Contents

[What is Hadoop? 2](#_Toc122087843)

[HDFS 2](#_Toc122087844)

[Clusters 2](#_Toc122087845)

[MapReduce 2](#_Toc122087846)

[YARN 3](#_Toc122087847)

[Why use Hadoop? 3](#_Toc122087848)

[Apache Spark architecture 4](#_Toc122087849)

[Why use Apache Spark? 5](#_Toc122087850)

[Spark RDD 5](#_Toc122087851)

[Similarities 5](#_Toc122087852)

[Differences 5](#_Toc122087853)

# What is Hadoop?

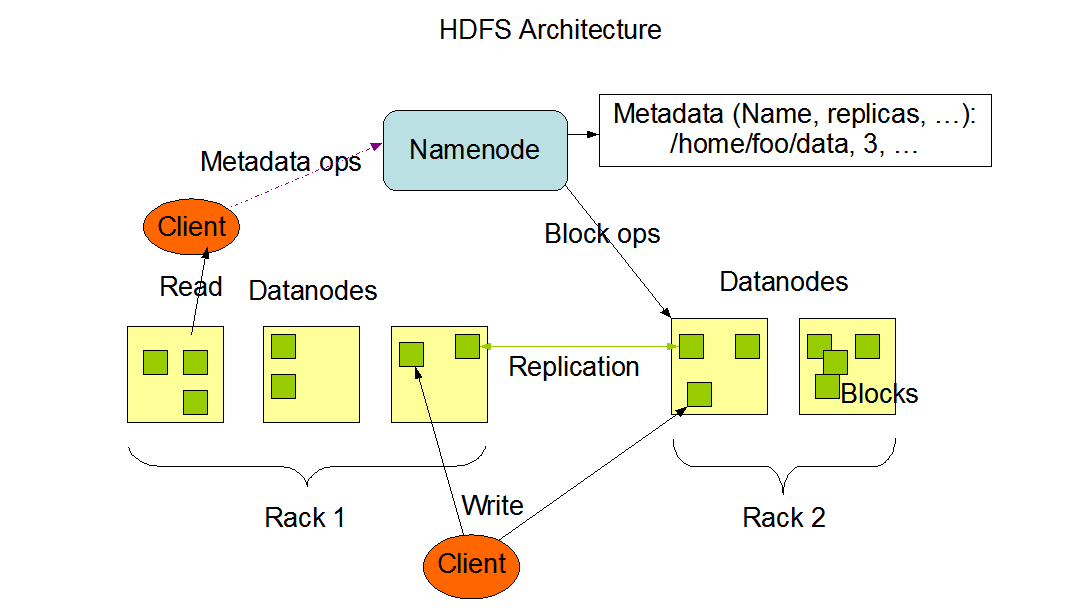
Hadoop is an open-source framework that stores and processes large amounts of data across clusters of distributed computer hardware. The four primary modules are Hadoop Distributed File System (HDFS), MapReduce, Yet Another Resource Negotiator (YARN), and Hadoop Common.

# HDFS

It is a storage unit that holds data across multiple machines and is the primary data storage system. HDFS has a block-structured file system that stores a few large files instead of many small files in blocks of size 128MB. These blocks are also replicated in case of corruption or machine failure which provides fault tolerance.

# Clusters

Clusters are a network of nodes that coordinate and perform various jobs. There are two components – a name node and a data node. While there is only one name node, there can be multiple data nodes. Name nodes, also known as the masters, are the primary component of HDFS and manage key operations and data storage. Data nodes, also known as the slaves, follow the instructions given by the name node, store data provided by the client, and follow the commands of the name node like reading, writing, processing, and replicating data.



# MapReduce

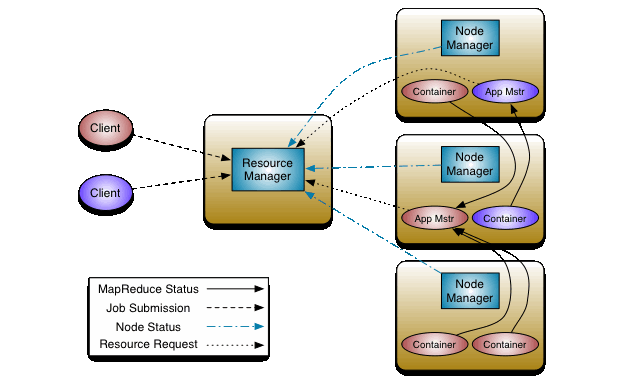
In MapReduce, the data is first assigned a key and value. Secondly, these key-value pairs are shuffled and sorted together based on their keys. And lastly, aggregation occurs, and the final output is obtained.

Diagram

Description automatically generated

# YARN

It is a resource management and job scheduling unit that runs distribution platforms. It is responsible for managing system resources for various applications ensuring the machine is not overloaded. It assigns small jobs to the slave nodes and prioritises important jobs over others.

