

Name: Renu Mariam Mathew

Student id: 16376

Hw# :6

Q1.Difference between group by and reduce by key

<code>groupByKey()</code>	<code>reduceByKey(func)</code>
Group values with the same key	Combine values with the same key
Eg. <i>Transformations on one pair RDD (example: {(1, 2), (3, 4), (3, 6)})</i>	<i>Transformations on one pair RDD (example: {(1, 2), (3, 4), (3, 6)})</i>
<code>rdd.groupByKey()</code>	<code>rdd.reduceByKey((x, y) => x + y)</code>
Output: <code>{(1, [2]), (3, [4, 6])}</code>	Output: <code>{(1, 2), (3, 10)}</code>
By applying this operation on the sample rdd it will return the new RDD which basically is made up with key and coalesces the value for one particular key.	To applying reduce by key transformation, the RDD on which reduce by key transformation will be applied need to have a key value pair. The output will be key and the sum of the values associated with that key.

Name: Renu Mariam Mathew

Student id: 16376

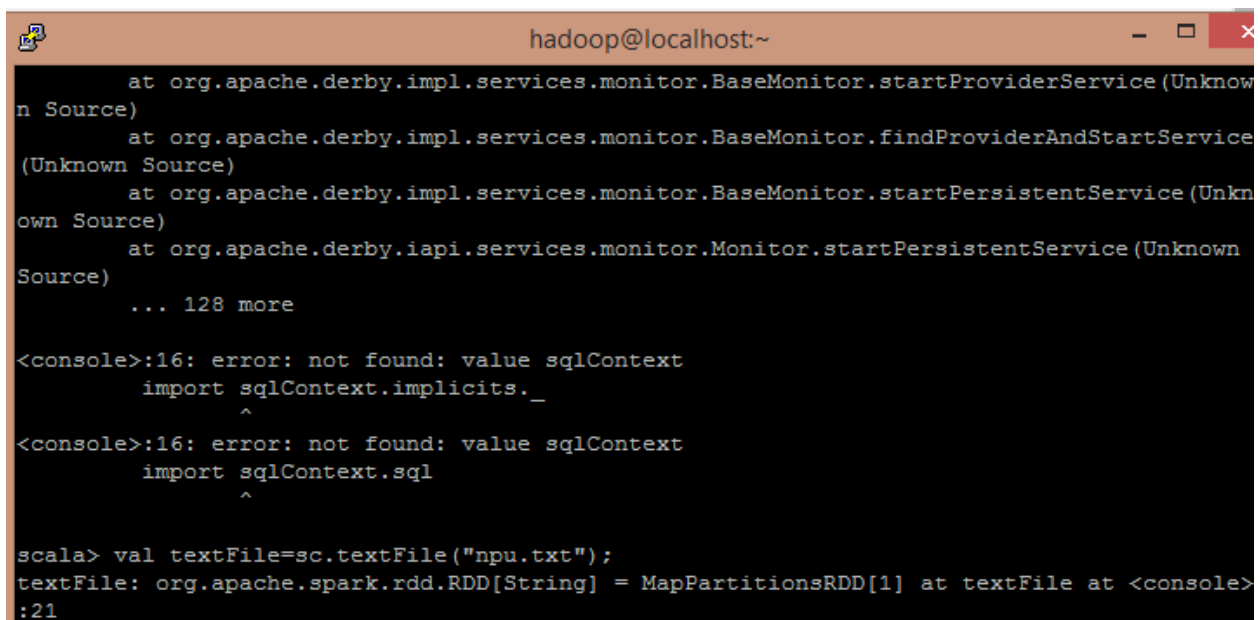
Hw# :6

groupByKey merely collects all values for a key together

reduceByKey could be used to sum values per key.

Q2. Spark wordcount on file in HDFS:

Step 1: `val textFile=sc.textfile("npu.txt");`

A screenshot of a terminal window with a title bar that reads 'hadoop@localhost:~'. The terminal shows several lines of Scala code and error messages. The first part shows a stack trace with 'at org.apache.derby.impl.services.monitor.BaseMonitor.startProviderService' and '... 128 more'. Then, there are two error messages: '<console>:16: error: not found: value sqlContext' followed by 'import sqlContext.implicitly._' with an arrow pointing to the underscore, and another similar error for 'sqlContext.sql'. Finally, the command 'scala> val textFile=sc.textFile("npu.txt");' is executed, resulting in 'textFile: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[1] at textFile at <console>:21'.

Step 2:

`Val words = textFile.flatMap(_.split("\\s+))`

Name: Renu Mariam Mathew

Student id: 16376

Hw# :6

Val wc= words.map(w ➔ (w,1)) .reduceByKey(_+)

Wc.saveAsTextFile("myWc.count")

Wc.collect

```
scala> val words = textFile.flatMap(_.split("\\s+"))
words: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[5] at flatMap at <console>:23

scala> val wc = words.map(w => (w,1)).reduceByKey(_+_ )
wc: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[7] at reduceByKey at <console>:25

scala> wc.saveAsTextFile("myWc.count")

scala> wc.collect
res3: Array[(String, Int)] = Array((science,,1), (university,2), (walking,1), (The,4), (classrooms;,1), (facilities.,1), (labs,,2), (sciences.,1), (library,1), (devoted,1), (Fremont,,2), (simulation,1), (housing,1), (house,1), (California,,1), (several,1), (Master's,1), (which,1), (offices;,1), (located,2), (recreation,1), (management.,1), (tend,1), (situated,1), (student,2), (buildings,1), (institution,1), (the,7), (Northwestern,1), (technology,2), (applied,1), (campus.,1), (companies.,1), (NPU,2), (be,1), (communication,1), (as,2), (corporations,1), (numerous,1), (area,1), (campus,2), (distance,1), (USA.,1), (is,3), (school,,1), (small,1), (Doctorate,1), (database,,1), (one,1), (higher,1), (schools,1), (midst,1), (Polytechnic,1), (center;,1), (States,1), (in,8), (founded,1), (awards,1...
```