

CS 644 Introduction to Big Data

Project Report

Flight Data Analysis



Submitted by -
Anish Gaikwad - aag79
Jaini Bhavsar - jkb37



Content

Introduction	2
Software Specifications	3
Dataset Information	4
Oozie Workflow	5
Algorithms	6
Performance Measurement Plots	9
References	11

Introduction

Flight delays and cancellations spiked at several points over the last year, costing U.S. carriers more than \$100 million combined and disrupting travel plans of hundreds of thousands of customers.

According to the Bureau of Transportation Statistics (BTS), most flight delays are caused by Maintenance, Crew changes (either full or partial), Checked baggage loading, High load rate (full flights), Late arriving feeder flights, Cleaning, Fueling, etc.

With an enormous reservoir of data at disposal, big data technology can transform the way airlines do business. By prioritizing data collection and analysis, even small airlines can respond to customer demands and market trends with precision and agility.

The ultimate benefits of big data analytics include timely responses to current and future market demands, improved planning and strategically aligned decision making, as well as crystal clear comprehension and monitoring of all main performance drivers relevant to the airline industry.

The goal of this project is to generate the insights from the airlines data to know the following and prevent you to stuck at the airport.

1. The 3 airlines with the highest and lowest probability, respectively, for being on schedule
2. The 3 airports with the longest and shortest average taxi time per flight - The most busy airports
3. The most common reason for flight cancellations

Big Data Analytics will completely overhaul the travel experience in the upcoming years.

Software Specifications

1. Mac OS - Version 12.2.1
2. Amazon EC2



3. Hadoop



4. Oozie



5. Java



6. Maven



DataSet Information

120 Million Records

Size - 1.6 Gigabytes (Compressed)

12 Gigabytes (Uncompressed)

Features:

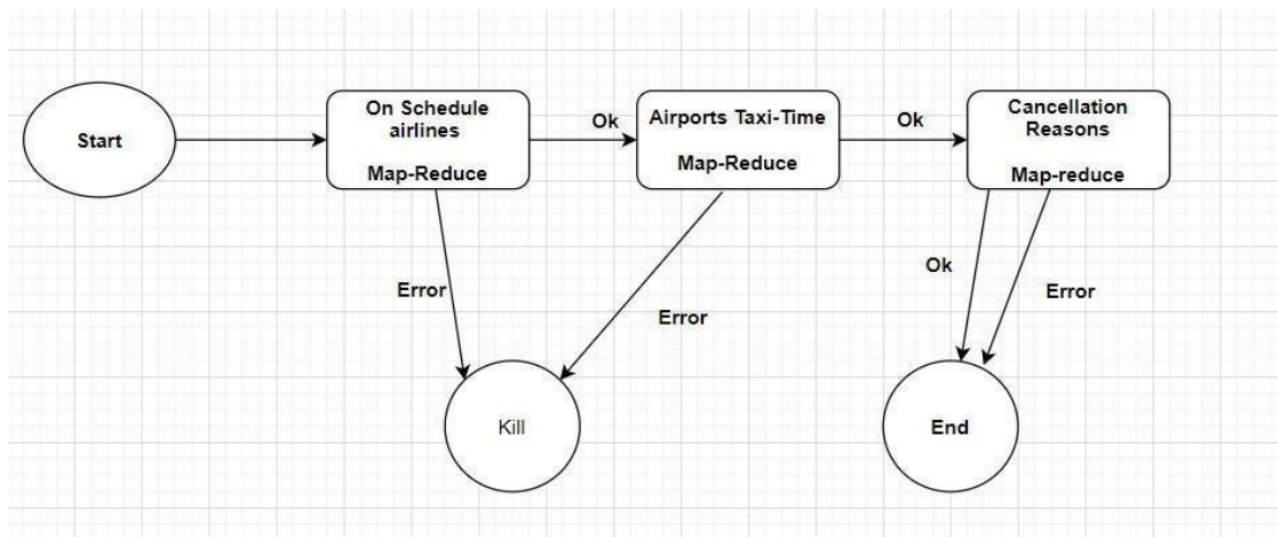
1. Year 1987 - 2008
2. Month 1-12
3. DayofMonth 1-31
4. DayOfWeek 1 (Monday) - 7 (Sunday)
5. DepTime - actual departure time (local, hhmm)
6. CRSDepTime - scheduled departure time (local, hhmm)
7. ArrTime - actual arrival time (local, hhmm)
8. CRSArrTime - scheduled arrival time (local, hhmm)
9. UniqueCarrier - unique carrier code
10. FlightNum - flight number
11. TailNum - plane tail number
12. ActualElapsedTime - in minutes
13. CRSElapsedTime - in minutes
14. AirTime - in minutes
15. ArrDelay - arrival delay, in minutes
16. DepDelay - departure delay, in minutes
17. Origin - origin IATA airport code
18. Dest - destination IATA airport code
19. Distance - in miles
20. TaxiIn - taxi in time, in minutes
21. TaxiOut - taxi out time in minutes
22. Cancelled - was the flight cancelled?
23. CancellationCode - reason for cancellation (A = carrier, B = weather, C = NAS, D = security)
24. Diverted - 1 = yes, 0 = no
25. CarrierDelay - in minutes
26. WeatherDelay - in minutes
27. NASDelay - in minutes
28. SecurityDelay - in minutes
29. LateAircraftDelay - in minutes