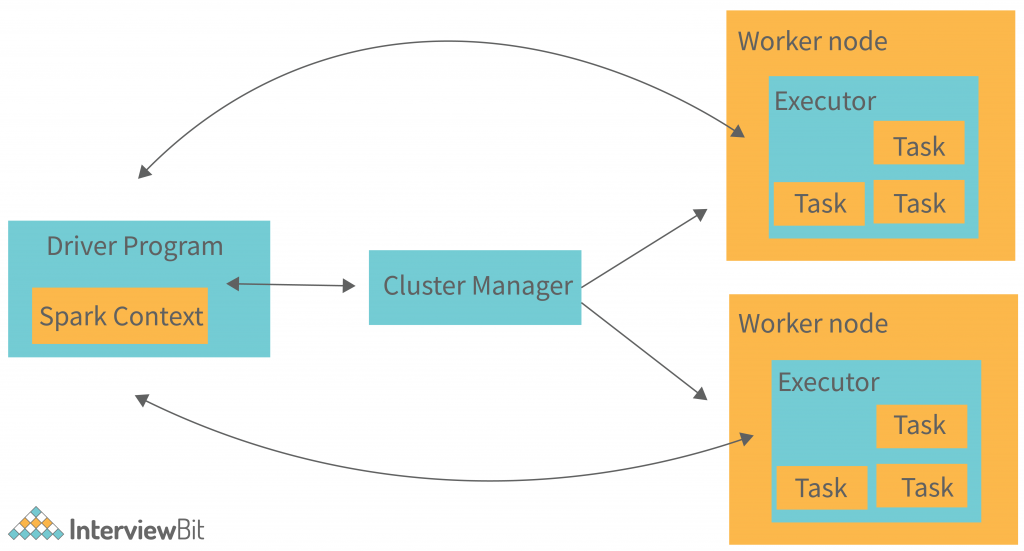


# Why use Hadoop?

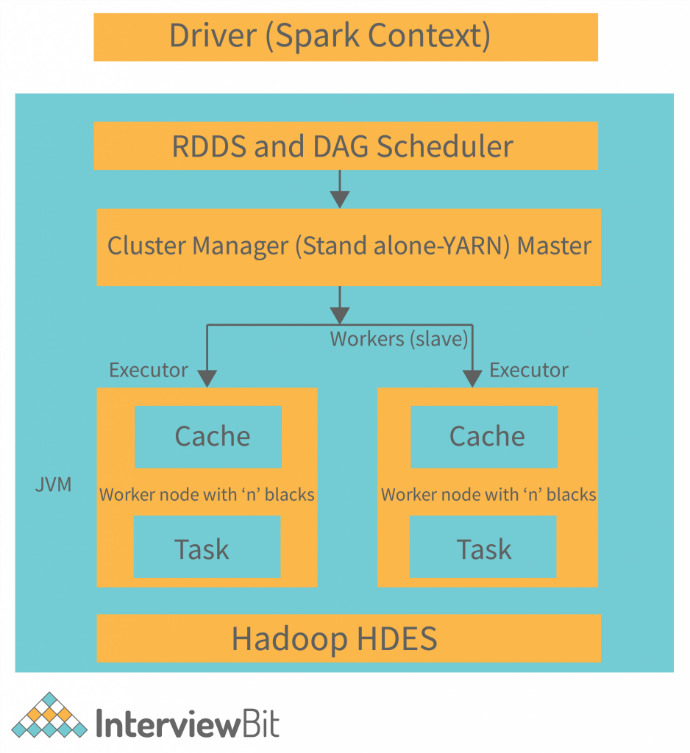
* Speed. HDFS divides the data into manageable smaller pieces and MapReduce runs parallel processes to store and retrieve data faster
* Diversity. HDFS can store different data formats
* Resilient. Data stored in a node is replicated in other cluster nodes, ensuring fault tolerance. Nodes that are down are automatically redirected to other nodes, allowing applications to run without interruption.

# Apache Spark architecture

A Spark application consists of two main components: a driver and executors. Drivers convert the user's code into multiple tasks that can be distributed across worker nodes, and executors execute the tasks assigned to them.



Spark builds the user’s data processing commands into a Directed Acyclic Graph or DAG. DAG is Spark’s scheduling layer and determines what tasks are executed on what nodes and in what sequence.



# Why use Apache Spark?

Due to two big advantages, Spark has become the framework of choice when processing big data, overtaking Hadoop’s MapReduce. The first advantage is speed. Spark’s in-memory data engine means it can perform tasks up to one hundred times faster than MapReduce in certain situations. MapReduce creates a two-stage execution graph consisting of data mapping and reducing, whereas Spark’s DAG has multiple stages that can be distributed more efficiently. The second advantage is the developer-friendly Spark API.

# Spark RDD

A Resilient Distributed Dataset (RDD) is a collection of objects that can be split across a computing cluster. Spark splits RDD operations into tasks and distributes them among many executor processes and executed in a parallel batch process, leading to fast and scalable parallel processing.



# Similarities

* Both Hadoop and Spark have forms of master and slave nodes. For Hadoop, they are the Name and Data Nodes. For Spark, they are the driver and executors.
* Both perform parallel processing as they use clustering to split up tasks.
* Both are data resilient. In Hadoop, the nodes file blocks are replicated. In Spark, if the RDD or a partition is lost or corrupted, it can be remade from the original fault-tolerant dataset or recomputed from past operations.

# Differences

* Hadoop relies on everyday hardware for storage while Spark uses RDDs to store data.
* Hadoop reads from and writes to HDFS while Spark processes data in RAM using RDD
* Spark is much faster than Hadoop. Spark uses in-memory processing via RAM while Hadoop reads from and writes to the disk.
* Instead of MapReduce, Spark uses DAG. One reason for Spark’s higher speed is that DAG has multiple stages that make distribution more efficient than MapReduce’s two-stage process.
* Hadoop manages batch processing while Spark manages real-time data.
* Hadoop is cheaper to scale up than Spark because Spark requires a lot of RAM to run in-memory processing.