Aviation Data Analysis Using Apache Pig

HARSHIT KRISHNAKUMAR[1,*] AND KARTHIK ANBAZHAGAN[2]

[1] School of Informatics and Computing, Bloomington, IN 47408, U.S.A.
[2] School of Informatics and Computing, Bloomington, IN 47408, U.S.A.
[*] Corresponding authors: harkrish@iu.edu, kartanba@iu.edu

Project-002, May 2, 2017

Data science is a challenging field which gives actionable insights into data, enabling businesses to take instant decisions. Big data techniques help and accelerate the analysis of data in real time. Big Data can be used to monitor things as diverse as flight data, traffic data, and financial transactions. With huge increase in the volume of air travel and drastic weather changes, flights delays and cancellation are on the rise. This project aims at tracking the aviation data and providing a list of the busiest airports by total flight traffic across the US. The system has been deployed in chameleon cloud platform. Apache Pig deployed on a Hadoop cluster is used to join multiple features across massive datasets to query and analyse the data across a Distributed File System.

Keywords: big data, Apache Pig, Apache Hadoop, Chameleon Cloud, Avaiation Analysis

https://github.com/cloudmesh/sp17-i524/blob/master/project/S17-IR-P002/report/report.pdf

CONTENTS

1. INTRODUCTION

Air travel is getting increasingly popular with the airlines providing cheaper fares and better services. However, owing to the increased congestion in traffic during and ever fluctuating weather conditions, there are a lot of cancellations and delays that happen to the flights. Cancellations tend to be costly for airline companies and pose difficulties for the customers who might have appointments or connecting flights. In the United States, we know that the December holiday period is the most busy time for airlines, and also owing to winter weather, it is the term where most of the cancellations occur.

The current project aims to deploy a Hadoop cluster with three nodes, run analysis on large data files related to airline cancellations and delays using Pig Scripts and benchmark the performance of the clusters for various data sizes and cluster configurations. We utilize Chameleon cloud to run the Hadoop cluster.

Skyscanner is a travel fare aggregator website and travel metasearch engine which helps users find the lowest rates from multiple travel sites, as well as instant comparisons for hotels and car hire removing the need for customers to search across different airlines for prices [**?**].

A metasearch engine (or aggregator) is a search tool that uses another search engine's data to produce their own results from the Internet [**?**]. Metasearch [**?**] engines take input from a user and simultaneously send out queries to third party search engines for results. Sufficient data is gathered, formatted by their ranks and presented to the users. The Skyscanner Live Pricing allows developers to access live pricing information on prices for different flights, by making requests to the Live Pricing API.

In this project, we would be querying the Skyscanner Live Pricing API using Apache HIVE and deploying the data on cloud (1-TBD & 2-TBD). Cloudmesh would be used for cloud

management and the software stack deployment would be done through Ansible. We would benchmark performance of our analysis across mutilple clouds. We would be presenting a real-time visualization of the cheapest air fare and the most likely travel destination analysis in D3.js.

2. WORKFLOW

The project will make use of Python APIs to retrieve live flight prices information from Skyscanner and dump it in Apache HIVE database [**?**]. SQL Analyses are performed on this data and the results of analyses are stored in HIVE and presented in an interactive dashboard or website. The dashboard will take the onward and return journey locations, and the date of travel as inputs from users and show different price ranges for different dates commencing from the next available flight, for a period of three months. This aims to provide the users an idea as to when is the safe time to book flight tickets and beyond which date will the prices shoot up.