Oozie Workflow

Apache Oozie is a Java Web application used to schedule Apache Hadoop jobs. Oozie combines multiple jobs sequentially into one logical unit of work. It is integrated with the Hadoop stack, with YARN as its architectural center, and supports Hadoop jobs for Apache MapReduce.

An Oozie Workflow is a collection of actions arranged in a Directed Acyclic Graph (DAG) . Control nodes define job chronology, setting rules for beginning and ending a workflow. In this way, Oozie controls the workflow execution path with decision, fork and join nodes. Action nodes trigger the execution of tasks.

Oozie workflow for this project is shown below.



Algorithms


A. First Map – Reduce: On Schedule Airlines

1. The Mapper starts first.
2. The mapper reads the data line by line but ignores the first line and the NA data. If the data of the ArrDelay is less than or equal to 10, it returns the output.
3. Now, the Reducer starts.
4. Reducer adds the values from the mapper of the same key and the sum will be the number of the airplanes of this airline on schedule. It then calculates the number of 0's and 1's and then calculates the on schedule probability of this airline.
5. The Reducer then uses the comparing functions to sort the data and then outputs the 3 airlines with the highest and lowest probability.
6. If the data is NULL, then the output will be a statement stating that the operation cannot be done.



B. Second Map – Reduce : Airport Taxi- Time

1. The Mapper starts first.
2. The Mapper reads the data line by line but ignores the first line. If the data of the TaxiIn or the TaxiOut column is NA, there will be no output.
3. Now, the reducer starts.
4. The Reducer adds the values from the mapper of the same key, calculates the total times the key is found. Then it does the calculation Normal/ All to calculate the Average time of each key.
5. The Reducer then uses the comparing functions to do the sorting. After sorting, it outputs the 3 airports with the shortest and longest waiting times.
6. If the data is NULL, the output will be a statement stating that there is no output.

