PRACTICE INTERVIEW QUESTIONS ON APACHE SPARK ,PYSPARK FOR DATA ENGINEERS (MADE BY RISHABH PANDEY)

SET OF 82 QUESTIONS

1. How is Apache Spark different from MapReduce?

Ans:

Apache Spark	MapReduce
Spark processes data in batches as well as in real-time	MapReduce processes data in batches only
Spark runs almost 100 times faster than Hadoop Map Reduce	Hadoop MapReduce is slower when it comes to large sc processing
Spark stores data in the RAM i.e. in-memory. So, it is easier to retrieve it	Hadoop MapReduce data is stored in HDFS and henc longer time to retrieve the data
Spark provides caching and in-memory data storage	Hadoop is highly disk-dependent

2. What are the important components of the Spark ecosystem?

Ans:

Apache Spark has 3 main categories that comprise its ecosystem. Those are:

- Language support: Spark can integrate with different languages to applications and perform analytics. These languages are Java, Python, Scala, and R.
- Core Components: Spark supports 5 main core components. There are Spark Core, Spark SQL, Spark Streaming, Spark MLlib, and GraphX.
- Cluster Management: Spark can be run in 3 environments. Those are the Standalone cluster, Apache Mesos, and YARN.

3. What are the different cluster managers available in Apache Spark?

Ans:

- Standalone Mode: By default, applications submitted to the standalone mode cluster will run in FIFO order, and each application will try to use all available nodes. You can launch a standalone cluster either manually, by starting a master and workers by hand or use our provided launch scripts. It is also possible to run these daemons on a single machine for testing.
- Apache Mesos: Apache Mesos is an open-source project to manage computer clusters, and can also run Hadoop applications. The advantages of deploying Spark with Mesos include dynamic partitioning between Spark and other frameworks as well as scalable partitioning between multiple instances of Spark.
- Hadoop YARN: Apache YARN is the cluster resource manager of Hadoop 2. Spark can be run on YARN as well.
- Kubernetes: Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.

4. What is a lazy evaluation in Spark?

Ans:

When Spark operates on any dataset, it remembers the instructions.When a transformation such as a map() is called on an RDD, the operation is not performed instantly. Transformations in Spark are not evaluated until you perform an action, which aids in optimizing the overall data processing workflow, known as lazy evaluation.

5. What makes Spark good at low latency workloads like graph processing and Machine Learning?

Ans:

Apache Spark stores data in-memory for faster processing and building machine learning models. Machine Learning algorithms require multiple iterations and different conceptual steps to create an optimal model. Graph algorithms traverse through all the nodes and edges to generate a graph. These low latency workloads that need multiple iterations can lead to increased performance.

6. How can you connect Spark to Apache Mesos?

Ans:

There are a total of 4 steps that can help you connect Spark to Apache Mesos.

- Configure the Spark Driver program to connect with Apache Mesos
- Put the Spark binary package in a location accessible by Mesos
- Install Spark in the same location as that of the Apache Mesos
- Configure the spark.mesos.executor.home property for pointing to the location where Spark is installed

7. What is a Parquet file and what are its advantages?

Ans:

Parquet is a columnar format that is supported by several data processing systems. With the Parquet file, Spark can perform both read and write operations.

Some of the advantages of having a Parquet file are:

- It enables you to fetch specific columns for access.
- It consumes less space
- It follows the type-specific encoding
- It supports limited I/O operations

8. What is shuffling in Spark? When does it occur?

Ans:

- Shuffling is the process of redistributing data across partitions that may lead to data movement across the executors. The shuffle operation is implemented differently in Spark compared to Hadoop.

