

# Crunching and visualizing Big Data on a Computer Cluster

Joana Simões

March 30, 2015



# Table of Contents

- 1 Introduction
- 2 Importing a Spatial-Temporal Series
- 3 Recovering the Spatial Attributes
- 4 Putting it All Together
- 5 Piping the Results into the Outside World

## Motivation

- Problem: the increasing volume of information, by explosion of traditional sources + new sources
- Target: fast query responses, which require a scalable architecture
- Possible solution: support clusters on a cost-effective architecture, such as commodity clusters or cloud environments



# Importing a Spatial-Temporal Series Recovering the Spatial Attributes Putting it All Together Piping the Results into the Outside World

## Cloud Services

### Compute

Amazon Elastic Compute Cloud (Amazon EC2)



### Storage

Amazon Simple Storage Service (Amazon S3)



Amazon Elastic Block Storage (Amazon EBS)



AWS Import/Export



AWS Storage Gateway Service



AWS Glacier



### Database

Amazon DynamoDB



Amazon Relational Database Service (Amazon RDS)



Amazon ElastiCache



### Networking

Amazon Route 53



Amazon Elastic Load Balancing



AWS Direct Connect



Amazon Virtual Private Cloud (VPC)



### Content Delivery

Amazon Cloudfront



Elastic Network Instance



### Application Services

Amazon Simple Queue Service (SQS)



Amazon Cloudsearch



Amazon Simple Email Service (SES)



Amazon Simple Workflow (SWF)



Amazon Simple Notification Service (SNS)



### Deployment and Management

Amazon Elastic Beanstalk



AWS Identity and Access Management (IAM)



AWS CloudFormation



### Monitoring

Amazon CloudWatch



### Non-Service Specific



## A thought...

First the use case, then the tools.

## Use Case

- Study spatial and temporal patterns of road traffic accidents.
- Relate target variable (accident) with context variables (e.g.: weather, proximity to SPI).

## Use Case

- Study spatial and temporal patterns of road traffic accidents.
- Relate target variable (accident) with context variables (e.g.: weather, proximity to SPI).
- In most (Big) Data Analysis 80% of the effort is devoted to the Extract-Transform-Load (ETL) process
- ETL: process responsible for pulling data out of the source systems and placing it into a data warehouse

## Use Case

- Study spatial and temporal patterns of road traffic accidents.
- Relate target variable (accident) with context variables (e.g.: weather, proximity to SPI).
- In most (Big) Data Analysis 80% of the effort is devoted to the Extract-Transform-Load (ETL) process
- ETL: process responsible for pulling data out of the source systems and placing it into a data warehouse
  - **Extract** data from different source systems and convert it into one consolidated data warehouse format which is ready for transformation processing
  - **Transform**: cleaning, filtering, splitting a column, joining data, apply validation, apply rules, etc
  - **Load**: into the data warehouse, repository or reporting applications

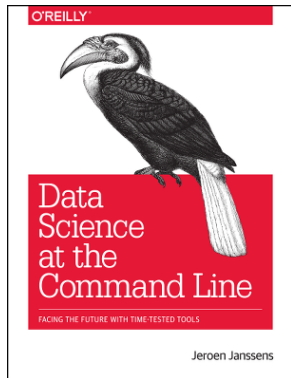


## Another thought...

There are no free lunches.

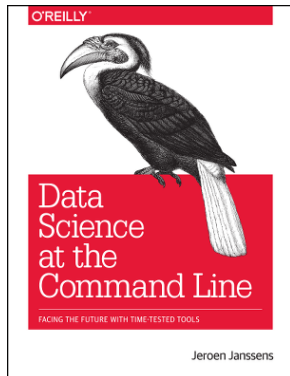
## Scalable ML Platforms

- ML platforms may provide high-level data import tools:



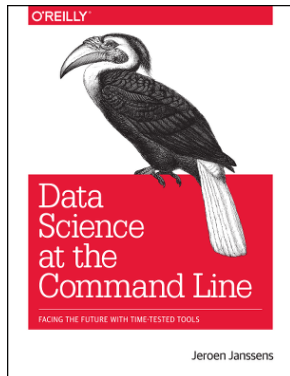
## Scalable ML Platforms

- ML platforms may provide high-level data import tools:
  - They generally trade ease of use for flexibility



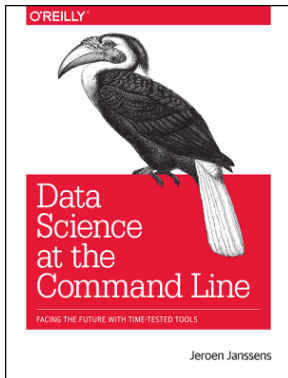
## Scalable ML Platforms

- ML platforms may provide high-level data import tools:
  - They generally trade ease of use for flexibility
  - They may have a cost (\$\$\$)

