Oracle Big Data Spatial and Graph: Spatial Features
January 28th, 2016

Siva Ravada
Senior Director of Development
Program Agenda

1. Overview
2. Vector Data Processing
3. Raster Data Processing
4. Discussion
Motivation for Oracle Big Data Spatial

Emergence of Hadoop for spatial analysis in business and spatial workflows

Existing Hadoop-based Spatial technologies are GeoSpatial-centric not Application-centric

No significant commercial offerings
What problems can Big Data Spatial analysis address?

- Data Harmonization using any location attribute (address, postal code, lat/long, placename, etc).
- Preparation, validation and cleansing of Spatial and Raster data.
- Categorization and filtering based on location and proximity.
- 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)
Use Case: Linking Information by Location

Insurance Industry

Of Insurance companies agree that analyzing multiple data sources together is crucial to making accurate predictions

86% Agree that linking information by location is key to combining disparate sources of Big Data

88%

Source: “The big data: How data analytics can yield underwriting gold.” Survey conducted by Ordnance Survey and Chartered Insurance Institute, 25 April 2013.
<table>
<thead>
<tr>
<th>What features does Big Data Spatial have?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data enrichment service API</strong> using GeoNames and geometry hierarchy data</td>
</tr>
<tr>
<td><strong>Spatial processing of data stored in HDFS. Raster processing operations:</strong> Mosaic and sub-set operations. Geodetic and Cartesian data</td>
</tr>
<tr>
<td><strong>MapReduce routines for distance calculations, PointInPolygon, buffer creation, Categorization, KMeansClustering, Binning, etc.</strong></td>
</tr>
<tr>
<td><strong>HTML5 Map Visualization API</strong></td>
</tr>
</tbody>
</table>
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 JGeometry class are supported on the Hadoop 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
    - Any interact, 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
Loading Vector Data

hadoop fs -put /opt/oracle/oracle-spatial-graph/spatial/vector/HOL/data/tweets.json
/user/oracle/HOL/tweets.json

hadoop fs -put /opt/oracle/oracle-spatial-
  graph/spatial/vector/HOL/data/USA_2012Q4_PCB3_PLY.dbf
/user/oracle/HOL/USA_2012Q4_PCB3_PLY.dbf

hadoop fs -put /opt/oracle/oracle-spatial-
  graph/spatial/vector/HOL/data/USA_2012Q4_PCB3_PLY.shp
/user/oracle/HOL/USA_2012Q4_PCB3_PLY.shp

hadoop fs -put /opt/oracle/oracle-spatial-
  graph/spatial/vector/HOL/data/USA_2012Q4_PCB3_PLY.shx
/user/oracle/HOL/USA_2012Q4_PCB3_PLY.shx
Store any business data with spatial information in HDFS

Oracle JGeometry

ANY BUSINESS DATA

HDFS

LOAD ANY FORMAT

USER-PROVIDED InputFormat/RecordReader Class
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: creating buffers on line geometry

• Consider a large line geometry data set that is loaded into HDFS
  – We want to read each line geometry, create a buffer around it using a specified distance value and store the resulting geometries back in HDFS

• 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.buffer() method for record processing
  – And the reducer stores the results back into HDFS

• Customers write their own map-reduce jobs using our APIs
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
Spatial Index for Spatial Queries

Spatial Index stored as a MapFile

HDFS

Data Blocks

Data Blocks

Data Blocks

Data Blocks

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
Spatial Server Console

• A sample J2EE application that can be deployed in Jetty
  – Create spatial indexes on data that is already loaded into HDFS
  – Run Hadoop jobs to do spatial processing
    • This will create and run the categorization MapReduce job that can use a provided spatial index or read each input record
  – Sample map application that can display results of a categorization job
Spatial Server Console

**New Index**

- Path of the data to index: `hdfs://hadoop.cluster.url:8020/user/myuser/twitter_datas`
- New index path: `hdfs://hadoop.cluster.url:8020/user/myuser/twitter_index`
- JAR with user classes: `userlibsamiple.jar`
- Input Format class: `org.apache.hadoop.mapred.TextInputFormat`
- Record Info Provider class: `mypackage.SampleTwitterLogRecordInfoProvider`
Spatial Server Console

Show Hadoop Results | Create Index | Run Job

New Job

With index
- *Index path: hdfs://hadoop.cluster.url:8020/user/myuser/twitter_index/part*/data
- *SRID: 8307
- *Tolerance: 0.5
- *Geodetic: Yes

Without index
- *Path of the data:
- JAR with user classes:
- *Input Format class:
- *Record Info Provider class:
- *MVsuggest service URL:

*Templates: World Continents, World Countries, World State Provinces
*Output path: hdfs://hadoop.cluster.url:8020/user/myuser/result/output
*Result name: Tweets of May
Outcome Notification email sent to: firstname.lastname@myemail.com

Create
Spatial Server Console

[Map of the United States with state boundaries highlighted in blue]
Big Data Spatial and Graph
Spatial Vector Processing Framework

- MapReduce job
- Spatial Operators, Functions
- Spatial Enrichment, Categorization API
- JGeometry format
- GeoNames and Hierarchy data
- RecordInfoProvider class
- Spatial Vector data (any format)
- Enrichments, Categorizations results
- Customer Application
- Sample Application
- MapReduce Framework, templates

- HDFS

- Customer data
- Generated data
- Oracle Provided
- Customer code
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
Loading Raster Data

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
Shaded Relief calculation

• Consider a 9X9 pixel DEM
• Split this image into blocks so that each bock is 3X3 pixels
• Write each block of cells to a HDFS block
Shaded Relief calculation

• We extend the original image with imaginary cells to make it a 11X11 image
  – These are imaginary cells with NODATA elevation

• Split this image into blocks so that each block is 5X5 pixels
  – When we write the HDFS blocks we add these additional pixels as physical pixels with NODATA elevation value

• This will make the processing phase later much easier as each block can be processed in the same way

• This will increase the number of blocks after the splits, but this increase in storage is acceptable as it makes processing easier
HDFS Splits

• Each split will have 5X5 pixels
• The middle 3X3 are the real pixels (red) in the block, and the rest are boundary pixels (orange)
  – shaded relief calculation is done only for the red pixels in each block
• Then each real pixel is processed by only one mapper
• And in reduce phase, there is only one new value for each real 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
Raster Processing

• Local Map algebra operations

```plaintext
localnot localif localadd localsubtract localmultiply
localdivide localpow localsqrt localround locallog
locallog10 localfloor localceil localnegate localabs
localsin localcos localatan 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 Server Console

• A sample J2EE application that can be deployed in Jetty
  – Load data into HDFS from NFS
  – Create catalogues from existing images on HDFS
  – Run Hadoop jobs to do subset operations
    • This will create and run the map-reduce job to the specified subset operation including changing resolution, changing coordinate system, etc.
  – Run Hadoop jobs to do raster analysis
    • This will create and run the map-reduce job to the specified raster analysis operations
    • Users will need to specify the java class that is used to process the pixels and produce new pixel values for the output image
Image Server Console
Image Server Console
Big Data Spatial and Graph
Spatial Raster Processing Framework

- Raster Analysis Application
- Sample Application
- Raster MapReduce framework
- S&G Java API
  - Subset
  - Mosaic
  - Raster Analysis API
- Analysis Algorithms
- HDFS
  - Raster catalog
  - Derived Rasters
  - Rasters
- GDAL Loader (GDAL formats)
- Raster Data Files on NFS
Program Agenda

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