Pre-requisites

After installing Kafka and Zookeeper create a kafka topic using (to use this be in bin directory of kafka else use full path of to run the command

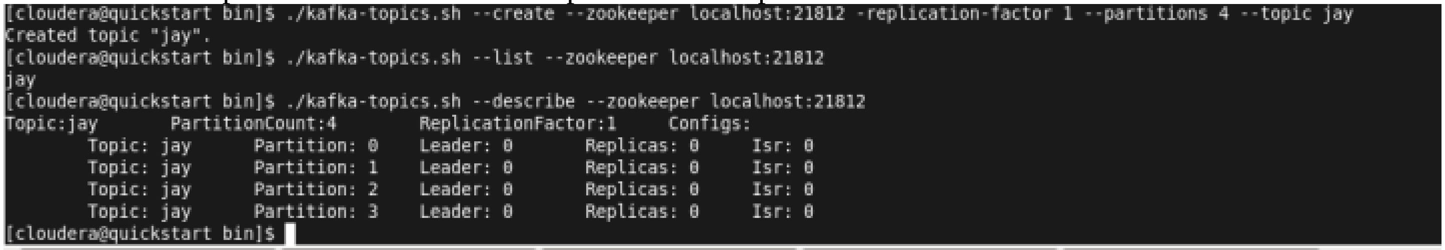
./kafka-topics.sh --create --zookeeper localhost:<port> -replication-factor --partitions 4 --topic <topic name>

and the check using:

./kafka-topics.sh --list --zookeeper localhost:<port>

and

./kafka-topics.sh --describe --zookeeper localhost:<port>



```
[cloudera@quickstart bin]$ ./kafka-topics.sh --create --zookeeper localhost:21812 -replication-factor 1 --partitions 4 --topic jay
Created topic "jay".
[cloudera@quickstart bin]$ ./kafka-topics.sh --list --zookeeper localhost:21812
jay
[cloudera@quickstart bin]$ ./kafka-topics.sh --describe --zookeeper localhost:21812
Topic:jay       PartitionCount:4        ReplicationFactor:1     Configs:
        Topic: jay      Partition: 0    Leader: 0       Replicas: 0     Isr: 0
        Topic: jay      Partition: 1    Leader: 0       Replicas: 0     Isr: 0
        Topic: jay      Partition: 2    Leader: 0       Replicas: 0     Isr: 0
        Topic: jay      Partition: 3    Leader: 0       Replicas: 0     Isr: 0
[cloudera@quickstart bin]$
```

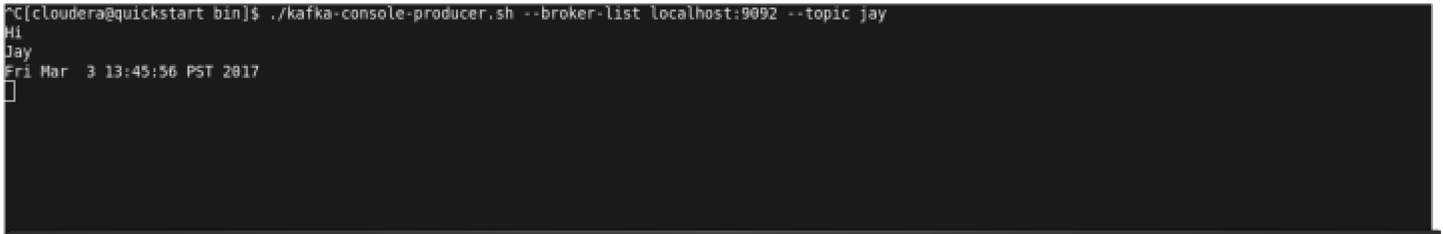
Run consumer and producer using commands :

