Assignment_46.1_Task3

Name : Silvi Dheer

Core Spark – Task3

Task 3

1) Which route is generating the most revenue per year

2) What is the total amount spent by every user on air-travel per year

3) Considering age groups of < 20 , 20-35, 35 > ,Which age group is travelling the most

every year.

Use the dataset given below:

https://drive.google.com/drive/folders/0B_P3pWagdIrrVThBaUdVSUtzbms

Dataset

Use the dataset given below:

https://drive.google.com/drive/folders/0B_P3pWagdIrrVThBaUdVSUtzbms

Dataset_Holidays.txt, Dataset_User_details.txt and Dataset_Transport.txt.

- The above dataset has the data column wise, userID, Destination, arrival, travel mode, travel distance and the year in the file S18_Dataset_Holidays.txt.
- The dataset Dataset_User_details.txt has columns user ID, name and the age.
- The dataset Dataset_Transport.txt has columns as travel mode and the cost respectively.

Dataset_Holidays.txt

```
1,CHN,IND,airplane,200,1990
2,IND,CHN,airplane,200,1991
3,IND,CHN,airplane,200,1992
4,RUS,IND,airplane,200,1990
5,CHN,RUS,airplane,200,1992
6,AUS,PAK,airplane,200,1991
7,RUS,AUS,airplane,200,1990
8,IND,RUS,airplane,200,1991
9,CHN,RUS,airplane,200,1992
10,AUS,CHN,airplane,200,1993
1,AUS,CHN,airplane,200,1993
2,CHN,IND,airplane,200,1993
3,CHN,IND,airplane,200,1993
4,IND,AUS,airplane,200,1991
5,AUS,IND,airplane,200,1992
6,RUS,CHN,airplane,200,1993
7,CHN,RUS,airplane,200,1990
8,AUS,CHN,airplane,200,1990
9,IND,AUS,airplane,200,1991
10,RUS,CHN,airplane,200,1992
1,PAK,IND,airplane,200,1993
2,IND,RUS,airplane,200,1991
3,CHN,PAK,airplane,200,1991
4,CHN,PAK,airplane,200,1990
```

Dataset_User_details.txt

```
1,mark,15
2,john,16
3,luke,17
4,lisa,27
5,mark,25
6,peter,22
7,james,21
8,andrew,55
9,thomas,46
10,annie,44
```

Dataset_Transport.txt

```
airplane,170
car,140
train,120
ship,200
```

Before that, we are loading the dataset into the spark context,

val baseRDD1 = sc.textFile("/home/acadgild/Desktop/Dataset_Holidays.txt")

val baseRDD2 = sc.textFile("/home/acadgild/Desktop/Dataset_Transport.txt")

val baseRDD3 = sc.textFile("/home/acadgild/Desktop/Dataset_User_details.txt")

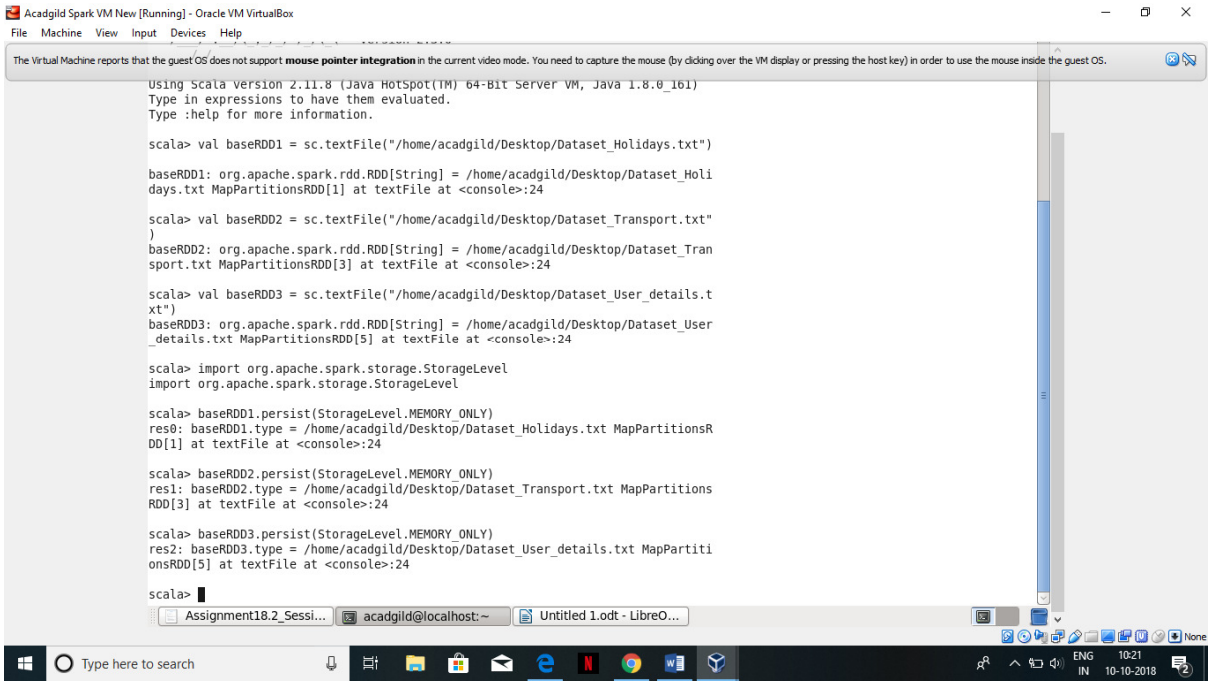
Importing the singleton object for controlling the storage of an RDD,

import org.apache.spark.storage.StorageLevel

baseRDD1.persist(StorageLevel.MEMORY_ONLY)

baseRDD2.persist(StorageLevel.MEMORY_ONLY)

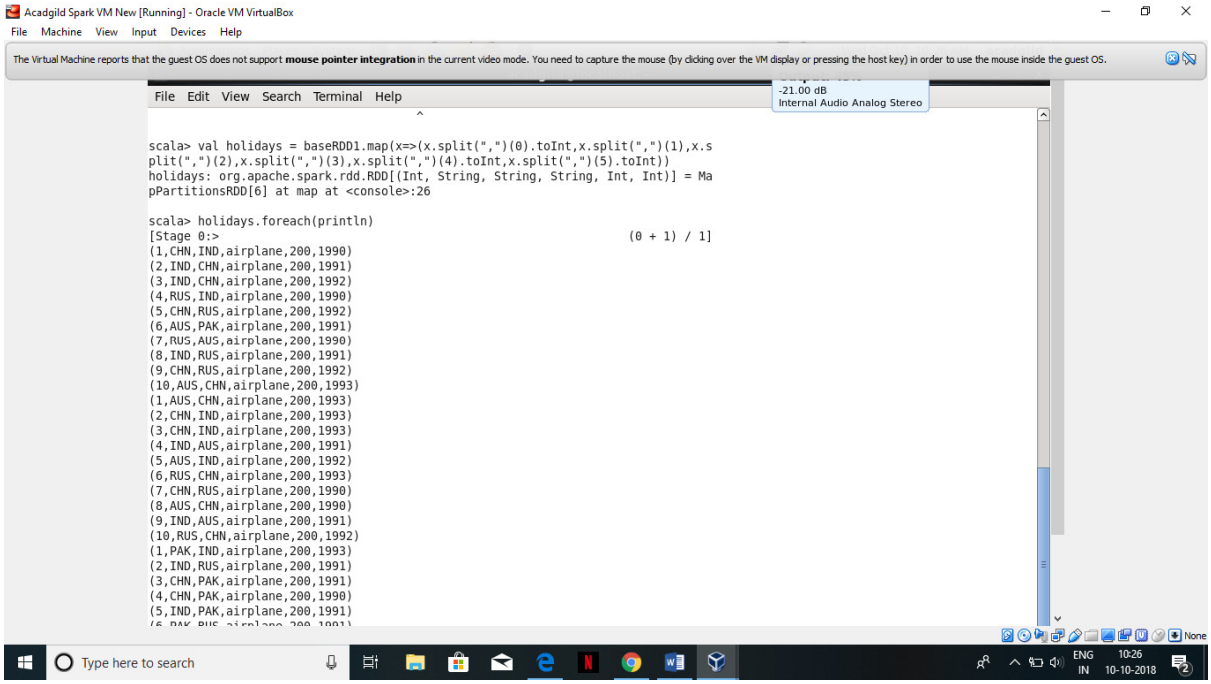
baseRDD3.persist(StorageLevel.MEMORY_ONLY)