  Shuffling has 2 important compression parameters:

- spark.shuffle.compress – checks whether the engine would compress shuffle outputs or not spark.shuffle.spill.compress – decides whether to compress intermediate shuffle spill files or not
- It occurs while joining two tables or while performing byKey operations such as GroupByKey or ReduceByKey

9. What are the various functionalities supported by Spark Core?

Ans:

Spark Core is the engine for parallel and distributed processing of large data sets. The various functionalities supported by Spark Core include:

- Scheduling and monitoring jobs
- Memory management
- Fault recovery
- Task dispatching
- eDataFrame

10. Explain the types of operations supported by RDDs.

Ans:

RDDs support 2 types of operation:

- **Transformations:** Transformations are operations that are performed on an RDD to create a new RDD containing the results (Example: map, filter, join, union)
- **Actions:** Actions are operations that return a value after running a computation on an RDD (Example: reduce, first, count)

11. How to programmatically specify a schema for DataFrame?

Ans:

DataFrame can be created programmatically with three steps:

- Create an RDD of Rows from the original RDD;
- Create the schema represented by a StructType matching the structure of Rows in the RDD created in Step 1.
- Apply the schema to the RDD of Rows via createDataFrame method provided by SparkSession.

12. What is a Lineage Graph?

Ans:

- A Lineage Graph is a dependencies graph between the existing RDD and the new RDD. It means that all the dependencies between the RDD will be recorded in a graph, rather than the original data.
- The need for an RDD lineage graph happens when we want to compute new RDD or if we want to recover the lost data from the lost persisted RDD.Spark does not support data replication in memory. So, if any data is lost, it can be rebuilt using RDD lineage. It is also called an RDD operator graph or RDD dependency graph.

13. Which transformation returns a new DStream by selecting only those records of the source DStream for which the function returns true?

1. map(func)

2. transform(func)

3. filter(func)

4. count()

Ans:

3) filter(func).

14. Does Apache Spark provide checkpoints?

Ans:

- Yes, Apache Spark provides an API for adding and managing checkpoints. Checkpointing is the process of making streaming applications resilient to failures. It allows you to save the data and metadata into a checkpointing directory. In case of a failure, the spark can recover this data and start from wherever it has stopped.
- There are 2 types of data for which we can use checkpointing in Spark.
- **Metadata Checkpointing:** Metadata means the data about data. It refers to saving the metadata to fault-tolerant storage like HDFS. Metadata includes configurations, DStream operations, and incomplete batches.
- **Data Checkpointing:** Here, we save the RDD to reliable storage because its need arises in some of the stateful transformations. In this case, the upcoming RDD depends on the RDDs of previous batches.

15. What are the different levels of persistence in Spark?

Ans:

- DISK_ONLY – Stores the RDD partitions only on the disk
- MEMORY_ONLY_SER – Stores the RDD as serialized Java objects with one-byte array per partition
- MEMORY_ONLY – Stores the RDD as deserialized Java objects in the JVM. If the RDD is not able to fit in the memory available, some partitions won't be cached
- OFF_HEAP – Works like MEMORY_ONLY_SER but stores the data in off-heap memory
- MEMORY_AND_DISK – Stores RDD as deserialized Java objects in the JVM. In case the RDD is not able to fit in the memory, additional partitions are stored on the disk
- MEMORY_AND_DISK_SER – Identical to MEMORY_ONLY_SER with the exception of storing partitions not able to fit in the memory to the disk

16. What is the difference between map and flatMap transformation in Spark Streaming?

Ans:

map()	flar map();
A map function returns a new DStream by passing each element of the source DStream through a function function	It is similar to map function and applies to each e RDD and it returns the result as new RDD
Spark Map function takes one element as input process it according to custom code (specified by the developer) and returns one element at a time	FlatMap allows returning 0, 1 or more elements fro function.In the FlatMap operation

17. How would you compute the total count of unique words in Spark?

Ans:

1. **Load the text file as RDD:**

- sc.textFile("hdfs://Hadoop/user/test_file.txt");

2. Function that breaks each line into words:

- def toWords(line):
- return line.split();

3. Run the toWords function on each element of RDD in Spark as flatMap transformation:

- words = line.flatMap(toWords);

4. Convert each word into (key,value) pair:

- def toTuple(word):
- return (word, 1);
- wordTuple = words.map(toTuple);

5. Perform reduceByKey() action:

- def sum(x, y):
- return x+y:
- counts = wordsTuple.reduceByKey(sum)

6. Print:

- counts.collect()

18. What are the different MLlib tools available in Spark?

Ans:

- ML Algorithms: Classification, Regression, Clustering, and Collaborative filtering
- Featurization: Feature extraction, Transformation, Dimensionality reduction, and Selection
- Pipelines: Tools for constructing, evaluating, and tuning ML pipelines
- Persistence: Saving and loading algorithms, models and pipelines
- Utilities: Linear algebra, statistics, data handling

19. What is a Sparse Vector?

Ans:

A Sparse vector is a type of local vector which is represented by an index array and a value array.

