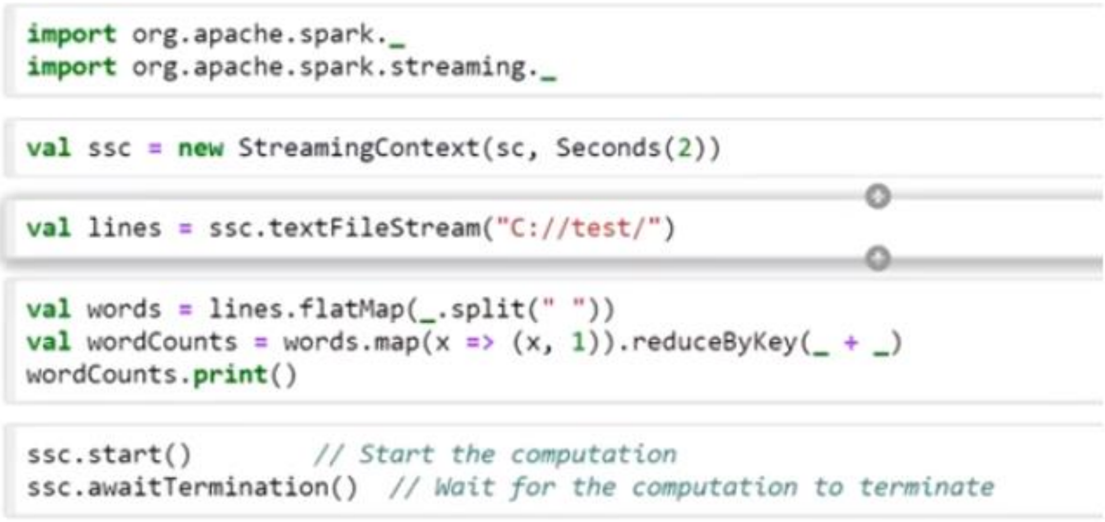
UE18CS322-Big Data- Unit 4

Question Bank

1. How is stream processing is different from Batch Processing?
2. Give examples for streamed data.
3. Comparison of data at motion and data at rest
4. DBMS Vs DSMS
5. Point out the differences between standing and adhoc queries. Given an example you should be able to identify if it is standing or adhoc query.
6. What Spark streaming library can do?
7. Do you find any difference between spark streaming process and batch processing? If so, draft the differences.
8. What do you mean by Dstream and what are the operations associated?
9. Is map (), reducebyKey() and filter() stateful transformation?
10. Explain the Spark streaming architecture.
11. Can we maintain long lived state on a DStream?
12. Consider a Dstream on twitter data
    - A sequence of tuples that contain <username, Password>
    - Need to get Hash tags from Twitter

    Show Streaming spark design for the same.
13. Show visually the Spark streaming flow.
14. **What is the different type of file sources for Spark Streaming?**
15. **What is the relationship between Dstream and RDD?**
16. Consider a Dstream on stock quotes generated similar to earlier that contains
    A sequence of tuples that contain <company name, stock sold>
    Need to find total shares sold per company in the last 1 minute
    Show Streaming spark design for the same.
17. **Do you understand stateful and stateless transformations in Spark Streaming? If so, identify the transformation that needs to be performed for the given scenario.**
    Consider a Dstream on stock quotes that contain a sequence of tuples that contain <company name, stock sold>
    What’s the max amount of stock sold across the whole day for a company?



```
import org.apache.spark._
import org.apache.spark.streaming._

val ssc = new StreamingContext(sc, Seconds(2))

val lines = ssc.textFileStream("C://test/")

val words = lines.flatMap(_.split(" "))
val wordCounts = words.map(x => (x, 1)).reduceByKey(_ + _)
wordCounts.print()

ssc.start()           // Start the computation
ssc.awaitTermination()  // Wait for the computation to terminate
```

18. What is the output of the above code?
19. Is UpdateStatebyKey operation is a stateful transformation?
20. **Given below a stream code, based on the snippet try to answer few questions:** tweets.updateStateByKey(tweet => updateMood(tweet))

    - What has to be the structure of the RDD *tweets*?
    - What does the function *updateMood* do?
    - How would you track sessions, maintaining the arbitrary state?
21. Do you think persist () and cache () perform similar operation as checkpointing?
22. Suppose if you try to find the word count for the last 30 minutes, typically there will be one receiver that receives all data and stores it in a executor and the processing happens here. Adding more node doesn't help here. So how do we achieve good throughput?
23. Enlist the Kafka Components.
24. Draw the process diagram of Kafka
25. Can we use Kafka without Zookeeper?
26. Is Kafka a mere messaging? If not how are they both different?
27. Consider a bookstore portal with various activities such as
    - Login
    - List books
    - Get book details
    - Buy book
    - Check status of order
    - Return book
    - Logout

Assume we have 3 backend modules

- Security
- Order processing
- Book information

Would you use a topic-based or content-based system? What would be the topics / content...?

28. Why are replications critical in Kafka?
29. What is the process for starting a Kafka Server?
30. Suppose we have a Kafka system
    - 1 topic
    - 3 servers
    - 3 partitions
    - 3 replicas per partition

    Consumer group with 3 instances
    Draw a diagram showing
    - Servers
    - Partitions
    - Consumer instances
    - Partition assignments
31. What is the role of leader?
32. How is the streaming algorithms differ from conventional algorithms?
33. What do you mean by cardinality problem? Does Flajolet Martin helps you to solve the cardinality problem?
34. Heavy hitter's problem demands an algorithm that can execute in constant time and occupying sub linear space. Which is the suitable algorithm to solve heavy hitters?
35. can we compute the majority element with a single left-to-right pass through the array? How?
36. Why Sampling algorithms or hash table implementations not ideal for heavy hitters? Justify in terms of space and time complexity.
37. To search for an element and finding its presence which data structure is an ideal choice?
38. Why do you say bloom filter as probabilistic data structure?