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

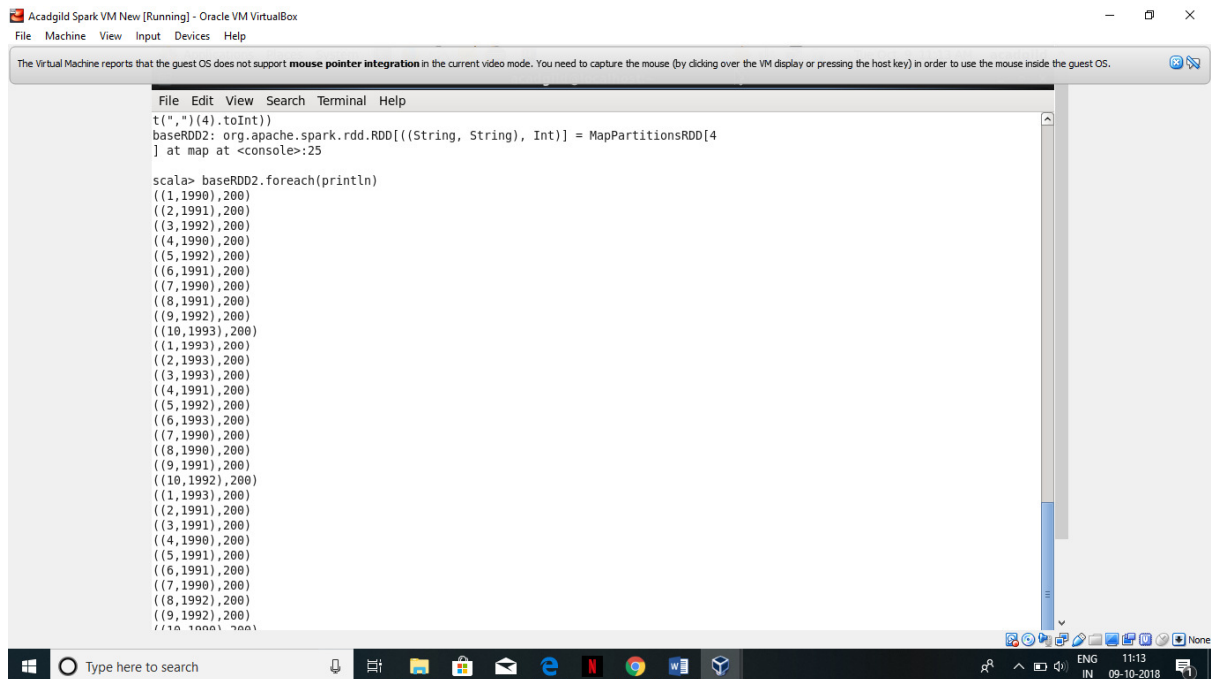### 1.) What is the distribution of the total number of air-travelers per year

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.

```
File  Edit  View  Search  Term  Browse the Web
ays.txt MapPartitionsRDD[1] at textFile at <console>:24

scala> val baseRDD1 = baseRDD.map(x => (x.split(",")(5).toInt,1))
baseRDD1: org.apache.spark.rdd.RDD[(Int, Int)] = MapPartitionsRDD[2] at map at <
console>:25

scala> baseRDD1.foreach(println)
[Stage 0:>                                              (0 + 1) / 1]
(1990,1)
(1991,1)
(1992,1)
(1990,1)
(1992,1)
(1991,1)
(1990,1)
(1991,1)
(1992,1)
(1993,1)
(1993,1)
(1993,1)
(1993,1)
(1991,1)
(1992,1)
(1993,1)
(1990,1)
(1990,1)
(1991,1)
(1992,1)
(1993,1)
(1991,1)
(1991,1)
(1990,1)
(1991,1)
(1991,1)
(1990,1)
```

