Problem Statement 1

Find out the top 5 most visited destinations.

Step1: Load the dataset from /home/acadgild/assignment\_23.2/DelayedFlights.csv and put to RDD delayed\_flights. Code is as below:

val delayed\_flights = sc.textFile("/home/acadgild/assignment\_23.2/DelayedFlights.csv")

Step2: Get the header

Get the header

val header = delayed\_flights.first()

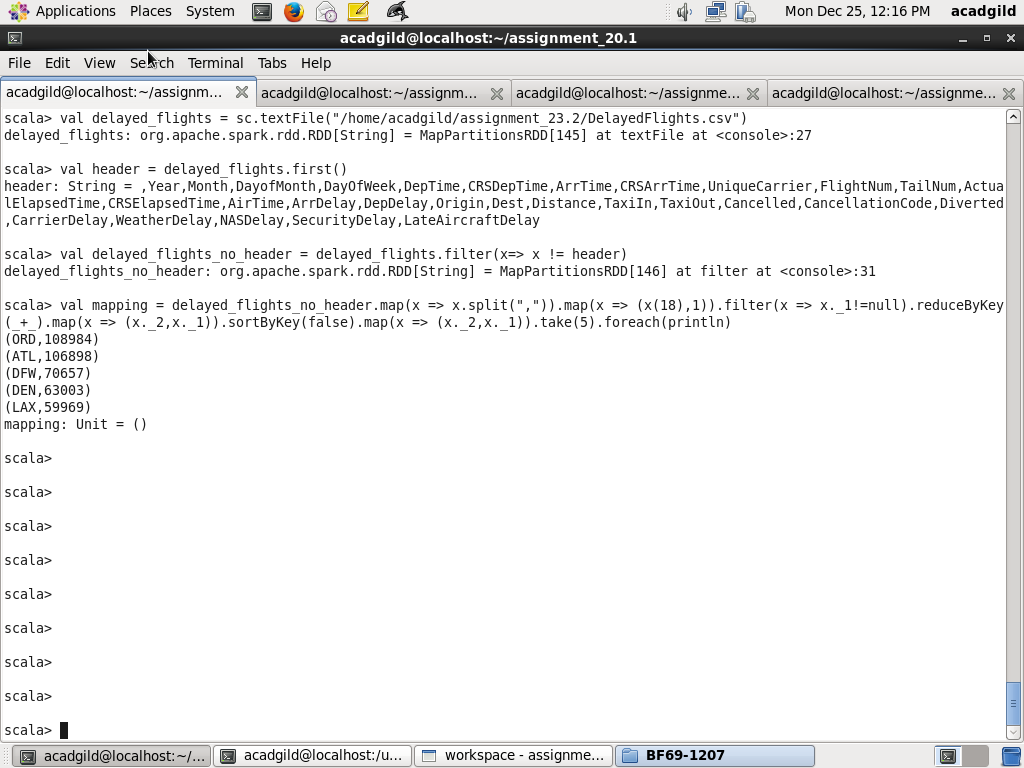
Step3: Remove header from delayed\_flights and put to delayed\_flights\_no\_header

val delayed\_flights\_no\_header = delayed\_flights.filter(x=> x != header)

Step4: Query to Find out the top 5 most visited destinations

* Split the fields of delayed\_flights\_no\_header based on separator comma.
* Map the Dest field (which is field 18) to 1
* Filter only Not Null records
* Use reduceByKey to get total number of time Dest appears group by Dest
* Sort the list by number of time Dest appears
* Take the first 5
* Print the tuples

val mapping = delayed\_flights\_no\_header.map(x => x.split(",")).map(x => (x(18),1)).filter(x => x.\_1!=null).reduceByKey(\_+\_).map(x => (x.\_2,x.\_1)).sortByKey(false).map(x => (x.\_2,x.\_1)).take(5).foreach(println)



Problem Statement 2

Which month has seen the most number of cancellations due to bad weather?

