

Assignment 46.1 Task2

Name : Silvi Dheer

Topic : Core Spark

Problem Statement

- 1) What is the distribution of the total number of air-travelers per year
- 2) What is the total air distance covered by each user per year
- 3) Which user has travelled the largest distance till date
- 4) What is the most preferred destination for all users.

Use below link to download the dataset:

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

```
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
```

1.) What is the distribution of the total number of air-travelers per year

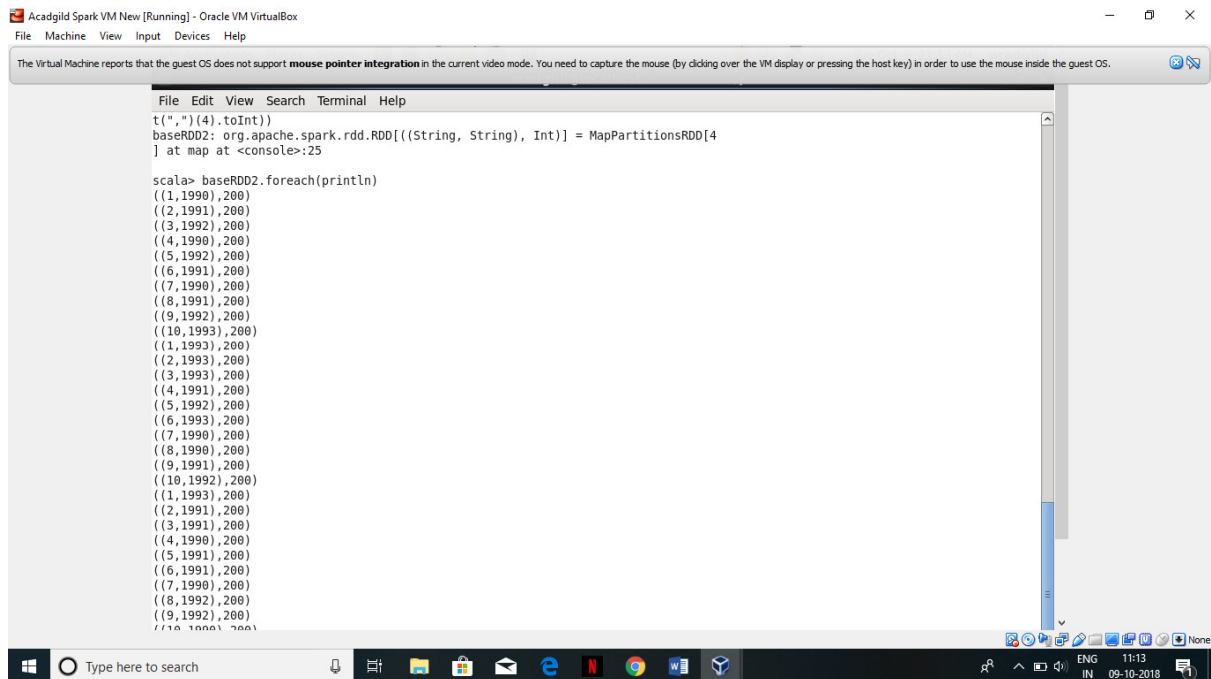
Codes used to achieve the above,

```
1. val baseRDD1 = baseRDD.map(x => (x.split(",")(5).toInt,1))
```

```
2. val no_air_travelers = baseRDD1.reduceByKey((x,y)=>(x+y)).foreach(println)
```

we are creating a tuple RDD baseRDD1 and mapping the key with numerical value 1.

We are creating a tuple rdd “baseRDD2” and mapping the key and value. Here the userID, year acts as key and the travel distance is value.

