HADOOP AND ITS ECOSYSTEM

Under the guidance of Prof. Yann Chang

IDS 521: Advanced Database Management

University of Illinois at Chicago

- Abhijeet Maheshwari

- Ankita Singla

- Devesh Sharma

Contents:

Executive Summary:

We have seen in recent years how Big data has evolved, enabling us to extract valuable knowledge from petabytes of data. This exploding size of the data has led to the development of distributed and parallel computing solutions. Apache Hadoop has been the biggest player, attracting a lot of users towards its capability of processing, analyzing and transforming Big data. HDFS and MapReduce being the main components of Hadoop architecture, enable parallel computation across thousands of hosts by partitioning the data. Hadoop enables easy scalability of clusters at a comparatively low cost for petabytes of data. Apache Hive is the data warehouse software built on top of Hadoop, providing SQL-like interface for data processing on file systems that integrate with Hadoop. While Hive provides this SQL compatibility, Apache Pig provides high-level platform for creating programs that run on Apache Hadoop, using Pig Latin language.

Since Apache does not provide support for scalability and has stability/security concerns, there arises a need for commercial vendors to help distribute Hadoop across various platforms. A few such vendors are Cloudera, Hortonworks, MapR. They ensure Hadoop is reliable, supported and complete.

1. Introduction to Hadoop

1.1 Hadoop – Origin and Overview

How did Hadoop originate?

Hadoop, the widely used text search library, was created by Doug Cutting, the creator of Apache Lucene, and Mike Cafarella in 2005. Hadoop has its origins in Apache Nutch, an open source web search engine, itself a part of the Lucene project.

Doug Cutting explained how it was a made-up name and not an acronym. In his own words, "*The name my kid gave a stuffed yellow elephant. Short, relatively easy to spell and pronounce, meaningless, and not used elsewhere: those are my naming criteria. Kids are good at generating such. Googol is a kid's term.*" Names of Sub projects and 'contrib' modules in Hadoop are also unrelated to their function, often after an elephant or other animal theme (for example: 'Pig').

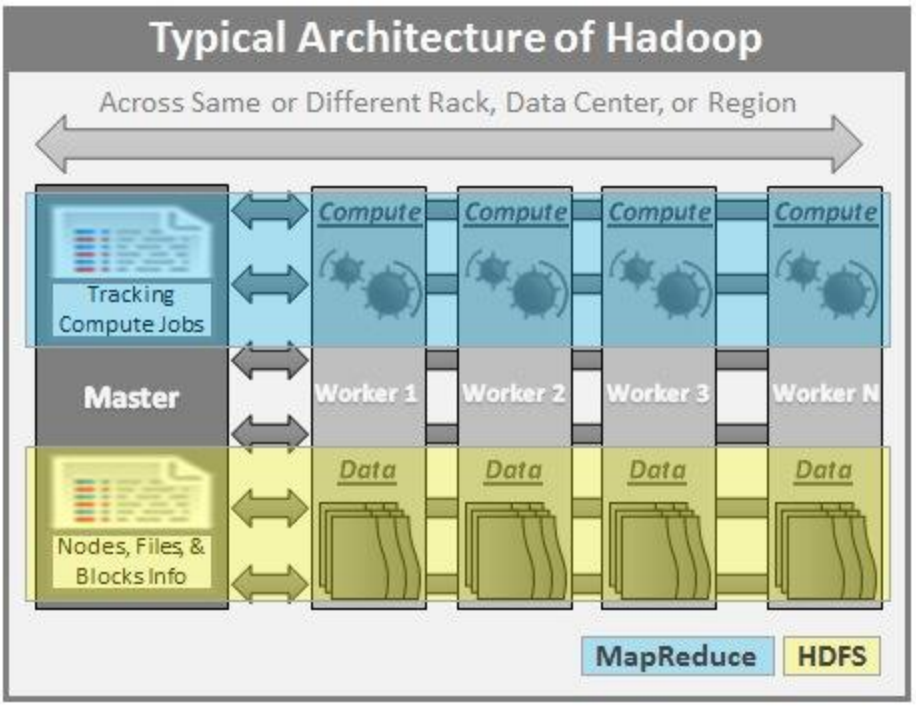
Nutch started operating in 2002, followed by working crawler and search system but the problem in front of them was their architecture not being able to scale to the billions of pages on the web. A ray of hope appeared for them when Google published their paper on Google's distributed filesystem, called GFS in 2003. It would free up time spent on administrative tasks such as managing storage nodes. After Google published their paper on MapReduce in 2004, Nutch developers had a working MapReduce implementation in Nutch and by mid-2005, all major Nutch algorithms were ported to run using MapReduce and NDFS.

In January 2008, Hadoop made its own top-level project at Apache, ensuring its success and diversity. It got licensed under Apache License 2.0. In April 2008, Hadoop broke a world record, becoming the fastest system to store a terabyte of data.

What is Hadoop?

Hadoop is an open source framework, capable of processing large amounts of heterogenous datasets in a distributed fashion. It provides a reliable shared storage and analysis system. Its framework is based on the following principle: "*In pioneer days they used oxen for heavy pulling, and when one ox couldn't budge a log, they didn't try to grow a larger ox. We shouldn't be trying for bigger computers, but for more systems of computers.* ***~Grace Hopper***".

What is Hadoop's architecture?



