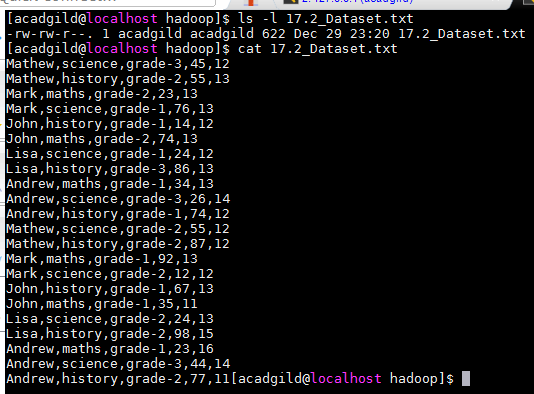
Assignment 17.2

Below is the dataset which we will be using for this Assignment in all problems. It has been kept in local file system:-



Problem Statement 1:-

1. Read the text file, and create a tupled rdd.
2. Find the count of total number of rows present.
3. What is the distinct number of subjects present in the entire school
4. What is the count of the number of students in the school, whose name is Mathew and marks is 55

Solution:-

# Read the text file, and create a tupled rdd

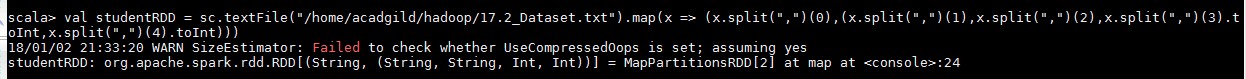
Below is the code used –

* val studentRDD = sc.textFile("/home/acadgild/hadoop/17.2\_Dataset.txt").map(x => (x.split(",")(0),(x.split(",")(1),x.split(",")(2),x.split(",")(3).toInt,x.split(",")(4

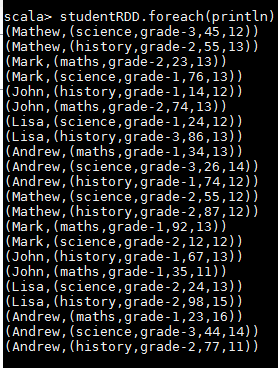
).toInt)))

* studentRDD.foreach(println)

Now first we are creating a RDD named as studentRDD using the spark context to read the input file and splitting the lines based on delimiter “,” to create a tuple.

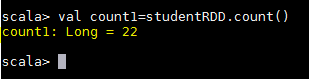


As shown below it as tupled RDD with name as key and subject,grade,marks and age as values:-



# The count of total number of rows present

We are using count() function to find the number of rows present as shown below-



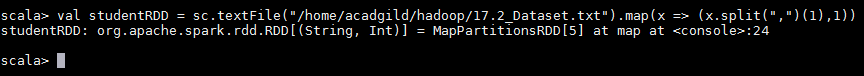
# The distinct number of subjects present in the entire school

Below is the code used:-

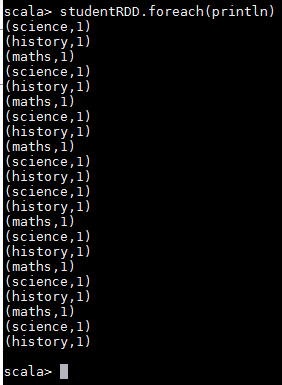
* val studentRDD = sc.textFile("/home/acadgild/hadoop/17.2\_Dataset.txt").map(x => (x.split(",")(1),1))
* val subCount = studentRDD.reduceByKey((x,y) => x + y)
* subCount.foreach(println)

Now we will try to understand each and every command one by one-

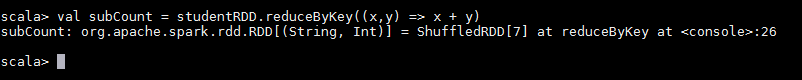
First we are creating a RDD to read the file and selecting only subject name and mapping them with value 1:-



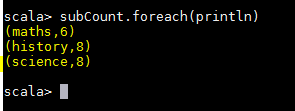
Below shows the result of above RDD:-



Now we are counting the values of occurrences using reduceByKey to get number of subjects



Below screenshot shows the subject name and their count:-



# The count of the number of students in the school, whose name is Mathew and marks is 55

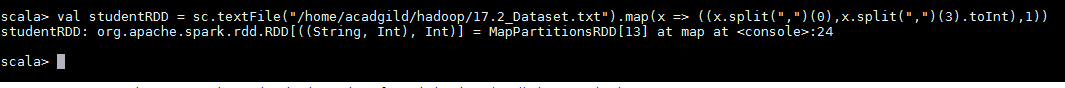
Below is the code used:-

* val studentRDD = sc.textFile("/home/acadgild/hadoop/17.2\_Dataset.txt").map(x => ((x.split(",")(0),x.split(",")(3).toInt),1))
* val filterRDD = studentRDD.filter(x => x.\_1.\_1 == "Mathew" && x.\_1.\_2