./kafka-console-producer.sh --broker-list localhost:<port> --topic <topic name>

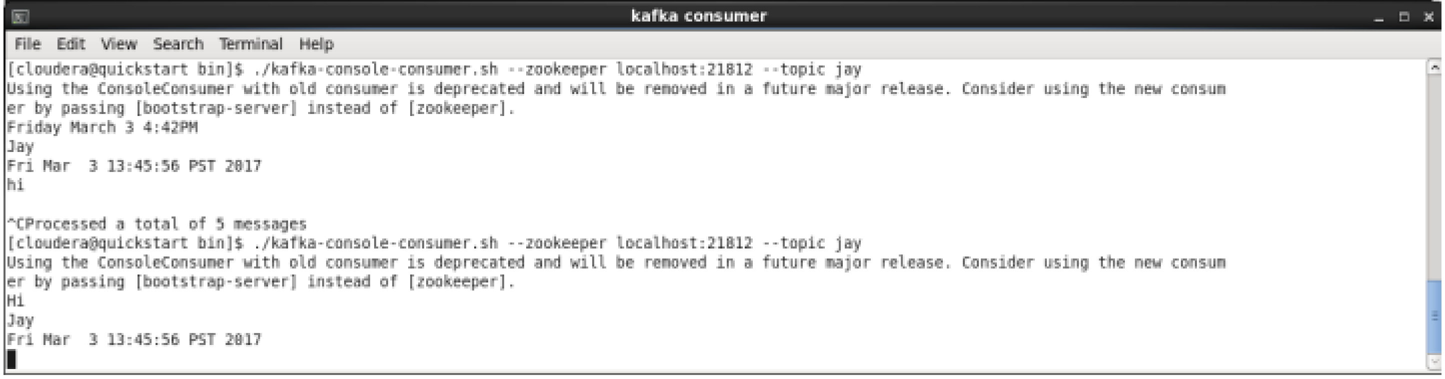
and

./kafka-console-consumer.sh --broker-list localhost:<port> --topic <topic name>

respectively



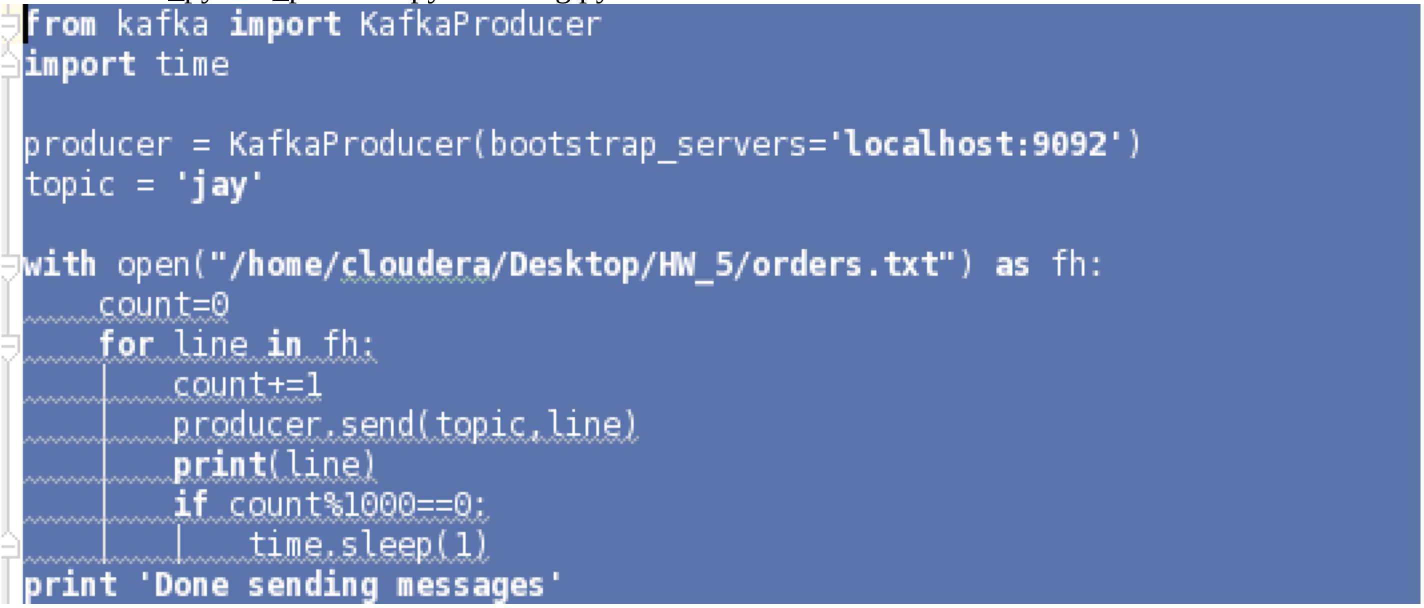
```
^C[cloudera@quickstart bin]$ ./kafka-console-producer.sh --broker-list localhost:9092 --topic jay
Hi
Jay
Fri Mar  3 13:45:56 PST 2017
```



```
[cloudera@quickstart bin]$ ./kafka-console-consumer.sh --zookeeper localhost:21812 --topic jay
Using the ConsoleConsumer with old consumer is deprecated and will be removed in a future major release. Consider using the new consum
er by passing [bootstrap-server] instead of [zookeeper].
Friday March 3 4:42PM
Jay
Fri Mar  3 13:45:56 PST 2017
hi

^CProcessed a total of 5 messages
[cloudera@quickstart bin]$ ./kafka-console-consumer.sh --zookeeper localhost:21812 --topic jay
Using the ConsoleConsumer with old consumer is deprecated and will be removed in a future major release. Consider using the new consum
er by passing [bootstrap-server] instead of [zookeeper].
Hi
Jay
Fri Mar  3 13:45:56 PST 2017
```