Few highlights of the Hadoop architecture:

- It works in a master-worker/ master-slave fashion
- It has 2 core components: HDFS and MapReduce

- **HDFS (Hadoop Distributed File System):** It replicates the data across multiple nodes, offering a highly reliable and distributed storage. Unlike a regular file system, it automatically splits the input data into multiple blocks, storing it across various data nodes. This ensures high availability and fault tolerance
- **MapReduce:** It provides an analysis system capable of performing complex computations on large datasets. It breaks these complex calculations into multiple tasks, assigning them to individual slave nodes and then takes care of coordinating and consolidating the result
- **The Master** consists of Namenode and Job Tracker components:
    - **Namenode** holds information useful for the operation of the Hadoop cluster. It contains information about all nodes in Hadoop cluster, files present in the cluster, building blocks of those files and their locations in the cluster
    - **Job Tracker** tracks individual tasks/ jobs assigned to all nodes and coordinates information exchange and results
- Each **Worker/ Slave** block in the framework shown above, contains the Task Tracker and a Datanode component:
    - **Task Tracker** runs the task assigned to it
    - **Datanode** holds the data
- There is no dependency on the physical server's location

1.2 Hadoop Characteristics:

- **Distributed Processing:** Data is processed in parallel on a cluster of nodes, given the data storage in a distributed manner in HDFS

- **Faster:** As mentioned above, due to parallel processing capability of Hadoop, it is extremely good at high-volume batch processing. It performs multiple times faster than a single thread server or on the mainframe
- **Fault Tolerance:** Since the data sent to one node is replicated in the other nodes of the same cluster, in case the original node fails to process the data, other nodes would process it
- **Reliability:** Again, because of the above characteristic of data replication, data is reliably stored on machine clusters, protecting it from machine failures
- **High Availability:** Due to these multiple copies of data, it is highly available and accessible from another path, even if hardware fails
- **Scalability:** Hadoop is highly scalable because it can store and distribute very large data sets across multiple parallel operating servers. It allows businesses to run processes involving thousands of terabytes of data. One big feature is its capability to add nodes during processing without any downtime, called hardware horizontal scalability
- **Flexibility:** Hadoop can handle both structured and unstructured, encoded or formatted type of data. It is of great value when business decision making requires handling unstructured data
- **Economic:** Since Hadoop runs on cluster of commodity hardware, it offers a cost-effective storage solution against gigantic data
- **Easy to Use:** It is relatively very easy to use since it takes care of distributed computing

- **Data Locality:** When a new MapReduce algorithm is submitted, it is moved to data in the cluster rather than moving the data to the location where the algorithm was submitted and then processing it

1.3 When to Use Hadoop:

- **Data Size and Data Diversity:** When there is huge volume of data coming in from multiple sources in a variety of formats, then Hadoop is the right technology to be used for such Big Data
- **Future Planning:** Hadoop supports us to do future cluster planning. If we need to implement Hadoop on our data, we should first understand its level complexity and the rate at which is going to grow. Accordingly, we can scale our cluster
- **Multiple Frameworks for Big Data:** When we need to various tools for various purposes, Hadoop is the go-to technology because it easily integrates with multiple tools like Mahout, R, Python, Spark, MongoDB and Hbase for Nosql database, etc.
- **Lifetime Data Availability:** When we want our data to be live and running forever, Hadoop's scalability comes in handy. There is no limit to cluster size, as and when requirement is there, we can add data nodes to the cluster at very minimal cost

1.4 When Not to Use Hadoop:

- **Real Time Analytics:** Since Hadoop works on batch processing, its response time is high and therefore it should not be used while working on real time analytics. To overcome this, we can use Spark mounted over Hadoop which makes it 100 times faster

- **Replacing Existing Infrastructure:** We cannot replace existing database with Hadoop but can use it parallelly. They are 2 different tools for different jobs. Hadoop can store and process the data, which can be passed into the relational database for BI, reporting, etc.
- **Multiple Smaller Datasets:** For smaller database, we should use tools like MS Excel because Hadoop would be both slower and costlier than these technologies, when implemented for smaller dataset. Although, if we can combine these smaller datasets (them being of exactly same format and type) and create a big data, then Hadoop can be used to store and process this data
- **Security, being the Main Concern:** When companies are dealing with sensitive data, they cannot move to Hadoop and Big data projects because of data breach possibility. If they want to use Hadoop, they would be required to encrypt their data and then load into Hadoop for further processing.

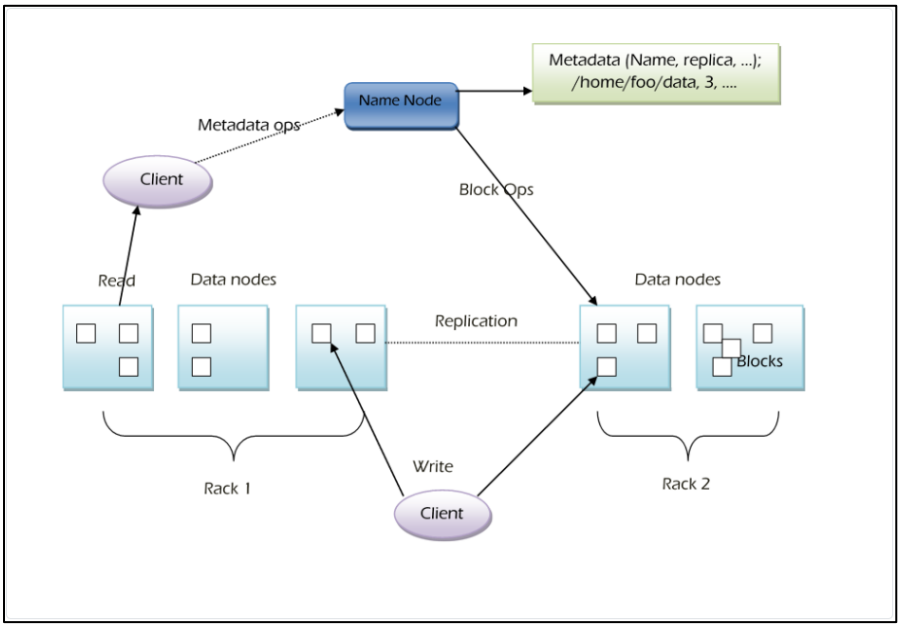
2. Key Technologies:

2.1 Hadoop Distributed File Systems (HDFS):

What is the HDFS?

