

Bring Location Intelligence To Big Data Applications on Hadoop, Spark, and NoSQL

Siva Ravada
Senior Director of Development

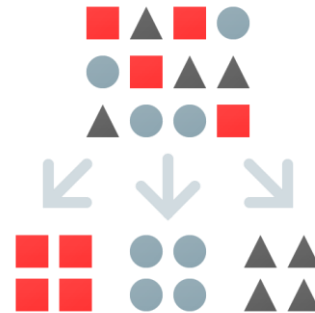
Program Agenda

- 1 Overview
- 2 Vector Data Processing
- 3 Raster Data Processing
- 4 Discussion

What problems can Big Data Spatial analysis address?



Data Harmonization using any location attribute (address, postal code, lat/long, placename, etc).



Categorization and filtering based on location and proximity



Preparation, validation and cleansing of Spatial and Raster data

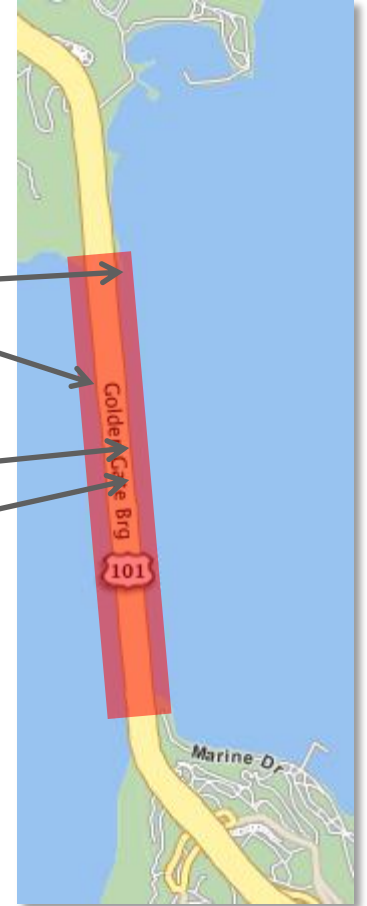


Visualizing and displaying results on a map

Data Harmonization: Linking information by location

Are these data points related?

- Tweet: sailing by #goldengate
- Instagram image subtitle: 골든게이트 교*
- Text message: Driving on 101 North , just reached border between Marin County and San Francisco County
- GPS Sensor: N 37°49'11" W 122°28'44"
- Now find all data points around Golden Gate Bridge ...

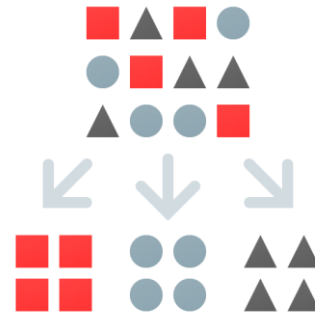


* Golden Gate Bridge (in Korean)

What features does Big Data Spatial have?



Data enrichment service API using GeoNames and geometry hierarchy data



MapReduce routines and for distance calculations, PointInPolygon, buffer creation, Categorization, KMeansClustering, binning, etc.. **Hive support (new)**



Spatial processing of data stored in HDFS. Raster processing operations: Mosaic and sub-set operations. Geodetic and Cartesian data



HTML5 Map Visualization API

Program Agenda

- 1 Overview
- 2 Vector Data Processing
- 3 Raster Data Processing
- 4 Discussion

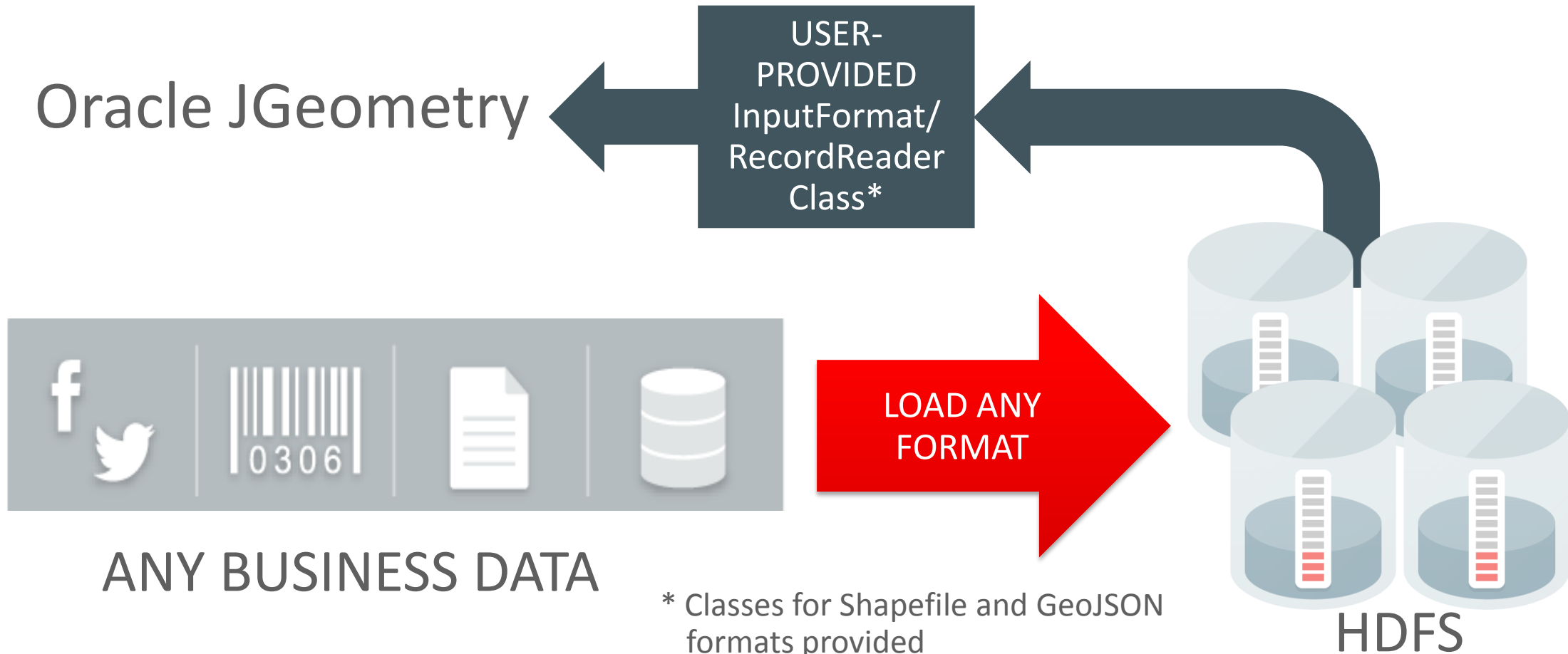
Vector Data Type support

- All of the data types we support with the SDO_GEOMETRY are supported on the Hadoop and Spark platform
 - Points, Lines, Polygons, Collections
 - Including Arcs, compound line strings, NURBs, compound polygons, etc.
 - Supports both 2D and 3D data types
 - Supports both Cartesian and Geodetic data models
 - Topological and distance operations
 - Anyinteract, inside, distance, length, simplify, buffer, PointInPolygon

