Top 99 Apache Spark Interview Questions and Answers 2021

Q1) What is Apache Spark?

Answer: Spark is an in-memory parallel data processing framework. It support batch, streaming processing also interactive analytic.

A parallel data framework for real-time processing that can be suitable for a wide range of circumstances is called Apache Spark. It can some programming languages like Python, Java, R, and Scala. The data scientists always incorporate the Spark in the applications to transform and analyze the data at scale.

The most frequently associated tasks with Spark include processing of streaming data from the Internet of Things and sensors. It is also known as the third generation data processing platform that can unify the problems of big data processing such as interacting, batch and streaming processing. Apache Spark is used for interactive data analytics, sensor data processing, iterative machine learning, and stream processing.

Q2) What are the three ways to create RDD in Spark?

Answer: The three ways to create RDD in Spark is:

. By using parallelized collection

. By loading an external dataset

. From an existing RDD

Q3) Can we create RDD from existing RDD?

Answer: Yes, by applying transformations on RDD we can create new RDD.

Q4) In how many ways we can create RDD?

Answer: There are three possible ways to create RDD.

Q5) Can we create RDD using Dataset like .txt file?

Answer: Yes, by loading dataset we can create RDD.

Q6) Can we run Spark without using HDFS?

Answer: Yes, we need HDFS just for storage purpose.

Q7) Does spark support stand alone mode?

Answer: Yes it supports standalone mode

Q8) What are the types of transformations in Spark?

Answer: Narrow and Wide Transformation are available in Spark.

Q9) Give some example for Narrow transformation?

Answer: Map and Filter.

Q10) Give some example of wide transformations?

Answer: GroupByKey and ReduceByKey

Q11) What are the Components in Spark?

Answer: Spark SQL, Spark Streaming, Mlib And Graph X

Q12) What is Spark SQL?

Answer: It is a component of Spark which provides support for structured and semi-structured data. Data Frame appeared in Spark Release 1.3.0

Q13) What is Spark Dataset?

Answer: Dataset is an extension of Data Frame API which provides type-safe, object-oriented programming interface.

Q14) What are the limitations of Data frame?

Answer: Data Frame does not have provision for compile-time type safety.

Q15) What is the transformation in Spark?

Answer: Transformation is a function that produces new RDD from the existing RDDs. It takes RDD as input and produces output as one or more RDD.

Q16) What is Action in Spark?

Answer: Actions return final results of RDD computations. It triggers execution using lineage graph and after carry out all intermediate transformations return the final results to the Driver.

Q17) Give some examples of Transformation in Spark?

Answer: Map, flatmap and filter.

Q18) Give some example of Action in Spark?

Answer: Count(),Collect() and reduce(func).

Q19) What collect does in Spark?

Answer: It returns all the elements in the RDD to Driver.

Q20) What are the different storage levels in Spark?

Answer: Memory_Only, MEMORY_AND_DISK, MEMORY_ONLY_SER , MEMORY_AND_DISK_SER & DISK_Only

Q21) What is the default storage level in Spark?

Answer: Memory_Only

Q22) What is coalesce ()?

Answer: It is used to decrease the number of partitions in an RDD. It avoids full shuffle of RDD.

Q23) What is re-partition ()?

Answer: It is used to increase the number of Partitions. It creates a new partition from the existing partition by shuffling of data.

Q24) What is the role of Driver in Spark?

Answer: The driver is the program which creates the Spark Context, connecting to a given Spark Master. It declares the transformations and actions on RDDs and submits such requests to the master.

Q25) What are the deployment modes in Spark?

Answer: Cluster mode and Client Mode

Q26)Why Spark is faster than Hadoop?

Answer: Spark is faster than Hadoop because it does processing in memory.

Q27) What is accumulator?

Answer: An accumulator is a shared variable which is used for aggregating information across the cluster.

Q28) What are the two types of shared variable available in Apache Spark?

Answer: Broadcast Variable and Accumulator.

Q29) What is the Broadcast variable?

Answer: It allows the programmer to keep a read-only variable cached on each machine instead of shipping a copy of it with tasks.

Q30) What is Spark D Stream?

Answer: Spark DStream is the basic abstraction of Spark Streaming. It is a continuous stream of data.

Q31) What is map transformation in Spark?

Answer: Map transformation takes a function as input after applying that function to each RDD return another RDD. Its return type can be different from its input type.

Q32) What is a flatmap transformation in Spark?

Answer: Flatmap is used when we want to produce multiple elements for each input element. The output of the flatmap is a List of the element through which we can iterate.

