# PROJECT REPORT OF CS-644

## Introduction to Big Data, Fall-2019

The Airline on Performance Flight data Analysis

Abhijit Chakraborty and Jhona Davied D Souza
ac687@njit.edu, jd655@njit.edu

## A.1 Structure of the Oozie Workflow

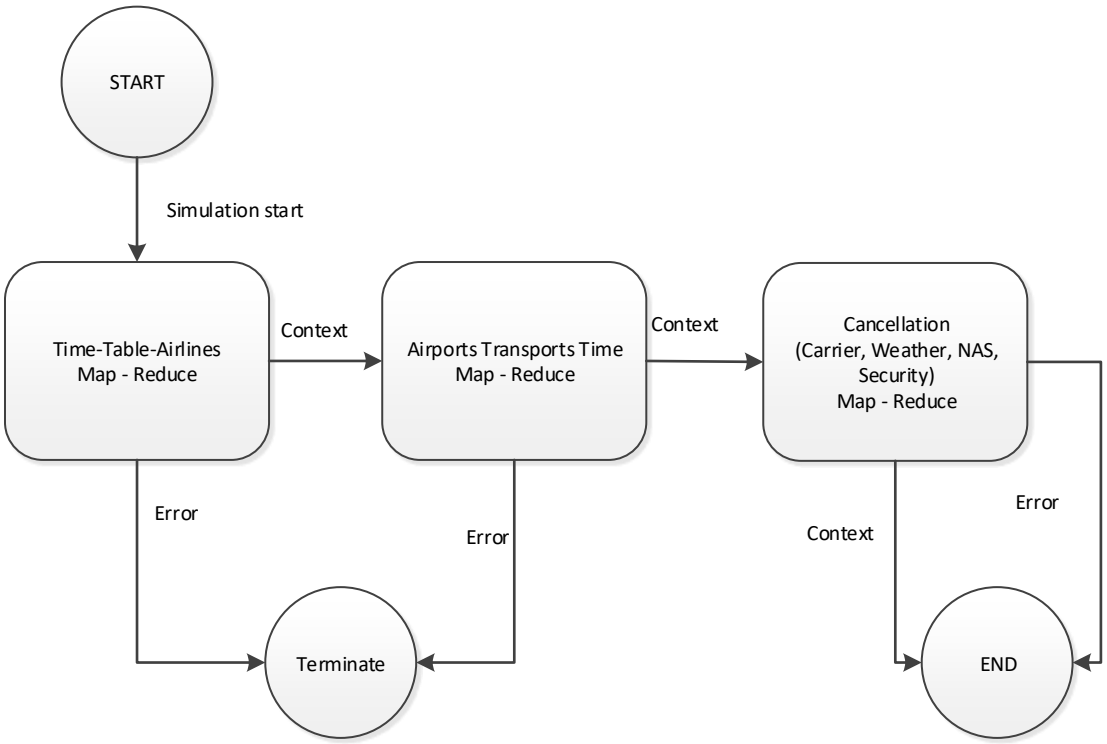



**Figure 1 : An Oozie workflow of airlines time-table with the transports time (taxi time) and flight cancellations.**

This Oozie workflow diagram (Figure 1) works with the three different combinations of algorithms that comprises with the following MapReduce programs:

1. First MapReduce: The Time-Table of Airlines:

2. Second Map Reduce: Airports Transport Time (Taxi-time)

3. Third Map Reduce: Stop Flight with Cancellation Reasons especially three

## A.2 Description of the Flight Algorithms

### First MapReduce: The Time-Table of Airlines:

1. Mapper <key,value>:<UniqueCarrier,1or 0>
2. The Mapper reads each line of data, ignore the first line and the NA data. If the data of the ArrDelay(min) ≤ 10min, output: <UniqueCarrier,1>, or output: <UniqueCarrier,0>
3. Reducer<key,value>:<UniqueCarrier,probability> Probability = (# of 1) / (# of 1 and 0)

4. Reducer collects the values from the mapper of the same key and creates the sum that will be the total number of this airlines when scheduled. Moreover, It calculates the time-table of on schedule probability of that particular airline.
5. Reducer then uses the Comparator function to do the sorting(decreasing). After sorting, the time-table shows the output of the 3 airlines with the highest and lowest probability.
6. If the data is NULL, then output shows “No Flights Found”.

