**Hadoop, Spark and MapReduce**

Hadoop and Spark, both developed by the Apache Software Foundation, are widely used open-source frameworks for big data architectures. Each framework contains an extensive ecosystem of open-source technologies that prepare, process, manage and analyze big data sets.

**HADOOP**

MapReduce is a Java-based distributed computing programming model within the Hadoop framework. It is used to access large amounts of data in the Hadoop File System (HDFS). The Mapper and Reducer are two jobs performed in MapReduce programming. Mapper is responsible for sorting all the available data, while Reducer is in charge of aggregating it and turning it into smaller chunks

**Background:**

According to its co-founders, [Doug Cutting](https://en.wikipedia.org/wiki/Doug_Cutting) and [Mike Cafarella](https://en.wikipedia.org/wiki/Mike_Cafarella), the genesis of Hadoop was the Google File System paper that was published in October 2003.This paper spawned another one from Google – "MapReduce: Simplified Data Processing on Large Clusters". Development started on the APACHE NUTCH project, but was moved to the new Hadoop subproject in January 2006 Doug Cutting, who was working at Yahoo! at the time, named it after his son's toy elephant. The initial code that was factored out of Nutch consisted of about 5,000 lines of code for HDFS and about 6,000 lines of code for MapReduce.

Hadoop consists of the *Hadoop Common* package, which provides file system and operating system level abstractions, a MapReduce engine (either MapReduce/MR1 or YARN/MR2) and the [Hadoop Distributed File System](https://en.wikipedia.org/wiki/Apache_Hadoop#Hadoop_distributed_file_system) (HDFS). The Hadoop Common package contains the [Java Archive (JAR)](https://en.wikipedia.org/wiki/JAR_(file_format)) files and scripts needed to start Hadoop.

For effective scheduling of work, every Hadoop-compatible file system should provide location awareness, which is the name of the rack, specifically the network switch where a worker node is. Hadoop applications can use this information to execute code on the node where the data is, and, failing that, on the same rack/switch to reduce backbone traffic. HDFS uses this method when replicating data for data redundancy across multiple racks. This approach reduces the impact of a rack power outage or switch failure; if any of these hardware failures occurs, the data will remain available

**ADVANTAGES**

1. **Ease of Use**

Apache Spark contains APIs for Scala, Java, and Python and Spark SQL for SQL users. Apache Spark offers basic building blocks that allow users to easily develop user-defined functions. You can use Apache Spark in interactive mode when running commands to get an instant response.

**2. Data Processing**

Apache Spark can perform many other tasks than just data processing. Apache Spark can handle graphs and has its own Machine Learning Library – MLlib. Because of its great performance, Apache Spark can be used for both – batch and near real-time processing. Apache Spark is a flexible, multiple-use platform that can be used to handle all activities rather than dividing them over many platforms.

MapReduce is an ideal solution for batch processing in Hadoop. If you want a real-time solution, you can use Impala or Apache Storm, and for graph processing, you can use Apache Giraph. Earlier, MapReduce used Apache Mahout for Machine Learning tasks, but Mahout was discontinued when Spark came in.

**3. Performance**

Apache Spark is very much popular for its speed. It runs 100 times faster in memory and ten times faster on disk than Hadoop MapReduce since it processes data in memory (RAM). At the same time, Hadoop MapReduce has to persist data back to the disk after every Map or Reduce action.

Spark takes a lot of RAM to operate effectively. Spark saves processes to memory and keeps them there if different instructions are not given. If Spark is used with other resource-demanding services, its performance may be hampered notably. Additionally, Spark’s performance will suffer if the data sources are too large to fit fully in memory.

**4. Failure Recovery**

MapReduce is more suitable for recovery after failure than Spark since it uses hard drives instead of RAM. When Spark comes back online after crashing in the middle of a data processing activity, it will have to start all over from the beginning. This process requires more time.

If MapReduce fails while doing a job, it will resume where it left off when it restarts. Because MapReduce is based on a hard disk, it can maintain its position if it fails in the middle of a job.  
Both Spark and Hadoop MapReduce have high failure tolerance, but Hadoop MapReduce is slightly more tolerant.