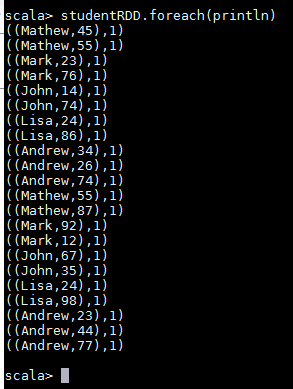
== "55")

* val reduceVal = filterRDD.reduceByKey((x,y) => x + y)

First we are creating a RDD to read the file and selecting name and marks as key and mapping them with value 1:-



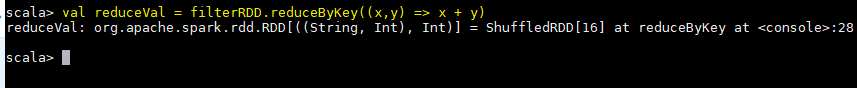
Below screenshot shows the result of the same:-



Now we are filtering above result which contains name as Mathew and marks equal to 55



Now we are counting each occurrences:-



Below screenshot shows the result for the same:-



Problem Statement 2:-

1. What is the count of students per grade in the school?
2. Find the average of each student (Note - Mathew is grade-1, is different from Mathew in some other grade!)
3. What is the average score of students in each subject across all grades?
4. What is the average score of students in each subject per grade?
5. For all students in grade-2, how many have average score greater than 50?

Solution:-

# The count of students per grade in the school

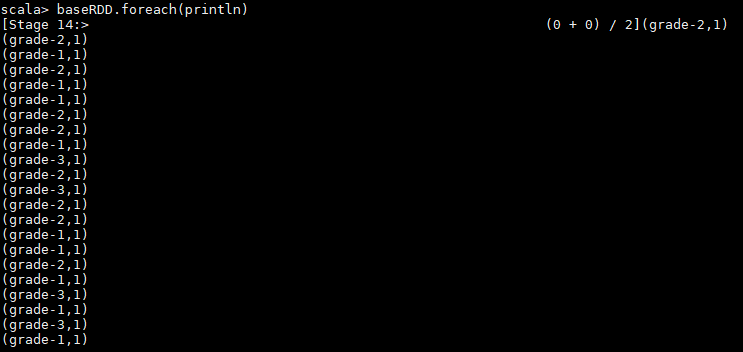
Below is the code used for same:-

* val baseRDD = sc.textFile("/home/acadgild/Assignment- 17.2/17.2\_Dataset.txt").map(x => (x.split(",")(2),1))
* val studCount = baseRDD.reduceByKey((x,y) => x + y )

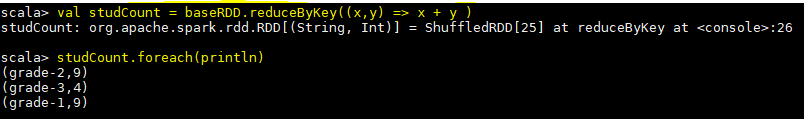
First we are creating a RDD to read the file and selecting only grade and mapping 1 with it:-



Below is the result of above RDD:-



Now are using reduuceByKey to add the occurrences of the grades. Below screenshot shows same along with the result:-



# The average of each student (Note - Mathew is grade-1, is different from Mathew in some other grade!)

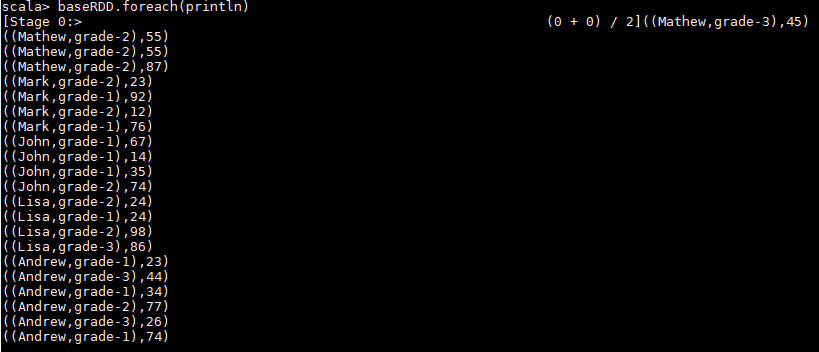
Below is the code used to find the result:-