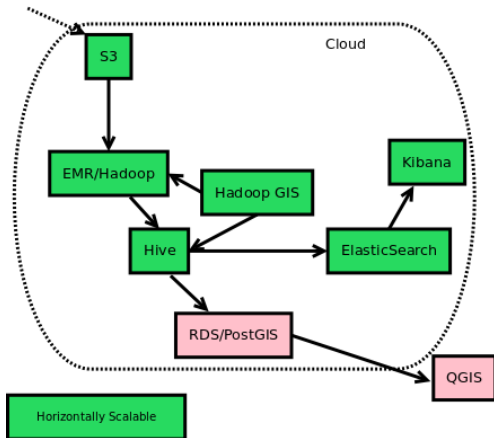


## Scalable ML Platforms

- ML platforms may provide high-level data import tools:
  - They generally trade ease of use for flexibility
  - They may have a cost (\$\$\$)
- To ensure maximum flexibility, we should be able to link together many tools, often using the command line.

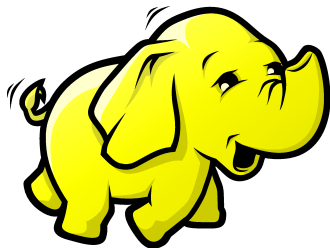


# Stack



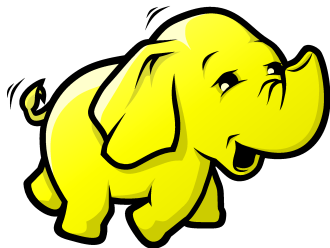
# Apache Hadoop

- framework for distributed storage and processing of (Big) Data on computer Clusters



# Apache Hadoop

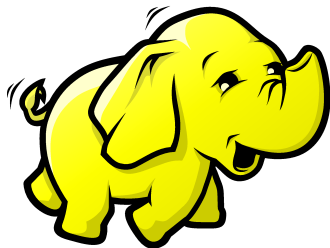
- framework for distributed storage and processing of (Big) Data on computer Clusters
  - **Storage:** HDFS
  - **Processing:** MapReduce





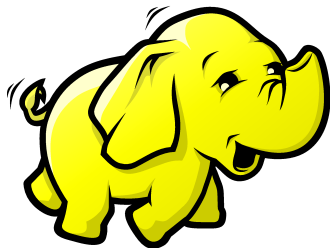
# Apache Hadoop

- framework for distributed storage and processing of (Big) Data on computer Clusters
  - **Storage:** HDFS
  - **Processing:** MapReduce
- It features a FOS license (Apache 2.0)



# Apache Hadoop

- framework for distributed storage and processing of (Big) Data on computer Clusters
  - **Storage:** HDFS
  - **Processing:** MapReduce
- It features a FOS license (Apache 2.0)
- EMR is an Amazon service that uses Hadoop



# Apache Hive

- Data warehouse infrastructure built on top of Hadoop for providing data summarization, query, and analysis.



# Apache Hive

- Data warehouse infrastructure built on top of Hadoop for providing data summarization, query, and analysis.



# Apache Hive

- Data warehouse infrastructure built on top of Hadoop for providing data summarization, query, and analysis.
- HiveQL: query language based on SQL
  - An internal compiler translates HiveQL statements into MapReduce jobs
- It features a FOS license (Apache 2.0)



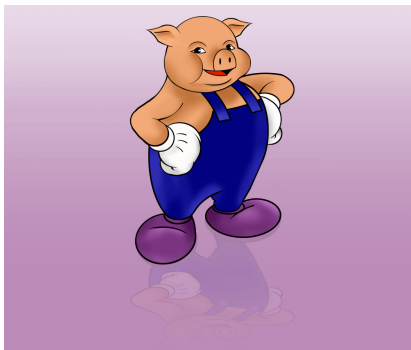
# Apache Hive

- Data warehouse infrastructure built on top of Hadoop for providing data summarization, query, and analysis.
- HiveQL: query language based on SQL
  - An internal compiler translates HiveQL statements into MapReduce jobs
- It features a FOS license (Apache 2.0)
- EMR features an Hive installation



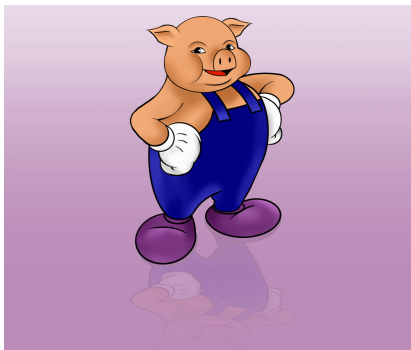
# Apache Pig

- High-level platform for creating MapReduce jobs over Hadoop.



