GIOVANNI BATTISTA FIOCCOLA

Cloud Computing Project – Task 1

by University of Illinois at Urbana-Champaign

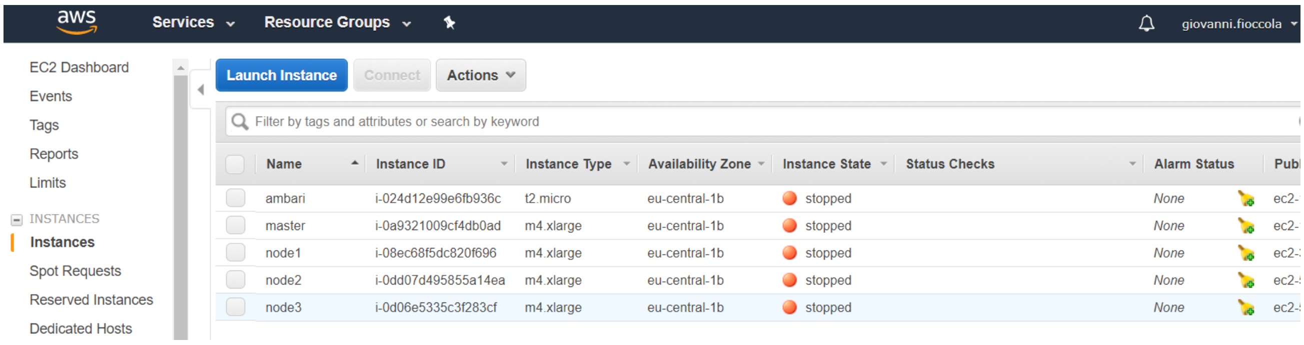
I. INTRODUCTION

The goal of the Capstone Project is to apply the knowledge acquired through the Cloud computing courses, in order to solve real-world Cloud computing challenges. In particular, the aim of task 1 is to:

- Extract and clean the transportation dataset from the US BTS (Bureau of Transportation Statistics) that is hosted as an Amazon EBS volume snapshot. The dataset contains data and statistics from the US Department of Transportation on aviation, maritime, highway, transit, rail, pipeline, bike/pedestrian and more in CSV format, but only the aviation part of the dataset is useful for this task. Finally, resulting data must be stored in HDFS (Hadoop Distributed File System).
- Answer a set of questions by using Apache Hadoop and/or Spark Streaming, which are two batch processing systems. The results related to questions from Group 2 and question 3.2 are stored in Cassandra.

II. VIRTUALIZATION ENVIRONMENT

In order to quickly deploy a complete Apache Hadoop stack, Apache Ambari has been used. Ambari is a completely open source management platform for provisioning, managing and monitoring Apache Hadoop clusters by using a graphical user interface. The Apache Hadoop cluster has been deployed on the EC2 Cloud service offered by Amazon Web Services (AWS), and consists of 1 NameNode (Master) and 3 DataNodes (Slaves – Node1, Node2 and Node3). The NameNode is a master server that manages the filesystem namespace and regulates access to files by clients; it executes namespace operations such as opening, closing, and renaming files and directories. The NameNode also determines the mapping of blocks to DataNodes. The DataNodes manage storage attached to the nodes that they run on, and they are responsible for serving read and write requests from the filesystem clients. The DataNodes also perform block creation, deletion, and replication upon instruction from the NameNode. Finally, the Ambari server runs on a fifth EC2 instance.



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Publ
ambari	i-024d12e99e6fb936c	t2.micro	eu-central-1b	● stopped		None	ec2-
master	i-0a9321009cf4db0ad	m4.xlarge	eu-central-1b	● stopped		None	ec2-
node1	i-08ec68f5dc820f696	m4.xlarge	eu-central-1b	● stopped		None	ec2-
node2	i-0dd07d495855a14ea	m4.xlarge	eu-central-1b	● stopped		None	ec2-
node3	i-0d06e5335c3f283cf	m4.xlarge	eu-central-1b	● stopped		None	ec2-

Figure 1 - Virtualization environment

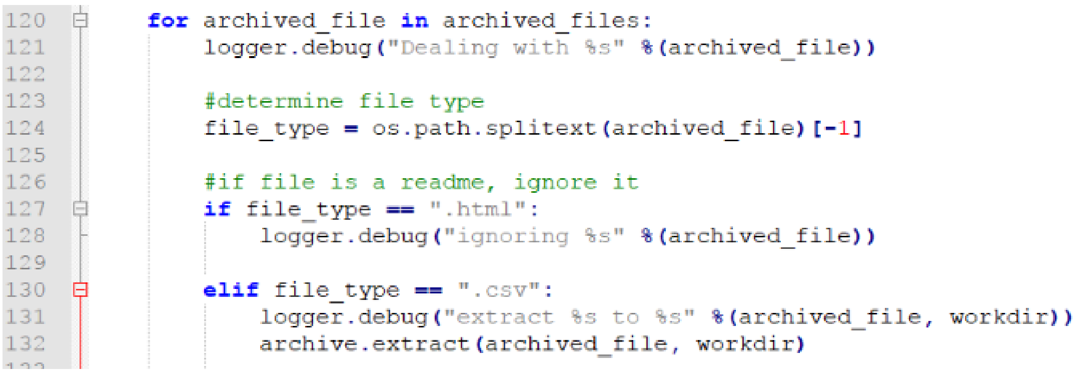
III. DATA ACQUISITION AND FILTERING

A new EBS volume from the existing EBS snapshot (ID snap-e1608d88 for Linux/Unix in the us-east-1 region) of the transportation dataset has been created and attached to the Master EC2 instance. By using the lsblk command, it is possible to view the available disk devices and their mount points. A new folder has been created (/mnt/data) and the EBS volume has been attached to the filesystem:

```
$ sudo mkdir /mnt/data/
```

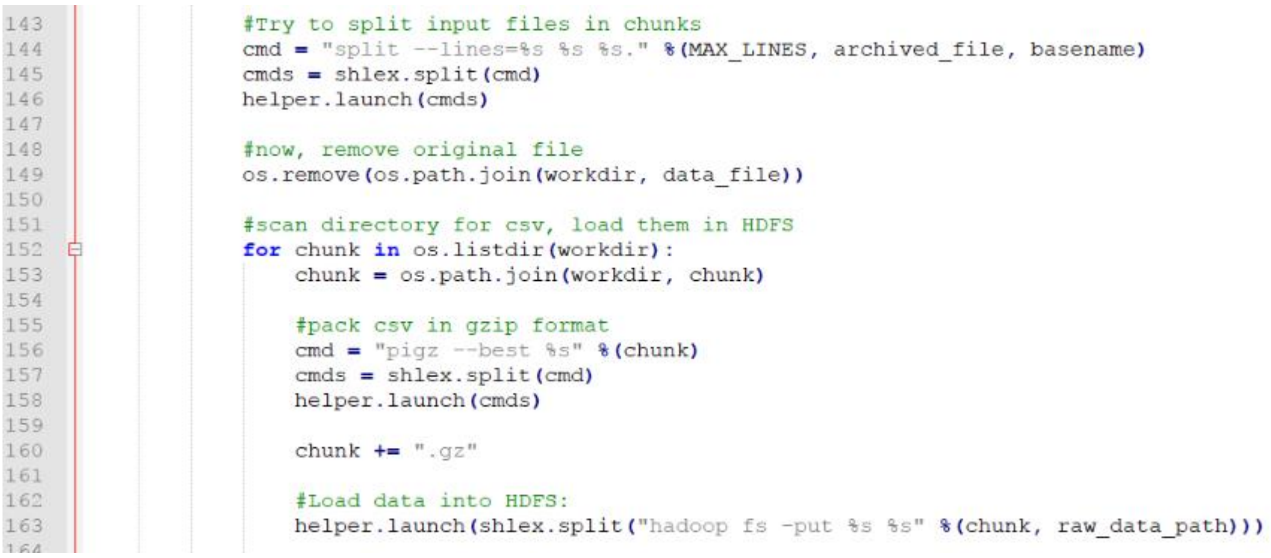
$ sudo mount /dev/xvdf /mnt/data

The attached EBS data volume includes data and statistics from the US Department of Transportation on aviation, maritime, highway, transit, rail, pipeline, bike/pedestrian and more, but only the aviation part of the dataset is useful for this task. Therefore, after mounting and attaching the volume, only the aviation directory has been considered, and more specifically the “airline ontime” data that contains on-time performance for each flight. A python script processes all subdirectories of “airline ontime” and the zip archives included in the subdirectories, ignores the “readme” files and searches for CSV extensions.



```
for archived_file in archived_files:
    logger.debug("Dealing with %s" %(archived_file))

    #determine file type
    file_type = os.path.splitext(archived_file)[-1]

    #if file is a readme, ignore it
    if file_type == ".html":
        logger.debug("ignoring %s" %(archived_file))

    elif file_type == ".csv":
        logger.debug("extract %s to %s" %(archived_file, workdir))
        archive.extract(archived_file, workdir)
```

Finally, the script divides input files in chunks and compresses CSV files in “.gz” format by using Pigz, performing a compression through concurrent threads, in order to use multiple processors and cores. Data is sent to HDFS by using put command.



```
#Try to split input files in chunks
cmd = "split --lines=%s %s %s." %(MAX_LINES, archived_file, basename)
cmds = shlex.split(cmd)
helper.launch(cmds)

#now, remove original file
os.remove(os.path.join(workdir, data_file))

#scan directory for csv, load them in HDFS
for chunk in os.listdir(workdir):
    chunk = os.path.join(workdir, chunk)

    #pack csv in gzip format
    cmd = "pigz --best %s" %(chunk)
    cmds = shlex.split(cmd)
    helper.launch(cmds)

    chunk += ".gz"

    #Load data into HDFS:
    helper.launch(shlex.split("hadoop fs -put %s %s" %(chunk, raw_data_path)))
```

Using mkdir in Apache Hadoop needs the "hadoop file permissions"; hdfs is a user that has permissions to create folders, so the new user is added to hdfs group. Then the folder is created and finally the owner of the folder is changed:

```
$ sudo usermod -aG hdfs $(whoami)

$ sudo -u hdfs hadoop fs -mkdir -p /user/giovanni/capstone/airline_ontime/raw_data/

$ sudo -u hdfs hadoop fs -chown -R $(whoami) /user/giovanni/
```