- public class SparseVector
- extends Object
- implements Vector

1. *Example: sparse1 = SparseVector(4, [1, 3], [3.0, 4.0])*
2. *where:*
3. *4 is the size of the vector*

4. *[1,3] are the ordered indices of the vector*
5. *[3,4] are the value*

20. What are the functions of Spark SQL?

Ans:

- Spark SQL is Apache Spark's module for working with structured data.
- Spark SQL loads the data from a variety of structured data sources.
- It queries data using SQL statements, both inside a Spark program and from external tools that connect to Spark SQL through standard database connectors (JDBC/ODBC).
- It provides a rich integration between SQL and regular Python/Java/Scala code, including the ability to join RDDs and SQL tables and expose custom functions in SQL.

---

21. What are the different types of operators provided by the Apache GraphX library?

Ans:

- **Property Operator:** Property operators modify the vertex or edge properties using a user-defined map function and produce a new graph.
- **Structural Operator:** Structure operators operate on the structure of an input graph and produce a new graph.
- **Join Operator:** Join operators add data to graphs and generate new graphs.

22. What are the analytic algorithms provided in Apache Spark GraphX?

Ans:

GraphX is Apache Spark's API for graphs and graph-parallel computation. GraphX includes a set of graph algorithms to simplify analytics tasks. The algorithms are contained in the

org.apache.spark.graphx.lib package and can be accessed directly as methods on Graph via GraphOps.

- **PageRank:** PageRank is a graph parallel computation that measures the importance of each vertex in a graph. Example: You can run PageRank to evaluate what the most important pages in Wikipedia are.
- **Connected Components:** The connected components algorithm labels each connected component of the graph with the ID of its lowest-numbered vertex. For example, in a social network, connected components can approximate clusters.
- **Triangle Counting:** A vertex is part of a triangle when it has two adjacent vertices with an edge between them. GraphX implements a triangle counting algorithm in the TriangleCount object that determines the number of triangles passing through each vertex, providing a measure of clustering.

23. Which Profilers do we use in PySpark?

Ans:

Custom profilers are PySpark supported in PySpark to allow for different Profilers to be used and for outputting to different formats than what is offered in the BasicProfiler.

We need to define or inherit the following methods, with a custom profiler:

- profile – Basically, it produces a system profile of some sort.
- stats – Well, it returns the collected stats.
- dump – Whereas, it dumps the profiles to a path.
- add – Moreover, this method helps to add a profile to the existing accumulated profile

Generally, when we create a SparkContext, we choose the profiler class.

24. Explain Basic Profiler.

Ans:

It is a default profiler, which we implement on the basis of cProfile and Accumulator.

25. Do we have a machine learning API in Python?

Ans:

As Spark provides a Machine Learning API, MLlib. Similarly, in Python as well, PySpark has this machine learning API.

26. Name algorithms supported in PySpark?

Ans:

There are several algorithms in PySpark:

- mllib.classification
- mllib.clustering
- mllib.fpm
- mllib.linalg
- mllib.recommendation
- spark.mllib
- Mllib.regression

27. Name parameter of SparkContext?

Ans:

The parameters of a SparkContext are:

- Master − URL of the cluster from which it connects.
- appName − Name of our job.
- sparkHome − Spark installation directory.
- pyFiles − It is the .zip or .py files, in order to send to the cluster and also to add to the PYTHONPATH.
- Environment − Worker nodes environment variables.
- Serializer − RDD serializer.
- Conf − to set all the Spark properties, an object of L{SparkConf}.
- JSC − It is the JavaSparkContext instance.

28. Which of the parameters of SparkContext we mostly use?

Ans:

Master and app name.

