Top Answers to Spark Interview Questions

Get Ready to Nail Your Spark Interview

1.Compare MapReduce and Spark

Criteria	MapReduce	Spark
Processing Speeds	Good	Excellent (up to 100 times faster)
Data caching	Hard disk	In-memory
Perform iterative jobs	Average	Excellent
Independent of Hadoop	No	Yes
Machine learning applications	Average	Excellent

2.What is Apache Spark?

Spark is a fast, easy-to-use and flexible data processing framework. It has an advanced execution engine supporting cyclic data flow and in-memory computing. Spark can run on Hadoop, standalone or in the cloud and is capable of accessing diverse data sources including HDFS, HBase, Cassandra and others. Learn more in this Apache Spark Tutorial **.**

3.Explain key features of Spark.

• Allows Integration with Hadoop and files included in HDFS.
• Spark has an interactive language shell as it has an independent Scala (the language in which Spark is written) interpreter
• Spark consists of RDD's (Resilient Distributed Datasets), which can be cached across computing nodes in a cluster.
• Spark supports multiple analytic tools that are used for interactive query analysis , real-time analysis and graph processing

4.Define RDD.

RDD is the acronym for Resilient Distribution Datasets – a fault-tolerant collection of operational elements that run parallel. The partitioned data in RDD is immutable and distributed. There are primarily two types of RDD:
• Parallelized Collections : The existing RDD's running parallel with one another
• Hadoop datasets: perform function on each file record in HDFS or other storage system

5.What does a Spark Engine do?

Spark Engine is responsible for scheduling, distributing and monitoring the data application across the cluster.

6.Define Partitions?

As the name suggests, partition is a smaller and logical division of data similar to 'split' in MapReduce. Partitioning is the process to derive logical units of data to speed up the processing process. Everything in Spark is a partitioned RDD.***Are you interested in learning Apache Spark? Well, find out more in this comprehensive*** *Apache Spark Course* ***to give your career a head start.***

7.What operations RDD support?

• Transformations
• Actions

8.What do you understand by Transformations in Spark?

Transformations are functions applied on RDD, resulting into another RDD. It does not execute until an action occurs. map() and filer() are examples of transformations, where the former applies the function passed to it on each element of RDD and results into another RDD. The filter() creates a new RDD by selecting elements form current RDD that pass function argument.

9. Define Actions.

An action helps in bringing back the data from RDD to the local machine. An action's execution is the result of all previously created transformations. reduce() is an action that implements the function passed again and again until one value if left. take() action takes all the values from RDD to local node.

10.Define functions of SparkCore.

Serving as the base engine, SparkCore performs various important functions like memory management, monitoring jobs, fault-tolerance, job scheduling and interaction with storage systems.

11.What is RDD Lineage?

Spark does not support data replication in the memory and thus, if any data is lost, it is rebuild using RDD lineage. RDD lineage is a process that reconstructs lost data partitions. The best is that RDD always remembers how to build from other datasets.

12.What is Spark Driver?

Spark Driver is the program that runs on the master node of the machine and declares transformations and actions on data RDDs. In simple terms, driver in Spark creates SparkContext, connected to a given Spark Master. The driver also delivers the RDD graphs to Master, where the standalone cluster manager runs.***Are you interested in the comprehensive*** *Apache Spark and Scala Videos* ***to take your career to the next level?***

13.What is Hive on Spark?

Hive contains significant support for Apache Spark, wherein Hive execution is configured to Spark:
hive> set spark.home=/location/to/sparkHome;
hive> set hive.execution.engine=spark;
Hive on Spark supports Spark on yarn mode by default.

14.Name commonly-used Spark Ecosystems.

- Spark SQL (Shark)- for developers
- Spark Streaming for processing live data streams
- GraphX for generating and computing graphs
- MLlib (Machine Learning Algorithms)
- SparkR to promote R Programming in Spark engine.

15.Define Spark Streaming.

Spark supports stream processing – an extension to the Spark API , allowing stream processing of live data streams. The data from different sources like Flume, HDFS is streamed and finally processed to file systems, live dashboards and databases. It is similar to batch processing as the input data is divided into streams like batches.*Learn about the Top Four Apache Spark use cases in this blog post.*

16.What is GraphX?

Spark uses GraphX for graph processing to build and transform interactive graphs. The GraphX component enables programmers to reason about structured data at scale.

17.What does MLlib do?

MLlib is scalable machine learning library provided by Spark. It aims at making machine learning easy and scalable with common learning algorithms and use cases like clustering, regression filtering, dimensional reduction, and alike.*Our in-depth* Scala Certification *Course can give your career a big boost!*

18.What is Spark SQL?

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 relational database.

19.What is a Parquet file?

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

20.What file systems Spark support?

• Hadoop Distributed File System (HDFS). Learn more about HDFS in these Top Interview questions.
• Local File system
• S3

21.What is Yarn?

Similar to Hadoop, Yarn is one of the key features in Spark, providing a central and resource management platform to deliver scalable operations across the cluster . Running Spark on Yarn necessitates a binary distribution of Spar as built on Yarn support.

22.List the functions of Spark SQL.

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. Get to know more about Tableau in this Tableau Tutorial.
• 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.

23.What are benefits of Spark over MapReduce?

• 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.Read more in this blog about the comparison of Spark and MapReduce.

24.Is there any benefit of learning MapReduce, then?

Yes, MapReduce is a paradigm used by many big data tools including Spark as well. It is extremely relevant to use MapReduce when the data grows bigger and bigger. Most tools like Pig and Hive convert their queries into MapReduce phases to optimize them better. Learn more in this MapReduce Tutorial.