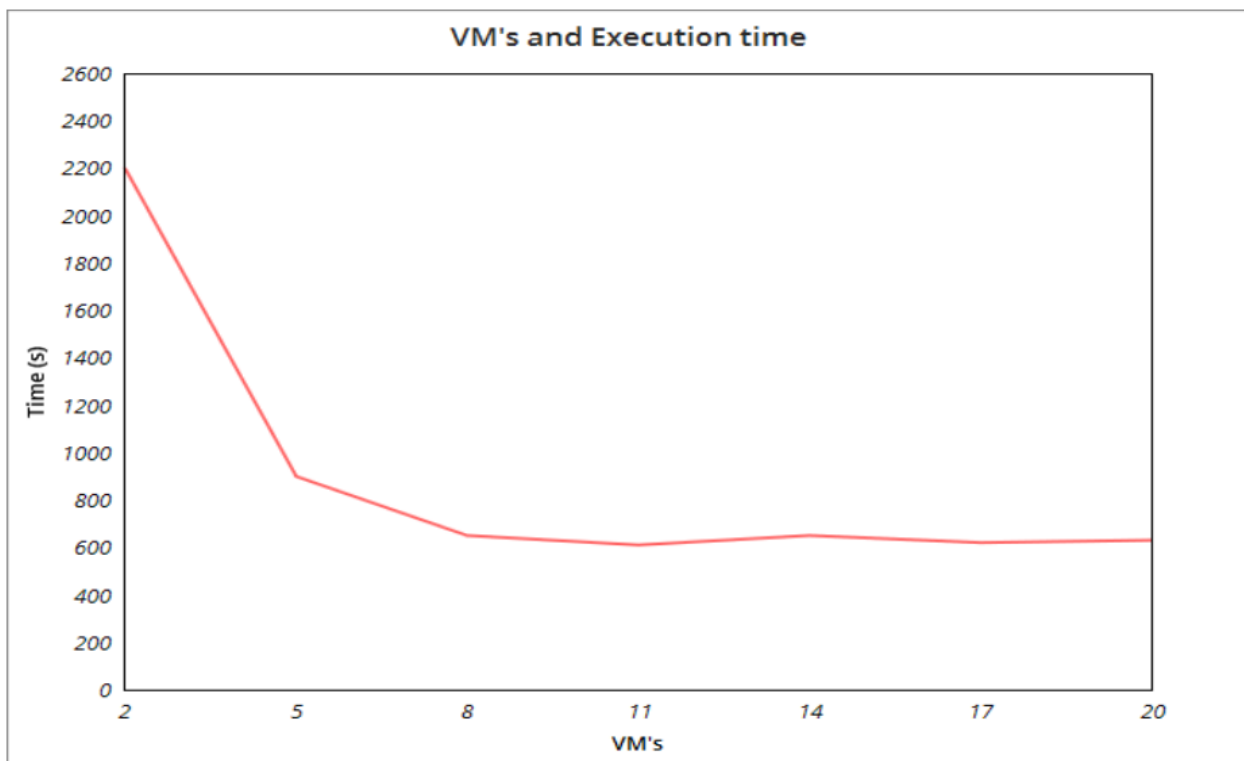


C. Third Map – Reduce : Cancellation Reasons

1. The Mapper starts first.
2. The Mapper reads the data line by line but ignores the first line. If the value of the cancel is 1 and the CancellationCode is NA, There is no output.
3. Now, the Reducer starts.
4. Reducer adds the values of the mapper of the same key.
5. The reducer then uses the comparing functions to do the sorting. After sorting, it outputs the most common reasons for the cancellations.
6. If the data is NULL, the output will be a statement that states that there is no output.