29. What Makes Apache Spark Good At Low-latency Workloads Like Graph Processing And Machine Learning?

Ans:

Apache Spark stores data in-memory for faster model building and training. Machine learning algorithms require multiple iterations to generate a resulting optimal model and similarly graph algorithms traverse all the nodes and edges.These low latency workloads that need multiple iterations can lead to increased performance. Less disk access and controlled network traffic make a huge difference when there is lots of data to be processed.

30. Is It Necessary To Start Hadoop To Run Any Apache Spark Application ?

Ans:

Starting hadoop is not mandatory to run any spark application. As there is no separate storage in Apache Spark, it uses Hadoop HDFS but it is not mandatory. The data can be stored in the local file system, can be loaded from the local file system and processed.

31. What Is The Default Level Of Parallelism In Apache Spark?

Ans:

If the user does not explicitly specify then the number of partitions are considered as default level of parallelism in Apache Spark.

32. Explain About The Common Workflow Of A Spark Program

Ans:

- The foremost step in a Spark program involves creating input RDD's from external data.
- Use various RDD transformations like filter() to create new transformed RDD's based on the business logic.
- persist() any intermediate RDD's which might have to be reused in future.
- Launch various RDD actions() like first(), count() to begin parallel computation , which will then be optimized and executed by Spark.

33. Name A Few Commonly Used Spark Ecosystems.

Ans:

- Spark SQL (Shark)
- Spark Streaming
- GraphX
- MLlib
- SparkR

34. What is Spark Streaming?

Ans:

At whatever point there is information streaming constantly and you need to process the information as right on time as could reasonably be expected, all things considered you can exploit Spark Streaming.

35. Can We Do Real-time Processing Using Spark Sql?

Ans:

Not directly but we can register an existing RDD as a SQL table and trigger SQL queries on top of that.

36. What Is Spark Sql?

Ans:

SQL Spark, better known as Shark is a novel module introduced in Spark to work with structured data and perform structured data processing. Through this module, Spark executes relational SQL queries on the data. The core of the component supports an altogether different RDD called SchemaRDD, composed of rows objects and schema objects defining data type of each column in the row. It is similar to a table in a relational database.

37. What Is A Parquet File?

Ans:

Parquet is a columnar format file supported by many other data processing systems. Spark SQL performs both read and write operations with Parquet files and considers it to be one of the best big data analytics formats so far.

38. List The Functions Of Spark Sql.

Ans:

Spark SQL is capable of:

- Loading data from a variety of structured sources
- Querying data using SQL statements, both inside a Spark program and from external tools that connect to Spark SQL through standard database connectors (JDBC/ODBC). For instance, using business intelligence tools like Tableau
- Providing rich integration between SQL and regular Python/Java/Scala code, including the ability to join RDDs and SQL tables, expose custom functions in SQL, and more

39. What Is Spark?

Ans:

Spark is a parallel data processing framework. It allows developers to develop fast, unified big data applications that combine batch, streaming and interactive analytics.

40. What Is Hive On Spark?

Ans:

- Hive is a component of Hortonworks' Data Platform (HDP). Hive provides an SQL-like interface to data stored in the HDP. Spark users will automatically get the complete set of Hive's rich features, including any new features that Hive might introduce in the future.
- The main task around implementing the Spark execution engine for Hive lies in query planning, where Hive operator plans from the semantic analyzer which is translated to a task plan that Spark can execute. It also includes query execution, where the generated Spark plan gets actually executed in the Spark cluster.

41. What Is A "Parquet" In Spark?

Ans:

"Parquet" is a columnar format file supported by many data processing systems. Spark SQL performs both read and write operations with the "Parquet" file.

42. What Are Benefits Of Spark Over Mapreduce?

Ans:

Due to the availability of in-memory processing, Spark implements the processing around 10-100x faster than Hadoop MapReduce. MapReduce makes use of persistence storage for any of the data processing tasks.

- Unlike Hadoop, Spark provides in-built libraries to perform multiple tasks form the same core like batch processing, Steaming, Machine learning, Interactive SQL queries. However, Hadoop only supports batch processing.
- Hadoop is highly disk-dependent whereas Spark promotes caching and in-memory data storage
- Spark is capable of performing computations multiple times on the same dataset. This is called iterative computation while there is no iterative computing implemented by Hadoop.