It is a distributed file system, following network-based approach to store files across systems. It is designed to work on product hardware and handle very large amount of data in Terabytes and Petabytes. It is designed to deploy on low cost hardware. It is a highly fault tolerant and self-healing distributed file system.

What is the HDFS's Architecture?



Hadoop follows a Master-Slave architecture, comprising of majorly 2 daemons (back-ground service that runs on Hadoop): Master and Slave Daemons. Master Daemons contain Name Node, Secondary Name Node and Job Tracker whereas Slave Daemons contain Data Node and Task Tracker. Name node keeps the directory tree of all files in the file system and tracks where the file is kept across the cluster. It maintains the file system and the file system further contains all the meta data. It is a single point of failure for HDFS cluster, when it goes down, the entire file system goes offline. Whenever the primary node is down, secondary name node comes into the picture. It helps to keep the file size containing HDFS modifications log within certain limits at name node. Task Tracker instantiates and monitors individual map and reduces work. Every task tracker has certain allocated slot which showcases how many tasks it can accommodate. It resides on top of data nodes. Job tracker takes care of scheduling and re-scheduling the tasks in the form of Map reduce jobs. It resides on top of the Name Node, managing map reduce tasks and distributing individual tasks to task tracker machine. Data Node stores the data in Hadoop

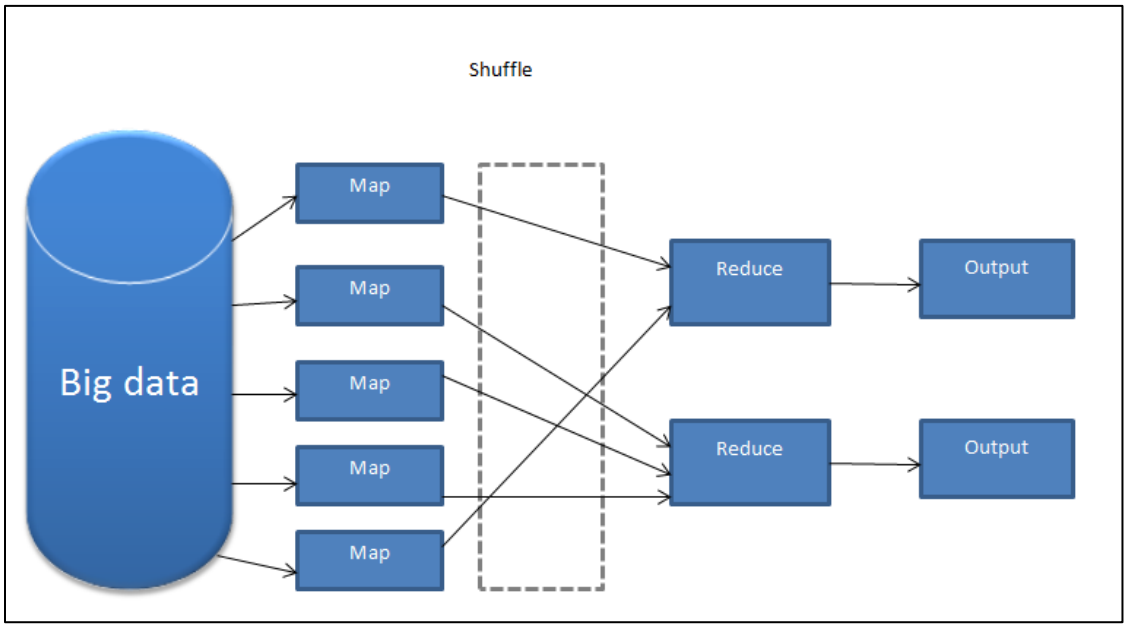
file system in the form of HDFS blocks, having default block size of 64MB. Initially, Data Node connects to the Name Node and establishes the service and then responds to the requests back from Name Node for file system operations.

2.2 MapReduce:

What is the MapReduce?

MapReduce is a parallel programming model for processing the huge amount of data. It allows writing applications to process massive amounts of data in parallel on large clusters. It provides automatic parallelization and distributed fault-tolerance, I/O scheduling, monitoring and status updates. Computational process occurs on both structured and unstructured data, making it fault-tolerant, reliable and supporting thousands of nodes.

What is the MapReduce's Algorithm and Programming Model?



A MapReduce job is divided into 4 phases:

- **Input Splits:** Input to a MapReduce job is divided into fixed-size pieces called input splits which is a chunk of input being consumed by a single map
- **Mapping:** Data in each split is passed to a mapping function to produce output values. Whatever job is to be done, would be executed over here
- **Shuffling:** It takes the output from Mapping phase and consolidates the relevant records for further processing
- **Reducing:** Here, the output from Shuffling phase is aggregated and returns a single value as output, summarizing the complete dataset

MapReduce cannot control the order in which maps or reductions are run, although a reduce job would not take place until all maps are complete.

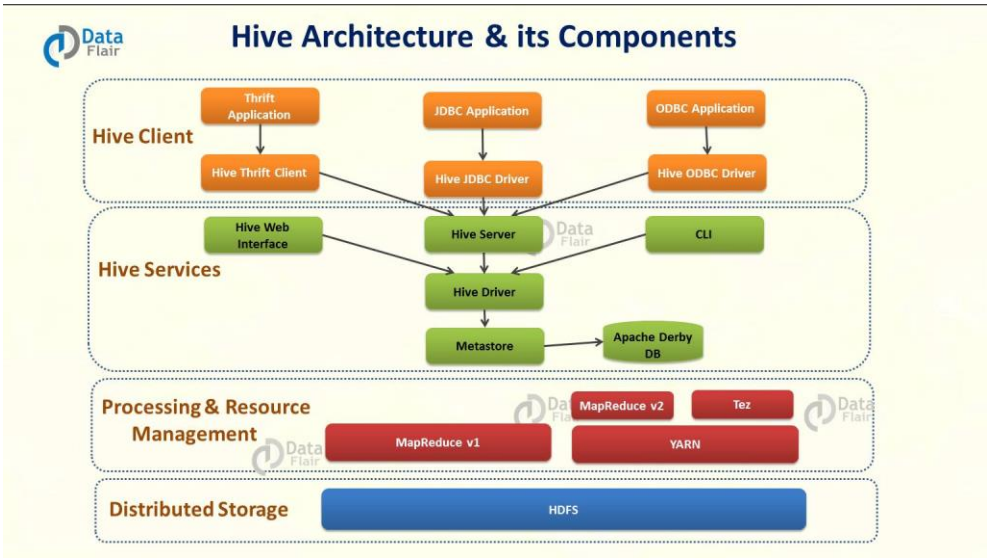
MapReduce algorithm facilitates parallelization, distribution, fault tolerance, status and monitoring the tools. These programs are generally written in Java or any other scripting language using Streaming API.

3.Projects on Hadoop

3.1 Hive:

Hive is a data warehouse open - source solution built on top of Hadoop. Hive supports queries in a SQL-like declarative language called HiveQL or HQL that are compiled into Hadoop-based MapReduce jobs. HiveQL also allows users to connect customized map scripts to queries.

Architecture & Components



The above diagram describes the flow in which a query is submitted into Hive and finally processed using the MapReduce framework.

As depicted in the above diagram the major components of Apache Hive are:

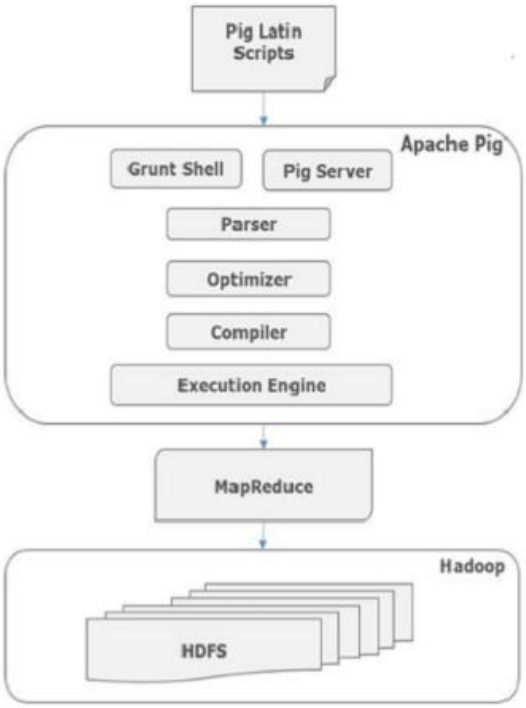
- **Hive Clients** – Apache Hive is very versatile as it supports all application written in languages like C++, Java, Python etc. using JDBC, Thrift and ODBC drivers. Thus, one can easily develop a Hive client application written in a language of their choice.
- **Hive Services** – Hive provides various services like web Interface, CLI etc. to perform queries.
- **Processing framework and Resource Management** – Hive internally uses Hadoop MapReduce framework to execute the queries.
- **Distributed Storage** – Hive is built on the top of Hadoop, so it uses the underlying HDFS for the distributed storage.

3.2 Apache Pig

Apache Pig is the language used for analyzing data in Hadoop using pig. It is a high - level data processing language that offers a wide range of data types and operators for different data processing operations.

In order to perform a specific task, programmers must write a Pig script using the Pig Latin language and execute it using any of the execution mechanisms (Grunt Shell, UDFs, Embedded). After execution, these scripts will undergo a series of transformations applied to produce the desired output by the Pig Framework.

Internally, Apache Pig converts these scripts to a number of MapReduce jobs, making the job easy for the programmer. The Apache Pig architecture is shown below.



Component of apache pig:

- **Parser**

The Pig Scripts are initially handled by the Parser. It checks the script syntax, types checks and other different checks. The parser output is a DAG (directed acyclic graph) that represents the statements of Pig Latin and the logical operators. The DAG represents the logical operators of the script as the nodes and the data flows as the edges.

- **Optimizer**

The logical plan (DAG) is transferred to the logical optimizer that performs logical optimizations such as projection and pushing.

- **Compiler**

The compiler compiles the optimized logical plan into a series of MapReduce jobs.

- **Execution engine**

The MapReduce jobs are finally submitted in a sorted order to Hadoop. Those MapReduce jobs are finally performed on Hadoop to produce the desired results.

3.3 HBase:

HBase is a column-oriented database management system with a Hadoop Distributed File System (HDFS) system. It is suitable for sparse data sets that are common in many cases of big data use. HBase doesn’t support a structured query language like SQL. HBase isn’t a relational data store at all. HBase applications are written in Java much like a typical Apache™ MapReduce application.

An HBase system comprises of a set of tables. Each table contains rows and columns, much like a traditional database. Each table must have a Primary Key, and all access attempts to HBase tables must use this Primary Key.