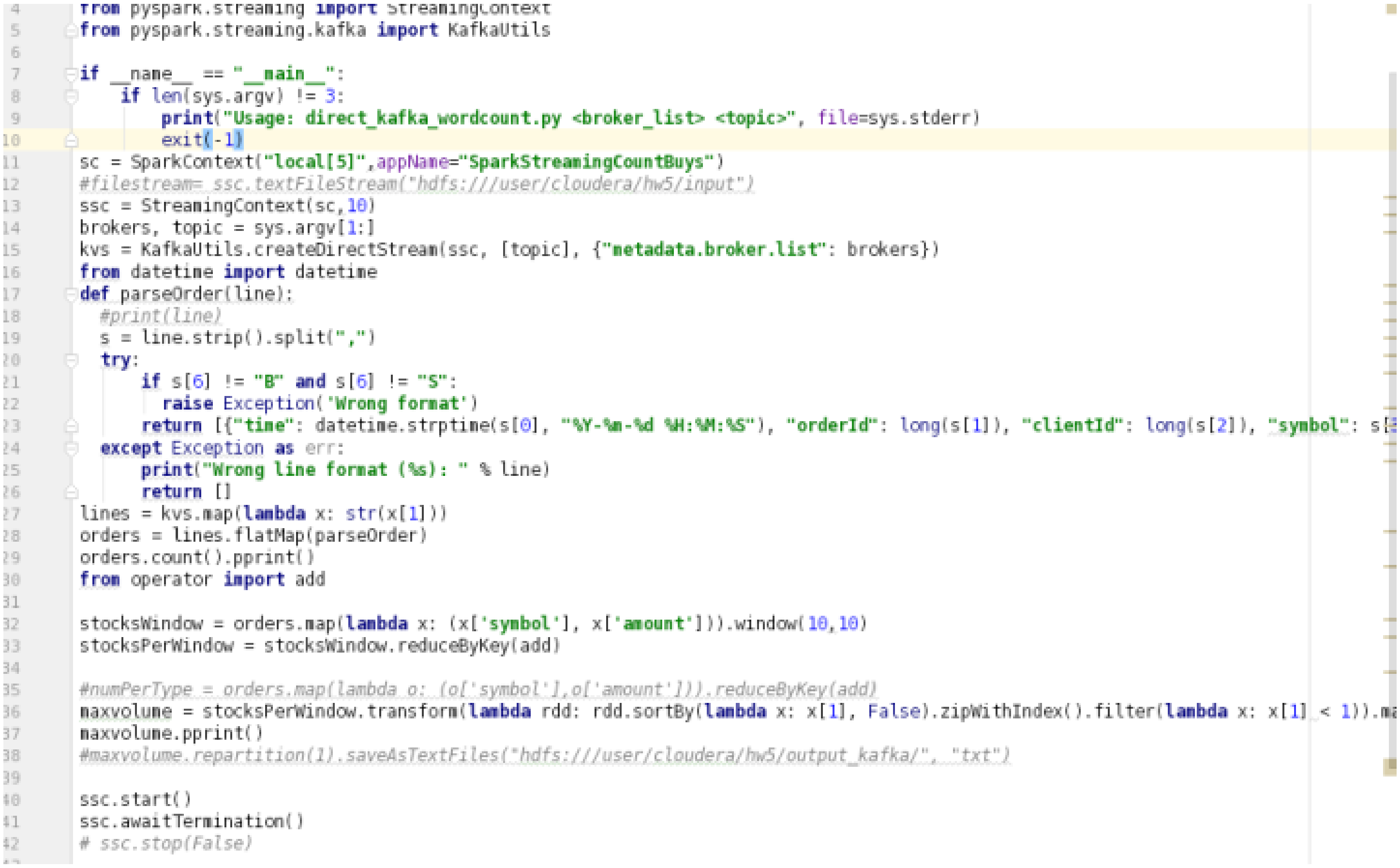
Start

1. Create a kafka producer to send 1000 messages and then wait 1 second code in kafka_python_producer.py run using python in 1 terminal



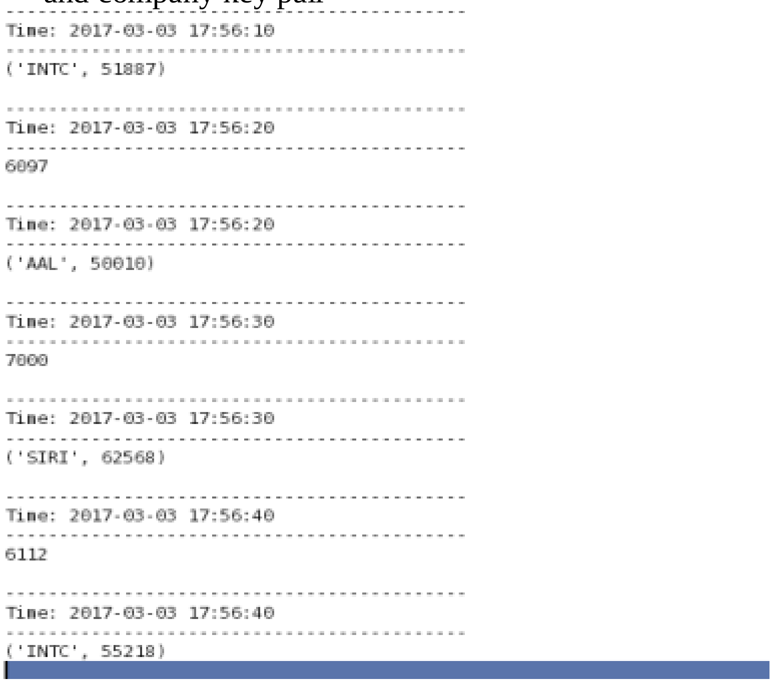
```
from kafka import KafkaProducer
import time

producer = KafkaProducer(bootstrap_servers='localhost:9092')
topic = 'jay'

with open("/home/cloudera/Desktop/HW_5/orders.txt") as fh:
    count=0
    for line in fh:
        count+=1
        producer.send(topic,line)
        print(line)
        if count%1000==0:
            time.sleep(1)
print 'Done sending messages'
```

2. In another terminal run Spark_streaming_consumer.py using spark-submit and parameters localhost:<port> and topic name respectively



