The Apache Hadoop software library based framework that gives permissions to distribute huge amount of data sets processing across clusters of computers using easy programming models.

## The Apache Hadoop Module

**Hadoop Common:** Includes the common utilities which supports the other Hadoop modules

[HDFS](https://intellipaat.com/interview-question/hdfs-interview-questions/)**:** Hadoop Distributed File System provides unrestricted, high-speed access to the data application.

**Hadoop YARN:** This technology is basically used for scheduling of job and efficient management of the cluster resource.

[MapReduce](https://intellipaat.com/tutorial/mapreduce-tutorial/)**:** This is a highly efficient methodology for parallel processing of huge volumes of data.

***Then there are other projects included in the Hadoop module which are less used:***

[**Apache Ambari**](https://intellipaat.com/apache-ambari-training/)**:**It is a tool for managing, monitoring and provisioning of the Hadoop clusters. Apache Ambari supports the HDFS and MapReduce programs. Major highlights of Ambari are:

* Managing of the [Hadoop framework](https://intellipaat.com/big-data-hadoop-training/) is highly efficient, secure and consistent.
* Management of cluster operations with an intuitive web UI and a robust API
* The installation and configuration of [Hadoop cluster](https://intellipaat.com/blog/working-with-hadoop-cluster-set-up/) are simplified effectively.
* It is used to support automation, smart configuration and recommendations
* Advanced cluster security set-up comes additional with this tool kit.
* The entire cluster can be controlled using the metrics, heat maps, analysis and troubleshooting
* Increased levels of customization and extension make this more valuable.

**Cassandra** - Introduction. **Apache Cassandra** is a highly scalable, high-performance distributed **database** designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure. It is a**type** of **NoSQL database**

[**HBase**](https://intellipaat.com/hbase-training/)**:**it is a non-relational, distributed database management system that works efficiently on sparse data sets and it is highly scalable.

[**Apache Spark**](https://intellipaat.com/tutorial/spark-tutorial/)**:**This is highly agile, scalable and secure the Big Data compute engine, versatiles the sufficient work on a wide variety of applications like real-time processing, machine learning, ETL and so on.

**Hive:** It is a data warehouse tool basically used for analyzing, querying and summarizing of analyzed data concepts on top of the Hadoop framework.

**Pig:** Pig is a high-level framework which ensures us to work in coordination either with Apache Spark or MapReduce to analyze the data. The language used to code for the frameworks are known as Pig Latin.

**Sqoop:** This framework is used for transferring the data to Hadoop from relational databases. This application is based on a [command-line interface](https://intellipaat.com/tutorial/hbase-tutorial/installation/).

**Oozie:** This is a scheduling system for workflow management, executing workflow routes for successful completion of the task in a Hadoop.

**Zookeeper:** Open source centralized service which is used to provide coordination between distributed applications of Hadoop. It offers the registry and synchronization service on a high level.

* **Hadoop Mapreduce (Processing/Computation layer) –**[MapReduce is a parallel programming model](https://intellipaat.com/tutorial/mapreduce-tutorial/introduction-of-mapreduce/) mainly used for writing large amount of data distribution applications devised from Google for efficient processing of large amounts of datasets, on large group of clusters.
* **Hadoop HDFS (Storage layer) –**[Hadoop Distributed File System](https://intellipaat.com/tutorial/hadoop-tutorial/hdfs-overview/)or[HDFS](https://intellipaat.com/tutorial/hadoop-tutorial/hdfs-overview/) is based on the Google File System (GFS) which provides a distributed file system that is especially designed to run on commodity hardware. It reduces the faults or errors and helps incorporate low-cost hardware. It gives high level processing throughput access to application data and is suitable for [applications with large datasets](https://intellipaat.com/blog/big-data-analytics-tools-performance-testing/).
* **Hadoop YARN –**[Hadoop YARN is a framework](https://intellipaat.com/tutorial/big-data-and-hadoop-tutorial/hadoop-yarn-technology/)  used for job scheduling and cluster resource management.
* **Hadoop Common –**This includes [Java libraries](https://intellipaat.com/tutorial/java-tutorial/introduction-java/) and utilities which provide those java files which are essential to start Hadoop.
* **Task Tracker –**It is a node which is used to accept the tasks such as shuffle and Mapreduce form job tracker.
* **Job Tracker –**It is a service provider which runs [Mapreduce jobs](https://intellipaat.com/jobs/jobs-in-big-data/) on cluster.
* **Name Node –**It is a node where Hadoop stores all file location information(data stored location) in Hadoop distributed file system.
* **Data Node –** The data is stored in the Hadoop distributed file system.
* **Data Node –**It stores data in the Hadoop distributed file system.

