Big Data Praktikum

Project „Flights“

Tutors:

Todor Ivanov

Kim Hee

Drafted by:
Vasil Radushev
Yavor Obreshkov

Project „Flights“

Even though commercial aviation is one of the most sophisticated and complex transportation systems, it is far away from perfect. Flight delays and cancellations are a well-known problem, which tends to increase in recent years. Thus, we decided to examine the problem further and to gain insight in the major causes and the busiest airports in the US. An article in The Economist (Dec 4th 2017) on take-off and landing slots on congested airports was the source of our inspiration to take a detailed look on data on flight delays. Our dataset consists of data on US flights for the full year 2015.

Our main goal is, using the tools discussed for big data streaming and processing, to provide at least partial answers to questions like:

- The airlines with the most delayed flights
- The main reasons behind delays in each month and their distribution
- The average number of delayed flights over the year/month/day
- Flights to which destination / from which destinations are characterized with the most extensive delays
- Finding a pattern (e.g. flights Dec – Feb mainly delayed due to weather conditions)

We consider the dataset to be appropriate to work with in the Apache Hadoop environment, as conventional tools like Microsoft Excel are simply not powerful enough. When trying to load the data in an Excel Sheet, we only managed to load partial data for the first three months.

The dataset we have already captures all the flights from 2015. However, the data itself is recorded, and already stored, thus making it difficult to spot problems in the mid-term. In other words, we have to wait for the 2015 data to be captured and stored in order to be able to tell ex-post where we have had any problems with delayed flights. Although useful, we would like to make our implementation of the project based on flow of data, allowing us to be able to monitor the events in “near-real time.” Therefore, we want to simulate a flow of data (streaming) that would be generated, transferred (processed), received and stored on the Hadoop Cluster.

In order to do that, we will use Kafka as our main tool, as it provides the ability to “produce”, store and stream data in a distributed manner on a cluster. For our implementation we will proceed as follows:

First, we will create a Kafka topic, to which producers would send some data:

Assuming the producer(s) to be the Airport Traffic Control Towers and the consumer would be the centralized monitoring system of the Air Traffic Organisation (ATO). As a full-scale implementation of this streaming process would require more than 320 producers, we will run the process with one producer streaming data to one topic. Our source would be the file flights.csv. We assume our data source to be stable, reliable and properly maintained, meaning that there are no errors such as missing flight numbers, wrong flight numbers, etc. We would try to simulate Kafka “messages” that would contain data for flights scheduled for departure every 10 seconds. Please note that we could adjust the time window depending on the number of flights to maintain a reasonable flow of messages.

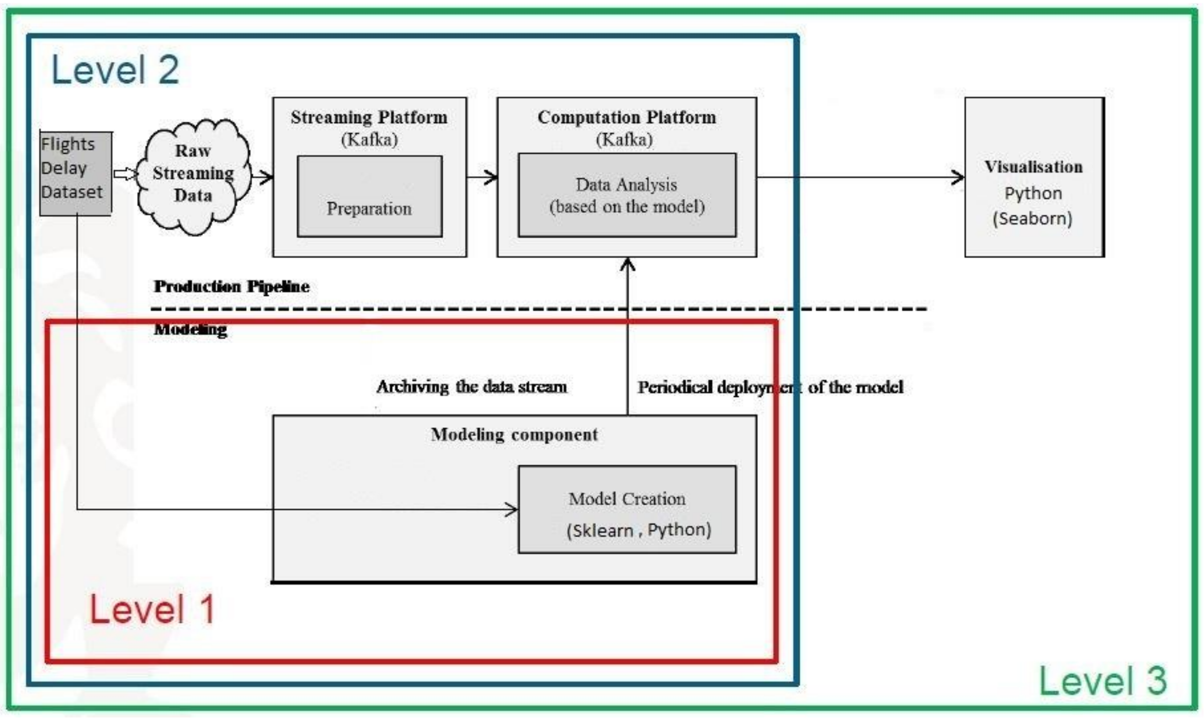
Another possibility, which we consider is using the Kafka Processing API, as this guarantees us a much more reliable streaming.