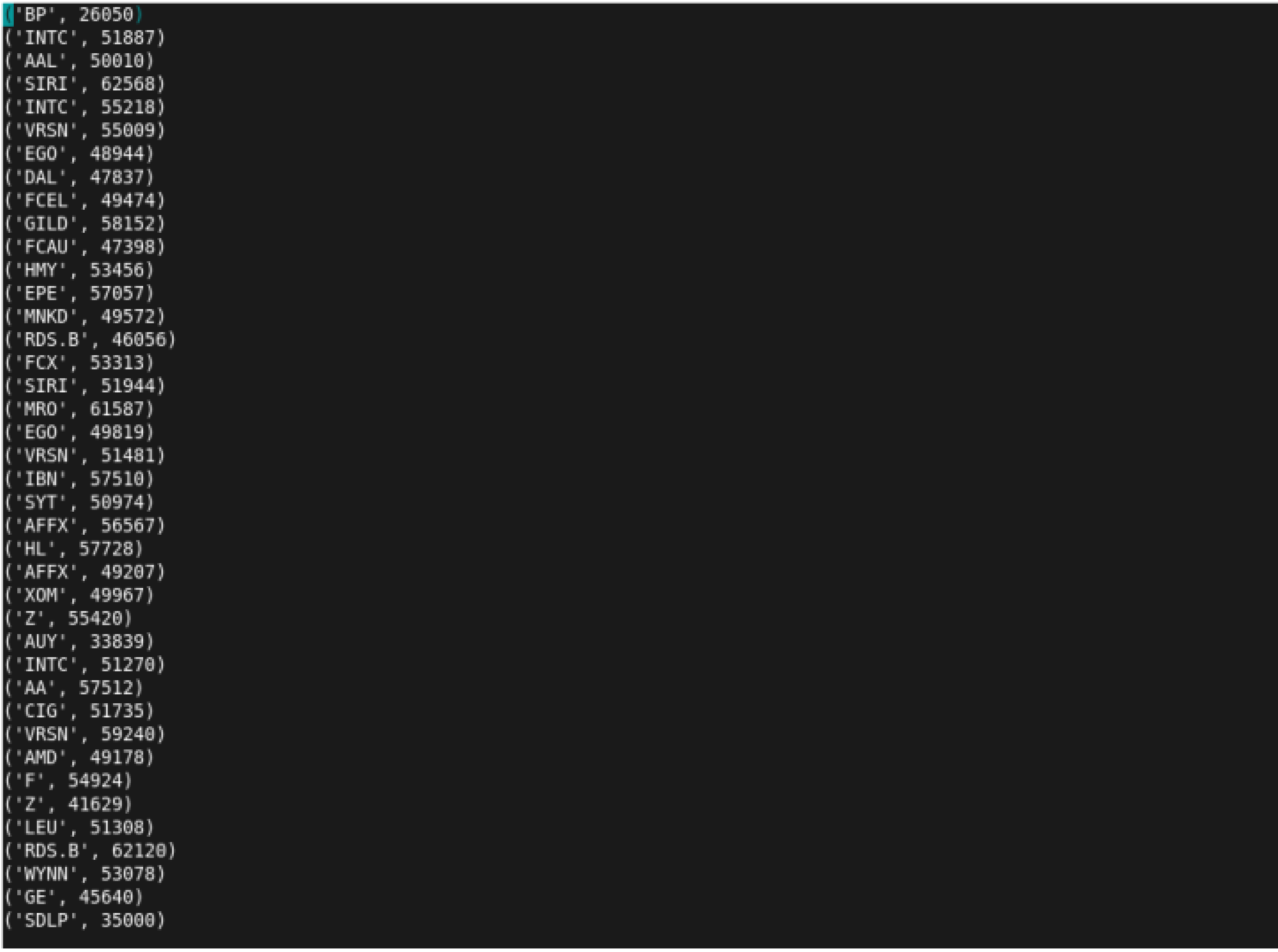
```
from pyspark.streaming import StreamingContext
from pyspark.streaming.kafka import KafkaUtils

if __name__ == "__main__":
    if len(sys.argv) != 3:
        print("Usage: direct_kafka_wordcount.py <broker_list> <topic>", file=sys.stderr)
        exit(-1)
sc = SparkContext("local[5]",appName="SparkStreamingCountBuys")
#filestream= ssc.textFileStream("hdfs:///user/cloudera/hw5/input")
ssc = StreamingContext(sc,10)
brokers, topic = sys.argv[1:]
kvs = KafkaUtils.createDirectStream(ssc, [topic], {"metadata.broker.list": brokers})
from datetime import datetime
def parseOrder(line):
  #print(line)
  s = line.strip().split(",")
  try:
      if s[6] != "B" and s[6] != "S":
        raise Exception('Wrong format')
      return [{"time": datetime.strptime(s[0], "%Y-%m-%d %H:%M:%S"), "orderId": long(s[1]), "clientId": long(s[2]), "symbol": s[
  except Exception as err:
      print("Wrong line format (%s): " % line)
      return []
lines = kvs.map(lambda x: str(x[1]))
orders = lines.flatMap(parseOrder)
orders.count().pprint()
from operator import add

stocksWindow = orders.map(lambda x: (x['symbol'], x['amount'])).window(10,10)
stocksPerWindow = stocksWindow.reduceByKey(add)

#numPerType = orders.map(lambda o: (o['symbol'],o['amount'])).reduceByKey(add)
maxvolume = stocksPerWindow.transform(lambda rdd: rdd.sortBy(lambda x: x[1], False).zipWithIndex().filter(lambda x: x[1] < 1)).ma
maxvolume.pprint()
#maxvolume.repartition(1).saveAsTextFiles("hdfs:///user/cloudera/hw5/output_kafka/", "txt")

ssc.start()
ssc.awaitTermination()
# ssc.stop(False)
```

