Relevant Projects

Machine Learning Projects

Data Science Projects

Python Projects for Data Science

Data Science Projects in R

Machine Learning Projects for Beginners

Deep Learning Projects

Neural Network Projects

Tensorflow Projects

NLP Projects

Kaggle Projects

IoT Projects

Big Data Projects

Hadoop Real-Time Projects Examples

Spark Projects

Data Analytics Projects for Students

Real-Time Log Processing in Kafka for Streaming Architecture

The goal of this apache kafka project is to process log entries from applications in real-time using Kafka for the streaming architecture in a microservice sense.

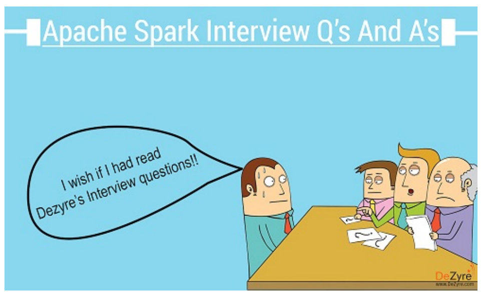
View Project Details

Online Hadoop Projects - Solving small file problem in Hadoop

In this hadoop project, we are going to be continuing the series on data engineering by discussing and implementing various ways to solve the hadoop small file problem.

Top 50 Spark Interview Questions and Answers for 2021

Last Updated: 25 Jan 2021


Apache Spark Interview Q's And A's
I wish if I had read Dezyre's Interview questions!!
DeZyre

The questions asked at a big data developer or apache spark developer job interview may fall into one of the following categori based on Spark Ecosystem Components -

- Spark Basic Interview Questions
- Spark SQL Interview Questions
- Spark MLlib Interview Questions
- Spark Streaming Interview Questions

In addition, displaying project experience in the following is key -

1. Spark Streaming projects
2. Spark MLlib projects
3. Apache Spark projects
4. PySpark projects

With the increasing demand from the industry, to process big data at a faster pace - Apache Spark is gaining huge momentum when it comes to enterprise adoption. Hadoop MapReduce well supported the need to process big data fast but there was always a need among developers to learn more flexible tools to keep up with the superior market of midsize big data sets, for real time data processing within seconds.

To support the momentum for faster big data processing, there is increasing demand for Apache Spark developers who can validate their expertise in implementing best practices for Spark - to build complex big data solutions. In collaboration with and big data industry experts -we have curated a list of top 50 Apache Spark Interview Questions and Answers that will help students/professionals nail a big data developer interview and bridge the talent supply for Spark Developers across various industry segments.

Click here to view a list of 50+ solved, end-to-end Big Data and Machine Learning Project Solutions (reusable code + videos)

Companies like Amazon, Shopify, Alibaba and eBay are adopting Apache Spark for their big data deployments- the demand for Spark developers is expected to grow exponentially. Google Trends confirm “hockey-stick-like-growth” in Spark enterprise adoption and awareness among organizations across various industries. Spark is becoming popular because of its ability to handle event streaming and processing big data faster than Hadoop MapReduce. 2017 is the best time to hone your Apache Spark skills and pursue a fruitful career as a data analytics professional, data scientist or big data developer.

These Apache Spark Projects will help you develop skills which will make you eligible to apply for Spark developer job roles.

Top 50 Apache Spark Interview Questions and

View Project Details

Data Warehouse Design for E-commerce Environments

In this hive project, you will design a data warehouse for e-commerce environments.

View Project Details

Yelp Data Processing using Spark and Hive Part 2

In this spark project, we will continue building the data warehouse from the previous project Yelp Data Processing Using Spark And Hive Part 1 and will do further data processing to develop diverse data products.

View Project Details

Tough engineering choices with large datasets in Hive Part - 1

Explore hive usage efficiently in this hadoop hive project using various file formats such as JSON, CSV, ORC, AVRO and compare their relative performances

View Project Details

Hadoop Project for Beginners-SQL Analytics with Hive

In this hadoop project, learn about the features in Hive that allow us to perform analytical queries over large datasets.