Then the python script is executed:

```
$ python AirlineToHDFS.py --input_path=/mnt/data/aviation/airline_ontime/ \
  --output_path=/user/giovanni/capstone/airline_ontime/raw_data/
```

A pig script allows to load data from HDFS, and the loaded data includes the header (the first row). The script gets rid of the header and only the following fields are considered:

FlightDate, AirlineID, FlightNum, Origin, OriginCityName, OriginStateName, Dest, DestCityName, DestStateName, CRSDepTime, DepDelay, CRSArrTime, ArrDelay, Cancelled, CancellationCode, Diverted, CRSElapsedTime, ActualElapsedTime, AirTime, Distance

Finally, filtered data are put into a new folder in HDFS:

```
$ pig -x mapreduce -p input=/user/giovanni/capstone/airline_ontime/raw_data/ \
 -p filtered=/user/giovanni/capstone/airline_ontime/filtered_data/  load_ontime.pig
```

IV. SYSTEM INTEGRATION

In order to install Cassandra, the following steps were performed:

- Add the DataStax Community repository to the /etc/apt/sources.list.d/cassandra.sources.list:
  ```
  $ echo "deb http://debian.datastax.com/community stable main" | sudo tee -a /etc/apt/sources.list.d/cassandra.sources.list
  ```
- Add the DataStax repository key to the aptitude trusted keys:
  ```
  $ curl -L http://debian.datastax.com/debian/repo_key | sudo apt-key add –
  ```
- Install the package:
  ```
  $ sudo apt-get update
  $ sudo apt-get install dsc20 cassandra=2.0.17
  ```
- Edit /etc/cassandra/cassandra.yaml:
  ```
  $ sudo nano /etc/cassandra/cassandra.yaml
  ```
  - The default cassandra.yaml uses the local, loopback address as its listen and Thrift addresses:
    ```
    listen_address: locahost
    rpc_address: localhost
    ```
  - listen_address and rpc_address must be set to the IP address of the master:
    ```
    listen_address: 172.31.44.199
    rpc_address: 172.31.44.199
    ```
  - Similarly for the the seed information:
    ```
    seeds: "node1,node2,node3"
    ```

The developers of Apache Spark have given thoughtful consideration to Python as a language of choice for data analysis. They have developed the PySpark API for working with RDDs in Python. Furthermore, it supports use of powerful IPythonshell instead of the builtin Python REPL. RDD (Resilient Distributed Dataset) is the basic abstraction in Spark and represents an immutable, partitioned collection of elements that can be operated on in parallel.

The developers of IPython have invested considerable effort in building the IPython Notebook, a system inspired by Mathematica that allows to create “executable documents.” IPython Notebooks can integrate formatted text (Markdown), executable code (Python), mathematical formulae (LaTeX), and graphics/visualizations (matplotlib) into a single document that captures the flow of an exploration and can be exported as a formatted report or an executable script.

The steps followed in order to use iPython with Spark are the following:

- First an IPython profile for use with PySpark has been created:
  ```
  ipython profile create pyspark
  ```

- The file ~/.ipython/profile_pyspark/startup/00-pyspark-setup.py with the following contents has been created:

```
import os
import sys

# Configure the environment
if 'SPARK_HOME' not in os.environ:
    os.environ['SPARK_HOME'] = '/usr/hdp/current/spark-client'

# Create a variable for our root path
SPARK_HOME = os.environ['SPARK_HOME']

# Add the PySpark/py4j to the Python Path
sys.path.insert(0, os.path.join(SPARK_HOME, "python", "lib"))
sys.path.insert(0, os.path.join(SPARK_HOME, "python"))

# Described here: http://blog.cloudera.com/blog/2014/08/how-to-use-ipython-notebook-
with-apache-spark/
sys.path.insert(0, os.path.join(SPARK_HOME, "python/lib/py4j-0.8.2.1-src.zip"))
execfile(os.path.join(SPARK_HOME, 'python/pyspark/shell.py'))
```

- An IPython shell with the profile we just created is started:
  `$ ipython --profile spark`

Finally, pyspark-cassandra is installed. This module provides python support for Apache Spark's Resillient Distributed Datasets from Apache Cassandra CQL rows using Cassandra Spark Connector within PySpark, both in the interactive shell and in python programmes submitted with spark-submit.

All MapReduce jobs are submitted to YARN, in order to prepare a dataset that can be processed by Spark. Finally, the Spark Cassandra Connector package allows to move loaded DataFrames from Spark to Apache Cassandra.

V. QUESTION 1.1

A pig script allows to load data from the filtered transportation dataset. Then, tuples that have the same Origin key are grouped together, and tuples that have the same Dest key are grouped together too. The number of elements that have the same Origin key are counted, the number of elements that have the same Dest key are counted too, and results are put in the same table. These results are grouped by airport code and the occurrences are summed up:

```
Total_counts = FOREACH United_group GENERATE FLATTEN($0), SUM(United_counts.count) AS count;
```

Finally, airports are ordered by occurrences thanks to multiple concurrent reduce tasks:

```
$ Popular = ORDER Total_counts BY count DESC PARALLEL 4;
```

This is the pig script:

```
$ pig -x mapreduce -p filtered=/user/giovanni/capstone/airline_ontime/filtered_data/  get_top10.pig
```

VI. QUESTION 1.2

A pig script allows to load data from the filtered transportation dataset. Then, cancelled and diverted flights are excluded. The tuples that have the same AirlineID key are grouped together, and for each airline the average arrival delay is computed:

```
airline_delays = FOREACH airlines GENERATE FLATTEN($0), AVG(flights.ArrDelay) AS avg_delay;
```

Finally, airlines are ordered by arrival delays thanks to multiple concurrent reduce tasks.

This is the pig script:

```
$ pig -x mapreduce -p filtered=/user/giovanni/capstone/airline_ontime/filtered_data/  top10_airlines.pig
```

VII. QUESTION 2.1

A pig script allows to load data from the filtered transportation dataset. The airlines lookup dictionary, previously downloaded (http://www.transtats.bts.gov/Download_Lookup.asp?Lookup=L_AIRLINE_ID) and put in hadoop filesystem, is loaded. A replicated join is performed:

```
lookup = LOAD '$lookup' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',', 'NO_MULTILINE', 'NOCHANGE', 'SKIP_INPUT_HEADER') AS (AirlineIDsec:chararray,Airline:chararray);

joined = JOIN filtered BY AirlineID, lookup BY AirlineIDsec USING 'replicated';
```

Then, cancelled and diverted flights are excluded. Only the following fields are considered: Origin, AirlineID, Airline, DepDelay:

```
flights = FOREACH arrived_flights GENERATE Origin, AirlineID, Airline, DepDelay;
```

The tuples that have the same Origin, AirlineID and Airline keys are grouped together, and then the average departure delay for each carrier is computed:

```
carrier_delays = FOREACH carrier GENERATE group.Origin, group.AirlineID, group.Airline, AVG(flights.DepDelay) AS avg_delay;
```

The carrier_delays tuples that have the same Origin key are grouped together. Finally, for each origin, carriers are ordered by their average departure delays by means of concurrent multiple reduce tasks. Data are then stored into Cassandra.

This is the pig script:

```
$ pig -x mapreduce -p filtered=/user/giovanni/capstone/airline_ontime/filtered_data/  -p
lookup=/user/giovanni/capstone/lookup/Lookup_AirlineID.csv -p
results=/user/giovanni/capstone/airline_ontime/top10_carriersByAirport/ top10_carriersByAirport.pig
```

Cassandra table definition is the following:

```
CREATE KEYSPACE capstone WITH REPLICATION = {'class': 'SimpleStrategy', 'replication_factor': 1};

USE capstone;

CREATE TABLE carriersbyairport ( origin TEXT, airlineid INT, airline TEXT, depdelay FLOAT, PRIMARY KEY(origin, depdelay));
```

To show the results, the following script is used:

```
$ cqlsh -f top10_carriersByAirport.cql 172.31.44.199
```

It includes for example:

```
SELECT origin, airlineid, depdelay, airline FROM carriersbyairport WHERE origin = 'CMI';)
```

VIII. QUESTION 2.2

A pig script allows to load data from the filtered transportation dataset. Then, cancelled and diverted flights are excluded. Only the following fields are considered: Origin, Dest, DepDelay:

```
flights = FOREACH arrived_flights GENERATE Origin, Dest, DepDelay;
```

The tuples that have the same Origin and Dest keys are grouped together, and then the average departure delay for each path is computed:

```
path_delays = FOREACH path GENERATE group.Origin, group.Dest, AVG(flights.DepDelay) AS avg_delay;
```

The path_delays tuples that have the same Origin key are grouped together. Finally, for each origin, path_delays are ordered by their average delays by means of concurrent multiple reduce tasks. Data are then stored into Cassandra.

This is the pig script:

```
$ pig -x mapreduce -p filtered=/user/giovanni/capstone/airline_ontime/filtered_data/ \
 -p results=/user/giovanni/capstone/airline_ontime/top10_airportsByAirport/ top10_airportsByAirport.pig
```

Cassandra table definition is the following:

```
CREATE KEYSPACE capstone WITH REPLICATION = {'class': 'SimpleStrategy', 'replication_factor': 1};

USE capstone;

CREATE TABLE airportsbyairport ( origin TEXT, destination TEXT, depdelay FLOAT, PRIMARY KEY(origin, depdelay));
```

To show the results, the following script is used:

```
$ cqlsh -f top10_ airportsByAirport.cql 172.31.44.199
```

It includes for example:

```
SELECT origin, destination, depdelay FROM airportsbyairport WHERE origin = 'CMI';
```

IX. QUESTION 2.3

A pig script allows to load data from the filtered transportation dataset. The airlines lookup dictionary, previously downloaded (http://www.transtats.bts.gov/Download_Lookup.asp?Lookup=L_AIRLINE_ID) and put in hadoop filesystem, is loaded. A replicated join is performed:

```
lookup = LOAD '$lookup' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',', 'NO_MULTILINE', 'NOCHANGE', 'SKIP_INPUT_HEADER') AS (AirlineIDsec:chararray,Airline:chararray);

joined = JOIN filtered BY AirlineID, lookup BY AirlineIDsec USING 'replicated';
```

Then, cancelled and diverted flights are excluded. Only the following fields are considered: Origin, Dest, AirlineID, Airline, ArrDelay:

```
flights = FOREACH arrived_flights GENERATE Origin, Dest, AirlineID, Airline, ArrDelay;
```

The tuples that have the same Origin, Dest, AirlineID and Airline keys are grouped together, and then the average arrival delay for each carrier is computed:

```
carrier_delays = FOREACH carrier GENERATE group.Origin, group.Dest, group.AirlineID, group.Airline, AVG(flights.ArrDelay) AS avg_delay;
```

The carrier_delays tuples that have the same Origin and Dest keys are grouped together. Finally, for each origin, carriers are ordered by their average arrival delays by means of concurrent multiple reduce tasks. Data are then stored into Cassandra.