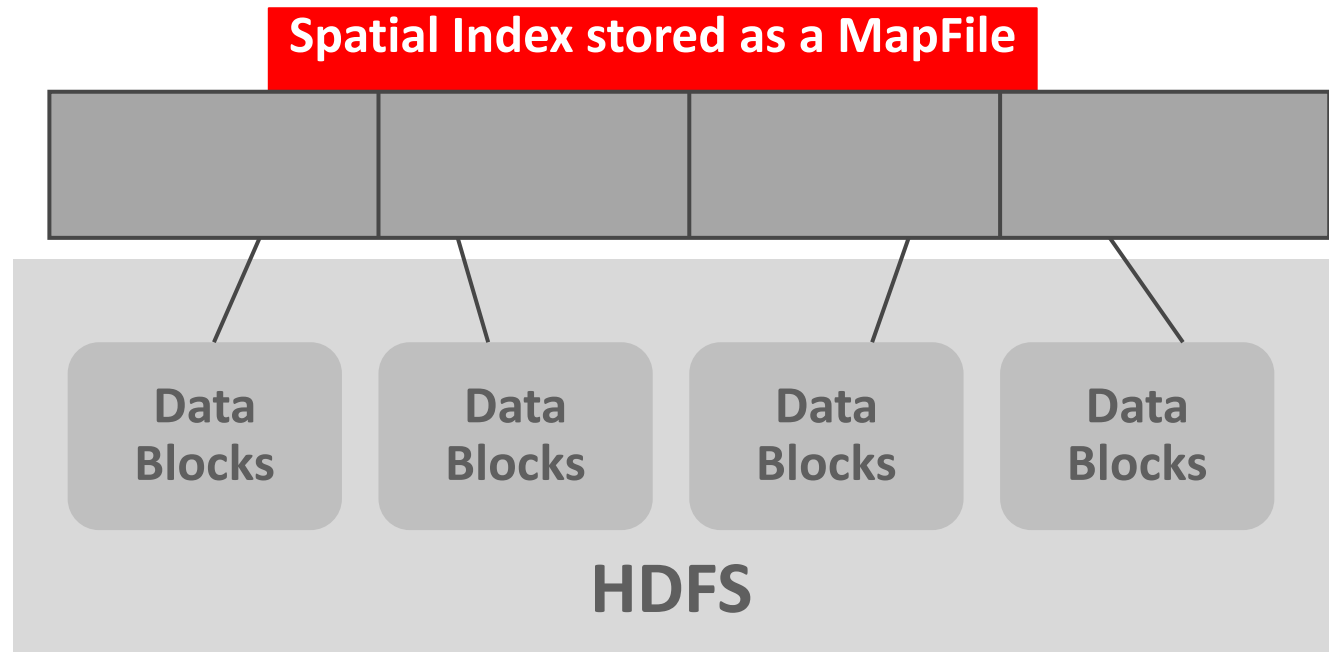
Vector spatial data storage in HDFS

- Customers load their data into HDFS using a loader of their choice
 - We do not require the data to be in a format that we specify
 - This makes it easy for customers to use any data format their applications prefer
 - And the data can have other business data and not just spatial data
- We require the customer to provide the InputFormat/RecordReader class
 - The RecordReader implementation reads the customer data record and produces an instance of JGeometry
 - With this model we can support any data format customers use for their data

Store any business data with spatial information in HDFS



Spatial Index for Spatial Queries



MapReduce Job with Index

Copy the index to distributed cache

Mapper reads the index data for the corresponding HDFS block

Process only those records that return hits from the index search

Vector Data Processing API

- Our API is broadly divided into three categories
 - Functions that operate on a single geometry at a time
 - Buffer, simplify, length, area, etc.
 - Functions that operate on a pair of geometries at a time
 - Range-Queries: these are the typical PointInPolygon, AnyInteract type of operations
 - Analogous to our operators and relate functions in the DB
 - Join-Queries: This is the typical Spatial-Join operation where two data sets are joined to find all the interacting pairs of geometries (next release)
 - Functions that categorize and enrich data
 - Associating a data set with a known geometry or named hierarchy
 - For example, process all tweets for a time period and count how many tweets are associated with each city, county, state, etc.

Vector Data Processing API Functions

Single Geometry

- Length
- Area
- Buffer
- Simplify

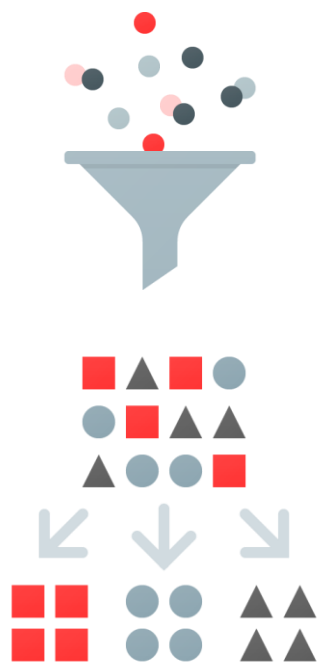
Geometry Pairs

- Range Queries
 - Point in Polygon
 - Touch, Overlap, Intersect, Contains, Any Interaction
- Join Queries
 - Interactions on sets of data
 - E.g.: Find all the dropped cell calls in all coverage areas

Categorization and Enrichment

- Associate a data set with a known geometry or named hierarchy
 - Process all Tweets for a period of time and count how many are associated with each city, county, state, etc.

Data Categorization Services



Any hierarchical geometry data set for reference

Customers choose a set of layers For example, they can select (continents, countries, cities) or (countries, states, counties) as the hierarchy

Big Data Spatial map-reduce job processes the customer data and produces a result file

Vector Data Processing: PointInPolygon operation

- Consider a large customer data set that is loaded into HDFS
 - We want to read each record, extract geometry information and check if the record is inside a given polygon geometry
- This can be run as a map-reduce job using our Java API from the Hadoop command line
 - Customer creates a map-reduce job and specifies the JGeometry.Inside() method for record processing
- Customers write their own map-reduce jobs using our APIs