25.What is Spark Executor?

When SparkContext connect to a cluster manager, it acquires an Executor on nodes in the cluster. Executors are Spark processes that run computations and store the data on the worker node. The final tasks by SparkContext are transferred to executors for their execution.

26.Name types of Cluster Managers in Spark.

The Spark framework supports three major types of Cluster Managers:
• Standalone: a basic manager to set up a cluster
• Apache Mesos: generalized/commonly-used cluster manager, also runs Hadoop MapReduce and other applications
• Yarn: responsible for resource management in Hadoop

27.What do you understand by worker node?

Worker node refers to any node that can run the application code in a cluster.

28.What is PageRank?

A unique feature and algorithm in graph, PageRank is the measure of each vertex in the graph. For instance, an edge from u to v represents endorsement of v's importance by u. In simple terms, if a user at Instagram is followed massively, it will rank high on that platform.

29.Do you need to install Spark on all nodes of Yarn cluster while running Spark on Yarn?

No because Spark runs on top of Yarn.

30.Illustrate some demerits of using Spark.

Since Spark utilizes more storage space compared to Hadoop and MapReduce, there may arise certain problems. Developers need to be careful while running their applications in Spark. Instead of running everything on a single node, the work must be distributed over multiple clusters.

31.How to create RDD?

Spark provides two methods to create RDD:• By parallelizing a collection in your Driver program. This makes use of SparkContext's 'parallelize' methodval IntellipaatData = Array(2,4,6,8,10) val distIntellipaatData = sc.parallelize(IntellipaatData)• By loading an external dataset from external storage like HDFS, HBase, shared file system.

1.Business Objects Characteristics

Criteria	Result
Querying and Data analysis	Ad-hoc querying & complex data analysis
Total cost of ownership	Low due to integrated, scalable, highly available, BI enterprise platform
Ease of use	Extremely user friendly, with quick up time

2.What do you understand by Business objects?

Business object detect the solutions for business professionals which could be beneficial to gain data from the corporate database directly from the desktop.

3.Write the advantages of using business objects.

There are many advantages of business objects
• Easy to use

- Dragging and dropping
- Graphical interface
- Business terms that are familiar
- Powerful reports for a lesser amount of time.Read this top-rated blog to find out why BusinessObjects is such a sought-after Business Intelligence tool.

4.Write the different products linked with Business Objects.

There are various products related with business objects
- Broadcast Agent
- Info View
- Supervisor
- Auditor
- Designer
- Set Analyzer
- User module

5.What is Designer.

Designer is a set which is linked with Business object IS utilized by designer for developing and maintaining universe .Here universe is the semantic layer which can resolves the problem of end user issues that are technical related to the database.

6.Name the kinds of modes related with designer and business objects?

There are two kinds of modes related with designer
- Workgroup mode
- Enterprise mode***Learn more about Business Objects in this*** *Business Objects Courses* ***to get ahead in your career!***

Wish to Learn Business Objects? Click Here

7.Name different kinds of procedure associated with multidimensional analysis that is inside business objects.

There are two different methods:
- Drill down
- Slice and Dice

8.Write the kinds of users linked with business objects.

There are various kinds of users
- Supervisor Designer
- Designer
- Graphical Interface
- Supervisor
- General Supervisor
- Versatile User
- End User

9. What all data sources are available?

BO allows us to review data from various sources. We can use data from RDBMS such as MS SQL server, Oracle, etc.

10.Write the various kind of data providers?

There are various kinds of data providers are:
- OLAP servers
- Queries over universe
- Personal data files
- SAP
- VBA procedures
- Stored procedures
- Free hand-SQL

11.What do you mean by drill mode?

It is a kind of study mode related to BO and helps in breaching data as well as in presenting data from all the available angles and the levels of details for analyzing the factors which caused the good and bad result.

12.Define personal connection?

It can be made only by one user an can't be used by the others. The details for same are kept in a PDAC.LSI file.

13.Define Shared connection?

It is opposite of personal connection as it can be used by others also through a server which is common one. In this details of connections are kept in SDAC>LSI file which is there in installation folder.

14.Define secured connection?

It is a kind of connection which overcomes the disadvantages related with the former connection. We can use universe in central repository by secured connection.

15.What is custom hierarchies?

It defines the universe for providing the drill down which is customized and occur between objects from same or different classes taking care of the user requirements.

16.How we can create Custom Hierarchies?

We can create custom hierarchies by clicking on tools then choosing hierarchies from BO designer.

17.What do you understand by context in the universe?

Context can be explained as the specific path of join between a specific group of joins or the tables for the purpose of a specific query.

18.How we can create Custom Hierarchies?

We can create custom hierarchies by clicking on tools then choosing hierarchies from BO designer.

19.What is Chasm Trap.

It is a situation which arises at the time when the value in the fact table get wrong when it is measured from two various fact tables.

20.How the problem of Chasm Trap be solved?

It can be solved by two methods:
- By using SQL parameter in universe which generates SQL statement for each measures and result into a correct output.
- The second one is involving two joints in different contexts.

21.Define Derived tables?

They are created in the universe. It is used for complex calculations which can't be sorted out in the report level. The other useful feature of derived table is that by using dblink we can use the table from other schema. ***Take charge of your career by going through our professionally designed*** *Sap Business Object Training Videos.*

22.what is User Objects.

It is a universe of classes and objects which is developed by the universe designer. Once the objects which is there in the universe does not matches our requirement, then the user can create his own objects which is known as User objects.

23.Name @functions.

The @functions are:

- @where
- @Prompt
- @script
- @Variable
- @Select

24.Write the uses of @functions.

