Crunching and visualizing Big Data

on a Computer Cluster

Joana Simões

March 29, 2015

Table of Contents

From S3 to HDFS

- Micro-task: Understand the dataset structure
  - A sample dataset is stored on an S3 bucket: `s3n://workshop-bdsd/accidents/`

From S3 to HDFS

- Micro-task: Understand the dataset structure
  - A sample dataset is stored on an S3 bucket: `s3n://workshop-bdsd/accidents/`
  - **`https://s3-eu-west-1.amazonaws.com/workshop-bdsd/accidents/accidents_sample.csv`**

From S3 to HDFS

- Micro-task: Understand the dataset structure
  - A sample dataset is stored on an S3 bucket: `s3n://workshop-bdsd/accidents/`
  - **`https://s3-eu-west-1.amazonaws.com/workshop-bdsd/accidents/accidents_sample.csv`**
  - Download and view dataset

From S3 to HDFS (cont.)

- Micro-task: Create a table linking to the data
    - Enter hive and create an external table linking to the S3 bucket

From S3 to HDFS (cont.)

- Micro-task: Create a table linking to the data
  - Enter hive and create an external table linking to the S3 bucket
  - Use the CSV serde to parse the table structure

From S3 to HDFS (cont.)

- Micro-task: Create a table linking to the data
  - Enter hive and create an external table linking to the S3 bucket
  - Use the CSV serde to parse the table structure
    - separator char
    - quote char
    - headers

From S3 to HDFS (cont.)

- Micro-task: Create a table linking to the data
    - Enter hive and create an external table linking to the S3 bucket
    - Use the CSV serde to parse the table structure
        - separator char
        - quote char
        - headers
    - View imported data

From S3 to HDFS (cont.)

- Micro-task: Type Mapping

From S3 to HDFS (cont.)

- Micro-task: Type Mapping
  - Create an empty table with correct types

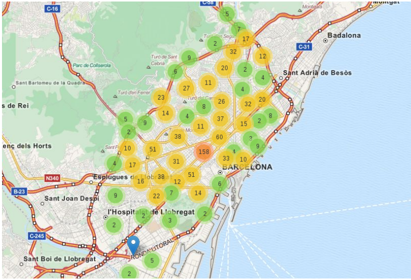
From S3 to HDFS (cont.)

- Micro-task: Type Mapping
  - Create an empty table with correct types
  - Insert data from accidents_import

From S3 to HDFS (cont.)

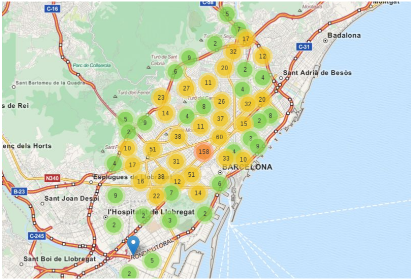
- Micro-task: Type Mapping
  - Create an empty table with correct types
  - Insert data from accidents_import
  - View table

What is so "Special" about Spatial



What is so "Special" about Spatial

- Location attributes allow us to detect spatial patterns
- Location also works as a "key", allowing us to connect with other datasets



Analysis of the Spatial Attributes

- Spatial Attributes are encoded as coordinates in ”d_coord_geo_impacte”

Analysis of the Spatial Attributes

- Spatial Attributes are encoded as coordinates in ”d_coord_geo_impacte”
- Problems with this field:

Analysis of the Spatial Attributes

- Spatial Attributes are encoded as coordinates in ”d_coord_geo_impacte”
- Problems with this field:
    - Unstructured: needs parsing
    - Inconsistent format (order of coordinates, separator)
    - Invalid values (e.g.: 77, names)
    - No metadata

Analysis of the Spatial Attributes

- Spatial Attributes are encoded as coordinates in "d_coord_geo_impacte"
- Problems with this field:
    - Unstructured: needs parsing
    - Inconsistent format (order of coordinates, separator)
    - Invalid values (e.g.: 77, names)
    - No metadata
    - Mixed CRS:

Analysis of the Spatial Attributes

- Spatial Attributes are encoded as coordinates in ”d_coord_geo_impacte”
- Problems with this field:
  - Unstructured: needs parsing
  - Inconsistent format (order of coordinates, separator)
  - Invalid values (e.g.: 77, names)
  - No metadata
  - Mixed CRS:
    - WGS84 (EPSG:4326)
    - UTM Grid (EPSG:5554)
    - UTM Grid encoded by the police using an *ad-hoc* format

Analysis of the Spatial Attributes

- Spatial Attributes are encoded as coordinates in "d_coord_geo_impacte"
- Problems with this field:
  - Unstructured: needs parsing
  - Inconsistent format (order of coordinates, separator)
  - Invalid values (e.g.: 77, names)
  - No metadata
  - Mixed CRS:
    - WGS84 (EPSG:4326)
    - UTM Grid (EPSG:5554)
    - UTM Grid encoded by the police using an *ad-hoc* format
    - $lon = y/1000 + 400000$
    - $lat = y/1000 + 4500000$

Objective

- Separate lat, long fields and map them to correct types
- Remove invalid values
- Convert all coordinates into a single CRS (WGS84)

Exporting Data to Pig

- Pig uses filters to subset the data
- To merge back the subsetted data, we can use joins by a common field
- Micro-task: Export the data

Exporting Data to Pig

- Pig uses filters to subset the data
- To merge back the subsetted data, we can use joins by a common field
- Micro-task: Export the data
    - Create a copy of the accidents table, with an id field (joins).