This is the pig script:

```
$ pig -x mapreduce -p filtered=/user/giovanni/capstone/airline_ontime/filtered_data/  -p
lookup=/user/giovanni/capstone/lookup/Lookup_AirlineID.csv -p
results=/user/giovanni/capstone/airline_ontime/top10_carriersByPath/top10_carriersByPath.pig
```

Cassandra table definition is the following:

```
CREATE KEYSPACE capstone WITH REPLICATION = {'class': 'SimpleStrategy', 'replication_factor': 1};
```

```
USE capstone;
```

```
CREATE TABLE carriersbypath ( origin TEXT, destination TEXT, airlineid INT, airline TEXT, arrdelay FLOAT, PRIMARY KEY(origin, destination, arrdelay));
```

To show the results, the following script is used:

```
$ cqlsh -f top10_ carriersByPath.cql 172.31.44.199
```

It includes for example:

```
SELECT origin, destination, airlineid, arrdelay, airline FROM carriersbypath WHERE origin = 'CMI' AND destination = 'ORD';
```

X. QUESTION 3.1

A pig script allows to load data from the filtered transportation dataset. Then, tuples that have the same Origin key are grouped together, and tuples that have the same Dest key are grouped together too. The number of elements that have the same Origin key are counted, the number of elements that have the same Dest key are counted too, and results are put in the same table. These results are grouped by airport code and the occurrences are summed up. Finally, airports are ordered by occurrences thanks to multiple concurrent reduce tasks:

```
$ Popular = ORDER Total_counts BY count DESC PARALLEL 4;
```

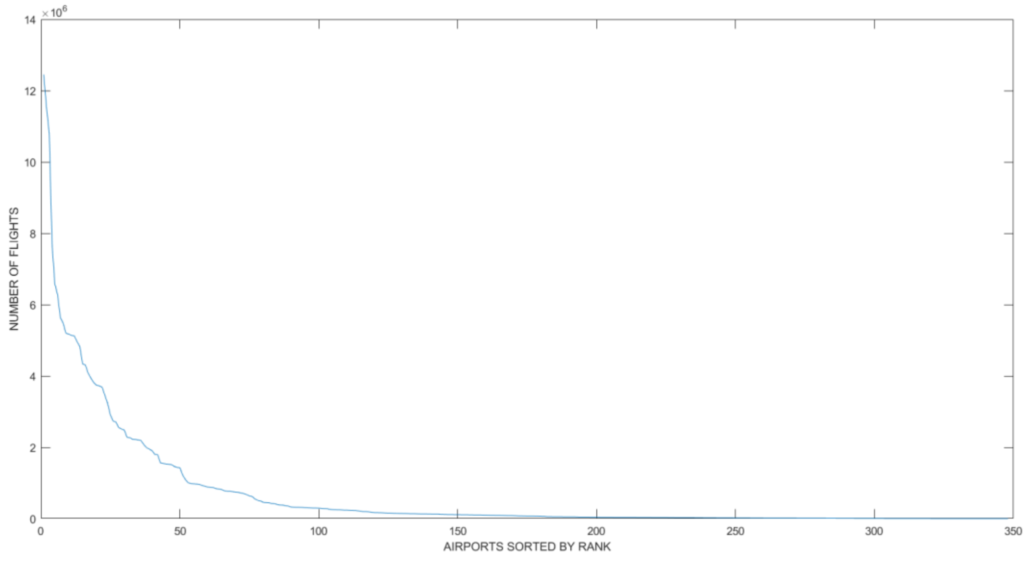
The RANK operator works with the count field. It sorts the relation on this field and prepends the rank value to each tuple. The rank of a tuple is one plus the number of different rank values preceding it. DESC sorts in descending order.

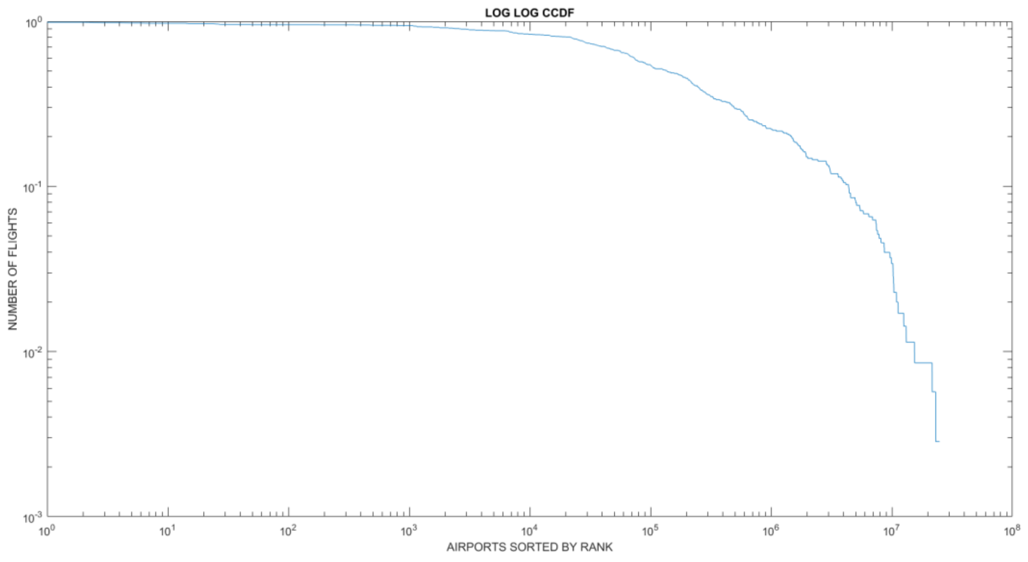
```
Ranked = RANK Popular BY count DESC;
```

This is the pig script:

```
$ pig -x mapreduce -p filtered=/user/giovanni/capstone/airline_ontime/filtered_data/  -p
popular=/user/giovanni/capstone/airline_ontime/popular/ get_popular.pig
```

Zipf's law is an empirical law formulated by using mathematical statistics, and it refers to the fact that many types of data, studied in the physical and social sciences, can be approximated with a Zipfian distribution, one of a family of related discrete power law probability distributions. Zipf's law states that, given some corpus of natural language utterances, the frequency of any word is inversely proportional to its rank in the frequency table. Therefore, the airport with a higher rank should have a double number of flights with respect to the next airport with lower rank. The CCDF of the popularity for airports should be a straight line if it had been a Zipf distribution. Instead, the popularity of the airports follows a lognormal distribution.





XI. QUESTION 3.2

A python script allows to read the CSV Data of filtered transportation dataset into an RDD:

```
ontime_data = sc.textFile(DATA_DIR).map(split).map(parse)
```

Then, cancelled and diverted flights are excluded:

```
arrived_data = ontime_data.filter(lambda x: x.Cancelled is False and x.Diverted is False and x.FlightDate.year == 2008 and x.CRSDepTime is not None and x.CRSArrTime is not None and x.ArrDelay is not None)
```

In order to extract date, departure/arrival time scheduled, and the delay, only the following fields are considered:

```
FlightData = arrived_data.map(lambda m: (m.FlightDate, m.Origin, m.Dest, m.FlightNum, m.CRSDepTime, m.CRSArrTime, m.ArrDelay))
```

Data are filtered because Tom wants his flights scheduled to depart airport X before 12:00 PM local time and to depart airport Y after 12:00 PM local time:

```
path1 = FlightData.filter(lambda (flightdate, origin, dest, flightnum, crsdeptime, crsarrtime, arrdelay): crsdeptime < time(hour=12, minute=00))
```

```
path2 = FlightData.filter(lambda (flightdate, origin, dest, flightnum, crsdeptime, crsarrtime, arrdelay): crsdeptime > time(hour=12, minute=00))
```

It is possible to transform path by Origin and Dest keys, in order to join path1 destination with path2 origin. Since the the second leg of the journey (flight Y-Z) must depart two days after the first leg (flight X-Y), it is possible to subtract two days from the second dataset, then join by two keys:

```
destPath1 = path1.keyBy(lambda (flightdate, origin, dest, flightnum, crsdeptime, crsarrtime, arrdelay): (dest, flightdate))
```

```
originPath2 = path2.keyBy(lambda (flightdate, origin, dest, flightnum, crsdeptime, crsarrtime, arrdelay): (origin, flightdate-timedelta(2)))
```

It is possible to do a join between the two paths. The airport Y (path1 dest, path2 origin) is the key. When a join is performed, RDD with the same keys are located on the same nodes. Calling a map to transform values could be inefficient since the new keys will invalidate partitioning by keys. So, dates are filtered out before calling a new key map phase:

```
joinedPath = destPath1.join(originPath2).values()
```

Tom wants to arrive at each destination with as little delay as possible, so it is possible to sum delays for each of the two paths, then it is possible to order by such values:

```
twoDaysPathFlat = joinedPath.map(lambda ((flightdate1, origin1, dest1, flightnum1, crsdeptime1, crsarrtime1, arrdelay1), (flightdate2, origin2, dest2, flightnum2, crsdeptime2, crsarrtime2, arrdelay2)): ((flightdate1, flightdate2, origin1, dest1, dest2), [(flightdate1, origin1, dest1, flightnum1, crsdeptime1, crsarrtime1, arrdelay1, flightdate2, origin2, dest2, flightnum2, crsdeptime2, crsarrtime2, arrdelay2, arrdelay1+arrdelay2)]))
```

First, flights from the same dataset with the same path are summed up. aggregatebykey first sums partitions on the same node, then intra-nodes:

```
twoDaysPathSorted = twoDaysPathFlat.aggregateByKey([], getBest, getBest)
```

Finally, data are stored into Cassandra.

