Introduction to Big Data - CS 644

Final Project Report

Flight Data Analysis

New Jersey Institute of Technology, Newark, NJ

GROUP MEMBERS:

HITESH SAAI MANANCHERY PANNEERSELVAM (UCID: hm374)

PAVANDEEP SINGH (UCID: ps355)

Table of Content

INTRODUCTION

The objective of this project is to do Analysis for the 22 years (1987-2008) of Airline Data (11.2 GB approximately). In this project, we will be using 3 MapReduce programs in a sequence using Oozie workflow.

Each MapReduce Oozie workflow finds out :

- The 3 airlines with highest and lowest probability, respectively, for being on schedule
- The 3 airports with the longest and shortest average taxi time per flight (both in and out).
- The most common reason for flight cancellations.

There are two tasks to be performed in order to analyze the data provided :

- Firstly, we run the data in an incremental manner and on fully distributed mode. In order to do that we will analyze the data with an increment of 1 year (1987), the first two years (1987-1988), the first three years (1987-1989) and so on up-to total 22 years (1987-2008) on the VMs (7 VMs in Fully-Distributed mode). Each time we do that we will measure execution time for each.
- Next we will Run Oozie workflow on the entire dataset (1987-2008) and increase VMs gradually . We will start with one time on two VMs and then gradually increase the system scale to the maximum VMs and measure each execution time for each workflow.

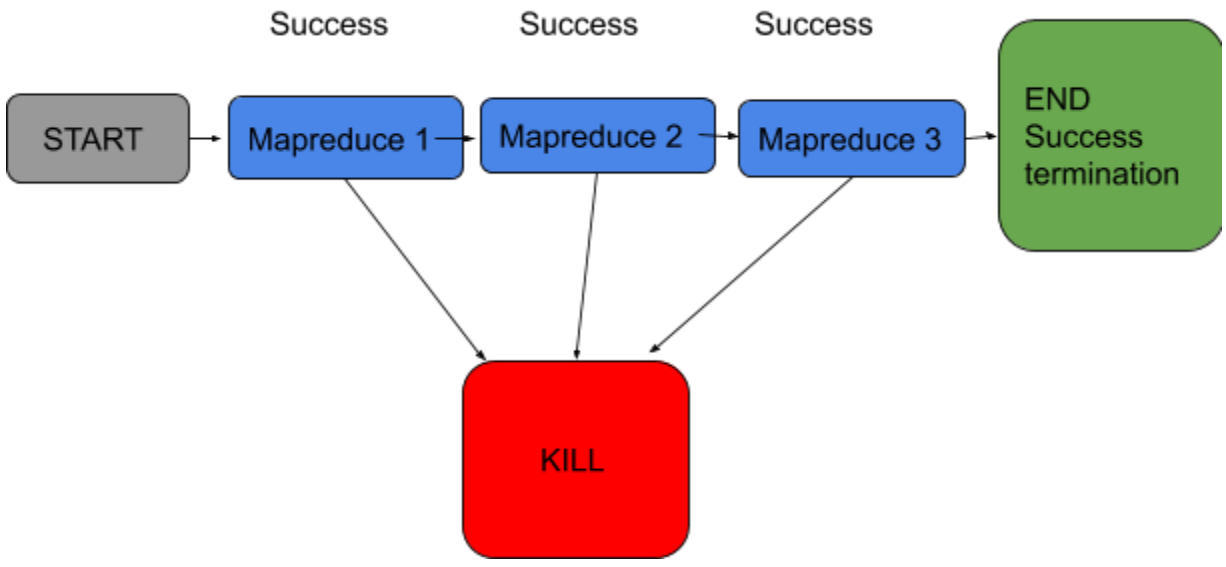
ABOUT OOZIE

Oozie is a server based *Workflow Engine* specialized in running workflow jobs with actions that run Hadoop Map / Reduce and Pig jobs. Apache Oozie is the tool in which programs can be pipelined in a desired order to work in Hadoop's distributed environment. Oozie also provides a mechanism to run the job at a given schedule.

Oozie is a Java Web-Application that runs in a Java servlet-container.

For the purposes of Oozie, a workflow is a collection of actions (i.e. Hadoop Map/Reduce jobs, Pig jobs) that are arranged in a control dependency DAG (Direct Acyclic Graph). "Control Dependency" from one action to another is defined as Second action is performed only after completion of first.

OOZIE WORKFLOW:



Map Reduce 1 - On Scheduled Flights (OnSchedule)

Map Reduce 2 - Average Taxi Time (TaxiAverage)

Map Reduce 3 - Reason for Cancellation (CancellationCauseReason)

The Oozie workflow has three MapReduce jobs. Once first job is successfully run, second is initiated, Same thing happens with third and that starts with the completion of second job. In case of errors, workflow is killed or terminated.

