INFO7250- Engg Of Big Data

Final Project

INFO7250- Engg Of Big Data

**Ankit Yadav**

**NUID- 001271369**

**Northeastern University**

**CONTENTS**

|  |  |
| --- | --- |
| Topics | PAge No |
| 0. Overview about the dataset | **2** |
| 1. Analysis of Flight Data using MapReduce on Hadoop | **4** |
| 2. Analysis of Flight Data using Apache PIG on Hadoop | **19** |
| 3. References | **26** |
| 4. APPENDICES | **27** |

**0. Overview about the dataset**

I am using the data on **Airline On-Time Statistics and Delay Causes from**

<http://stat-computing.org/dataexpo/2009/the-data.html>

This is dataset containing information about airline schedule with following columns:

**Variable descriptions**

|  |  |  |
| --- | --- | --- |
|  | **Name** | **Description** |
| 1 | Year | 1987-2008 |
| 2 | Month | 1-12 |
| 3 | DayofMonth | 1-31 |
| 4 | DayOfWeek | 1 (Monday) - 7 (Sunday) |
| 5 | DepTime | actual departure time (local, hhmm) |
| 6 | CRSDepTime | scheduled departure time (local, hhmm) |
| 7 | ArrTime | actual arrival time (local, hhmm) |
| 8 | CRSArrTime | scheduled arrival time (local, hhmm) |
| 9 | UniqueCarrier | [unique carrier code](http://stat-computing.org/dataexpo/2009/supplemental-data.html) |
| 10 | FlightNum | flight number |
| 11 | TailNum | plane tail number |
| 12 | ActualElapsedTime | in minutes |
| 13 | CRSElapsedTime | in minutes |
| 14 | AirTime | in minutes |
| 15 | ArrDelay | arrival delay, in minutes |
| 16 | DepDelay | departure delay, in minutes |
| 17 | Origin | origin [IATA airport code](http://stat-computing.org/dataexpo/2009/supplemental-data.html) |
| 18 | Dest | destination [IATA airport code](http://stat-computing.org/dataexpo/2009/supplemental-data.html) |
| 19 | Distance | in miles |
| 20 | TaxiIn | taxi in time, in minutes |
| 21 | TaxiOut | taxi out time in minutes |
| 22 | Cancelled | was the flight cancelled? |
| 23 | CancellationCode | reason for cancellation (A = carrier, B = weather, C = NAS, D = security) |
| 24 | Diverted | 1 = yes, 0 = no |
| 25 | CarrierDelay | in minutes |
| 26 | WeatherDelay | in minutes |
| 27 | NASDelay | in minutes |
| 28 | SecurityDelay | in minutes |
| 29 | LateAircraftDelay | in minutes |

The reason of selections this data set is that it has many numbers of columns which will enable me to use various MapReduce algorithms studies in the course for different types of analysis.

Also, the data is evenly segregated in yearly basis. So, in case If I can am unable to load complete data in my computer then too I can do the same analysis on small portion of same data more easily.

**The complete data was downloaded using following script:**



**1. Analysis of Flight Data using MapReduce on Hadoop**

**1- Getting total count of all the data:**

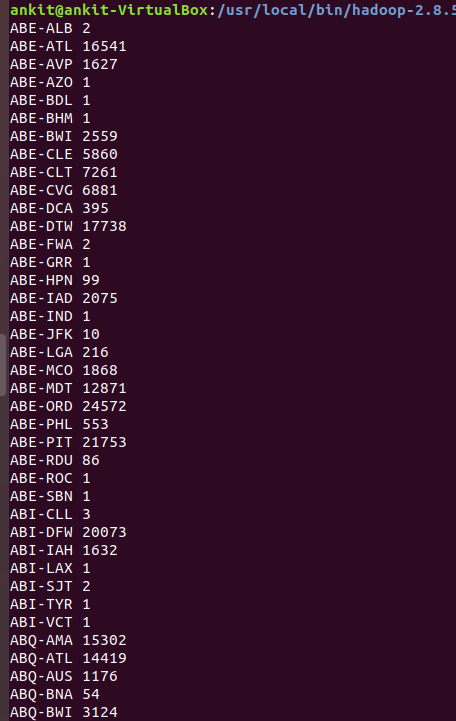
This is a very basic map reduce use case in which we count the whole data to get a sense of how many total records are there:

hadoop jar ~/Downloads/ProjectJars/count.jar hadoop.project.total\_count.MRCount /flight-data /FinalProjectMROutput/2-Total-Data-Count

The final count is: **123534970**

**2- Getting the total flights from all source destinations pairs in from 1987 to 2008:**

This was a huge data and MapReduce made this analysis quite simple and fast:



**3: Top 30 source destination pairs**

Sorting the above data to get top 30 most busy Source Destination pair:



**4: Delay in flight percentage**

We considered the delay greater than or equal to 15 minutes as delay . Now we need to count those flights which had delay greater than or equal to 15 minutes.

Delayed flight Count:

Percentage of departure delayed flights: Total Flight Count/ Departure Delayed Flight count

**= (19690422/123534970) \* 100 = 15.94 %**

So, this shows that the actual delay greater that 15 minutes is very less and generally flights depart on time.

Let’s check the same for arrival delay

Percentage of departure delayed flights: Total Flight Count/ Delayed Flight count

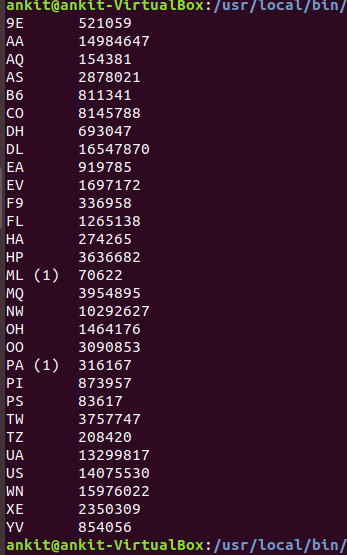
**= (24627925/123534970) \* 100 = 19.9 %**

**So, the delay in departure and arrival is between 15 to 20 % range.**

**So, it shows that overall flights are mostly on time from/to all source destination**

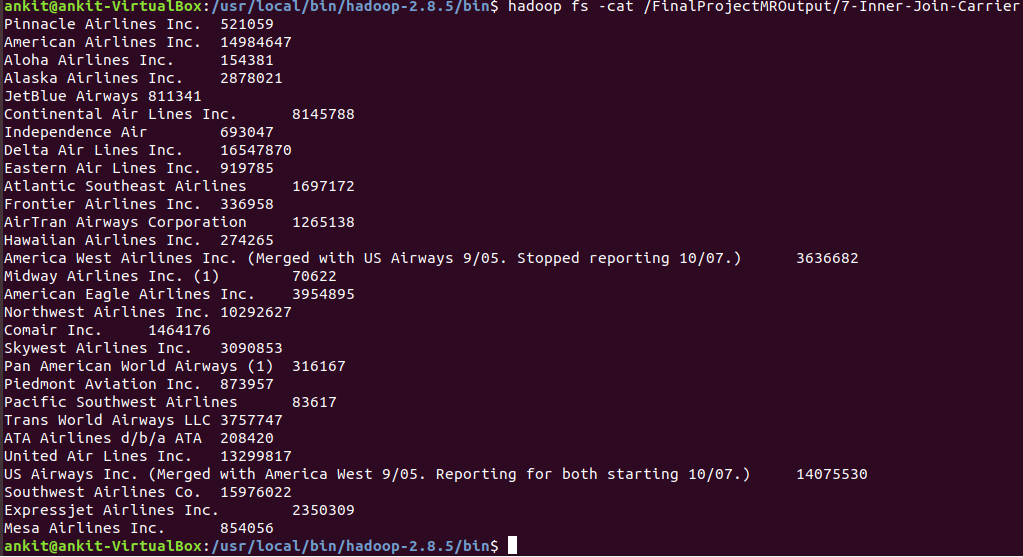
**5- Count of unique carrier’s flights**

The data for unique carriers are as follows:

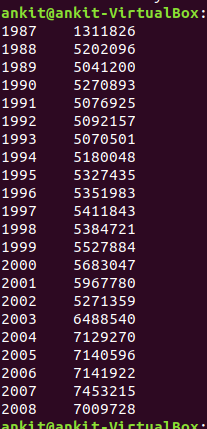


**6- Inner Join to get the full name for unique carriers**

We did inner join with between two files to get carrier names instead of carrier codes

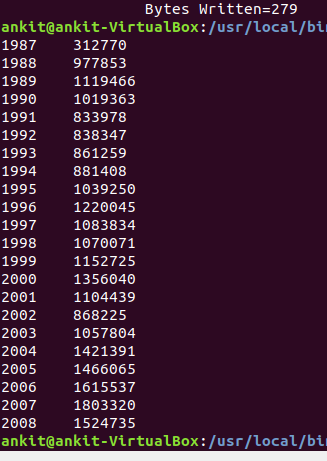


**7- Getting Flight data by year**

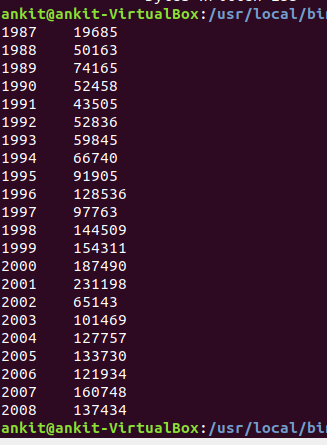


**8- Delayed flights per year**

In this we will check delayed flights per year( we will count flights as delayed only is the delay time if greater than equal to 15 minutes)

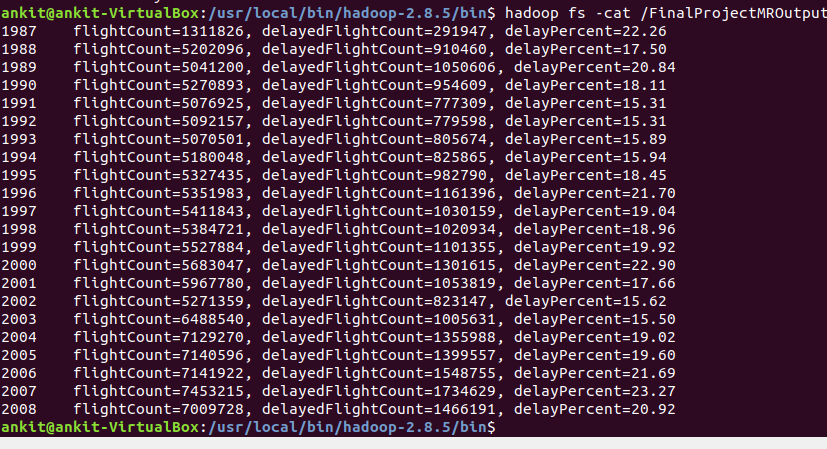


**9- Cancelled flights by year**



**10- Ratio of delayed flights per year to total flights**

We can get percentage of delayed flights per year also

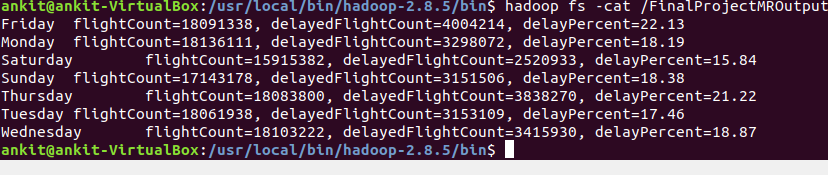


It shows that the years 1991-1994, 2002,2003 were best years to fly as they had least delayed flights (less than 16%).

Years with most delays were- 1987, 2000, 2007 with more than 22% flights delayed

**11- Total flights by day of week and ratio to delayed**

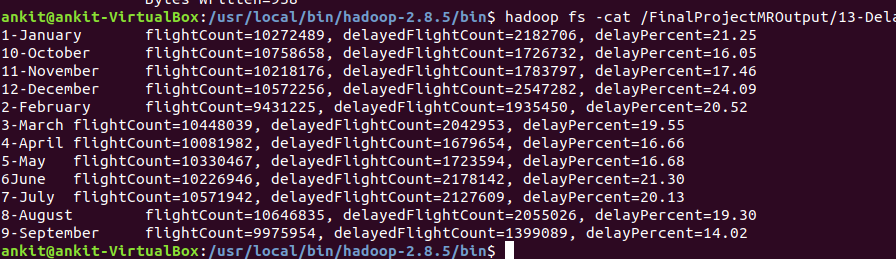
Following is the data of total flights , delayed flights and their ratio.



From this data we can infer that maximum delay is on Thursday and Fridays that is when weekends are starting .

Best day to fly are when the weekends ends like Saturday, Sunday or on weekdays

**12- Total flights by months of year and ratio to delayed**

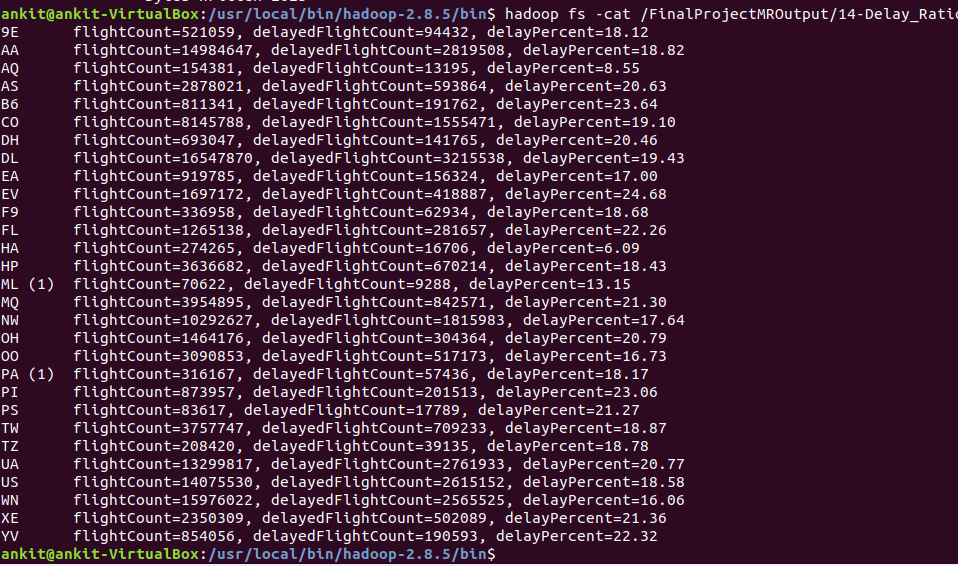


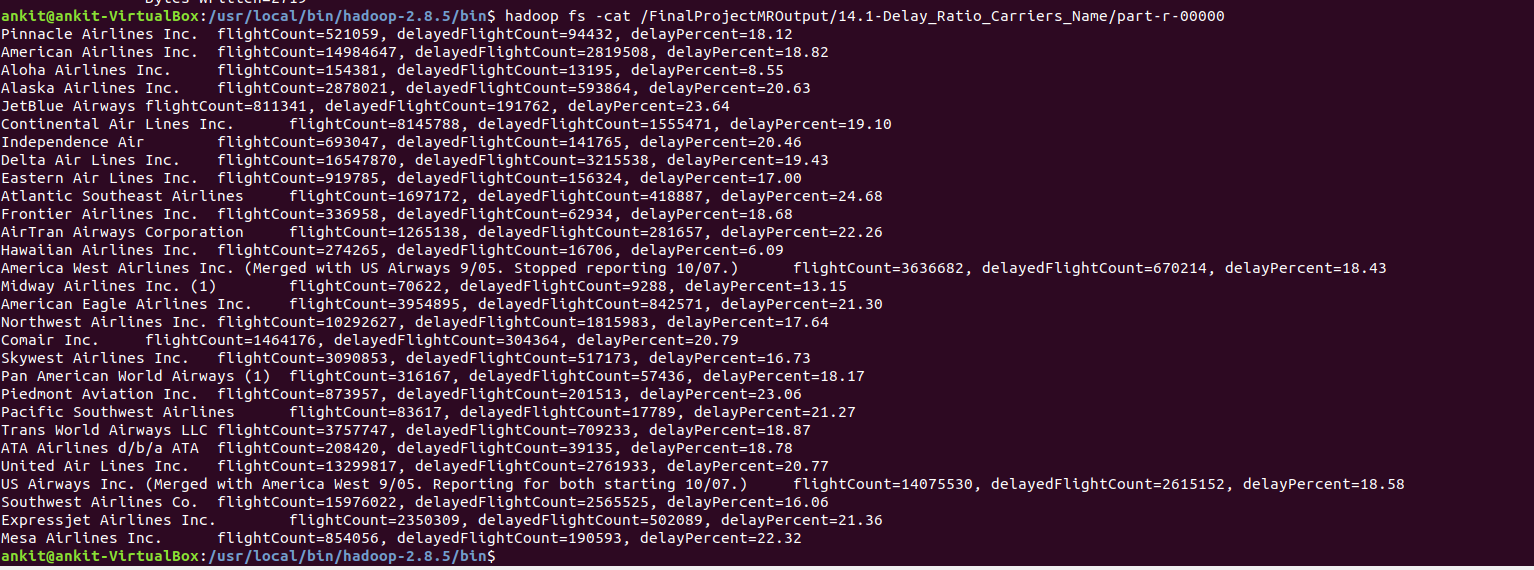
We can infer from this data that best months to fly was September with least delay- 14 %