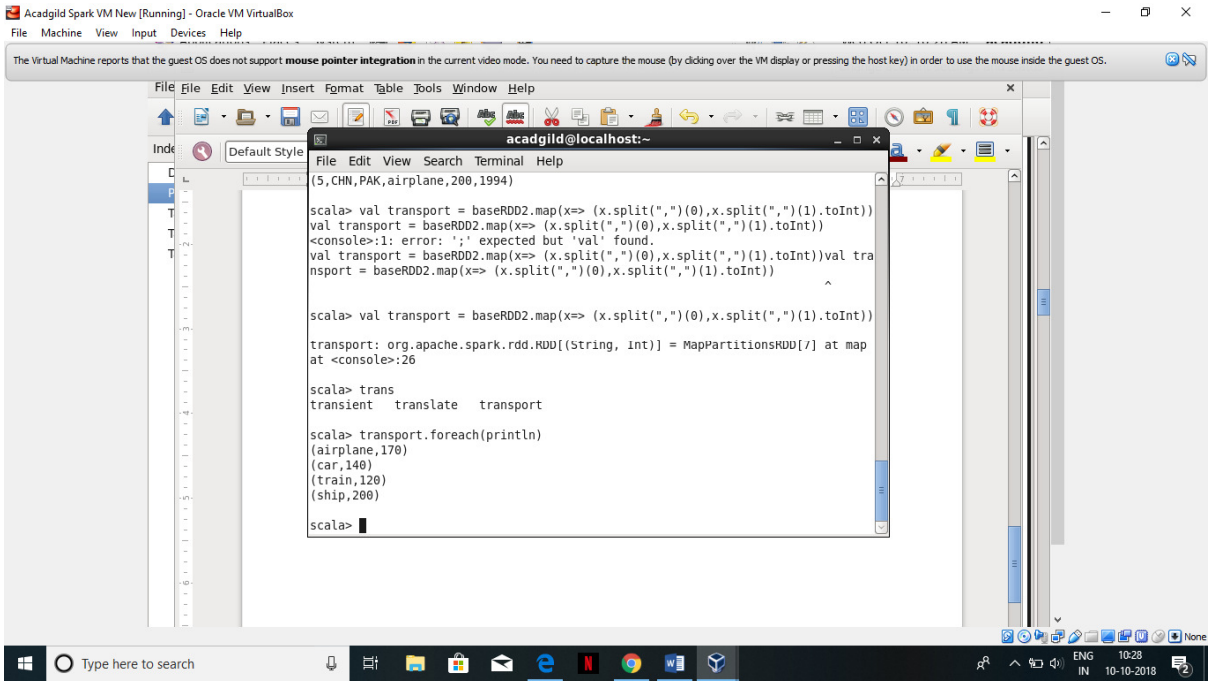
The given dataset's have been loaded and we are creating the tuple RDD columns wise in the spark context,

We are loading the dataset's in the name of holidays, transport and user RDD's.

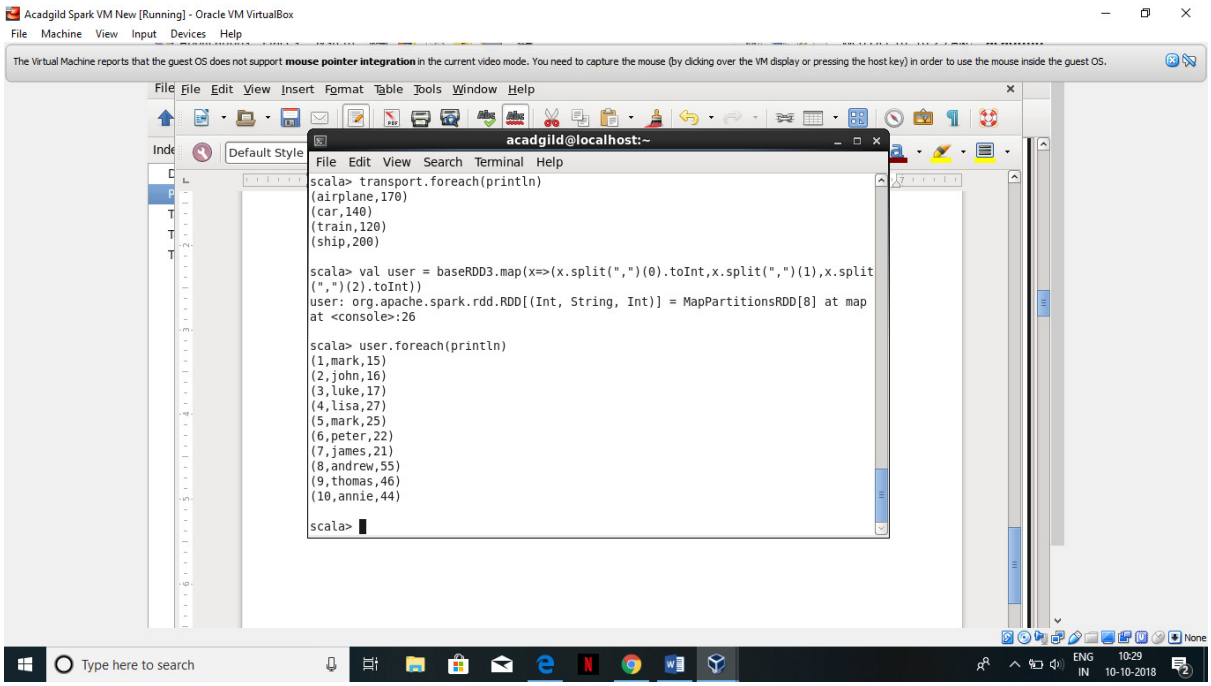
val holidays = baseRDD1.map(x=>(x.split(",")(0).toInt,x.split(",")(1),x.split(",")(2),x.split(",")(3),x.split(",")(4).toInt,x.split(",")(5).toInt))



val transport = baseRDD2.map(x=> (x.split(",")(0),x.split(",")(1).toInt))



val user = baseRDD3.map(x=>(x.split(",")(0).toInt,x.split(",")(1),x.split(",")(2).toInt))



Task 1 - Which route is generating the most revenue per year?