* val baseRDD = sc.textFile("/home/acadgild/Assignment- 17.2/17.2\_Dataset.txt").map(x => ((x.split(",")(0),x.split(",")(2)),x.split(",")(3).toInt))
* val studAvg = baseRDD.mapValues(x => (x,1))
* val studReduce = studAvg.reduceByKey((x,y) => (x.\_1 + y.\_1, x.\_2 + y.\_2))
* val calcAvg = studReduce.mapValues { case (sum, count) => (1.0 \* sum) / count }

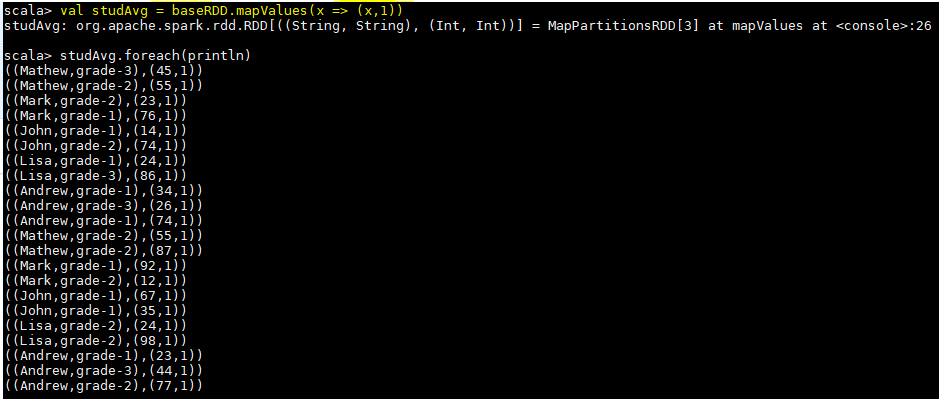
First we are creating the baseRDD to read the file and selecting name and grade as key and marks as value:-



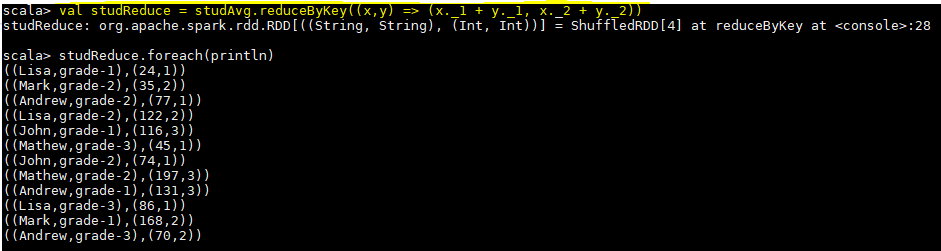
Below screenshot shows the result for above RDD:-



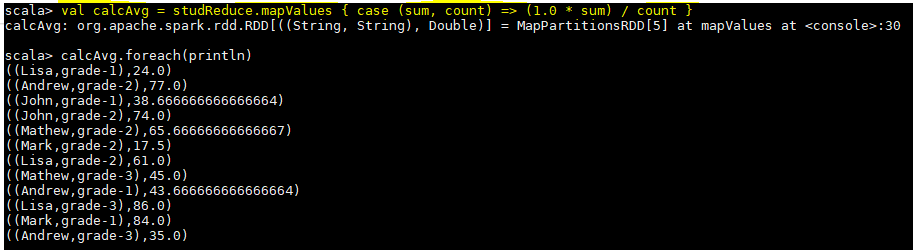
Now we are mapping the values of above paired RDD with 1 using mapValues function:-



Here we are using reduceByKey to add the occurrences of marks for each key which is student name and grade:-



Now we are calculating average by summing the marks and dividing by its count for each key. Below screenshot shows the final result:-

