**Frequently asked Data Engineering questions:**

Spark 

1. Does spark streaming provide at most once, exactly once and at least once delivery semantics?

Yes, spark streaming can provide all the 3. But for exactly once, the output database should handle duplicates. So, with correct database, exactly once is also possible.

1. Which exception will we get if we try to initialize a database driver at driver and use it in executors?

NotSerializable exception

1. When an RDD lineage grows too much, how can we cut it short or reduce it?  
   Checkpoint the RDD
2. Which of these tend to have more shuffling of data across executors- reduceByKey or groupByKey?

groupByKey

1. Which kind of an RDD supports groupByKey operations?

Pair RDD

1. What’s the difference between persist() and cache()?

persist() allows the user to specify the storage level whereas cache() uses the default storage level of MEMORY ONLY.

1. Which framework does Spark use internally for communication between different executors and for scheduling tasks?

Akka

1. Which Kafka consumer does Spark streaming uses- High level or low level?  
   Low level
2. Where does Spark streaming store Kafka’s offsets?

In files during checkpointing

1. Which is an alternative better serializer to use in Spark other than the default Java serializer?

Kryo serializer

Scala 

1. What’s the alternative recommended way of treating nulls in Scala? What’s the advantage?

Use an “Option” which can either be a “Some” or a “None”. Using ”Option” allows for a functional style of programming and eliminates “NullPointerException”.

1. By which Scala feature does a “String” have a “toInt” function?  
   Implicit class “StringOps”
2. What’s the difference between “val” and “var”?  
   “val” is an immutable variable and cannot change its value. “var” is mutable and can be changed any number of times.
3. What’s “Nil”?

An empty list

1. What’s the difference between “map” and “flatMap”?

Map works by applying a function to each element in the list and expects another element as output. Flatmap works by applying a function that returns a sequence for each element in the list, and flattening the results.

Hadoop/Big Data 

1. Which of these are columnar or row oriented file formats- Avro, Parquet, ORC, JSON?

Row oriented- Avro and JSON

Column oriented- ORC and Parquet

1. Where does HDFS lie in the CAP theorem? Does it choose availability or consistency?

HDFS is a CP (Consistency + Partitioning) system.

1. What about Cassandra?

It has tunable consistency. By using quorums, we can choose either of consistency or availability

1. What’s the difference between ORDER BY and SORT BY in Hive?  
   ORDER BY guarantees total ordering of data. SORT BY only ensures ordering of each reducer’s output.
2. Does Kafka allow multiple consumers to consume from a single partition?

1 partition can be consumed by only 1 consumer within a consumer group. Different consumer groups can consume the same partition. However, a consumer can consume multiple partitions.

1. What’s the O-notation time complexity for retrieval from a HashMap in Java- Best and worst?  
   Best- O(1) and Worst- O(n)

**Some other topics you can read:**

**Hadoop/Big Data:**CAP Theorem

MR lifecycle

RDD fault tolerance

Speculation Mode

Hive Query Flow

override-hashcode and equals

kafka partitioning

LRU cache

Spark (standalone mode, Fault tolerance, transformation and action, map and flatmap, Avro & Parquet, Shuffle, spilling, partitioner )

HBase Data Modeling

disadvantage of kafka

parallelism in spark and kafka

coalesce

for java - performance tuning/jvm internals

caching vs broadcast

caching in low memory

Implementation of LRU Cache,

Mutable list & map

Parquet file - limit i/o operation,columnal fast, consumes less space

Checkpointing- Streaming use case

Role of driver,executor, clustr manager

Lazy Evaluation

RDD - Transformation & Action

Mode of execution - yarn/local

Executors

Cache vs persist

Lineage graph: track if RDD1 -> RDD2

reduce & reducebykey

Broadcast variable

Acummulators: data from worker to driver

Flow of job: DAG->executors register->jar copied->job run->data sent to driver by executor

Indexes: Hash index/Btree

Adapter/Decorator design pattern

Map vs Flatmap

**Java/DS/Algo:**

Garbage collection

Externalization(Custom Serialization)

Pre-emptive Scheduling/ Time slicing

Hash map, Concurrent Hash map, Hash table

Implementation of Queue using Array

fail-fast vs fail-safe

recurssion(sum of 2 linked list)

maximum sum of sub array