Hadoop foundation of analytics:

HADOOP

Hadoop is an Apache open source framework written in java that allows distributed processing of large datasets across clusters of computers using simple programming models. The Hadoop framework application works in an environment that provides distributed *storage* and *computation* across clusters of computers. Hadoop is designed to scale up from single server to thousands of machines, each offering local computation and storage.

Hadoop Architecture

At its core, Hadoop has two major layers namely −

- Processing/Computation layer (MapReduce), and
- Storage layer (Hadoop Distributed File System).

Hadoop | History or Evolution

Hadoop is an open source framework overseen by Apache Software Foundation which is written in Java for storing and processing of huge datasets with the cluster of commodity hardware. There are mainly two problems with the big data. First one is to store such a huge amount of data and the second one is to process that stored data. The traditional approach like RDBMS is not sufficient due to the heterogeneity of the data. So Hadoop comes as the solution to the problem of big data i.e. storing and processing the big data with some extra capabilities. There are mainly two components of Hadoop which are **Hadoop Distributed File System (HDFS)** and **Yet Another Resource Negotiator(YARN)**.

Hadoop History

Hadoop was started with **Doug Cutting and Mike Cafarella** in the year 2002 when they both started to work on Apache Nutch project. Apache Nutch project was the process of building a search engine system that can index 1 billion pages. After a lot of research on Nutch, they concluded that such a system will cost around half a million dollars in hardware, and along with a monthly running cost of $30, 000 approximately, which is very expensive. So, they realized that their project architecture will not be capable enough to the workaround with billions of pages on the web. So they were looking for a feasible solution which can reduce the implementation cost as well as the problem of storing and processing of large datasets.

In 2003, they came across a paper that described the architecture of Google's distributed file system, called **GFS (Google File System)** which was published by Google, for storing the large data sets. Now they realize that this paper can solve their problem of storing very large files which were being generated because of web crawling and indexing processes. But this paper was just the half solution to their problem.

In 2004, Google published one more paper on the technique **MapReduce**, which was the solution of processing those large datasets. Now this paper was another half solution for Doug Cutting and Mike

Cafarella for their Nutch project. These both techniques (GFS & MapReduce) were just on white paper at Google. Google didn't implement these two techniques. Doug Cutting knew from his work on Apache Lucene (It is a free and open-source information retrieval software library, originally written in Java by Doug Cutting in 1999) that open-source is a great way to spread the technology to more people. So, together with Mike Cafarella, he started implementing Google's techniques (GFS & MapReduce) as open-source in the Apache Nutch project.
In 2005, Cutting found that Nutch is limited to only 20-to-40 node clusters. He soon realized two problems:
(a) Nutch wouldn't achieve its potential until it ran reliably on the larger clusters
(b) And that was looking impossible with just two people (Doug Cutting & Mike Cafarella).
The engineering task in Nutch project was much bigger than he realized. So he started to find a job with a company who is interested in investing in their efforts. And he found Yahoo!.Yahoo had a large team of engineers that was eager to work on this there project.
So **in 2006,** Doug Cutting joined Yahoo along with Nutch project. He wanted to provide the world with an open-source, reliable, scalable computing framework, with the help of Yahoo. So at Yahoo first, he separates the distributed computing parts from Nutch and **formed a new project Hadoop (He gave name Hadoop it was the name of a yellow toy elephant which was owned by the Doug Cutting's son. and it was easy to pronounce and was the unique word.**) Now he wanted to make Hadoop in such a way that it can work well on thousands of nodes. So with GFS and MapReduce, he started to work on Hadoop.
In 2007, Yahoo successfully tested Hadoop on a 1000 node cluster and start using it.
In January of 2008, Yahoo released Hadoop as an open source project to ASF(Apache Software Foundation). And in **July of 2008, Apache Software Foundation successfully tested a 4000 node cluster with Hadoop.**
In 2009, Hadoop was successfully tested to sort a PB (PetaByte) of data in less than 17 hours for handling billions of searches and indexing millions of web pages. And **Doug Cutting left the Yahoo and joined Cloudera to fulfill the challenge of spreading Hadoop to other industries.**
In **December of 2011,** Apache Software Foundation released **Apache Hadoop version 1.0.**
And later in Aug 2013, **Version 2.0.6 was available**.
And currently, we have **Apache Hadoop version 3.0** which released in **December 2017**.