The @prompt function tells the end user to write any specific values.
The @script function recovers the result of the Visual Basics for applications macro's.
The @select function re-used the existing statements.
25. business object consists of how many domains? Write them.
Three Domains are there in Business Objects

- Document
- Security
- Universe

26.Define secured connection?

It is a kind of connection which overcomes the disadvantages related with the former connection. We can use universe in central repository by secured connection.

27.How we can use one derived table from another?

Using @Derived_table function, we can use one derived table from another. The syntax is: @derived_table(derived table name)

28.What is Slice in Business Objects.

It is used to reset, rename and omit the blocks. It work with the master or detailed reports.

29.Differentiate between Dice and Slice.

- Dice: It shows the data and removes the data.
- Slice: It reset, rename and delete the blocks.

30.Define master/detail report?

Big block of data can be break into small sections by using master or detail report. Replication in values can be avoided by using this.

31.What is class?

Collection of objects in a universe is known as class. Subclasses are derived from the class.

32.How many possible ways are there for linking universes?

There are three ways:
- The Master approach.
- The Kernal approach.
- The Component approach.

33.What is data mining?

It is a process by which we can extract the needed details from the database.

34.Write the available Drill modes.

The available Drill modes are;
- Drill down.
- Drill up
- Drill through.
- Drill by

35.What is aggregate_awarness?

We use aggregate_awarness function to define one object for measures in fact tables when we have a same fact tables in different grains, the syntax is as, @aggregate_aware(highest_level.lower level)

36.What is fan trap?

A one to many join links to a table which answer with another one to many join links is known as fan trap.

37.What is Data provider?

The question or the data source is known as the data provider.

38.At what time we use a context?

Context is developed when the dimension objects are their in one or both fact tables.

39.Define standard mode?

Only the clients inside the group can be accessed in the standard mode.

40.Write the schema supported by Business Objects Designer.

There are five different schemas supported by Business Objects designer
- Snowflake schema.
- Star schema
- Normalized production schema
- Multistar schema.
- Data warehouse with aggregates.

41.What is Channel?

It is to make the users know up-to-date information. Channel is a website with ‘push’ technology.

42.What are the limitations over user objects?

User objects are not common between the end users. It is kept in a specific user object definition file. Hence if any end-user tries to refresh or edit the query contains another user’s user object, it will be routinely cleaned and removed.***Give your career a big boost by going through our*** *Business Objects Certification* ***now!***

43.Name the tasks of universe designer.

The tasks are:
- Creating the universe.
- Designing the universe.
- Distributing the universe.
- Maintaining the universe

44.Write the main components of designer interface.

The main components are:
- The structure pane.
- The table browser
- The universe pane.

45.Define report bursting?

We use report bursting for maintaining the version documents according to the user profiles.

46.What is WEBI?

It is a solution that is particular in supporting the decisions related with reports, queries, and analysis.

47.What is the full form of DSS?

Decision Support Systems.

48.What is strategies?

Strategies is used for extracting automatically structural data from database or from a flat file.

49.Define universe?

Universe is a group of objects and classes. These objects and classes will be projected for an application or a set of users.

50.What is secured mode?

Secured mode blocked the contact of specific users over specific commands.

51.Define Drill by?

Drill by is used to move to other hierarchy and examine the other data, which belongs to another hierarchy.

52.Define list of values?

It is file which have the data values linked with an object.This blog will help you get a better understanding of Business Objects !

Compare Spark vs Hadoop MapReduce

Criteria	Hadoop MapReduce	Apache Spark
Memory	Does not leverage the memory of the hadoop cluster to maximum.	Let's save data on memory with the use of RDD's.
Disk usage	MapReduce is disk oriented.	Spark caches data in-memory and ensures low latency.
Processing	Only batch processing is supported	Supports real-time processing through spark streaming.
Installation	Is bound to hadoop.	Is not bound to Hadoop.

Spark vs Hadoop

Simplicity, Flexibility and Performance are the major advantages of using Spark over Hadoop.

- Spark is 100 times faster than Hadoop for big data processing as it stores the data in-memory, by placing it in Resilient Distributed Databases (RDD).
- Spark is easier to program as it comes with an interactive mode.
- It provides complete recovery using lineage graph whenever something goes wrong.

Refer Spark vs Hadoop

2) What is Shark?

Most of the data users know only SQL and are not good at programming. Shark is a tool, developed for people who are from a database background - to access Scala MLib capabilities through Hive like SQL interface. Shark tool helps data users run Hive on Spark - offering compatibility with Hive metastore, queries and data.

3) List some use cases where Spark outperforms Hadoop in processing.

i. Sensor Data Processing –Apache Spark's 'In-memory computing' works best here, as data is retrieved and combined from different sources.

ii. Spark is preferred over Hadoop for real time querying of data

iii. Stream Processing – For processing logs and detecting frauds in live streams for alerts, Apache Spark is the best solution.

4) What is a Sparse Vector?

A sparse vector has two parallel arrays –one for indices and the other for values. These vectors are used for storing non-zero entries to save space.

5) What is RDD?

RDDs (Resilient Distributed Datasets) are basic abstraction in Apache Spark that represent the data coming into the system in object format. RDDs are used for in-memory computations on large clusters, in a fault tolerant manner. RDDs are read-only portioned, collection of records, that are –

- Immutable – RDDs cannot be altered.
- Resilient – If a node holding the partition fails the other node takes the data.

6) Explain about transformations and actions in the context of RDDs.

Transformations are functions executed on demand, to produce a new RDD. All transformations are followed by actions. Some examples of transformations include map, filter and reduceByKey.

