Фильтрация пустых строчек

Берем код примера с лекции:

```
from pyspark import SparkContext
from pyspark.streaming import StreamingContext

sc = SparkContext(master='local[1]')
ssc = StreamingContext(sc, batchDuration=10)

dstream = ssc.socketTextStream(hostname='localhost', port=9999)

result = dstream.filter(bool).count()

result.pprint()

ssc.start()
ssc.awaitTermination()
```

Требуется его запустить. Заслать немного информации через сокет и убедиться что он правильно считает строчки в каждом батче.

- Вход: строчки засланные через сокет
- Выход: число непустых строк в каждом батче

Для того, чтобы отправить информацию, можете использовать следующую команду:

```
nc -lk 127.0.0.1 -p {user_port}
```

Фильтрация пустых строчек с hdfs

Задача аналогична предыдущей только надо переопределить получение входных данных. Требуется получать текстовые файлы с HDFS.

На основе этих файлов можно породить список RDD и с помощью функции queueStream подать его на вход в spark Streaming.

- Вход: hdfs:///data/griboedov/ (один файл - один батч)
- Выход: число непустых строк в каждом батче

Пример эталонного решения:

```
def ls(directory):
    hadoop = sc._jvm.org.apache.hadoop

    fs = hadoop.fs.FileSystem
    conf = hadoop.conf.Configuration()
    path = hadoop.fs.Path(directory)

    return [f.getPath() for f in fs.get(conf).listStatus(path)]
ls_result = ls('hdfs:///data/griboedov/')

batches = [sc.textFile(f) for f in map(str, ls_result)]
dstream = ssc.queueStream(rdds=batches)

result = dstream.filter(bool).count()

result.pprint()

ssc.start()
ssc.awaitTermination()
```

Word Count

На основе данных из предыдущей задачи посчитать частоту слов в каждом батче.

- Вход: файлы из hdfs:///data/griboedov/
- Выход: 10 самых популярных слов вместе с их частотой для каждого батча

Подсказка: для вывода результата вам наверняка потребуется функция foreachRDD.

Stateful Word Count

На основе предыдущей задачи требуется преобразовать stateless wordcount в stateful с помощью функции updateStateByKey. Вывод результата происходит после обработки каждого батча. Результат содержит статистику для всех предыдущих батчей.

- Вход: файлы из hdfs:///data/griboedov/
- Выход: 10 самых популярных слов вместе с их частотой, посчитанный на основе всех предыдущих батчей

Пример stateful wordcount

```
def update_func(new_values, old_value):
 return (old_value or 0) + sum(new_values)

dstream \
 .flatMap(lambda row: row.split()) \
 .map(lambda word: (word, 1)) \
 .updateStateByKey(update_func)
```

Дополнительные примеры

официальная репа spark:
https://github.com/apache/spark/tree/master/examples/src/main/python/streaming