ID2221 DATA INTENSIVE COMPUTING
LAB ASSIGNMENT 3

Rithika Harish Kumar(rihk@kth.se)
Catherine Rakama(rakama@kth.se)

TASK

To implement a spark streaming application that reads streaming data i.e (key,value) pairs from Kafka and stores the result i.e the continuous update of average value of each key or the (key, average value) pairs in Cassandra.

GIVEN

The (key,value) pairs from Kafka can be generated by running the code provided within src/generator folder. Also the base code is provided within src/sparkstreaming folder with build files.

STEPS

1. Edit the KafkaSpark.scala file. It is explained below:

- Inside the main function connect to cassandra using:

```
val cluster = Cluster.builder().addContactPoint("127.0.0.1").build()
val session = cluster.connect()
```

- Build a keyspace(avg_space) and table(avg) with two columns (key, average value) in cassandra using:

```
session.execute("CREATE KEYSPACE IF NOT EXISTS avg_space WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor' : 1 };")
session.execute("CREATE TABLE IF NOT EXISTS avg_space.avg (word text PRIMARY KEY, count float);")
```

- Connect to Kafka using:

```
val kafkaConf = Map("metadata.broker.list" -> "localhost:9092","zookeeper.connect" -> "localhost:2181","group.id" -> "kafka-spark-streaming","zookeeper.connection.timeout.ms" -> "1000")
```

- Configure Spark and create a streaming context. Checkpointing is to save the generated RDDs to reduce recovery time.

```
val sparkConf = new SparkConf().setAppName("KafkaSpark").setMaster("local[2]")
val ssc = new StreamingContext(sparkConf, Seconds(1))
ssc.checkpoint("checkpoint")
```

- Read the key, value pairs from Kafka.

```
val messages = KafkaUtils.createDirectStream[String, String, StringDecoder, StringDecoder](ssc, kafkaConf, "avg".split(",").toSet)
```

- mapWithState is used with a function as its input to sum the values for each key and calculate the average.

- Store the result in cassandra.

```
stateDstream.saveToCassandra("avg_space","avg", SomeColumns("word", "count"))
```

2. Save the file and then Open new terminal and run the Zookeeper server.

```
$KAFKA_HOME/bin/zookeeper-server-start.sh $KAFKA_HOME/config/zookeeper.properties
```

3. Open new terminal and run the Kafka server.

```
$KAFKA_HOME/bin/kafka-server-start.sh $KAFKA_HOME/config/server.properties
```

ID2221 DATA INTENSIVE COMPUTING
LAB ASSIGNMENT 3

Rithika Harish Kumar(rihk@kth.se)
Catherine Rakama(rakama@kth.se)

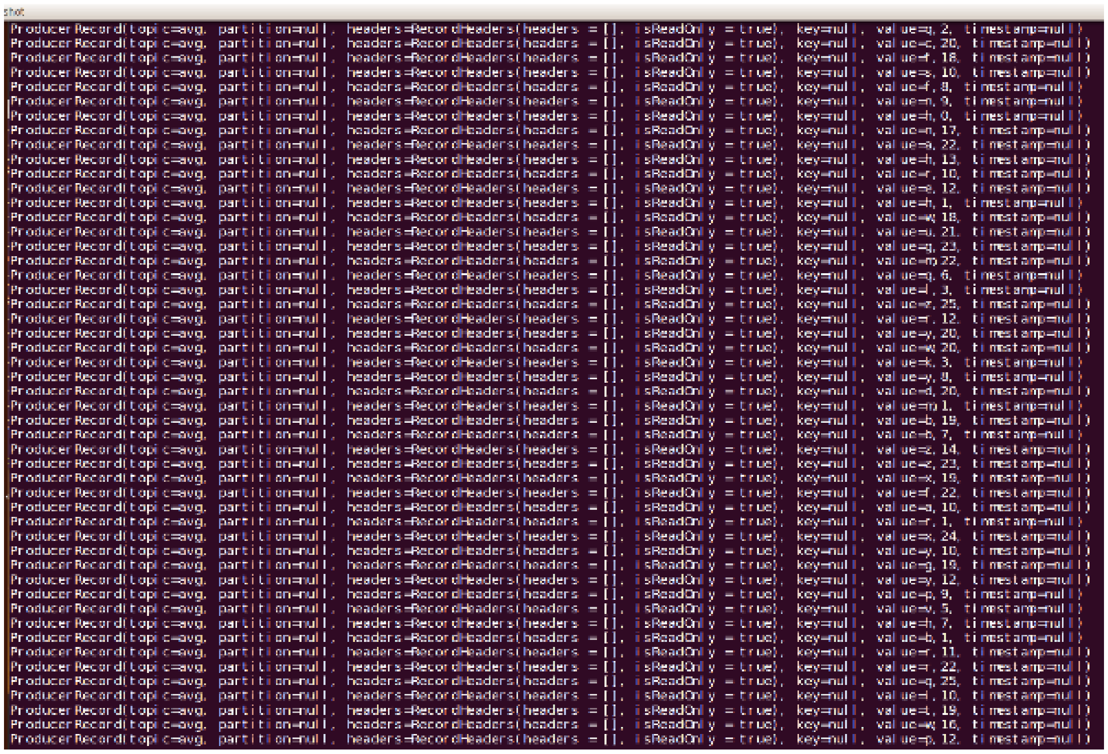
4. Create a topic ‘avg’.