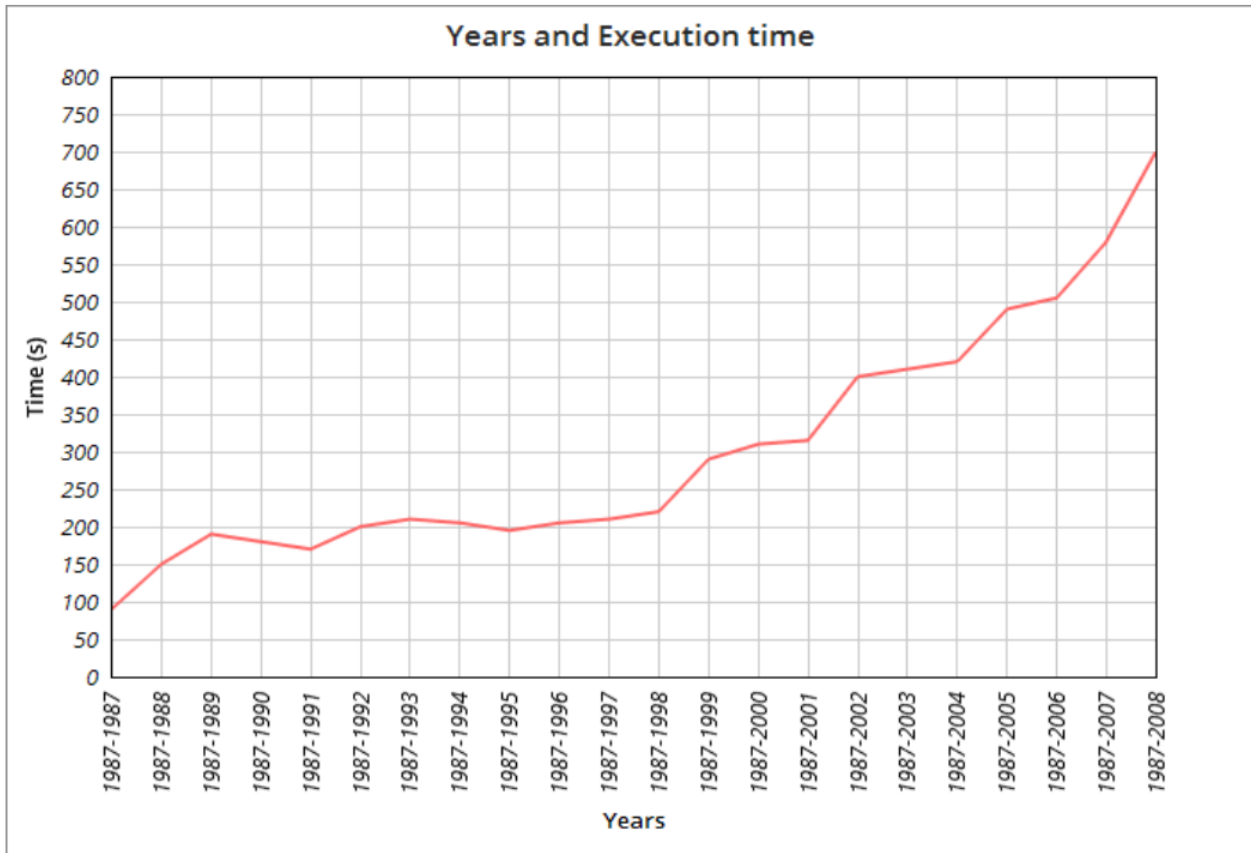
Performance Measurement Plots

1. Increasing number of VM's (Entire data set)



According to the above figure, along with the increasing number of VM's, the workflow execution time will decrease. By increasing the number of VM's the processing ability of the cluster also increases which is obvious because the number of machines working on the dataset is more than the previous number, every time we increase. But this will not be the case forever because as we increase the number of VM's at a point, the processing ability becomes stable. Though the ability keeps increasing, the interaction between the data nodes also increases with the number, which makes the cluster slower.

2. Increasing size of data (20 Vm's)



According to the above figure, along with the increasing data size, the execution time of the workflow will always increase too. As the data increases, the time consumption also increases. As the trend goes by, we can observe that the time has a gradual increase until 1998 because the amount of data generated after 1998 is bigger than the previous years and the time consumed is also high. This also tells us that the number of people preferring airways has also increased after 1998.

References

1. *'9 INCREDIBLE WAYS DATA ANALYTICS IS TRANSFORMING AIRLINES'*, ANASTASIIA ZAMIATINA
2. *'BIG DATA CASE STUDY: 5 RELEVANT EXAMPLES FROM THE AIRLINE INDUSTRY'*, CARLOTA FELIU
3. *'Sky is the limit for Big Data Analytics in the Aviation Industry'*, TechVidvan
4. *'How airlines use your personal data to reduce delays – and why you should let them have it'*, Arnon Shimoni, July 28th 2019
5. *'Apache Oozie'*, Cloudera