# TASK1

Video Link: https://youtu.be/G_vcZFWWsG8

## SYSTEM INTEGRATION

I decided to build Hadoop cluster containing one Master (m4.large) node and 3 slave nodes (m4.2xlarge). Cassandra cluster contained only two nodes (m4.large). Both clusters were integrated on rather low level, I did perform Cassandra inserts from Hadoop reduce jobs.

## CLEANUP

Airline-ontime dataset seems to be sufficient to answer all questions. Each row from each file, except header, was transformed into following format and stored in HDFS.

ORIGIN, DESTINATION, DEPDELAYMIN, ARRDELMIN, DAYOFWEEK, FLIGHTDATE, CRSDEPARTURE, CARRIER, AIRLINE, FLIGH

To do so I unzipped all files using simple bash script and loaded to HDFS, then used Hadoop job to do transformation.

## Q 1.2 RANK THE TOP 10 AIRLINES BY ON-TIME ARRIVAL

I considered on time arrival when arrival delay minutes was less than 15.

To get only top 10 airlines I used approach called secondary sorting it is a technique that allows the MapReduce programmer to control the order that the values show up within a reduce function call.

We can achieve this using a composite key that contains both the information needed to sort by key and the information needed by value, and then decoupling the grouping of the intermediate data from the sorting of the intermediate data.

### Q 1.2 RESULTS

```
AIRLINE (ON-TIME #)

19393 (11653055)
19790 (11617449)
19805 (10429826)
20355 (10069525)
19977 (9080416)
19386 (7321432)
19704 (5760074)
20211 (2751559)
19991 (2574988)
20398 (2469438)
```

## Q 1.3 RANK DAYS OF WEEK BY ON-TIME ARRIVAL

Same assumptions were made here as in Q 1.2 and also same algorithms were used.

### Q 1.3 RESULTS

```
DAY OF WEEK (ON-TIME #)

2(12908937)
1(12862198)
3(12678529)
4(12248323)
7(12178435)
5(12167955)
6(11666461)
```

## Q 2.3 FOR X-Y PAIR RANK TOP 10 CARRIERS

In addition to secondary sorting I used grouping technique which allowed me to receive in reducer different grouped keys.

### Q 2.3 RESULTS

```
ATL -> PHX  DL(23679)
ATL -> PHX HP(10298)
ATL -> PHX US(3073)
ATL -> PHX EA(1400)
ATL -> PHX FL(1387)
```

```
DFW -> IAH  AA(52067)
DFW -> IAH CO(51575)
DFW -> IAH DL(19991)
DFW -> IAH RU(2921)
DFW -> IAH XE(2593)
DFW -> IAH EV(1344)
DFW -> IAH MQ(1218)
DFW -> IAH OO(356)
DFW -> IAH UA(139)
DFW -> IAH PA(51)

IND -> CMH  CO(3992)
IND -> CMH US(2529)
IND -> CMH HP(1058)
IND -> CMH AL(557)
IND -> CMH NW(538)
IND -> CMH DL(165)
IND -> CMH EA(150)
IND -> CMH AA(4)
```

## Q 2.4 RESULTS

```
ATL -> PHX  US(12)
ATL -> PHX HP(13)
ATL -> PHX FL(12)
ATL -> PHX EA(14)
ATL -> PHX DL(13)

DFW -> IAH XE(14)
DFW -> IAH UA(8)
DFW -> IAH RU(11)
DFW -> IAH PA(9)
DFW -> IAH OO(8)
DFW -> IAH MQ(12)
DFW -> IAH EV(10)
DFW -> IAH DL(10)
DFW -> IAH CO(9)
DFW -> IAH AA(12)

IND -> CMH US(7)
IND -> CMH NW(7)
IND -> CMH HP(7)
IND -> CMH EA(11)
IND -> CMH DL(12)
IND -> CMH CO(4)
IND -> CMH AL(8)
IND -> CMH AA(8)
```