View Project Details

AWS Project - Build an ETL Data Pipeline on AWS EMR Cluster

Build a fully working scalable, reliable and secure AWS EMR complex data pipeline from scratch that provides support for all data stages from data collection to data analysis and visualization.

View Project Details

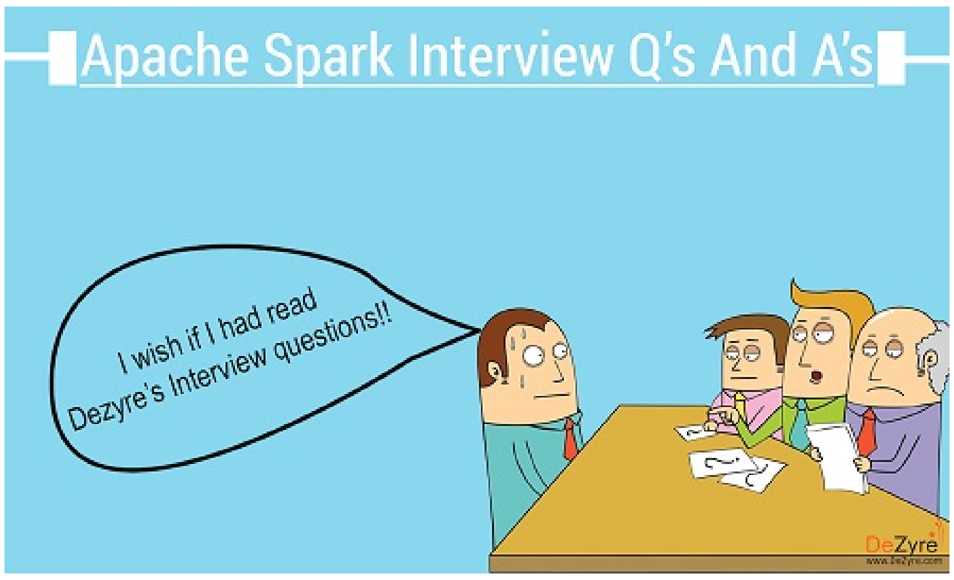
Data processing with Spark SQL

In this Apache Spark SQL project, we will go through provisioning data for retrieval using Spark SQL.

View Project Details

Top 50 Apache Spark Interview Questions and Answers

Preparation is very important to reduce the nervous energy at any big data job interview. Regardless of the big data expertise and skills one possesses, every candidate dreads the face to face big data job interview. Though there is no way of predicting exactly what questions will be asked in any big data or spark developer job interview- these Apache spark interview questions and answers might help you prepare for these interviews better.


Apache Spark Interview Q's And A's
I wish if I had read Dezyre's Interview questions!!
DeZyre
www.DeZyre.com

1) 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

Explore features of Spark SQL in practice on Spark 2.0

The goal of this spark project for students is to explore the features of Spark SQL in practice on the latest version of Spark i.e. Spark 2.0.

View Project Details

Real-Time Log Processing using Spark Streaming Architecture

In this Spark project, we are going to bring processing to the speed layer of the lambda architecture which opens up capabilities to monitor application real time performance, measure real time comfort with applications and real time alert in case of security

