**Introduction to Big Data**

**CS644**

**Project: Flight Data Analysis**

**Semester: Spring 2019**

Group Members:

Shawn Rahul D’Souza

UCID: srd59

NJIT ID: 31476009

Venkata Sai Vinay Malapati

UCID: vm436

NJIT ID: 31482320

**Introduction:**

For this project, we have analyzed the patterns of 22 years flight data using Hadoop in a fully distributed mode with an Oozie workflow to solve following three problems:

a. The 3 airlines with the highest and lowest probability, respectively, for being on schedule

b. The 3 airports with the longest and shortest average taxi time per flight (both in and out), respectively;

c. The most common reason for flight cancellations.

**A. Oozie Workflow:**

A close up of a map

Description automatically generated

**B. Algorithm:**

1. Flight Delay:

* This uses one mapper and one reducer.
* In mapper class, we first read the file. We have consider arrival delay with threshold of 5 minutes.
* For entries that have an arrival delay time greater than 5 min are assigned the value 0. Otherwise commit entries that falls under 5 min with value 1.
* In reducer, take the sum of all values under same key airline. And find the probability of it being rescheduled.
* In cleanup function, sort them based on probability and the top 3 and bottom 3 values on being scheduled.

1. Flight Taxi Time:

* This uses one mapper and one reducer.
* Read the file in the mapper.
* For the taxi time, we have ignored the NA values, and commit (origin, taxiOut) and (destination, taxiIn) pairs.
* In reducer, find the average taxi time by counting total taxi time and divide it with total count value.
* In cleanup function, them based on the average taxi time and then commit longest 3 and shortest 3 values of average taxi time with airport.

.

1. Flight Cancellation:

* This uses one mapper and one reducer.
* In mapper class, read the file and check for canceled value. If it is 1 and value of cancellation code is not NA, then commit that entry with value 1 (cancellation\_code, 1)
* In reducer class, count the number of 1’s by cancellation code and then commit the cancellation code with highest value.

**C. Performance by increasing number of EC2 instances:**

We’ve start from 2 instances which have taken more than 300 seconds. After increasing the number of instances gradually, to find how long it would take to run whole 22 year data decreased. Since we are increasing the number of instances, this means that it will take less time to process the data.

**D. Performance by increasing data size (from 1 to 22 years) :**

We first run single year 1987 data. Then, we have increase the data size by increasing the years data upto 1987-2008. In the chart we can see that as the data size increases, the time it would take to run the program is also increases.