CSP554—Big Data Technologies

Assignment #8

Worth: 20 points

Exercise 1) 5 points

1.(1 point) Extract-transform-load (ETL) is the process of taking transactional business data (think of data collected about the purchases you make at a grocery store) and converting that data into a format more appropriate for reporting or analytic exploration. What problems was encountering with the ETL process at Twitter (and more generally) that impacted data analytics?

Answer: Due to the latency caused by ETL pipelines, business intelligence was executed on the day-old data because nightly jobs were the norm. Organizations started requesting increasingly recent data as business activity picked up speed.

2.(1 point) What example is mentioned about Twitter of a case where thelambda architecture would be appropriate?

Answer: The example used with Twitter is that if we wanted to track tweet impressions, we would like real-time updates as users are currently tapping, swiping, and clicking, but we would also like historical counts going back to the moment a tweet was posted. For instance, take a look at a tweet from Donald Trump from last year that is currently experiencing a surge in engagement.

3.(2 points) What did Twitter find were the two of the limitations of using thelambda architecture?

Answer: Logs are added to a Hadoop data warehouse by Twitter. Even in the best case scenario, the logging pipeline imposed a delay, so the logs were a few hours old.
As a result, a dashboard showing tweet impressions generated solely by MapReduce would always be a few hours old. These are the two limitations that Twitter faced while using lambda architecture.

4.(1 point) What is the Kappa architecture?

Answer: Everything is stream in Kappa architecture and we only need stream processing engine to process data. It basically is a distributed log. The table abstraction is added by Kafka Streams as a firstclass citizen and is cleverly implemented as compacted topics. This log/table duality is therefore effectively captured by Kafka.

5.(1 point) Apache Beam is one framework that implements a kappaarchitecture. What is one of the distinguishing features of Apache Beam?

Answer: There is no distinction between batch and streaming calculations in the Beam concept. Instead, the distinction between bounded and unbounded datasets is the main one. Simply streaming across bounded datasets is what we would typically refer to as batch processing.

Exercise 2) 5 points

a).(1.25 points) What is the Kappa architecture and how does it differ from the lambda architecture?

Answer: Kappa is a stream processing architecture designed to work with Samza(Stream processor) . The fundamental notion of Kappa is to only do recomputation when the business logic changes by replaying historical data, as opposed to performing all computation in the batch layer on a regular basis. That is the main difference between Lambda and Kappa architecture.

b). (1.25 points) What are the advantages and drawbacks of pure streaming versusmicro-batch real-time processing systems?

Answer: While batch-oriented systems offer exceptional resource-efficiency at the cost of latency that is too high for real-time applications, pure stream-oriented systems, such as Storm and Samza, give very low latency and very high per-item costs. Some systems, like Storm Trident and Spark Streaming, use micro-batching techniques to balance throughput and latency: While Spark Streaming limits batch size in a native batch processor to decrease latency, Trident aggregates tuples into batches to relax the one-at-a-time processing model in favor of improved throughput.

c). (1.25 points) In few sentences describe the data processing pipeline in Storm.

Answer: The nodes upstream, known as bolts, perform processing, write data to external storage, and occasionally transfer tuples farther downstream themselves. Spouts are the nodes that ingest input and therefore start the data flow in the topology. Storm includes a number of built-in groupings to manage data flow across nodes, such as those for shuffling or hash-partitioning a stream of tuples according to an attribute value, but it also supports arbitrary new groupings.
Storm by default distributes spouts and bolts in a round-robin method throughout the cluster's nodes.

d).(1.25 points) How does Spark streaming shift the Spark batch processing approach to work on real-time data streams?