Spatial Binning

Background

World Map ▾

Result Color



Apply color to results

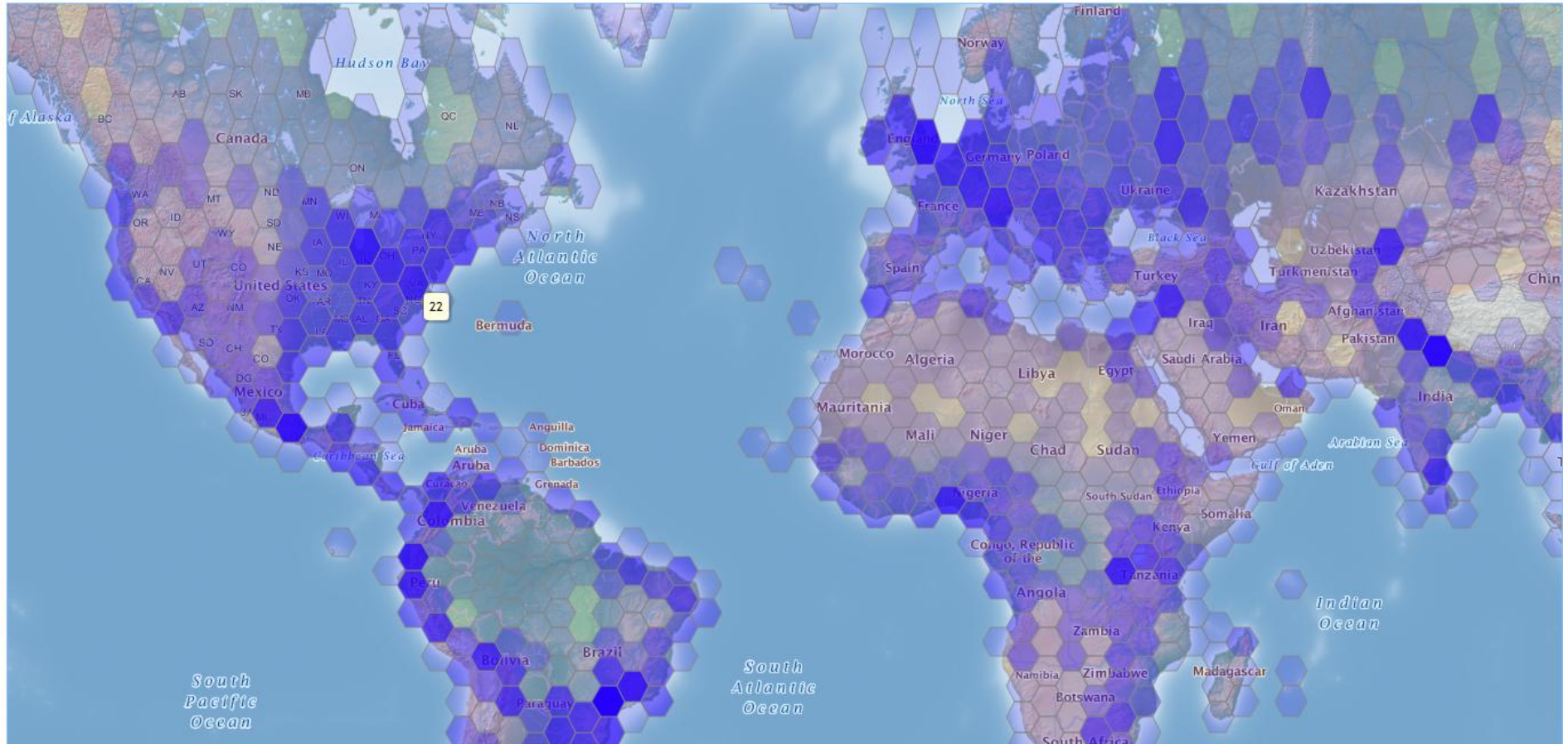


Results

[sample](#)

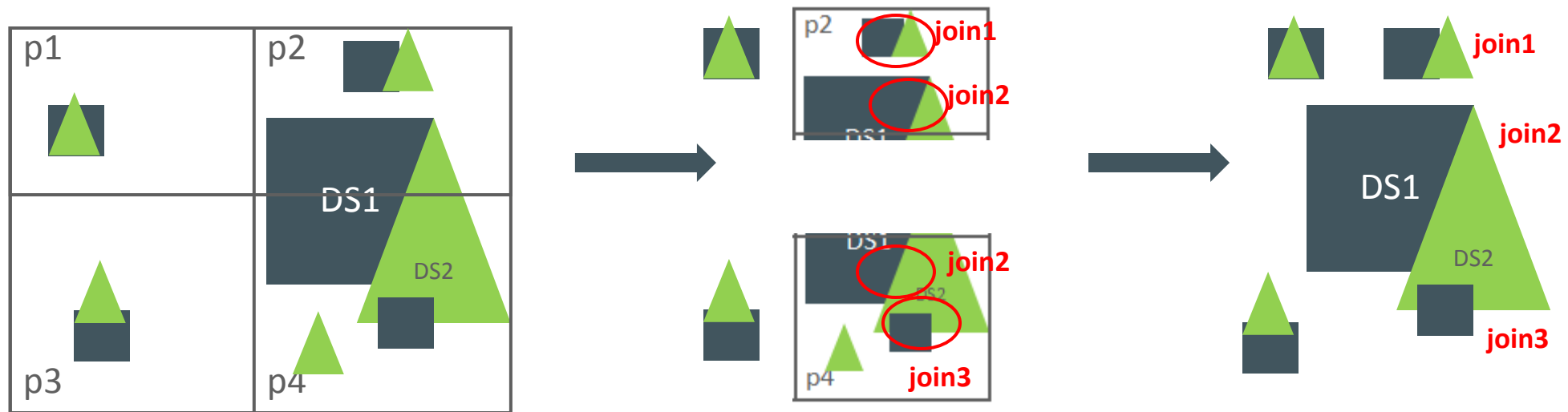
[Tweets January](#)

Tweets January (min: 1, max: 34)



Spatial Join Process

- The spatial join process consists of the following phases:
 1. Partitioning
 2. Join
 3. Duplicate removal