**Second Map Reduce: Airports Transport Time (Taxi-time):**

1. Mapper <key,value>: <IATA airport code, TransportTime>: <Origin,TaxiOut>or <Dest,TaxiIn>
2. The Mapper reads each line of data but ignores the first line. If the data of the TaxiIn or the TaxiOut column is not NA, output: <IATA airport code, TransportTime>
3. Reducer <key,value>: <IATA airport code, Average TransportTime>
4. Reducer sums the value from the mapper of the same key (initial), and calculate the total times this key is found (all). Then identifies the equation: initial/tuple to calculate the average TransportTime of each key.
5. Reducer uses the Comparator function to do the sorting in decreasing order. After sorting, output the 3 airports with the longest and shortest average TransportTime(Taxi time).
6. If the data is NULL, then output: No Transport Time Found.

**Third Map Reduce: Stop Flight with Cancellation Reasons especially three**

1. Mapper <key, value>: < CancellationCode,
2. The Mapper reads each line of data but ignores the first line. If the value of the Cancelled is 1 and the CancellationCode is not NA, output: < CancellationCode, 1>
3. Reducer <key, value>: < CancellationCode, sum of the1s>.
4. Reducer sums the value from the mapper of the same key.
5. Reducer then uses the Comparator function to do the sorting in decreasing order. After sorting, output the most common reason for flight stops.
6. If the data is NULL, then output: No Reason for flight Stop.

## B. 1 VMs Vs Executing Time

The Following Figure 2 describes the workflow execution time that will decrease along with the increasing the number of the VM s. By increasing the number of the VMs, the processing ability of the Hadoop cluster will also increase because the data can be dealt with in parallel one or more data nodes. The execution time of every map reduce job will be shorter than that of before. Therefore, the execution time of the oozie workflow will be less than the previous too. However, the execution time of pacts with the same data size that will not be always decreased by growing the number of VMs.

Although trying to increase the number of VM, the execution time will no longer be decreasing anymore when the execution time decreases to a certain range. The motive is more VMs means more information interaction time between the data-nodes of a Hadoop cluster. Information interaction time of a Hadoop cluster growths when the number of VMs upsurges.

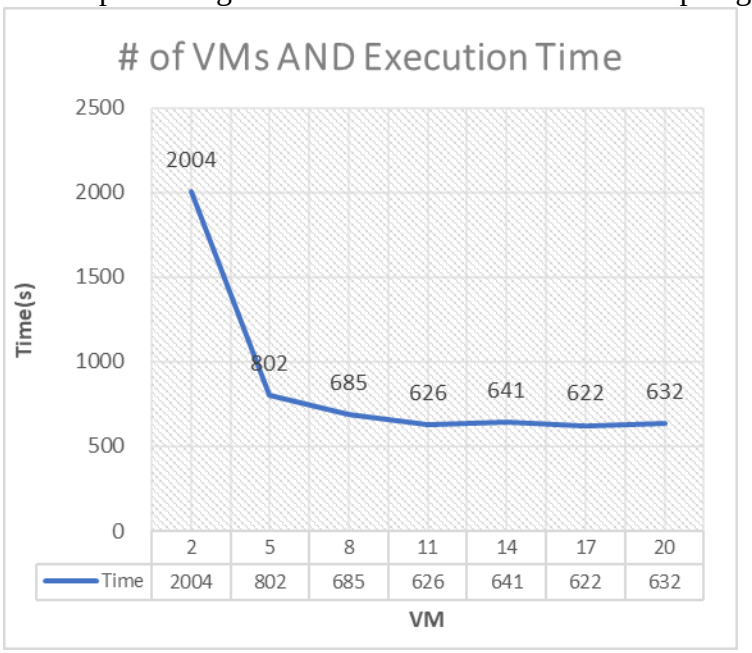
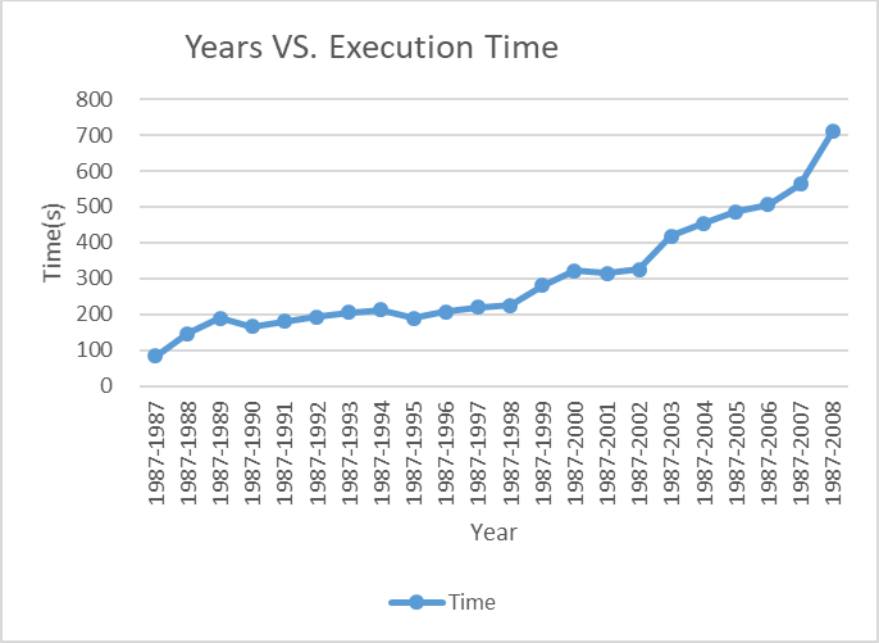