Step1: Load the dataset from /home/acadgild/assignment\_23.2/DelayedFlights.csv and put to RDD delayed\_flights. Code is as below

val delayed\_flights = sc.textFile("/home/acadgild/assignment\_23.2/DelayedFlights.csv")

Step2: Get the header

val header = delayed\_flights.first()

Step3: Remove the header

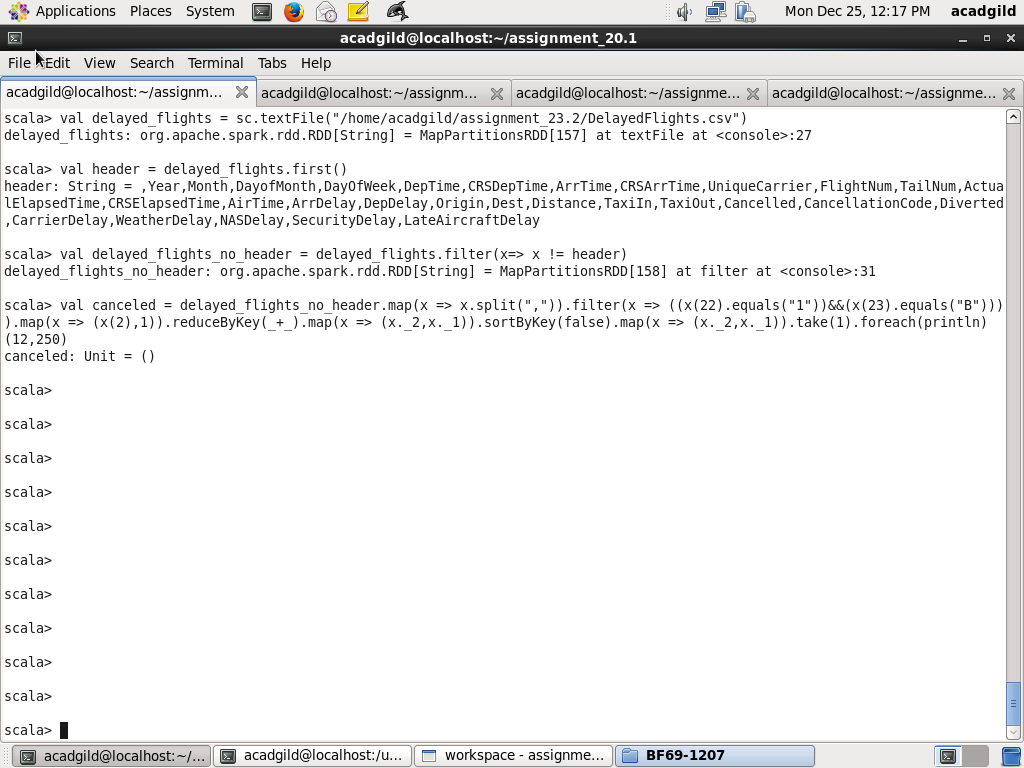
Remove the heading by filtering tuples which are not header and put to delayed\_flights\_no\_header

val delayed\_flights\_no\_header = delayed\_flights.filter(x=> x != header)

Step4: Query the delayed\_fligts to get Which month has seen the most number of cancellations due to bad weather.

* Split the fields of delayed\_flights\_no\_header based on separator comma.
* Map the Dest field (which is field 18) to 1
* Filter those records for which Cancelled field (field 22) is 1
* Filter those records for which Cancellation field (field 230 is B
* Use map to get Month , count as 1
* Use reduceByKey to get total number of time Month appears group by Month
* Sort the list by number of time Month appears
* Take the first 1
* Print the tuple

val canceled = delayed\_flights\_no\_header.map(x => x.split(",")).filter(x => ((x(22).equals("1"))&&(x(23).equals("B")))).map(x => (x(2),1)).reduceByKey(\_+\_).map(x => (x.\_2,x.\_1)).sortByKey(false).map(x => (x.\_2,x.\_1)).take(1).foreach(println)



Problem Statement 3

Top ten origins with the highest AVG departure delay

Step1: Load the dataset from /home/acadgild/assignment\_23.2/DelayedFlights.csv and put to RDD delayed\_flights. Code is as below

val delayed\_flights = sc.textFile("/home/acadgild/assignment\_23.2/DelayedFlights.csv")

Step2: Get the header

val header = delayed\_flights.first()

Step3: Remove the header

Remove the heading by filtering tuples which are not header and put to delayed\_flights\_no\_header

