Dattamsha

Big Data Made Easy

map vs flatMap in Spark

September 24, 2014 Big Data example, spark

In the previous blogs around Spark examples, RDD.flatMap() has been used. In this blog we will look at the differences between RDD.map() and RDD.flatMap().

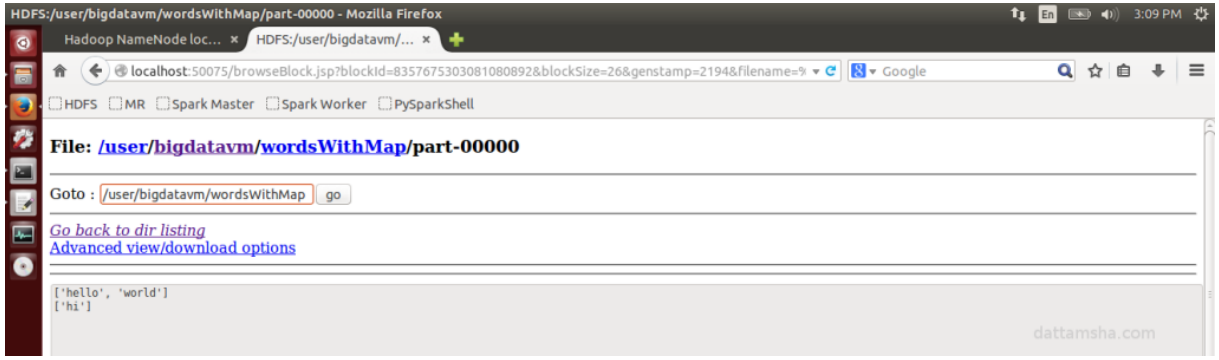
map and flatMap are similar, in the sense they take a line from the input RDD and apply a function on it. The way they differ is that the function in map returns only one element, while function in flatMap can return a list of elements (0 or more) as an iterator.

Also, the output of the flatMap is flattened. Although the function in flatMap returns a list of elements, the flatMap returns an RDD which has all the elements from the list in a flat way (not a list).

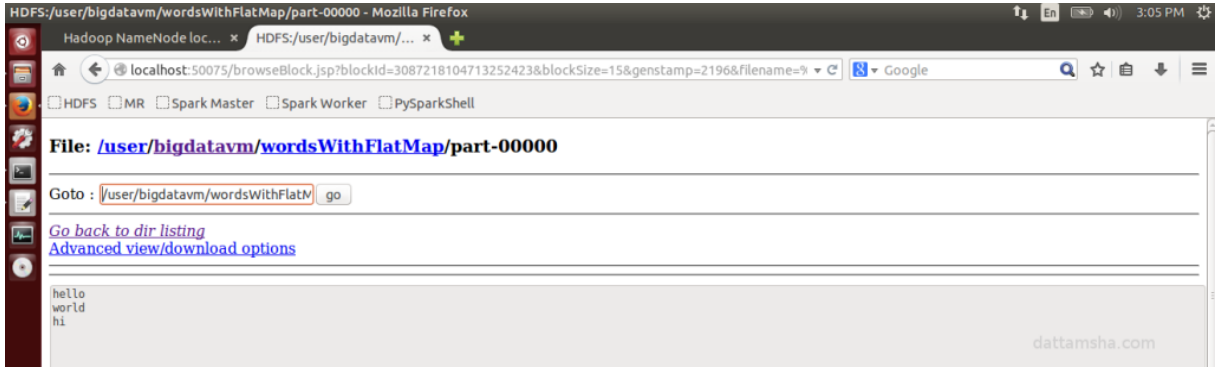
Sounds a bit confusing. In the below code snippet, on the input lines both map and flatMap are applied and output dumped in HDFS to wordsWithMap and wordsWithFlatMap folder.

```
1  from pyspark import SparkContext
2
3  sc = SparkContext("spark://bigdata-vm:7077", "Map")
4  lines = sc.parallelize(["hello world", "hi"])
5
6  wordsWithMap = lines.map(lambda line: line.split(" ")).coalesce(1)
7  wordsWithFlatMap = lines.flatMap(lambda line: line.split(" ")).coalesce(1)
8
9  wordsWithMap.saveAsTextFile("hdfs://localhost:9000/user/bigdatavm/wordsWithMap")
10 wordsWithFlatMap.saveAsTextFile("hdfs://localhost:9000/user/bigdatavm/wordsWithF
```

The output of the map function in HDFS



The output of the flatMap function in HDFS



Conclusion

The input function to map returns a single element, while the flatMap returns a list of elements (0 or more). And also, the output of the flatMap is flattened.

In the case of word count, where the input line is split into multiple words, flatMap can be used. Also, in the case of weather data set, the extractData nethod will validate the record and might or might not return a value. In this case also, flatMap can be used.

One thought on "map vs flatMap in Spark"

April 22, 2015 at 4:28 am

Thank you…very useful explanation

Rohan