Adam Reilly
W205
Lab 10

SUBMISSION 1

```
from pyspark import SparkContext
from pyspark.streaming import StreamingContext
ssc = StreamingContext(sc, 1)
lines= ssc.textFileStream("file:///tmp/datastreams")
uclines = lines.map(lambda word: word.upper())
overfive = uclines.filter(lambda word: len(word) > 5)
overfive.pprint()
ssc.start()
```

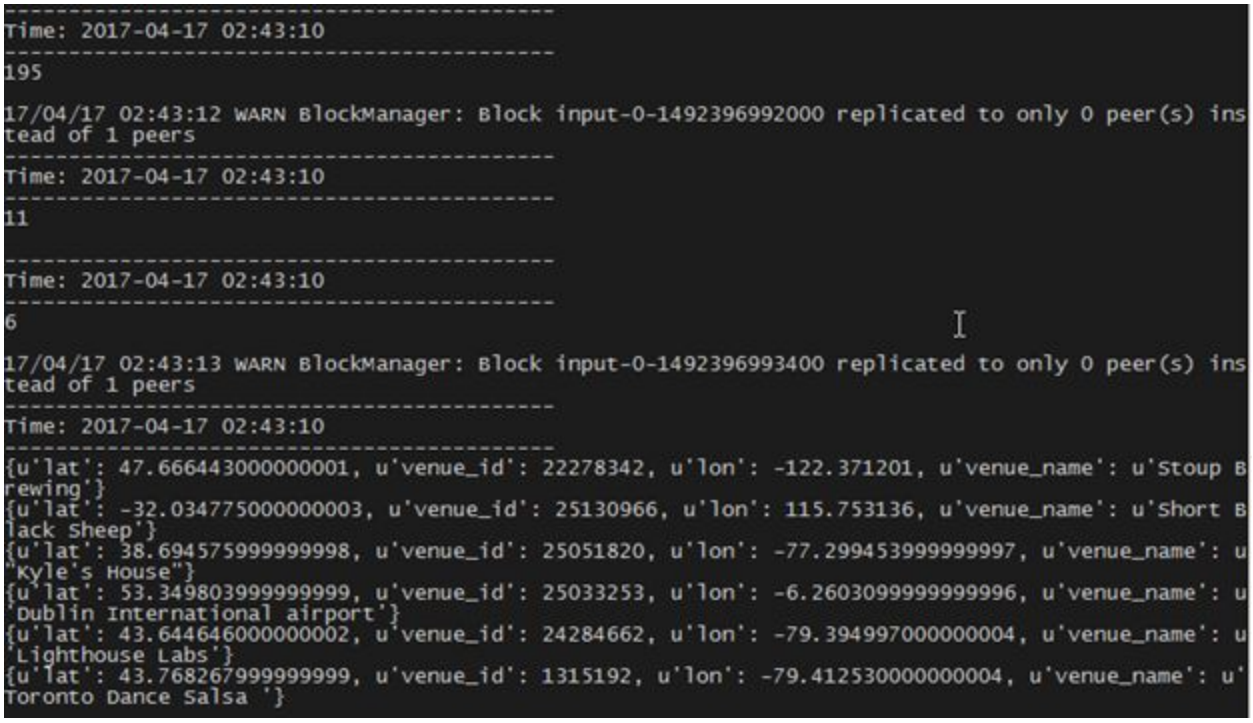
SUBMISSION 2.

```
$MASTER=local[2] pyspark
from pyspark import SparkContext
from pyspark.streaming import StreamingContext
import json
ssc = StreamingContext(sc, 10)
lines = ssc.textFileStream("file:///tmp/datastreams")
slines = lines.flatMap(lambda x: [ j['venue'] for j in json.loads('['+x+']') if 'venue' in j] )
cnt=slines.count()
cnt.pprint()
slines.pprint()
prefix= ("file:///tmp/venues.txt")
slines.saveAsTextFiles(prefix)
ssc.start()
```

SUBMISSION 3

1. In this scenario, I would increase the batch processing time so that it would be better able to fit the flow of data.
2. One side effect of needing to update a streaming program is that you likely won't be able to access the data that is being produced while the program updates. You could handle this by writing an identical program before updating and run the data through that program (and you could even write the data to an outside file in case you would like to reimport it back into the updated program).

SUBMISSION 4a



SUBMISSION 4B. The difference between a 10 sec batch with a 30 second sliding window and a 30 second batch is with the first option, the sorting is done in advance. Additionally, the sliding window can can compare more data because the 30 second batch can only cover items added since the last batch was run, while the window has more flexibility.