Advantages of Hadoop:

1. Scalable

Hadoop is a highly scalable storage platform, because it can stores and distribute very large data sets across hundreds of inexpensive servers that operate in parallel. Unlike traditional relational database systems (RDBMS) that can't scale to process large amounts of data, Hadoop enables businesses to run applications on thousands of nodes involving many thousands of terabytes of data.

2. Cost effective

Hadoop also offers a cost effective storage solution for businesses' exploding data sets. The problem with traditional relational database management systems is that it is extremely cost prohibitive to scale to such a degree in order to process such massive volumes of data. In an effort to reduce costs, many companies in the past would have had to down-sample data and classify it based on certain assumptions as to which data was the most valuable. The raw data would be deleted, as it would be too cost-prohibitive to keep. While this approach may have worked in the short term, this meant that when business priorities changed, the complete raw data set was not available, as it was too expensive to store.

3. Flexible

Hadoop enables businesses to easily access new data sources and tap into different types of data (both structured and unstructured) to generate value from that data. This means businesses can use Hadoop to derive valuable business insights from data sources such as social media, email conversations. Hadoop can be used for a wide variety of purposes, such as log processing, recommendation systems, data warehousing, market campaign analysis and fraud detection.

4. Fast

Hadoop's unique storage method is based on a distributed file system that basically 'maps' data wherever it is located on a cluster. The tools for data processing are often on the same servers where the data is located, resulting in much faster data processing. If you're dealing with large volumes of unstructured data, Hadoop is able to efficiently process terabytes of data in just minutes, and petabytes in hours.

5. Resilient to failure

A key advantage of using Hadoop is its fault tolerance. When data is sent to an individual node, that data is also replicated to other nodes in the cluster, which means that in the event of failure, there is another copy available for use.

Disadvantages of Hadoop:

As the backbone of so many implementations, Hadoop is almost synomous with big data.

1. Security Concerns

Just managing a complex applications such as Hadoop can be challenging. A simple example can be seen in the Hadoop security model, which is disabled by default due to sheer complexity. If whoever managing the platform lacks of know how to enable it, your data could be at huge risk. Hadoop is also missing encryption at the storage and network levels, which is a major selling point for government agencies and others that prefer to keep their data under wraps.

2. Vulnerable By Nature

Speaking of security, the very makeup of Hadoop makes running it a risky proposition. The framework is written almost entirely in Java, one of the most widely used yet controversial programming languages in existence. Java has been heavily exploited by cybercriminals and as a result, implicated in numerous security breaches.

3. Not Fit for Small Data

While big data is not exclusively made for big businesses, not all big data platforms are suited for small data needs. Unfortunately, Hadoop happens to be one of them. Due to its high capacity design, the Hadoop Distributed File System, lacks the ability to efficiently support the random reading of small files. As a result, it is not recommended for organizations with small quantities of data.

4. Potential Stability Issues

Like all open source software, Hadoop has had its fair share of stability issues. To avoid these issues, organizations are strongly recommended to make sure they are running the latest stable version, or run it under a third-party vendor equipped to handle such problems.

5. General Limitations

The article introduces Apache Flume, MillWheel, and Google's own Cloud Dataflow as possible solutions. What each of these platforms have in common is the ability to improve the efficiency and reliability of data collection, aggregation, and integration. The main point the article stresses is that companies could be missing out on big benefits by using Hadoop alone.

Hadoop Versions:

Hadoop is a Software which on an open-source framework storing data using a distributed network rather than a centralized one thereby processing the data in a parallel transition. This enables Hadoop to act as one of the most reliable batch processing engine and layered storage and resource management system. As the data beings stored and processed increases in its complexity so do Hadoop where the developers bring out various versions to address the issues (bug fixes) and simplify the complex data processes. The updates are automatically implemented as Hadoop development follows the trunk (base code) – branch (fix)model. Hadoop has two versions: a) Hadoop 1.x (Version 1) and b) Hadoop 2 (Version 2)

Below are the two Hadoop Versions:

- Hadoop 1.x (Version 1)
- Hadoop 2 (Version 2)

1. Hadoop 1.x

Below are the Components of 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.