Actions are the results of RDD computations or transformations. After an action is performed, the data from RDD moves back to the local machine. Some examples of actions include reduce, collect, first, and take.

7) What are the languages supported by Apache Spark for developing big data applications?

Scala, Java, Python, R and Clojure

8) Can you use Spark to access and analyse data stored in Cassandra databases?

Yes, it is possible if you use Spark Cassandra Connector.

9) Is it possible to run Apache Spark on Apache Mesos?

Yes, Apache Spark can be run on the hardware clusters managed by Mesos.

10) Explain about the different cluster managers in Apache Spark

The 3 different clusters managers supported in Apache Spark are:

- YARN
- Apache Mesos -Has rich resource scheduling capabilities and is well suited to run Spark along with other applications. It is advantageous when several users run interactive shells because it scales down the CPU allocation between commands.
- Standalone deployments – Well suited for new deployments which only run and are easy to set up.

11) How can Spark be connected to Apache Mesos?

To connect Spark with Mesos-

- Configure the spark driver program to connect to Mesos. Spark binary package should be in a location accessible by Mesos. (or)
- Install Apache Spark in the same location as that of Apache Mesos and configure the property 'spark.mesos.executor.home' to point to the location where it is installed.

12) How can you minimize data transfers when working with Spark?

Minimizing data transfers and avoiding shuffling helps write spark programs that run in a fast and reliable manner. The various ways in which data transfers can be minimized when working with Apache Spark are:

1. Using Broadcast Variable- Broadcast variable enhances the efficiency of joins between small and large RDDs.
2. Using Accumulators – Accumulators help update the values of variables in parallel while executing.
3. The most common way is to avoid operations ByKey, repartition or any other operations which trigger shuffles.

13) Why is there a need for broadcast variables when working with Apache Spark?

These are read only variables, present in-memory cache on every machine. When working with Spark, usage of broadcast variables eliminates the necessity to ship copies of a variable for every task, so data can be processed faster. Broadcast variables help in storing a lookup table inside the memory which enhances the retrieval efficiency when compared to an RDD lookup ().

14) Is it possible to run Spark and Mesos along with Hadoop?

Yes, it is possible to run Spark and Mesos with Hadoop by launching each of these as a separate service on the machines. Mesos acts as a unified scheduler that assigns tasks to either Spark or Hadoop.

15) What is lineage graph?

The RDDs in Spark, depend on one or more other RDDs. The representation of dependencies in between RDDs is known as the lineage graph. Lineage graph information is used to compute each RDD on demand, so that whenever a part of persistent RDD is lost, the data that is lost can be recovered using the lineage graph information.

16) How can you trigger automatic clean-ups in Spark to handle accumulated metadata?

You can trigger the clean-ups by setting the parameter 'spark.cleaner.ttl' or by dividing the long running jobs into different batches and writing the intermediary results to the disk.

17) Explain about the major libraries that constitute the Spark Ecosystem

- **Spark MLib-** Machine learning library in Spark for commonly used learning algorithms like clustering, regression, classification, etc.
- **Spark Streaming** – This library is used to process real time streaming data.
- **Spark GraphX** – Spark API for graph parallel computations with basic operators like joinVertices, subgraph, aggregateMessages, etc.
- **Spark SQL** – Helps execute SQL like queries on Spark data using standard visualization or BI tools.

18) What are the benefits of using Spark with Apache Mesos?

It renders scalable partitioning among various Spark instances and dynamic partitioning between Spark and other big data frameworks.

19) What is the significance of Sliding Window operation?

Sliding Window controls transmission of data packets between various computer networks. Spark Streaming library provides windowed computations where the transformations on RDDs are applied over a sliding window of data. Whenever the window slides, the RDDs that fall within the particular window are combined and operated upon to produce new RDDs of the windowed DStream.

20) What is a DStream?

Discretized Stream is a sequence of Resilient Distributed Databases that represent a stream of data. 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.

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

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.

22) What is Catalyst framework?

Catalyst framework is a new optimization framework present in Spark SQL. It allows Spark to automatically transform SQL queries by adding new optimizations to build a faster processing system.

23) Name a few companies that use Apache Spark in production.

Pinterest, Conviva, Shopify, Open Table

24) Which spark library allows reliable file sharing at memory speed across different cluster frameworks?

Tachyon

25) Why is BlinkDB used?

BlinkDB is a query engine for executing interactive SQL queries on huge volumes of data and renders query results marked with meaningful error bars. BlinkDB helps users balance ‘query accuracy’ with response time.

26) How can you compare Hadoop and Spark in terms of ease of use?

Hadoop MapReduce requires programming in Java which is difficult, though Pig and Hive make it considerably easier. Learning Pig and Hive syntax takes time. Spark has interactive APIs for different languages like Java, Python or Scala

and also includes Shark i.e. Spark SQL for SQL lovers - making it comparatively easier to use than Hadoop.

27) What are the common mistakes developers make when running Spark applications?

Developers often make the mistake of-

- Hitting the web service several times by using multiple clusters.
- Run everything on the local node instead of distributing it.

Developers need to be careful with this, as Spark makes use of memory for processing.

28) What is the advantage of a Parquet file?

Parquet file is a columnar format file that helps –

- Limit I/O operations
- Consumes less space
- Fetches only required columns.

29) What are the various data sources available in SparkSQL?

- Parquet file
- JSON Datasets
- Hive tables

30) How Spark uses Hadoop?

Spark has its own cluster management computation and mainly uses Hadoop for storage.

For the complete list of big data companies and their salaries- CLICK HERE

31) What are the key features of Apache Spark that you like?