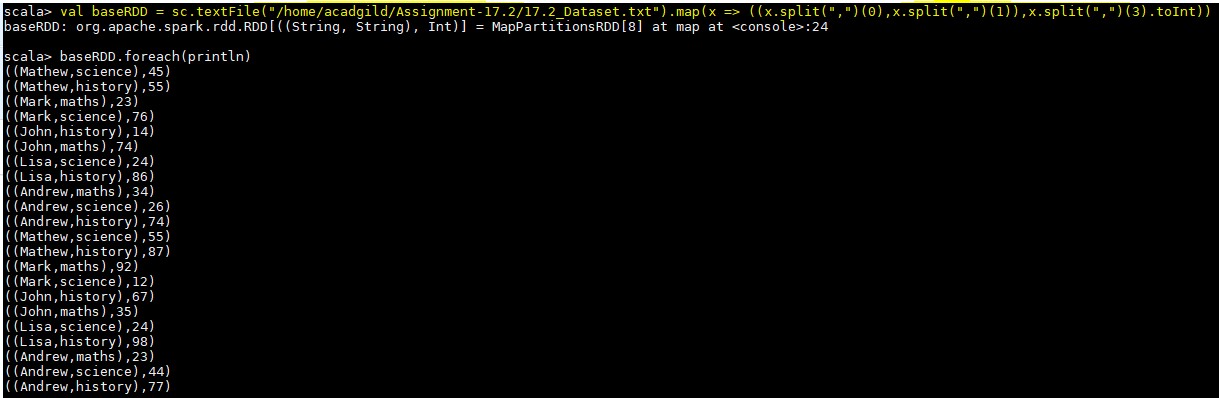


# The average score of students in each subject across all grades

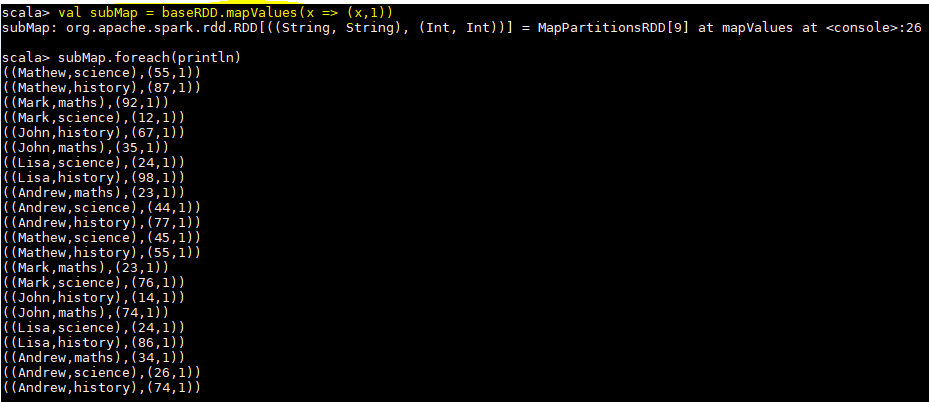
Below is the code used to find the result:-

* val baseRDD = sc.textFile("/home/acadgild/Assignment- 17.2/17.2\_Dataset.txt").map(x => ((x.split(",")(0),x.split(",")(1)),x.split(",")(3).toInt))
* val subMap = baseRDD.mapValues(x => (x,1))
* val subReduce = subMap.reduceByKey((x,y) => (x.\_1 + y.\_1, x.\_2 + y.\_2))
* val subAvg = subReduce.mapValues { case (sum, count) => (1.0 \* sum) / count }

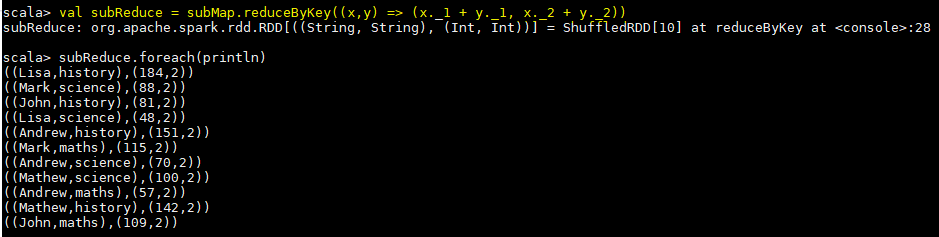
We are first creating baseRDD to read the text file and we are extracting name and subject as key and marks as value:-



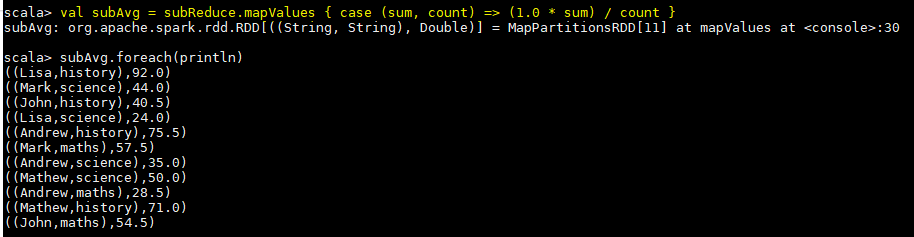
Now using mapValues we are mapping each value with 1:-



Now we are adding the marks and number of occurrences for each key using reducebyKey:-



In below step we are calculating average by dividing the sum of marks and count of occurrences for each key:-