**Figure 2 : The VM vs TimeTable Graph**

### B.2 The Execution time of Each Year

The Figure 3 describes the execution time of the oozie workflow along with the increasing data size. Initially, the time-consuming increases with the rise in the amount of data, but the time-consuming increment is leisurely because the data increment of first few years is less. Quite the reverse, after year 1999, the time-consuming increases very rapidly with the pace and the slope develops much vertical compare to the first few years. The purpose is the flight data between year 1999 to the year 2008 that is in increment order always and is faster than that of the previous years. This correspondingly means that, the people are choosing flights more than other conveyances.




## C. The Excel Files for the Above-Mentioned Graphs

| | | |
|---|---|---|
| 1987-1987 | 84 | 1 minutes 24 seconds |
| 1987-1988 | 147 | 2 minutes 27 seconds |
| 1987-1989 | 190 | 3 minutes 10 seconds |
| 1987-1990 | 167 | 2 minutes 47 seconds |
| 1987-1991 | 181 | 3 minutes 01 seconds |
| 1987-1992 | 194 | 3 minutes 14 seconds |
| 1987-1993 | 206 | 3 minutes 26 seconds |
| 1987-1994 | 212 | 3 minutes 32 seconds |
| 1987-1995 | 190 | 3 minutes 10 seconds |
| 1987-1996 | 207 | 3 minutes 27 seconds |
| 1987-1997 | 220 | 3 minutes 40 seconds |
| 1987-1998 | 225 | 3 minutes 45 seconds |
| 1987-1999 | 281 | 4 minutes 41 seconds |
| 1987-2000 | 321 | 5 minutes 21 seconds |
| 1987-2001 | 315 | 5 minutes 15 seconds |
| 1987-2002 | 325 | 6 minutes 25 seconds |
| 1987-2003 | 419 | 6 minutes 59 seconds |
| 1987-2004 | 454 | 7 minutes 34 seconds |
| 1987-2005 | 487 | 8 minutes 07 seconds |
| 1987-2006 | 506 | 8 minutes 26 seconds |
| 1987-2007 | 564 | 9 minutes 24 seconds |
| 1987-2008 | 710 | 11 minutes 50 seconds |

| VM | Time |
|---|---|
| 2 | 2004 |
| 5 | 802 |
| 8 | 685 |
| 11 | 626 |
| 14 | 641 |
| 17 | 622 |
| 20 | 632 |