43. How Spark Sql Is Different From Hql And Sql?

Ans:

SparkSQL is a special component on the spark Core engine that supports SQL and Hive Query Language without changing any syntax. It's possible to join SQL table and HQL table.

44. What do you understand about SchemaRDD in Apache Spark RDD?

Ans:

- *SchemaRDD* is an RDD that consists of row objects (wrappers around the basic string or integer arrays) with schema information about the type of data in each column.
- SchemaRDD was designed as an attempt to make life easier for developers in their daily routines of code debugging and unit testing on SparkSQL core module. The idea can boil down to describing the data structures inside RDD using a formal description similar to the relational database schema. On top of all basic functions provided by common RDD APIs, SchemaRDD also provides some straightforward relational query interface functions that are realized through SparkSQL.
- Now, it is officially renamed to *DataFrame API* on Spark's latest trunk.

45. How is Spark SQL different from HQL and SQL?

Ans:

Spark SQL is a special component on the Spark Core engine that supports SQL and Hive Query Language without changing any syntax. It is possible to join SQL table and HQL table to Spark SQL.

46. What is a DStream in Apache Spark?

Ans:

- *Discretized Stream* (DStream) is the basic abstraction provided by Spark Streaming. It is a continuous stream of data. It is received from a data source or from a processed data stream generated by transforming the input stream. Internally, a DStream is represented by a continuous

series of RDDs and each RDD contains data from a certain interval. Any operation applied on a DStream translates to operations on the underlying RDDs.

DStreams can be created from various sources like Apache Kafka, HDFS, and Apache Flume.

DStreams have two operations:

- Transformations that produce a new DStream.
- Output operations that write data to an external system.

There are many DStream transformations possible in Spark Streaming. Let us look at filter(*func*). filter(*func*) returns a new DStream by selecting only the records of the source DStream on which *func* returns true.

47. When running Spark applications, is it necessary to install Spark on all the nodes of the YARN cluster?

Ans:

Spark need not be installed when running a job under YARN or Mesos because Spark can execute on top of YARN or Mesos clusters without affecting any change to the cluster.

48. What are the various data sources available in Spark SQL?

Ans:

Parquet file, JSON datasets and Hive tables are the data sources available in Spark SQL.

49. What are the various levels of persistence in Apache Spark?

Ans:

Apache Spark automatically persists the intermediary data from various shuffle operations, however, it is often suggested that users call persist () method on the RDD in case they plan to reuse it. Spark has various persistence levels to store the RDDs on disk or in memory or as a combination of both with different replication levels.

The various storage/persistence levels in Spark are:

- **MEMORY_ONLY:** Store RDD as deserialized Java objects in the JVM. If the RDD does not fit in memory, some partitions will not be cached and will be recomputed on the fly each time they're needed. This is the default level.
- **MEMORY_AND_DISK:** Store RDD as deserialized Java objects in the JVM. If the RDD does not fit in memory, store the partitions that don't fit on disk, and read them from there when they're needed.
- **MEMORY_ONLY_SER:** Store RDD as *serialized* Java objects (one byte array per partition).
- **MEMORY_AND_DISK_SER:** Similar to MEMORY_ONLY_SER, but spill partitions that don't fit in memory to disk instead of recomputing them on the fly each time they're needed.
- **DISK_ONLY:** Store the RDD partitions only on disk.
- **OFF_HEAP:** Similar to MEMORY_ONLY_SER, but store the data in off-heap memory.

50. How Spark uses Akka?

Ans:

Spark uses Akka basically for scheduling. All the workers request for a task to master after registering. The master just assigns the task. Here Spark uses Akka for messaging between the workers and masters.

51. What do you understand by Lazy Evaluation?

Spark is intellectual in the manner in which it operates on data. When you tell Spark to operate on a given dataset, it heeds the instructions and makes a note of it, so that it does not forget – but it does nothing, unless asked for the final result. When a transformation like map*()* is called on an RDD, the operation is not performed immediately. Transformations in Spark are not evaluated till you perform an action. This helps optimize the overall data processing workflow.

52. How is AI executed in Spark?

Ans:

MLlib is an adaptable AI library given by Spark. It goes for making AI simple and adaptable with normal learning calculations and use cases like bunching, relapse separating, dimensional decrease, and alike.