Codes used below,

1. val holidaysmap = holidays.map(x=>x._4->(x._2,x._5,x._6))

2. val transportmap = transport.map(x=>x._1->x._2)

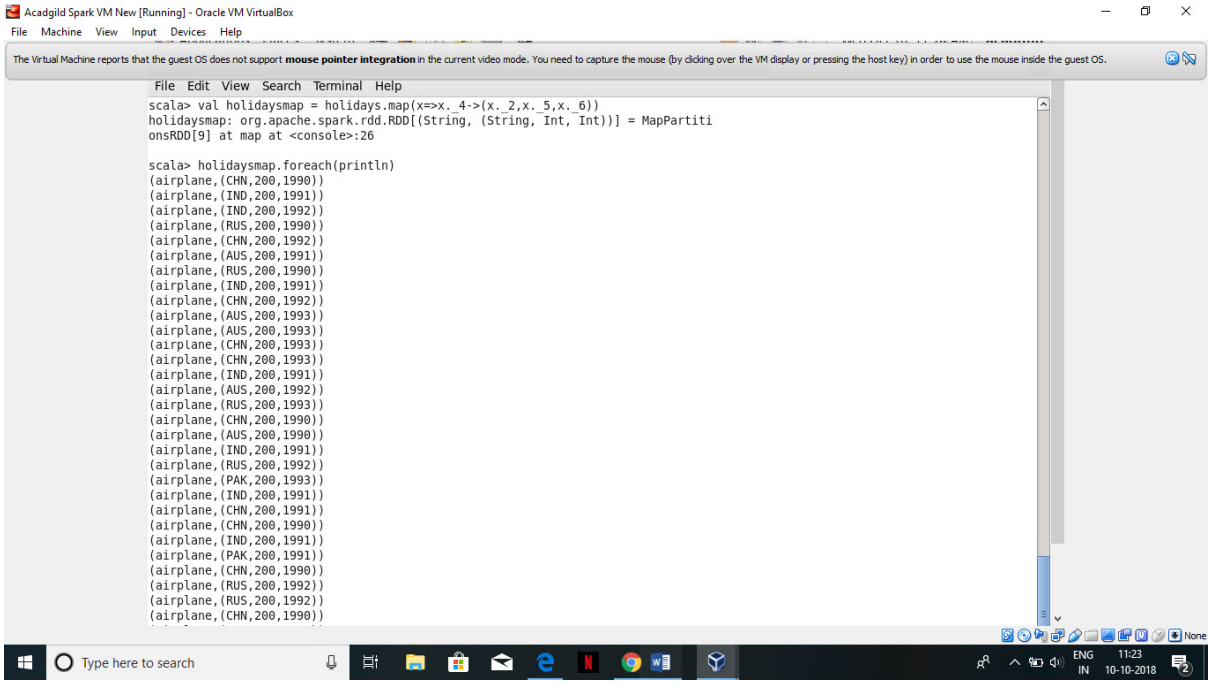
3. val join1 = holidaysmap.join(transportmap)

4. val route = join1.map(x=>(x._2._1._1->x._2._1._3)->(x._2._1._2*x._2._2))

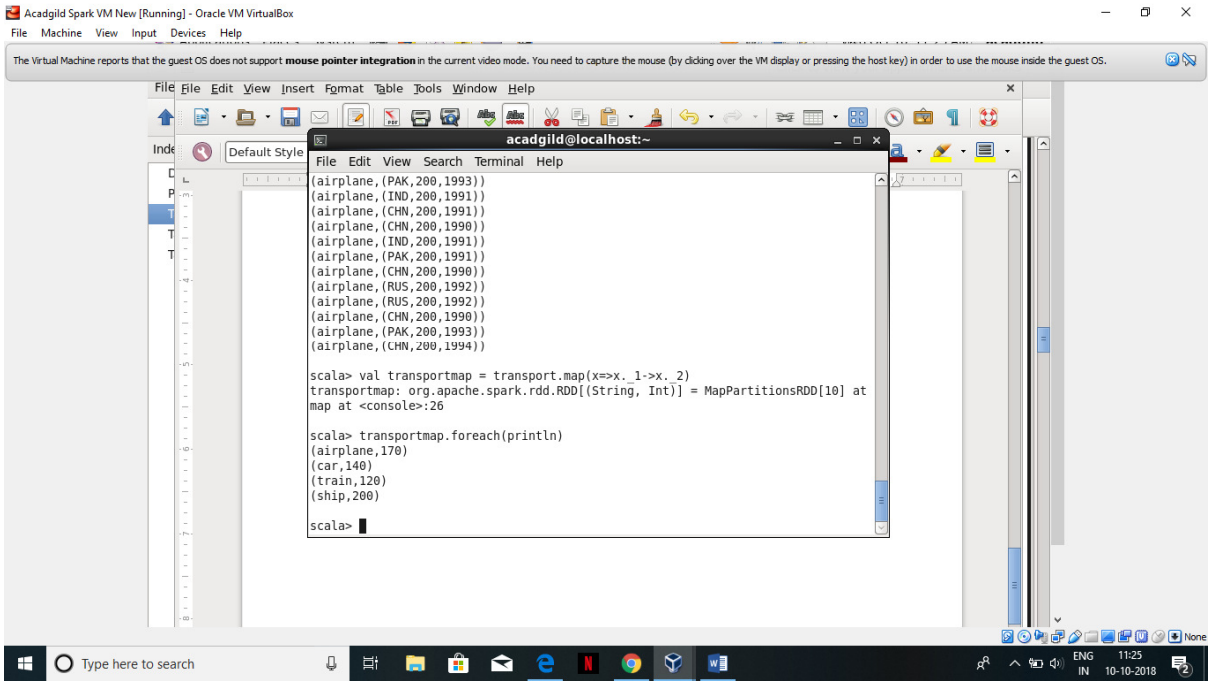
5. val revenue = route.groupByKey().map(x=>x._2.sum->x._1)

6. val routemostrevenue = revenue.sortByKey(false).first()

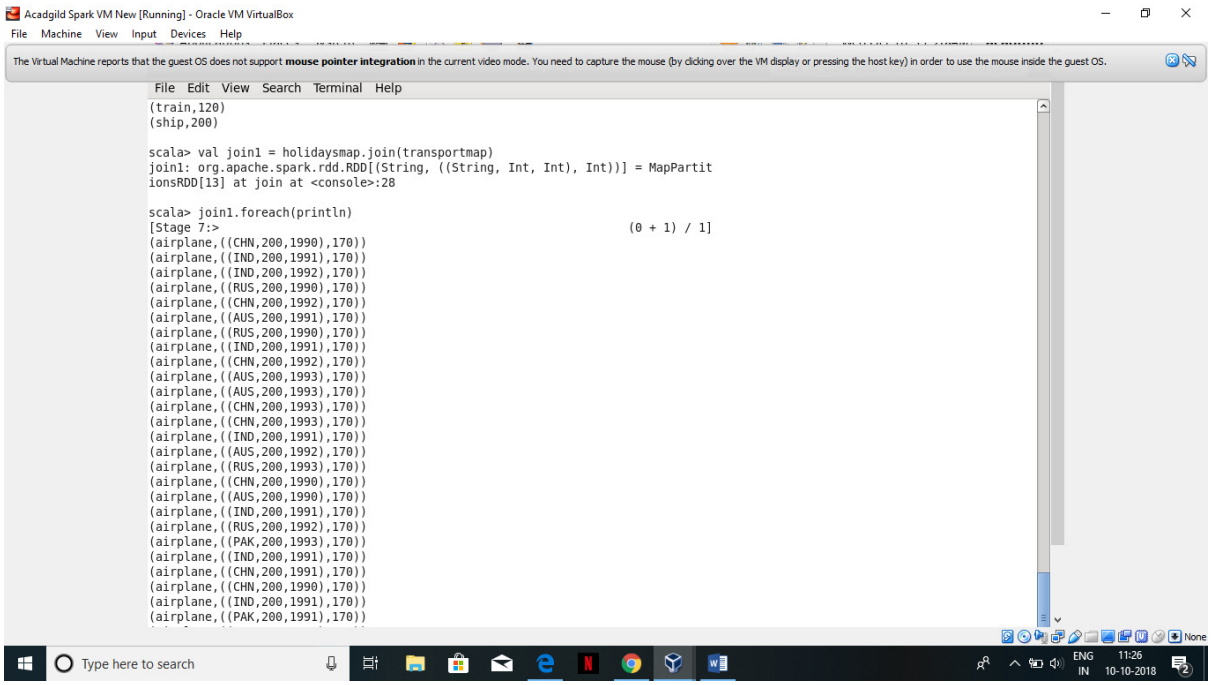
Step 1 – we are mapping the key and value from the base RDD holidays as travel mode as key and the destination, distance and the year as value.