Q33) What is action Reduce in Spark?

Answer: Reduce takes a function as an input which has two parameters which are of same type and output a single value of the input type.

Q34) What is lazy evaluation in Spark?

Answer: When we apply transformation on RDD it does not immediately gives output it will make DAG of all transformation. Transformations in Spark are evaluated after you perform an action. This is called Lazy Evaluation.

Q35) What is MLib in Spark?

Answer: MLlib is a distributed machine learning framework built on top of Spark.

Q36) What is GraphX in Spark?

Answer: GraphX is a distributed graph-processing framework built top of Spark. It provides different APIs for expressing graph computation.

Q37) What is spark shell?

Answer: Spark Shell is a Spark Application which is written in Scala. It offers a command line environment with auto-completion.
which is helpful in developing our own Standalone Spark Application.

Q38) Write some function of Spark Context?

Answer: Used to create Spark RDDs, accumulators, and broadcast variables, access all Spark services and run jobs also to get the status of spark application. Starting and cancelling of Job etc.

Q39) Write some function of Spark Executor?

Answer: To run a task that makeup application and to return the result to Driver. It Provides in-memory storage for RDDs cached by user.

Q40) Which are the Programming languages supported by Spark?

Answer: Java, Python, Scala, SQL and R.

Q41) What is DAG in Spark?

Answer: DAG is a set of Vertices and Edges, where vertices represent the RDDs and the edges represent the Operation to be applied on RDD.

Q42) What is Caching in Spark Streaming?

Answer: Caching Streaming is storing streaming RDD in memory. It is a mechanism to speed up applications that access the same RDD multiple times.

Q43) Which file systems are supported by Spark?

Answer: HDFS, Local File system & Amazon S3.

Q44) What is RDD?

Answer: Resilient Distribution Datasets (RDD) is a fault-tolerant collection of partitioned data that run in parallel. RDD is immutable and distributed in nature.

Q45) Write some input sources for Spark Streaming.

Answer: TCP Sockets, Stream of files, Apache Kafka, Apache Flume, Kinesis etc.

Q46) Can we use Hive on Spark?

Answer: Yes, by creating Hive context

Q47) What is a pipe () operation in Spark?

Answer: Spark is using Scala, Java, and Python to write the program. However, if one wants to pipe (inject) the data
which is written in other languages Spark provides a general mechanism for that in the form of pipe() method.

Q48) What are the data sources available in Spark SQL?

Answer: Parquet, Avro, JSON and Hive tables

Q49) What is a partition in Spark?

Answer: A partition in spark is a logical division of data stored on a Node in the cluster. Partitions are basic units of parallelism in Apache Spark.

Q50) What are the types of Partitioning in Apache Spark?

Answer: The types of portioning in Apache Spark are as follows:

- Hash Partitioning
- Range Partitioning

Q51) What are the types of Cluster managers in Spark?

Answer: Standalone,Yarn & Mesos.

Q52) Comparing Hadoop?

Answer: Comparing Hadoop it is reliable fast and easily process highly difficult data

Q53) what is Hive metastore and where it is stored.?

Answer: Hive Metastore is the schemas of the hsql data definitions of underlying data for Hive benches.

Q54) Hadoop MapReduce vs spark MapReduce?

Answer:

- It does multi retention and dispensation the data through map reduce
- spark extracts the data by overlaying in yarn environment

Q55) RDD how It's work?

Answer: Resilient Distribution Datasets which runs parallel for fault tolerance It is prepared by recipes through dual tactics adapting Spark Framework's parallelize

Q56) Elements in spark?

Answer:

- Spark Core: Vile locomotive for large-scale parallel and dispersed data dispensation
- Spark Streaming: Castoff IN dispensation real-time flowing data
- Spark SQL: Integrates interpersonal dispensation with Spark's functional programming API
- GraphX: Graphs and graph-parallel reckoning
- MLlib: Completes machine erudition trendy Apache Spark

Q57) what are the different storage formats you have used in your project and compression techniques.?

Answer: Csv,json

Q58) Dstream in spark?

Answer: (DStream) is the emergent broad view on condition that by Spark Streaming

Q59) what is Hive metastore and where it is stored.?

Answer: Hive Metastore is the schemas of the hsql data definitions of underlying data for Hive tables.

Q60) Dstreams catches the data?

Answer: Dstreams is transmittable data and vittles in recollection

Q61)Dataset castoff in spark?

Answer: Json dataset are castoffed in spark

Q62) Difference between RDD and DataFrame?

