What is Spark?

Spark is scheduling, monitoring and distributing engine for big data.It is a cluster computing platform designed to be fast and general purpose.Spark extends the popular MapReduce model.One of the main features Spark offers for speed is the ability to run computations in memory, but the system is also more efficient than MapReduce for complex applications running on disk.

What is Standalone mode?

In standalone mode, Spark uses a Master daemon which coordinates the efforts of the Workers, which run the executors. Standalone mode is the default, but it cannot be used on secure clusters.When you submit an application, you can choose how much memory its executors will use, as well as the total number of cores across all executors.

What is YARN mode?

In YARN mode, the YARN ResourceManager performs the functions of the Spark Master. The functions of the Workers are performed by the YARN NodeManager daemons, which run the executors. YARN mode is slightly more complex to set up, but it supports security.

What is Apache Spark?

Spark is a fast, easy-to-use and flexible data processing framework. 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.

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.

What are client mode and cluster mode?

Each application has a driver process which coordinates its execution. This process can run in the foreground (client mode) or in the background (cluster mode). Client mode is a little simpler, but cluster mode allows you to easily log out after starting a Spark application without terminating the application.

Define 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.

How to run spark in Standalone client mode?

```
spark-submit \
–class org.apache.spark.examples.SparkPi \
–deploy-mode client \
–master spark//$SPARK_MASTER_IP:$SPARK_MASTER_PORT \
$SPARK_HOME/examples/lib/spark-examples_version.jar 10
```

How to run spark in Standalone cluster mode?

```
spark-submit \
–class org.apache.spark.examples.SparkPi \
–deploy-mode cluster \
–master spark//$SPARK_MASTER_IP:$SPARK_MASTER_PORT \
$SPARK_HOME/examples/lib/spark-examples_version.jar 10
```

How to run spark in YARN client mode?

```
spark-submit \
–class org.apache.spark.examples.SparkPi \
–deploy-mode client \
–master yarn \
$SPARK_HOME/examples/lib/spark-examples_version.jar 10
```

How to run spark in YARN cluster mode?

```
spark-submit \
–class org.apache.spark.examples.SparkPi \
–deploy-mode cluster \
–master yarn \
$SPARK_HOME/examples/lib/spark-examples_version.jar 10
```

What is Executor memory?

You can configure this using the –executor-memory argument to sparksubmit. Each application will have at most one executor on each worker, so this setting controls how much of that worker's memory the application will claim. By default, this setting is 1 GB—you will likely want to increase it on most servers.

What is the maximum number of total cores?

This is the total number of cores used across all executors for an application. By default, this is unlimited; that is, the application will launch executors on every available node in the cluster. For a multiuser workload, you should instead ask users to cap their usage. You can set this value through the –total-execution cores argument to spark-submit, or by configuring spark.cores.max in your Spark configuration file.

What does a Spark Engine do?

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

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.

What operations RDD support?

Transformations

Actions

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.

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.

Define functions of Spark Core.

Spark Core performs various important functions like memory management, monitoring jobs, fault-tolerance, job scheduling and interaction with storage systems.

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.

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.

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.

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.

What are the main components of Spark?

Spark Core: Spark Core contains the basic functionality of Spark, including components for task scheduling, memory management, fault recovery, interacting with storage systems, and more. Spark Core is also home to the API that defines RDDs,

Spark SQL: Spark SQL is Spark's package for working with structured data. It allows querying data via SQL as well as the HQL.

Spark Streaming: Spark Streaming is a Spark component that enables processing of live streams of data. Examples of data streams include logfiles generated by production web servers.

MLlib: Spark comes with a library containing common machine learning (ML) functionality, called MLlib. MLlib provides multiple types of machine learning algorithms.

GraphX: GraphX is a library for manipulating graphs (e.g., a social network's friend graph) and performing graph-parallel computations.

How Spark Streaming works?

Spark Streaming receives live input data streams and divides the data into batches, which are then processed by the Spark engine to generate the final stream of results in batches.Spark Streaming provides a high-level abstraction called discretized stream or DStream, which represents a continuous stream of data. DStreams can be created either from input data streams from sources such as Kafka, Flume, or by applying high-level operations on other DStreams. Internally, a DStream is represented as a sequence of RDDs.

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 tobatch processing as the input data is divided into streams like batches.

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.

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.