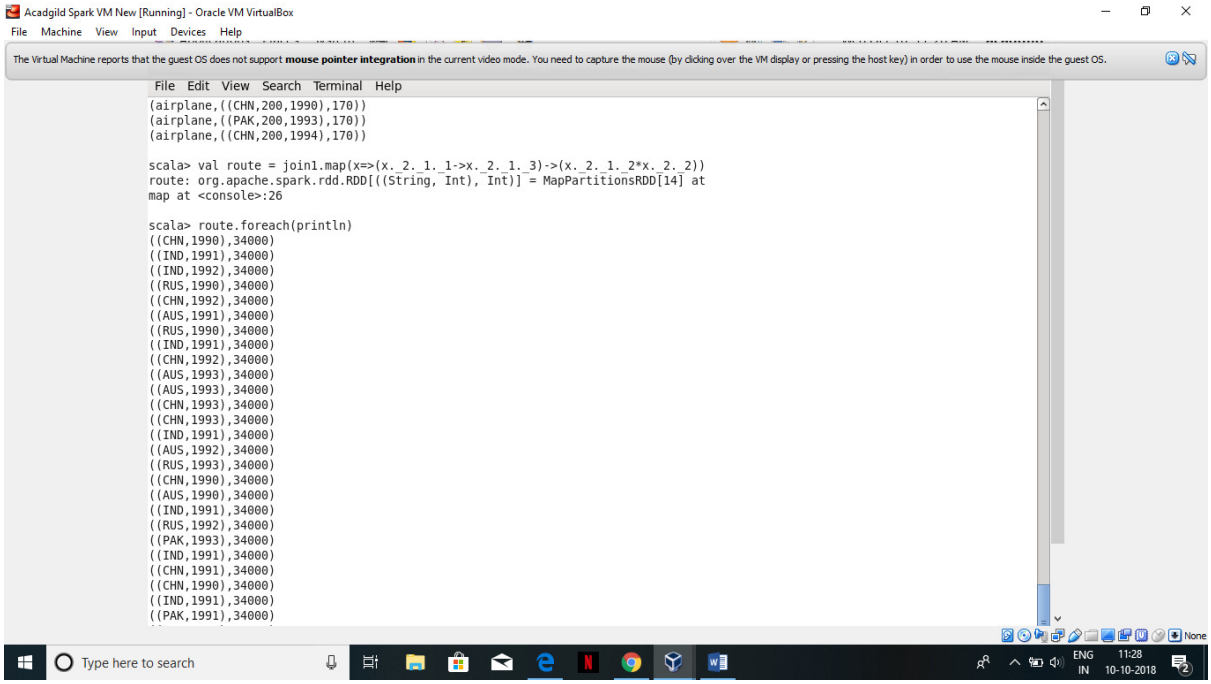
Step -2 – same as, we are creating a tuple RDD as travel mode as key and the rate as values, shown below.



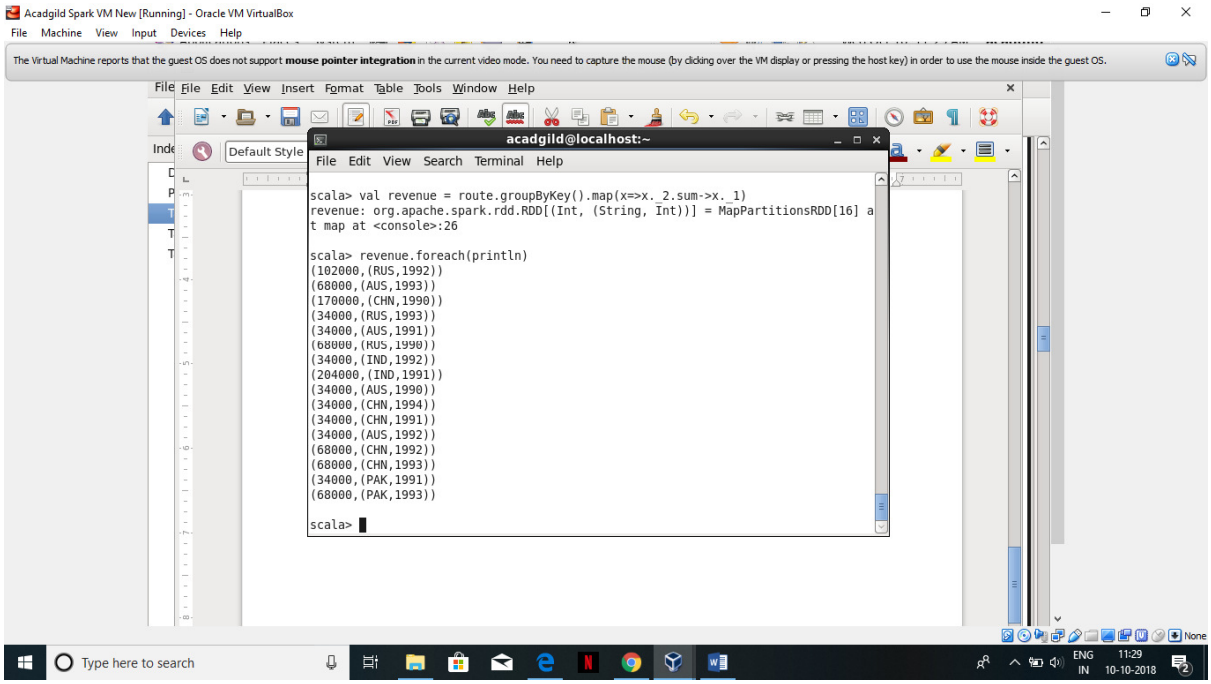
Step 3 – we are joining the 2 RDD's holidaysmap and the transport using join function,



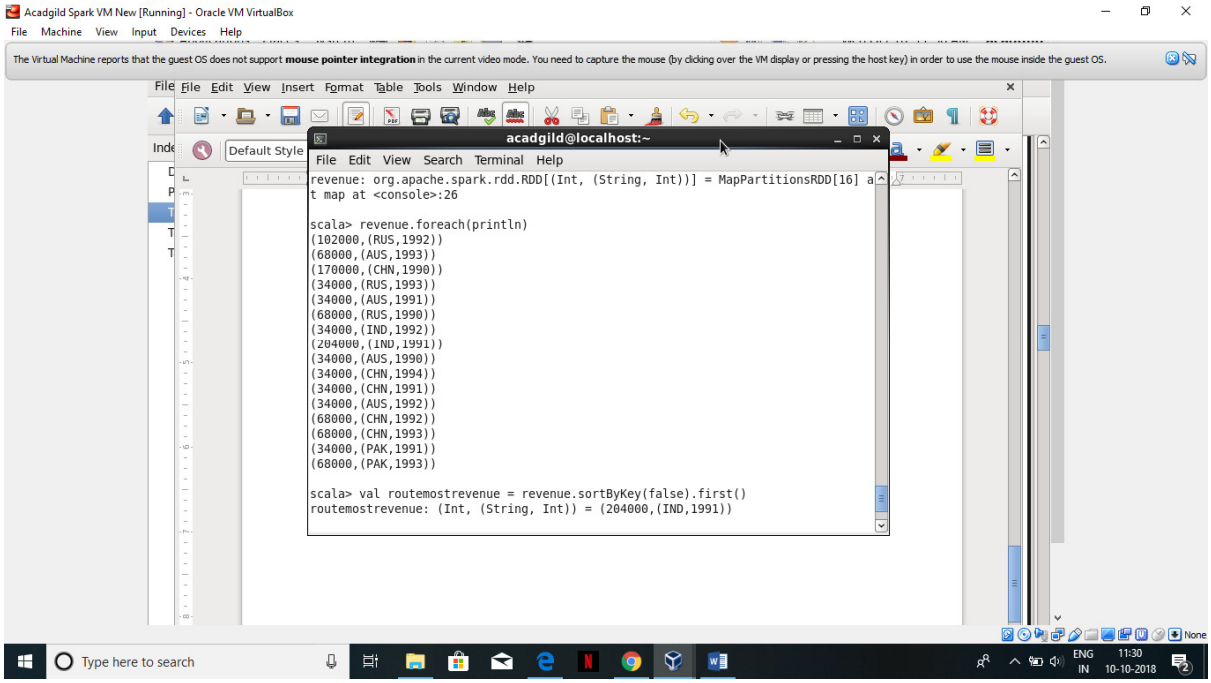
Step 4 – we are mapping the new RDD join1 as below, destination & year as key and the values as multiplication of the cost and the distance.