53. What is Spark Executor?

Ans:

At the point when SparkContext associates with a group chief, it obtains an Executor on hubs in the bunch. Representatives are Spark forms that run controls and store the information on the laborer hub. The last assignments by SparkContext are moved to agents for their execution.

54. Name kinds of Cluster Managers in Spark.

Ans:

The Spark system underpins three noteworthy sorts of Cluster Managers:

- **Standalone:** An essential administrator to set up a group.
- **Apache Mesos:** Generalized/regularly utilized group administrator, additionally runs Hadoop MapReduce and different applications.
- **YARN:** Responsible for asset the board in Hadoop.

55. Show some utilization situations where Spark beats Hadoop in preparing.

Ans:

- Sensor Data Processing: Apache Spark's "In-memory" figuring works best here, as information is recovered and joined from various sources.
- Real Time Processing: Spark is favored over Hadoop for constant questioning of information. for example Securities exchange Analysis, Banking, Healthcare, Telecommunications, and so on.
- Stream Processing: For preparing logs and identifying cheats in live streams for cautions, Apache Spark is the best arrangement.
- Big Data Processing: Spark runs upto multiple times quicker than Hadoop with regards to preparing medium and enormous estimated datasets.

56. By what method can Spark be associated with Apache Mesos?

Ans:

To associate Spark with Mesos:

- Configure the sparkle driver program to associate with Mesos.

- Spark paired bundle ought to be in an area open by Mesos.
- Install Apache Spark in a similar area as that of Apache Mesos and design the property ‘spark.mesos.executor.home’ to point to the area where it is introduced.

57. How is Spark SQL not the same as HQL and SQL?

Ans:

Flash SQL is a unique segment on the Spark Core motor that supports SQL and Hive Query Language without changing any sentence structure. It is conceivable to join SQL table and HQL table to Spark SQL.

58. What is ancestry in Spark? How adaptation to internal failure is accomplished in Spark utilizing Lineage Graph?

Ans:

- At whatever point a progression of changes are performed on a RDD, they are not assessed promptly, however languidly.
- At the point when another RDD has been made from a current RDD every one of the conditions between the RDDs will be signed in a diagram.
- This chart is known as the ancestry diagram.
- Consider the underneath situation

Ancestry chart of every one of these activities resembles:

- First RDD
- Second RDD (applying map)
- Third RDD (applying channel)
- Fourth RDD (applying check)

This heredity diagram will be helpful on the off chance that if any of the segments of information is lost.

- Need to set spark.logLineage to be consistent with empowering the Rdd.toDebugString() gets empowered to print the chart logs.

59. What is the contrast between RDD , DataFrame and DataSets?

Ans:

RDD :

- It is the structure square of Spark. All Dataframes or Dataset is inside RDDs.
- It is lethargically assessed permanent gathering objects
- RDDS can be effectively reserved if a similar arrangement of information should be recomputed.

DataFrame :

- Gives the construction see (lines and segments). It tends to be thought of as a table in a database.
- Like RDD even the dataframe is sluggishly assessed.
- It offers colossal execution due to a.) Custom Memory Management – Data is put away in off load memory in twofold arrangement .No refuse accumulation because of this.
- **Optimized Execution Plan –** Query plans are made utilizing Catalyst analyzer.
- **DataFrame Limitations :** Compile Time wellbeing , i.e no control of information is conceivable when the structure isn't known.
- **DataSet :** Expansion of DataFrame
- DataSet Features – Provides best encoding component and not at all like information edges supports arrange time security.

60. What is DStream?

Ans:

- Discretized Stream (DStream)
- Apache Spark Discretized Stream is a gathering of RDDS in grouping .
- Essentially, it speaks to a flood of information or gathering of Rdds separated into little clusters. In addition, DStreams are based on Spark RDDs, Spark's center information reflection. It likewise enables Streaming to flawlessly coordinate with some other Apache Spark segments. For example, Spark MLlib and Spark SQL.

61. What is the connection between Job, Task, Stage ?

Ans:

- **Errand**

  An errand is a unit of work that is sent to the agent. Each stage has some assignment, one undertaking for every segment. The Same assignment is done over various segments of RDD.