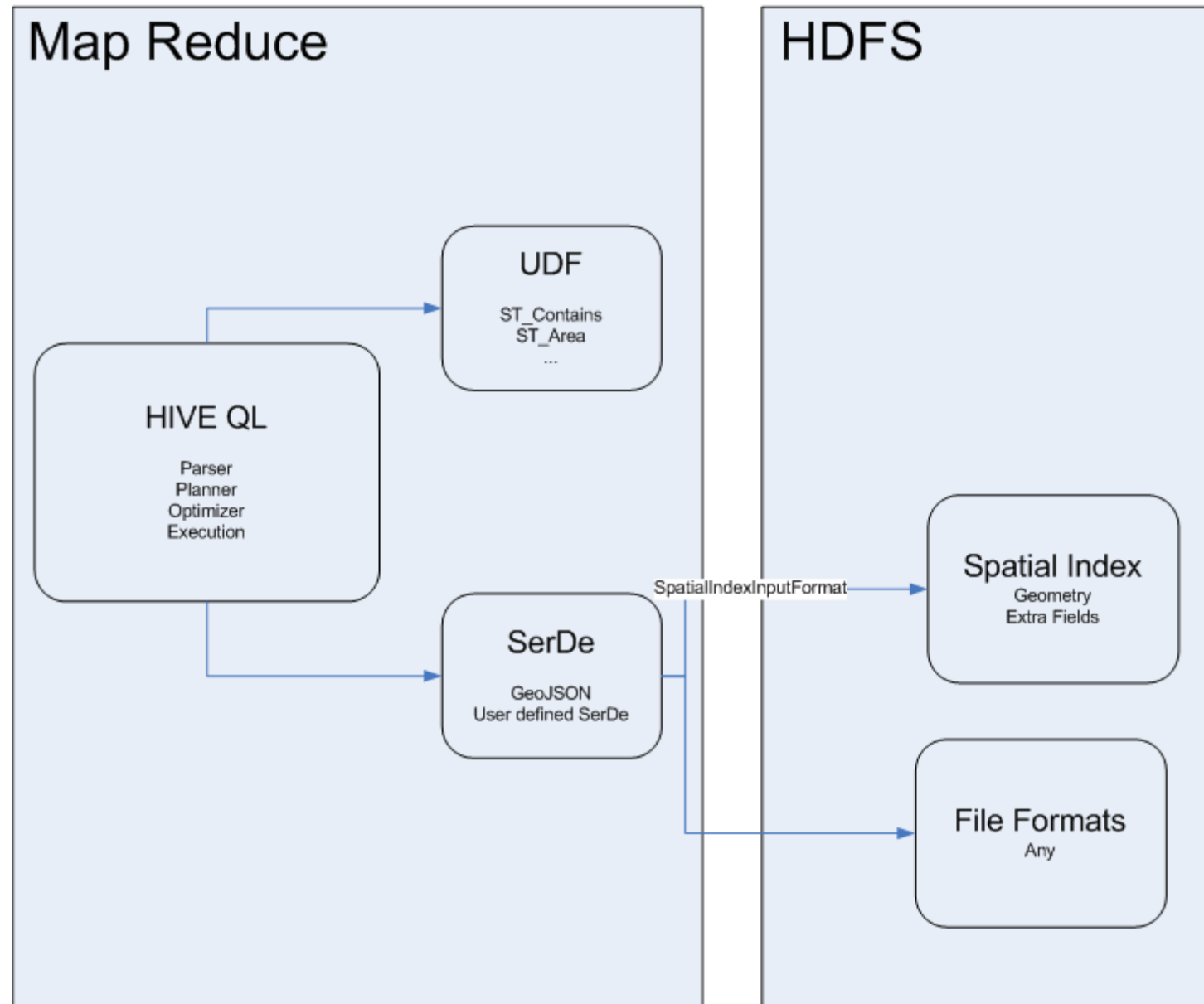


HIVE Support for Vector Data



- Provides support for standards based spatial data types
 - ST_GEOMETRY type with constructors
 - GeoJSON and WKT are natively supported
- Provides support for standard based spatial operators
 - ST_RELATE, ST_CONTAINS, ST_DISTANCE, ST_BUFFER, etc.
- ST_GEOMETRY has member methods to output geometry as text
 - AsJSON, AsWKT
- Provides spatial indexing capability
 - Create a Spatial Index with the java API and use it in HIVE queries

Analysis



Spatial – Hive integration

Table creation



```
CREATE EXTERNAL TABLE IF NOT EXISTS sample_tweets_index (id STRING, geometry
STRING, followers_count STRING, friends_count STRING, location STRING)
ROW FORMAT SERDE
'oracle.spatial.hadoop.vector.hive.json.GeoJsonSerDe'
STORED AS INPUTFORMAT
'oracle.spatial.hadoop.vector.mapred.input.SpatialIndexTextInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION '/user/oracle/twitter/data'
```

Spatial – Hive integration

HiveQL Query



```
SELECT id, followers_count, friends_count, location FROM sample_tweets
WHERE ST_Contains(
  ST_Polygon('{"type": "Polygon","coordinates": [[[-106, 25], [-106, 30],
    [-104, 30], [-104, 25], [-106, 25]]]}', 8307)
  , ST_Point(geometry, 8307)
  , 0.5)
and followers_count > 50;
```

Accessing BDSG data from the Database

- Oracle SQL Connectors for HDFS (OSCH)
 - Files in HDFS are delimited text files (fields must be delimited using single-character markers, such as commas or tabs)
 - Spatial data is stored as GeoJSON or WKT format
 - An external table is defined to access the data from HDFS
- Spatial operations can be performed from the database using SQL
- JOIN between a DB table: CINEMA and HDFS table: TWEETS_EXT_TAB_FILE

```
select sdo_geom.SDO_DISTANCE(ci.geometry, SDO_GEOMETRY(tw.geometry, 4326), 0.05, 'UNIT=MILE'),
       ci.name, tw.user_id
from CINEMA ci, TWEETS_EXT_TAB_FILE tw
where SDO_WITHIN_DISTANCE(ci.geometry, SDO_GEOMETRY(tw.geometry, 4326),
                        'DISTANCE=0.25 UNIT=MILE') = 'TRUE'
```

Accessing BDSG data from the Database

- Big Data SQL to access Spatial data in HDFS
 - Files in HDFS are delimited text files (fields must be delimited using single-character markers, such as commas or tabs) or custom formatted files
 - If Spatial data is stored as GeoJSON or WKT format, we provide the SerDe
 - Spatial data stored in custom format, customers provide the SerDe
 - An external table is defined to access the data from HDFS
- Spatial operations can be performed from the database using SQL
- JOIN between a DB table: CINEMA and HDFS table: TWEETS_EXT_TAB_FILE

```
select sdo_geom.SDO_DISTANCE(ci.geometry, SDO_GEOMETRY(tw.geometry, 4326), 0.05, 'UNIT=MILE'),  
       ci.name, tw.user_id  
from CINEMA ci, TWEETS_EXT_TAB_FILE tw  
where SDO_WITHIN_DISTANCE(ci.geometry, SDO_GEOMETRY(tw.geometry, 4326),  
                          'DISTANCE=0.25 UNIT=MILE') = 'TRUE'
```

NoSQL Support In Vector API

- A NoSQL data source can be used as the input data for a Vector API Job
- A NoSQL input can be specified to a Vector Job in one the following ways
 - Using the Oracle NoSQL Key-Value API Hadoop classes
 - Using the Oracle NoSQL Table API Hadoop classes
 - Using the special NoSQL classes provided by the Vector API

Spatial in Spark

Spatial Resilient Distributed Datasets (RDD)

- A Spatial RDD provides the regular RDD functionality plus spatial transformations and actions
- Current Spatial RDD implementations are SpatialJavaRDD and SpatialJavaPairRDD which are subclasses of JavaRDD and JavaPairRDD respectively
- Spatial transformations take a spatial predicate which can be used to spatially filter the RDD's data
- The following spatial operations can be used to perform a spatial transformations and actions
 - IsInside
 - Contains
 - AnyInteract
 - WithinDistance
 - MBR
 - Nearest Neighbors

SpatialJavaRDD Transformation Example

```
// load text file RDD
JavaRDD<String> csvRDD = sc.textFile(file);

// create a spatial RDD
SpatialJavaRDD<String> spatialCSV RDD = SpatialJavaRDD.fromJavaRDD(csvRDD, new CSVRecordInfoProvider(srid),
    String.class);

//Configure spatial operation
JGeometry qryWindow = JGeometry.createLinearPolygon(new double[] { 2, 1, 2, 3, 6, 3, 6, 1, 2, 1 }, 2, srid);
SpatialOperationConfig spatialOpConf = new SpatialOperationConfig(SpatialOperation.AnyInteract, qryWindow,
    0.0005);

//perform spatial filter transformation
SpatialJavaRDD<String> filteredSpatialSCVRDD = spatialCSV RDD.filter((record) -> {
    //filter function provided by the user to perform a non-spatial filter
    String[] tokens = record.split(",");
    String id = tokens[0];
    return Integer.parseInt(id) > 2;
} , spatialOpConf);
```

Distributed Spatial Index

- A Distributed Spatial Index is used to perform faster spatial searches over a Spatial
- A spatial index may redistribute and repartition an existing spatial RDD in order to speed up queries
- A local index can be built for each RDD's partition in order to further accelerate spatial searches
 - This feature is optional when creating a distributed spatial index.
- An existing Distributed Spatial RDD can be saved and loaded from a local or distributed file system using the save and load DistributedSpatialRDD's method
- The current Distributed Spatial Index implementation is based on QuadTree

HTML 5 API for MapVisualization

ORACLE®

Show Hadoop Results

Show Hadoop Results

Create index

Run Job

Templates

[USA Cities](#)

[USA Counties](#)

[USA States](#)

[World Cities](#)

[World Continents](#)