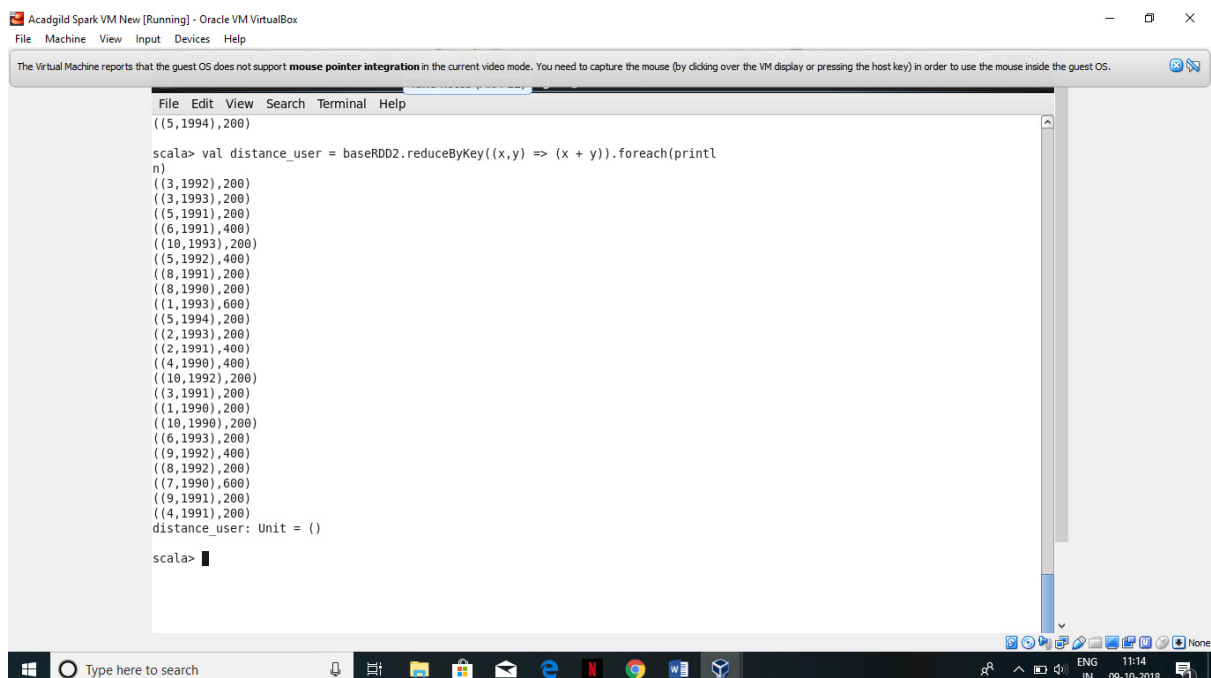


The screenshot shows a virtual machine window titled "Acadgild Spark VM New [Running] - Oracle VM VirtualBox". The terminal window displays the following code and output:

```
File Edit View Search Terminal Help
t(",")(4).toInt))
baseRDD2: org.apache.spark.rdd.RDD[(String, String), Int] = MapPartitionsRDD[4
] at map at <console>:25

scala> baseRDD2.foreach(println)
((1,1990),200)
((2,1991),200)
((3,1992),200)
((4,1990),200)
((5,1992),200)
((6,1991),200)
((7,1990),200)
((8,1991),200)
((9,1992),200)
((10,1993),200)
((1,1993),200)
((2,1993),200)
((3,1993),200)
((4,1991),200)
((5,1992),200)
((6,1993),200)
((7,1990),200)
((8,1990),200)
((9,1991),200)
((10,1992),200)
((1,1993),200)
((2,1991),200)
((3,1991),200)
((4,1990),200)
((5,1991),200)
((6,1991),200)
((7,1990),200)
((8,1992),200)
((9,1992),200)
((10,1993),200)
```

In the second step, we are reducing the number of occurrences using reduceByKey and printing the result, therefore the total air distance covered by each user per year is show below in the screenshot,



The screenshot shows the same virtual machine window. The terminal window displays the following code and output:

```
File Edit View Search Terminal Help
((5,1994),200)

scala> val distance_user = baseRDD2.reduceByKey((x,y) => (x + y)).foreach(println)
((3,1992),200)
((3,1993),200)
((5,1991),200)
((6,1991),400)
((10,1993),200)
((5,1992),400)
((8,1991),200)
((8,1990),200)
((1,1993),600)
((5,1994),200)
((2,1993),200)
((2,1991),400)
((4,1990),400)
((10,1992),200)
((3,1991),200)
((1,1990),200)
((10,1990),200)
((6,1993),200)
((9,1992),400)
((8,1992),200)
((7,1990),600)
((9,1991),200)
((4,1991),200)
distance_user: Unit = ()

scala>
```

3.) Which user has travelled the largest distance till date?