View Project Details

-->

You might also like

Data Scientist Salary

How to Become a Data Scientist

Data Analyst vs Data Scientist

Data Scientist Resume

Data Science Projects for Beginners

Machine Learning Engineer

Machine Learning Projects for Beginners

Datasets

Pandas Dataframe

Machine Learning Algorithms

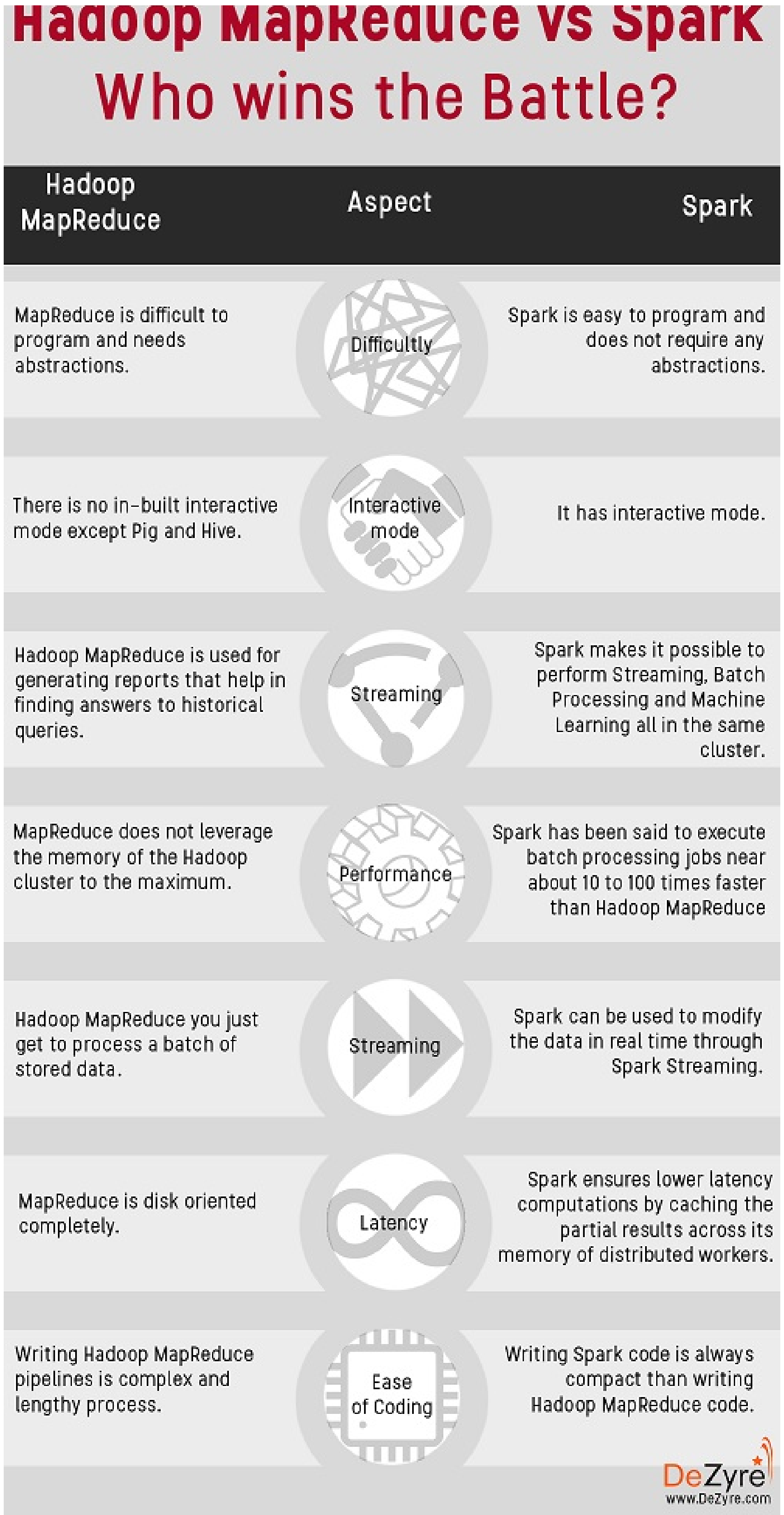
Regression Analysis

MNIST Dataset

Data Science Interview Questions

Python Data Science Interview Questions

Spark Interview Questions



Hadoop MapReduce vs Spark
Who wins the Battle?

Hadoop MapReduce	Aspect	Spark
MapReduce is difficult to program and needs abstractions.	Difficulty	Spark is easy to program and does not require any abstractions.
There is no in-built interactive mode except Pig and Hive.	Interactive mode	It has interactive mode.
Hadoop MapReduce is used for generating reports that help in finding answers to historical queries.	Streaming	Spark makes it possible to perform Streaming, Batch Processing and Machine Learning all in the same cluster.
MapReduce does not leverage the memory of the Hadoop cluster to the maximum.	Performance	Spark has been said to execute batch processing jobs near about 10 to 100 times faster than Hadoop MapReduce
Hadoop MapReduce you just get to process a batch of stored data.	Streaming	Spark can be used to modify the data in real time through Spark Streaming.
MapReduce is disk oriented completely.	Latency	Spark ensures lower latency computations by caching the partial results across its memory of distributed workers.
Writing Hadoop MapReduce pipelines is complex and lengthy process.	Ease of Coding	Writing Spark code is always compact than writing Hadoop MapReduce code.