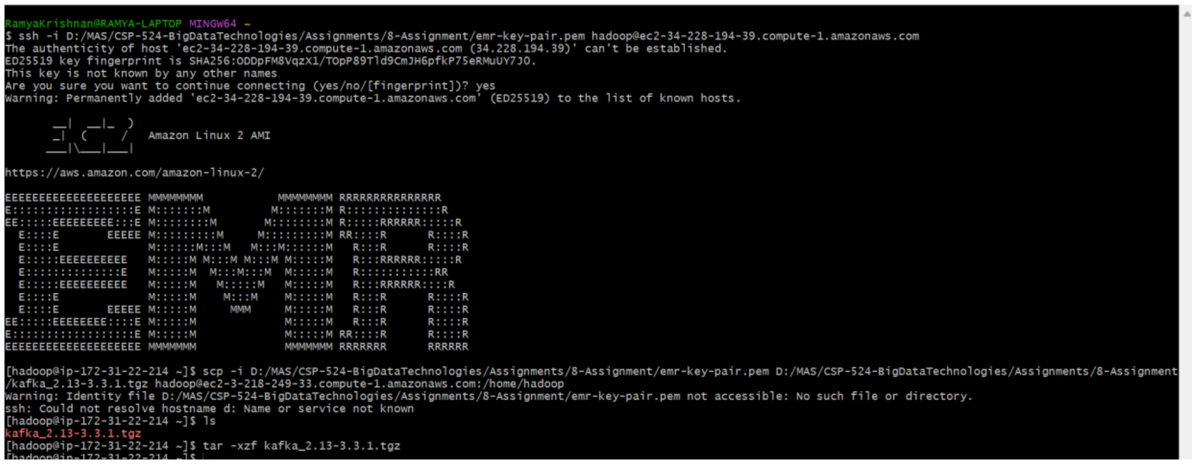
Answer: By dividing the stream of incoming data items into manageable batches, converting them into RDDs, and processing them as usual, Spark Streaming adapts Spark's batch-processing strategy to real-time requirements. Additionally, it handles data distribution and flow automatically.

Exercise 3) 5 points

Enter the following command:

```
tar -xzf kafka_2.13-3.3.1.tgz
```

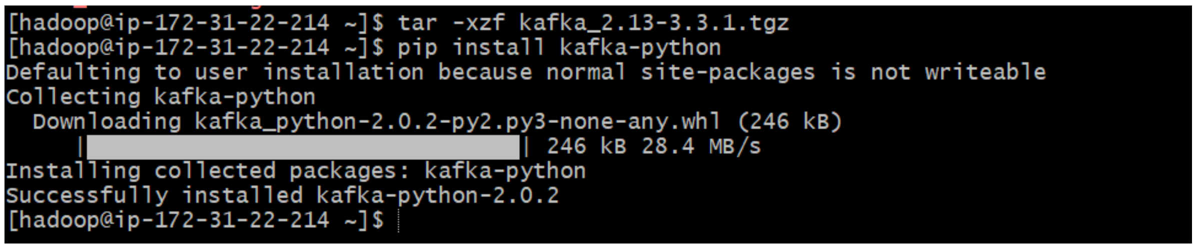
Note, this will create a new directory (kafka_2.13-3.3.1) holding the kakfa software release.



```
RamyaKrishnan@RAMYA-LAPTOP MINGW64 ~
$ ssh -i D:/MAS/CSP-524-BigDataTechnologies/Assignments/8-Assignment/emr-key-pair.pem hadoop@ec2-34-228-194-39.compute-1.amazonaws.com
The authenticity of host 'ec2-34-228-194-39.compute-1.amazonaws.com (34.228.194.39)' can't be established.
ED25519 key fingerprint is SHA256:ODDpFM8VqzX1/TOpP89Tld9CmJH6pfkP75eRMuUY7JO.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-34-228-194-39.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-22-214 ~]$ scp -i D:/MAS/CSP-524-BigDataTechnologies/Assignments/8-Assignment/emr-key-pair.pem D:/MAS/CSP-524-BigDataTechnologies/Assignments/8-Assignment
/kafka_2.13-3.3.1.tgz hadoop@ec2-3-218-249-33.compute-1.amazonaws.com:/home/hadoop
Warning: Identity file D:/MAS/CSP-524-BigDataTechnologies/Assignments/8-Assignment/emr-key-pair.pem not accessible: No such file or directory.
ssh: Could not resolve hostname d: Name or service not known
[hadoop@ip-172-31-22-214 ~]$ ls
kafka_2.13-3.3.1.tgz
[hadoop@ip-172-31-22-214 ~]$ tar -xzf kafka_2.13-3.3.1.tgz
[hadoop@ip-172-31-22-214 ~]$
```

Then enter this command:

```
pip install kafka-python
```

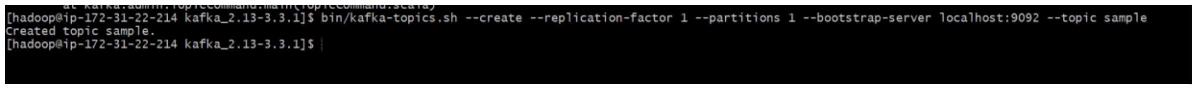


```
[hadoop@ip-172-31-22-214 ~]$ tar -xzf kafka_2.13-3.3.1.tgz
[hadoop@ip-172-31-22-214 ~]$ pip install kafka-python
Defaulting to user installation because normal site-packages is not writeable
Collecting kafka-python
  Downloading kafka_python-2.0.2-py2.py3-none-any.whl (246 kB)
     |████████████████████████████████| 246 kB 28.4 MB/s
Installing collected packages: kafka-python
Successfully installed kafka-python-2.0.2
[hadoop@ip-172-31-22-214 ~]$
```