Exporting Data to Pig

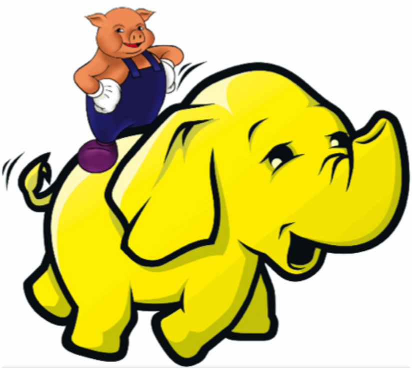
- Pig uses filters to subset the data
- To merge back the subsetted data, we can use joins by a common field
- Micro-task: Export the data
  - Create a copy of the accidents table, with an id field (joins).
  - Export this table into a tsv

Exporting Data to Pig

- Pig uses filters to subset the data
- To merge back the subsetted data, we can use joins by a common field
- Micro-task: Export the data
    - Create a copy of the accidents table, with an id field (joins).
    - Export this table into a tsv
    - Store it in HDFS (if needed)
    - View exported data

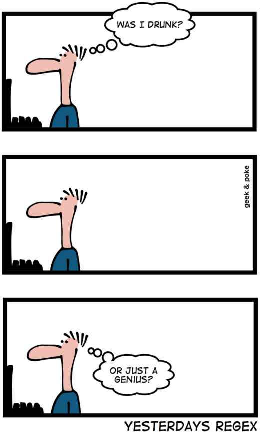
Presenting the Pig Script

- Subsets the coordinate list, using filters
- Detects each coordinate "type", using regular expressions
- In the case of UTM encoded, it applies a formula to decode back into UTM
- Stores the results into separate files, in HDFS



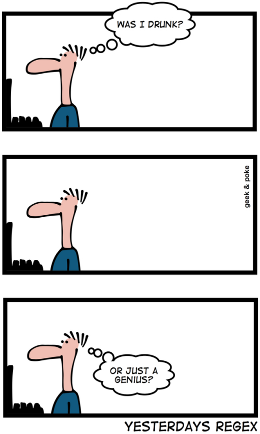
REGEX

- Sequence of characters that forms a search pattern, mainly for use in pattern matching with strings, or string matching



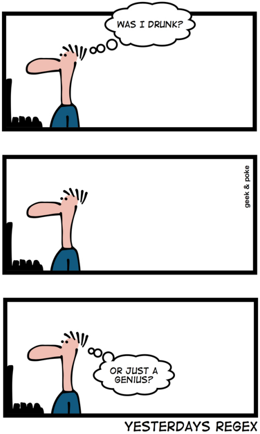
REGEX

- Sequence of characters that forms a search pattern, mainly for use in pattern matching with strings, or string matching
- REGEX_EXTRACT(D_COORD_C
  z]',0)



REGEX

- Sequence of characters that forms a search pattern, mainly for use in pattern matching with strings, or string matching
- REGEX_EXTRACT(D_COORD_(z]',0)
- '[A-z]'



Running Pig

- Micro-task: run pig script

Running Pig

- Micro-task: run pig script
  - Download script from S3: `https://s3-eu-west-1.amazonaws.com/workshop-bdsd/recover_geography.pig`

Running Pig

- Micro-task: run pig script
    - Download script from S3: `https://s3-eu-west-1.amazonaws.com/workshop-bdsd/recover_geography.pig`
    - Edit script and **ammend paths**

Running Pig

- Micro-task: run pig script
  - Download script from S3: `https://s3-eu-west-1.amazonaws.com/workshop-bdsd/recover_geography.pig`
  - Edit script and **ammend paths**
  - Run script

Running Pig

- Micro-task: run pig script
  - Download script from S3: `https://s3-eu-west-1.amazonaws.com/workshop-bdsd/recover_geography.pig`
  - Edit script and **ammend paths**
  - Run script
  - Check output files

Importing Data Back into Hive

- Micro-task: Create tables linking to pig output

Importing Data Back into Hive

- Micro-task: Create tables linking to pig output
  - Create table with wgs84 data
  - Create table with UTM data
  - Create table with police-decoded data

Exporting data into PostGIS

- As of Hadoop GIS 2.0, CRS transformation is **not supported**
- We need to rely on another tool: PostGIS on RDS
- Micro-task: Export UTM data to PostGIS