The PYSPARK_CASSANDRA variable has been set to the pyspark directory:

```
PYSPARK_CASSANDRA=/home/giovanni/capstone/pyspark-cassandra/target
```

Then jar files are added:

```
export PYSPARK_SUBMIT_ARGS="--jars ${PYSPARK_CASSANDRA}/pyspark_cassandra-0.1.5.jar \
  --driver-class-path ${PYSPARK_CASSANDRA}/pyspark_cassandra-0.1.5.jar  \
  --py-files ${PYSPARK_CASSANDRA}/pyspark_cassandra-0.1.5-py2.7.egg \
  --conf spark.cassandra.connection.host=172.31.44.199"
```

The pyspark_cassandra library has been imported in order to use Cassandra:

```
import pyspark_cassandra
```

The application is launched by using the spark-submit command:

```
spark-submit $PYSPARK_SUBMIT_ARGS --master yarn --executor-cores=3 --num-executors 3 --driver-memory=3G --executor-memory=4G best2pathsnokafka.py
```

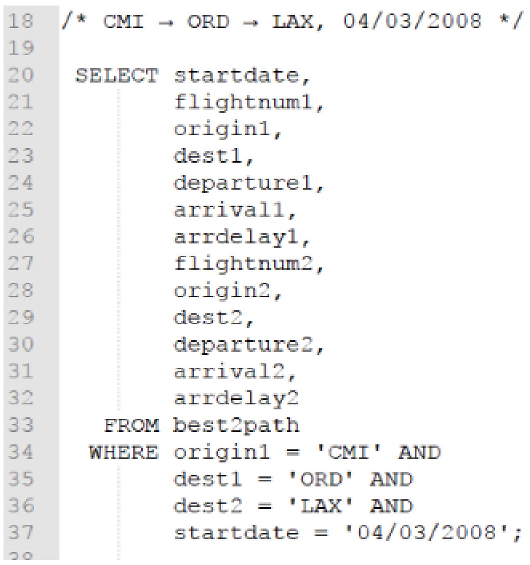
Cassandra table definition is the following:

```
CREATE KEYSPACE capstone WITH REPLICATION = {'class': 'SimpleStrategy', 'replication_factor': 1};

USE capstone;

CREATE TABLE best2path (startdate TEXT, flightnum1 INT, origin1 TEXT, dest1 TEXT, departure1 TIMESTAMP, arrival1 TIMESTAMP, arrdelay1 FLOAT, flightnum2 INT, origin2 TEXT, dest2 TEXT, departure2 TIMESTAMP, arrival2 TIMESTAMP, arrdelay2 FLOAT, PRIMARY KEY(origin1, dest1, dest2, startdate));
```

To show the results, the following script is used:

```
$ cqlsh -f best2paths.cql 172.31.44.199
```

It includes for example:



```
/* CMI → ORD → LAX, 04/03/2008 */

 SELECT startdate,
        flightnum1,
        origin1,
        dest1,
        departure1,
        arrival1,
        arrdelay1,
        flightnum2,
        origin2,
        dest2,
        departure2,
        arrival2,
        arrdelay2
   FROM best2path
  WHERE origin1 = 'CMI' AND
        dest1 = 'ORD' AND
        dest2 = 'LAX' AND
        startdate = '04/03/2008';
```

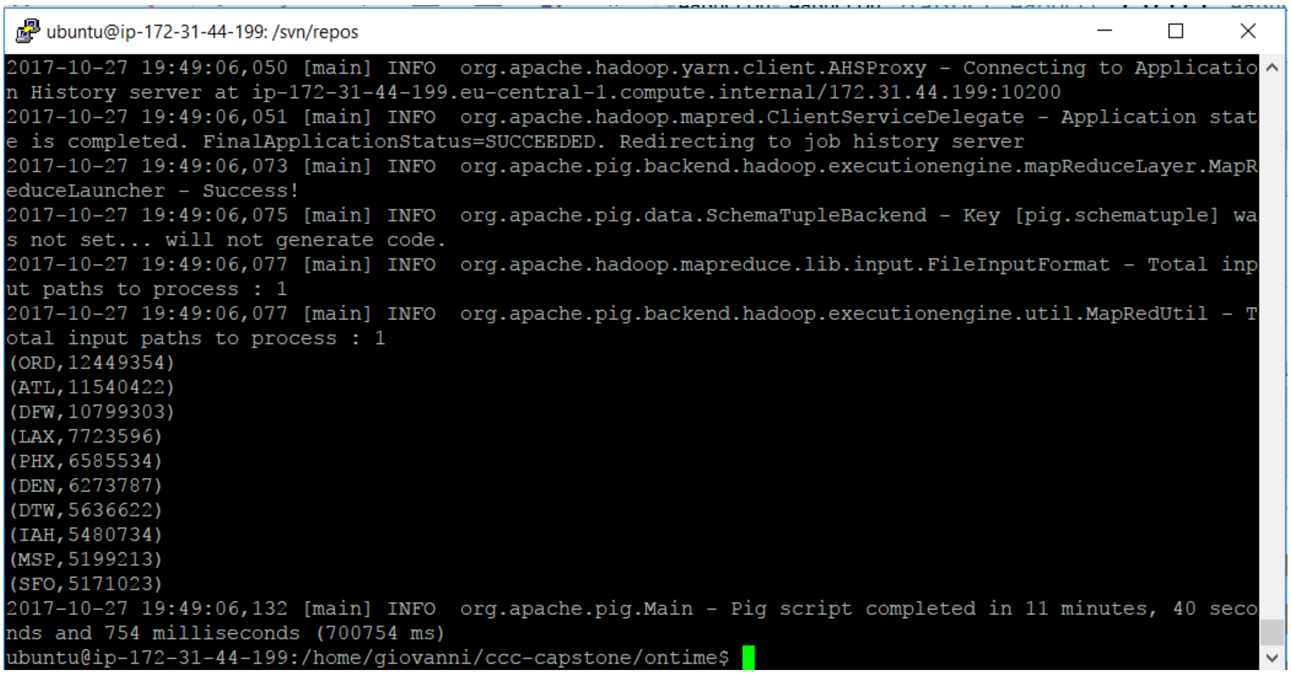
XII. OPTIMIZATIONS

The MapReduce algorithm has two important tasks: Map and Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). On the other hand, Reduce takes the output from a map as an input and combines the data tuples into smaller set of tuples. In MapReduce, the data is distributed over the cluster and processed. The difference in Spark is that it performs in-memory processing of data. This in-memory processing is a faster process as there is no time spent in moving the data/processes in and out of the disk, whereas MapReduce requires a lot of time to perform these input/output operations, thereby increasing latency. So Spark is used when a large set of files needs to be processed.

In HDFS, the zip files are converted into plain-text CSV files because zip files cannot be split, while CSV files are split into blocks and multiple concurrent map tasks can read the same file that is distributed among different nodes.

