7.6 输出操作

在Spark应用中，外部系统经常需要使用到Spark DStream处理后的数据，因此，需要采用输出操作把DStream的数据输出到数据库或者文件系统中

7.6.1 把DStream输出到文本文件中
7.6.2 把DStream写入到MySQL数据库中

7.6.1 把DStream输出到文本文件中



请在NetworkWordCountStatefulText.py代码文件中输入以下内容：

```
# coding:utf8

from __future__ import print_function
import sys
from pyspark import SparkContext
from pyspark.streaming import StreamingContext

if __name__ == "__main__":
    sc = SparkContext(appName="PythonStreamingStatefulNetworkWordCount")
    ssc = StreamingContext(sc, 1)
    ssc.checkpoint("file:///usr/local/spark/mycode/streaming/stateful/")
    # RDD with initial state (key, value) pairs
    initialStateRDD = sc.parallelize([(u'hello', 1), (u'world', 1)])
    def updateFunc(new_values, last_sum):
        return sum(new_values) + (last_sum or 0)
    lines = ssc.socketTextStream("hadoop01", 9999)
    running_counts = lines.flatMap(lambda line: line.split(" "))₩
                          .map(lambda word: (word, 1))₩
                          .updateStateByKey(updateFunc, initialRDD=initialStateRDD)
    running_counts.saveAsTextFiles(₩
        "file:///usr/local/spark/mycode/streaming/stateful/output")
    running_counts.pprint()
    ssc.start()
    ssc.awaitTermination()
```

7.6.1 把DStream输出到文本文件中



按照7.5.2节中的方法运行NetworkWordCountStatefulText.py，就可以把词频统计结果写入到“/usr/local/spark/mycode/streaming/stateful”目录中，可以发现，在这个目录下，生成了很多目录，结果如下：

```
output-1694936607000
output-1694936622000
output-1694936637000
output-1694936743000
……
```

7.6.2 把DStream写入到关系数据库中



启动MySQL数据库，并完成数据库和表的创建：

```
$ service mysql start
$ mysql -u root -p
$ #屏幕会提示你输入密码
```

在此前已经创建好的“spark”数据库中创建一个名称为“wordcount”的表：

```
mysql> use spark
mysql> create table wordcount (word char(20), count int(4));
```

7.6.2 把DStream写入到关系数据库中



由于需要让Python连接数据库MySQL，因此，需要首先安装Python连接MySQL的模块PyMySQL，在Linux终端中执行如下命令（如果以前已经安装，就不用重复执行下面命令）：

```
$ conda activate pyspark
$ pip install pymysql
```

7.6.2 把DStream写入到关系数据库中



NetworkWordCountStatefulDB.py

```
# coding:utf8

from __future__ import print_function
import sys
import pymysql
from pyspark import SparkContext
from pyspark.streaming import StreamingContext

if __name__ == "__main__":
    sc = SparkContext(appName="PythonStreamingStatefulNetworkWordCount")
    ssc = StreamingContext(sc, 1)
    ssc.checkpoint("file:///usr/local/spark/mycode/streaming/stateful")
    # RDD with initial state (key, value) pairs
    initialStateRDD = sc.parallelize([(u'hello', 1), (u'world', 1)])

    def updateFunc(new_values, last_sum):
        return sum(new_values) + (last_sum or 0)
```

7.6.2 把DStream写入到关系数据库中



```
lines = ssc.socketTextStream("hadoop01", 9999)
running_counts = lines.flatMap(lambda line: line.split(" "))₩
                      .map(lambda word: (word, 1))₩
                      .updateStateByKey(updateFunc, initialRDD=initialStateRDD)
running_counts.pprint()
```

7.6.2 把DStream写入到关系数据库中



```
    def dbfunc(records):
        host = "localhost"
        port = 3306
        user = "root"
        password = "123456"
        database = "spark"
        # 创建MySQL连接
        db = pymysql.connect(host=host, port=port, user=user, password=password, d
atabase=database)
        cursor = db.cursor()
        def doinsert(p):
            sql = "insert into wordcount(word,count) values ('%s', '%s')" % (str(p[0]), str
(p[1]))
            try:
                cursor.execute(sql)
                db.commit()
            except:
                db.rollback()
        for item in records:
            doinsert(item)
```

7.6.2 把DStream写入到关系数据库中



```
def func(rdd):
    repartitionedRDD = rdd.repartition(3)
    repartitionedRDD.foreachPartition(dbfunc)

running_counts.foreachRDD(func)
ssc.start()
ssc.awaitTermination()
```

7.7 本章小结



Spark Streaming是Spark生态系统中实现流计算功能的组件。本章介绍了Spark Streaming的设计原理，它把连续的数据流切分成多个分段，每个分段采用Spark引擎进行批处理，从而间接地实现了流处理的功能。由于Spark是基于内存的计算框架，因此，Spark Streaming具有较好的实时性。

本章还介绍了开发Spark Streaming程序的基本步骤，阐述了创建StreamingContext对象的方法。同时，分别以文件流、套接字流和RDD队列流等作为基本数据源，详细描述了流计算程序的编写方法。Spark Streaming还可以与Kafka、Flume等数据采集工具组合起来使用，由这些数据采集工具提供数据源。

DStream包含无状态转换操作和有状态转换操作，前者无法维护历史批次的状态信息，而后者可以在跨批次数据之间维护历史状态信息。

本章最后介绍了如何把DStream数据输出到文本文件和关系数据库中。