# The average score of students in each subject per grade

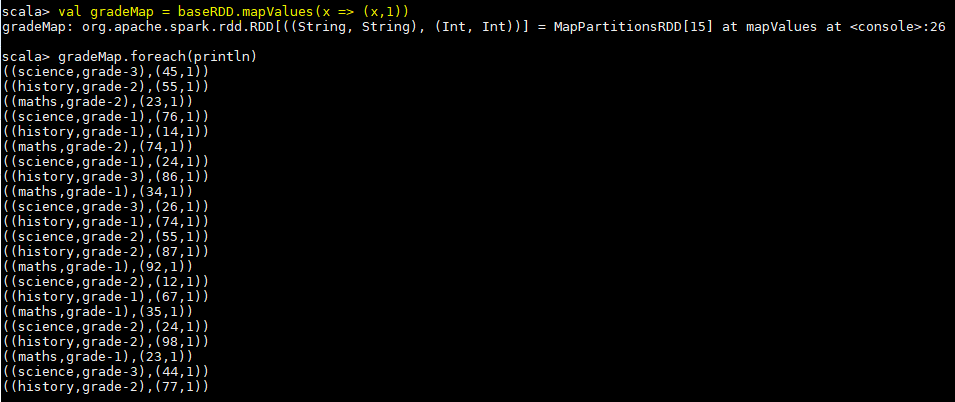
Below is the code used to find the result:-

* val baseRDD = sc.textFile("/home/acadgild/Assignment- 17.2/17.2\_Dataset.txt").map(x => ((x.split(",")(1),x.split(",")(2)),x.split(",")(3).toInt))
* val gradeMap = baseRDD.mapValues(x => (x,1))
* val gradeReduce = gradeMap.reduceByKey((x,y) => (x.\_1 + y.\_1, x.\_2 + y.\_2))
* val gradeAvg = gradeReduce.mapValues { case (sum,count) => (1.0 \* sum) / count }

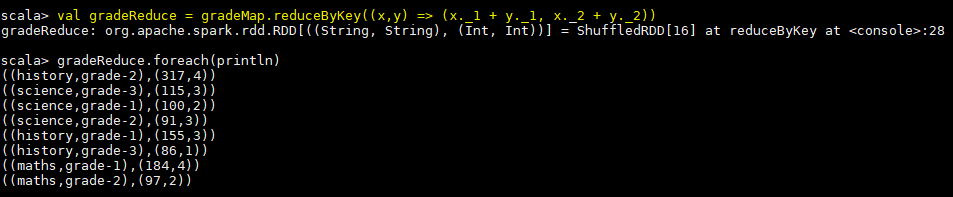
In first step we are creating paired RDD named as baseRDD to read the file and extracting subject and grade as key and marks as value:-



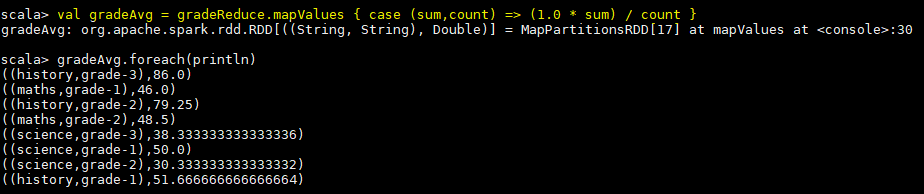
Then we are mapping the values of baseRDD with 1using function mapValues:-



Now we are adding marks and number of occurrences for each key using reduceByKey:-



Then we are calculating average by dividing the sum of marks with number of occurrences:-



# For all students in grade-2, how many have average score greater than 50

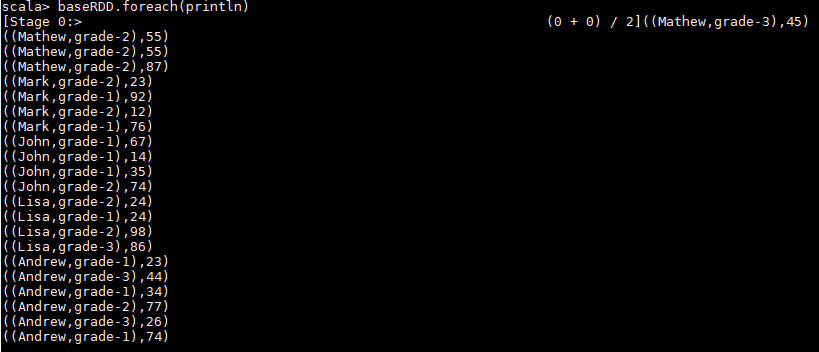
Below code has been used to find the required result:-