DeZyre
www.DeZyre.com

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

Click here to view 52+ solved, reusable project solutions in Big Data - Spark

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.

Hadoop Interview Questions

Data Analyst Interview Questions

Machine Learning Interview Questions

AWS vs Azure

Hadoop Architecture

Spark Architecture

Blog Categories

- Big Data
- CRM
- Data Science
- Data Science Projects in Python
- Live Courses
- Machine Learning Projects in Python
- Mobile App Development
- NoSQL Database
- Web Development

Tutorials

- Hadoop Online Tutorial – Hadoop HDFS Commands Guide
- MapReduce Tutorial–Learn to implement Hadoop WordCount Example
- Hadoop Hive Tutorial-Usage of Hive Commands in HQL
- Hive Tutorial-Getting Started with Hive Installation on Ubuntu
- Learn Java for Hadoop Tutorial: Inheritance and Interfaces
- Learn Java for Hadoop Tutorial: Classes and Objects
- Learn Java for Hadoop Tutorial: Arrays
- Apache Spark Tutorial - Run your First Spark Program
- PySpark Tutorial-Learn to use Apache

- PySpark Tutorial- Learn to use Apache Spark with Python
- R Tutorial- Learn Data Visualization with R using GGVIS
- Neural Network Training Tutorial
- Python List Tutorial
- MatPlotLib Tutorial
- Decision Tree Tutorial
- Neural Network Tutorial
- Performance Metrics for Machine Learning Algorithms
- R Tutorial: Data.Table
- SciPy Tutorial
- Step-by-Step Apache Spark Installation Tutorial
- Introduction to Apache Spark Tutorial
- R Tutorial: Importing Data from Web
- R Tutorial: Importing Data from Relational Database
- R Tutorial: Importing Data from Excel
- Introduction to Machine Learning Tutorial
- Machine Learning Tutorial: Linear Regression
- Machine Learning Tutorial: Logistic Regression
- Support Vector Machine Tutorial (SVM)
- K-Means Clustering Tutorial
- dplyr Manipulation Verbs
- Introduction to dplyr package
- Importing Data from Flat Files in R
- Principal Component Analysis Tutorial
- Pandas Tutorial Part-3
- Pandas Tutorial Part-2
- Pandas Tutorial Part-1
- Tutorial- Hadoop Multinode Cluster Setup on Ubuntu
- Data Visualizations Tools in R
- R Statistical and Language tutorial
- Introduction to Data Science with R
- Apache Pig Tutorial: User Defined

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.

- [Apache Pig Tutorial: User Defined Function Example]
- Apache Pig Tutorial Example: Web Log Server Analytics
- Impala Case Study: Web Traffic
- Impala Case Study: Flight Data Analysis
- Hadoop Impala Tutorial
- Apache Hive Tutorial: Tables
- Flume Hadoop Tutorial: Twitter Data Extraction
- Flume Hadoop Tutorial: Website Log Aggregation
- Hadoop Sqoop Tutorial: Example Data Export
- Hadoop Sqoop Tutorial: Example of Data Aggregation
- Apache Zookepeer Tutorial: Example of Watch Notification
- Apache Zookepeer Tutorial: Centralized Configuration Management
- Hadoop Zookeeper Tutorial
- Hadoop Sqoop Tutorial
- Hadoop PIG Tutorial
- Hadoop Oozie Tutorial
- Hadoop NoSQL Database Tutorial
- Hadoop Hive Tutorial
- Hadoop HDFS Tutorial
- Hadoop hBase Tutorial
- Hadoop Flume Tutorial
- Hadoop 2.0 YARN Tutorial
- Hadoop MapReduce Tutorial
- Big Data Hadoop Tutorial for Beginners- Hadoop Installation

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.