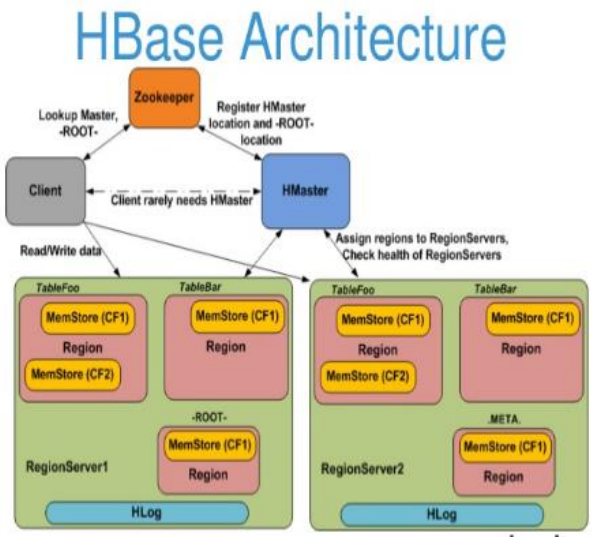


Image Sources:Cloudera

HBase architecture has 3 important components- HMaster, Region Server.

I. HMaster

The process that assigns regions to region servers in the Hadoop cluster for load balancing is known as **HBase HMaster**. Responsibilities of HMaster :–

- Monitors and Manages the Hadoop Cluster.
- Performs Administration (Interface for creating, updating and deleting tables.)
- All the DDL operations are handled by the HMaster
- HMaster is responsible for all the changes in the schema as well as any metadata Operations.

II. Region Server

Regional servers are working nodes that handle client requests to read, write, update and delete. Region Server Process runs in the hadoop cluster on every node. The region server is running on HDFS DataNode and includes the following components –

- Block Cache – This is the cache read. The most frequently read data is stored in the read cache and when the block cache is complete, the data that is recently used is disposed.
- MemStore: This is the write cache and stores new data that has not been written to the disk yet. Every column family has a MemStore in a region
- Write Ahead Log (WAL): It is a file that stores new data that are not permanently stored.
- HFile is the actual storage file which saves rows on a disk as sorted key values.

III. Zookeeper

HBase uses ZooKeeper as a distributed coordinating service for regional assignments and recovers any regional server crashes by loading them to other functioning region servers. ZooKeeper is a centralized monitoring server that maintains configuration and distributes synchronization information.

Whenever a customer wants to contact regions, he must first approach Zookeeper. HMaster and Region servers are registered with ZooKeeper service, clients need ZooKeeper quorum to connect to region servers and HMaster servers.

4. Hadoop Vendors providing Big Data Solutions

The data we create every day is enormous and in recent years its speed has reached its ultimate level, resulting in an increase of almost 90 percent in the data size. The attributes of high variety, speed and volume have increased the number of Hadoop vendors. As the big data technologies increase, their demands are growing rapidly. It has a revolutionary venture information administration and a great structural design. Cloud and venture merchants are on the threshold of competing with the best Vendors. The core components of the free source big data tools are HDFS, MapReduce, YARN and Common.

The following are the recent upgradations by the vendor circulations:

- Support which assists with technical solutions and turns it simple for users at various levels.
- They are consistent for a swift response to patches, fixes and bug detection.
- They also give an opportunity for extra add-on instruments to customize their apps for users.

4.1 Need for Commercial Hadoop Vendors:

While improvements are constantly being made, like all open source technologies, Apache Hadoop has its share of stability problems or minor security concerns. Not every big data platform is suitable for small data requirements. Sadly, Hadoop is one of them. To avoid such problems, organizations have developed distributions of Hadoop.

- **Support:**

  Hadoop vendors provide assistance and technical guidance in order to facilitate the adoption of Hadoop for their big data problems.

- **Reliability**:

  Whenever a bug is detected, Hadoop vendors react promptly and solve it and provide customers with a reliable resource.

- **Completeness**:

  Hadoop suppliers partner with other distributions and provide additional tools to help consumers customize Hadoop applications to fulfill their specific tasks.

4.2 Hadoop Vendors Market Share

Hadoop suppliers' partner with other distributions and provide additional tools to help consumers customize Hadoop applications to fulfill their specific tasks.

Worldwide Big Data software and services market revenues are projected to increase from $ 42B in 2018 to $ 103B in 2027, with a compound annual growth rate (CAGR) of 10.48%. As part of this forecast, Wikibon estimates that the global big data market is growing at 11.4% CAGR between

2017 and 2027, from $ 35 billion to $ 103 billion.

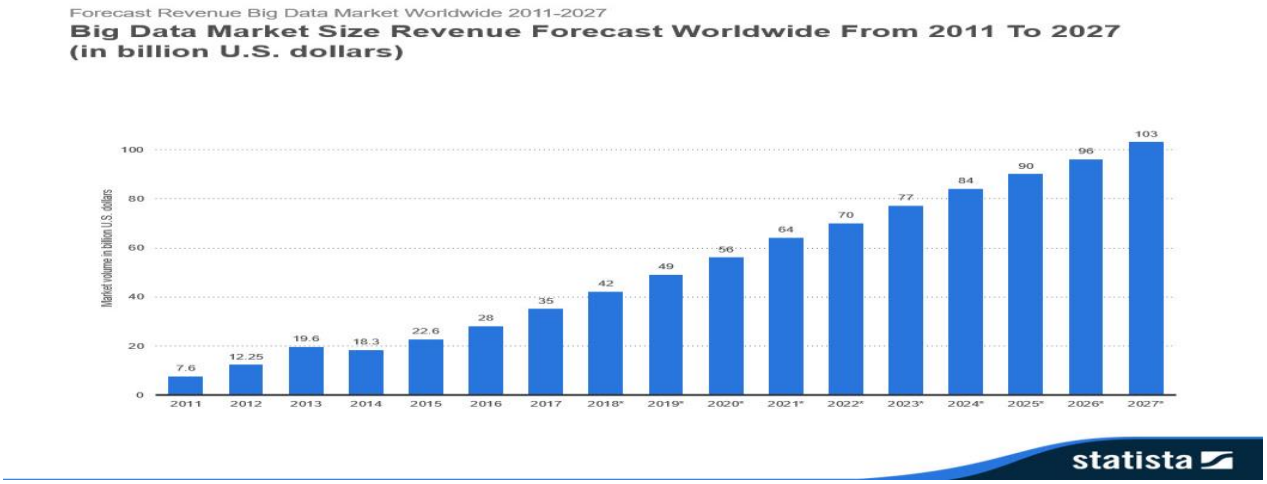


Image source: Wikibon and reported by Statista.