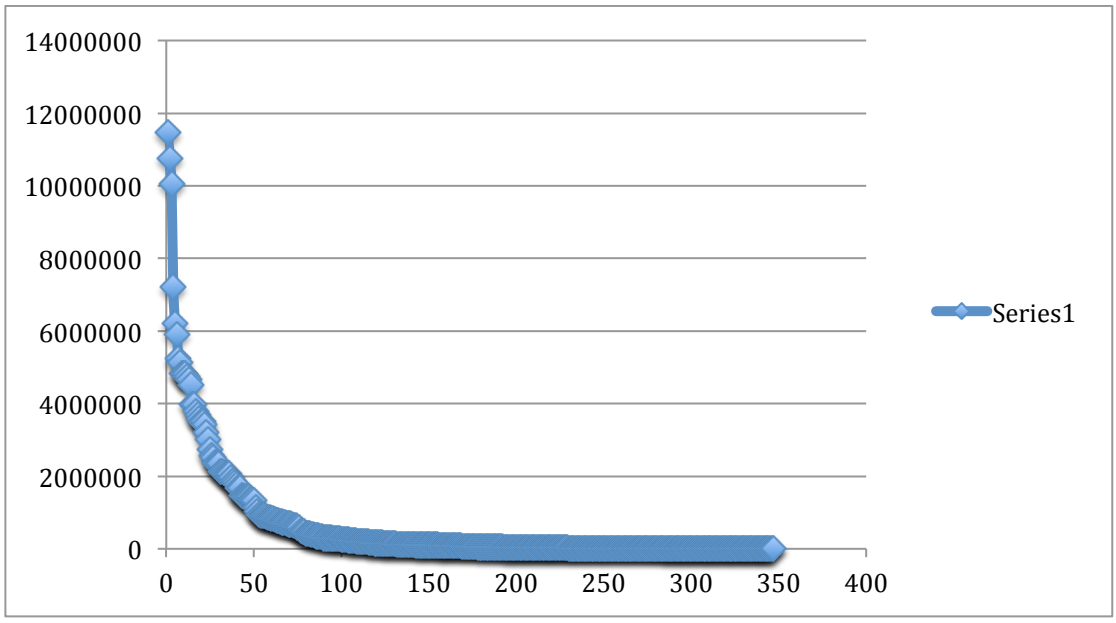
## Q 2.1 RESULTS

```
CMI MQ(11997)
CMI   US(2858)
CMI   OH(1405)
CMI   PI(1353)
CMI   TW(1306)
CMI   DH(406)
CMI   EV(179)

BWI   WN(401778)
BWI   US(334434)
BWI   AA(69732)
BWI   DL(61405)
BWI   UA(60211)
BWI   FL(54892)
BWI   CO(50593)
BWI   NW(50591)
BWI   PI(47599)
BWI   TW(21188)
```

## Q 3.1 RESULTS

Airport popularity doesn't look like ZIPF distribution, It's more like exponential distribution.




## Q 3.2 TOM TRAVEL

In addition to techniques used in previous questions I did use data joining technique. I joined flights from X – Y and Y – Z on Y mid airport as a key, then applied date filtering.

### Q 3.2 RESULTS

```
ORIGIN | MID | DESTINATION | DATE     | FLIGHT1 | FLIGHT2 | RANK
--------+-----+------------+-----------+---------+---------+------
  CMI | ORD |       LAX | 2008-03-04 |   4278 |    129 |   9
ORIGIN | MID | DESTINATION | DATE     | FLIGHT1 | FLIGHT2 | RANK
--------+-----+------------+-----------+---------+---------+------
  JAX | DFW |       CRP | 2008-09-09 |    845 |   3627 |   4
ORIGIN | MID | DESTINATION | DATE     | FLIGHT1 | FLIGHT2 | RANK
--------+-----+------------+-----------+---------+---------+------
  LAX | ORD |       JFK | 2008-01-01 |    944 |   5366 |   9
```

## SUMMARY

Overall information retrieved might be useful to on planning trip, or companies to improve flights performance.

Methodology I used to get results was very low level/ raw I did spend a lot of time writing MapReduce jobs but it makes me appreciate technologies like PIG and HIVE even more