Step E – Create a Kafka topic

In the Producer-Term, enter the following command:

```
cd kafka_2.13-3.3.1

bin/kafka-topics.sh --create --replication-factor 1 --partitions 1 --bootstrap-server localhost:9092 --topic sample
```



```
[hadoop@ip-172-31-22-214 kafka_2.13-3.3.1]$ bin/kafka-topics.sh --create --replication-factor 1 --partitions 1 --bootstrap-server localhost:9092 --topic sample
Created topic sample.
[hadoop@ip-172-31-22-214 kafka_2.13-3.3.1]$
```

Here we create a new kafka topic called ‘sample’. You can use this command to create a topic with any name you like. Try creating a few more topics.

To list the topics that you created you can enter the following into the Producer-Term (note some default topics already exist):

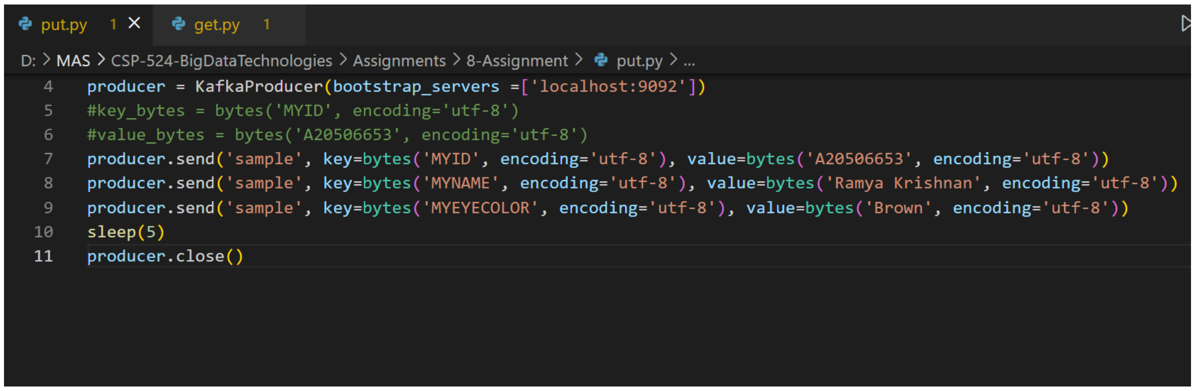
```
bin/kafka-topics.sh --list --bootstrap-server localhost:9092
```

a)

Execute this program in the Producer-Term, use the command line (you might need to provide a full pathname depending on where your python program is such as /home/hadoop/someplace/put.py):

```
python put.py
```

Submit the program as your answer to ‘part a’ of this exercise.



```
put.py 1 ×   get.py 1
D: > MAS > CSP-524-BigDataTechnologies > Assignments > 8-Assignment > put.py > ...
producer = KafkaProducer(bootstrap_servers =['localhost:9092'])
#key_bytes = bytes('MYID', encoding='utf-8')
#value_bytes = bytes('A20506653', encoding='utf-8')
producer.send('sample', key=bytes('MYID', encoding='utf-8'), value=bytes('A20506653', encoding='utf-8'))
producer.send('sample', key=bytes('MYNAME', encoding='utf-8'), value=bytes('Ramya Krishnan', encoding='utf-8'))
producer.send('sample', key=bytes('MYEYECOLOR', encoding='utf-8'), value=bytes('Brown', encoding='utf-8'))
sleep(5)
producer.close()
```

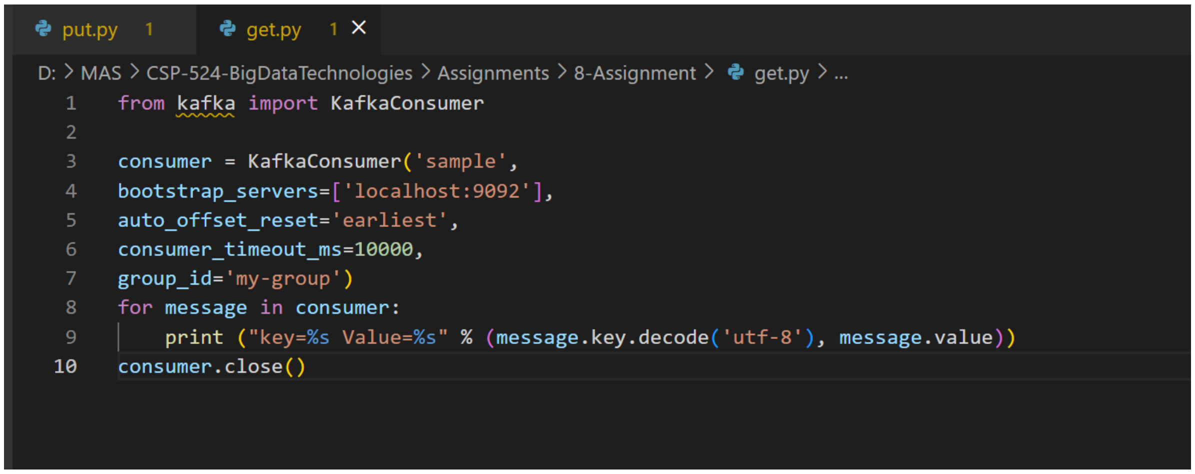
b)