The width of the sector for the respective vendor is their market share in comparison with others. As the earliest Hadoop distributions, Cloudera has the largest user base and a higher market share, as shown in the figure. The bold companies are active and leading suppliers on the Hadoop market.

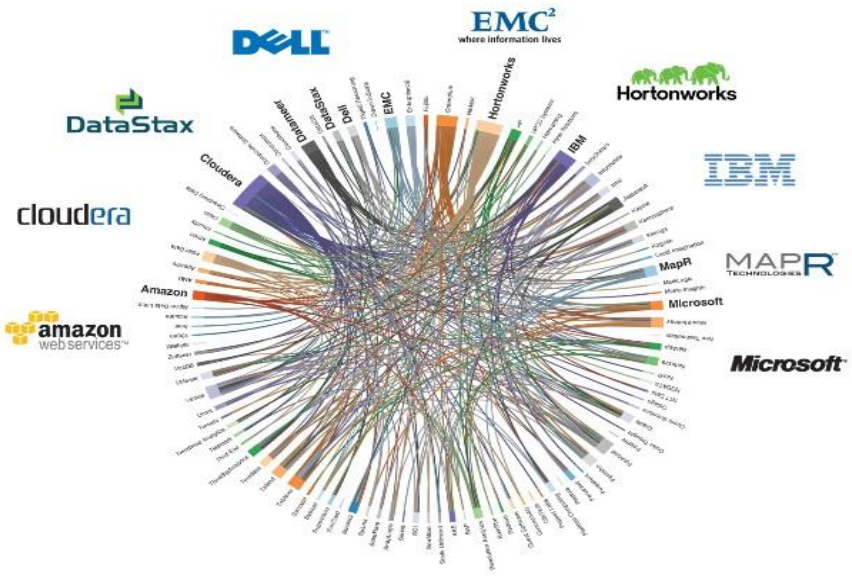


Image Source: randomramblings.postach.io

The top 6 vendors offering Big Data Hadoop solution are:

- **Cloudera**

This tops amongst the big data vendors to make Hadoop a reliable data platform for big data. The Cloudera Hadoop vendor has about 350 + customers, including the US Army, All State and Monsanto. Cloudera occupies 53 percent of the Hadoop market.

- **Amazon Web Services Elastic MapReduce Hadoop Distribution**

Amazon Elastic MapReduce is part of Amazon Web Services(AWS) and has been active since Hadoop 's earliest days. I.e. AWS. Amazon Elastic MapReduce Elastic MapReduce loads a simple to use data analysis stand built on an influential structural design of HDFS.

- **HortonWorks**

Hortonworks is one of the leading suppliers because it promises an open source distribution of 100 percent. It acquires companies that provide or meet company gaps and contribute code to the Apache project community instantly.

- **Microsoft Hadoop Distribution**

Based on the current Hadoop distribution strategy of the Big Data vendors and the market presence, it is an IT company that is not prominent for free foundation software solutions. Microsoft offers it as a manufactured goods from a community cloud.

Another special feature of Microsoft is that the Polybase feature helps customers track data on the SQL server during the query implementation.

- **MapR**

  MapR technologies were designed to enable Hadoop to perform well with minimal effort and potential. MapR Technologies linchpin, the HDFS API inheriting MapR filesystem, is fully read / written and can save trillions of files. MapR delivered reliable and efficient distribution for the large cluster implementation predominantly than any other vendor.

- **IBM InfoSphere Insights**

IBM assimilates key data management parts and analytics assets into open - source distribution. The company has also launched an open - source project, Apache System ML for machine learning. IBM Big Insights enables customers to market their apps to integrate advanced Big Data analytics in a very rapid manner.

5. Cloudera Hadoop Distribution

Cloudera distribution including Apache Hadoop provides an analytics platform and the latest open source technologies to store, process, discover, model and serve large amounts of data. By integrating Hadoop with more than a dozen other critical open source projects, Cloudera has created a functionally advanced system that helps perform end-to-end Big Data workflows. It ensures that all independent ecosystems such as Hive, Pig etc co-exist as independent clusters

and are provided to consumers as Cloudera Hadoop Distribution(CDH). Unlike Apache, Cloudera and others provide commercial support for their own versions of Hadoop.

As of 2017, Cloudera was the leading Hadoop distribution vendor with 53% of market share while Hortonworks had 16% of the market share. In October 2018, the two open-source and commercial-license software distribution companies announced an all-stock merger of equals.