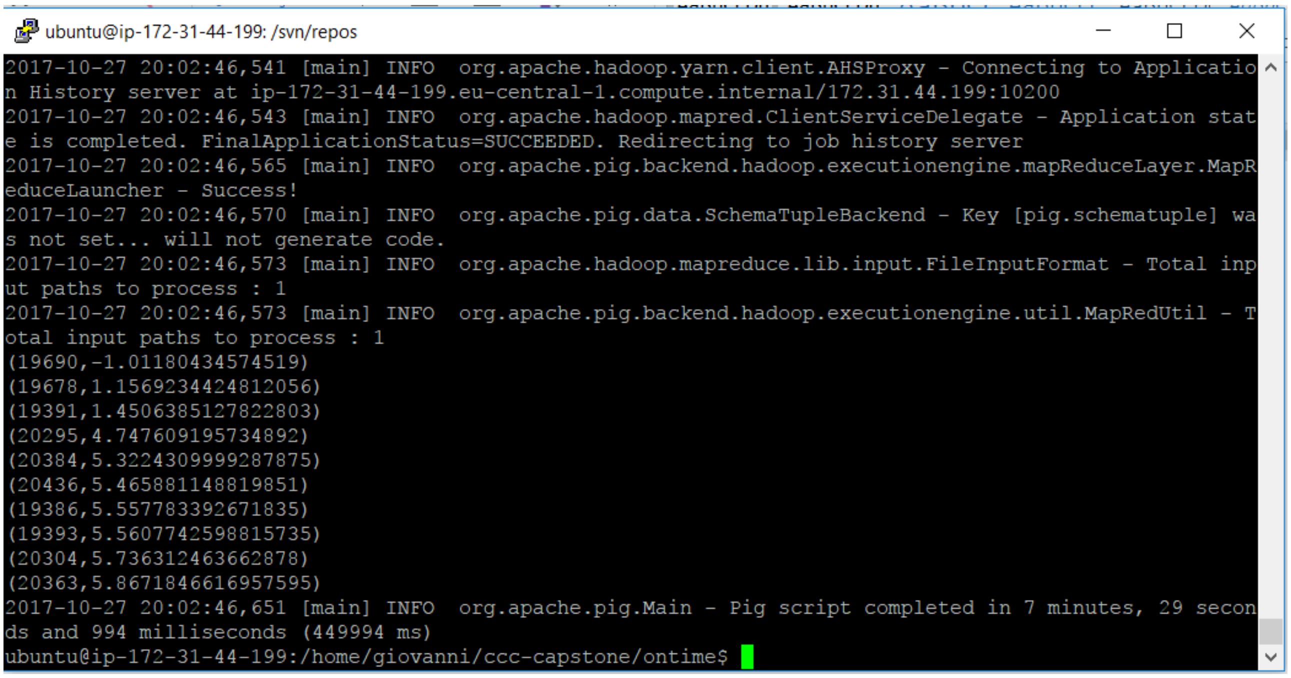
RESULTS

RESULTS 1.1


ubuntu@ip-172-31-44-199: /svn/repos

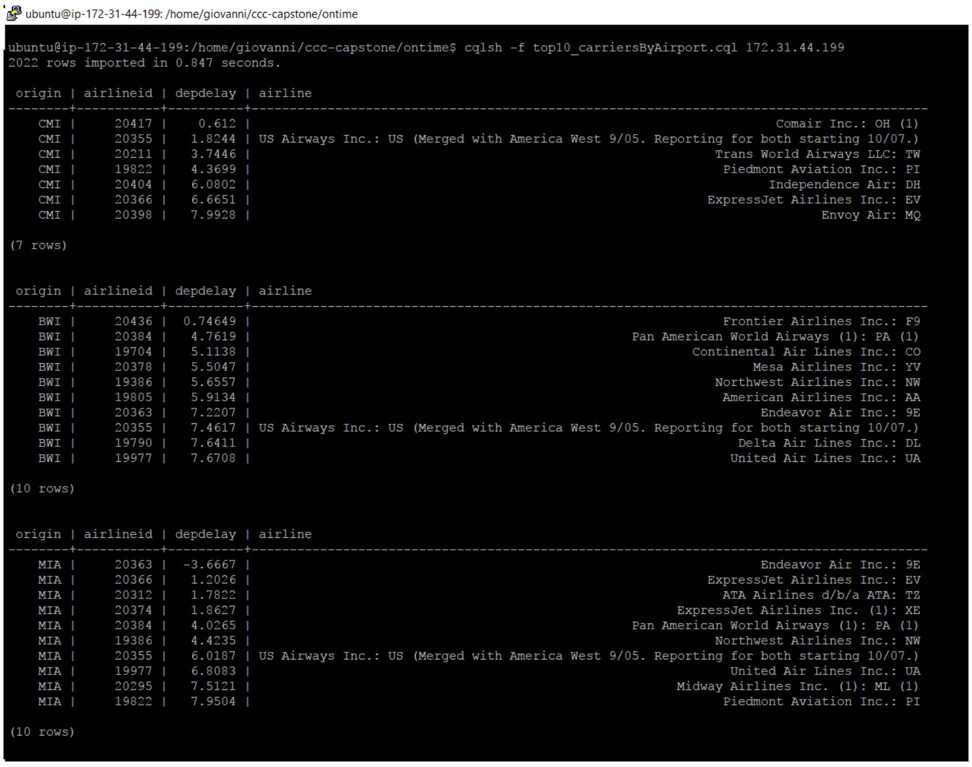
```
2017-10-27 19:49:06,050 [main] INFO  org.apache.hadoop.yarn.client.AHSProxy - Connecting to Applicatio
n History server at ip-172-31-44-199.eu-central-1.compute.internal/172.31.44.199:10200
2017-10-27 19:49:06,051 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application stat
e is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2017-10-27 19:49:06,073 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapR
educeLauncher - Success!
2017-10-27 19:49:06,075 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] wa
s not set... will not generate code.
2017-10-27 19:49:06,077 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total inp
ut paths to process : 1
2017-10-27 19:49:06,077 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - T
otal input paths to process : 1
(ORD,12449354)
(ATL,11540422)
(DFW,10799303)
(LAX,7723596)
(PHX,6585534)
(DEN,6273787)
(DTW,5636622)
(IAH,5480734)
(MSP,5199213)
(SFO,5171023)
2017-10-27 19:49:06,132 [main] INFO  org.apache.pig.Main - Pig script completed in 11 minutes, 40 seco
nds and 754 milliseconds (700754 ms)
ubuntu@ip-172-31-44-199:/home/giovanni/ccc-capstone/ontime$
```

RESULTS 1.2

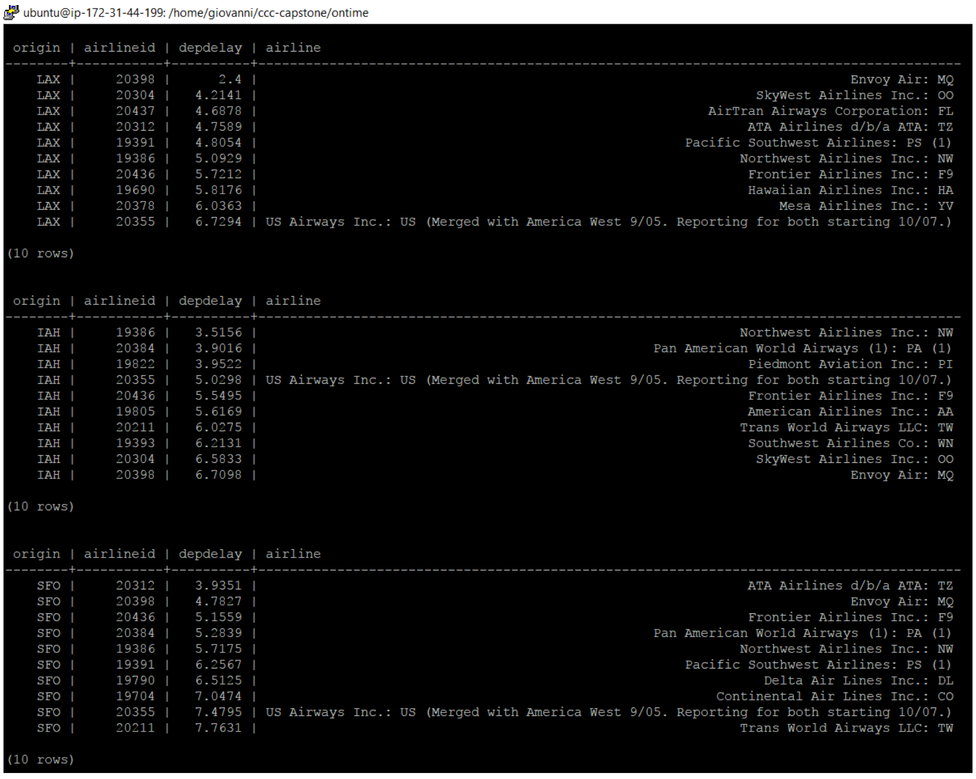

ubuntu@ip-172-31-44-199: /svn/repos

```
2017-10-27 20:02:46,541 [main] INFO  org.apache.hadoop.yarn.client.AHSProxy - Connecting to Applicatio
n History server at ip-172-31-44-199.eu-central-1.compute.internal/172.31.44.199:10200
2017-10-27 20:02:46,543 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application stat
e is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2017-10-27 20:02:46,565 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapR
educeLauncher - Success!
2017-10-27 20:02:46,570 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] wa
s not set... will not generate code.
2017-10-27 20:02:46,573 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total inp
ut paths to process : 1
2017-10-27 20:02:46,573 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - T
otal input paths to process : 1
(19690,-1.01180434574519)
(19678,1.1569234424812056)
(19391,1.4506385127822803)
(20295,4.747609195734892)
(20384,5.3224309999287875)
(20436,5.465881148819851)
(19386,5.557783392671835)
(19393,5.5607742598815735)
(20304,5.736312463662878)
(20363,5.8671846616957595)
2017-10-27 20:02:46,651 [main] INFO  org.apache.pig.Main - Pig script completed in 7 minutes, 29 secon
ds and 994 milliseconds (449994 ms)
ubuntu@ip-172-31-44-199:/home/giovanni/ccc-capstone/ontime$
```