We are reducing the number of occurrences using reduceByKey and printing the result. Therefore, Total no of air travelers per year is,


```
acadgild@localhost:~
File  Edit  View  Search  Terminal  Help
(1993,1)
(1991,1)
(1991,1)
(1990,1)
(1991,1)
(1991,1)
(1990,1)
(1992,1)
(1992,1)
(1990,1)
(1993,1)
(1994,1)

scala> val no_air_travelers = baseRDD1.reduceByKey((x,y)=>(x+y)).foreach(println
)
[Stage 1:>                                              (0 + 1) / 1]
(1994,1)
(1992,7)
(1990,8)
(1991,9)
(1993,7)
no_air_travelers: Unit = ()

scala>
```

```
val baseRDD = sc.textFile("/home/acadgild/Desktop/Dataset_Holidays.txt")
val baseRDD1 = baseRDD.map(x => (x.split(",")(5).toInt,1))
val no_air_travelers = baseRDD1.reduceByKey((x,y)=>(x+y)).foreach(println)
```

2.) What is the total air distance covered by each user per year?

Codes used to achieve the above,

1. val baseRDD2 = baseRDD.map(x => ((x.split(",")(0),x.split(",")(5)),x.split(",")(4).toInt))

2. val distance_user = baseRDD2.reduceByKey((x,y) => (x + y)).foreach(println)

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.
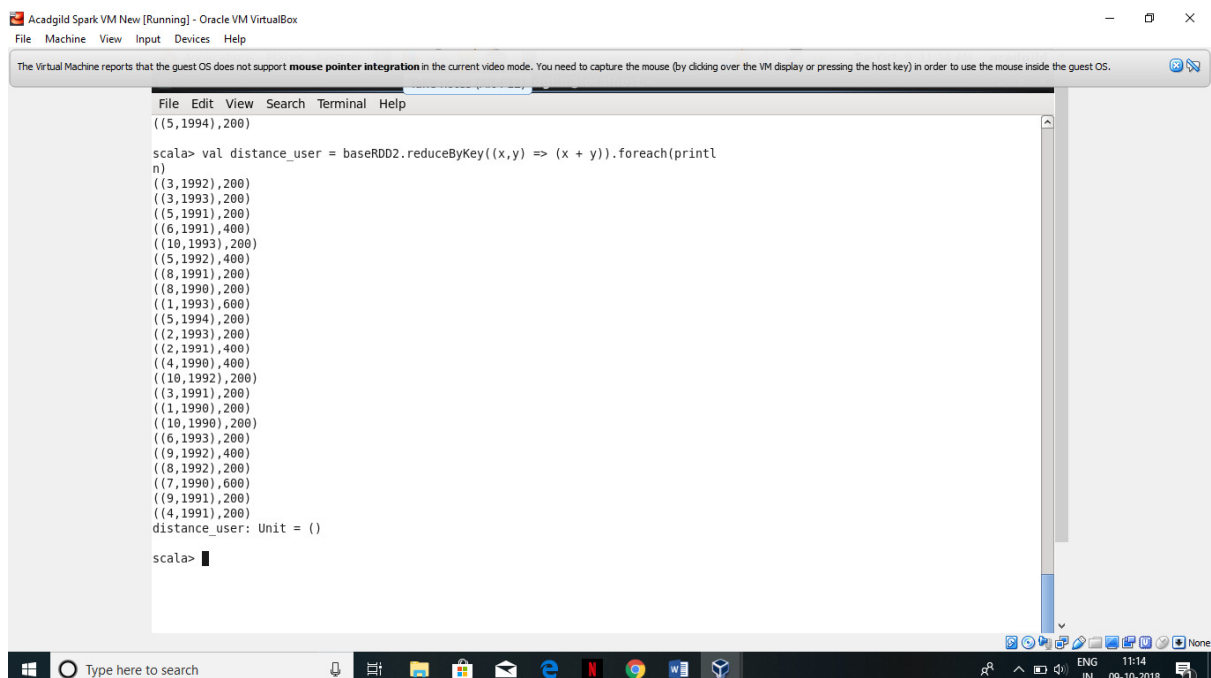

In the second step, we are reducing the number of occurrences using reduceByKey and printing the result, therefore the total air distance covered by each user per year is show below in the screenshot,