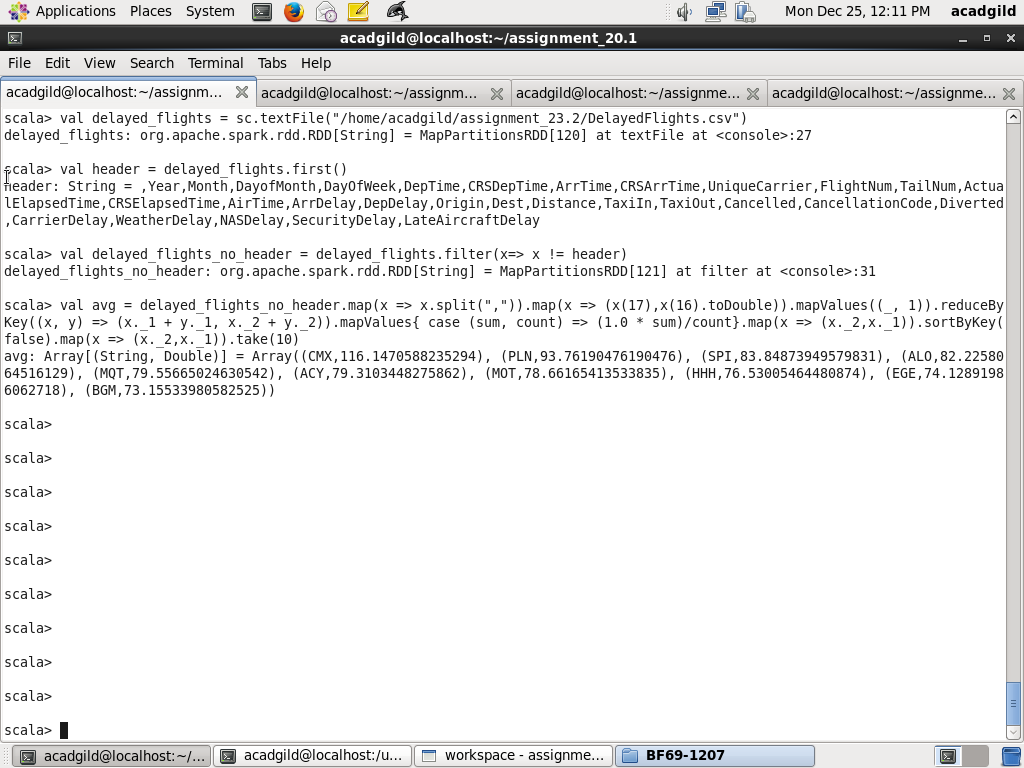
val delayed\_flights\_no\_header = delayed\_flights.filter(x=> x != header)

Step4: Query to get Top ten origins with the highest AVG departure delay

* Split the fields of delayed\_flights\_no\_header based on separator comma.
* Use map to take fields DepDelay, Origin
* Map each tuple to 1
* Use reduceByKey to get total number of dalay and total depDelay
* Calculate Average delay, by dividing total depDelay by total number of delay
* Sort by average delay descending
* Take first 10
* Print the Origin, Average depDelay

val avg = delayed\_flights\_no\_header.map(x => x.split(",")).map(x => (x(17),x(16).toDouble)).mapValues((\_, 1)).reduceByKey((x, y) => (x.\_1 + y.\_1, x.\_2 + y.\_2)).mapValues{ case (sum, count) => (1.0 \* sum)/count}.map(x => (x.\_2,x.\_1)).sortByKey(false).map(x => (x.\_2,x.\_1)).take(10).forach(println)

Screenshot is as below:



Problem Statement 4

Which route (origin & destination) has seen the maximum diversion?

Step1: Load the dataset from /home/acadgild/assignment\_23.2/DelayedFlights.csv and put to RDD delayed\_flights. Code is as below

val delayed\_flights = sc.textFile("/home/acadgild/assignment\_23.2/DelayedFlights.csv")

Step2: Get the header

val header = delayed\_flights.first()

Step3: Remove the header

Remove the heading by filtering tuples which are not header and put to delayed\_flights\_no\_header

val delayed\_flights\_no\_header = delayed\_flights.filter(x=> x != header)

Step4: Query to get Which route (origin & destination) has seen the maximum diversion

* Split the fields of delayed\_flights\_no\_header based on separator comma.
* Filter tuples for which Diverted field is 1
* Using map, concatenate Origin, Dest to 1
* Use reduceByKey to get total number of diversions by Source, Destination
* Sort by total number of diversions descending
* Take first 10
* Print the Origin,Desitnation Number of Diversions

val diversion = delayed\_flights\_no\_header.map(x => x.split(",")).filter(x => ((x(24).equals("1")))).map(x => ((x(17)+","+x(18)),1)).reduceByKey(\_+\_).map(x => (x.\_2,x.\_1)).sortByKey(false).map(x => (x.\_2,x.\_1)).take(10).foreach(println)