RESULTS 2.1

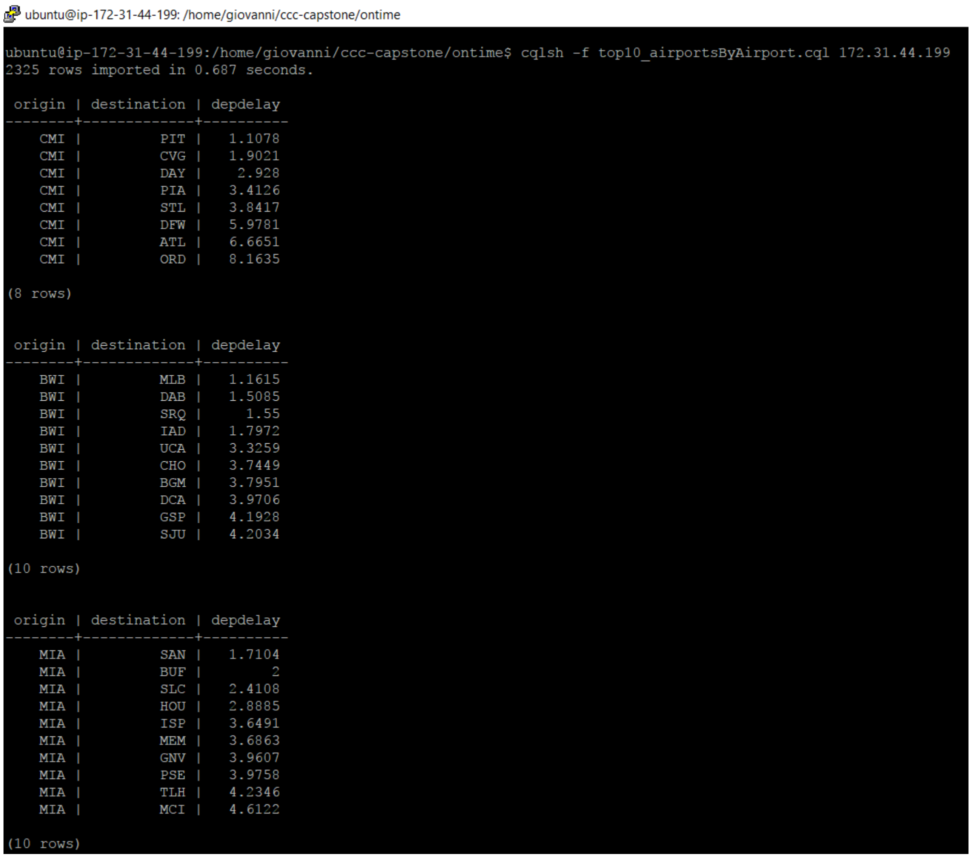

ubuntu@ip-172-31-44-199: /home/giovanni/ccc-capstone/ontime

```
ubuntu@ip-172-31-44-199:/home/giovanni/ccc-capstone/ontime$ cqlsh -f top10_carriersByAirport.cql 172.31.44.199
2022 rows imported in 0.847 seconds.

 origin | airlineid | depdelay | airline
--------+-----------+----------+-----------------------------------------------------------------------------------------
    CMI |     20417 |    0.612 |                                                                     Comair Inc.: OH (1)
    CMI |     20355 |   1.8244 | US Airways Inc.: US (Merged with America West 9/05. Reporting for both starting 10/07.)
    CMI |     20211 |   3.7446 |                                                             Trans World Airways LLC: TW
    CMI |     19822 |   4.3699 |                                                              Piedmont Aviation Inc.: PI
    CMI |     20404 |   6.0802 |                                                                    Independence Air: DH
    CMI |     20366 |   6.6651 |                                                            ExpressJet Airlines Inc.: EV
    CMI |     20398 |   7.9928 |                                                                           Envoy Air: MQ

(7 rows)

 origin | airlineid | depdelay | airline
--------+-----------+----------+-----------------------------------------------------------------------------------------
    BWI |     20436 |  0.74649 |                                                              Frontier Airlines Inc.: F9
    BWI |     20384 |   4.7619 |                                                  Pan American World Airways (1): PA (1)
    BWI |     19704 |   5.1138 |                                                          Continental Air Lines Inc.: CO
    BWI |     20378 |   5.5047 |                                                                  Mesa Airlines Inc.: YV
    BWI |     19386 |   5.6557 |                                                             Northwest Airlines Inc.: NW
    BWI |     19805 |   5.9134 |                                                              American Airlines Inc.: AA
    BWI |     20363 |   7.2207 |                                                                   Endeavor Air Inc.: 9E
    BWI |     20355 |   7.4617 | US Airways Inc.: US (Merged with America West 9/05. Reporting for both starting 10/07.)
    BWI |     19790 |   7.6411 |                                                                Delta Air Lines Inc.: DL
    BWI |     19977 |   7.6708 |                                                               United Air Lines Inc.: UA

(10 rows)

 origin | airlineid | depdelay | airline
--------+-----------+----------+-----------------------------------------------------------------------------------------
    MIA |     20363 |  -3.6667 |                                                                   Endeavor Air Inc.: 9E
    MIA |     20366 |   1.2026 |                                                            ExpressJet Airlines Inc.: EV
    MIA |     20312 |   1.7822 |                                                              ATA Airlines d/b/a ATA: TZ
    MIA |     20374 |   1.8627 |                                                        ExpressJet Airlines Inc. (1): XE
    MIA |     20384 |   4.0265 |                                                  Pan American World Airways (1): PA (1)
    MIA |     19386 |   4.4235 |                                                             Northwest Airlines Inc.: NW
    MIA |     20355 |   6.0187 | US Airways Inc.: US (Merged with America West 9/05. Reporting for both starting 10/07.)
    MIA |     19977 |   6.8083 |                                                               United Air Lines Inc.: UA
    MIA |     20295 |   7.5121 |                                                        Midway Airlines Inc. (1): ML (1)
    MIA |     19822 |   7.9504 |                                                              Piedmont Aviation Inc.: PI

(10 rows)
```

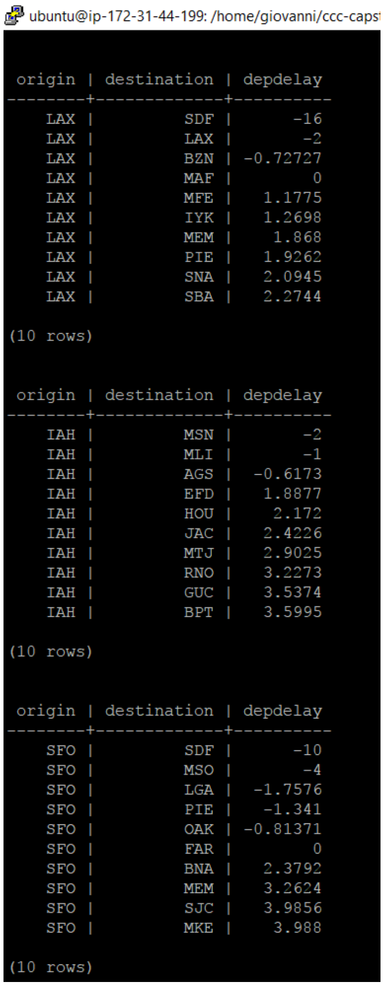

ubuntu@ip-172-31-44-199: /home/giovanni/ccc-capstone/ontime

```
 origin | airlineid | depdelay | airline
--------+-----------+----------+-----------------------------------------------------------------------------------------
    LAX |     20398 |      2.4 |                                                                           Envoy Air: MQ
    LAX |     20304 |   4.2141 |                                                               SkyWest Airlines Inc.: OO
    LAX |     20437 |   4.6878 |                                                         AirTran Airways Corporation: FL
    LAX |     20312 |   4.7589 |                                                              ATA Airlines d/b/a ATA: TZ
    LAX |     19391 |   4.8054 |                                                      Pacific Southwest Airlines: PS (1)
    LAX |     19386 |   5.0929 |                                                             Northwest Airlines Inc.: NW
    LAX |     20436 |   5.7212 |                                                              Frontier Airlines Inc.: F9
    LAX |     19690 |   5.8176 |                                                              Hawaiian Airlines Inc.: HA
    LAX |     20378 |   6.0363 |                                                                  Mesa Airlines Inc.: YV
    LAX |     20355 |   6.7294 | US Airways Inc.: US (Merged with America West 9/05. Reporting for both starting 10/07.)

(10 rows)

 origin | airlineid | depdelay | airline
--------+-----------+----------+-----------------------------------------------------------------------------------------
    IAH |     19386 |   3.5156 |                                                             Northwest Airlines Inc.: NW
    IAH |     20384 |   3.9016 |                                                  Pan American World Airways (1): PA (1)
    IAH |     19822 |   3.9522 |                                                              Piedmont Aviation Inc.: PI
    IAH |     20355 |   5.0298 | US Airways Inc.: US (Merged with America West 9/05. Reporting for both starting 10/07.)
    IAH |     20436 |   5.5495 |                                                              Frontier Airlines Inc.: F9
    IAH |     19805 |   5.6169 |                                                              American Airlines Inc.: AA
    IAH |     20211 |   6.0275 |                                                             Trans World Airways LLC: TW
    IAH |     19393 |   6.2131 |                                                              Southwest Airlines Co.: WN
    IAH |     20304 |   6.5833 |                                                               SkyWest Airlines Inc.: OO
    IAH |     20398 |   6.7098 |                                                                           Envoy Air: MQ

(10 rows)

 origin | airlineid | depdelay | airline
--------+-----------+----------+-----------------------------------------------------------------------------------------
    SFO |     20312 |   3.9351 |                                                              ATA Airlines d/b/a ATA: TZ
    SFO |     20398 |   4.7827 |                                                                           Envoy Air: MQ
    SFO |     20436 |   5.1559 |                                                              Frontier Airlines Inc.: F9
    SFO |     20384 |   5.2839 |                                                  Pan American World Airways (1): PA (1)
    SFO |     19386 |   5.7175 |                                                             Northwest Airlines Inc.: NW
    SFO |     19391 |   6.2567 |                                                      Pacific Southwest Airlines: PS (1)
    SFO |     19790 |   6.5125 |                                                                Delta Air Lines Inc.: DL
    SFO |     19704 |   7.0474 |                                                          Continental Air Lines Inc.: CO
    SFO |     20355 |   7.4795 | US Airways Inc.: US (Merged with America West 9/05. Reporting for both starting 10/07.)
    SFO |     20211 |   7.7631 |                                                             Trans World Airways LLC: TW

(10 rows)
```