In a next step, to visualize the results and generate reports would be to use Python as integrated in the Anaconda Development environment using the Jupyter notebook. We could then use Python libraries such as *numpy* or *matplotlib* to try to identify and visualize a pattern or a trend. Once we are able to identify a sort of pattern or a trend, we would be in a better position to provide answers to the questions in this proposal.

This sums up the main idea of our project. For now, we will refrain from making "bold" statements and we would focus on getting useful, reliable results, that could help us better understand the seemingly never-ending problem with flight delays. We will then focus on finding solutions or possible measures that could be taken in combating the problems.

Advanced methods of machine learning and predictive analysis could be eventually by data scientists and business intelligence specialists used to predict airport congestions, providing both airports and air carriers with useful, up-to-date data and allowing them to take the precautions and measures to possibly limit costs and decrease the number of passengers affected from this problem.

Our System Architecture


Level 2
Flights Delay Dataset
Raw Streaming Data
Streaming Platform (Kafka)
Preparation
Computation Platform (Kafka)
Data Analysis (based on the model)
Visualisation Python (Seaborn)
Production Pipeline
Modeling
Archiving the data stream
Periodical deployment of the model
Modeling component
Model Creation (Sklearn , Python)
Level 1
Level 3

Preparing the VM for the project

1. We are working in the virtual machine Cloudera-Quickstart-VM-5.12.0-0.
2. We are using Python 2.7.13.
3. Download and install Kafka using parcels in the Cloudera Manager
4. Download and install Anaconda Navigator 4.3.1 via cloudera@quickstart terminal:
   $ conda install -c conda-forge
5. Create in Anaconda Navigator new environment my_root:
   $ conda create -n my_root --clone=/opt/cloudera/parcels/Anaconda
6. Download and Install jupyter-notebook via cloudera@quickstart terminal:
   $ python -m pip install jupyter
7. Download and install kafka library 1.3.5 for python in the new my_root environment:
   $ conda update kafka-python
8. Download, install and update in Anaconda Navigator (**my_root environment**) all the necessary libraries: pandas 0.22.0, numpy 1.13.1, matlibplot 2.0.2, json 2.6.0, seaborn 0.8, scikit-learn 0.19.0
9. Start jupyter-notebook in **my_root environment** via **terminal@bash-4.1$**
   $ jupyter-notebook
10. Start the Cloudera Manager via cloudera@quickstart terminal:
    $ sudo /home/cloudera/cloudera-manager --pause --express --force
11. Start all the services in the Cloudera Manager.
12. Uploading all the necessary datasets (flights.csv, airlines_num.csv, airports.csv) in the jupyter-notebook in **my_root environment**.

Data Streaming

1. Create a kafka topic via cloudera@quickstart terminal:

$ kafka-topics --create --zookeeper quickstart.cloudera:2181 --replication-factor 1 --partitions 1 --topic topic0302

2. Check if the topic was created via cloudera@quickstart terminal

$ kafka-topics --list --zookeeper localhost:2181

3. Create a kafka-producer and kafka-consumer using jupyter-notebook in the my_root environment
4. Consumer script: Model+KafkaConsumer.ipynb
5. Producer script: KafkaProducer.ipynb
6. Run the producer script (sometimes it does not start, so try running the consumer first and then producer)
7. Monitoring the streaming via the terminal:

$ kafka-console-consumer --zookeeper localhost:2181 --topic topi0302 --from-beginning

Model, Predictions and Plots

1. Start the Model+KafkaConsumer.ipynb
2. For the modelling part where the model is trained we are using the dataset flights with 100 000 rows.
3. After this we are training the model on a dataset with 1 million rows.
4. At the end we are using the full dataset with ~ 5.8 million rows.
5. We are using the consumer script and we are changing the source file.
6. When the consumer script is opened in jupyter-notebook on the second line starts the predictions part from incoming messages.
7. After running it as a result a user could see the predicted delay on every flight (every message in the consumer) based on our model.