- **Occupation**

  The activity is a parallel calculation consisting of numerous undertakings that get produced in light of activities in Apache Spark.

- **Stage**

  Each activity gets isolated into littler arrangements of assignments considered stages that rely upon one another. Stages are named computational limits. All calculation is impossible in a single stage. It is accomplished over numerous stages.

62. Clarify quickly about the parts of Spark Architecture?

Ans:

Flash Driver: The Spark driver is the procedure running the sparkle setting . This driver is in charge of changing over the application to a guided diagram of individual strides to execute on the bunch. There is one driver for each application.

63. How might you limit information moves when working with Spark?

Ans:

- The different manners by which information moves can be limited when working with Apache Spark are:
- Communicate and Accumulator factors

64. When running Spark applications, is it important to introduce Spark on every one of the hubs of YARN group?

Ans:

Flash need not be introduced when running a vocation under YARN or Mesos in light of the fact that Spark can execute over YARN or Mesos bunches without influencing any change to the group.

65. Which one will you decide for an undertaking – Hadoop MapReduce or Apache Spark?

Ans:

The response to this inquiry relies upon the given undertaking situation – as it is realized that Spark utilizes memory rather than system and plate I/O. In any case, Spark utilizes enormous measure of RAM and requires devoted machines to create viable outcomes. So the choice to utilize Hadoop or Spark changes powerfully with the necessities of the venture and spending plan of the association.

66. What is the distinction among continue() and store()

Ans:

endure () enables the client to determine the capacity level while reserve () utilizes the default stockpiling level.

67. What are the different dimensions of constancy in Apache Spark?

Ans:

Apache Spark naturally endures the mediator information from different mix tasks, anyway it is regularly proposed that clients call persevere () technique on the RDD on the off chance that they intend to reuse it. Sparkle has different tirelessness levels to store the RDDs on circle or in memory or as a mix of both with various replication levels.

68. What are the disservices of utilizing Apache Spark over Hadoop MapReduce?

Ans:

Apache Spark's in-memory ability now and again comes a noteworthy barrier for cost effective preparing of huge information. Likewise, Spark has its own record the board framework and consequently should be incorporated with other cloud based information stages or apache hadoop.

69. What is the upside of Spark apathetic assessment?

Ans:

Apache Spark utilizes sluggish assessment all together the advantages:

- Applying Transformations tasks on RDD or "stacking information into RDD" isn't executed quickly until it sees an activity. Changes on RDDs and putting away information in RDD are languidly assessed. Assets will be used in a superior manner if Spark utilizes sluggish assessment.

- Lazy assessment advances the plate and memory utilization in Spark.
- The activities are activated just when the information is required. It diminishes overhead.

70. What are the advantages of Spark over MapReduce?

Ans:

- Because of the accessibility of in-memory handling, Spark executes the preparing around 10 to multiple times quicker than Hadoop MapReduce while MapReduce utilizes diligence stockpiling for any of the information handling errands.
- Dissimilar to Hadoop, Spark gives inbuilt libraries to play out numerous errands from a similar center like cluster preparing, Steaming, Machine learning, Interactive SQL inquiries. Be that as it may, Hadoop just backings cluster handling.
- Hadoop is very plate subordinate while Spark advances reserving and in-memory information stockpiling.

71. How DAG functions in Spark?

Ans:

- At the point when an Action is approached Spark RDD at an abnormal state, Spark presents the heredity chart to the DAG Scheduler.
- Activities are separated into phases of the errand in the DAG Scheduler. A phase contains errands dependent on the parcel of the info information. The DAG scheduler pipelines administrators together. It dispatches tasks through the group chief. The conditions of stages are obscure to the errand scheduler.The Workers execute the undertaking on the slave.

72. What is the hugeness of the Sliding Window task?

Ans:

Sliding Window controls transmission of information bundles between different PC systems. Sparkle Streaming library gives windowed calculations where the changes on RDDs are connected over a sliding window of information. At whatever point the window slides, the RDDs that fall inside the specific window are consolidated and worked upon to create new RDDs of the windowed DStream.