Step 5 – using groupByKey function, we are grouping the destination & year with the sum of the costs.



Expected output,



Task 2 - What is the total amount spent by every user on air-travel per year?

The codes used for this task below,

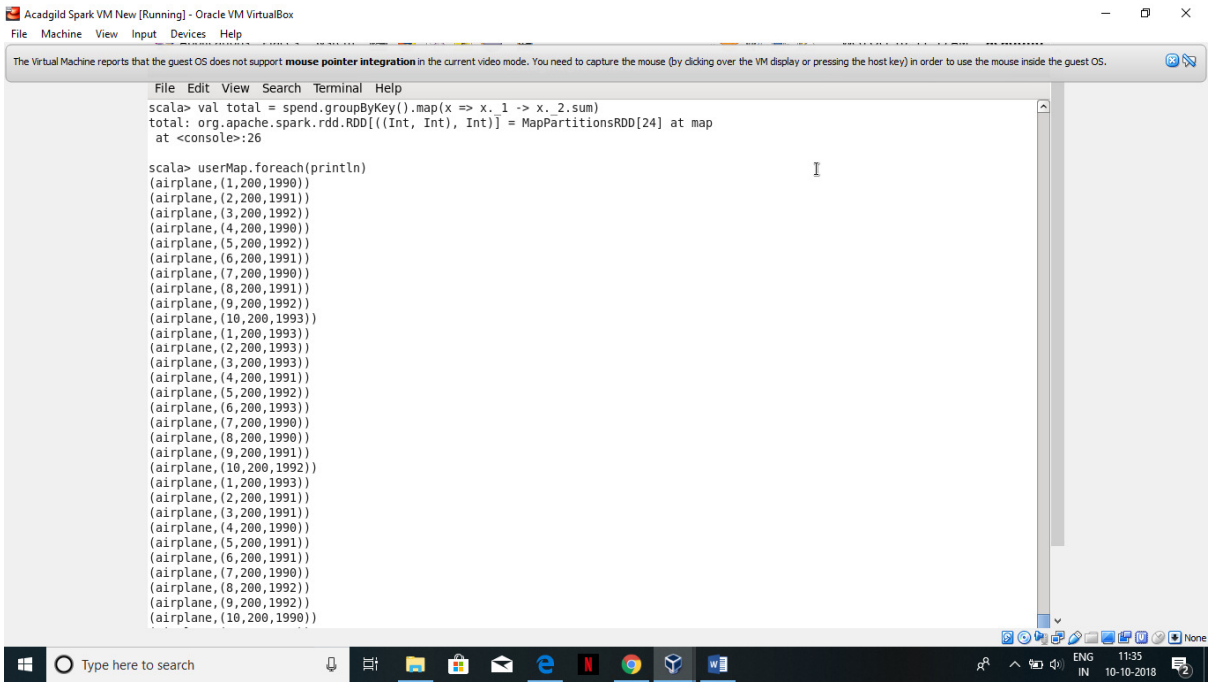
1. val userMap = holidays.map(x => x._4 -> (x._1,x._5,x._6))

2. val amount = userMap.join(transportmap)

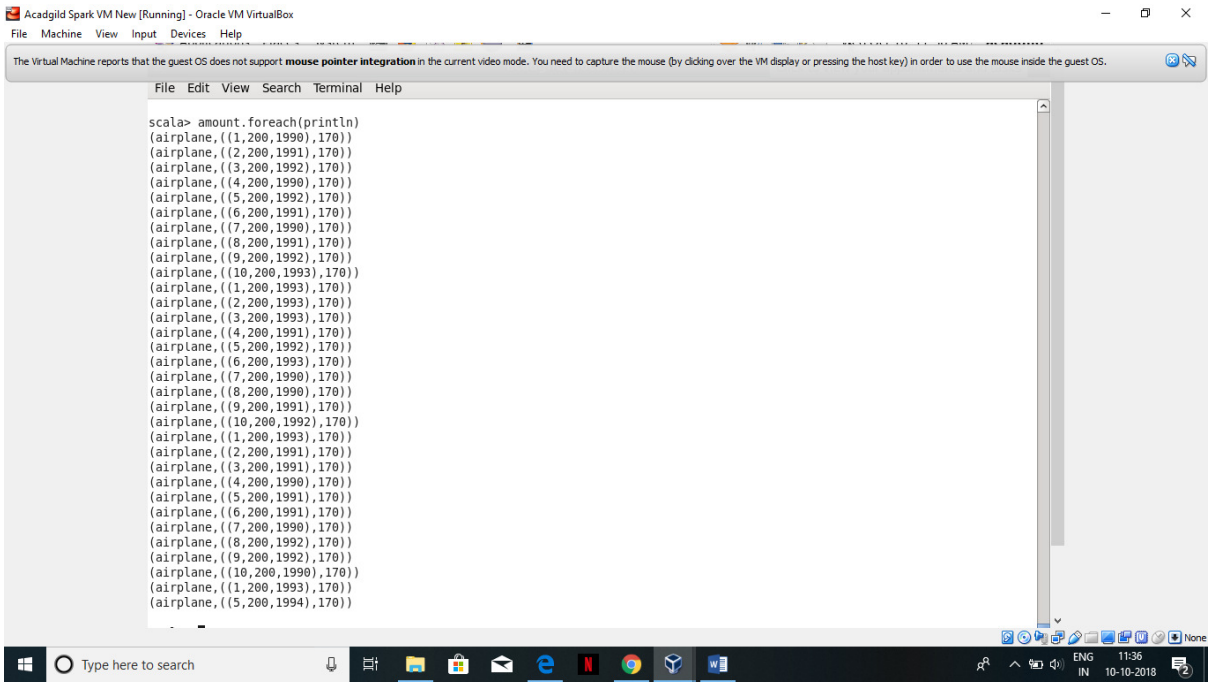
3. val spend = amount.map(x => (x._2._1._1, x._2._1._3) -> (x._2._1._2 * x._2._2))

4. val total = spend.groupByKey().map(x => x._1 -> x._2.sum)