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?

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.

54) What does the Spark Engine do?

Spark engine schedules, distributes and monitors the data application across the spark cluster.

55) What makes Apache Spark good at low-latency workloads like graph processing and machine learning?

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

56) Is it necessary to start Hadoop to run any Apache Spark Application ?

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

57) What is the default level of parallelism in apache spark?

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

58) Explain about the common workflow of a Spark program

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

59) In a given spark program, how will you identify whether a given operation is Transformation or Action ?

One can identify the operation based on the return type -

i) The operation is an action, if the return type is other than RDD.

ii) The operation is transformation, if the return type is same as the RDD.

60) What according to you is a common mistake apache spark developers make when using spark ?

- Maintaining the required size of shuffle blocks.
- Spark developer often make mistakes with managing directed acyclic graphs (DAG's.)

61) Suppose that there is an RDD named ProjectPrordd that contains a huge list of numbers. The following spark code is written to calculate the average -

```
def ProjectProAvg(x, y):
return (x+y)/2.0;
avg = ProjectPrordd.reduce(ProjectProAvg);
```

What is wrong with the above code and how will you correct it ?

Average function is neither commutative nor associative. The best way to compute average is to first sum it and then divide it by count as shown below -

```
def sum(x, y):
return x+y;
total =ProjectPrordd.reduce(sum);
avg = total / ProjectPrordd.count();
```

However, the above code could lead to an overflow if the total becomes big. So, the best way to compute average is divide each number by count and then add up as shown below -

```
cnt = ProjectPrordd.count();
def divideByCnt(x):
return x/cnt;
myrdd1 = ProjectPrordd.map(divideByCnt);
avg = ProjectPrordd.reduce(sum);
```

Q. Say I have a huge list of numbers in a file in HDFS. Each line has one number.And I want to com

Spark SQL Interview Questions

1) Explain the difference between Spark SQL and Hive.

- Spark SQL is faster than Hive.
- Any Hive query can easily be executed in Spark SQL but vice-versa is not true.
- Spark SQL is a library whereas Hive is a framework.
- It is not mandatory to create a metastore in Spark SQL but it is mandatory to create a Hive metastore.
- Spark SQL automatically infers the schema whereas in Hive schema needs to be

- Spark SQL automatically infers the schema whereas in Hive schema needs to be explicitly declared..

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.

4) What do you understand by receivers in Spark Streaming ?

Receivers are special entities in Spark Streaming that consume data from various data sources and move them to Apache Spark. Receivers are usually created by streaming contexts as long running tasks on various executors and scheduled to operate in a round robin manner with each receiver taking a single core.

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.

5) How will you calculate the number of executors required to do real-time processing using Apache Spark? What factors need to be connsidered for deciding on the number of nodes for real-time processing?

The number of nodes can be decided by benchmarking the hardware and considering multiple factors such as optimal throughput (network speed), memory usage, the execution frameworks being used (YARN, Standalone or Mesos) and considering the other jobs that are running within those execution frameworks along with spark.

6) What is the difference between Spark Transform in DStream and map ?

tranform function in spark streaming allows developers to use Apache Spark transformations on the underlying RDD's for the stream. map function in hadoop is used for an element to element transform and can be implemented using transform.Ideally , map works on the elements of Dstream and transform allows developers to work with RDD's of the DStream. map is an elementary transformation whereas transform is an RDD transformation.

Check Out Top Scala Interview Questions for Spark Developers.

**Click here to view 52+ solved, end-to-end project solutions in
Big Data - Spark**

PREVIOUS NEXT

About us

Release your Data Science projects faster and get just-in-time learning. Get access to 100+ code recipes and project use-cases.

Contact

601, Allerton Street

Redwood City, California, 94063, USA

Email: contact@dezyre.com

Links

Mini Projects

Recipes

Data Science Blog

Plans & pricing

About Us

Contact Us

Privacy Policy

User Policy