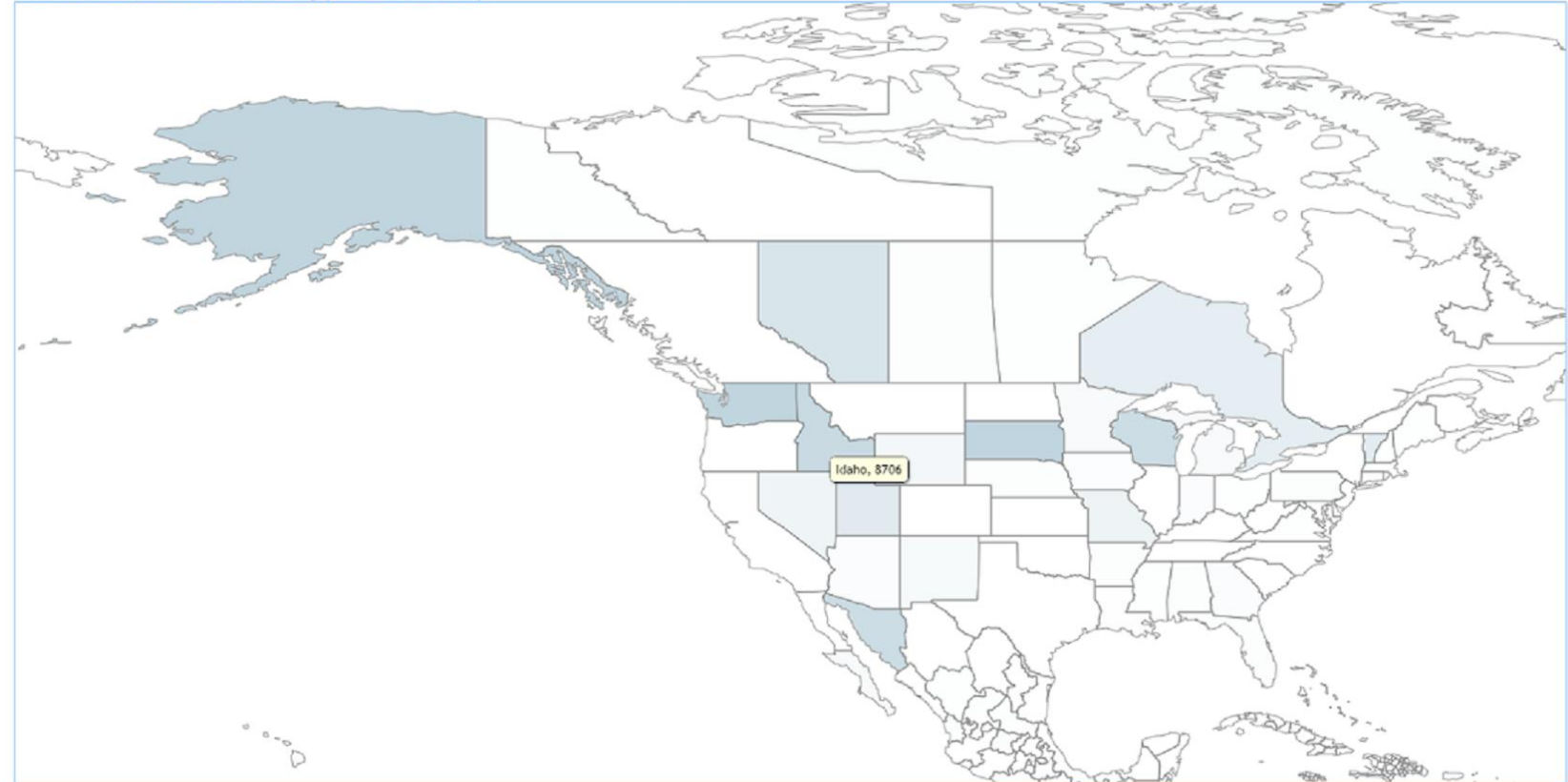
[World Countries](#)

[World State Provinces](#)

Results  

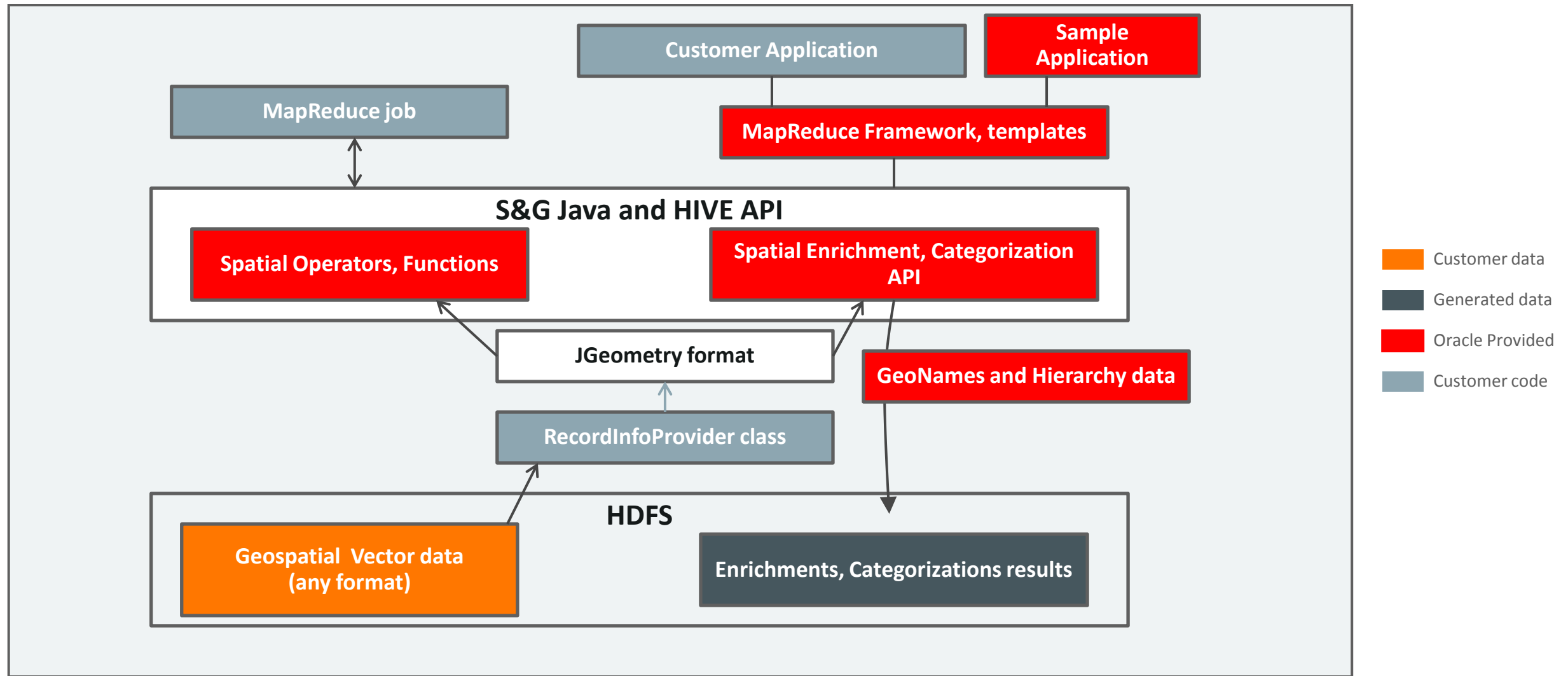
[Tweets of May](#)