3. EXECUTION SUMMARY

The schedule for completion of this project has been outlined below:

1. Mar 06-Mar 12, 2017 Creating virtual machines on Chameleon cloud using Cloudmesh and coming up with a project proposal
2. Mar 13-Mar 19, 2017 using cloudmesh to set up Hadoop clusters and installing the required software packages
3. Mar 20-Mar 26, 2017 Fetching the data from Skyscanner API and adding it to our HIVE database
4. Mar 27-Apr 02, 2017 Running few data mining/time series models to predict the ticket prices
5. Apr 03-Apr 09, 2017 Review the work done and find out scopes for improvement and creating a benchmark report
6. Apr 10-Apr 16, 2017 Presenting the work in D3.Js in real-time as a visualization of the analysis
7. Apr 17-Apr 23, 2017 Complete the Project Report

4. HADOOP DEPLOYMENT

The deployment was done from an Ubuntu14.04 instance running on Oracle Virtual Box. Hadoop instance with three nodes was deployed on Chameleon cloud using Cloudmesh Client was used to automatically create the cluster and deploy Hadoop software with Pig and Spark addons. Three virtual machines with Ubuntu14.04 version were installed on Chameleon Cloud, and Hadoop along with Pig and Spark addons was deployed to the clouds with one name node and two data nodes.

5. CLUSTER CONFIGURATIONS

Hadoop was deployed on a cluster with three nodes. Three virtual machines were created on Chameleon Cloud for this purpose, with one instance as namenode (master) and rest of the instances as datanodes (slaves). All the three virtual machines were created with Ubuntu14.04 OS. Each of the vms had 20 GB space, with 2GB ram and one CPU. This configuration is called the "small" flavour in Chameleon Cloud. Each vm was assigned a separate floating public IP which could be used to SSH and connect.

6. ANSIBLE PLAYBOOK

Ansible was installed in the oracle virtual box and configured to automatically perform file transfers and to run Pig code on the cluster. All our interactions are with the namenode of Hadoop, so the public IP of the namenode of our installation was given in the hosts file of Ansible for all purposes.

Five Ansible Playbooks were setup, each with the following purposes:

1. Transfer data and Pig Script files to namenode from local location
2. Transfer data files to hdfs location on namenode using *hdfs dfs* command
3. Run pig code using *pig <any_pig_script>* shell command on the name node
4. Transfer the output files from hdfs back to namenode
5. Transfer output files to local location

Each of these would run on the locations mentioned in hosts file on usernames CC and hadoop, which were the default users created by cloudmesh client.

7. AIRLINE ANALYSIS

7.1. Datasets

The current project focuses on analysing flight cancellations and delays data to find trends. The dataset is taken from the US Department of Transportation's Bureau of Statistics. Their website releases monthly air travel statistics and summary report of all the flights information of previous month. Along with this report, they release the raw data which is openly available to be downloaded and analysed. The two datasets we use are taken from the raw data provided in the website. A brief description about the datasets is given below.

1. **Delayed Flights** contains information about all the cancelled or delayed flights ranging across the years 1987 to 2008. It has 29 columns which are described in the Table 1.
2. **Airports** is a reference dataset, which gives the airport names and locatons. The column description is shown in the Table 2.

The Delayed Flights dataset is around 250 mb in size and the airports dataset is around 250 kb since it is reference data.

7.2. Pig Script

Pig installation runs on the Hadoop hdfs system. The script needed for this analysis was based out of the idea from the blog post written by Sumit Anand [https://acadgild.com/blog/aviation-data-analysis-using-apache-pig/]. It takes the two datasets, performs basic joins, orders the rows and returns top five airports which cause most delays in all the years. This script was run on the namenode on Pig hadoop mode, with the input files given from hdfs dfs locations. This script analyses the data and outputs into hdfs location. The codes are written in Pig Latin language and can be run in Pig grunt shell.

Column Name	Description
Year	1987-2008
Month	1-12
DayofMonth	1-31
DayOfWeek	1 (Monday) - 7 (Sunday)
DepTime	actual departure time
CRSDepTime	scheduled departure time
ArrTime	actual arrival time
CRSArrTime	scheduled arrival time
UniqueCarrier	unique carrier code
FlightNum	flight number
TailNum	plane tail number
ActualElapsedTime	in minutes
CRSElapsedTime	in minutes
AirTime	in minutes
ArrDelay	arrival delay in minutes
DepDelay	departure delay in minutes
Origin	origin IATA airport code
Dest	destination IATA airport code
Distance	in miles
TaxiIn	taxi in time, in minutes
TaxiOut	taxi out time in minutes
Cancelled	was the flight cancelled?
CancellationCode	reason for cancellation
Diverted	1 = yes, 0 = no
CarrierDelay	in minutes
WeatherDelay	in minutes
NASDelay	in minutes
SecurityDelay	in minutes
LateAircraftDelay	in minutes