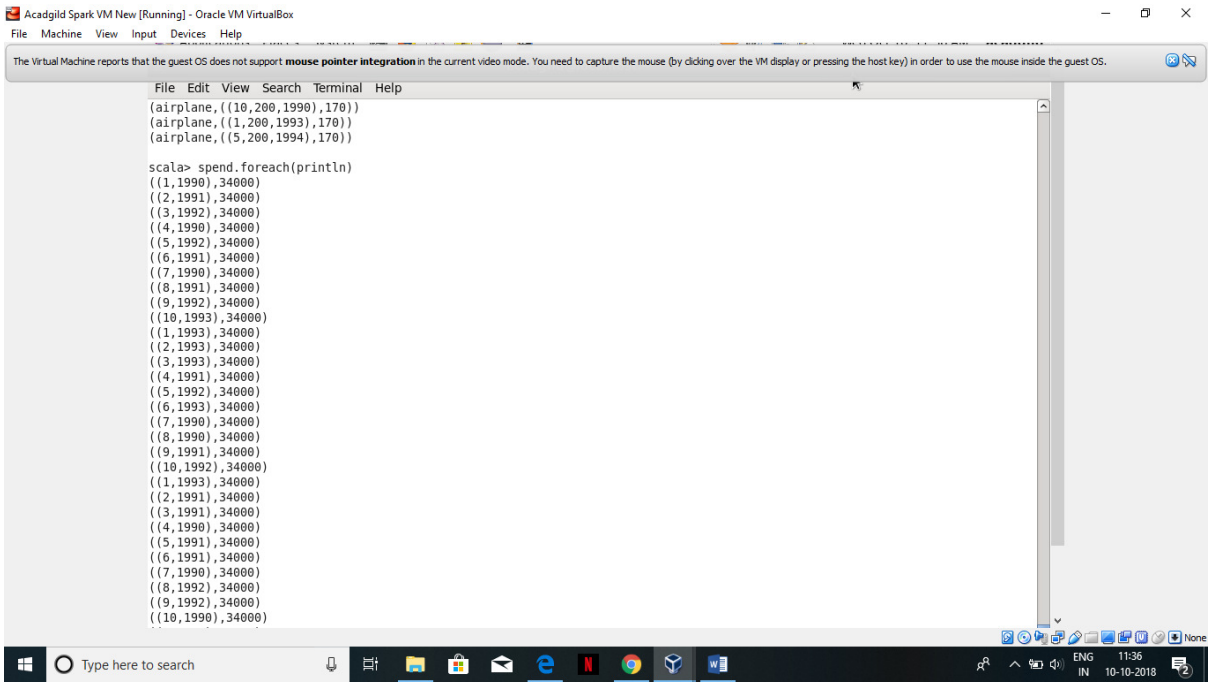
Step -1 – we are creating a tuple rdd from a baseRDD holidays making the travel mode as Key and the userID, distance & year as values.



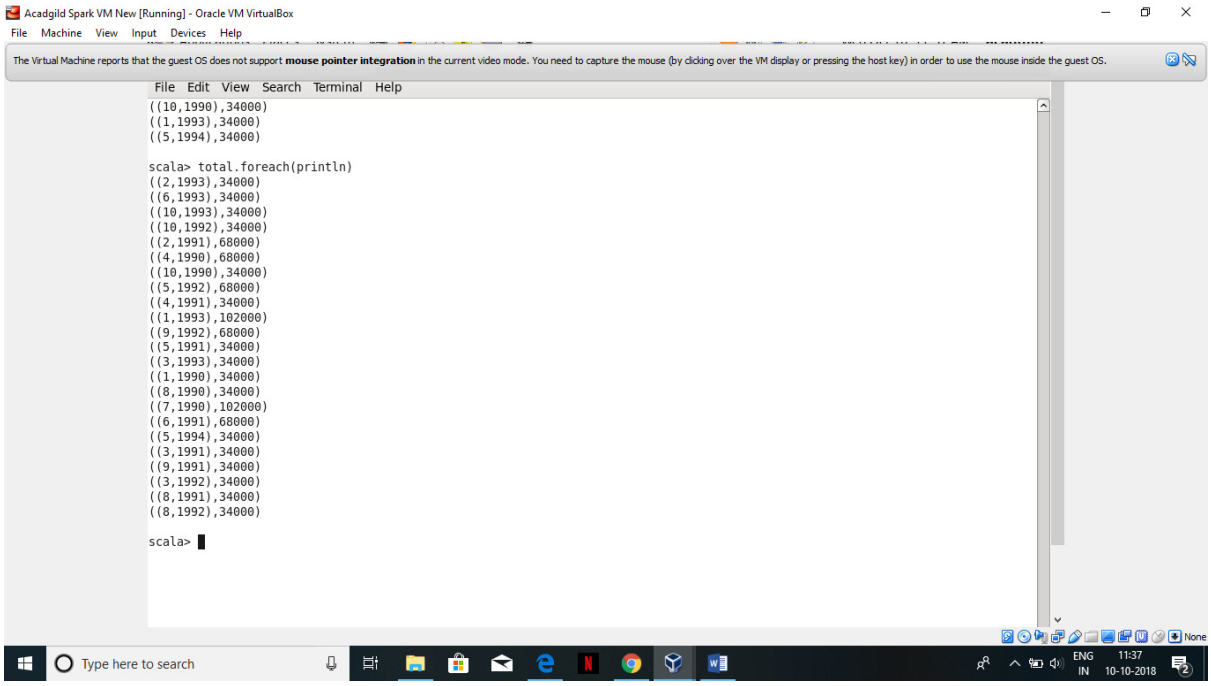
Step -2 – we are joining the created tuple RDD userMap with the already created tuple RDD transportMap using the join function.



Step – 3 – now, we are calculating the expenditure for each user by multiplying the distance and the amount spent for the travel mode airplane,



In the final step, we are summing the total value for each user yearly wise, please see the expected result in the below screen shot.

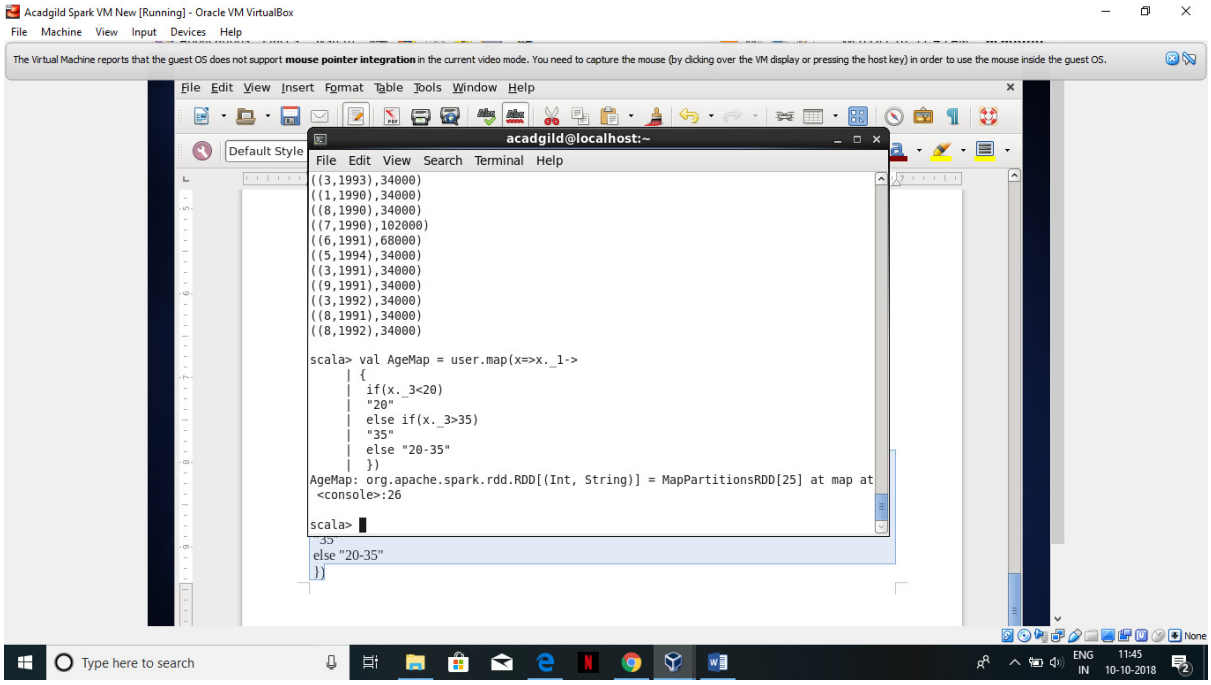


Task 3 - Considering age groups of < 20, 20-35, 35 >, which age group is travelling the most every year.