# Apache Pig

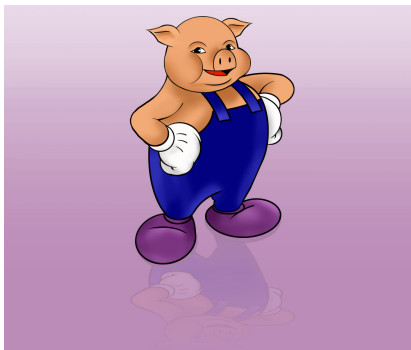
- High-level platform for creating MapReduce jobs over Hadoop.
- It features an interactive mode and a batch mode





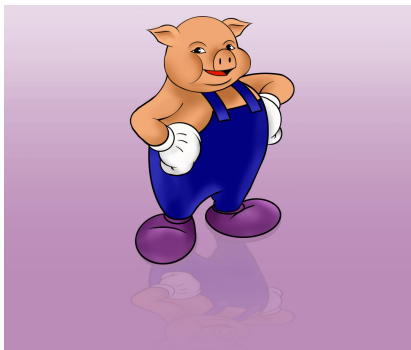
# Apache Pig

- High-level platform for creating MapReduce jobs over Hadoop.
- It features an interactive mode and a batch mode
- It uses lazy evaluation



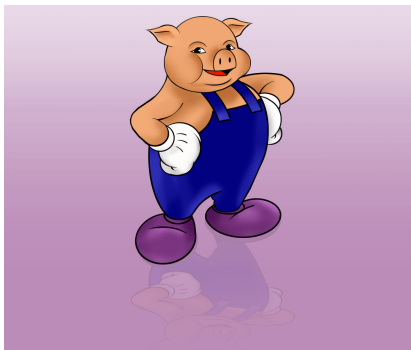
# Apache Pig

- High-level platform for creating MapReduce jobs over Hadoop.
- It features an interactive mode and a batch mode
- It uses lazy evaluation
- Pig Latin is procedural



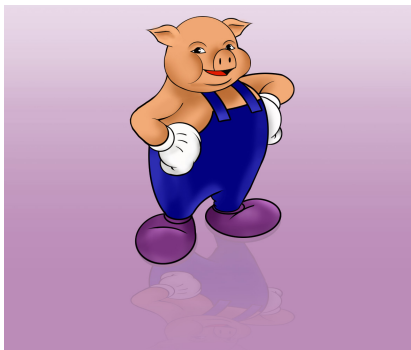
# Apache Pig

- High-level platform for creating MapReduce jobs over Hadoop.
- It features an interactive mode and a batch mode
- It uses lazy evaluation
- Pig Latin is procedural
- It features a FOS license (Apache 2.0)



# Apache Pig

- High-level platform for creating MapReduce jobs over Hadoop.
- It features an interactive mode and a batch mode
- It uses lazy evaluation
- Pig Latin is procedural
- It features a FOS license (Apache 2.0)
- EMR may install Pig



Importing a Spatial-Temporal Series  
Recovering the Spatial Attributes  
Putting it All Together  
Piping the Results into the Outside World

# Hadoop GIS

Importing a Spatial-Temporal Series  
Recovering the Spatial Attributes  
Putting it All Together  
Piping the Results into the Outside World

# PostgreSQL + PostGIS

Importing a Spatial-Temporal Series  
Recovering the Spatial Attributes  
Putting it All Together  
Piping the Results into the Outside World

E(L)K

# 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



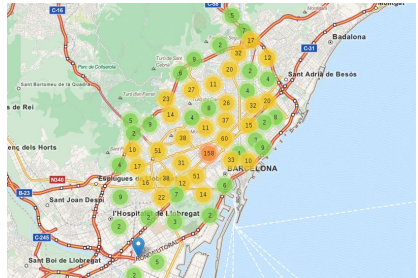
## From S3 to HDFS (cont.)

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

## From S3 to HDFS (cont.)

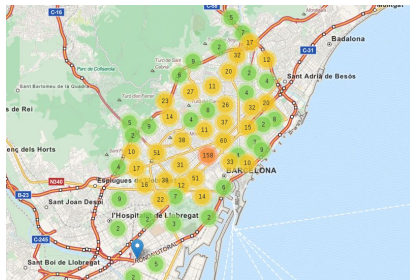
- Micro-task: Type Mapping
  - Create an empty table with correct types
  - Insert data from accidents\_import
  - View table

# 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)
    - European Grid (EPSG:5554)
    - European Grid encoded by the police using an *ad-hoc* format

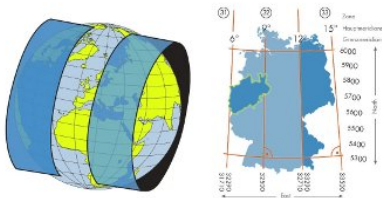
# Analysis of the Spatial Attributes

- 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)
    - European Grid (EPSG:5554)
    - European Grid encoded by the police using an *ad-hoc* format
    - $lon = y/1000 + 400000$
    - $lat = y/1000 + 4500000$

# CRS

**World Geodetic System (WGS84, EPSG:4326):** standard for use in cartography, geodesy, and navigation; reference CRS for GPS.