Cloudera provides its services both on-premise and on cloud. Various bundles provided are:

- **Cloudera Enterprise Data Hub** - Cloudera's comprehensive data management platform including all of Data Science & Engineering, Operational DB, Analytic DB, and Cloudera Essentials.
- **Cloudera Analytic DB** - Cloudera's technologies that enable fast, flexible, and scalable Business Intelligence (BI) and SQL analytics built on the core Cloudera Essentials platform.
- **Cloudera Operational DB** - Cloudera's high-scale NoSQL technologies for real-time, data applications built on the core Cloudera Essentials platform.
- **Cloudera Data Science and Engineering** - Cloudera's technologies that enable efficient, high-scale data processing, data science, and machine learning on top of the Core Essentials platform.
- **Cloudera Essentials** - Cloudera's core data management platform for fast, easy, and secure large-scale data processing that includes Cloudera's enterprise-ready management capabilities (Cloudera Manager) and open source platform distribution (CDH).

CDH OVERVIEW:

CDH is the most complete, tested, and popular distribution of Apache Hadoop and related projects that delivers the core elements of Hadoop – scalable storage and distributed computing – along with a Web-based user interface and vital enterprise capabilities. CDH is Apache-licensed open source and is the only Hadoop solution to offer unified batch processing, interactive SQL and interactive search, and role-based access controls.

CDH provides:

- **Flexibility**: Store any type of data and manipulate it with a variety of different computation frameworks including batch processing, interactive SQL, free text search, machine learning and statistical computation.
- **Integration:** Get up and running quickly on a complete Hadoop platform that works with a broad range of hardware and software solutions.
- **Security**: Process and control sensitive data.
- **Scalability**: Enable a broad range of applications and scale and extend them to suit your requirements.
- **High availability:** Perform mission-critical business tasks with confidence.
- **Compatibility:** Leverage your existing IT infrastructure and investment.

CDH meets enterprise demands with its 100% open source distribution over Hadoop framework. It integrates Apache Hadoop with critical open source projects and creates a functionally advanced system that helps perform end-to-end Big Data workflows.

The below figure shows core-components of CDH including Spark, MapReduce etc.

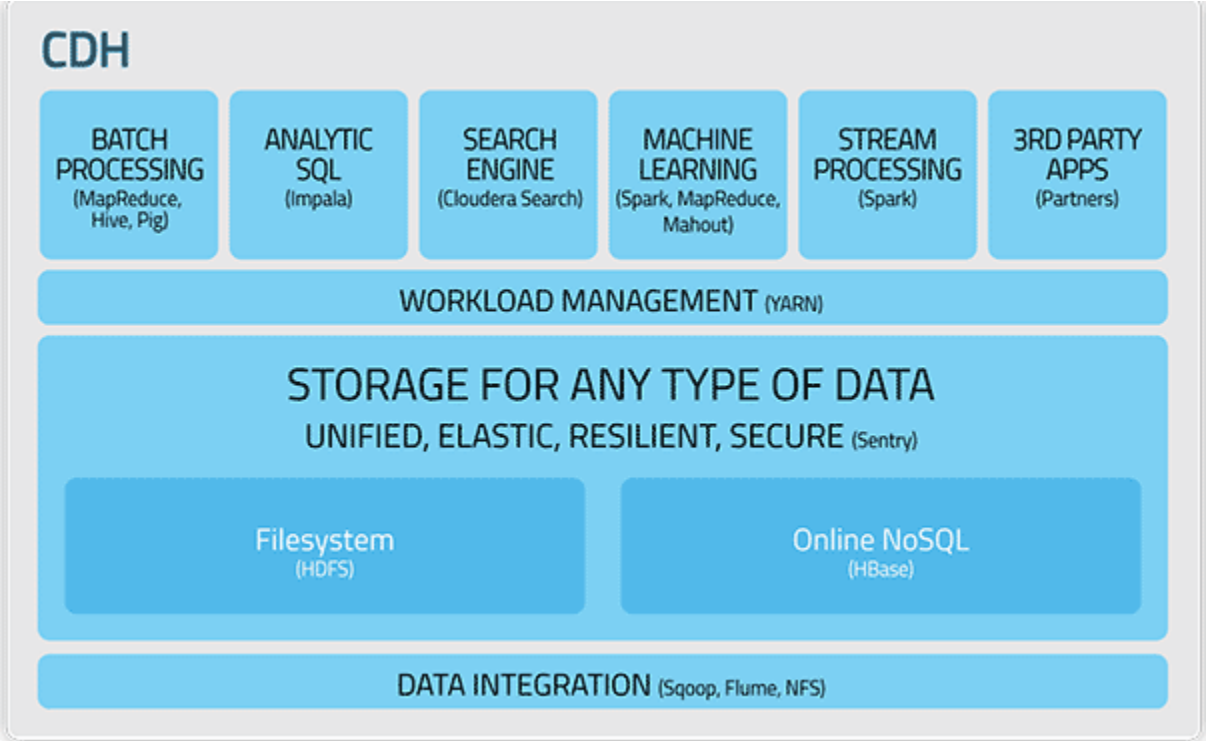


Figure 6.1. (a): CDH and its Components (source: Cloudera.com)

6. MapR Distribution

MapR provides a user-friendly interface and fast performing Hadoop distribution. Its technology runs on both commodity hardware and public cloud computing services.

MapR addresses the limitations of Hadoop with an underlying data platform with no Java dependencies or reliance on the Linux file system. It provides a dynamic read-write data layer that brings unprecedented dependability, ease-of-use, and high speed to Hadoop, NoSQL, database and streaming applications in one converged big data platform.

The MapR Converged Data Platform provides distinct capabilities for management, data protection, and business continuity. It has been developed to converge Hadoop with Spark, web-scale storage, NoSQL into one unified cluster. It enables direct processing of files, tables, and event streams. MapR implements a hybrid approach of using a mixture of data sources for data storage known as "Polyglot Persistence", thus having the ability to leverage multiple data types and formats as per a consumer's use case.

