

IERG 4330/ ESTR4316 / IEMS 5730 Spring 2022

Homework 4

Release date: Mar 31, 2022

Due date: 23:59:00, Apr 19, 2022

The solution will be posted right after the deadline, so no late homework will be accepted!

Every Student **MUST** include the following statement, together with his/her signature in the submitted homework.

I declare that the assignment submitted on Blackboard system is original except for source material explicitly acknowledged, and that the same or related material has not been previously submitted for another course. I also acknowledge that I am aware of University policy and regulations on honesty in academic work, and of the disciplinary guidelines and procedures applicable to breaches of such policy and regulations, as contained in the website

<http://www.cuhk.edu.hk/policy/academichonesty/>.

Signed (Student *June*) Date: *19-4-22*

Name *Chen Kri Yin* SID *1155124983*

Submission notice:

- Submit your homework via the elearning system

General homework policies:

A student may discuss the problems with others. However, the work a student turns in must be created COMPLETELY by oneself ALONE. A student may not share ANY written work or pictures, nor may one copy answers from any source other than one's own brain.

Each student **MUST LIST** on the homework paper the **name of every person he/she has discussed or worked with**. If the answer includes content from any other source, the student **MUST STATE THE SOURCE**. Failure to do so is cheating and will result in sanctions. Copying answers from someone else is cheating even if one lists their name(s) on the homework.

If there is information you need to solve a problem but the information is not stated in the problem, try to find the data somewhere. If you cannot find it, state what data you need, make a reasonable estimate of its value and justify any assumptions you make. You will be graded not only on whether your answer is correct but also on whether you have done an intelligent analysis.

Q2:

i)

The Kafka producer:

```
File Edit View Help kafka_producer.py - ...Program\hw4
kafka_producer.py x
1 import os
2 import random
3 import time
4 from datetime import datetime
5
6 # modify to convert the ts to seconds
7 def convert_to_seconds(ts):
8     # print(ts)
9     date_time = datetime.strptime(ts, "%Y-%m-%d %H:%M:%S")
10    # print(date_time)
11
12    timedelta = date_time - datetime(1980, 1, 1)
13    seconds = timedelta.total_seconds()
14
15    # print(seconds)
16    return seconds
17
18
19 # modify this function, the sleep time should based on the time in the data
20 def random_sleep(ts):
21     # t = random.randint(1, 4) # modify this line
22     time.sleep(ts)
23
24
25 def main():
26     last_ts = None
27     with open('new_tweets.txt', 'rb') as f:
28         for line in f:
29
30             # split the text and timestamp
31             parts = line.rstrip().split(',')
32             text = ' '.join(parts[:-1])
33             ts = parts[-1]
34
35             ts = convert_to_seconds(ts)
36
37             cmd = 'echo "' + text + '" ./bin/kafka-console-producer.sh --broker-list localhost:9092 --topic bitcoin'
38
39             os.system(cmd)
40             # print(text)
41
42             if last_ts is not None:
43                 ts_delta = ts - last_ts
44                 # print(ts_delta)
45                 random_sleep(ts_delta)
46
47             last_ts = ts
48
49
50 if __name__ == '__main__':
51     main()
```

ii)

Output:

Showing 4096 bytes. Click [here](#) for full log

```
(u'#socialmedia', 2), (u'#xrp', 2), (u'#Earn0nLatoken', 1), (u'#linkedin', 1), (u'#government.', 1), (u'#medium', 1), (u'#gifts\u
----- 2022-04-18 17:27:00 -----
[(u'#TRON', 7), (u'#btc', 4), (u'#BitcoinTo', 4), (u'#cryptocurrency', 3), (u'#crypto', 3), (u'#twitter', 2), (u'#BTC', 2), (u'#N
----- 2022-04-18 17:29:00 -----
[(u'#TRON', 7), (u'#cryptocurrency', 3), (u'#Miami!Fabiano', 3), (u'#Bitcoin2021', 3), (u'#btc', 2), (u'#BTC', 2), (u'#NFT', 2),
----- 2022-04-18 17:31:00 -----
[(u'#BTC', 5), (u'#Miami!Fabiano', 5), (u'#Bitcoin2021', 5), (u'#BitcoinTo', 4), (u'#TRON', 3), (u'#NFT', 2), (u'#trading', 2), (
----- 2022-04-18 17:33:00 -----
[(u'#BitcoinTo', 6), (u'#BTC', 5), (u'#Miami!Fabiano', 4), (u'#Bitcoin2021', 4), (u'#BitcoinHoping', 3), (u'#LMCHB
----- 2022-04-18 17:35:00 -----
[(u'#BitcoinTo', 5), (u'#TRON', 5), (u'#BitcoinHoping', 4), (u'#BTC', 2), (u'#btc', 2), (u'#electroneum', 1), (u'#DCA', 1), (u'#C
----- 2022-04-18 17:37:00 -----
[(u'#BitcoinHoping', 4), (u'#BitcoinTo', 4), (u'#btc', 3), (u'#TRON', 3), (u'#1SG', 2), (u'#Ethereum', 2), (u'#LatokenApp', 2), (
```