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.

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.

What file systems Spark support?

Hadoop Distributed File System (HDFS)

Local File system

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.

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

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

What is persist()?

Spark's RDDs are by default recomputed each time you run an action on them. If you would like to reuse an RDD in multiple actions, you can ask Spark to persist it using RDD.persist().After computing it the first time, Spark will store the RDD contents in memory (partitioned across the machines in your cluster), and reuse them in future actions. Persisting RDDs on disk instead of memory is also possible.

Write common workflow of a Spark program?

Every Spark program and shell session will work as follows:

Create some input RDDs from external data.

Transform them to define new RDDs using transformations like filter().

Ask Spark to persist() any intermediate RDDs that will need to be reused.

Launch actions such as count() and first() to kick off a parallel computation, which is then optimized and executed by Spark.

Difference between cache() and persist()?

With cache(), you use only the default storage level MEMORY_ONLY. With persist(), you can specify which storage level you want.So cache() is the same as calling persist() with the default storage level.Spark has many levels of persistence to choose from based on what our goals are.The default persist() will store the data in the JVM heap as unserialized objects. When we write data out to disk, that data is also always serialized.Different levels of persistence are MEMORY_ONLY, MEMORY_ONLY_SER, MEMORY_AND_DISK, MEMORY_AND_DISK_SER, DISK_ONLY.

What is lineage graph?

As you derive new RDDs from each other using transformations, Spark keeps track of the set of dependencies between different RDDs, called the lineage graph. It uses this information to compute each RDD on demand and to recover lost data if part of a persistent RDD is lost.

Difference between map() and flatMap()?

The map() transformation takes in a function and applies it to each element in the RDD with the result of the function being the new value of each element in the resulting RDD. Sometimes we want to produce multiple output elements for each input element. The operation to do this is called flatMap(). As with map(), the function we provide to flatMap() is called individually for each element in our input RDD. Instead of returning a single element, we return an iterator with our return values.

What is reduce() action?

It takes a function that operates on two elements of the type in your RDD and returns a new element of the same type. A simple example of such a function is +, which we can use to sum our RDD. With reduce(), we can easily sum the elements of our RDD, count the number of elements, and perform other types of aggregations.

What is Pair RDD?

Spark provides special operations on RDDs containing key/value pairs. These RDDs are called pair RDDs. Pair RDDs are a useful building block in many programs, as they expose operations that allow you to act on each key in parallel.For example, pair RDDs have a reduceByKey() method that can aggregate data separately for each key, and a join() method that can merge two RDDs together by grouping elements with the same key.

What are Accumulators?

Accumulators, provides a simple syntax for aggregating values from worker nodes back to the driver program. One of the most common uses of accumulators is to count events that occur during job execution for debugging purposes.

What is Broadcast Variables?

Spark's second type of shared variable, broadcast variables, allows the program to efficiently send a large, read-only value to all the worker nodes for use in one or more Spark operations. They come in handy, for example, if your application needs to send a large, read-only lookup table to all the nodes.

What is Piping?

Spark provides a pipe() method on RDDs. Spark's pipe() lets us write parts of jobs using any language we want as long as it can read and write to Unix standard streams. With pipe(), you can write a transformation of an RDD that reads each RDD element from standard input as a String, manipulates that String however you like, and then writes the result(s) as Strings to standard output.

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.

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.

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.

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

What are Executors?

Spark executors are worker processes responsible for running the individual tasks in a given Spark job. Executors are launched once at the beginning of a Spark application and typically run for the entire lifetime of an application.Executors have two roles. First, they run the tasks that make up the application and return results to the driver.Second, they provide in-memory storage for RDDs that are cached by user programs.

What are the steps that occur when you run a Spark application on a cluster?

The user submits an application using spark-submit.

Spark-submit launches the driver program and invokes the main() method specified by the user.

The driver program contacts the cluster manager to ask for resources to launch executors.

The cluster manager launches executors on behalf of the driver program.

The driver process runs through the user application. Based on the RDD actions and transformations in the program, the driver sends work to executors in the form of tasks.

Tasks are run on executor processes to compute and save results.

If the driver's main() method exits or it calls SparkContext.stop(),it will terminate the executors and release resources from the cluster manager.

What is Spark SQL?