Other good months were- April, May and October with delay – 16%

Worst month was December- 24% flights delayed. It may be due to big holidays season in December.

**13- Total delayed flights by flight carriers and ratio of delayed flights**



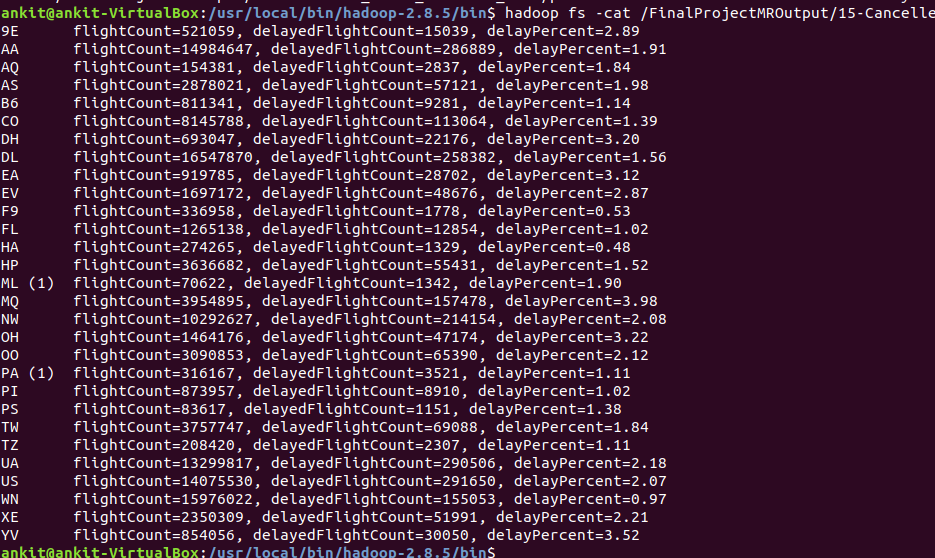


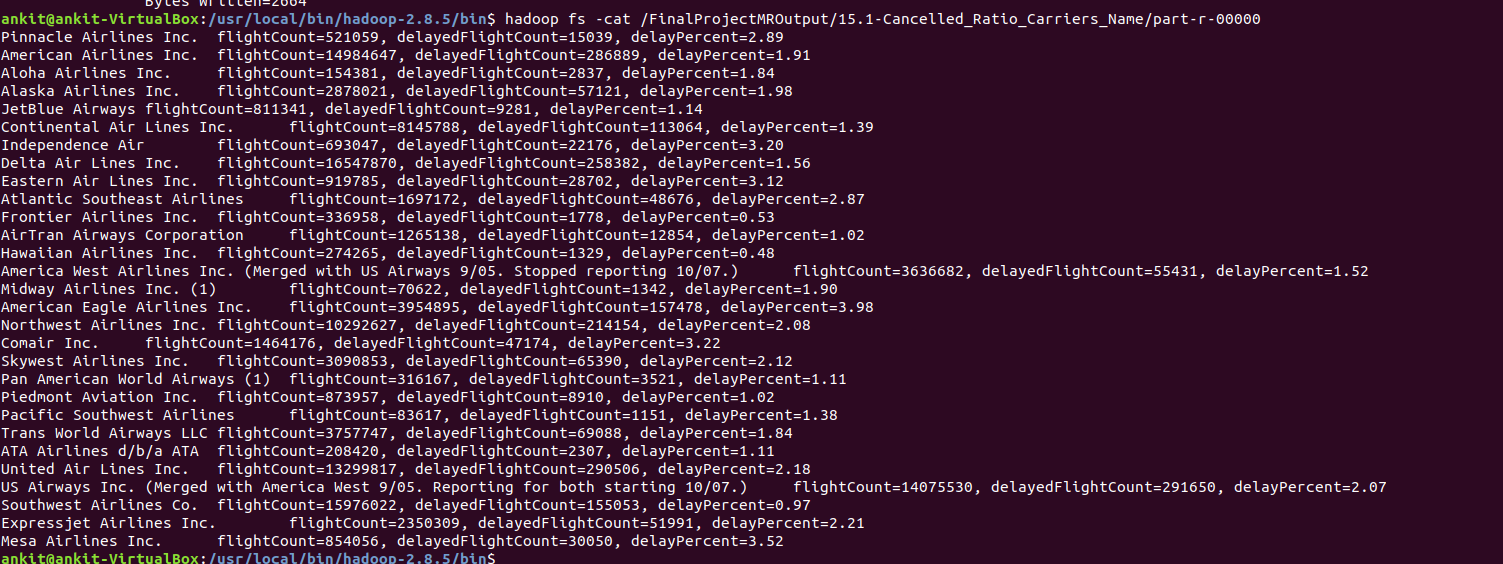
By using this analysis we can check which carriers are more prone to delays and can plan flights with those carriers who are less prone to delays.

Carriers with least delays- **Hawaiian Airlines, Aloha Airlines** with **6%** and **8%** flights delayed respectively.

Carriers with most delays- **JetBlue Airways, Atlantic Southeast Airlines** with around **24%** flights delayed.

**14- Total cancelled flights by flight carriers and ratio of cancelled flights**



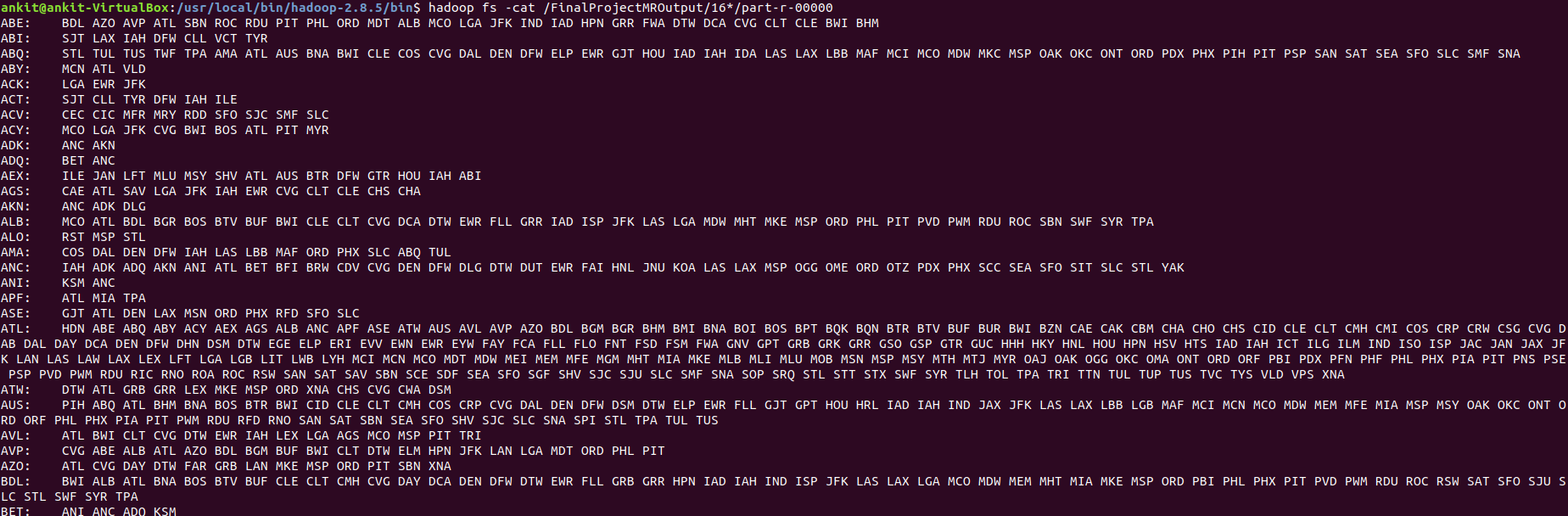


The number of cancelled flights are very less for almost all the carriers less than 4%.