- Spark provides advanced analytic options like graph algorithms, machine learning, streaming data, etc
- It has built-in APIs in multiple languages like Java, Scala, Python and R
- It has good performance gains, as it helps run an application in the Hadoop cluster ten times faster on disk and 100 times faster in memory.

32) What do you understand by Pair RDD?

Special operations can be performed on RDDs in Spark using key/value pairs and such RDDs are referred to as Pair RDDs. Pair RDDs allow users to access each key in parallel. They have a reduceByKey () method that collects data based on each key and a join () method that combines different RDDs together, based on the elements having the same key.

33) Which one will you choose for a project –Hadoop MapReduce or Apache Spark?

The answer to this question depends on the given project scenario - as it is known that Spark makes use of memory instead of network and disk I/O. However, Spark uses large amount of RAM and requires dedicated machine to produce effective results. So the decision to use Hadoop or Spark varies dynamically with the requirements of the project and budget of the organization.

34) Explain about the different types of transformations on DStreams?

- Stateless Transformations- Processing of the batch does not depend on the output of the previous batch. Examples – map (), reduceByKey (), filter ().
- Stateful Transformations- Processing of the batch depends on the intermediary results of the previous batch. Examples –Transformations that depend on sliding windows.

35) Explain about the popular use cases of Apache Spark

Apache Spark is mainly used for

- Iterative machine learning.
- Interactive data analytics and processing.
- Stream processing
- Sensor data processing

36) Is Apache Spark a good fit for Reinforcement learning?

No. Apache Spark works well only for simple machine learning algorithms like clustering, regression, classification.

37) What is Spark Core?

It has all the basic functionalities of Spark, like - memory management, fault recovery, interacting with storage systems, scheduling tasks, etc.

38) How can you remove the elements with a key present in any other RDD?

Use the subtractByKey () function

39) What is the difference between persist() and cache()

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

40) What are the various levels of persistence in Apache Spark?

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
- MEMORY_ONLY_SER

- MEMORY_AND_DISK
- MEMORY_AND_DISK_SER, DISK_ONLY
- OFF_HEAP

41) How Spark handles monitoring and logging in Standalone mode?

Spark has a web based user interface for monitoring the cluster in standalone mode that shows the cluster and job statistics. The log output for each job is written to the work directory of the slave nodes.

42) Does Apache Spark provide check pointing?

Lineage graphs are always useful to recover RDDs from a failure but this is generally time consuming if the RDDs have long lineage chains. Spark has an API for check pointing i.e. a REPLICATE flag to persist. However, the decision on which data to checkpoint - is decided by the user. Checkpoints are useful when the lineage graphs are long and have wide dependencies.

43) How can you launch Spark jobs inside Hadoop MapReduce?

Using SIMR (Spark in MapReduce) users can run any spark job inside MapReduce without requiring any admin rights.

44) How Spark uses Akka?

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.

45) How can you achieve high availability in Apache Spark?

- Implementing single node recovery with local file system
- Using StandBy Masters with Apache ZooKeeper.

46) Hadoop uses replication to achieve fault tolerance. How is this achieved in Apache Spark?

Data storage model in Apache Spark is based on RDDs. RDDs help achieve fault tolerance through lineage. RDD always has the information on how to build from other datasets. If any partition of a RDD is lost due to failure, lineage helps build only that particular lost partition.

47) Explain about the core components of a distributed Spark application.

- Driver- The process that runs the main () method of the program to create RDDs and perform transformations and actions on them.
- Executor –The worker processes that run the individual tasks of a Spark job.
- Cluster Manager-A pluggable component in Spark, to launch Executors and Drivers. The cluster manager allows Spark to run on top of other external managers like Apache Mesos or YARN.

48) 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 a 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.

49) Define a worker node.

A node that can run the Spark application code in a cluster can be called as a worker node. A worker node can have more than one worker which is configured by setting the SPARK_ WORKER_INSTANCES property in the spark-env.sh file. Only one worker is started if the SPARK_ WORKER_INSTANCES property is not defined.

50) What do you understand by SchemaRDD?

An RDD that consists of row objects (wrappers around basic string or integer arrays) with schema information about the type of data in each column.

51) What are the disadvantages of using Apache Spark over Hadoop MapReduce?

Apache spark does not scale well for compute intensive jobs and consumes large number of system resources. Apache Spark's in-memory capability at times comes a major roadblock for cost efficient processing of big data. Also, Spark does have its own file management system and hence needs to be integrated with other cloud based data platforms or apache hadoop.

52) Is it necessary to install spark on all the nodes of a YARN cluster while running Apache Spark on YARN ?

No , it is not necessary because Apache Spark runs on top of YARN.

53) What do you understand by Executor Memory in a Spark application?

Every spark application has same fixed heap size and fixed number of cores for a spark executor. The heap size is what referred to as the Spark executor memory which is controlled with the spark.executor.memory property of the –executor-memory flag. Every spark application will have one executor on each worker node. The executor memory is basically a measure on how much memory of the worker node will the application utilize.

Spark Streaming Interview Questions

1) Name some sources from where Spark streaming component can process real-time data.

Apache Flume, Apache Kafka, Amazon Kinesis

2) Name some companies that are already using Spark Streaming.

Uber, Netflix, Pinterest.

3) What is the bottom layer of abstraction in the Spark Streaming API ?

DStream.

We invite the big data community to share the most frequently asked Apache Spark Interview questions and answers, in the comments below - to ease big data job interviews for all prospective analytics professionals.

