Big Data Praktikum

Project „Flights“ - Proposal

Tutors:

Todor Ivanov

Kim Hee

Drafted by:

Vasil Radushev	Student ID: 4654326	E-mail: r_vasil@yahoo.com
Yavor Obreshkov	Student ID: 4718741	E-mail: yav.obreshkov@gmail.com

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:

- Which months are characterized as the months with most delayed flights
- Which months are characterized as the months with most cancelled flights
- The airlines with the most delayed flights
- The airlines with the biggest delays
- The main reasons behind delays in each month and their distribution
- What % of flights delayed due to one of the four (five) reasons
- Where was the delay aggregated: was it on the runway, or while boarding passengers?
- Was the delay due to the airport or the carrier (Who is to blame?)
- 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.

Therefore, our **first step** will be transferring the files into HDFS. We prefer HUE as our main HDFS Browser, as it also includes a resource manager, which allows us to monitor the jobs on the cluster.

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 30 minutes. Please note that we could adjust the time window depending on the number of flights to maintain a reasonable flow of messages.

Having set up the Kafka environment, we could use a Flume agent to stream the data into Hive. We consider using Hive, as it allows processing of data using SQL queries on the Hadoop cluster, thus providing us with the possibility to run multiple queries fast and reliable. Here, we should note that creating an empty external table in Hive containing the header (the fields) from the flights.csv is essential and we would do it upfront, before starting the streaming process. Having the data streamed to Hive, we could run multiple queries and try to answer some of our questions.

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 we could use Tableau because it is functioning well with Hive.

Another way around 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.