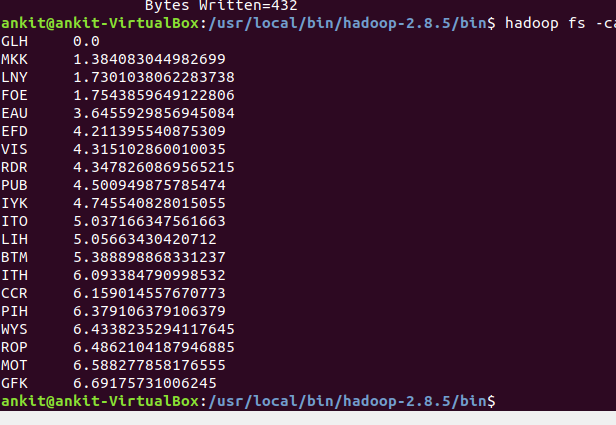
Among them best are **Frontier Airlines, Hawaiian Airlines** with **0.5%** cancelled flights and worst are **American Eagle Airlines, Mesa Airlines** with more than **3.5%** cancelled flights.

**15- Inverted index for all source and destination**

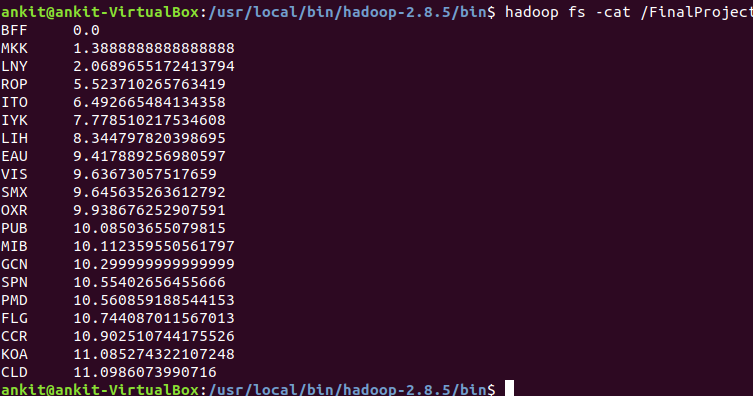
This data can help to search for all the destination stations from a particular source stations.



**16- Top 20 best source station with least departure delayed flight percent**



**17- Top 20 best destination station with least arrival delayed flight percent**



**18- Delay groups- grouping amount of flights per delay groups**



Between 1 hour and 2 hour **2.93**

Between 15 and 30 minutes **7.33**

Between 30 minutes and 1 hour **4.55**

Less than 15 Minutes **84.06**

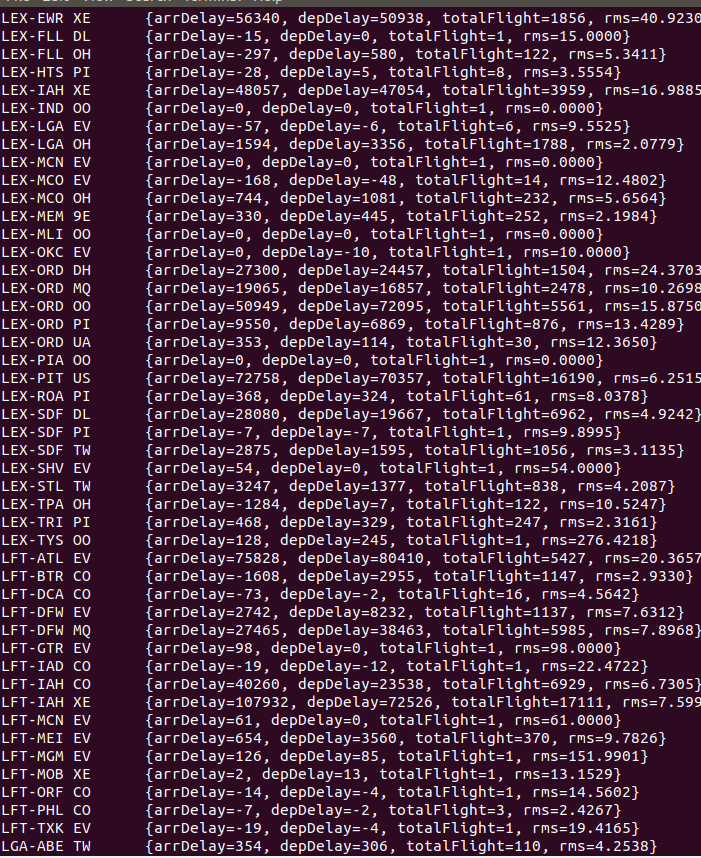
More than 2 hours **1.13**

**19- Recommendation system- Best carrier for a source destination route**

I first calculate average arrival delay and average departure delay for each source destination pair for each carrier.

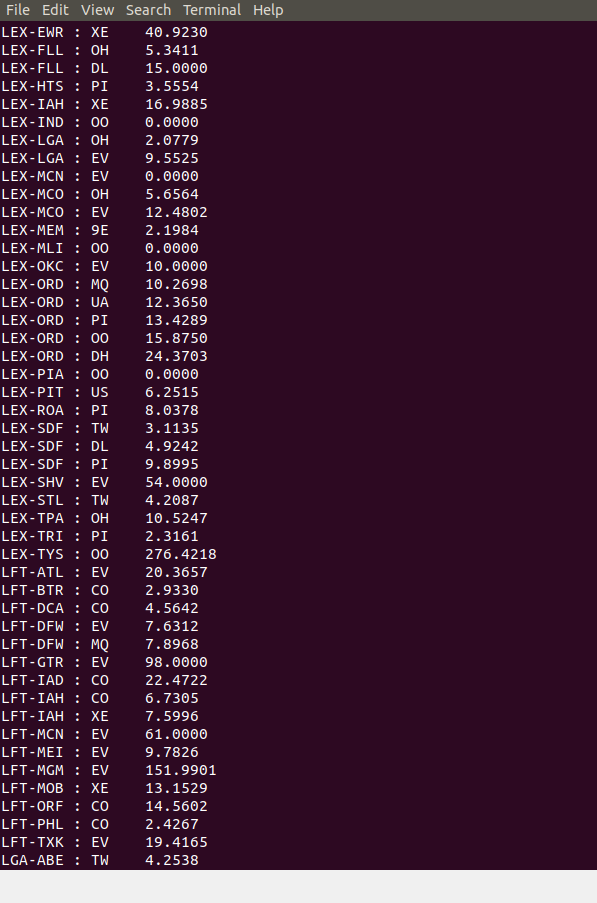
After this I calculate root mean square value for average arrival delay and average departure delay.