* val baseRDD = sc.textFile("/home/acadgild/Assignment- 17.2/17.2\_Dataset.txt").map(x => ((x.split(",")(0),x.split(",")(2)),x.split(",")(3).toInt))
* val studAvg = baseRDD.mapValues(x => (x,1))
* val studReduce = studAvg.reduceByKey((x,y) => (x.\_1 + y.\_1, x.\_2 + y.\_2))
* val calcAvg = studReduce.mapValues { case (sum, count) => (1.0 \* sum) / count }
* val filterGrade = calcAvg.filter(x => x.\_1.\_2 == "grade-2" && x.\_2 > 50)
* val countStud = filterGrade2.count()

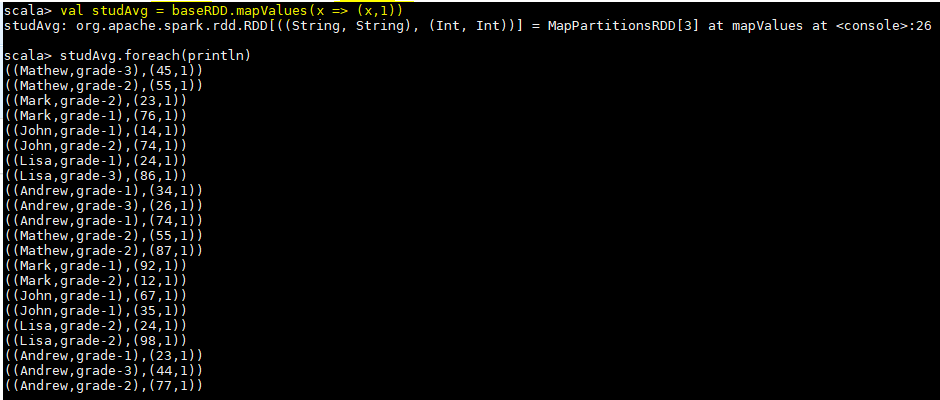
First we are creating a paired RDD named as baseRDD to read the file and extracting name and grade as key and marks as value:-



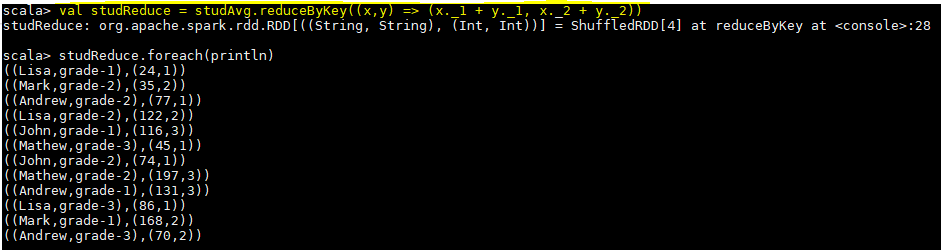
Below screenshot shows the result for above RDD:-



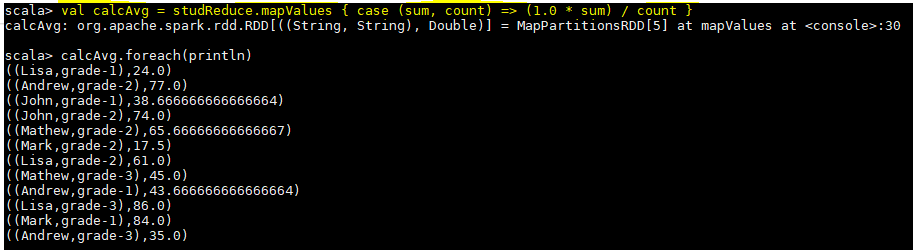
Now we are mapping each value of baseRDD with 1 as shown below:-



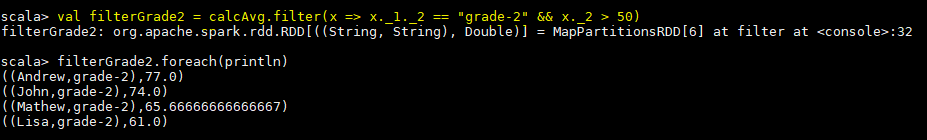
In below step we are adding the marks of subject and number of occurrences per key using reduceByKey function:-



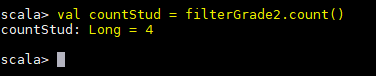
Here we are calculating the average of each student:-



Now in below step we are filtering the above result with student belonging to grade-2 and having marks greater than 50:-