## How does Hadoop Work?

Hadoop helps to execute large amount of processing where the user can connect together multiple commodity computers to a single-CPU, as a single functional distributed system and have the particular set of clustered machines that reads the dataset in parallel and provide intermediate, and after integration gets the desired output.

Hadoop runs code across a cluster of computers and performs the following tasks:

* Data are initially divided into files and directories. Files are divided into consistent sized blocks ranging from 128M and 64M.
* Then the files are distributed across various cluster nodes for further processing of data.
* Job tracker starts its scheduling programs on individual nodes.
* Once all the nodes are done with scheduling then the output is return back.

## The Challenges facing Data at Scale and the Scope of Hadoop

Big Data are categorized into:

* **Structured –**which stores the data in rows and columns like relational data sets
* **Unstructured – here**data cannot be stored in rows and columns like video, images, etc.
* **Semi-structured –**data in format XML are readable by machines and human

There is a standardized methodology that Big Data follows highlighting usage[methodology of ETL](https://intellipaat.com/interview-question/etl-interview-questions/).

**ETL – stands for Extract, Transform, and Load.**

**Extract** –fetching the data from multiple sources

**Transform** – convert the existing data to fit into the analytical needs

**Load** –right systems to derive value in it.

## Advantages of Hadoop

* It give access to the user to rapidly write and test the distributed systems and then automatically distributes the data and works across the machines and in turn utilizes the primary parallelism of the CPU cores.
* Hadoop library are developed to find/search and handle the failures at the application layer.
* Servers can be added or removed from the cluster dynamically at any point of time.
* It is open source based on Java applications and hence compatible on all the platforms.

## Hadoop Features and Characteristics

Apache Hadoop is the most popular and powerful big data tool, which  provides world’s best reliable storage layer –HDFS(Hadoop Distributed File System), a batch Processing engine namely  MapReduce and a Resource Management Layer like YARN.**Open-source** – Apache Hadoop is an open source project. It means its code can be modified according to business requirements.

* **Distributed Processing**– The data storage is maintained in a distributed manner in HDFS across the cluster, data is processed in parallel on cluster of nodes.
* **Fault Tolerance**– By default the three replicas of each block is stored across the cluster in Hadoop and it’s changed only when required. Hadoop’s fault tolerant can be examined in such cases when any node goes down, the data on that node can be recovered easily from other nodes. Failures of a particular node or task are recovered automatically by the framework.
* **Reliability**– Due to replication of data in the cluster, data can be reliable which is stored on the cluster of machine despite machine failures .Even if your machine goes down, and then also your data will be stored reliably.
* **High Availability**– Data is  available and accessible even there occurs a hardware failure due to multiple copies of data. If any incidents occurred such as if your machine or few hardware crashes, then data will be accessed from other path.
* **Scalability**– Hadoop is highly scalable and in a unique way hardware can be easily added to the nodes. It also provides horizontal scalability which means new nodes can be added **on the top** without any downtime**.**
* **Economic**– Hadoop is not very expensive as it runs on cluster of commodity hardware. We do not require any specialized machine for it. Hadoop provides huge cost reduction since it is very easy to add more nodes on the top here. So if the requirement increases, then there is an increase of nodes, without any downtime and without any much of pre planning.
* **Easy to use**– No need of client to deal with distributed computing, framework takes care of all the things. So it is easy to use.
* **Data Locality**– Hadoop works on data locality principle which states that the movement of computation of data instead of data to computation. When client submits his algorithm, then the algorithm is moved to data in the cluster instead of bringing data to the location where algorithm is submitted and then processing it.

## Hadoop Assumptions

Hadoop is written with huge amount of clusters of computers in mind and is built upon the following assumptions:

* Hardware may fail due to any external or technical malfunction where instead commodity hardware can be used.
* Processing will be run in batches and there exits an emphasis on high throughput as opposed to low latency.
* Applications which run on HDFS have large sets of data. A typical file in HDFS may be of gigabytes to terabytes in size.
* Applications require a **write-once-read-many**access model.
* Moving Computation is cheaper compared to the Moving Data.

## Hadoop Design Principles

The following are the design principles on which Hadoop works:

* System shall manage and heal itself as per the requirement occurred.
* Fault Tolerant are automatically and transparently route are managed around failures speculatively execute redundant tasks if certain nodes are detected to be running of slower phase.
* Performance is scaled based on linearity.
* Proportional change in terms of capacity with resource been change (Scalability)
* Compute must be moved to data.
* Data Locality is termed as lower latency, lower bandwidth.
* It is based on simple core, modular and extensible (Economical).

## Reference

<https://intellipaat.com/tutorial/hadoop-tutorial/>