Exporting data into PostGIS

- As of Hadoop GIS 2.0, CRS transformation is **not supported**
- We need to rely on another tool: PostGIS on RDS
- Micro-task: Export UTM data to PostGIS
    - Merge UTM (UTM + police) tables in a single table

Exporting data into PostGIS

- As of Hadoop GIS 2.0, CRS transformation is **not supported**
- We need to rely on another tool: PostGIS on RDS
- Micro-task: Export UTM data to PostGIS
    - Merge UTM (UTM + police) tables in a single table
    - Exported merged table into TSV

Importing Data into PostGIS

- Micro-task: Import UTM data into PostGIS

Importing Data into PostGIS

- Micro-task: Import UTM data into PostGIS
  - Install the PSQL client
  - Log into RDS:

Importing Data into PostGIS

- Micro-task: Import UTM data into PostGIS
  - Install the PSQL client
  - Log into RDS:
    - host: bdigitaldb.celqzuwfokoe.eu-west-1.rds.amazonaws.com
    - user: workshop
    - password: geohipster
    - database: workshop_bdigital

Importing Data into PostGIS

- Micro-task: Import UTM data into PostGIS
  - Install the PSQL client
  - Log into RDS:
    - host: bdigitaldb.celqzuwfokoe.eu-west-1.rds.amazonaws.com
    - user: workshop
    - password: geohipster
    - database: workshop_bdigital
  - Create table to accomodate data

Importing Data into PostGIS

- Micro-task: Import UTM data into PostGIS
  - Install the PSQL client
  - Log into RDS:
    - host: bdigitaldb.celqzuwfokoe.eu-west-1.rds.amazonaws.com
    - user: workshop
    - password: geohipster
    - database: workshop_bdigital
  - Create table to accomodate data
  - Copy data into table

CRS Transformation

- Micro-task: Convert all features in UTM grid to WGS84

CRS Transformation

- Micro-task: Convert all features in UTM grid to WGS84
    - Create geometry fields to accomodate geometry in the two CRS (UTM, WGS84)

CRS Transformation

- Micro-task: Convert all features in UTM grid to WGS84
  - Create geometry fields to accomodate geometry in the two CRS (UTM, WGS84)
    - Add columns
    - Set SRID
    - Create geometry index

CRS Transformation

- Micro-task: Convert all features in UTM grid to WGS84
  - Create geometry fields to accomodate geometry in the two CRS (UTM, WGS84)
    - Add columns
    - Set SRID
    - Create geometry index
  - Instantiate UTM geometry

CRS Transformation

- Micro-task: Convert all features in UTM grid to WGS84
  - Create geometry fields to accomodate geometry in the two CRS (UTM, WGS84)
    - Add columns
    - Set SRID
    - Create geometry index
  - Instantiate UTM geometry
  - Transform UTM geometry into another CRS

CRS Transformation

- Micro-task: Convert all features in UTM grid to WGS84
    - Create geometry fields to accomodate geometry in the two CRS (UTM, WGS84)
        - Add columns
        - Set SRID
        - Create geometry index
    - Instantiate UTM geometry
    - Transform UTM geometry into another CRS
    - Export projected geometry in GeoJSON

Importing Data back into Hive

- Micro-task: Import transformed data

Importing Data back into Hive

- Micro-task: Import transformed data
    - Enter Hive

Importing Data back into Hive

- Micro-task: Import transformed data
  - Enter Hive
  - Create table linking to the PostGIS export

Importing Data back into Hive

- Micro-task: Import transformed data
  - Enter Hive
  - Create table linking to the PostGIS export
  - Create new table and instantiate geometry from GeoJSON

Joining Data

- Micro-task: Join imported coordinates with WGS84 coordinates and the rest of the dataset

Joining Data

- Micro-task: Join imported coordinates with WGS84 coordinates and the rest of the dataset
  - Join imported records with original table with all fields

Joining Data

- Micro-task: Join imported coordinates with WGS84 coordinates and the rest of the dataset
  - Join imported records with original table with all fields
  - Merge imported records with WGS84 records, for a single table with unified geometry

References

- http://tweettracker.fulton.asu.edu/tda/TwitterDataAnalytics.pdf
- http://www2.qgis.org
- http://plugins.qgis.org/plugins/
- http://geokoder.com/mongodb-plugin-for-quantum-gis
- http://www.gislounge.com/heat-maps-in-gis/
- https://alastaira.wordpress.com/2011/02/23/heat-mapping-crime-data-with-bing-maps-and-html5-canvas/
- http://docs.qgis.org/2.0/en/docs/user_manual/plugins/plugins_heatmap.html
- http://en.wikipedia.org/wiki/Kernel_%28statistics%29#Kernel_functions_in_common_use
- http://en.wikipedia.org/wiki/Cluster_analysis
- https://plugins.qgis.org/plugins/clusterpy_qgis_plugin/
- http://www.rise-group.org/section/Software/clusterPy/
- http://threejs.org/
- http://anitagraser.com/2014/03/15/3d-viz-with-qgis-three-js/

Thank you for Listening!