Below screenshot shows the final result which is the count which is:-



Problem Statement 3:-

Are there any students in the college that satisfy the below criteria :

1. Average score per student\_name across all grades is same as average score per student\_name per grade

Solution:-

# Average score per student\_name across all grades is same as average score per student\_name per grade

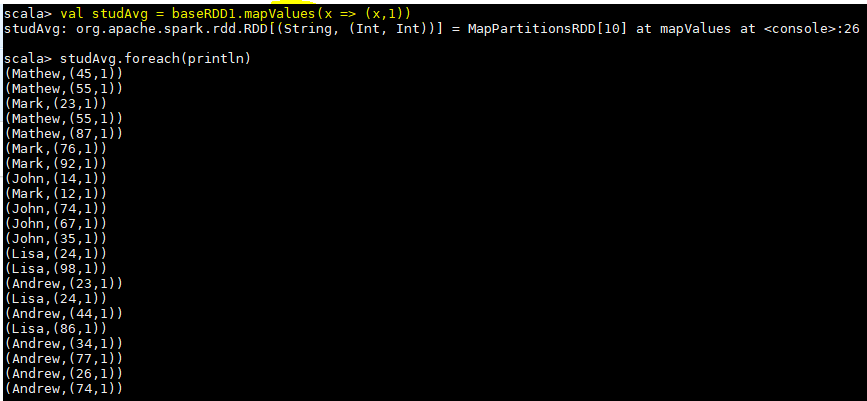
To find the solution of above problem we will first calculate average of each student across all grades i.e. irrespective of grade. Below is the code used to find the same:-

* val baseRDD1 = sc.textFile("/home/acadgild/Assignment- 17.2/17.2\_Dataset.txt").map(x => (x.split(",")(0),x.split(",")(3).toInt))
* val studAvg = baseRDD1.mapValues(x => (x,1))
* val studReduce = studAvg.reduceByKey((x,y) => (x.\_1 + y.\_1, x.\_2 + y.\_2))
* val nameAvg = studReduce.mapValues { case (sum, count) => (1.0 \* sum) / count }

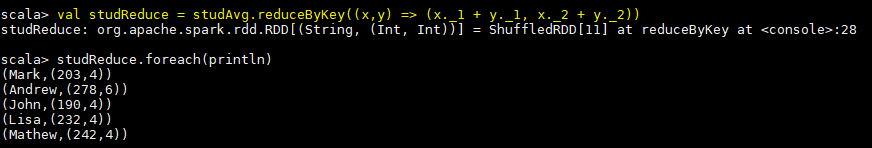
First we create a paired RDD named as baseRDD1 by extracting only name and marks:-



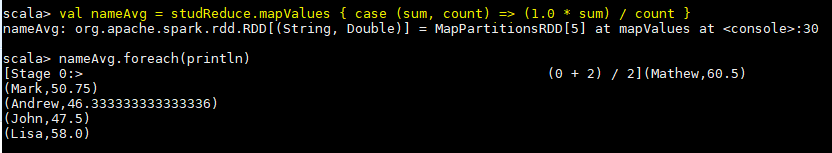
Then we are mapping each value of above RDD with 1:-



Then we are adding the marks and number of occurrences for each student using redducebyKey as shown below:-



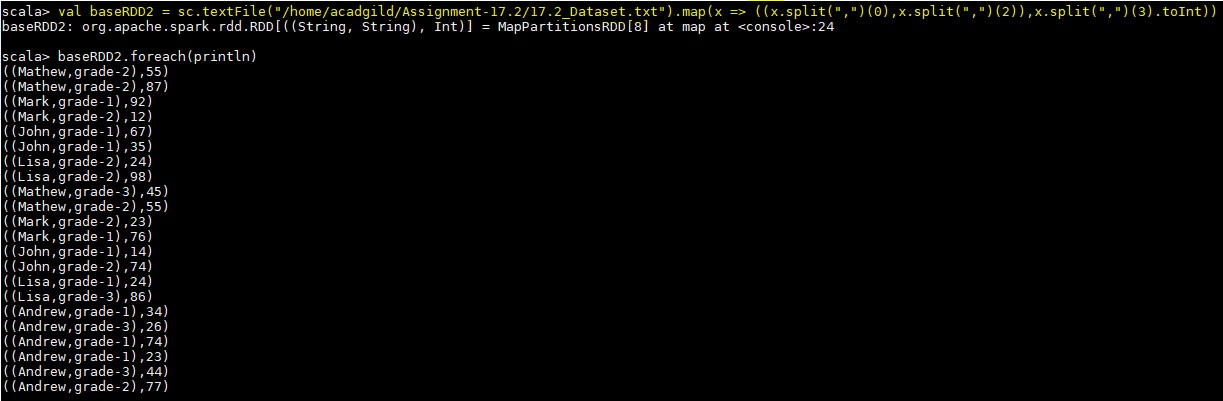
In below step we are calculating the average of each student:-