**5. Security**

Apache Spark’s security is set to “OFF” by default, leaving you vulnerable to threats. Spark supports RPC channel authentication through a shared secret. Event logging is a feature of Spark, and Web UIs can be protected using java extensions. Furthermore, because Spark can operate on YARN and use HDFS, it can benefit from Kerberos authentication, HDFS file permissions, and encryption between nodes.

 Cost

While Spark and MapReduce are open-source solutions, you will still need to spend more on computers and staff. Spark and MapReduce can both run on commodity systems and in the cloud.

MapReduce requires a larger number of devices with higher disk space but little RAM capacity. On the other hand, Spark requires fewer devices with standard disk space but definitely higher RAM capacity. An adequate capacity is necessary to hold all the available data. Because disk space is less expensive than RAM space, MapReduce is the less expensive alternative.

**7. Scheduler**

Apache Spark can schedule all its tasks by itself, while Hadoop MapReduce requires external schedulers like Oozie.

**HADOOP VERSION AND IT’S ADVANTAGES.**

**Hadoop 1.x**

1. The Hadoop Common Module is a jar file which acts as the base API on top of which all the other components work.

2. Version one being the first one to come in existence is rock solid and has got no new updates

3. It has a limitation on the scaling nodes with just a maximum of 4000 nodes for each cluster

4. The functionality is limited utilizing the slot concept, i.e., the slots are capable of running a map task or a reduce task.

**Hadoop Version 2**

[HDFS Federation which has improved](https://www.educba.com/hdfs-federation/) to provide for horizontal scalability for the name node. Moreover, the name node was available for a single point of failure only, it is available on varied points. This is going to the Hadoop stat has been increased to include the stacks such as Hive, Pig, which make this tap well equipped enabling me to handle failures pertaining to Name Node.

YARN stands for Yet Another Resource Network has been improved with the new ability to process data in the larger term that is petabyte and terabyte to make it available for the HDFS while using the applications which are not MapReduce based. These include applications like MPI and GIRAPH.

**Version – 2.7.x Released on 31st May 2018:** The update focused to provide for two major functionalities that are providing for your application and providing for a global resource manager, thereby improving its overall utility and versatility, increasing scalability up to 10000 nodes for each cluster.

**Version 2.8.x – Released in September 2018:** The updated provided improvements include the capacity scheduler which is designed to provide multi-tenancy support for processing data over Hadoop and it has been made to be accessible for window uses so that there is an increase in the rate of adoption for the software across the industry for dealing with problems related to big data.

**Version 3**

Below is the latest running Hadoop Updated Version

**Version 3.1.x – released on 21 October 2019:**This update enables Hadoop to be utilized as a platform to serve a big chunk of Data Analytics Functions and utilities to be performed over event processing alongside using real-time operations give a better result.

It has now improved feature work on the container concept which enables had to perform generic which were earlier not possible with version 1.

The latest **version 3.2.1 released on 22nd September 2019** addresses issues of non-functionality (in terms of support) of data nodes for multi-Tenancy, limitation to you only MapReduce processing and the biggest problem than needed for an alternate data storage which is needed for the real-time processing and graphical analysis.

The ever-increasing Avalanche of data and [Big Data Analytics](https://www.educba.com/what-is-big-data-analytics/) pertaining to just business standing at an estimated 169 billion dollars (USD), the predicted growth to 274 billion dollars by 2022, the market seems to be growing ecstatically.

This all the more calls for a system that is integrable in its functioning for the abandoned Utah which is growing day by day. Hadoop app great to store, process and access the great solution which works to store process and access this heterogeneous set of data which can be unstructured/ structure in an organized manner.

With the feature of constant updates which act as tools to rectify the bugs that developers say while using Hadoop, and the improved versions increase the scope of application and improve the dimension and flexibility of using Hadoop, increases the chances of it is the next biggest to for all functions related to big data processing and Analytics.