1. What is Apache Spark?

Wikipedia defines Apache Spark “an open source cluster computing framework originally developed in the AMPLab at University of California, Berkeley but was later donated to the Apache Software Foundation where it remains today. In contrast to Hadoop’s two-stage disk-based MapReduce paradigm, Spark’s multi-stage in-memory primitives provides performance up to 100 times faster for certain applications. By allowing user programs to load data into a cluster’s memory and query it repeatedly, Spark is well-suited to machine learning algorithms.”

Spark is essentially a fast and flexible data processing framework. It has an advanced execution engine supporting cyclic data flow with in-memory computing functionalities. Apache Spark can run on Hadoop, as a standalone system or on the cloud. Spark is capable of accessing diverse data sources including HDFS, HBase, Cassandra among others

2. Explain the key features of Spark.

• Spark allows Integration with Hadoop and files included in HDFS.

• It has an independent language (Scala) interpreter and hence comes with an interactive language shell.

• It consists of RDD’s (Resilient Distributed Datasets), that can be cached across computing nodes in a cluster.

• It supports multiple analytic tools that are used for interactive query analysis, real-time analysis and graph processing. Additionally, some of the salient features of Spark include:

Lighting fast processing: When it comes to Big Data processing, speed always matters, and Spark runs Hadoop clusters way faster than others. Spark makes this possible by reducing the number of read/write operations to the disc. It stores this intermediate processing data in memory.

Support for sophisticated analytics: In addition to simple “map” and “reduce” operations, Spark supports SQL queries, streaming data, and complex analytics such as machine learning and graph algorithms. This allows users to combine all these capabilities in a single workflow.

Real-time stream processing: Spark can handle real-time streaming. MapReduce primarily handles and processes previously stored data even though there are other frameworks to obtain real-time streaming. Spark does this in the best way possible.

3. What is “RDD”?

RDD stands for Resilient Distribution Datasets: a collection of fault-tolerant operational elements that run in parallel. The partitioned data in RDD is immutable and is distributed in nature.

4. How does one create RDDs in Spark?

In Spark, parallelized collections are created by calling the SparkContext “parallelize” method on an existing collection in your driver program.

```
val data = Array(4,6,7,8)

val distData = sc.parallelize(data)
```

Text file RDDs can be created using SparkContext’s “textFile” method. Spark has the ability to create distributed datasets from any storage source supported by Hadoop, including your local file system, HDFS, Cassandra, HBase, Amazon S3, among others. Spark supports text files, “SequenceFiles”, and any other Hadoop “InputFormat” components.

```
val inputfile = sc.textFile(“input.txt”)
```

5. What does the Spark Engine do?

Spark Engine is responsible for scheduling, distributing and monitoring the data application across the cluster.

6. Define “Partitions”.

A “Partition” is a smaller and logical division of data, that is similar to the “split” in Map Reduce. Partitioning is the process that helps derive logical units of data in order to speed up data processing.

Here’s an example: val someRDD = sc.parallelize(1 to 100, 4)

Here an RDD of 100 elements is created in four partitions, which then distributes a dummy map task before collecting the elements back to the driver program.

7. What operations does the “RDD” support?

- Transformations
- Actions

8. Define “Transformations” in Spark.

“Transformations” are functions applied on RDD, resulting in a new RDD. It does not execute until an action occurs. ***map***() and ***filer***() are examples of “transformations”, where the former applies the function

assigned to it on each element of the RDD and results in another RDD. The ***filter***() creates a new RDD by selecting elements from the current RDD.

9. Define “Action” in Spark.

An “action” helps in bringing back the data from the RDD to the local machine. Execution of “action” is the result of all transformations created previously. ***reduce***() is an action that implements the function passed again and again until only one value is left. On the other hand, the ***take***() action takes all the values from the RDD to the local node.

10. What are the functions of “Spark Core”?

The “SparkCore” performs an array of critical functions like memory management, monitoring jobs, fault tolerance, job scheduling and interaction with storage systems.

It is the foundation of the overall project. It provides distributed task dispatching, scheduling, and basic input and output functionalities. RDD in Spark Core makes it fault tolerance. RDD is a collection of items distributed across many nodes that can be manipulated in parallel. Spark Core provides many APIs for building and manipulating these collections.

11. What is an “RDD Lineage”?

Spark does not support data replication in the memory. In the event of any data loss, it is rebuilt using the “RDD Lineage”. It is a process that reconstructs lost data partitions.

12. What is a “Spark Driver”?

“Spark Driver” is the program that runs on the master node of the machine and declares transformations and actions on data RDDs. The driver also delivers RDD graphs to the “Master”, where the standalone cluster manager runs.

13. What is SparkContext?

“SparkContext” is the main entry point for Spark functionality. A “SparkContext” represents the connection to a Spark cluster, and can be used to create RDDs, accumulators and broadcast variables on that cluster.

14. What is Hive on Spark?

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.

15. Name a few commonly used Spark Ecosystems.

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

16. What is “Spark Streaming”?

Spark supports stream processing, essentially an extension to the Spark API. This allows stream processing of live data streams. The data from different sources like Flume and HDFS is streamed and processed to file systems, live dashboards and databases. It is similar to batch processing as the input data is divided into streams like batches.
Business use cases for Spark streaming: Each Spark component has its own use case. Whenever you want to analyze data with the latency of less than 15 minutes and greater than 2 minutes i.e. near real time is when you use Spark streaming

17. What is “GraphX” in Spark?

“GraphX” is a component in Spark which is used for graph processing. It helps to build and transform interactive graphs.

18. What is the function of “MLlib”?

“MLlib” is Spark’s machine learning library. It aims at making machine learning easy and scalable with common learning algorithms and real-life use cases including clustering, regression filtering, and dimensional reduction among others.