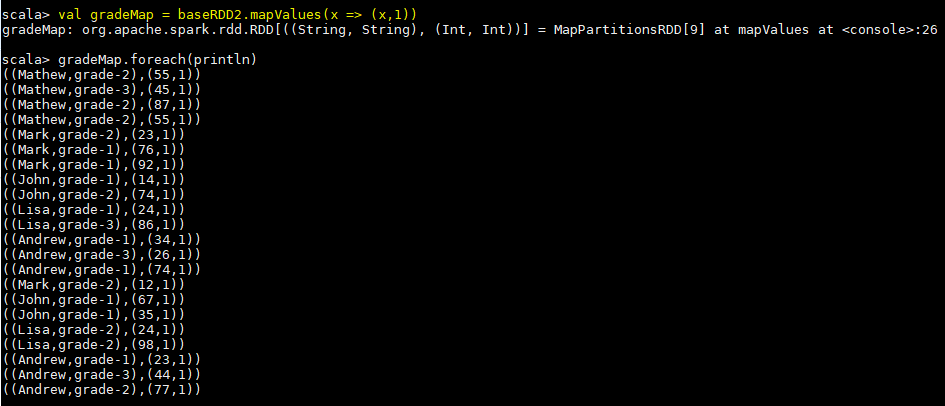
Now the second step of this problem is to find the average of each student per grade.We have used below code to find the same:-

* val baseRDD2 = sc.textFile("/home/acadgild/Assignment- 17.2/17.2\_Dataset.txt").map(x => ((x.split(",")(0),x.split(",")(2)),x.split(",")(3).toInt))
* val gradeMap = baseRDD2.mapValues(x => (x,1))
* val gradeReduce = gradeMap.reduceByKey((x,y) => (x.\_1 + y.\_1, x.\_2 + y.\_2))
* val gradeAvg = gradeReduce.mapValues { case (sum, count) => (1.0 \* sum) / count }

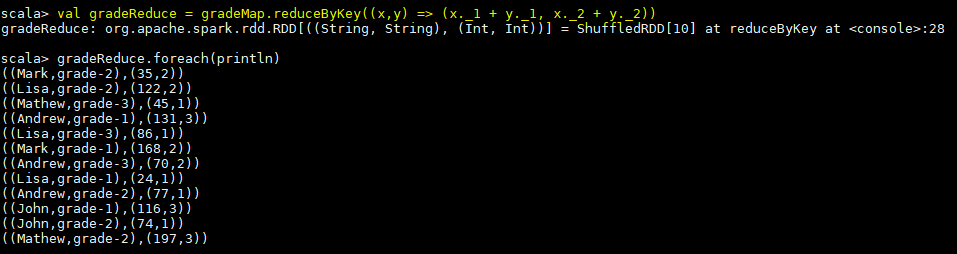
So first we are creating another paired RDD named as baseRDD2 by extracting name and grade as key and marks as value from the input file:-



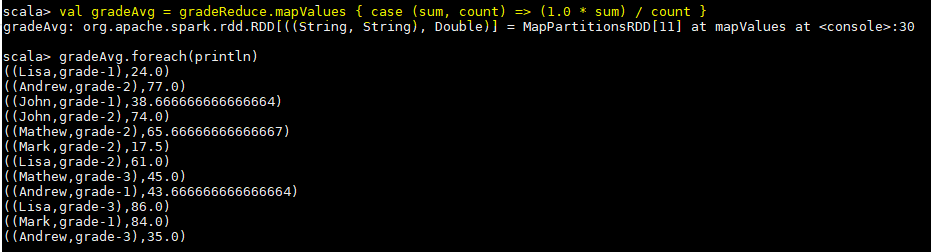
Then we are mapping each value of baseRDD2 with 1 using mapValues function:-



Then we are adding the marks and number of occurrences of 1 for each key using reduceByKey() function:-



In below step we are calculating average of each key by dividing the sum of marks with the count:-



Now to proceed further we are extracting name and marks from above RDD



In below step we are using intersection function between flatgradeAvg and flatnameAvg rdd’s to find whether any common student is there.So the command comman.foreach(println) shows that no common students are there having average score per student\_name across all grades is same as average score per student\_name per grade:-