- The oozie workflow starts with the MapReduce1 - OnSchedule which analyzes the data to find the top 3 airlines that have the lowest and highest probability of being on schedule. Oozie workflow on the Success of this job, sends confirmation(OK) to MapReduce2. Any error leads to the termination of workflow.
- Success of MapReduce 1 initiates second job MapReduce2 (TaxiAverage). This job computes the 3 airports with the longest and shortest average taxi time. Success of this job, sends confirmation(OK) to MapReduce3. Any error leads to the termination of workflow.
- On getting success from MapReduce2, MapReduce3 (CancellationCauseReason) starts. This job computes the most common reason for flight cancellation. On completion of this job, the workflow ends the job (signals the end of the job). The outputs are available in respective files.

ALGORITHM

1. **First Map-Reduce: On Schedule Airlines**
    a. Mapper <key, value>:<UniqueCarrier,1 or 0>
    b. The Mapper read the data line by line, ignore the first line and the NA data. If the data of the ArrDelay column which is less than or equal to 10 minutes, output: <UniqueCarrier,1>, otherwise output: <UniqueCarrier,0>
    c. Reducer <key, value>:<UniqueCarrier, probability>
    d. Reducer sum the values from the mapper of the same key, the sum will be the number of this airline. when it is on schedule. And calculate the total number of 0 and 1, then calculate the on schedule probability of this airline.
    e. Reducer then use the Comparator function do the sorting. After sorting, output the 3 airlines with the highest and lowest probability.
    f. If the data is NULL, then output: There is no value can be used, so no output
2. **Second Map-Reduce: Airports Taxi Time:**
    a. Mapper <key,value>: <IATA airport code, TaxiTime>: <Origin,TaxiOut> or <Dest,TaxiIn>
    b. The Mapper read the data line by line, ignore the first line. If the data of the TaxiIn or the
    c. TaxiOut column is not NA, output: <IATA airport code, TaxiTime>
    d. Reducer <key,value>: <IATA airport code, Average TaxiTime>

e. Reducer sum the value from the mapper of the same key (normal), and calculate the total times this key is found (all). Then do the equation: normal/all to calculate the average TaxiTime of each key.

f. Reducer then use the Comparator function do the sorting. After sorting, output the 3 airports with the longest and shortest average taxi time.

g. If the data is NULL, then output: There is no value can be used, so no output.

3. Third Map-Reduce: Cancellation Reasons:

a. Mapper <key,value>: < CancellationCode, 1>

b. The Mapper read the data line by line, ignore the first line. If the value of the Cancelled is 1 and the CancellationCode is not NA, output: < CancellationCode, 1>

c. Reducer <key,value>: < CancellationCode, sum of the 1s>

d. Reducer sum the value from the mapper of the same key.

e. Reducer then use the Comparator function do the sorting. After sorting, output the most common reason for flight cancellations.

f. If the data is NULL, then output: There is no the most common reason for flight cancellations.

PERFORMANCE MEASUREMENTS

Increasing the VM with the entire dataset:

The performance measure checks the workflow time duration for the different numbers of virtual machines for entire flight dataset of 22 years (1987-2008) .