Answer:

RDD:-

- Optimization – No inbuilt optimization engine is available in RDD
- Serialization- it does so use Java serialization
- Compile-time type safety
- Efficiently process data, which is structured as well as unstructured
- Need to define the schema (manually)

- RDD API is slower to perform simple grouping and aggregation operations

DataFrame :-

- Optimization- Optimization takes place using catalyst optimizer, Analyzing a logical plan, Logical plan, Physical planning and Code generation to compile java bytecode
- Serialization– it uses off-heap storage (in memory) in binary format
- Run-time type validation
- Efficiently process data, which is structured as well as semi-structured
- Shema is automatically defined
- DataFrame API is slower to perform simple grouping and aggregation operations

Q63)How to convert RDD to DataFrame?

Answer:

```
case class Customers(custid: Int,cname: String,lname: String)
valpeopleDF =
spark.sparkContext.textFile("D:/Hadoop/Spark/SparkScala/customer_data.csv")
.map(_.split(","))
.map(attribute => Customers(attribute(0).toInt,attribute(1),attribute(2)))
.toDF()
```

Q64) How to programmatically specifying schema for DataFrame?

Answer:

```
valschemaMap = List("id","name","salary").map(field =>StructField(field,StringType,true))
val schema    = StructType(schemaMap)
```

Q65) How to remove Special character "#" from 100 of columns in DataFrame?

```
val columns = "#cust_id|#cust_name| #odr_date|   #shipdt|  #Courer| #recvd_dt|#returned or
not|#returned dt|#reson of return"
.split('|').map(_.stripMargin('#'))
valcreateNewDF = createDF.toDF(columns:_*)
```

Q66) Load a csv/textFile and remove header and footer?

```
valheaderFooterRemovedDF = loadDF.take(loadDF.count.toInt).drop(1).dropRight(1)
valschemaDefine = "id|name|date|type|status".split('|').map(col
=>StructField(col.toString,StringType,true))
val schema = StructType(schemaDefine)
valfinalDF =
spark.createDataFrame(spark.sparkContext.parallelize(headerFooterRemovedDF),schema)
```

Q67) Take first 10 record and last 10 record of file and combine both using DataFrame?

```
val loadDF =
spark.read.format("csv").option("path","file:///home/maria_dev/Files/assignment_table.csv").
load()
val combineDF = loadDF.take(10) ++ loadDF.take(loadDF.count.toInt).takeRight(10)
val schemaDesign = loadDF.first.toSeq.map(c => c.toString.trim).map(col =>
StructField(col,StringType,true))
val schema = StructType(schemaDesign)
val createDF = spark.createDataFrame(spark.sparkContext.parallelize(combineDF),schema)
createDF.show()
```

Q68) How to calculate executor memory?

Answer:

Configuration of the cluster is as below :

Nodes = 10

Each Node has core = 16 cores (-1 for operating systems)

Each Node Ram = 61 GB Ram (-1 for Hadoop Deamons)

Number of cores identification:

Number of cores is, number of concurrent tasks an executor can run in parallel so the general rule of thumb for optimal value is 5 (–num-cores 5)

Number of executor identification :

No.of.executor = No.of.cores / concurrent tasks (5 in general)

15/5 = 3 is no.of.executor in each node

No.of.nodes * no.of.executor in each node = no.of.executor (for spark job)

10 * 3 = 30 (-–num-executors 30)

Q69) Definition of “Spark SQL”?

Answer: Spark SQL is a Spark interface to operate with structured as great as the semi-structured data value. It should this ability to place data value of various structured data specialists like “text files value”, JSON files value, Parquet files value, among other data.

Q70) What is the name of a few commonly used Spark Ecosystems?

Answer:

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

Q71) What is meant by “Parquet fie”?

Answer: Parquet is defined by a columnar format file supported many of data value system processing. Spark SQL has been performing both of the read and write data operations function with Parquet file it’s supposed to be one of the best high data analytics formats so greatly.

Q72) Defined by Catalyst Framework?

Answer: Catalyst framework defined as a unique optimization dataset system framework present in “Spark SQL”. It allows Spark SQL catalyst framework data value has to automatically modify the SQL data value queries by adding new optimizations to data to produce a faster data processing system.

Q73) How do using BlinkDB?

Answer: BlinkDB is a query engine transfers the data for producing interactive data system SQL queries about huge numbers of data value including renders difficulty returns identified including significant error bars. BlinkDB helps users data balance ‘query accuracy’.

Q74) What are the various data sources available in Parquet file JSON Datasets Hive tables?