RESULTS 2.2

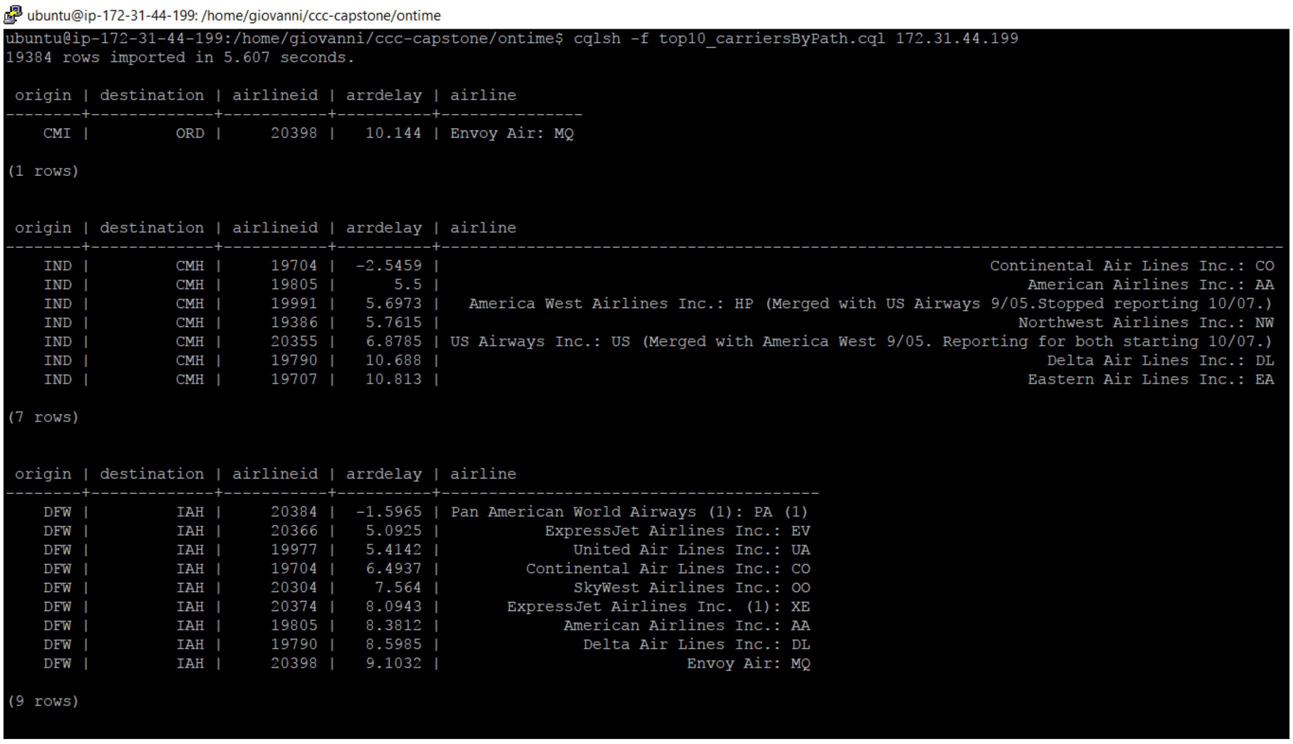

ubuntu@ip-172-31-44-199: /home/giovanni/ccc-capstone/ontime

```
ubuntu@ip-172-31-44-199:/home/giovanni/ccc-capstone/ontime$ cqlsh -f top10_airportsByAirport.cql 172.31.44.199
2325 rows imported in 0.687 seconds.

 origin | destination | depdelay
--------+-------------+----------
    CMI |         PIT |   1.1078
    CMI |         CVG |   1.9021
    CMI |         DAY |    2.928
    CMI |         PIA |   3.4126
    CMI |         STL |   3.8417
    CMI |         DFW |   5.9781
    CMI |         ATL |   6.6651
    CMI |         ORD |   8.1635

(8 rows)

 origin | destination | depdelay
--------+-------------+----------
    BWI |         MLB |   1.1615
    BWI |         DAB |   1.5085
    BWI |         SRQ |     1.55
    BWI |         IAD |   1.7972
    BWI |         UCA |   3.3259
    BWI |         CHO |   3.7449
    BWI |         BGM |   3.7951
    BWI |         DCA |   3.9706
    BWI |         GSP |   4.1928
    BWI |         SJU |   4.2034

(10 rows)

 origin | destination | depdelay
--------+-------------+----------
    MIA |         SAN |   1.7104
    MIA |         BUF |        2
    MIA |         SLC |   2.4108
    MIA |         HOU |   2.8885
    MIA |         ISP |   3.6491
    MIA |         MEM |   3.6863
    MIA |         GNV |   3.9607
    MIA |         PSE |   3.9758
    MIA |         TLH |   4.2346
    MIA |         MCI |   4.6122

(10 rows)
```

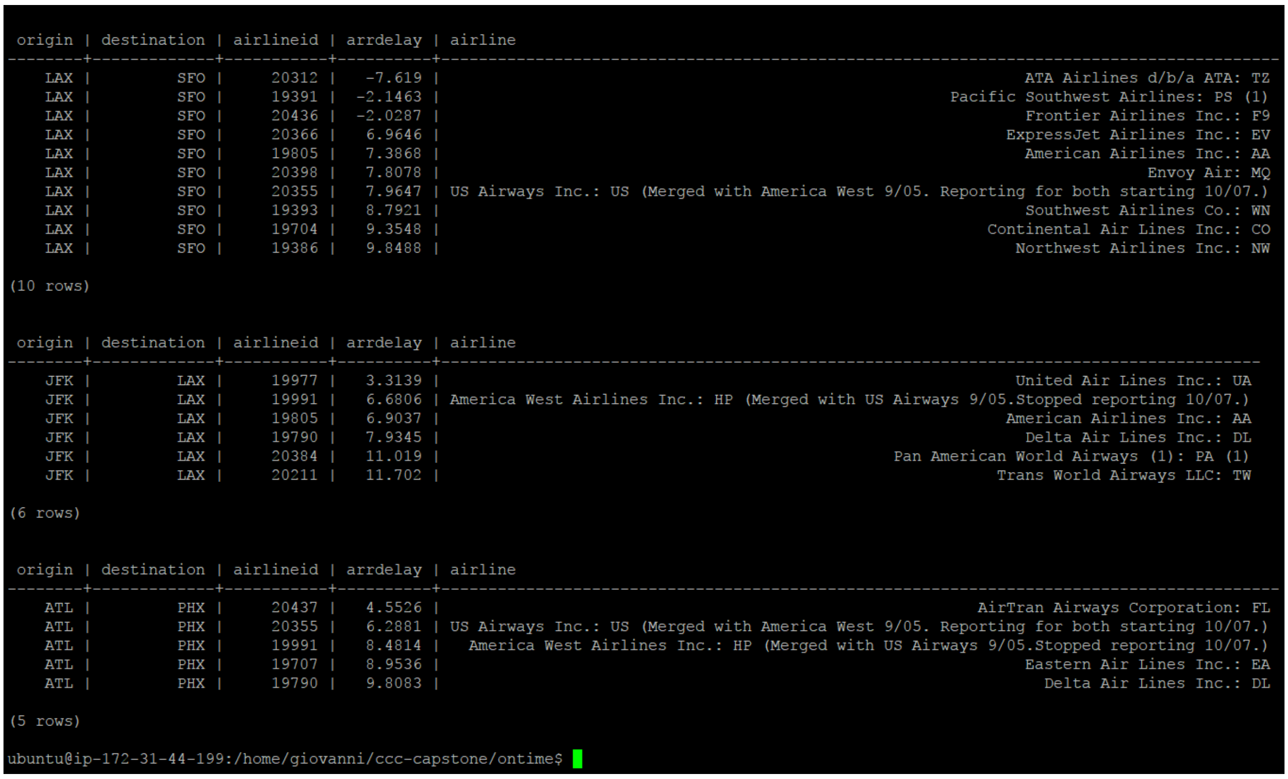

ubuntu@ip-172-31-44-199: /home/giovanni/ccc-caps

```
 origin | destination | depdelay
--------+-------------+----------
    LAX |         SDF |      -16
    LAX |         LAX |       -2
    LAX |         BZN | -0.72727
    LAX |         MAF |        0
    LAX |         MFE |   1.1775
    LAX |         IYK |   1.2698
    LAX |         MEM |    1.868
    LAX |         PIE |   1.9262
    LAX |         SNA |   2.0945
    LAX |         SBA |   2.2744

(10 rows)

 origin | destination | depdelay
--------+-------------+----------
    IAH |         MSN |       -2
    IAH |         MLI |       -1
    IAH |         AGS |  -0.6173
    IAH |         EFD |   1.8877
    IAH |         HOU |    2.172
    IAH |         JAC |   2.4226
    IAH |         MTJ |   2.9025
    IAH |         RNO |   3.2273
    IAH |         GUC |   3.5374
    IAH |         BPT |   3.5995

(10 rows)

 origin | destination | depdelay
--------+-------------+----------
    SFO |         SDF |      -10
    SFO |         MSO |       -4
    SFO |         LGA |  -1.7576
    SFO |         PIE |   -1.341
    SFO |         OAK | -0.81371
    SFO |         FAR |        0
    SFO |         BNA |   2.3792
    SFO |         MEM |   3.2624
    SFO |         SJC |   3.9856
    SFO |         MKE |    3.988

(10 rows)
```