**Rms =**

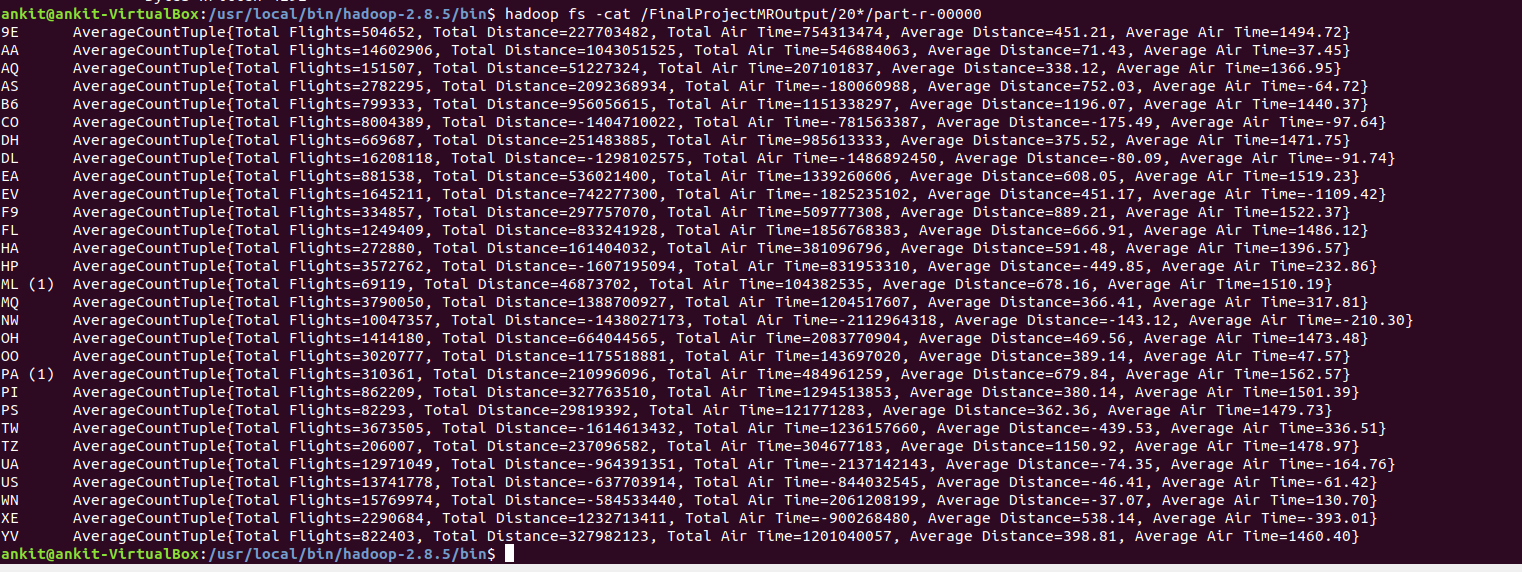


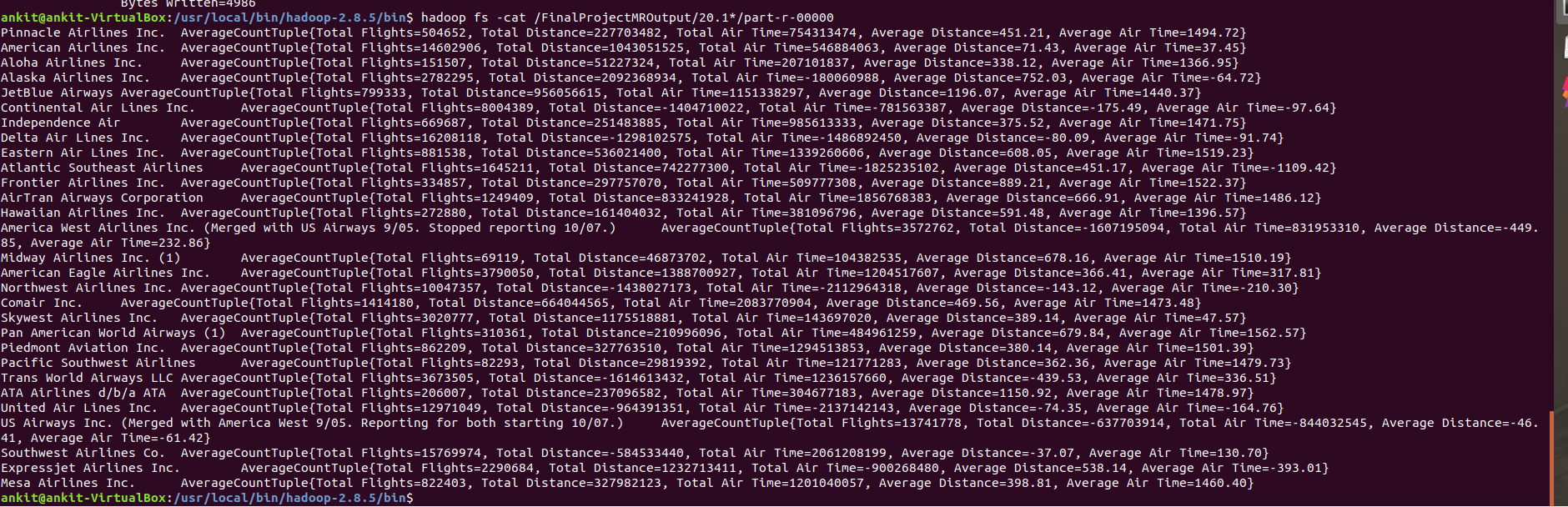
After this I sorted the carriers for all source destination pair in ascending order by RMS value.

It gives user a recommendation for choosing a carrier between a source destination pair with least arrival and/or departure delay.

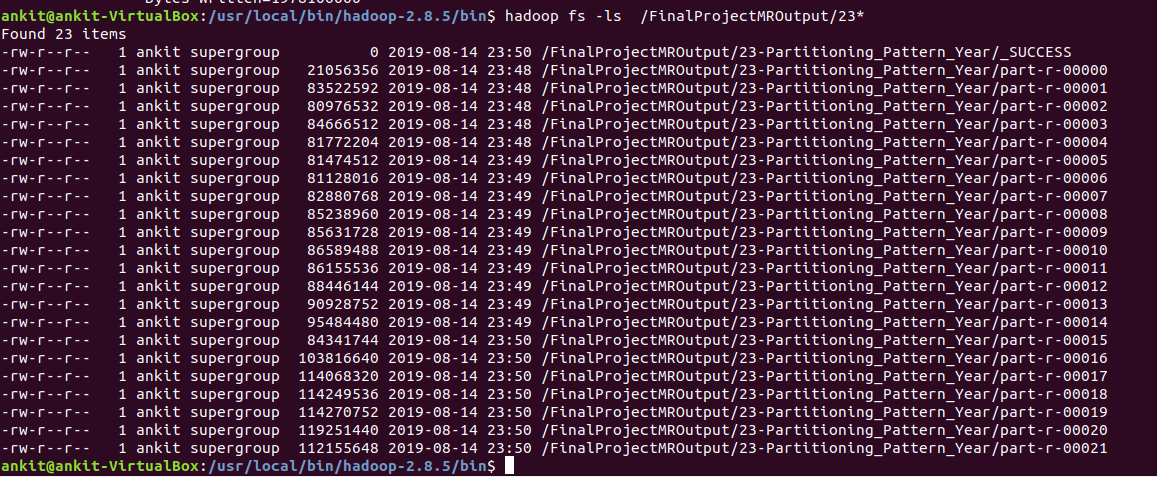


**20- Average flying distance per carrier**





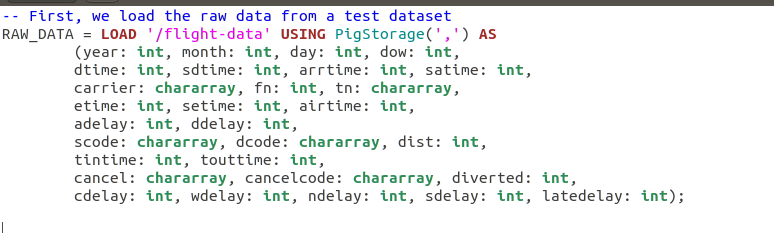
**21- Using partitioning pattern on the basis of year**



**2. Analysis of Flight Data using Apache PIG on Hadoop**

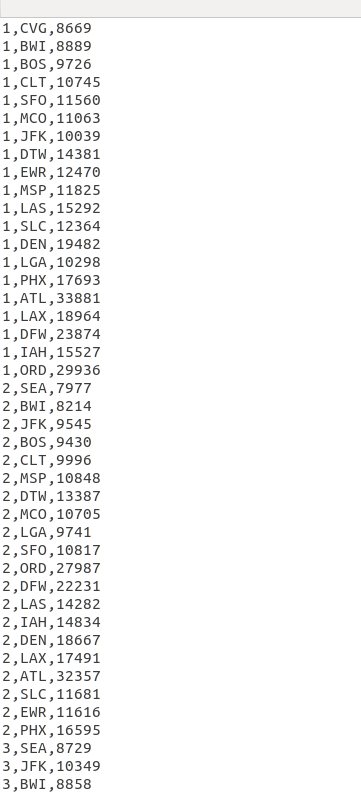
**Analysis 1: Top 20 cities by total volume of flights**

What are the busiest cities by total flight traffic? For each airport code I computed the number of inbound, outbound and all flights.