19. What is “Spark SQL”?

Spark SQL is a Spark interface to work with structured as well as semi-structured data. It has the capability to load data from multiple structured sources like “textfiles”, JSON files, Parquet files, among others. Spark SQL provides a special type of RDD called SchemaRDD. These are row objects, where each object represents a record.

Here’s how you can create an SQL context in Spark SQL:

SQL context: scala> var sqlContext=new SqlContext

HiveContext: scala> var hc = new HIVEContext(sc)

20. What is a “Parquet” in Spark?

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

21. What is an “Accumulator”?

“Accumulators” are Spark’s offline debuggers. Similar to “Hadoop Counters”, “Accumulators” provide the number of “events” in a program.

Accumulators are the variables that can be added through associative operations. Spark natively supports accumulators of numeric value types and standard mutable collections. “AggregrateByKey()” and “combineByKey()” uses accumulators.

22. Which file systems does Spark support?

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

23. What is “YARN”?

“YARN” is a large-scale, distributed operating system for big data applications. It is one of the key features of Spark, providing a central and resource management platform to deliver scalable operations across the cluster.

24. List the benefits of Spark over MapReduce.

- Due to the availability of in-memory processing, Spark implements the processing around 10-100x faster than Hadoop MapReduce.
- Unlike MapReduce, Spark provides in-built libraries to perform multiple tasks form the same core; like batch processing, steaming, machine learning, interactive SQL queries among others.
- MapReduce is highly disk-dependent whereas Spark promotes caching and in-memory data storage
- Spark is capable of iterative computation while MapReduce is not.

Additionally, Spark stores data in-memory whereas Hadoop stores data on the disk. Hadoop uses replication to achieve fault tolerance while Spark uses a different data storage model, resilient distributed datasets (RDD). It also uses a clever way of guaranteeing fault tolerance that minimizes network input and output.

25. What is a “Spark Executor”?

When “SparkContext” connects to a cluster manager, it acquires an “Executor” on the cluster nodes. “Executors” are Spark processes that run computations and store the data on the worker node. The final tasks by “SparkContext” are transferred to executors.

26. List the various types of “Cluster Managers” in Spark.

The Spark framework supports three kinds of Cluster Managers:

- Standalone
- Apache Mesos
- YARN

27. What is a “worker node”?

“Worker node” refers to any node that can run the application code in a cluster.

28. Define “PageRank”.

“PageRank” is the measure of each vertex in a graph.

29. Can we do real-time processing using Spark SQL?

30. What is the biggest shortcoming of Spark?

Spark utilizes more storage space compared to Hadoop and MapReduce.

Also, Spark streaming is not actually streaming, in the sense that some of the window functions cannot properly work on top of micro batching.

Got a question for us? Please mention it in the comments section and we will get back to you.

- In Spark, you can basically do everything using single application / console (pyspark or scala console) and get the results immediately. Switching between 'Running something on cluster' and 'doing something locally' is fairly easy and straightforward. This also leads to less context switch of the developer and more productivity.

Q2: Is there are point of learning Mapreduce, then?
A: Yes. For the following reason:

Mapreduce is a paradigm used by many big data tools including Spark. So, understanding the MapReduce paradigm and how to convert a problem into series of MR tasks is very important.

When the data grows beyond what can fit into the memory on your cluster, the Hadoop Map-Reduce paradigm is still very relevant.
Almost, every other tool such as Hive or Pig converts its query into MapReduce phases. If you understand the Mapreduce then you will be able to optimize your queries better.

Q3: When running Spark on Yarn, do I need to install Spark on all nodes of Yarn Cluster? A: Since spark runs on top of Yarn, it utilizes yarn for the execution of its commands over the cluster's nodes. So, you just have to install Spark on one node.

Q4: What are the downsides of Spark?
A:
Spark utilizes the memory. The developer has to be careful. A casual developer might make following mistakes:
She may end up running everything on the local node instead of distributing work over to the cluster.
She might hit some webservice too many times by the way of using multiple clusters.

The first problem is well tackled by Hadoop Map reduce paradigm as it ensures that the data your code is churning is fairly small a point of time thus you can make a mistake of trying to handle whole data on a single node.
The second mistake is possible in Map-Reduce too. While writing Map-Reduce, user may hit a service from inside of map() or reduce() too many times. This overloading of service is also possible while using Spark.

Q5: What is a RDD?
A:
The full form of RDD is resilience distributed dataset. It is a representation of data located on a network which is
Immutable - You can operate on the rdd to produce another rdd but you can’t alter it.
Partitioned / Parallel - The data located on RDD is operated in parallel. Any operation on RDD is done using multiple nodes.
Resilience - If one of the node hosting the partition fails, another nodes takes its data.

RDD provides two kinds of operations: Transformations and Actions.

Q6: What is Transformations?
A: The transformations are the functions that are applied on an RDD (resilient distributed data set). The transformation results in another RDD. A transformation is not executed until an action follows.

The example of transformations are:
map() - applies the function passed to it on each element of RDD resulting in a new RDD.
filter() - creates a new RDD by picking the elements from the current RDD which pass the function argument.

Q7: What are Actions?
A: An action brings back the data from the RDD to the local machine. Execution of an action results in all the previously created transformation. The example of actions are:
reduce() - executes the function passed again and again until only one value is left. The function should take two argument and return one value.

take() - take all the values back to the local node form RDD.

http://www.knowbigdata.com/blog/interview-questions-apache-spark-part-2
Q1: Say I have a huge list of numbers in RDD(say myrdd). And I wrote the following code to compute average:

```
def myAvg(x, y):
 return (x+y)/2.0;
avg = myrdd.reduce(myAvg);
```
What is wrong with it? And How would you correct it?