The codes used for this task are below,

In Order to considering particular age groups, we are using a below if, else logic to define a RDD AgeMap which gives you a set of age groups,

```
val AgeMap = user.map(x=>x._1->
    | {
    | if(x._3<20)
    | "20"
    | else if(x._3>35)
    | "35"
    | else "20-35"
    | })
```



Actual codes,

1. val UserID = holidays.map(x => x._1 -> 1)

2. val joinMap1 = AgeMap.join(UserID)

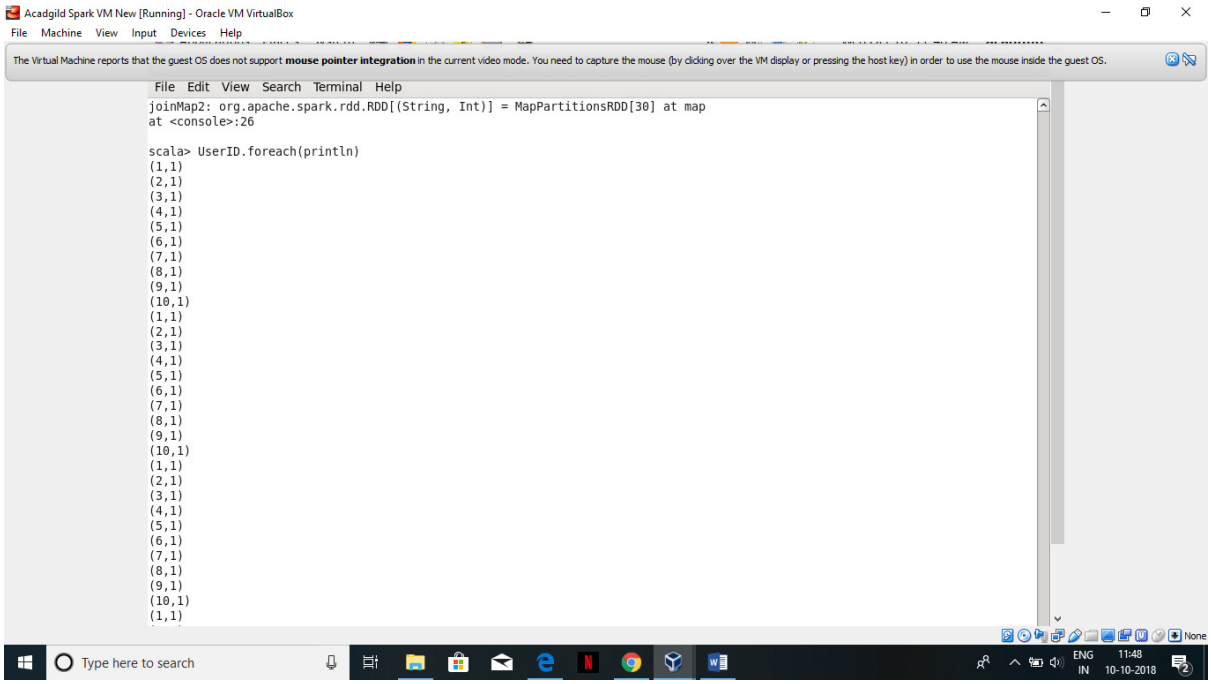
3. val joinMap2 = joinMap1.map(x => x._2._1 -> x._2._2)

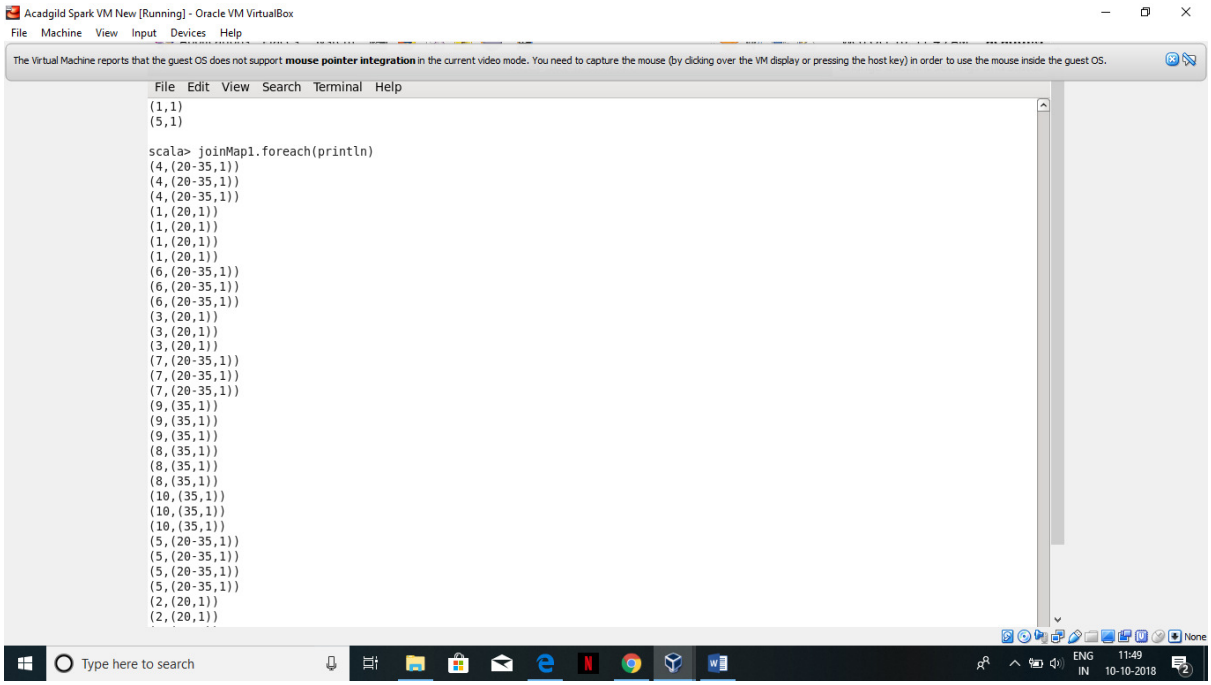
4. val groupKey = joinMap2.groupByKey.map(x => x._1 -> x._2.sum)

5. val mostGroup = groupKey.sortBy(x => -x._2).first()

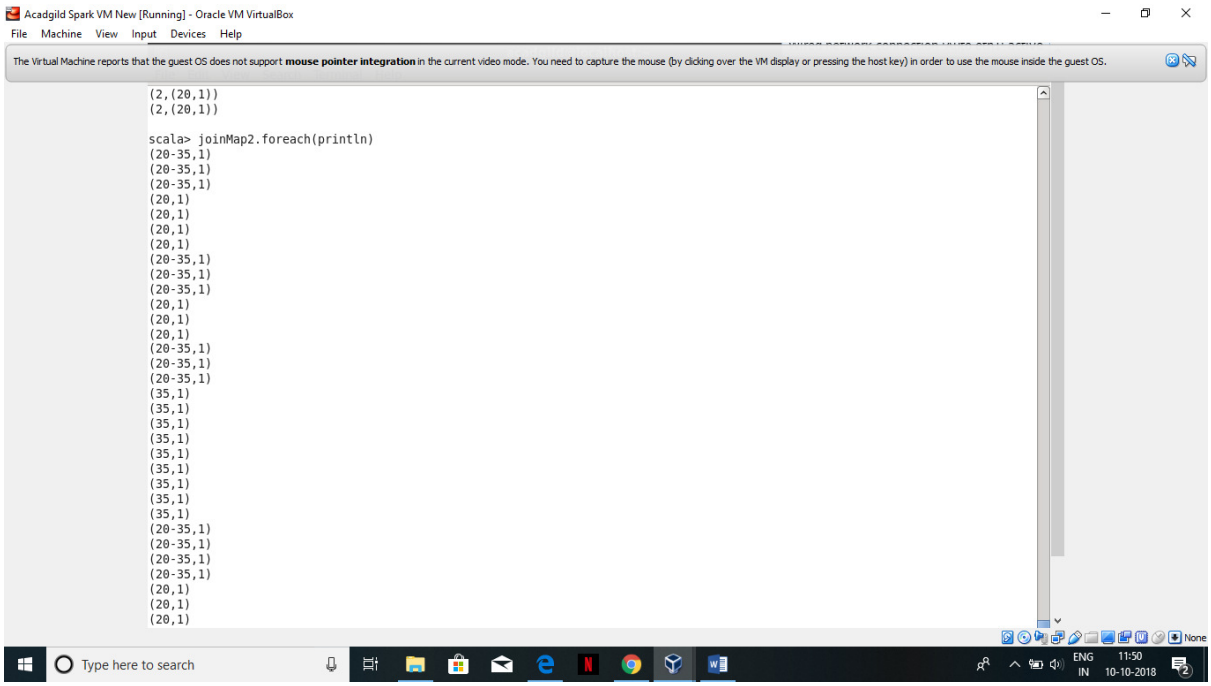
Step – 1- we are just mapping the user ID from the RDD holidays with the numerical 1.