World State Provinces - Tweets of May (min: 1, max: 34817)



Big Data Spatial and Graph

Spatial Vector Processing Framework



Program Agenda

- 1 Overview
- 2 Vector Data Processing
- 3 Raster Data Processing
- 4 Discussion

Image Server

- HDFS storage for the image or raster files
 - We can support dozens of file formats (GDAL supported formats)
 - Images are geo-referenced
 - Images can be in different coordinate systems and resolutions
- Three main capabilities
 - Loader to load raster data from NFS to HDFS
 - Mosaic and subset operations based on a virtual mosaic
 - Image processing framework for raster analysis

Image Server Loader

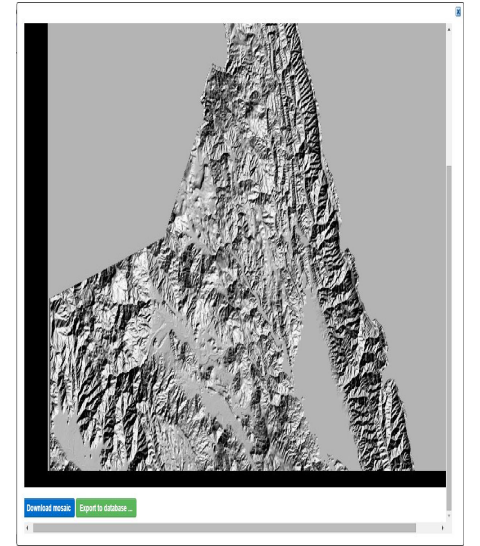
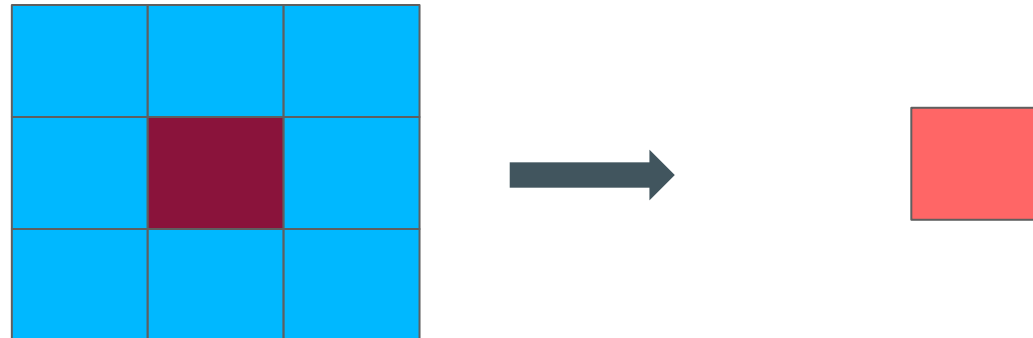
- Customers usually have large volumes of raster data in traditional file systems
 - We provide a GDAL based loader to load the data into HDFS such that the resulting HDFS blocks are organized for map-reduce jobs
 - Many formats supported by GDAL are supported
- In V1, only support 3 band and single band images with float and byte data types
- In V2 added support for multi-band images

Raster Data Stored on HDFS

- When data is moved from traditional file system to HDFS, data should be organized such that a map-reduce job can process it with minimum amount of data transfer between data nodes
- We explain this concept with a raster data analysis example
- Raster Analysis with Hadoop
 - Given a DEM, find the shaded relief model from it
 - Storage Models considered
 - Pixel data is partitioned over different HDFS blocks
 - What are the best options for storing the DEM data on HDFS to enable this type of raster processing ?

Shaded Relief calculation

- Input: NXM pixels where each pixel is a floating point number denoting elevation
- Find the shaded relief from the DEM
- Algorithm
 - Look at the values of 8 neighbors and the current pixel value and generate a new pixel
 - Needs the neighboring pixel values to calculate the new pixel value corresponding to the current pixel



Raster Data Analysis Framework

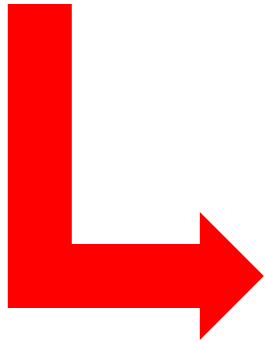
- Traditional algorithm will work very well in Map-Reduce framework
- Customers specify the pixel overlap required while loading the data into HDFS
- Once the data is loaded according to this overlap, rest of the processing can be done using standard algorithms
- Very effective for raster data processing as many map-reduce nodes can work together to produce the result in a short amount of time
- All the operations are performed on a catalogue of images
 - Customers can logically combine certain number of images into a catalogue or into a virtual mosaic

Image Server Features

- Subset operation
 - Find the set of images from a given catalogue covering a user specified region and generate a new image from the source files
 - The new images will have user specified resolution and coordinate system
 - These can be different for different images in the source catalogue and the resulting image can have a different value
 - Mosaic the input images to deal with gaps and overlaps
 - Create a new file with the specified file format
- Image Processing Functions

Raster Processing

- Local Map algebra operations



localnot	localif	localadd	localsubtract	localmultiply	
localdivide	localpow	localsqrt	localround	locallog	
locallog10	localfloor	localceil	localnegate	localabs	
localsin	localcos	localtan	localsinh	localcosh	localtanh
localasin	localacos	localatan	localdefined	localundefined	

Shaded Relief Generation from DEM

- Hypothetical illumination of a surface by determining illumination values for each pixel
- Custom shaded relief algorithm can be plugged into our framework
- Users need to implement a few classes
 - ImageProcessorInterface
 - write results back in the ImageBandWritable data type