A:The average function is not commutative and associative;
```
cnt = myrdd.count();
def devideByCnd(x):
 return x/cnt;
myrdd1 = myrdd.map(devideByCnd);
avg = myrdd.reduce(sum);
```
Q2: Say I have a huge list of numbers in a file in HDFS. Each line has one number.And I want to compute the square root of sum of squares of these numbers. How would you do it?
We would first load the file as RDD from HDFS on spark

```
numsAsText = sc.textFile("hdfs://hadoop1.knowbigdata.com/user/student/sgiri/mynumbersfile.txt");

# Define the function to compute the squares
def toSqInt(str):
 v = int(str);
 return v*v;
#Run the function on spark rdd as transformation
nums = numsAsText.map(toSqInt);

#Run the summation as reduce action
total = nums.reduce(sum)

#finally compute the square root. For which we need to import math.
import math;
print math.sqrt(total);
```

3: Is the following approach correct? Is the sqrtOfSumOfSq a valid reducer?

```
numsAsText =sc.textFile("hdfs://hadoop1.knowbigdata.com/user/student/sgiri/mynumbersfile.txt");
def toInt(str):
 return int(str);
nums = numsAsText.map(toInt);
def sqrtOfSumOfSq(x, y):
 return math.sqrt(x*x+y*y);
total = nums.reduce(sum)
import math;
print math.sqrt(total);
```
A: Yes. The approach is correct and sqrtOfSumOfSq is a valid reducer.

Q4: Could you compare the pros and cons of the your approach (in Question 2 above) and my approach

(in Question 3 above)?

A:

You are doing the square and square root as part of reduce action while I am squaring in map() and summing in reduce in my approach.

My approach will be faster because in your case the reducer code is heavy as it is calling math.sqrt() and reducer code is generally executed approximately n-1 times the spark RDD.
The only downside of my approach is that there is a huge chance of integer overflow because I am computing the sum of squares as part of map.

Q5: If you have to compute the total counts of each of the unique words on spark, how would you go about it?

A:

```
#This will load the bigtextfile.txt as RDD in the spark

lines = sc.textFile("hdfs://hadoop1.knowbigdata.com/user/student/sgiri/bigtextfile.txt");


#define a function that can break each line into words

def toWords(line):

      return line.split();
# Run the toWords function on each element of RDD on spark as flatMap transformation.
# We are going to flatMap instead of map because our function is returning multiple values.
words = lines.flatMap(toWords);
# Convert each word into (key, value) pair. Her key will be the word itself and value will be 1.
def toTuple(word):
     return (word, 1);
wordsTuple = words.map(toTuple);
# Now we can easily do the reduceByKey() action.
def sum(x, y):
    return x+y;

counts = wordsTuple.reduceByKey(sum)
# Now, print
counts.collect()
```

Q6: In a very huge text file, you want to just check if a particular keyword exists. How would you do this using Spark?

A:

```
lines = sc.textFile("hdfs://hadoop1.knowbigdata.com/user/student/sgiri/bigtextfile.txt");
def isFound(line):
 if line.find(“mykeyword”) > -1:
  return 1;
 return 0;
```

```
foundBits = lines.map(isFound);
sum = foundBits.reduce(sum);
if sum > 0:
 print “FOUND”;
else:
 print “NOT FOUND”;
```

Q7: Can you improve the performance of this code in previous answer?

A: Yes. The search is not stopping even after the word we are looking for has been found. Our map code would keep executing on all the nodes which is very inefficient.

We could utilize accumulators to report whether the word has been found or not and then stop the job. Something on these line:

```
import thread, threading
from time import sleep
result = "Not Set"
lock = threading.Lock()
accum = sc.accumulator(0)
def map_func(line):
 #introduce delay to emulate the slowness
 sleep(1);
 if line.find("Adventures") > -1:
  accum.add(1);
  return 1;
 return 0;
def start_job():
 global result
 try:
  sc.setJobGroup("job_to_cancel", "some description")
  lines = sc.textFile("hdfs://hadoop1.knowbigdata.com/user/student/sgiri/wordcount/input/big.txt");
  result = lines.map(map_func);
  result.take(1);
 except Exception as e:
  result = "Cancelled"
 lock.release()
def stop_job():
 while accum.value < 3 :
  sleep(1);
 sc.cancelJobGroup("job_to_cancel")
supress = lock.acquire()
supress = thread.start_new_thread(start_job, tuple())
supress = thread.start_new_thread(stop_job, tuple())
supress = lock.acquire()
```

http://spark.apache.org/docs/latest/spark-standalone.html#high-availability

By default, standalone scheduling clusters are resilient to Worker failures (insofar as Spark itself is resilient to losing work by moving it to other workers). However, the scheduler uses a Master to make scheduling decisions, and this (by default) creates a single point of failure: if the Master crashes, no new applications can be created. In order to circumvent this, we have two high availability schemes, detailed below.

Standby Masters with ZooKeeper

Overview

Utilizing ZooKeeper to provide leader election and some state storage, you can launch multiple Masters in your cluster connected to the same ZooKeeper instance. One will be elected “leader” and the others will remain in standby mode. If the current leader dies, another Master will be elected, recover the old Master’s state, and then resume scheduling. The entire recovery process (from the time the first leader goes down) should take between 1 and 2 minutes. Note that this delay only affects scheduling *new* applications – applications that were already running during Master failover are unaffected.

https://jaceklaskowski.gitbooks.io/mastering-apache-spark/content/exercises/spark-exercise-standalone-master-ha.html