5. The next component if the Hadoop Distributed File System commonly known as HDFS, which plays the role of a distributed storage system that is designed to cater to large data, with a block size of 64 MegaBytes (64MB) for supporting the architecture. It is further divided into two components:

- Name Node which is used to store metadata about the Data node, placed with the Master Node. They contain details like the details about the slave note, indexing and their respective locations along with timestamps for timelining.
- Data Nodes used for storage of data related to the applications in use placed in the Slave Nodes.

6. Hadoop 1 uses Map Reduce (MR) data processing model It is not capable of supporting other non-MR tools.

MR has two components:

- Job Tracker is used to assigning or reassigning task-related (in case scenario fails or shutdown) to MapReduce to an application called task tracker is located in the node clusters. It additionally maintains a log about the status of the task tracker.
- The Task Tracker is responsible for executing the functions which have been allocated by the job tracker and sensor cross the status report of those task to the job tracker.

7. The network of the cluster is formed by organizing the master node and slave nodes. Which of this cluster is further divided into tracks which contain a set of commodity computers or nodes.

8. Whenever a large storage operation for big data set is is received by the Hadoop system, the data is divided into decipherable and organized blocks that are distributed into different nodes.

2. Hadoop Version 2

Version 2 for Hadoop was released to provide improvements over the lags which the users faced with version 1. Let's throw some light over the improvements that the new version provides:

- HDFS Federation which has improved to provide for horizontal scalability for the name node. Moreover, the namenode 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 NameNode.
- YARN stands for Yey 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 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.

Hadoop Ecosystem

Overview: Apache Hadoop is an open source framework intended to make interaction with **big data** easier, However, for those who are not acquainted with this technology, one question arises that what is big data ? Big data is a term given to the data sets which can't be processed in an efficient manner with the help of traditional methodology such as RDBMS. Hadoop has made its place in the industries and companies that need to work on large data sets which are sensitive and needs efficient handling. Hadoop is a framework that enables processing of large data sets which reside in the form of clusters. Being a framework, Hadoop is made up of several modules that are supported by a large ecosystem of technologies.

Introduction: *Hadoop Ecosystem* is a platform or a suite which provides various services to solve the big data problems. It includes Apache projects and various commercial tools and solutions. There are *four major elements of Hadoop* i.e. **HDFS, MapReduce, YARN, and Hadoop Common**. Most of the tools or solutions are used to supplement or support these major elements. All these tools work collectively to provide services such as absorption, analysis, storage and maintenance of data etc.

Following are the components that collectively form a Hadoop ecosystem:

- **HDFS:** Hadoop Distributed File System
- **YARN:** Yet Another Resource Negotiator
- **MapReduce:** Programming based Data Processing
- **Spark:** In-Memory data processing
- **PIG, HIVE:** Query based processing of data services
- **HBase:** NoSQL Database
- **Mahout, Spark MLLib:** Machine Learning algorithm libraries
- **Solar, Lucene:** Searching and Indexing
- **Zookeeper:** Managing cluster
- **Oozie:** Job Scheduling

Note: Apart from the above-mentioned components, there are many other components too that are part of the Hadoop ecosystem.

All these toolkits or components revolve around one term i.e. *Data*. That's the beauty of Hadoop that it revolves around data and hence making its synthesis easier.

HDFS:

- HDFS is the primary or major component of Hadoop ecosystem and is responsible for storing large data sets of structured or unstructured data across various nodes and thereby maintaining the metadata in the form of log files.
- HDFS consists of two core components i.e.
    1. Name node
    2. Data Node
- Name Node is the prime node which contains metadata (data about data) requiring comparatively fewer resources than the data nodes that stores the actual data. These data nodes are commodity hardware in the distributed environment. Undoubtedly, making Hadoop cost effective.

- HDFS maintains all the coordination between the clusters and hardware, thus working at the heart of the system.

