CSP – 554 Big Data Technologies

Assignment #9

Rutul Mehta
A20476293

Exercise – 1

a.

- The primary idea behind the Kappa design is to avoid recompiling all data in the batch layer on a regular basis, instead doing all computation in the stream processing system and only do computation when the business logic changes by replaying past data.
- Lambda architecture, on the other hand, includes both batch-oriented and real-time systems. As a result, it addresses both the volume and velocity challenges of big data. Batch Layer, Speed Layer, and Serving Layer are the three layers of Lambda architecture.

b.

- Pure Streaming takes care of data as soon as it arrives. It reduces delay at the expense of a high per-item price.
- Data is buffered and processed in batches in micro-batch real-time processing systems. It improves efficiency while also increasing the time an individual item spends in the data flow.

c.

- In Storm, a data processing pipeline (topology) is a directed network in which data flow is represented by directed edges between nodes. Individual processing steps are represented by nodes.
- Spouts and bolts are the two types of nodes. Spouts are responsible for ingesting and emitting tuples to bolts, which are downstream nodes. Processing, writing data to external storage, and delivering tuples farther downstream are all handled by bolts.

d.

- Spark streaming shifts spark's batch-processing approach make chunks of incoming stream data into small batches. Then transform into RDDs and process them. Distribution of the data flow is also taken care.

Exercise: 2

List of topics created

```
Created topic A20476293.
[hadoop@ip-172-31-28-125 kafka_2.13-3.0.0]$ bin/kafka-topics.sh --list --bootstr
ap-server localhost:9092
A20476293
Rutul
sample
[hadoop@ip-172-31-28-125 kafka_2.13-3.0.0]$
```

a.

put.py

```
producer.close()[hadoop@ip-172-31-28-125 ~]$ cat put.py
from kafka import KafkaProducer
import json
producer = KafkaProducer(value_serializer=lambda v: json.dumps(v).encode('utf-8'))
data = {'MYID':'A20476293', 'MYNAME':'Rutul Mehta', 'MYEYECOLOR':'black'}
producer.send('sample', data)
producer.close()[hadoop@ip-172-31-28-125 ~]$
```

b.

get.py with output

```
[hadoop@ip-172-31-28-125 ~]$ cat get.py
from kafka import KafkaConsumer
from json import loads
consumer = KafkaConsumer('sample', auto_offset_reset='earliest', consumer_timeout_ms=1000, value_deserializer=lambda x: loads(x.decode('utf-8')))
for msg in consumer:
    #print(msg.value)
    print("Key=MYID, Value={}".format(msg.value['MYID']))
    print("Key=MYNAME, Value={}".format(msg.value['MYNAME']))
    print("Key=MYEYECOLOR, Value={}".format(msg.value['MYEYECOLOR']))
consumer.close()[hadoop@ip-172-31-28-125 ~]$ python get.py
Key=MYID, Value=A20476293
Key=MYNAME, Value=Rutul Mehta
Key=MYEYECOLOR, Value=black
[hadoop@ip-172-31-28-125 ~]$
```

c.

I have deleted my cluster.