Table 1. Delayed Flights Column Information

Column Name	Description
iata	Airport Code
airport	Airport Name
city	Airport City
state	State Code
country	Country Code
lat	Lattitude
long	Longitude

Table 2. Airports Column Information

8. BENCHMARKING

Benchmarking is a process in software development which allows developers to determine the performance of their systems. This can be done for multiple reasons including looking for improvements and future planning. The basic requirements of a benchmark are that the environmental conditions of test should be same each time it is run, and the test should be repeatable.

The current paper presents multiple levels of benchmarking, starting from the time taken to deploy Hadoop cluster with three nodes, the time taken to move files from local to namenode and hdfs, time taken to run the pig code and time taken to move files from hdfs to namenode and local. Deployment of Hadoop cluster is done using Cloudmesh Client, and the rest of the tasks are done using Ansible. For all Ansibls steps, there are individual playbook files which can be called from separate Ansible commands. Each of these steps has been timed using Ubuntu Shell's "*time*" command.

Excluding the time taken to download, install and configure Cloudmesh Client, the time taken to create three vms with the "small" configuration, deploy Hadoop with three nodes, Spark and Pig addons is 8 minutes 46 seconds. This deployment was done from Oracle Virtual Box with a ram capacity of 8 GB.

The experiment was performed on the entire dataset, and the tests were repeated with 50% and 25% of the data. The results of the benchmarking tests are given in the tables 3, 4 and 5.

Task	Time Taken
Copy Data to cloud	1 min 33 sec
Copy Data to HDFS	23 sec
run pig script	3min 1 sec
Copy output to cloud	25 sec
Copy output to local	17 sec

Table 3. Benchmark Results for the entire data

9. SUMMARY

Airline industry is rapidly growing as the customers who take flights are increasing. Considering this trend, the cancellations and delays come into focus. It is imperative that Big Data technologies are deployed in this sector for quick results. This project aims at using Hadoop and Pig to run a basic analysis on Flight Delays data and benckmark the clouds' performance.

Task	Time Taken
Copy Data to cloud	54 sec
Copy Data to HDFS	20 sec
run pig script	1 min 55 sec
Copy output to cloud	24 sec
Copy output to local	16 sec

Table 4. Benchmark Results for the 50% of the data

Task	Time Taken
Copy Data to cloud	27 sec
Copy Data to HDFS	19 sec
run pig script	1 min 6 sec
Copy output to cloud	22 sec
Copy output to local	17 sec

Table 5. Benchmark Results for the 25% of the data

As we can see from the benchmark results, there is a drastic decrease in the first three steps (copy data to cloud, copy data to hdfs and run pig script) when the input file size is decreased. This is expected since there is less data for Hadoop to crunch. However in each of these cases, the output is top 3 airports, which is a textfile with three records. Thus we do not see much of a change in the time to copy the output (last two steps).

Overall, the time taken to run this entire process starting from copying data to namenode to getting output to local is taking less than 6 minutes for the entire dataset. Keeping in mind that our benchmark was for the "small" flavor of Chameleon Cloud (2GB ram) this is a good score, and if the cluster is vertically or horizontally scaled up, this analysis will yield results quicker. In order to take full advantage of Hadoop's parallel processing capabilities, it would be ideal if the number of nodes in the cluster were increased (horizontal scaling).

ACKNOWLEDGEMENTS

The author thanks Professor Gregor Von Lazewski for providing us with the guidance and topics for the Project. The author also thanks the AIs of Big Data Class for providing the technical support.

REFERENCES

AUTHOR BIOGRAPHIES

Harshit Krishnakumar is pursuing his MSc in Data Science from Indiana University Bloomington
Karthik Anbazhagan is pursuing his MSc in Data Science from Indiana University Bloomington