Code:

```
from pyspark.streaming.kafka import KafkaUtils
from pyspark import SparkConf, SparkContext
from pyspark.streaming import StreamingContext
from pyspark.sql import Row, SQLContext
import sys
import json
import time

def process_rdd(time, rdd):
    print("----- %s -----" % str(time))
    try:
        print(rdd.top(10, key=lambda x: x[1]))
    except:
        e = sys.exc_info()
        print("Error: ", e)

if __name__ == '__main__':
    sc = SparkContext(appName="ini")
    sc.setLogLevel("WARN")

    ssc = StreamingContext(sc, 10)
    ssc.checkpointIntervalInMinutes = 1
    kafkaStream = KafkaUtils.createStream(ssc, 'dicvd/ie.cuhk.edu.hk:2181', 'test', {'985-test': 1})

    words = kafkaStream.flatMap(lambda x: x[1].split()).filter(lambda x: x.startswith("#") and x[1].lower() != "bitcoin" and len(x) > 1).map(lambda x: (x[1]).reduceByKeyAndOrder(lambda x, y: x + y, lambda x, y: x - y, 100, 128))
    words.foreachRDD(process_rdd)

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

Q3

The producer is the same as in q2

Code:

```
1  from pyspark.streaming.kafka import KafkaUtils
2  from pyspark import SparkConf, SparkContext
3  from pyspark.streaming import StreamingContext
4  from pyspark.sql import Row, SQLContext
5  import sys
6  import json
7  import time
8
9  from pyspark.sql import SparkSession
10 from pyspark.sql.functions import explode
11 from pyspark.sql.functions import split
12 from pyspark.sql import functions as F
13 from pyspark.sql.functions import window
14 from pyspark.sql.functions import size
15
16
17 if __name__ == "__main__":
18
19     spark = SparkSession \
20         .builder \
21         .appName("Q3_t") \
22         .getOrCreate()
23
24     lines = spark \
25         .readStream \
26         .format("kafka") \
27         .option("kafka.bootstrap.servers", "dicvmd7.ie.cuhk.edu.hk:6667") \
28         .option("subscribe", '983-ft') \
29         .load()
30
31     # print(lines.isStreaming())
32     # lines.isStreaming()
33
34     words = lines.select(
35         explode(
36             split(lines.value, " ")
37         ).alias("word"), "timestamp"
38     )
39
40
41     wordCounts = words.select("*")\
42         .filter(words.word.startswith("#")) \
43         .withWatermark("timestamp", "2 minutes") \
44         .groupBy(
45             window("timestamp", "10 minutes", "5 minutes"),
46             "word") \
47         .count().sort(F.col("count").desc())
48
49
50     query = wordCounts \
51         .writeStream \
52         .option("checkpointLocation", "./checkpoint") \
53         .outputMode("complete") \
54         .format("console") \
55         .start()
56
57
58     query.awaitTermination()
59
60
```

Output:

Batch: 4360

window		word count	
[1970-01-01 08:00...		126	
[,]			115
[,]	111		
[,]		87	
[1970-11-02 18:04...	82		
[,]		79	
[1970-11-02 18:04...	... 78		
[,]		75	
[1970-01-01 08:00...		74	
[1970-01-01 08:00...		66	
[,]	... 63		
[1970-11-02 18:04...	... 58		
[,]	... 55		
[1970-11-02 18:04...	6... 54		
[,]	Y... 51		
[1970-11-02 18:04...	'... 50		
[,]		47	
[1970-11-02 18:04...	E... 46		
[,]	◆... 43		
[1970-01-01 08:00...		42	

only showing top 20 rows