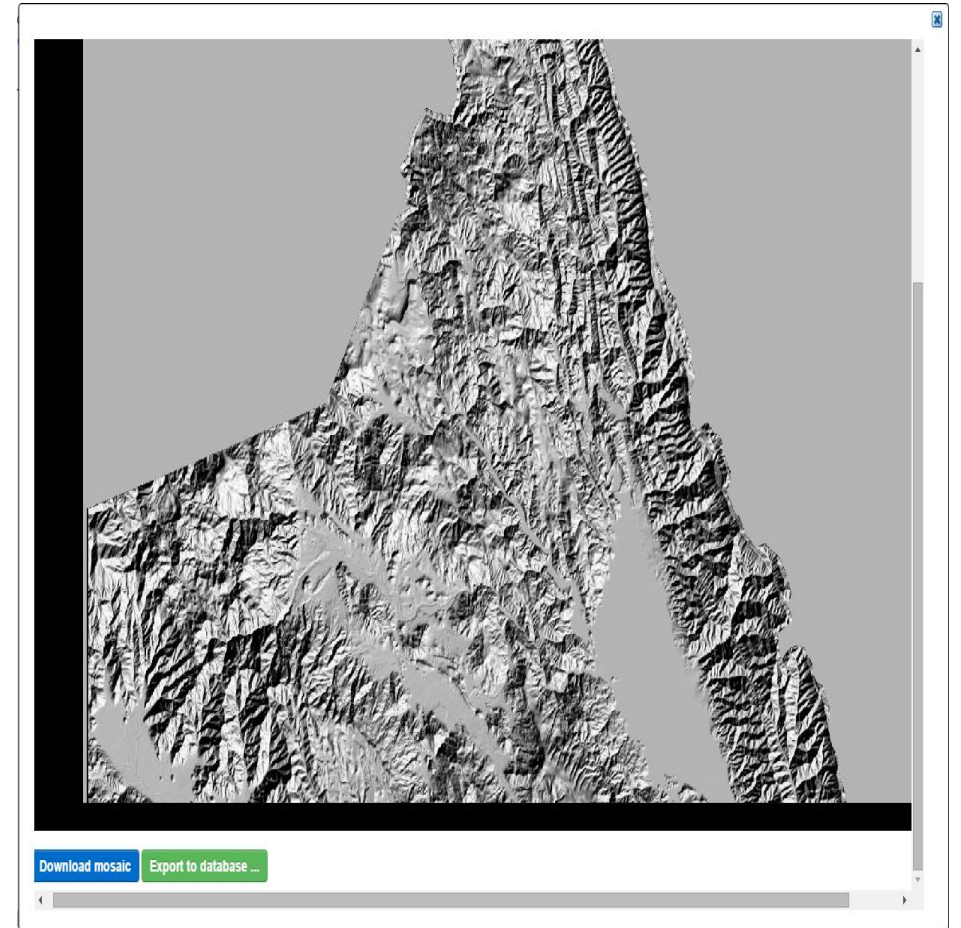


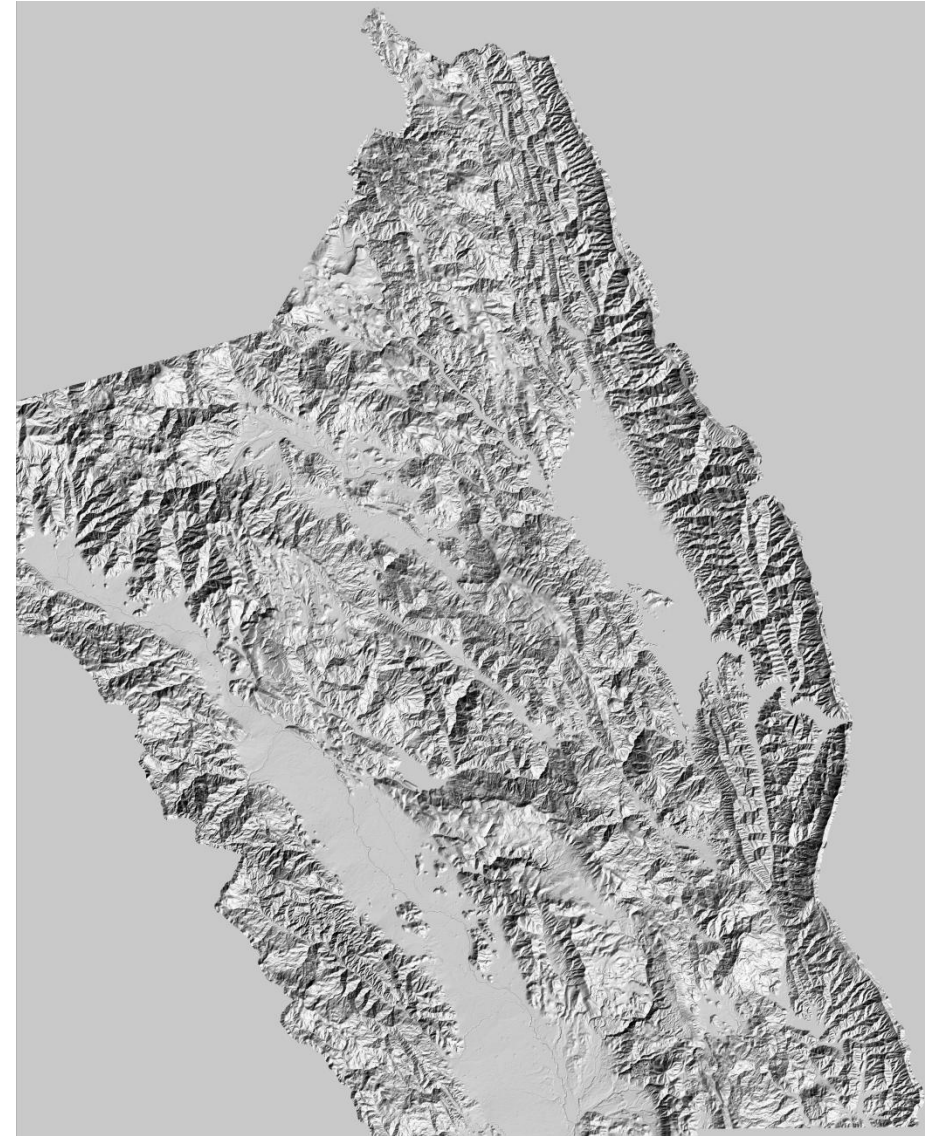
Image Processing on Hadoop

- This framework allows for complex image processing on raster data
- In this example we describe how a DEM can be processed to generate a shaded relief in addition to doing some raster algebra operations
- In this demo, we start with a DEM for the NAPA valley area



Image Processing on Hadoop

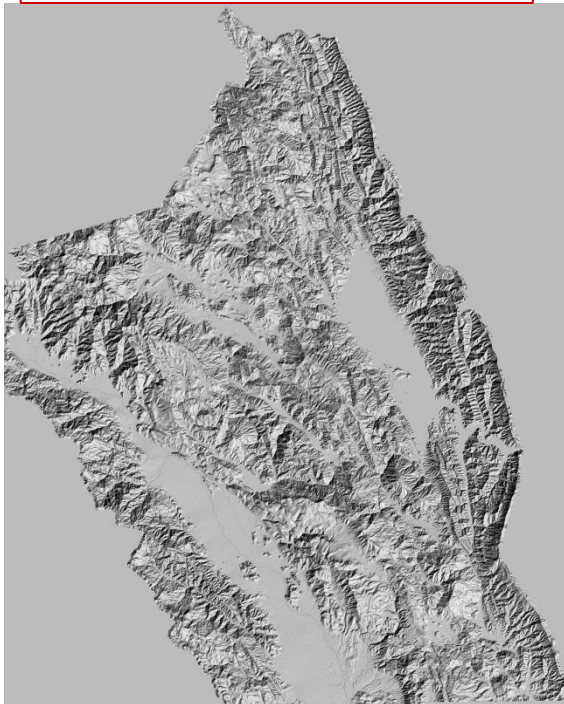
- The DEM can be processed using custom algorithms to generate a hillshade
- The whole process will execute as a map reduce program
- Users can take a non map-reduce custom algorithm and plug it into the framework



