

## **CS5543 Real-Time Big Data Analytics**

### **LAB ASSIGNMENT #2**

**Latha Muddu**

#### **1) Question**

##### **Spark Programming:**

**Write a spark program with an interesting use case using text data as the input and program should have at least Two Spark Transformations and Two Spark Actions.**

##### **Transformations:**

###### **Map:**

The Map transformation in spark, the input is given by passing elements as a function where the map transformation returns a distributed dataset.

###### **SortByKey:**

The SortByKey transformation in spark where key value pairs are paired based on the order of K. The key value pairs could be sorted in ascending or descending order based on the requirement.

###### **ReduceByKey:**

In this transformation the values for each keys are reduced and returns corresponding key value pairs. The number of reduce function could be specified as per the requirement.

##### **Actions:**

###### **takeOrdered(n, [ordering]):**

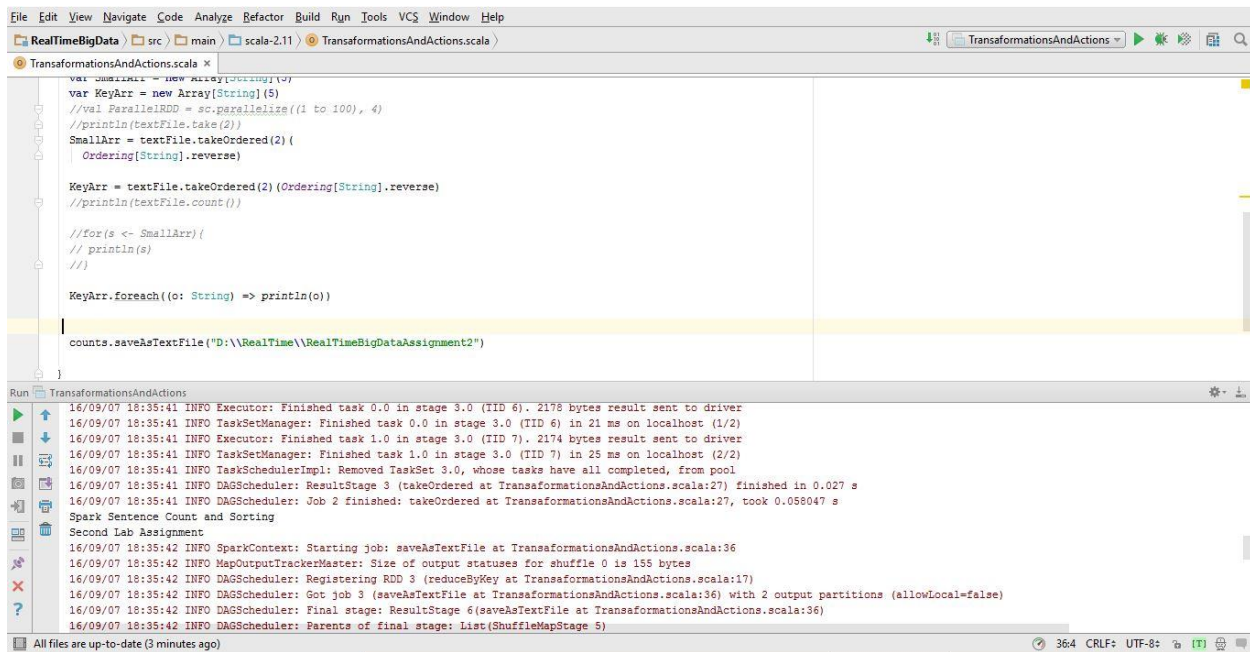
Returns the first n elements in a particular order as specified in the action.

###### **foreach(func):**

This will run a function on each element of the dataset. This action is usually done for interacting with the External Storage Systems.

## Screenshots:

The Actions used are takeOrder and foreach



The screenshot shows an IDE window with a Scala file named `TransaformationsAndActions.scala`. The code defines a `KeyArr` and a `SmallArr`, then uses `takeOrdered` and `foreach` to process them. The output console shows the execution of the program, including task completion messages and the final output of the `saveAsTextFile` action.

```
var SmallArr = new Array[Long](4)
var KeyArr = new Array[String](5)
//val ParallelRDD = sc.parallelize((1 to 100), 4)
//println(textFile.take(2))
SmallArr = textFile.takeOrdered(2)(
  Ordering[String].reverse)

KeyArr = textFile.takeOrdered(2)(Ordering[String].reverse)
//println(textFile.count())

//for(s <- SmallArr){
//  println(s)
//}

KeyArr.foreach((o: String) => println(o))

counts.saveAsTextFile("D:\\RealTime\\RealTimeBigDataAssignment2")
```