$KAFKA_HOME/bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1
--topic avg

5. Open new terminal and run Cassandra.

$CASSANDRA_HOME/bin/cassandra -f

6. Open new terminal and change the path to the src/generator folder and execute the command ‘sbt run’. The following stream was generated :



```
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=q, 2, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=c, 20, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=f, 18, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=s, 10, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=f, 8, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=n, 9, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=h, 0, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=n, 17, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=a, 22, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=h, 13, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=r, 10, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=e, 12, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=h, 1, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=w, 18, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=u, 21, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=q, 23, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=m, 22, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=q, 6, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=l, 3, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=z, 25, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=t, 12, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=y, 20, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=w, 20, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=k, 3, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=y, 8, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=d, 20, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=m, 1, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=b, 19, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=b, 7, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=z, 14, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=z, 23, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=x, 19, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=f, 22, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=a, 10, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=r, 1, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=x, 24, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=y, 10, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=g, 19, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=y, 12, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=p, 9, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=v, 5, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=h, 7, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=b, 1, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=r, 11, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=t, 22, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=q, 25, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=l, 10, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=t, 19, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=w, 16, timestamp=null)
ProducerRecord(topic=avg, partition=null, headers=RecordHeaders(headers = [], isReadOnly = true), key=null, value=p, 12, timestamp=null)
```

7. Open new terminal and change the path to src/sparkstreaming and execute the command ‘sbt run’.
8. Results can be viewed in Cassandra.

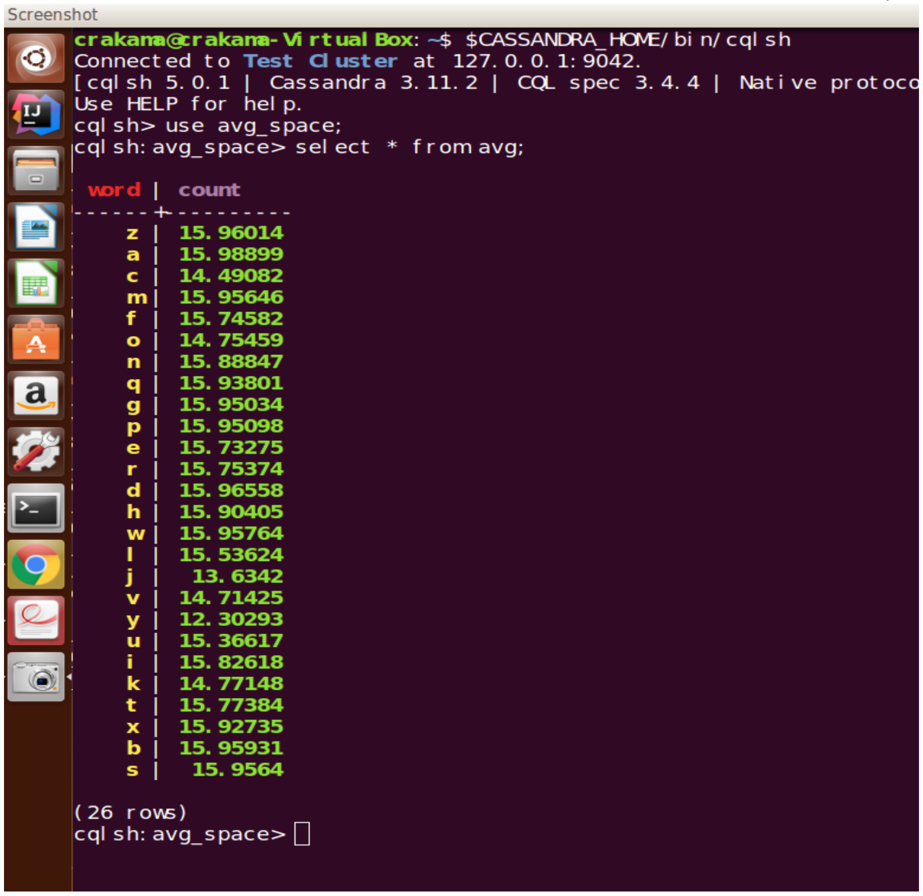
RESULT

To view the result:
Open a terminal and run the commands.
$CASSANDRA_HOME/bin/cqlsh
Use avg_space;
Select * from avg;

A screenshot of the obtained result is shown below.

ID2221 DATA INTENSIVE COMPUTING
LAB ASSIGNMENT 3

Rithika Harish Kumar(rihk@kth.se)
Catherine Rakama(rakama@kth.se)



```
crakama@crakama-VirtualBox:~$ $CASSANDRA_HOME/bin/cqlsh
Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.2 | CQL spec 3.4.4 | Native protoco
Use HELP for help.
cqlsh> use avg_space;
cqlsh:avg_space> select * from avg;

 word | count
------+---------
    z | 15.96014
    a | 15.98899
    c | 14.49082
    m | 15.95646
    f | 15.74582
    o | 14.75459
    n | 15.88847
    q | 15.93801
    g | 15.95034
    p | 15.95098
    e | 15.73275
    r | 15.75374
    d | 15.96558
    h | 15.90405
    w | 15.95764
    l | 15.53624
    j |  13.6342
    v | 14.71425
    y | 12.30293
    u | 15.36617
    i | 15.82618
    k | 14.77148
    t | 15.77384
    x | 15.92735
    b | 15.95931
    s |  15.9564

(26 rows)
cqlsh:avg_space>
```

Clear pictures can be found in the images folder.