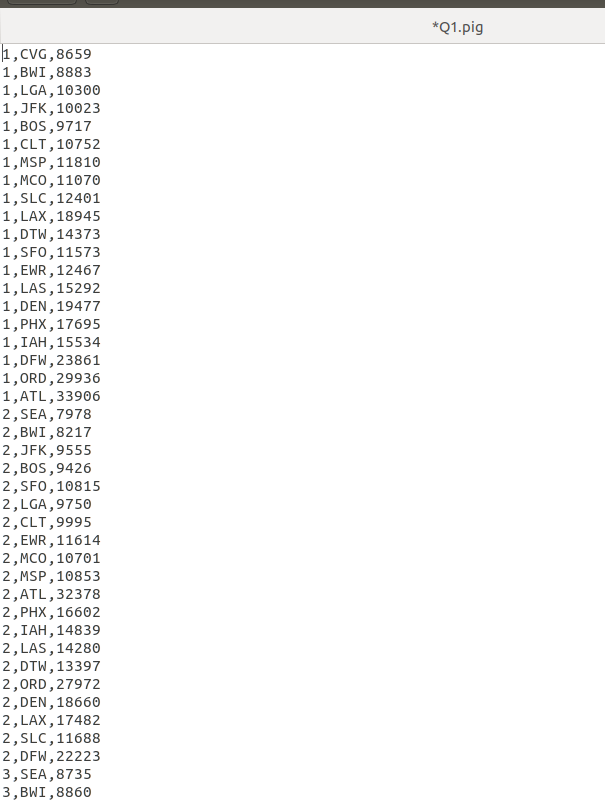




**OUTPUT: INBOUNT\_TOP**

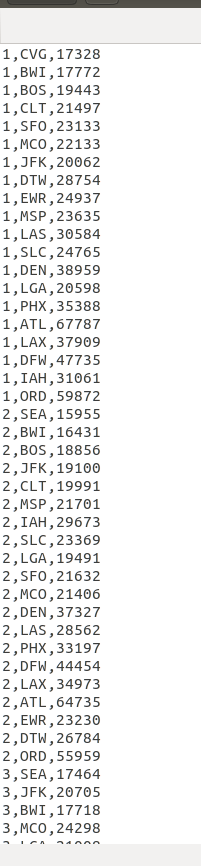


**OUTPUT: OUTBOUND TOP**



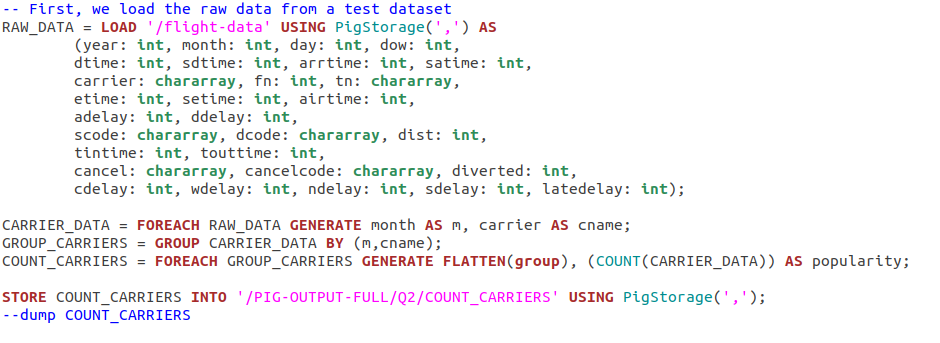


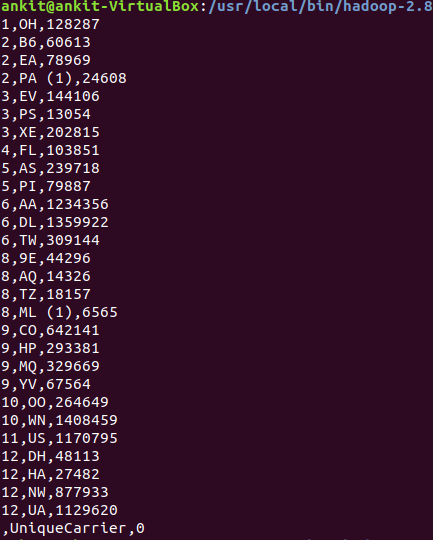
**Output Monthly Traffic Top**



**Analysis 2: Carrier Popularity**

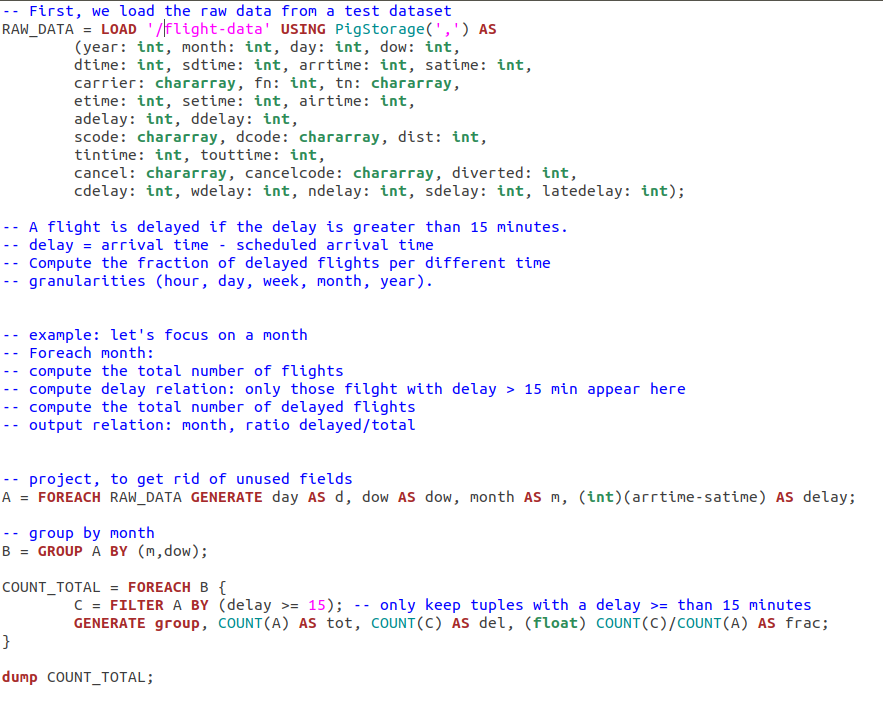
Computing the volume -- total flights -- over each year, by carrier. The carriers are ranked by their median volume (over the 10-year span).