Answer:

- Parquet file
- JSON Datasets
- Hive tables

Q75) Different SparkSQL from HQL & SQL?

Answer: SparkSQL is a unique element information use on the spark Core engine that executes SQL including Hive Query Language of destroying any syntax. It's now to join the SQL report table and HQL table.

Q76) What does a Spark Engine do?

Answer: Spark Engine is held for scheduling, distributing and monitoring the data application across the cluster.

Q77)What did operations support for RDD?

Answer:

- Transformations.
- Actions

Q78)What are the file systems support for Spark?

Answer:

- Hadoop Distributed File System (HDFS).
- Local File system.
- S3

Q79) What are the features of Apache Spark?

Answer: The following are the important features of Apache Spark:

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

- Real-Time Computation
- Machine Learning

Q80) What is meant by RDD?

Answer: Resilient Distribution Datasets (RDD) is an operational element fault-tolerant collection which satisfies the properties like distributed, catchable, immutable, etc..,

There are two types of RDD:

- Hadoop datasets
- Parallelized Collections

Q81) How to create RDDs in Spark?

Answer: There are two methods to create RDD:

They are:

Parallelizing driver program collection and this can be used SparkContext's 'parallelize' method val Array = Array(3,6,9)

val RDD = sc.parallelize(Array)

By loading dataset from the HBase, HDFS and other external storages.

Q82) What is YARN?

Answer: YARN is one of the important features in Spark and it is very much similar to Hadoop that provides a resource management platform to deliver operations that are scalable across the cluster. YARN is termed as the distributed container manager and Spark as the data processing tool. Both the Spark and Hadoop MapReduce can run on YARN. Spark can run independently from its installation process. There is no need to install Spark on YARN cluster nodes because it runs on the top of YARN.

Q83) What are the common Ecosystems in Spark?

Answer:

- Spark Streaming for streaming data
- Spark SQL (Shark)- for developers
- GraphX for Graph computation
- Machine Learning Algorithms (MLlib)

- SparkR to run R programming in a Spark engine

Q84) What are the differences between Spark and Hadoop?

Answer:

Parameter	Apache Spark	Hadoop
Processing	It can support both Batch processing & Real-time processing	Only Batch processing
Recovery	It can allow partitions recovery	Fault-tolerant
Speed	Faster than Hadoop	Maintain decent speed
Difficulty	Very easy to learn when compared to Hadoop because pf high-level modules	Difficult to learn
Interactivity	It has interactive modes	No interactive modes

Q85) Explain about partitions in Apache Spark?

Answer: As the name indicates partition means a logical and smaller division of data which is similar to the 'split' in MapReduce. It is the process to derive data logical units to speed up the process. Spark can manage the partitions to minimize the network traffic between the executors and also it can read data from the nodes into RDD. Partitions can also be known as the data set in the large distributed chunk and it can be used to optimize the operations to hold chunks. Everything in the Spark can be performed through partition RDD.

Q86) Define Spark Streaming?

Answer: Spark Streaming is very much useful for streaming data and also it enables fault-tolerant and high-throughput for live data. DStream is the fundamental unit which is the series of Resilient Distributed Datasets (RDD). The data from HDFS, Flume can be streamed to process them in live dashboards, file systems, and databases. The real-time data from different sources like Geographical systems, Stock market, and Twitter can be streamed by using this Spark streaming process.

Q87) Write about the Actions in Spark?

Answer: The data from the RDD can bring back to the local machine by using Action execution in the Spark. Action execution is the result of the transformations that were created previously. With the help of lineage graph, these Actions can trigger the data execution into original RDD which can have the ability to carry out all the transformations in intermediate and return the final result in Driver program.

Q88) How to minimize data transfers working with Spark?

Answer: The data transfer minimization and shuffling removal can be very much helpful in writing Spark programs that can run reliably. There are different ways of minimizing data transfers with Spark:

They are:

- Using Accumulators
- Using Broadcast Variable
- Avoiding operations by key

Q89) Why we have to use broadcast variables while working with Spark?

Answer: The broadcast variables can be referred to as read-only variables present on every machine(in-memory cache). The broadcast variables usage can eliminate variable ship copy necessity, in this way the data can process at high speed. Storing the lookup table in the memory can be possible through broadcast variables. It is used to enhance the efficiency of retrieval compared to Resilient Distribution Datasets loops.

Q90) What are the functions of SparkCore?

Answer: Spark core is nothing but a space engine for distributed data processing and large-scale parallel process. The Spark core is also known as the distributed execution engine while the Python APIs, Java, and Scala offer ETL application platform. Spark core can perform different functions like monitoring jobs, storage system interactions, job scheduling, memory management, and fault-tolerance. Further, it can allow workload streaming, machine learning, and SQL.