**European Terrestrial Reference System 1989 (ETRS89, EPSG:5554):** proposed, multipurpose Pan-European mapping standard; based on the ETRS89 Lambert Azimuthal Equal-Area projection coordinate reference system



# 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 subsetting 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 subsetting 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 subsetting 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

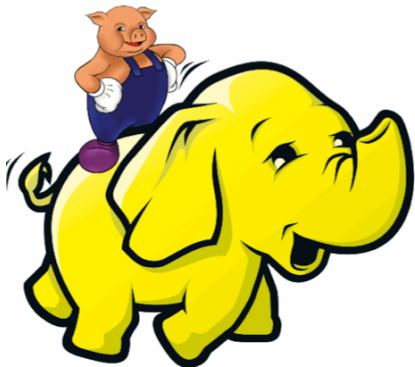
## Exporting Data to Pig

- Pig uses filters to subset the data
- To merge back the subsetting 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



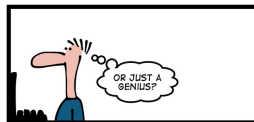
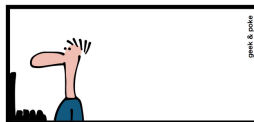
# Presenting the Pig Script

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



# REGEX

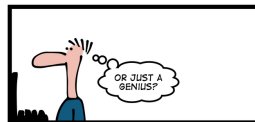
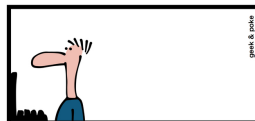
- Sequence of characters that forms a search pattern, mainly for use in pattern matching with strings, or string matching



YESTERDAYS REGEX

# REGEX

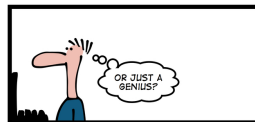
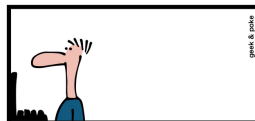
- Sequence of characters that forms a search pattern, mainly for use in pattern matching with strings, or string matching
- `REGEX_EXTRACT(D_COORD_Cz]',0)`



YESTERDAYS REGEX

# REGEX

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



YESTERDAYS REGEX

# 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](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](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 grid 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 grid 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 grid data to PostGIS
  - Merge grid (grid + 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 grid data to PostGIS
  - Merge grid (grid + police) tables in a single table
  - Exported merged table into TSV



# Importing Data into PostGIS

- Micro-task: Import grid data into PostGIS

# Importing Data into PostGIS

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



# Importing Data into PostGIS

- Micro-task: Import grid 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 grid 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 grid 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 European grid to WGS84

# CRS Transformation

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

# CRS Transformation

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

# CRS Transformation

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

# CRS Transformation

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



# CRS Transformation

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

# GeoJSON

**GeoJSON:** is an open standard format for encoding collections of simple geographical features along with their non-spatial attributes using JavaScript Object Notation.

The screenshot shows the GeoJSONLint website. The browser address bar displays 'geojsonlint.com'. The site has a dark navigation bar with the following links: Point, LineString, Polygon, Feature, FeatureCollection, and GeometryCollection. Below the navigation bar, a message reads: 'Use this site to validate and view your GeoJSON. For details about GeoJSON, read the spec.' On the left, a text area contains a valid GeoJSON 'FeatureCollection' with two features: a point in Sydney, Australia, and a line string connecting Sydney to Cape Town, South Africa. Below the text area are two buttons: 'Test GeoJSON' and 'Clear'. A checkbox labeled 'Clear Current Features' is checked. On the right, a world map displays the geographical features defined in the GeoJSON: a blue pin in Sydney, Australia, and a blue line connecting Sydney to Cape Town, South Africa. The map includes labels for continents (North America, South America, Europe, Africa, Asia, Australia) and oceans (North Atlantic Ocean, South Atlantic Ocean, Indian Ocean). The footer of the map states: 'Powered by Leaflet - Tiles Courtesy of MapQuest, Map data (c) OpenStreetMap contributors, CC-BY-SA'.

# 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

# Understanding Indexes

# Generating Indexes

## 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://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/Kernel_%28statistics%29#Kernel_functions_in_common_use)
- [http://en.wikipedia.org/wiki/Cluster\\_analysis](http://en.wikipedia.org/wiki/Cluster_analysis)
- [https://plugins.qgis.org/plugins/clusterpy\\_qgis\\_plugin/](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!