YARN:

- Yet Another Resource Negotiator, as the name implies, YARN is the one who helps to manage the resources across the clusters. In short, it performs scheduling and resource allocation for the Hadoop System.
- Consists of three major components i.e.
    1. Resource Manager
    2. Nodes Manager
    3. Application Manager
- Resource manager has the privilege of allocating resources for the applications in a system whereas Node managers work on the allocation of resources such as CPU, memory, bandwidth per machine and later on acknowledges the resource manager. Application manager works as an interface between the resource manager and node manager and performs negotiations as per the requirement of the two.

MapReduce:

- By making the use of distributed and parallel algorithms, MapReduce makes it possible to carry over the processing's logic and helps to write applications which transform big data sets into a manageable one.
- MapReduce makes the use of two functions i.e. Map() and Reduce() whose task is:
    1. *Map()* performs sorting and filtering of data and thereby organizing them in the form of group. Map generates a key-value pair based result which is later on processed by the Reduce() method.
    2. *Reduce()*, as the name suggests does the summarization by aggregating the mapped data. In simple, Reduce() takes the output generated by Map() as input and combines those tuples into smaller set of tuples.

PIG:

- Pig was basically developed by Yahoo which works on a pig Latin language, which is Query based language similar to SQL.
- It is a platform for structuring the data flow, processing and analyzing huge data sets.
- Pig does the work of executing commands and in the background, all the activities of MapReduce are taken care of. After the processing, pig stores the result in HDFS.
- Pig Latin language is specially designed for this framework which runs on Pig Runtime. Just the way Java runs on the JVM.

- Pig helps to achieve ease of programming and optimization and hence is a major segment of the Hadoop Ecosystem.

HIVE:

- With the help of SQL methodology and interface, HIVE performs reading and writing of large data sets. However, its query language is called as HQL (Hive Query Language).
- It is highly scalable as it allows real-time processing and batch processing both. Also, all the SQL datatypes are supported by Hive thus, making the query processing easier.
- Similar to the Query Processing frameworks, HIVE too comes with two components: *JDBC Drivers* and *HIVE Command Line*.
- JDBC, along with ODBC drivers work on establishing the data storage permissions and connection whereas HIVE Command line helps in the processing of queries.

Mahout:

- Mahout, allows Machine Learnability to a system or application. Machine Learning, as the name suggests helps the system to develop itself based on some patterns, user/environmental interaction or om the basis of algorithms.
- It provides various libraries or functionalities such as collaborative filtering, clustering, and classification which are nothing but concepts of Machine learning. It allows invoking algorithms as per our need with the help of its own libraries.

Apache Spark:

- It's a platform that handles all the process consumptive tasks like batch processing, interactive or iterative real-time processing, graph conversions, and visualization, etc.
- It consumes in memory resources hence, thus being faster than the prior in terms of optimization.
- Spark is best suited for real-time data whereas Hadoop is best suited for structured data or batch processing, hence both are used in most of the companies interchangeably.

Apache HBase:

- It's a NoSQL database which supports all kinds of data and thus capable of handling anything of Hadoop Database. It provides capabilities of Google's BigTable, thus able to work on Big Data sets effectively.
- At times where we need to search or retrieve the occurrences of something small in a huge database, the request must be processed within a short quick span of time. At such times, HBase comes handy as it gives us a tolerant way of storing limited data.