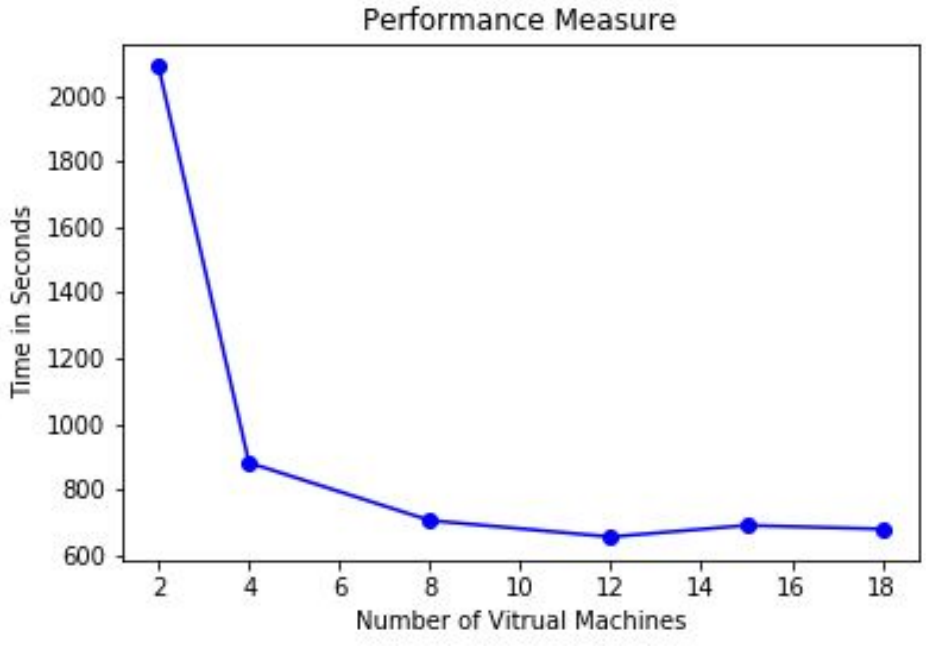


Figure 1

From the Figure 1, we can see that , as we increase the number of VM's there is an gradual decrease in the time duration for the workflow execution. By increasing the number of VM's the processing ability of the hadoop is also increasing because the data is handled in the parallel processing mode. When the execution time decrease to a certain range, although trying to increase the number of VM, the execution time is no longer decreasing. The reason is more VMs means more information interaction time between the datanodes of a hadoop cluster. Information interaction time of a hadoop cluster increases when the number of VMs increases.

Increasing the Dataset size with same number of VM's :

Running the workflow to analyze the data in a progressive manner with an increment of 1 year, i.e. the first year (1987), the first 2 years (1987-1988), the first 3 years (1987-1989), and the total 22 years (1987-2008), on the maximum allowed number of VMs, and measuring each corresponding workflow execution time.

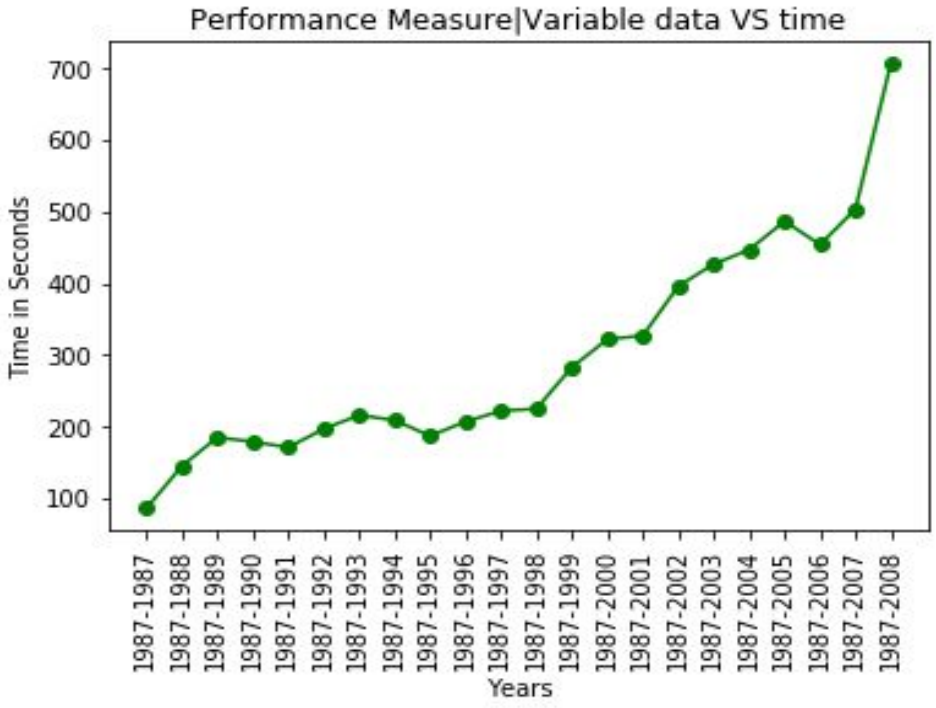


Figure 2

From the figure 2, we can see that with the fixed number of virtual machine says 20 Virtual Machine and with increase in the size of the dataset in regular interval, the time duration it takes to go through the workflow increases. Initially with the one year data of 1987 it takes less than 100 seconds to run the process, and gradually the time consumed started to increase and we can see after the year 2007 there is a sudden peak in the time duration. Overall, there is a linear increase in the time with the increase in the data size.

REFERENCES

AWS EC2 Setup:

https://www.youtube.com/watch?v=XkDhf1pwPtA&list=PLWsYJ2ygHmWjPsg-+6MnQO6WxVWFI8OzK_

Hadoop Installation & Setup:

https://www.youtube.com/watch?v=nnGIGbnqYas&list=PLWsYJ2ygHmWhOvtHIPxGJDtki9SJIINCw

Oozie Installation:

https://acadgild.com/blog/beginners-guide-for-oozie-installation