Flight Delay Analysis

Ayushi

November 13, 2017

Objective

Implement a pipeline that runs a set of pseudo distributed map reduce tasks to plot the mean delay of the five most active airlines and for the five most active airports in the country.

System Specification

Amazon EMR cluster

Instance Type: m3.xlarge

Memory: 15GB

Storage: 2 x 40 GB SSD

vCPU: 4

Execution

Code took approximately 10 mins to run on EMR with the above mentioned specifications.

Implementation

The program is divided into 2 Map-Reduce jobs.

Job 1: Find top 5 Airlines and Airports

Map - TopAirlineAirportMap.ActiveAirportAirlinesMapper:

- Reads the input CSV and counts the flight frequency per airline and per airport. String "0" or "1" is appended to the keys, 0 for Airlines and 1 for Aiports.

Partition - TopAirlineAirportMap.TopPartitioner:

- Partitions the key from the mapper into two different reducers by using "0" and "1" appended to the keys.

Reduce - TopAirlineAirportReducer:

- Receives airline/airport as key and list of frequncies as value.
- Adds the frequency value and calculates the total frequency count of each airline and airport.

Job 2: Calculate monthly flight delay

Map - CalculateMonthlyDelay.AverageMonthlyDelayMapper:

- Reads the output of job 1, sorts it and finds the top 5 airlines and airports.
- Then map function in the first stage reads each record and pchecks if it passes the sanity test and then sets it to the approriate Airline or Airport hashmap based on the calculated top 5 airport and airline.

Partition - CalculateMonthlyDelay.AverageMonthlyPartitioner:

- Partitions each airport or airline by the month (will result in 24 reducers)

- The partitioner differentiates the record between airport or airline by using the first character in the key which is either "0" or "1" and finds out the month which is the 4th element in the key.

Reduce - MonthlyDelayReducer:

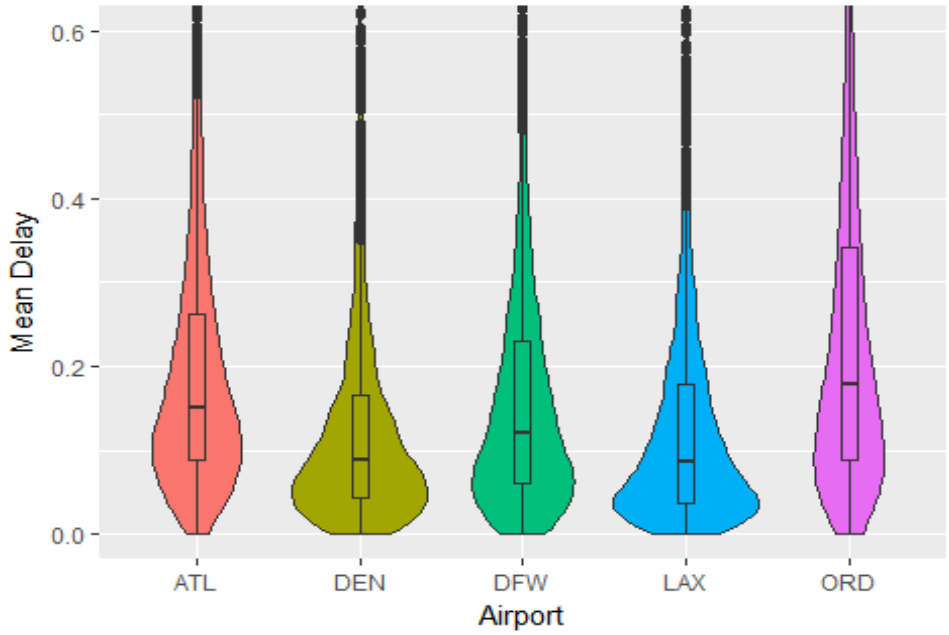
- Aggregrates all the normalized delays for each airport and airline and finds the mean normalized delay for the same. The result is written in a CSV file.

Results

Figure 1. Top 5 Airlines and Airports

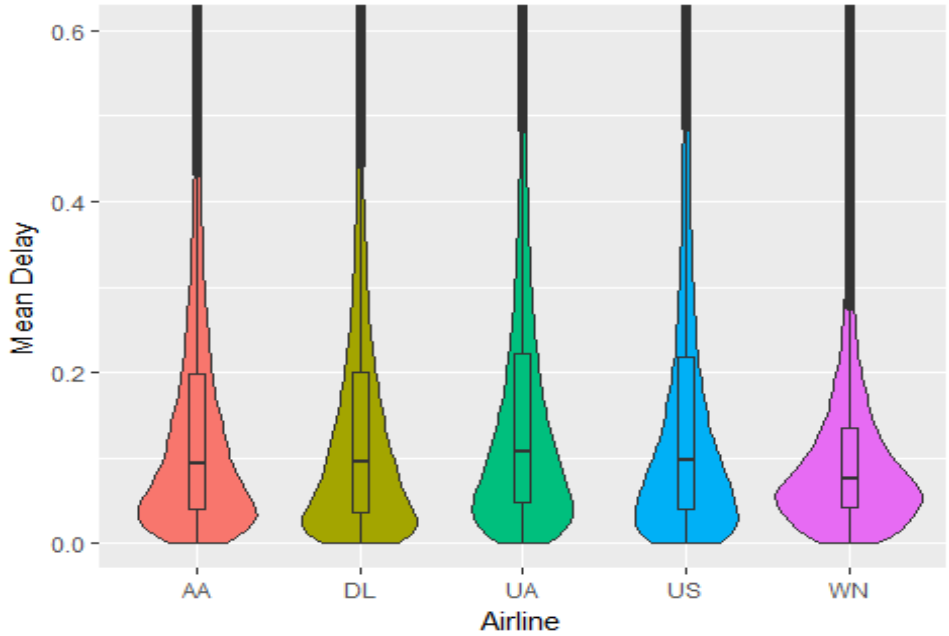
Airlines	Airports
AA	ATL
DL	DEN
UA	DFW
US	LAX
WN	ORD

