**Spark**

* In Hadoop, the main concern is to maintain speed in processing large datasets in terms of waiting time between queries and waiting time to run the program.
* Spark was introduced by Apache Software Foundation for speeding up the Hadoop computational computing software process.
* **Spark is not a modified version of Hadoop** and is not, really, dependent on Hadoop because it has its own cluster management.
* Hadoop is just one of the ways to implement Spark.
* Spark uses Hadoop in two ways – one is **storage** and second is **processing**.
* Since Spark has its own cluster management computation, it uses Hadoop for storage purpose only.

**Apache Spark**

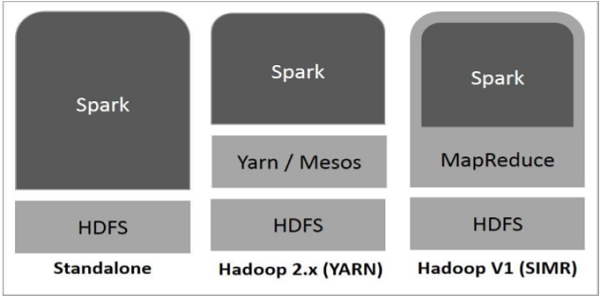
* Apache Spark is a lightning-fast cluster computing technology, designed for fast computation.
* It is based on Hadoop MapReduce and it extends the MapReduce model to efficiently use it for more types of computations.
* Spark is designed to cover a wide range of workloads such as batch applications, iterative algorithms, interactive queries and streaming.
* It reduces the management burden of maintaining separate tools.

## Evolution of Apache Spark

* Spark is one of Hadoop’s sub project developed in 2009 in UC Berkeley’s AMPLab by Matei Zaharia.

## Features of Apache Spark

* **Speed** − Spark helps to run an application in Hadoop cluster, up to 100 times faster in memory, and 10 times faster when running on disk.
* **Supports multiple languages** − Spark provides built-in APIs in Java, Scala, or Python. Therefore, you can write applications in different languages.
* **Advanced Analytics** − Spark not only supports ‘Map’ and ‘reduce’. It also supports SQL queries, Streaming data, Machine learning (ML), and Graph algorithms.



* **Standalone** − Spark Standalone deployment means Spark occupies the place on top of HDFS(Hadoop Distributed File System) and space is allocated for HDFS, explicitly. Here, Spark and MapReduce will run side by side to cover all spark jobs on cluster.
* **Hadoop Yarn** − Hadoop Yarn deployment means, simply, spark runs on Yarn without any pre-installation or root access required. It helps to integrate Spark into Hadoop ecosystem or Hadoop stack. It allows other components to run on top of stack.
* **Spark in MapReduce (SIMR)** − Spark in MapReduce is used to launch spark job in addition to standalone deployment. With SIMR, user can start Spark and uses its shell without any administrative access.

## Resilient Distributed Datasets

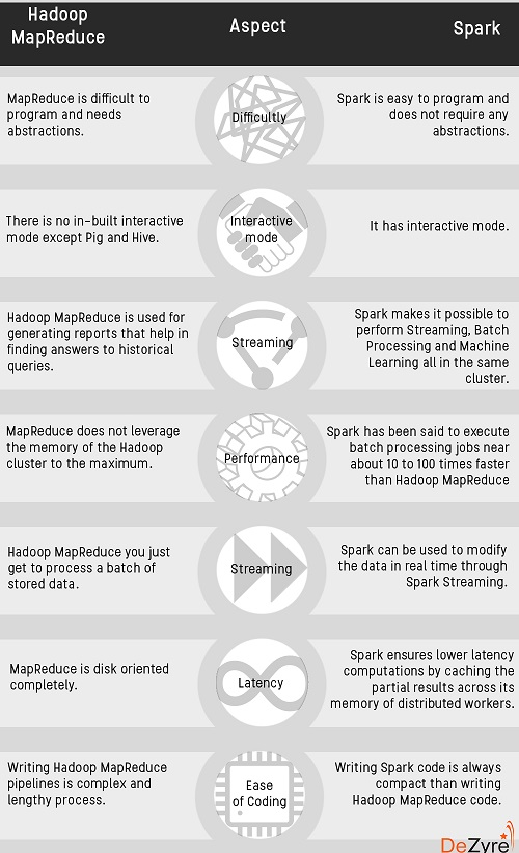
* Resilient Distributed Datasets (RDD) is a fundamental data structure of Spark.
* It is an immutable distributed collection of objects. Each dataset in RDD is divided into logical partitions, which may be computed on different nodes of the cluster. RDDs can contain any type of Python, Java, or Scala objects, including user-defined classes.
* Formally, an RDD is a read-only, partitioned collection of records.
* RDD is a fault-tolerant collection of elements that can be operated on in parallel.

**INTERVIEW QUESTIONS**

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. |

**Hadoop vs Spark:**

****

**Additional Terms:**

* 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.

1. **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.

1. **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.
4. **Where Apache Spark is a good fit for?**

Apache Spark works well for simple machine learning algorithms like clustering, regression, classification. But it is not suitable for Reinforcement learning.

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 explicitly declared.