Java 8的lambda使得编写Spark应用更简单

Spark的其中一个目标就是使得大数据应用程序的编写更简单。Spark的Scala和Python的API接口很简洁；但由于Java缺少函数表达式（function expressions），使得Java API有些冗长。Java 8里面增加了lambda表达式，Spark开发者们更新了Spark的API来支持Java8的lambda表达式，而且与旧版本的Java保持兼容。这些支持将会在Spark 1.0可用。

来看看下面几个例子：这些例子将展示用Java 8编写Spark程序将更加简明。在Java 7，我们可能是这样写：

```
JavaRDD<String> lines = sc.textFile("hdfs://log.txt").filter(
  new Function<String, Boolean>() {
    public Boolean call(String s) {
      return s.contains("error");
    }
});

long numErrors = lines.count();
```

但是在Java 8我们可以这样写：

```
/**
 * User: 过往记忆
 * Date: 14-7-09
 * Time: 下午23:46
 * bolg:
 * 本文地址：/archives/1065
```

```
 * 过往记忆博客，专注于hadoop、hive、spark、shark、flume的技术博客，大量的干货
 * 过往记忆博客微信公共帐号：iteblog_hadoop
 */

JavaRDD<String> lines = sc.textFile("hdfs://log.txt")
                        .filter(s -> s.contains("error"));
long numErrors = lines.count();
```

当代码更长时，对比更明显。比如读取一个文件，得出其中的单词数。在Java 7中，实现代码如下：

```
JavaRDD<String> lines = sc.textFile("hdfs://log.txt");

// Map each line to multiple words
JavaRDD<String> words = lines.flatMap(
  new FlatMapFunction<String, String>() {
    public Iterable<String> call(String line) {
      return Arrays.asList(line.split(" "));
    }
});

// Turn the words into (word, 1) pairs
JavaPairRDD<String, Integer> ones = words.mapToPair(
  new PairFunction<String, String, Integer>() {
    public Tuple2<String, Integer> call(String w) {
      return new Tuple2<String, Integer>(w, 1);
    }
});

// Group up and add the pairs by key to produce counts
JavaPairRDD<String, Integer> counts = ones.reduceByKey(
  new Function2<Integer, Integer, Integer>() {
    public Integer call(Integer i1, Integer i2) {
      return i1 + i2;
    }
});

counts.saveAsTextFile("hdfs://counts.txt");
```

但是在Java 8，我们可以这样写：

```
/**
 * User: 过往记忆
 * Date: 14-7-09
 * Time: 下午23:46
 * bolg:
 * 本文地址：/archives/1065
 * 过往记忆博客，专注于hadoop、hive、spark、shark、flume的技术博客，大量的干货
 * 过往记忆博客微信公共帐号：iteblog_hadoop
 */

JavaRDD<String> lines = sc.textFile("hdfs://log.txt");
JavaRDD<String> words =
    lines.flatMap(line -> Arrays.asList(line.split(" ")));
JavaPairRDD<String, Integer> counts =
    words.mapToPair(w -> new Tuple2<String, Integer>(w, 1))
        .reduceByKey((x, y) -> x + y);
counts.saveAsTextFile("hdfs://counts.txt");
```

从上面的几个例子可以看出，Java 8的lambda表达式确实比之前版本更加简洁。支持Java 8的lambda将会在Spark 1.0版本提供支持。