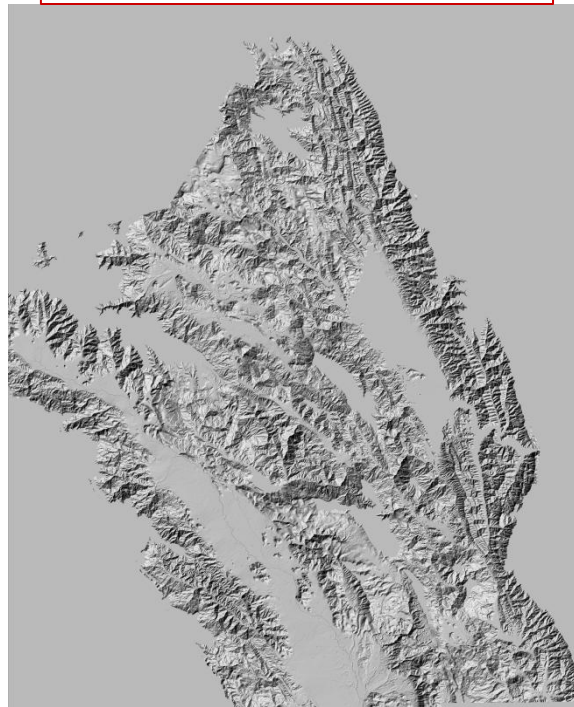
Raster Algebra on Hadoop

- We also support pixel level raster algebra operations on this raster data
- We can take the DEM and generate different versions of hillshaded rasters by selectively removing pixels with certain value

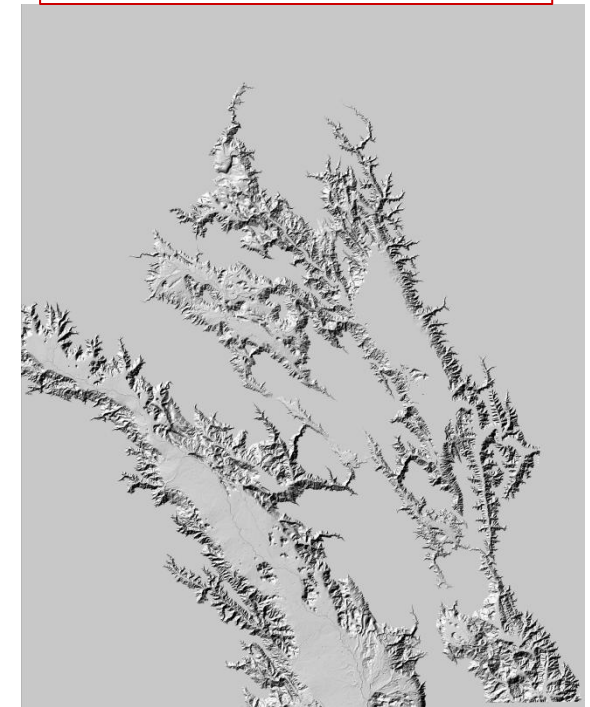
Elevation>1000m



Elevation>1800m

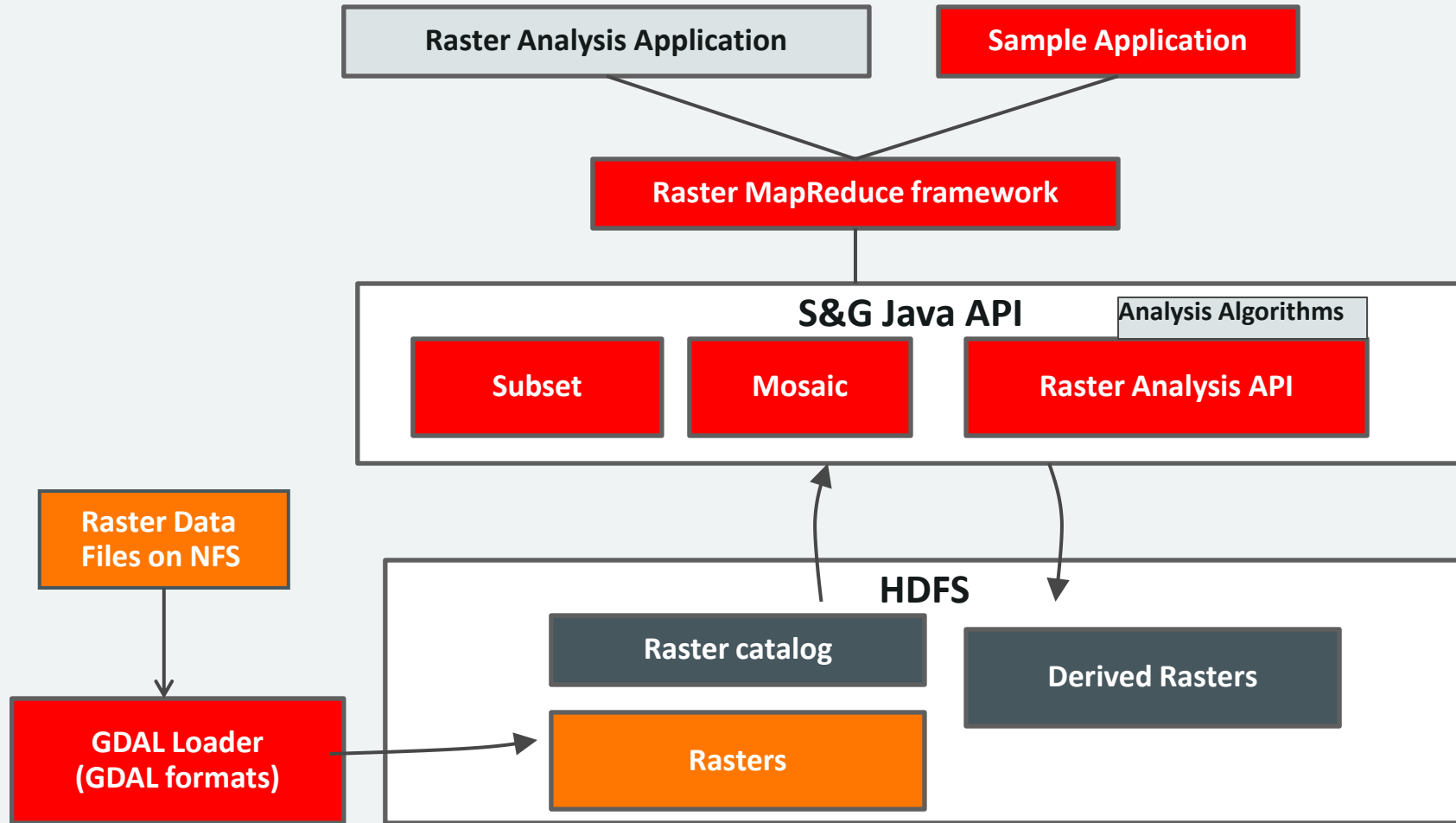


Elevation>2800m



Big Data Spatial and Graph

Spatial Raster Processing Framework




- Customer data
- Generated data
- Oracle Provided
- Customer code

Program Agenda

- 1 Overview
- 2 Vector Data Processing
- 3 Raster Data Processing
- 4 Discussion



Resources

- Oracle Big Data Spatial and Graph OTN product page: www.oracle.com/technetwork/database/database-technologies/bigdata-spatialandgraph
– White papers, software downloads, documentation and videos
- Oracle Big Data Lite Virtual Machine - a free sandbox to get started: www.oracle.com/technetwork/database/bigdata-appliance/oracle-bigdatalite-2104726.html
- Hands On Lab for Big Data Spatial: tinyurl.com/BDSG-HOL
- Blog – examples, articles & tips: blogs.oracle.com/bigdataspatialgraph
- Oracle By Example tutorials: www.oracle.com/goto/oll
(search “Big Data Spatial and Graph”)
-  @OracleBigData, @SpatialHannes, @JeanIhm  Oracle Spatial and Graph Group