Execute this program in the Consumer-Term. Use the command line:

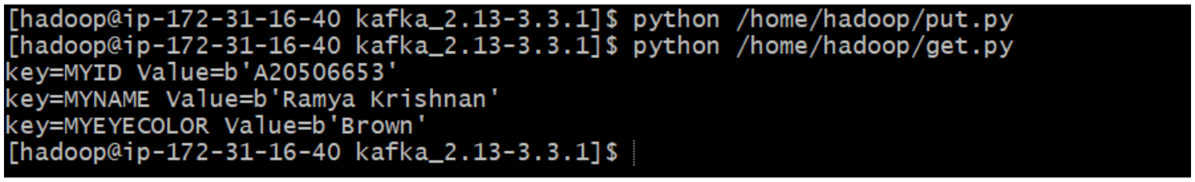
python get.py

Note, if needed you can terminate the program by entering ‘ctrl-c’.

Submit the program and a screenshot of its output as your answer to ‘part b’ of this exercise.

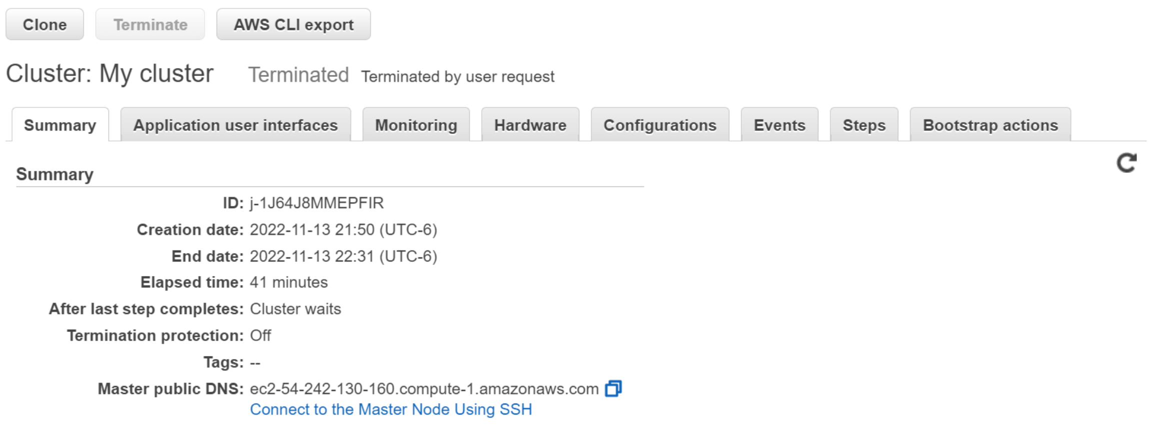


```
put.py 1   get.py 1 ×
D: > MAS > CSP-524-BigDataTechnologies > Assignments > 8-Assignment > get.py > ...
from kafka import KafkaConsumer

consumer = KafkaConsumer('sample',
bootstrap_servers=['localhost:9092'],
auto_offset_reset='earliest',
consumer_timeout_ms=10000,
group_id='my-group')
for message in consumer:
    print ("key=%s Value=%s" % (message.key.decode('utf-8'), message.value))
consumer.close()
```



```
[hadoop@ip-172-31-16-40 kafka_2.13-3.3.1]$ python /home/hadoop/put.py
[hadoop@ip-172-31-16-40 kafka_2.13-3.3.1]$ python /home/hadoop/get.py
key=MYID Value=b'A20506653'
key=MYNAME Value=b'Ramya Krishnan'
key=MYEYECOLOR Value=b'Brown'
[hadoop@ip-172-31-16-40 kafka_2.13-3.3.1]$
```

c) Remember to terminate your EMR cluster!