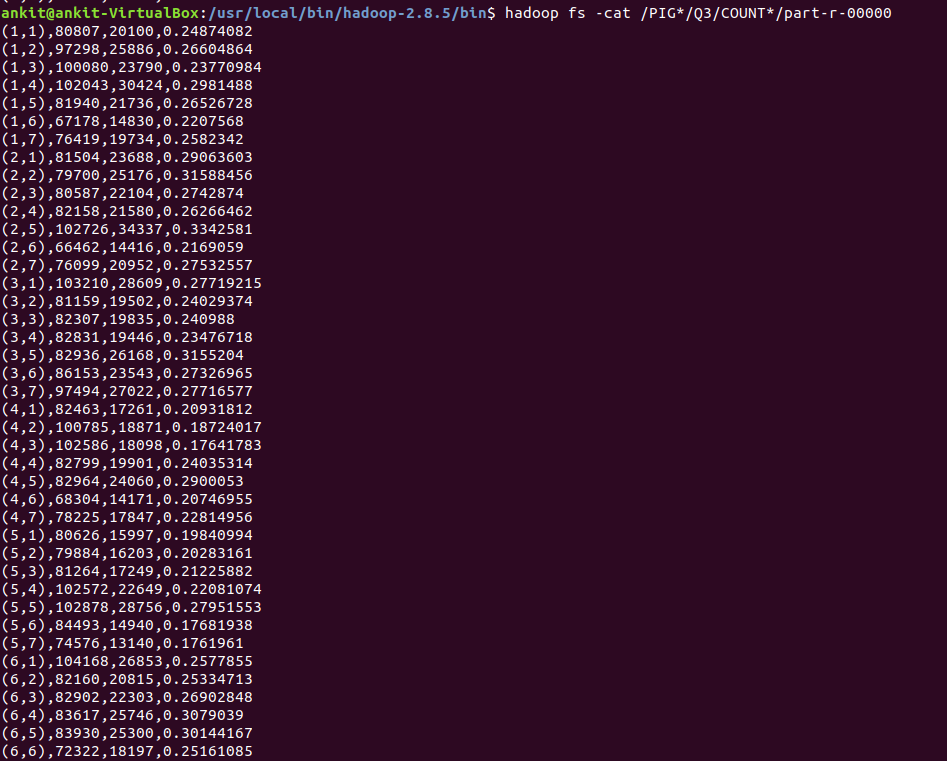




**Analysis 3: Proportion of Flights Delayed**

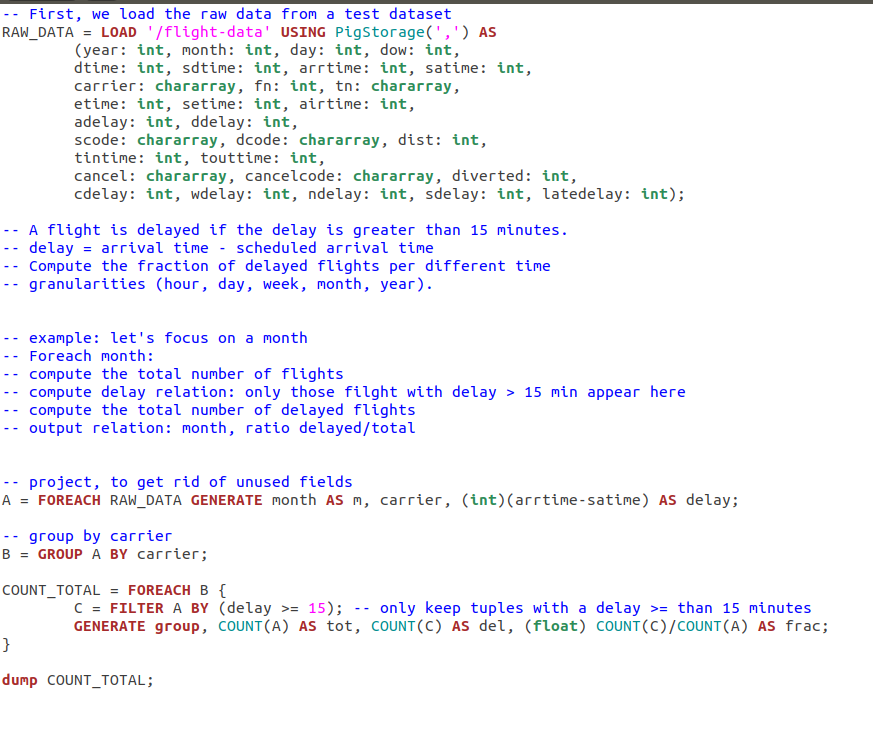
A flight is delayed if the delay is greater than 15 minutes. I am calculating the fraction of delayed flights per different time limits (hour, day, week, month, year).





**Analysis 4: Carrier Delays**

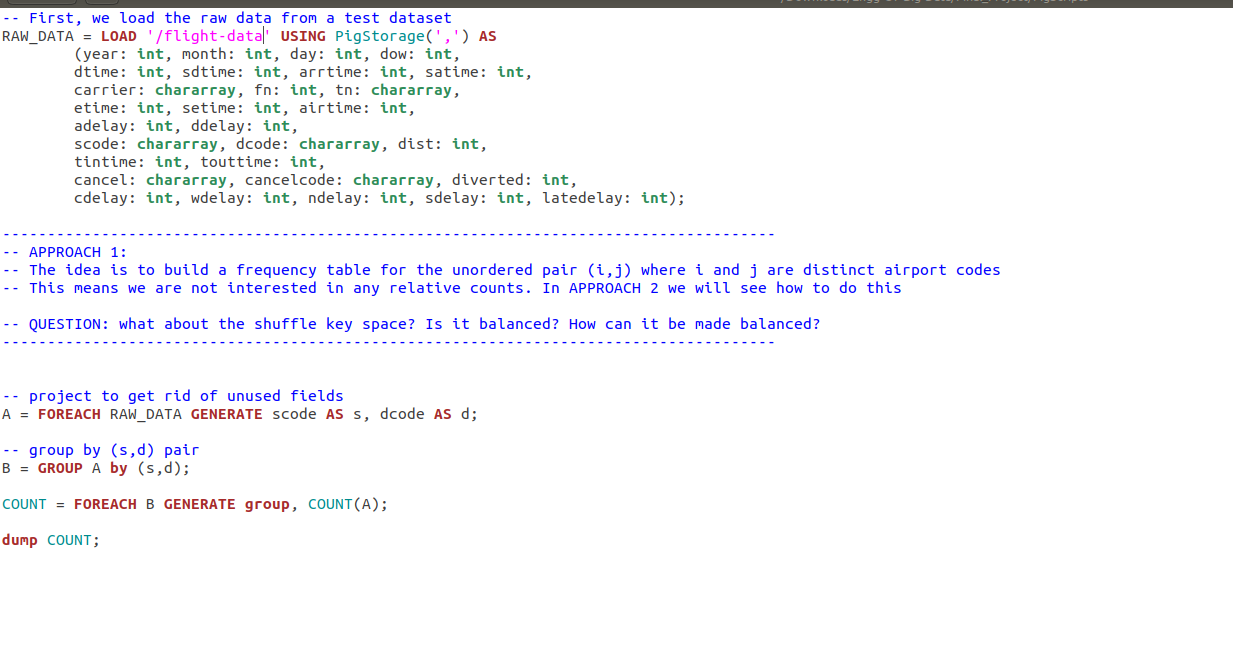
Calculating the proportion of delayed flights by carrier, ranked by carrier, at different time (hour, day, week, month year). Again, a flight is delayed if the delay is greater than 15 minutes.

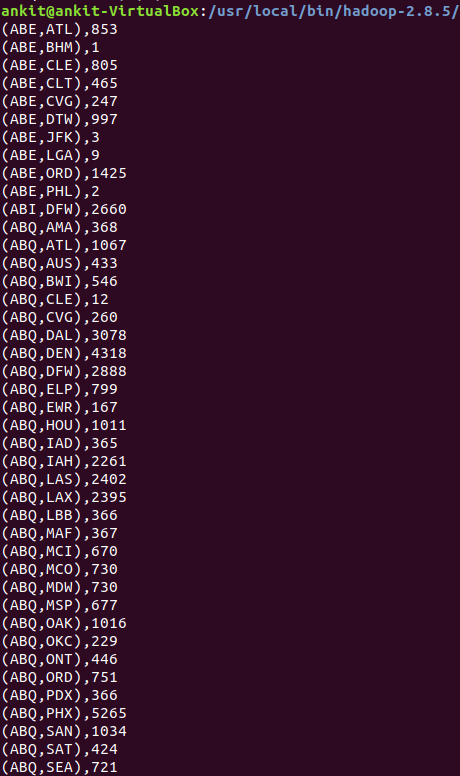




**Analysis 5: Routes that were most busy**

The approach is to create a frequency table for the unordered pair (m,n) where m and n are distinct airport codes which will help in finding the routes that are more busy.