73. What are communicated and Accumulators?

Ans:

- **Communicate variable:**

  On the off chance that we have an enormous dataset, rather than moving a duplicate of informational collection for each assignment, we can utilize a communication variable which can be replicated to every hub at one time and share similar information for each errand in that hub. Communicate variable assistance to give a huge informational collection to every hub.

- **Collector:**

  Flash capacities utilized factors characterized in the driver program and nearby replicated factors will be produced. Aggregators are shared factors which help to refresh factors in parallel during execution and offer the outcomes from specialists to the driver.

74. What are activities ?

Ans:

An activity helps in bringing back the information from RDD to the nearby machine. An activity's execution is the aftereffect of all recently made changes. lessen() is an activity that executes the capacity passed over and over until one esteem assuming left. take() move makes every one of the qualities from RDD to nearby hub.

75. What is YARN?

Ans:

Like Hadoop, YARN is one of the key highlights in Spark, giving a focal and asset the executives stage to convey adaptable activities over the bunch. YARN is a conveyed holder chief, as Mesos for instance, while Spark is an information preparing instrument. Sparkle can keep running on YARN, a similar way Hadoop Map Reduce can keep running on YARN. Running Spark on YARN requires a double dispersion of Spark as based on YARN support.

76. Clarify the key highlights of Apache Spark.

Ans:

Coming up next are the key highlights of Apache Spark:

- Polyglot
- Speed
- Multiple Format Support
- Lazy Evaluation
- Hadoop Integration
- Machine Learning

77. How is Streaming executed in Spark? Clarify with precedents.

Ans:

Sparkle Streaming is utilized for handling constant gushing information. Along these lines it is a helpful expansion into the Spark API. It empowers high-throughput and shortcoming tolerant stream handling of live information streams. The crucial stream unit is DStream which is fundamentally a progression of RDDs (Resilient Distributed Datasets) to process the constant information. The information from various sources like Flume, HDFS is spilled lastly to document frameworks, live

dashboards and databases. It is like a bunch preparing as the information is partitioned into streams like clusters.

78. What are the enhancements that engineers can make while working with flash?

Ans:

- Flash is memory serious, whatever you do it does in memory.
- Initially, you can alter to what extent flash will hold up before it times out on every one of the periods of information region information neighborhood process nearby hub nearby rack neighborhood Any.
- Channel out information as ahead of schedule as could be allowed. For reserving, pick carefully from different capacity levels.
- Tune the quantity of parcels in sparkle.

79. List some use cases where Spark outperforms Hadoop in processing.

Ans:

- Sensor Data Processing: Apache Spark's "In-memory" computing works best here, as data is retrieved and combined from different sources.
- Real Time Processing: Spark is preferred over Hadoop for real-time querying of data. e.g. Stock Market Analysis, Banking, Healthcare, Telecommunications, etc.
- Stream Processing: For processing logs and detecting frauds in live streams for alerts, Apache Spark is the best solution.
- Big Data Processing: Spark runs upto 100 times faster than Hadoop when it comes to processing medium and large-sized datasets.

80. What is a Data Frame?

Ans:

An information casing resembles a table, it got some named sections which are composed into segments. You can make an information outline from a document or from tables in hive, outside databases SQL or NoSQL or existing RDD's. It is practically equivalent to a table.

81. How might you associate Hive to Spark SQL?

Ans:

- The principal significant thing is that you need to place the hive-site.xml record in the conf index of Spark.
- At that point with the assistance of Spark session object we can develop an information outline as,

82. What is GraphX?

Ans:

- Ordinarily you need to process the information as charts, since you need to do some examination on it. It endeavors to perform Graph calculation in Spark in which information is available in documents or in RDD's.
- GraphX is based on the highest point of Spark center, so it has got every one of the abilities of Apache Spark like adaptation to internal failure, scaling and there are numerous inbuilt chart calculations too. GraphX binds together ETL, exploratory investigation and iterative diagram calculation inside a solitary framework.
- You can see indistinguishable information from the two charts and accumulations, change and unite diagrams with RDD effectively and compose custom iterative calculations utilizing the pregel API.
- GraphX contends on execution with the quickest diagram frameworks while holding Spark's adaptability, adaptation to internal failure and convenience.