The Spark core can also be responsible for:

- Monitoring, Scheduling, and distributing jobs on a cluster
- Fault recovery and memory management

- Ecosystems interactions

Q91) What is meant by Parquet file?

Answer: A columnar format file that can support different data processing systems is known as Parquet. Both read and write operations can be performed by Spark SQL with the help of the Parquet file. It can be considered as one of the biggest data analytics formats.

Many data processing systems support this Parquet columnar format because of the advantages that it has. The following are the advantages of columnar storage:

Columnar storage can fetch specific columns which the user wants to access

- It can also give good summarized data
- It has the limit of 10 operations
- Space consumption is very less in columnar storage

Q92) What are the functions of Spark SQL?

Answer: Spark SQL supports HiveQueryLanguage and SQL without syntax changes on the Sparkcore engine. It is a new model in Spark which integrates functional programming API of Spark with relational processing. It can support querying data either via Hive Query Language or via SQL.

The Spark SQL has the capability of querying the data by using the Spark program and SQL statements. It can also provide rich interactions between regular Java/Python/Scala code and SQL. Data loading from various structured sources can be possible through Spark SQL.

Q93) What are the advantages of Spark over Hadoop MapReduce?

Answer: Spark is 100 times faster than the Hadoop MapReduce in terms of memory and RAM can be utilized to get faster results.

The MapReduce mechanism is very time taking process because the user can write many tasks and tie these tasks using shell/Oozie script.

Always, it is a problem in MapReduce in translating MP output into the input of another MP may require different code. It is because the Oozie script didn't support.

The user can do everything in Spark by using a single console and can get the output immediately. Switching between 'Doing something locally' and 'Running something on cluster' is very easy. It leads to more productivity.

Q94) How can we use Spark as the alongside Hadoop?

Answer: Apache Spark has the capability to compatible with Hadoop. It leads to technologies as very powerful combinations.

The components of Hadoop can be used alongside spark in different ways.

They are:

MapReduce: In the same Hadoop cluster the Spark can be used with MapReduce as a framework processing.

Real-Time & Batch Processing: Both MapReduce & Spark can be used together. Of those Spark is used for real-time processing and MapReduce is used for batch processing.

HDFS: The Spark ability is to run on HDFS top to level the replicated storage.

YARN: YARN is termed as the next generation of Hadoop in which we can run the Spark applications.

Q95) Define Worker node?

Answer: The node which can able to run the application even in the cluster is known as a Worker node. The driver program must accept incoming connections and listen from its executors which can be addressable. Worker nodes can be referred to the slave nodes because the worker node's work can be assigned by the master node. The processing of data can be done by worker node and it reports to the master node. The master node always schedules the tasks.

Q96) What are the demerits of Spark?

Answer: The demerits of Spark are:

Developers are always careful while running Spark applications.

There is a possibility of having a bottleneck in the cost-efficient big data process with the capability of Spark in-memory.

Spark uses more storage space when compared to MapReduce and Hadoop. It leads to certain problems.

The work should distribute over multiple clusters instead of running on a single node.

A huge amount of data can be consumed by Spark when compared to Hadoop.

Q97) How to connect Spark with Apache Mesos?

Answer: The following are the steps to connect Spark with Apache Mesos:

Initially, configure the program of Spark driver to connect Mesos.

The binary package of Spark should be in a position to accessible by Mesos.

Then finally, in the location of Mesos, we have to install the Spark and configure the property by pointing to the location.

Q98) What is meany by DStream in Spark?

Answer: A continuous data stream and an abstraction that is provided by Spark streaming are known as DStream. It can receive from the data stream processing or from the data source which was generated by the stream input. The internal structure of DStream can be represented by Resilient Distribution Datasets continuous series. The DStream operations are translated to underlying RDD operations.

The user can create these DStreams in various sources like HDFS, Apache Kafka, and Apache Flume. These DStreams can do two operations:

They are:

These can write the data to the external systems. Hence, it is known as output operation.

New DStream can be produced with the transformations.

Q99) Explain the types of DStream transformations?

Answer:

There are two types of transformations on DStream.

- Stateless Transformations
- Stateful Transformations

Stateless Transformations: The batch processing doesn't depend on the previous output batch.

Example: reduceByKey(), map(), filter().

Stateful Transformations: The batch processing depends on the previous batch intermediate results.

Example: Sliding windows transformations.