Codes used below,

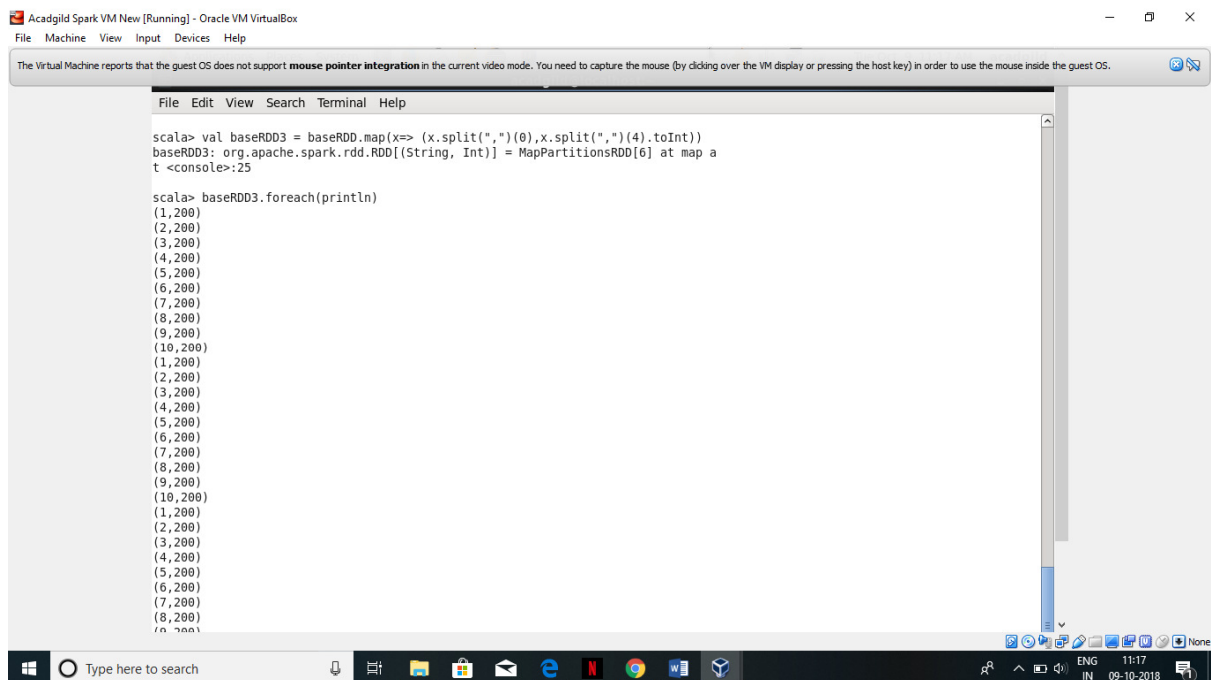
```
1. val baseRDD3 = baseRDD.map(x=> (x.split(",")(0),x.split(",")(4).toInt))
```

```
2. val largest_dist = baseRDD3.reduceByKey((x,y)=>(x+y)).takeOrdered(1)
```

The tuple rdd “baseRDD3” is created to map the key and value from the baseRDD. Here the userID and is key and the travel distance is value,

In the 2nd step, we are reducing the number of occurrences using reduceByKey and using the takeOrdered function to get the result,

```
largest_dist: Array[(String, Int)] = Array((1,800))
```

