Spark 分组 TOPN 排序

作者：hadoop
原文链接：https://ld246.com/article/1474537135143
来源网站：链滴

```
<pre class="lang:java decode:true">/**
 * Created by zhangshuai on 2016/9/22.
 */
//        输入文件
//        Spark 100
//        Hadoop 65
//        Spark 99
//        Hadoop 61
//        Spark 195
//        Hadoop 60
//        Spark 98
//        Hadoop 69
//        Spark 91
//        Hadoop 64
//        Spark 89
//        Hadoop 98
//        Spark 88
//        Hadoop 99
//        Spark 68
//        Hadoop 60
//        Spark 79
//        Hadoop 97
//        Spark 69
//        Hadoop 96



//        结果输出
//        Group key :Spark
//        195
//        100
//        99
//        98
//        91
//        **********
//        Group key :Hadoop
//        99
//        98
//        97
//        96
//        69
//        **********

import org.apache.spark.SparkConf;
import org.apache.spark.api.java.JavaPairRDD;
import org.apache.spark.api.java.JavaRDD;
import org.apache.spark.api.java.JavaSparkContext;
import org.apache.spark.api.java.function.PairFunction;
import org.apache.spark.api.java.function.VoidFunction;

import scala.Tuple2;

import java.util.Arrays;
import java.util.Iterator;
```

```
public class TopNGroupJava {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        SparkConf conf = new SparkConf().setAppName("TopNGroupJava")
                .setMaster("local");

        JavaSparkContext sc = new JavaSparkContext(conf); //其底层实际上就是Scala的SparkCont
xt
        JavaRDD&lt;String&gt; lines = sc.textFile(
                "E://topn.txt");

        JavaPairRDD&lt;String, Integer&gt; pairs = lines.mapToPair(new PairFunction&lt;String, S
ring, Integer&gt;() {
            private static final long serialVersionUID = 1L;

            @Override
            public Tuple2&lt;String, Integer&gt; call(String line)
                    throws Exception {
                // TODO Auto-generated method stub
                String[] splitedLine = line.split(" ");
                System.out.println(splitedLine[0]);

                return new Tuple2&lt;String, Integer&gt;(splitedLine[0],Integer.valueOf(splitedLine[1
));
            }
        });

        JavaPairRDD&lt;String, Iterable&lt;Integer&gt;&gt; groupedPairs = pairs.groupByKey();

        JavaPairRDD&lt;String, Iterable&lt;Integer&gt;&gt; top5 = groupedPairs.mapToPair(new
PairFunction&lt;Tuple2&lt;String, Iterable&lt;Integer&gt;&gt;, String, Iterable&lt;Integer&gt;
gt;() {
            /**
             *
             */
            private static final long serialVersionUID = 1L;

            @Override
            public Tuple2&lt;String, Iterable&lt;Integer&gt;&gt; call(Tuple2&lt;String, Iterable&lt;I
teger&gt;&gt; groupedData) throws Exception {
                Integer[] top5 = new Integer[5]; //保存top5本身
                String groupedKey = groupedData._1(); //获取分组组名
                Iterator&lt;Integer&gt; groupedValue = groupedData._2().iterator(); //获取每组的内
集合

                while (groupedValue.hasNext()) { //查看下一个元素，如果有继续循环
                    Integer value = groupedValue.next(); //获取当前循环的元素本身内容

                    for (int i = 0; i &lt; 5; i++) {
                        if (top5[i] == null) {
                            top5[i] = value;
```

```
                    break;
                } else if (value &gt; top5[i]) {
                    for (int j = 4; j &gt; i; j--) {
                        top5[j] = top5[j - 1];
                    }

                    top5[i] = value;

                    break;
                }
            }
        }

        return new Tuple2&lt;String, Iterable&lt;Integer&gt;&gt;(groupedKey,
                Arrays.asList(top5));
    }
});

top5.foreach(new VoidFunction&lt;Tuple2&lt;String, Iterable&lt;Integer&gt;&gt;&gt;() {
    @Override
    public void call(Tuple2&lt;String, Iterable&lt;Integer&gt;&gt; topped)
            throws Exception {
        System.out.println("Group key :" + topped._1());

        Iterator&lt;Integer&gt; toppedValue = topped._2().iterator();

        while (toppedValue.hasNext()) {
            Integer value = toppedValue.next();
            System.out.println(value);
        }

        System.out.println("**********************");
    }
});
  }
}
</pre>

<pre class="lang:scala decode:true ">import org.apache.spark.{SparkContext, SparkConf}

/**
 * Created by zhangshuai on 2016/9/22.
 */
//输入文件
//Spark,100
//Hadoop,62
//Flink,77
//Kafka,91
//Hadoop,93
//Spark,78
//Hadoop,69
//Spark,98
//Hadoop,62
//Spark,99
```

```
//Hadoop,61
//Spark,70
//Hadoop,75
//Spark,88
//Hadoop,68
//Spark,90
//Hadoop,61



//结果输出
//Flink:
//77
//Hadoop:
//61
//61
//62
//62
//68
//Kafka:
//91
//Spark:
//70
//78
//88
//90
//98
object TopNGroupScala {
  def main(args: Array[String]) {
    val conf=new SparkConf().setAppName("TopNGroupScala").setMaster("local")

    val sc=new SparkContext(conf)
    sc.setLogLevel("WARN")
    val lines=sc.textFile("E://topn.txt",1)

    val pairs=lines.map{(line =&gt;(line.split(",")(0),line.split(",")(1).toInt))}

    val grouped=pairs.groupByKey()

    val groupedTop5=grouped.map(grouped =&gt;
     {
       (grouped._1,grouped._2.toList.sortWith(_&lt;_).take(5))
     }
    )
    val groupedKeySorted=groupedTop5.sortByKey()

    groupedKeySorted.collect().foreach(pair =&gt;
    {
      println(pair._1+":")
      pair._2.foreach{println}
    }
    )
   sc.stop()
```

```
  }
}
</pre>

<p>&amp;nbsp;</p>
```