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))

```
acadgild@localhost:~

File  Edit  View  Search  Terminal  Help
(7,200)
(8,200)
(9,200)
(10,200)
(1,200)
(5,200)

scala> val largest_dist = baseRDD3.reduceByKey((x,y)=>(x+y)).takeOrdered(1)
largest_dist: Array[(String, Int)] = Array((1,800))

scala> largest_dist.collect
<console>:26: error: missing argument list for method collect in trait Traversab
leLike
Unapplied methods are only converted to functions when a function type is expect
ed.
You can make this conversion explicit by writing `collect _` or `collect(_)(_)`
instead of `collect`.
       largest_dist.collect
                    ^

scala> largest_dist
res4: Array[(String, Int)] = Array((1,800))

scala>
```

`.takeOrdered(1).foreach(println)`

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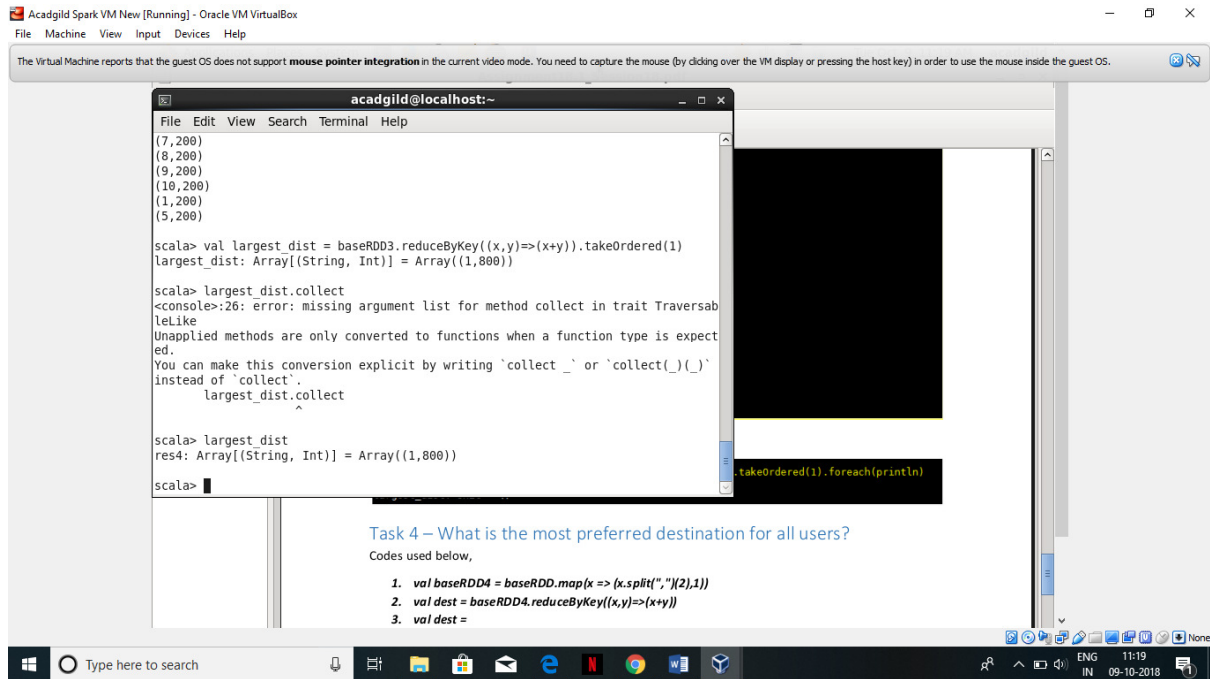
4.) What is the most preferred destination for all users?

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

File   Edit   View   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)
(AUS,1)
(CHN,1)
(IND,1)
(RUS,1)
(PAK,1)
(PAK,1)
(PAK,1)
(RUS,1)
(IND,1)
(IND,1)
(IND,1)
(AUS,1)
```

acadgild@localhost:~

File   Edit   View   Search   Terminal   Help

```
(RUS,1)
(PAK,1)
(PAK,1)
(PAK,1)
(RUS,1)
(IND,1)
(IND,1)
(IND,1)
(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>
```

] at reduceByKey at <console>:29

(PAK,5)
(RUS,6)
(AUS,5)

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))
```

The Required Output -

```
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))