Spark SQL is a module in Apache Spark that integrates relational processing(e.g., declarative queries and optimized storage) with Spark's procedural programming API. Spark SQL makes two main additions.First, it offers much tighter integration between relational and procedural processing, through a declarative DataFrame API.Second, it includes a highly extensible optimizer, Catalyst.

Big data applications require a mix of processing techniques, data sources and storage formats. The earliest systems designed for these workloads, such as MapReduce, gave users a powerful, but low-level, procedural programming interface. Programming such systems was onerous and required manual optimization by the user to achieve high performance. As a result, multiple new systems sought to provide a more productive user experience by offering relational interfaces to big data. Systems like Pig, Hive and Shark all take advantage of declarative queries to provide richer automatic optimizations.

What is a schema RDD/DataFrame?

A SchemaRDD is an RDD composed of Row objects with additional schema information of the types in each column. Row objects are just wrappers around arrays of basic types (e.g., integers and strings).

What are Row objects?

Row objects represent records inside SchemaRDDs, and are simply fixed-length arrays of fields.Row objects have a number of getter functions to obtain the value of each field given its index. The standard getter, get (or apply in Scala), takes a column number and returns an Object type (or Any in Scala) that we are responsible for casting to the correct type. For Boolean, Byte, Double, Float, Int, Long, Short, and String, there is a getType() method, which returns that type. For example, get String(0) would return field 0 as a string.

What are DStreams?

Much like Spark is built on the concept of RDDs, Spark Streaming provides an abstraction called DStreams, or discretized streams. A DStream is a sequence of data arriving over time. Internally, each DStream is represented as a sequence of RDDs arriving at each time step. DStreams can be created from various input sources, such as Flume, Kafka, or HDFS. Once built, they offer two types of operations: transformations, which yield a new DStream, and output operations, which write data to an external system.

Explain Spark Streaming Architecture?

Spark Streaming uses a "micro-batch" architecture, where Spark Streaming receives data from various input sources and groups it into small batches. New batches are created at regular time intervals. At the beginning of each time interval a new batch is created, and any data that arrives during that interval gets added to that batch. At the end of the time interval the batch is done growing. The size of the time intervals is determined by a parameter called the batch interval. Each input batch forms an RDD, and is processed using Spark jobs to create other RDDs. The processed results can then be pushed out to external systems in batches.

What are the types of Transformations on DStreams?

In stateless transformations the processing of each batch does not depend on the data of its previous batches. They include the common RDD transformations like map(), filter(), and reduceByKey().

• Stateful transformations, in contrast, use data or intermediate results from previous batches to compute the results of the current batch. They include transformations based on sliding windows and on tracking state across time.

What is Receiver in Spark Streaming?

Every input DStream is associated with a Receiver object which receives the data from a source and stores it in Spark's memory for processing.

How Spark achieves fault tolerance?

Spark stores data in-memory whereas Hadoop stores data on disk. Hadoop uses replication to achieve fault tolerance whereas Spark uses different data storage model, RDD. RDDs achieve fault tolerance through a notion of lineage: if a partition of an RDD is lost, the RDD has enough information to rebuild just that partition.This removes the need for replication to achieve fault tolerance.

What are Spark's main features?

Speed : Spark enables applications in Hadoop clusters to run up to 100x faster in memory, and 10x faster even when running on disk. Spark makes it possible by reducing number of read/write to disc. It stores this intermediate processing data in-memory. It uses the concept of an Resilient Distributed Dataset (RDD), which allows it to transparently store data on memory and persist it to disc only it's needed. This helps to reduce most of the disc read and write – the main time consuming factors – of data processing.

Combines SQL, streaming, and complex 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 out-of-the-box. Not only that, users can combine all these capabilities seamlessly in a single workflow.

Ease of Use:Spark lets you quickly write applications in Java, Scala, or Python. This helps developers to create and run their applications on their familiar programming languages and easy to build parallel apps.

Runs Everywhere: Spark runs on Hadoop, Mesos, standalone, or in the cloud. It can access diverse data sources including HDFS, Cassandra, HBase, S3.

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

Is Apache Spark a good fit for Reinforcement learning?

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

What is Spark Core?

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

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

Use the subtractByKey () function

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

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

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

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.

Does Apache Spark provide checkpointing?

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.

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.

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.

How can you achieve high availability in Apache Spark?

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

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.

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.

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.

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.

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.

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.