Run TransaformationsAndActions

```
16/09/07 18:35:41 INFO Executor: Finished task 0.0 in stage 3.0 (TID 6). 2178 bytes result sent to driver
16/09/07 18:35:41 INFO TaskSetManager: Finished task 0.0 in stage 3.0 (TID 6) in 21 ms on localhost (1/2)
16/09/07 18:35:41 INFO Executor: Finished task 1.0 in stage 3.0 (TID 7). 2174 bytes result sent to driver
16/09/07 18:35:41 INFO TaskSetManager: Finished task 1.0 in stage 3.0 (TID 7) in 25 ms on localhost (2/2)
16/09/07 18:35:41 INFO TaskSchedulerImpl: Removed TaskSet 3.0, whose tasks have all completed, from pool
16/09/07 18:35:41 INFO DAGScheduler: ResultStage 3 (takeOrdered at TransaformationsAndActions.scala:27) finished in 0.027 s
16/09/07 18:35:41 INFO DAGScheduler: Job 2 finished: takeOrdered at TransaformationsAndActions.scala:27, took 0.058047 s
Spark Sentence Count and Sorting
Second Lab Assignment
16/09/07 18:35:42 INFO SparkContext: Starting job: saveAsTextFile at TransaformationsAndActions.scala:36
16/09/07 18:35:42 INFO MapOutputTrackerMaster: Size of output statuses for shuffle 0 is 155 bytes
16/09/07 18:35:42 INFO DAGScheduler: Registering RDD 3 (reduceByKey at TransaformationsAndActions.scala:17)
16/09/07 18:35:42 INFO DAGScheduler: Got job 3 (saveAsTextFile at TransaformationsAndActions.scala:36) with 2 output partitions (allowLocal=false)
16/09/07 18:35:42 INFO DAGScheduler: Final stage: ResultStage 6 (saveAsTextFile at TransaformationsAndActions.scala:36)
16/09/07 18:35:42 INFO DAGScheduler: Parents of final stage: List(ShuffleMapStage 5)
```

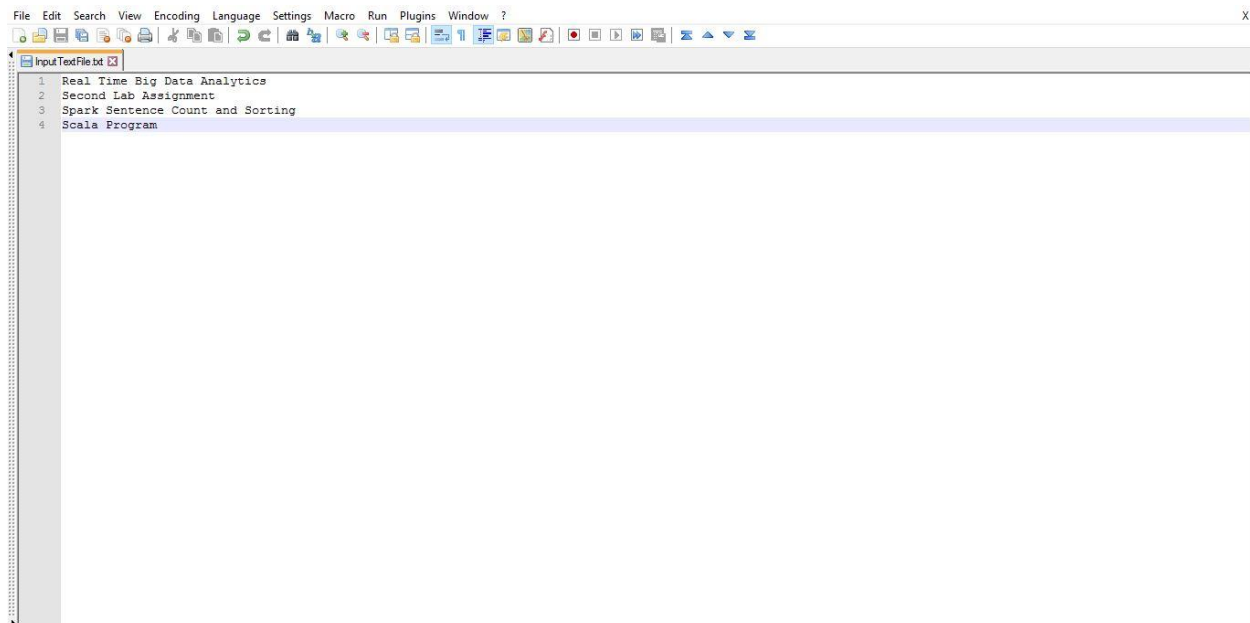
The input file given is

Real Time Big Data Analytics

Second Lab Assignment

Spark Sentence Count and Sorting

Scala Program



The screenshot shows a text file named `InputTextFile.txt` with the following content:

```
1 Real Time Big Data Analytics
2 Second Lab Assignment
3 Spark Sentence Count and Sorting
4 Scala Program
```

## Map Reduce Diagram:

