CS-644 Introduction to Big Data

PROJECT: FLIGHT DATA ANALYSIS

TEAM MEMBERS

1.HARSHITHA SARIPILLI(HS759)

2.AKARSH KANAPARTHI(AK2856)

3.BHARGAVI REDDY KAKIREDDY(BK385)

4.DAMODHAR MAHIDHAR TANNIRU(DT378)

Project: **Flight Data Analysis**

Contents

Introduction and Overview

Three map-reduce scripts are used in this project to analyze the flight data utilizing an Oozie-based workflow. Click the following link to download the flight data analysis: https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/HG7NV7

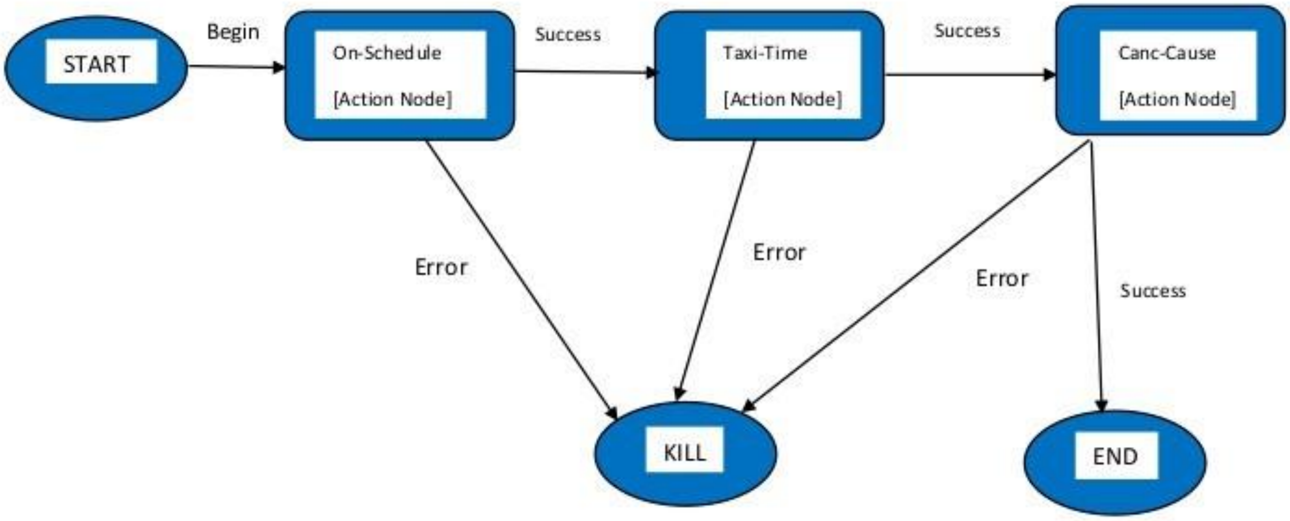
Every map-reduce software tackles one of the following problems:

a. the three airlines with the greatest and lowest likelihood of operating on time;

b. the three airports where the average taxi wait time, for both arriving and outgoing planes, is longest and lowest;

c. the main cause of aircraft cancellations.

Oozie Workflow



Algorithms Description:

a. On Time Flights

1. Decide on 10 as the delayThreshold value.
2. The Mapper program determines the total ArrDelay and DepDelay for every flight.

3. The flight is deemed to be on schedule if the total is less than the delay threshold. Emit (Carrier, 1) to show that the flight is proceeding according to plan.
4. The flight is deemed delayed if the total exceeds or equals the delay threshold. To let the pilotknow if the flight is not on time, emit (Carrier, 0).
5. Compute the TotalCount in the Reducer by summing the values for every Carrier key.
6. If the value at the Reducer is 1, increase the onSchedule count for each Carrier key by one.
7. Calculate onSchedule/totalCount to determine the likelihood of being on time.
8. Concatenate the probability and carrier into an array list.
9. Repeat steps 5-8 for each carrier key received by the reducer.
10. Sort the arraylist in decreasing probability order while doing context cleanup.
11. Transfer the arraylist's values (Carrier, probability) to HDFS.

b. Average Taxi Time

1. The Mapper emits (Origin, TaxiOutTime) & (Destination, TaxiInTime) for every trip.
2. To get totalCount, add together all of the values for each Airport (Origin/Destination) key at the Reducer.
3. To determine the overall taxi time, add up all of the values (TaxiInTime/TaxiOutTime) for each Airport key at the Reducer.
4. Divide the total taxi time by the totalCount to find the average taxi time for that airport.
5. Include the average taxi ride time and the airport in an array list.
6. Repeat steps 3-5 for each airport key that is given to the reducer.
7. Sort the arraylist by typical taxi times in decreasing order during context cleanup.
8. Upload the values (Airport, avgTaxiTime) from the arraylist to HDFS.