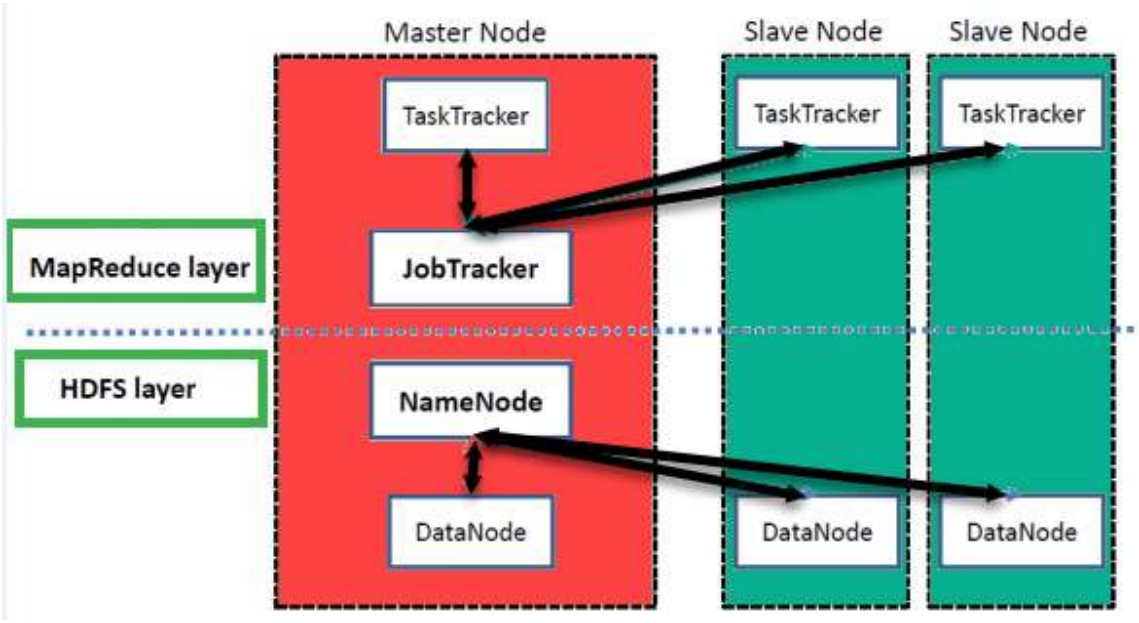
Other Components: Apart from all of these, there are some other components too that carry out a huge task in order to make Hadoop capable of processing large datasets. They are as follows:

- **Solr, Lucene:** These are the two services that perform the task of searching and indexing with the help of some java libraries, especially Lucene is based on Java which allows spell check mechanism, as well. However, Lucene is driven by Solr.
- **Zookeeper:** There was a huge issue of management of coordination and synchronization among the resources or the components of Hadoop which resulted in inconsistency, often. Zookeeper overcame all the problems by performing synchronization, inter-component based communication, grouping, and maintenance.
- **Oozie:** Oozie simply performs the task of a scheduler, thus scheduling jobs and binding them together as a single unit. There is two kinds of jobs .i.e Oozie workflow and Oozie coordinator jobs. Oozie workflow is the jobs that need to be executed in a sequentially ordered manner whereas Oozie Coordinator jobs are those that are triggered when some data or external stimulus is given to it.

Hadoop vs RDBMS

Hadoop software framework work is very well structured semi-structured and unstructured data. This also supports a variety of data formats in real-time such as XML, JSON, and text-based flat file formats. RDBMS works efficiently when there is an entity-relationship flow that is defined perfectly and therefore, the database schema or structure can grow and unmanaged otherwise. i.e., An RDBMS works well with structured data. Hadoop will be a good choice in environments when there are needs for big data processing on which the data being processed does not have dependable relationships.

Hadoop Architecture



High Level Hadoop Architecture

Hadoop has a Master-Slave Architecture for data storage and distributed data processing using MapReduce and HDFS methods.

NameNode:

NameNode represented every files and directory which is used in the namespace

DataNode:

DataNode helps you to manage the state of an HDFS node and allows you to interacts with the blocks

MasterNode:

The master node allows you to conduct parallel processing of data using Hadoop MapReduce.

Slave node:

The slave nodes are the additional machines in the Hadoop cluster which allows you to store data to conduct complex calculations. Moreover, all the slave node comes with Task Tracker and a DataNode. This allows you to synchronize the processes with the NameNode and Job Tracker respectively.

In Hadoop, master or slave system can be set up in the cloud or on-premise

Features Of 'Hadoop'

• Suitable for Big Data Analysis

As Big Data tends to be distributed and unstructured in nature, HADOOP clusters are best suited for analysis of Big Data. Since it is processing logic (not the actual data) that flows to the computing nodes, less network bandwidth is consumed. This concept is called as **data locality concept** which helps increase the efficiency of Hadoop based applications.

• Scalability

HADOOP clusters can easily be scaled to any extent by adding additional cluster nodes and thus allows for the growth of Big Data. Also, scaling does not require modifications to application logic.

• Fault Tolerance

HADOOP ecosystem has a provision to replicate the input data on to other cluster nodes. That way, in the event of a cluster node failure, data processing can still proceed by using data stored on another cluster node.