Figure 2. Mean Delay For 5 Most Active Airports



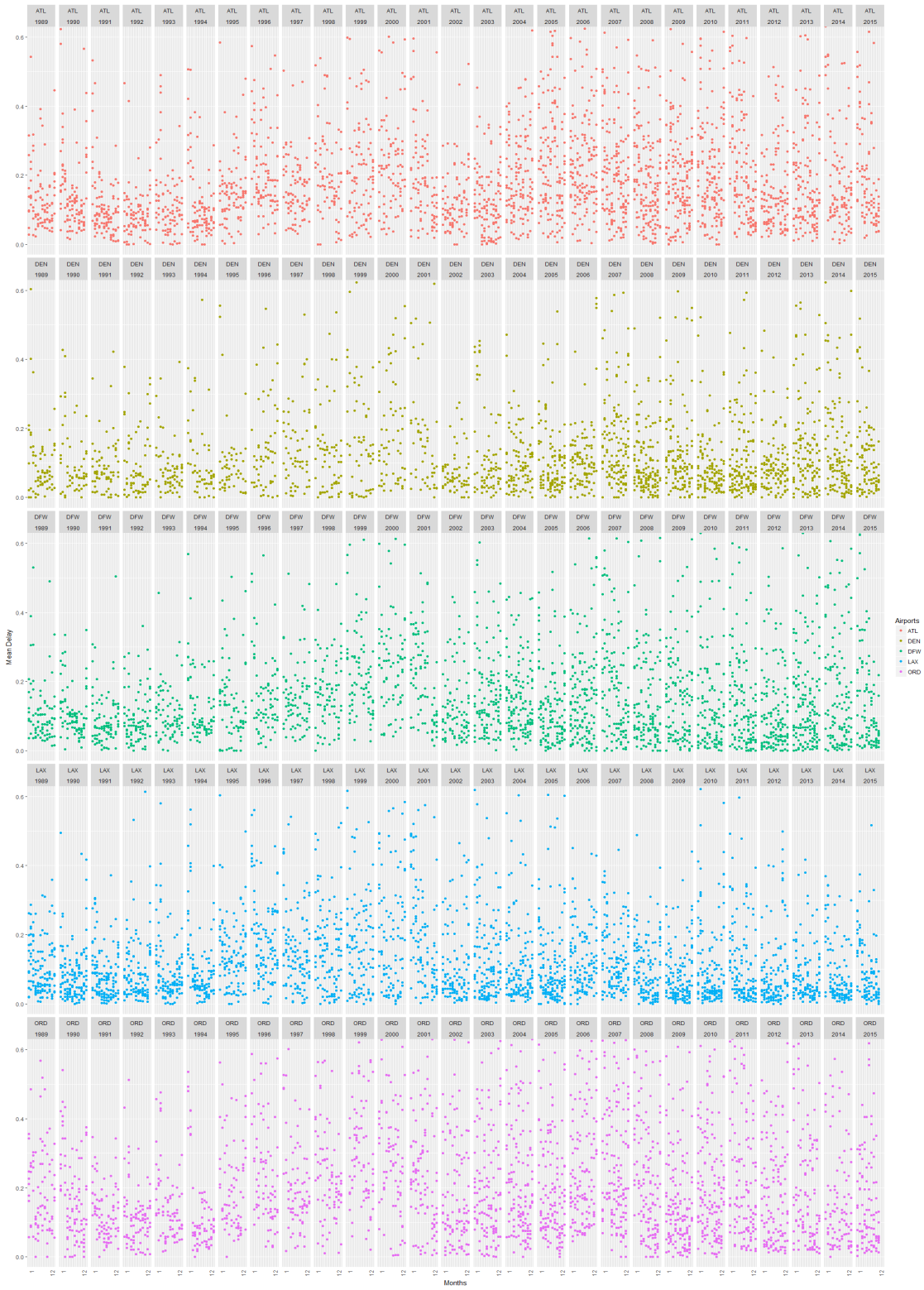
The above Figure presents mean delay for 5 most active airport from 1987-2015. It can be seen that Airport ORD has the highest median with frequency distribution along the complete range. This shows that ORD has performed poorly as compared to other 4 airports.

Figure 3. Mean Delay For 5 Most Active Airlines



The above figure shows mean delay for 5 most active airlines. DL,AA,WN have median around .0175, but as US is more elongated, this shows that it is has maximum number of delays over the given period.

Figure 4. Mean Delay For 5 Most Active Airports



The above Figure represents the mean delay of top five most active airports for each month from year 1987-2015. It can be seen that there is a gradual decrease in delay across years.

The same Figure can be created for Top 5 most active airlines however, due to space constraints it is not included here.However, even the Figure for airlines shows a decrease in delay across years.

Conclusion

The MR jobs for this implementation not only required custom partitioners but also used a lot of util classes like CSVparser,NonSplitableTextInputFormat etc.
Having these utils helped in controlling the number of MR jobs and increased efficiency.