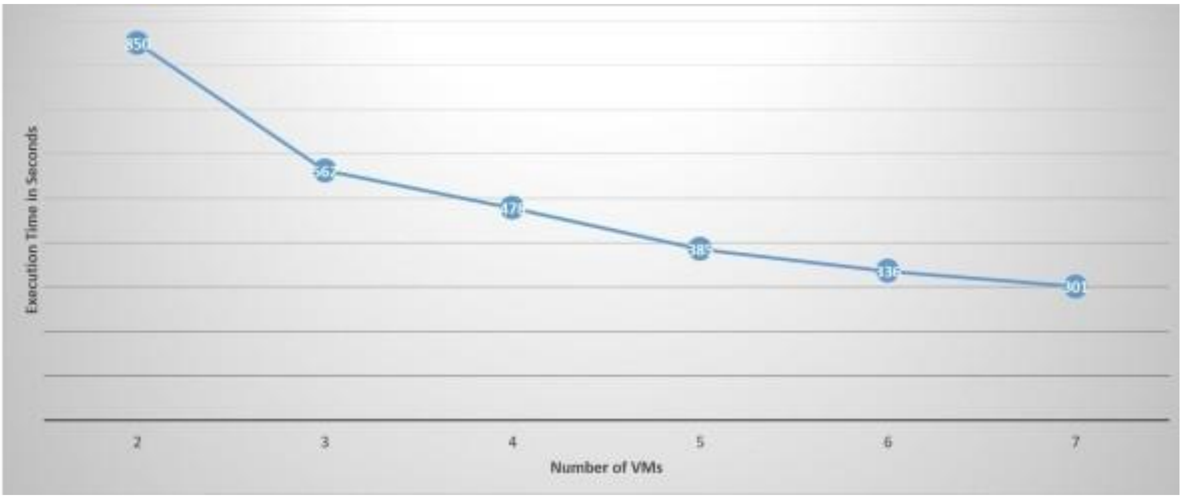
c. Most Common Cancellation Cause

1. The Mapper emits (CancellationCode, 1) for each flight (Cancelled = 1) in the event of a cancellation.

2. To obtain the totalCount in the Reducer, sum up all of the values for every CancellationCode key.
3. Commit (CancellationCode, totalCount) to HDFS.
4. Repeat steps 2-3 for each CancellationCode key received by the Reducer.

Performance Measurement Plot -1

examines how long it takes for a workflow to execute over a 22-year period when more virtual machines (VMs) are employed to process the whole data set.



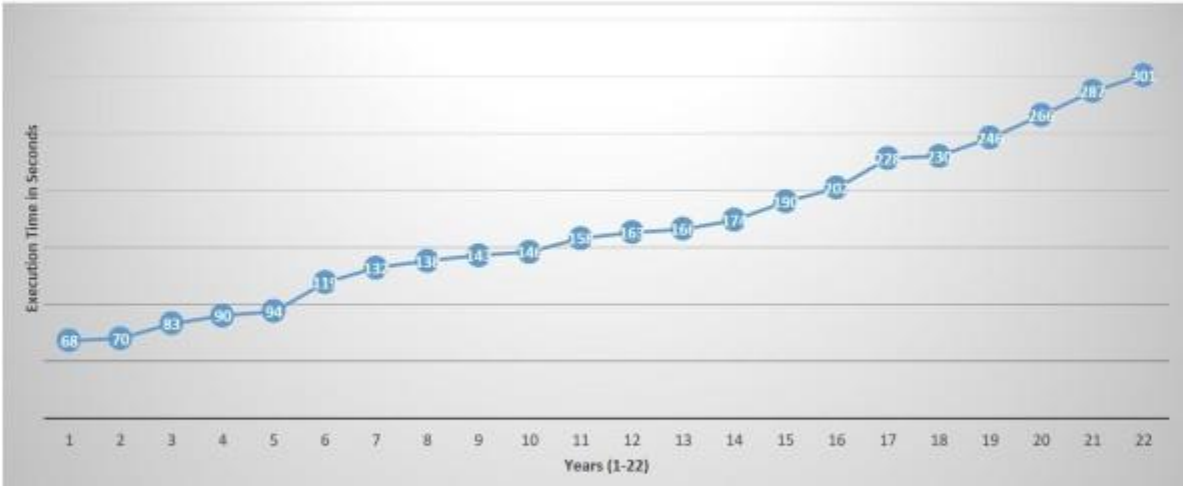
The Hadoop/Oozie cluster had two virtual machines before, but now there are seven. Every step inserts a new virtual machine.

The expansion from two to three virtual machines (VMs) results in a significant reduction in execution time. After then, the increase of virtual machines (VMs) reduces the time required to finish a workflow virtually linearly. A cluster with seven virtual machines has a total time of 301 seconds, over 2.83 times faster than the first cluster, which had only two virtual machines (850 seconds).

The previously described workflow execution periods provide credence to the idea. Increasing the number of virtual machines makes more computational resources available, which promotes parallel processing and expedites data processing.

Performance Measurement Plot -2

demonstrates how the execution time of the workflow varies with the amount of data (from one year to 22 years).



For the statistic, the maximum number of virtual machines (VMs), in this case seven, were utilized.

Initially, the data from 1987 is the only ones examined. More data for the following year is contributed to the Hadoop/Oozie cluster at each stage. All the data from 1987 to 2008 are examined at the end.

Processing time grows as data volume does, as would be expected. When the data size is expanded from one year to twenty-two years, the method execution time increases by a factor of about 4.4 (from 68 seconds to 301 seconds).

It makes sense that increasing the amount of input will require more processing to obtain the result because the computational resources in this case are fixed.