Step -2 – In this step, we are joining the 2 RDD's UserID and the AgeMap, So we are getting the below tuple RDD

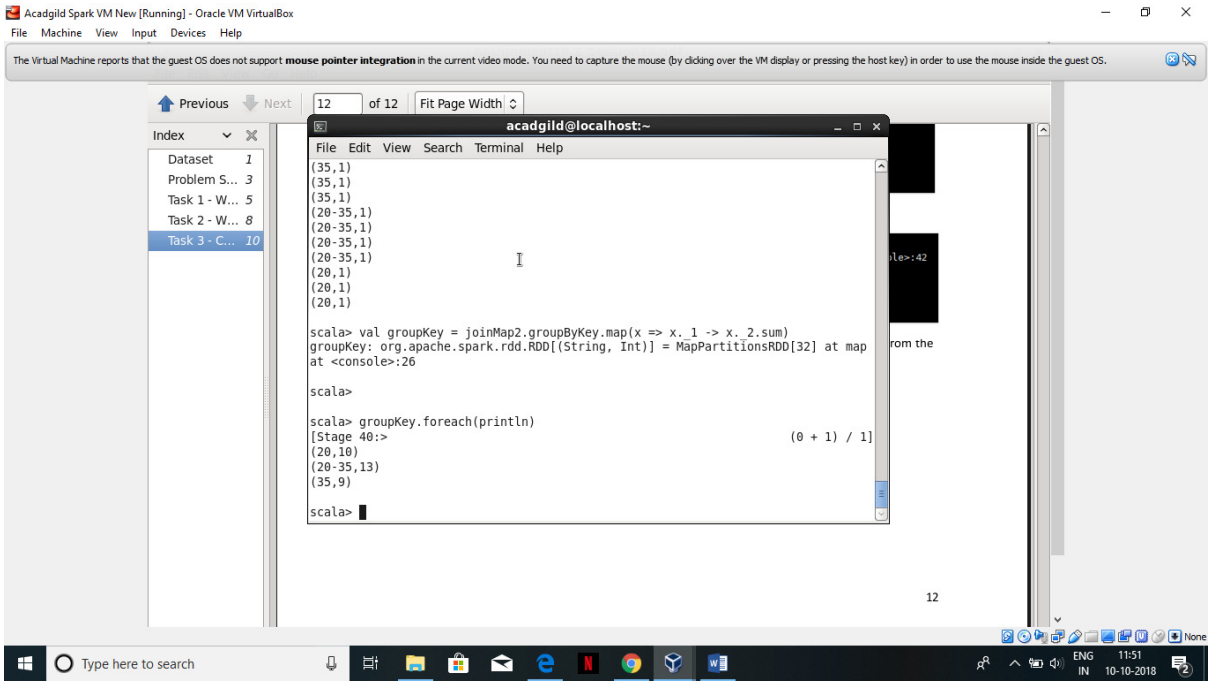




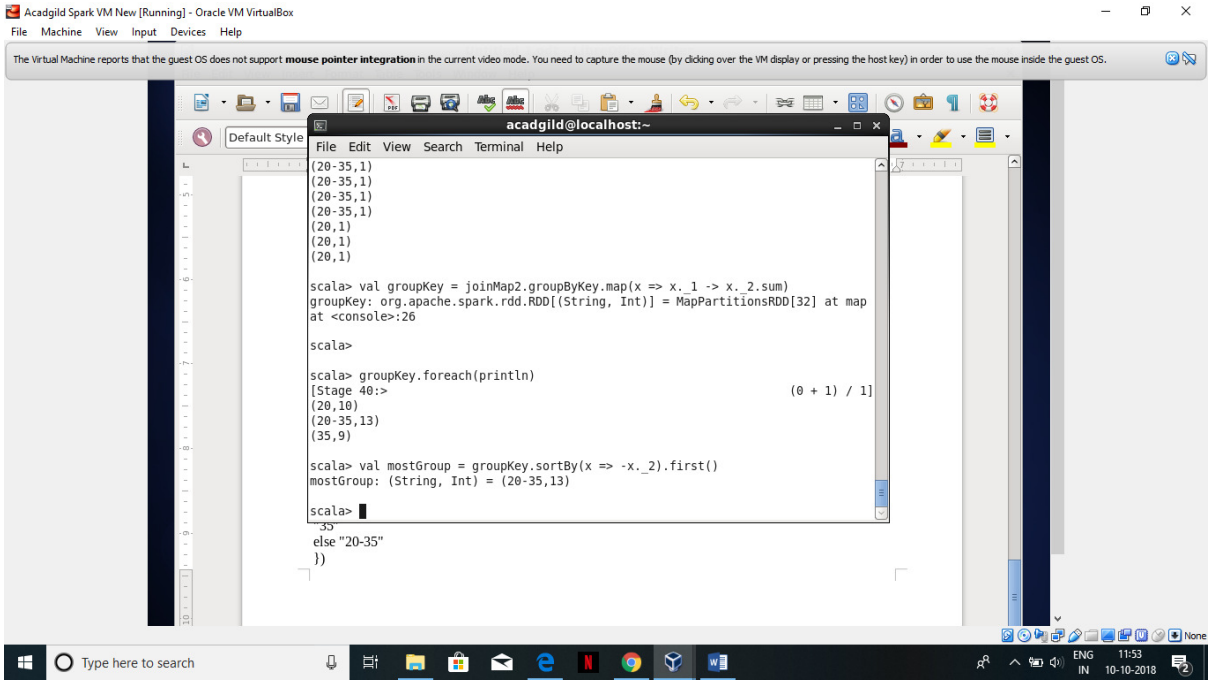
Step -3 – we are just eliminating the user ID in this step



Step – 4 – we just summed the total value by grouping the Age Group,



Step -5 - We use the function first() to find the age group who is travelling the most every year from the given dataset. The expected output shown below,


Acadgild Spark VM New [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
The Virtual Machine reports that the guest OS does not support mouse pointer integration in the current video mode. You need to capture the mouse (by clicking over the VM display or pressing the host key) in order to use the mouse inside the guest OS.
acadgild@localhost:~
File Edit View Search Terminal Help
(20-35,1)
(20-35,1)
(20-35,1)
(20-35,1)
(20,1)
(20,1)
(20,1)
scala> val groupKey = joinMap2.groupByKey.map(x => x._1 -> x._2.sum)
groupKey: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[32] at map at <console>:26
scala>
scala> groupKey.foreach(println)
[Stage 40:> (0 + 1) / 1]
(20,10)
(20-35,13)
(35,9)
scala> val mostGroup = groupKey.sortBy(x => -x._2).first()
mostGroup: (String, Int) = (20-35,13)
scala>
35"
else "20-35"
})
Default Style
Type here to search
ENG
IN
11:53
10-10-2018