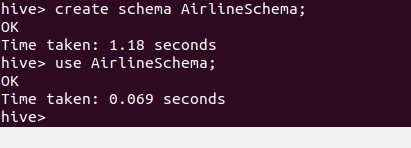


**3. Analysis of Flight Data using Apache HIVE on Hadoop**

**Creating schema for flight data**

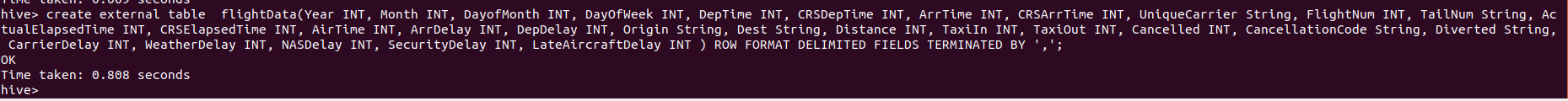
create schema AirlineSchema;

use AirlineSchema;



**Creating table to store flight data**

create external table flightData(Year INT, Month INT, DayofMonth INT, DayOfWeek INT, DepTime INT, CRSDepTime INT, ArrTime INT, CRSArrTime INT, UniqueCarrier String, FlightNum INT, TailNum String, ActualElapsedTime INT, CRSElapsedTime INT, AirTime INT, ArrDelay INT, DepDelay INT, Origin String, Dest String, Distance INT, TaxiIn INT, TaxiOut INT, Cancelled INT, CancellationCode String, Diverted String, CarrierDelay INT, WeatherDelay INT, NASDelay INT, SecurityDelay INT, LateAircraftDelay INT ) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';



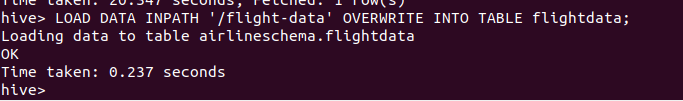
**Set some hive properties**

SET hive.exec.dynamic.partition = true;

SET hive.exec.dynamic.partition.mode = nonstrict;

**Load flight data from HDFS into table**

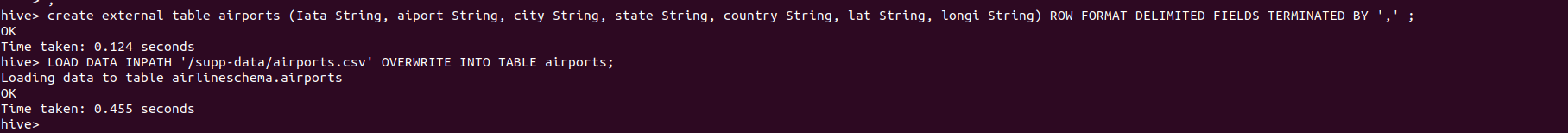
LOAD DATA INPATH '/flight-data' OVERWRITE INTO TABLE flightData;



**Create table and load airports data from HDFS**

create external table airports (Iata String, aiport String, city String, state String, country String, lat String, longi String) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' ;

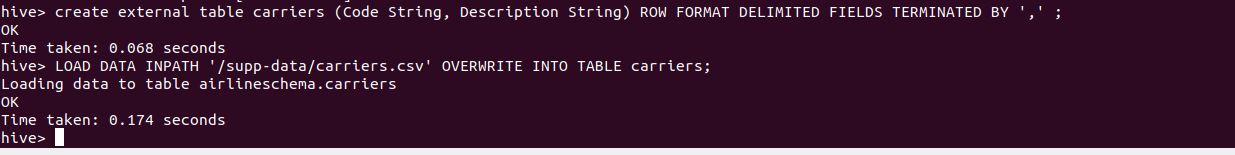
LOAD DATA INPATH '/supp-data/airports.csv' OVERWRITE INTO TABLE airports;



**Create table and load carrier’s data from HDFS**

create external table carriers (Code String, Description String) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' ;

LOAD DATA INPATH '/supp-data/carriers.csv' OVERWRITE INTO TABLE carriers;



Now, All data is loaded. So, we can proceed to analysis.

**1: FLIGHTS THAT TRAVELLED LESS THAN OR MORE THAN 500 AIRTIME**

INSERT OVERWRITE DIRECTORY '/ HiveMROutput /1.1' select count(\*) from flightData where AirTime > 500;

INSERT OVERWRITE DIRECTORY '/ HiveMROutput /1.2' select count(\*) from flightData where AirTime >= 500;

**2. COUNT OF ALL THE FLIGHTS THAT WERE ON TIME WHILE ARRIVING AND DEPARTURE**

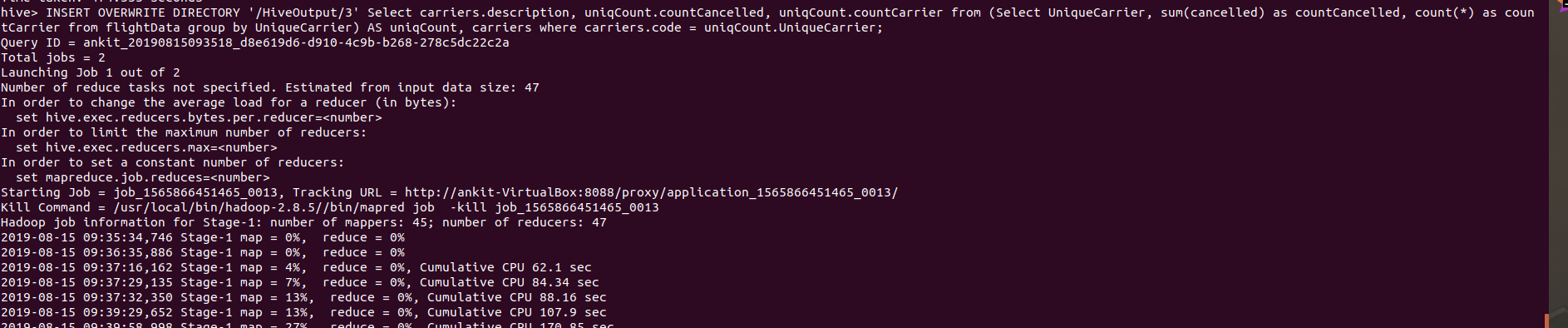
INSERT OVERWRITE DIRECTORY '/HiveMROutput/2' select Year,Month,DayofMonth,Origin,Dest,AirTime,Distance,TaxiIn,TaxiOut from flightData where DepTime<=CRSDepTime and ArrTime<=CRSArrTime;

**3: COUNT OF ALL THE FLIGHTS THAT TOOK MORE THAN 30 MINS TO DEP AND ARRIVAL DELAY**

select count(\*) from flightData where ArrDelay + DepDelay >30;

**4: COUNT OF FLIGHTS FOR EACH CARRIER**

INSERT OVERWRITE DIRECTORY '/ HiveMROutput /3' Select carriers.description, uniqCount.countCancelled, uniqCount.countCarrier from (Select UniqueCarrier, sum(cancelled) as countCancelled, count(\*) as countCarrier from flightData group by UniqueCarrier) AS uniqCount, carriers where carriers.code = uniqCount.UniqueCarrier;



**4. References**

1. <https://learning.oreilly.com/library/view/mapreduce-design-patterns/9781449341954/>
2. <https://gitlab.eurecom.fr/yonghui.feng/clouds-lab>
3. <https://learning.oreilly.com/library/view/data-algorithms/9781491906170/ch01.html>
4. <http://cs229.stanford.edu/proj2013/MathurNagaoNg-PredictingFlightOnTimePerformance.pdf>

**5. APPENDIX**

The code of this project can be found at GitHub repository for this project at

<https://github.com/ankit08015/Engg-Of-Big-Data/tree/master/Final_Project>