RESULTS 2.3



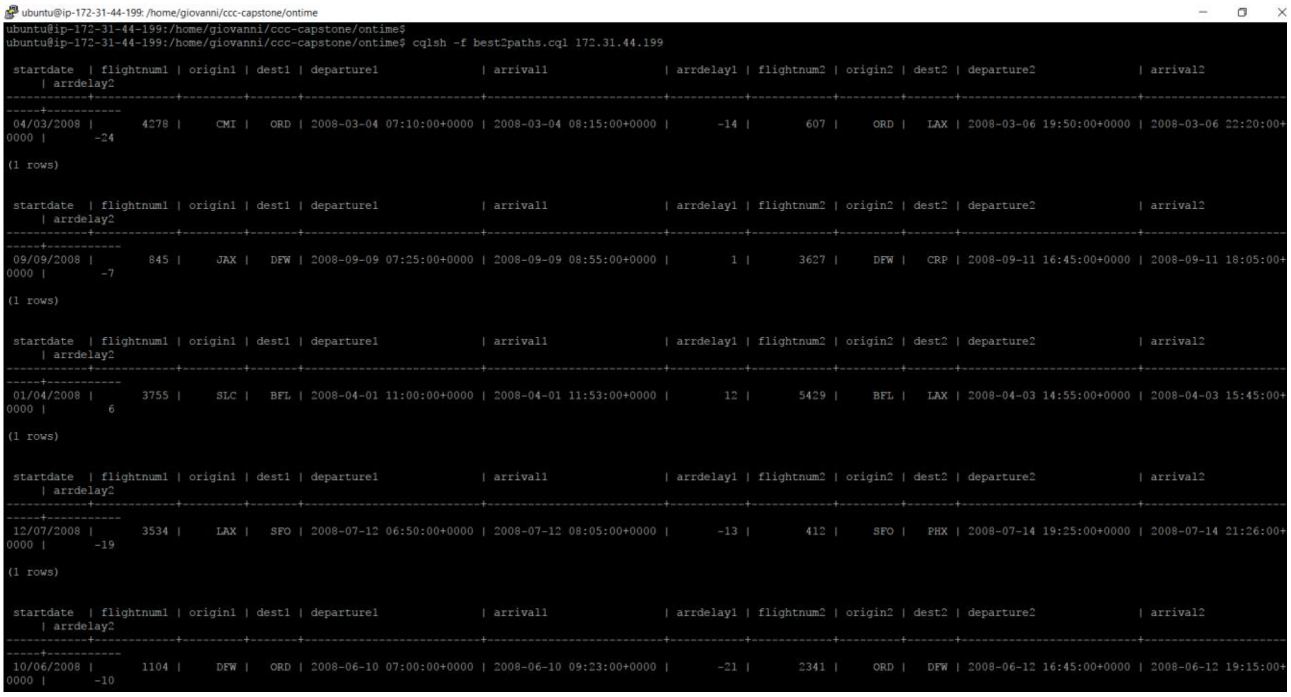
```
ubuntu@ip-172-31-44-199: /home/giovanni/ccc-capstone/ontime
ubuntu@ip-172-31-44-199:/home/giovanni/ccc-capstone/ontime$ cqlsh -f top10_carriersByPath.cql 172.31.44.199
19384 rows imported in 5.607 seconds.

 origin | destination | airlineid | arrdelay | airline
--------+-------------+-----------+----------+---------------
    CMI |         ORD |     20398 |   10.144 | Envoy Air: MQ

(1 rows)

 origin | destination | airlineid | arrdelay | airline
--------+-------------+-----------+----------+----------------------------------------------------------------------------------------
    IND |         CMH |     19704 |  -2.5459 |                                                         Continental Air Lines Inc.: CO
    IND |         CMH |     19805 |      5.5 |                                                             American Airlines Inc.: AA
    IND |         CMH |     19991 |   5.6973 |  America West Airlines Inc.: HP (Merged with US Airways 9/05.Stopped reporting 10/07.)
    IND |         CMH |     19386 |   5.7615 |                                                            Northwest Airlines Inc.: NW
    IND |         CMH |     20355 |   6.8785 | US Airways Inc.: US (Merged with America West 9/05. Reporting for both starting 10/07.)
    IND |         CMH |     19790 |   10.688 |                                                               Delta Air Lines Inc.: DL
    IND |         CMH |     19707 |   10.813 |                                                             Eastern Air Lines Inc.: EA

(7 rows)

 origin | destination | airlineid | arrdelay | airline
--------+-------------+-----------+----------+----------------------------------------
    DFW |         IAH |     20384 |  -1.5965 | Pan American World Airways (1): PA (1)
    DFW |         IAH |     20366 |   5.0925 |           ExpressJet Airlines Inc.: EV
    DFW |         IAH |     19977 |   5.4142 |              United Air Lines Inc.: UA
    DFW |         IAH |     19704 |   6.4937 |         Continental Air Lines Inc.: CO
    DFW |         IAH |     20304 |    7.564 |              SkyWest Airlines Inc.: OO
    DFW |         IAH |     20374 |   8.0943 |       ExpressJet Airlines Inc. (1): XE
    DFW |         IAH |     19805 |   8.3812 |             American Airlines Inc.: AA
    DFW |         IAH |     19790 |   8.5985 |               Delta Air Lines Inc.: DL
    DFW |         IAH |     20398 |   9.1032 |                          Envoy Air: MQ

(9 rows)
```



```
 origin | destination | airlineid | arrdelay | airline
--------+-------------+-----------+----------+----------------------------------------------------------------------------------------
    LAX |         SFO |     20312 |   -7.619 |                                                             ATA Airlines d/b/a ATA: TZ
    LAX |         SFO |     19391 |  -2.1463 |                                                     Pacific Southwest Airlines: PS (1)
    LAX |         SFO |     20436 |  -2.0287 |                                                             Frontier Airlines Inc.: F9
    LAX |         SFO |     20366 |   6.9646 |                                                           ExpressJet Airlines Inc.: EV
    LAX |         SFO |     19805 |   7.3868 |                                                             American Airlines Inc.: AA
    LAX |         SFO |     20398 |   7.8078 |                                                                          Envoy Air: MQ
    LAX |         SFO |     20355 |   7.9647 | US Airways Inc.: US (Merged with America West 9/05. Reporting for both starting 10/07.)
    LAX |         SFO |     19393 |   8.7921 |                                                             Southwest Airlines Co.: WN
    LAX |         SFO |     19704 |   9.3548 |                                                         Continental Air Lines Inc.: CO
    LAX |         SFO |     19386 |   9.8488 |                                                            Northwest Airlines Inc.: NW

(10 rows)

 origin | destination | airlineid | arrdelay | airline
--------+-------------+-----------+----------+---------------------------------------------------------------------------------------
    JFK |         LAX |     19977 |   3.3139 |                                                             United Air Lines Inc.: UA
    JFK |         LAX |     19991 |   6.6806 | America West Airlines Inc.: HP (Merged with US Airways 9/05.Stopped reporting 10/07.)
    JFK |         LAX |     19805 |   6.9037 |                                                            American Airlines Inc.: AA
    JFK |         LAX |     19790 |   7.9345 |                                                              Delta Air Lines Inc.: DL
    JFK |         LAX |     20384 |   11.019 |                                               Pan American World Airways (1): PA (1)
    JFK |         LAX |     20211 |   11.702 |                                                           Trans World Airways LLC: TW

(6 rows)

 origin | destination | airlineid | arrdelay | airline
--------+-------------+-----------+----------+----------------------------------------------------------------------------------------
    ATL |         PHX |     20437 |   4.5526 |                                                        AirTran Airways Corporation: FL
    ATL |         PHX |     20355 |   6.2881 | US Airways Inc.: US (Merged with America West 9/05. Reporting for both starting 10/07.)
    ATL |         PHX |     19991 |   8.4814 |  America West Airlines Inc.: HP (Merged with US Airways 9/05.Stopped reporting 10/07.)
    ATL |         PHX |     19707 |   8.9536 |                                                             Eastern Air Lines Inc.: EA
    ATL |         PHX |     19790 |   9.8083 |                                                               Delta Air Lines Inc.: DL

(5 rows)

ubuntu@ip-172-31-44-199:/home/giovanni/ccc-capstone/ontime$
```

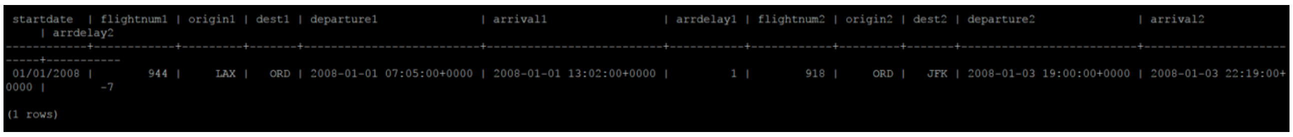
RESULTS 3.1

RESULTS 3.2



ubuntu@ip-172-31-44-199: /home/giovanni/ccc-capstone/ontime

```
ubuntu@ip-172-31-44-199:/home/giovanni/ccc-capstone/ontime$
ubuntu@ip-172-31-44-199:/home/giovanni/ccc-capstone/ontime$ cqlsh -f best2paths.cql 172.31.44.199

 startdate  | flightnum1 | origin1 | dest1 | departure1               | arrival1                 | arrdelay1 | flightnum2 | origin2 | dest2 | departure2               | arrival2
     | arrdelay2
------------+------------+---------+-------+--------------------------+--------------------------+-----------+------------+---------+-------+--------------------------+-------------------------
-----+-----------
 04/03/2008 |       4278 |     CMI |   ORD | 2008-03-04 07:10:00+0000 | 2008-03-04 08:15:00+0000 |       -14 |        607 |     ORD |   LAX | 2008-03-06 19:50:00+0000 | 2008-03-06 22:20:00+
0000 |       -24

(1 rows)

 startdate  | flightnum1 | origin1 | dest1 | departure1               | arrival1                 | arrdelay1 | flightnum2 | origin2 | dest2 | departure2               | arrival2
     | arrdelay2
------------+------------+---------+-------+--------------------------+--------------------------+-----------+------------+---------+-------+--------------------------+-------------------------
-----+-----------
 09/09/2008 |        845 |     JAX |   DFW | 2008-09-09 07:25:00+0000 | 2008-09-09 08:55:00+0000 |         1 |       3627 |     DFW |   CRP | 2008-09-11 16:45:00+0000 | 2008-09-11 18:05:00+
0000 |        -7

(1 rows)

 startdate  | flightnum1 | origin1 | dest1 | departure1               | arrival1                 | arrdelay1 | flightnum2 | origin2 | dest2 | departure2               | arrival2
     | arrdelay2
------------+------------+---------+-------+--------------------------+--------------------------+-----------+------------+---------+-------+--------------------------+-------------------------
-----+-----------
 01/04/2008 |       3755 |     SLC |   BFL | 2008-04-01 11:00:00+0000 | 2008-04-01 11:53:00+0000 |        12 |       5429 |     BFL |   LAX | 2008-04-03 14:55:00+0000 | 2008-04-03 15:45:00+
0000 |         6

(1 rows)

 startdate  | flightnum1 | origin1 | dest1 | departure1               | arrival1                 | arrdelay1 | flightnum2 | origin2 | dest2 | departure2               | arrival2
     | arrdelay2
------------+------------+---------+-------+--------------------------+--------------------------+-----------+------------+---------+-------+--------------------------+-------------------------
-----+-----------
 12/07/2008 |       3534 |     LAX |   SFO | 2008-07-12 06:50:00+0000 | 2008-07-12 08:05:00+0000 |       -13 |        412 |     SFO |   PHX | 2008-07-14 19:25:00+0000 | 2008-07-14 21:26:00+
0000 |       -19

(1 rows)

 startdate  | flightnum1 | origin1 | dest1 | departure1               | arrival1                 | arrdelay1 | flightnum2 | origin2 | dest2 | departure2               | arrival2
     | arrdelay2
------------+------------+---------+-------+--------------------------+--------------------------+-----------+------------+---------+-------+--------------------------+-------------------------
-----+-----------
 10/06/2008 |       1104 |     DFW |   ORD | 2008-06-10 07:00:00+0000 | 2008-06-10 09:23:00+0000 |       -21 |       2341 |     ORD |   DFW | 2008-06-12 16:45:00+0000 | 2008-06-12 19:15:00+
0000 |       -10
```



```
 startdate  | flightnum1 | origin1 | dest1 | departure1               | arrival1                 | arrdelay1 | flightnum2 | origin2 | dest2 | departure2               | arrival2
     | arrdelay2
------------+------------+---------+-------+--------------------------+--------------------------+-----------+------------+---------+-------+--------------------------+-------------------------
-----+-----------
 01/01/2008 |        944 |     LAX |   ORD | 2008-01-01 07:05:00+0000 | 2008-01-01 13:02:00+0000 |         1 |        918 |     ORD |   JFK | 2008-01-03 19:00:00+0000 | 2008-01-03 22:19:00+
0000 |        -7

(1 rows)
```