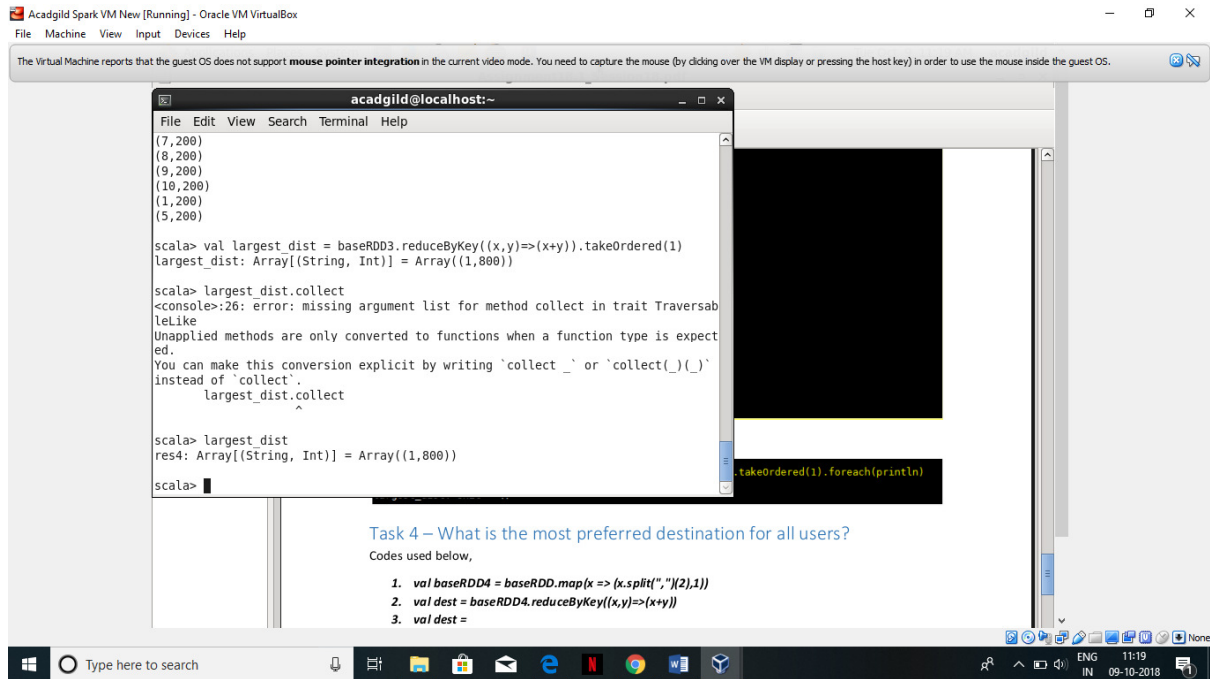


The screenshot shows a Scala REPL window titled "Acadgild Spark VM New [Running] - Oracle VM VirtualBox". The window contains the following code and output:

```
scala> val baseRDD3 = baseRDD.map(x=> (x.split(",")(0),x.split(",")(4).toInt))
baseRDD3: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[6] at map a
t <console>:25

scala> baseRDD3.foreach(println)
(1,200)
(2,200)
(3,200)
(4,200)
(5,200)
(6,200)
(7,200)
(8,200)
(9,200)
(10,200)
(1,200)
(2,200)
(3,200)
(4,200)
(5,200)
(6,200)
(7,200)
(8,200)
(9,200)
(10,200)
(1,200)
(2,200)
(3,200)
(4,200)
(5,200)
(6,200)
(7,200)
(8,200)
(9,200)
(10,200)
```

The output shows a list of 25 tuples, each representing a user ID and a travel distance of 200. The window also displays a message from the virtual machine: "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."



4.) What is the most preferred destination for all users?

Codes used below,

1. `val baseRDD4 = baseRDD.map(x => (x.split(",")(2),1))`
2. `val dest = baseRDD4.reduceByKey((x,y)=>(x+y))`
3. `val dest =`
`baseRDD4.reduceByKey((x,y)=>(x+y)).takeOrdered(1)(Ordering[Int].reverse.on(_._2))`

A tuple rdd created with the destination as key and numerical 1 as value, and we are reducing the number of occurrences using the reduceByKey. Now, the most preferred destination is taken by using the function takeOrdered and ordering the values descending so that we can get the required output.

The output of the each step is shown below

Acadgild Spark VM New [Running] - Oracle VM VirtualBox

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.

```
File Edit View Search Term Browse the Web

scala> val baseRDD4 = baseRDD.map(x => (x.split(",")(2),1))
baseRDD4: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[9] at map a
t <console>:25

scala> baseRDD4.foreach(println)
(IND,1)
(CHN,1)
(CHN,1)
(IND,1)
(RUS,1)
(PAK,1)
(AUS,1)
(RUS,1)
(RUS,1)
(CHN,1)
(CHN,1)
(IND,1)
(IND,1)
(AUS,1)
(IND,1)
(CHN,1)
(RUS,1)
(CHN,1)
(CHN,1)
(AUS,1)
(IND,1)
(RUS,1)
(PAK,1)
(PAK,1)
(PAK,1)
(RUS,1)
(IND,1)
(IND,1)
(IND,1)
(AUS,1)
```

Type here to search

Acadgild Spark VM New [Running] - Oracle VM VirtualBox

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.

```
File Edit View Search Terminal Help

(RUS,1)
(PAK,1)
(PAK,1)
(PAK,1)
(RUS,1)
(IND,1)
(IND,1)
(IND,1)
(AUS,1)
(PAK,1)
(PAK,1)

scala> val dest = baseRDD4.reduceByKey((x,y)=>(x+y))
dest: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[10] at reduceByKey a
t <console>:25

scala> dest.foreach(println)
(CHN,7)
(IND,9)
(PAK,5)
(RUS,6)
(AUS,5)

scala>

The required output,

scala> val dest = baseRDD4.reduceByKey((x,y)=>(x+y)).takeOrdered(1)(Ordering[Int].reverse.on(_._2))
dest: Array[(String, Int)] = Array((IND,9))
```

Type here to search

The Required Output -

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  
(AUS,1)  
(AUS,1)  
(PAK,1)  
  
scala> val dest = baseRDD4.reduceByKey((x,y)=>(x+y))  
dest: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[10] at reduceByKey a  
t <console>:25  
  
scala> dest.foreach(println)  
(CHN,7)  
(IND,9)  
(PAK,5)  
(RUS,6)  
(AUS,5)  
  
scala> val dest =  
    | baseRDD4.reduceByKey((x,y)=>(x+y)).takeOrdered(1)(Ordering[Int].reverse.o  
n(_._2))  
dest: Array[(String, Int)] = Array((IND,9))  
  
scala> dest.foreach(println)  
(IND,9)  
  
scala>   
  
val dest =  
baseRDD4.reduceByKey((x,y)=>(x+y)).takeOrdered(1)(Ordering[Int].reverse.on(_._2))
```