3. However, using 10 window and 10 slide it was expected to take in 10000 order because of latency in kafka it sent in variable amount of data you can see the amount of data in the outputs above the max data and company key pair



```
-------------------------------------------
Time: 2017-03-03 17:56:10
-------------------------------------------
('INTC', 51887)

-------------------------------------------
Time: 2017-03-03 17:56:20
-------------------------------------------
6097

-------------------------------------------
Time: 2017-03-03 17:56:20
-------------------------------------------
('AAL', 50010)

-------------------------------------------
Time: 2017-03-03 17:56:30
-------------------------------------------
7000

-------------------------------------------
Time: 2017-03-03 17:56:30
-------------------------------------------
('SIRI', 62568)

-------------------------------------------
Time: 2017-03-03 17:56:40
-------------------------------------------
6112

-------------------------------------------
Time: 2017-03-03 17:56:40
-------------------------------------------
('INTC', 55218)
```

4. Due to this latency when I used the method similar to Problem 1 and took data I got different data as given in screen shot below



```
('BP', 26050)
('INTC', 51887)
('AAL', 50010)
('SIRI', 62568)
('INTC', 55218)
('VRSN', 55009)
('EGO', 48944)
('DAL', 47837)
('FCEL', 49474)
('GILD', 58152)
('FCAU', 47398)
('HMY', 53456)
('EPE', 57057)
('MNKD', 49572)
('RDS.B', 46056)
('FCX', 53313)
('SIRI', 51944)
('MRO', 61587)
('EGO', 49819)
('VRSN', 51481)
('IBN', 57510)
('SYT', 50974)
('AFFX', 56567)
('HL', 57728)
('AFFX', 49207)
('XOM', 49967)
('Z', 55420)
('AUY', 33839)
('INTC', 51270)
('AA', 57512)
('CIG', 51735)
('VRSN', 59240)
('AMD', 49178)
('F', 54924)
('Z', 41629)
('LEU', 51308)
('RDS.B', 62120)
('WYNN', 53078)
('GE', 45640)
('SDLP', 35000)
```