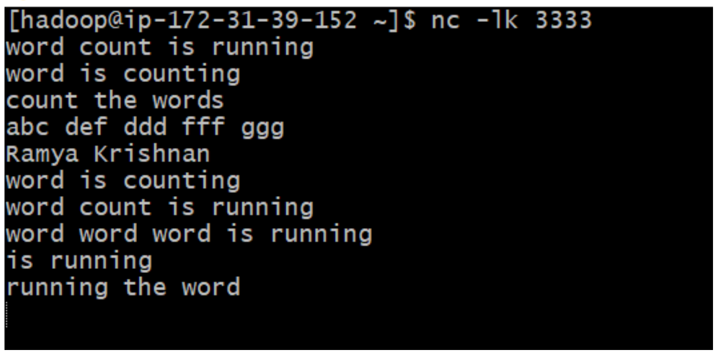
Cluster: My cluster Terminated Terminated by user request

Summary

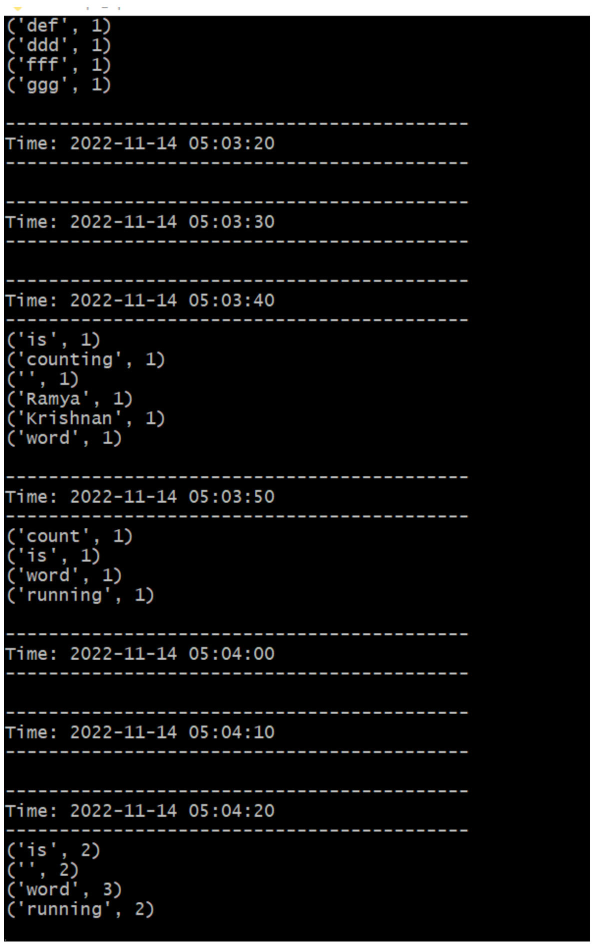
ID: j-1J64J8MMEPFIR
Creation date: 2022-11-13 21:50 (UTC-6)
End date: 2022-11-13 22:31 (UTC-6)
Elapsed time: 41 minutes
After last step completes: Cluster waits
Termination protection: Off
Tags: --
Master public DNS: ec2-54-242-130-160.compute-1.amazonaws.com
Connect to the Master Node Using SSH

Exercise 4) 5 points

11) Now in the EC2-1 window enter one or more lines of text and press Enter/Return after each one including the last. You should see the word count results scroll by in the EC2 window



```
[hadoop@ip-172-31-39-152 ~]$ nc -lk 3333
word count is running
word is counting
count the words
abc def ddd fff ggg
Ramya Krishnan
word is counting
word count is running
word word word is running
is running
running the word
```


```
('def', 1)
('ddd', 1)
('fff', 1)
('ggg', 1)

-------------------------------------------
Time: 2022-11-14 05:03:20
-------------------------------------------

-------------------------------------------
Time: 2022-11-14 05:03:30
-------------------------------------------

-------------------------------------------
Time: 2022-11-14 05:03:40
-------------------------------------------
('is', 1)
('counting', 1)
('', 1)
('Ramya', 1)
('Krishnan', 1)
('word', 1)

-------------------------------------------
Time: 2022-11-14 05:03:50
-------------------------------------------
('count', 1)
('is', 1)
('word', 1)
('running', 1)

-------------------------------------------
Time: 2022-11-14 05:04:00
-------------------------------------------

-------------------------------------------
Time: 2022-11-14 05:04:10
-------------------------------------------

-------------------------------------------
Time: 2022-11-14 05:04:20
-------------------------------------------
('is', 2)
('', 2)
('word', 3)
('running', 2)
```