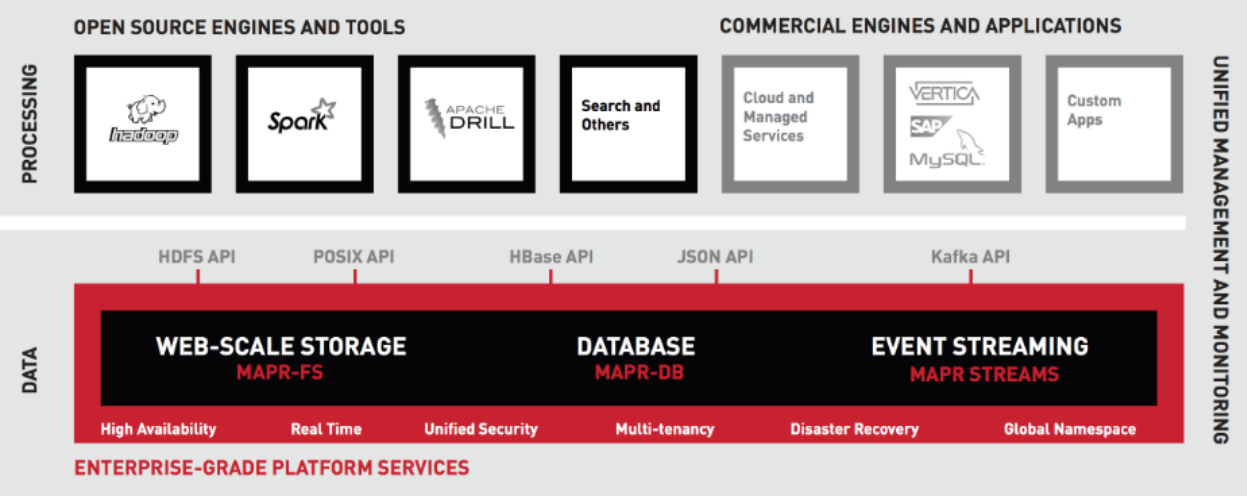


Figure 7 (a): The MapR Converged Data Platform ***(Source:*** *MapR Technologies)*

MapR enables running analytical workloads (dealing with Analytical data) in the same cluster with operational workloads (dealing with historical data), thus avoiding the cost and errors of resource allocation, separate management frameworks and security models.

Web scale storage – File system with full read-write semantics, which can scale to exabytes of data and trillions of files in a single cluster.

NoSQL Database. MapR-DB is a multi-model NoSQL database that natively supports JSON document and wide column data models with high performance, consistent low latency, strong consistency, multi-master replication, granular security, and completely automatic self-tuning.

Event Streaming. MapR Streams is a publish-subscribe, event stream transport engine for reliably delivering ordered messages at high volumes and velocities.

7. Hortonworks Data Platform

Founded in 2011, Hortonworks, provides 100% open source Hadoop distribution platform. It provides Hortonworks Data Platform, powered by Apache Hadoop, which is a massively scalable open source platform for storing, processing and analyzing large volumes of multi-source data. It has YARN placed at the center of its architecture, thus providing a data platform for multi-workload data processing across an array of processing methods. It includes Hadoop Distributed File system [HDFS], MapReduce, Pig, Hive, HBase, Zookeeper and a few other components. HCatalog, a new metadata system, allows Hive and Pig to work together easily by sharing schemas.

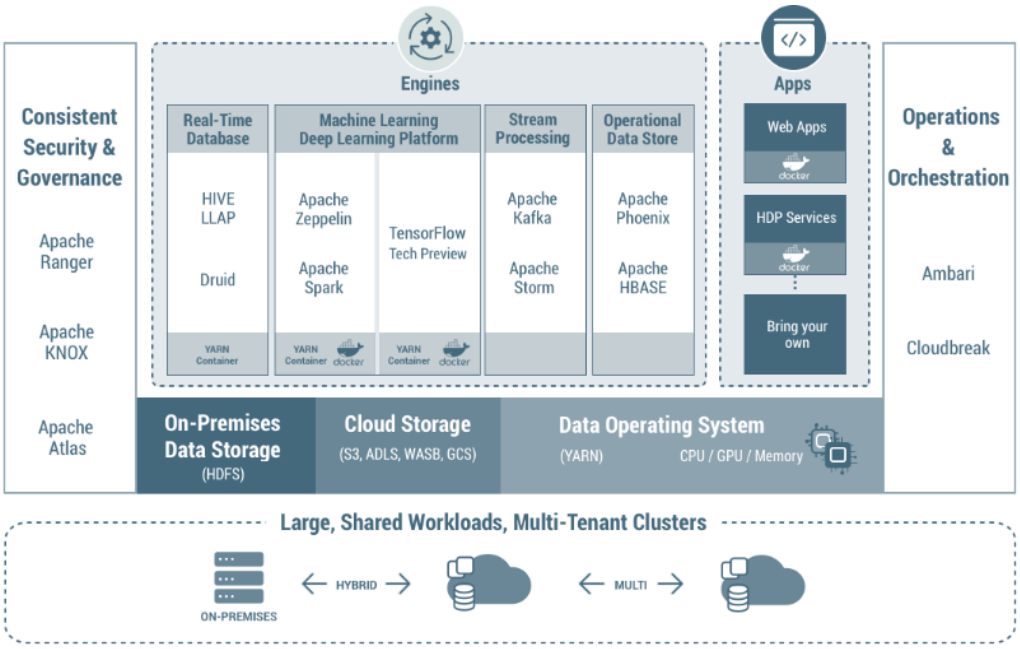
KEY CAPABILITIES:

1. 3rd party applications can run on Apache Hadoop since **Containerization** provides YARN support for Docker containers, thus reducing deployment time.

2. **GPU Pooling and isolation,** enables first-class resource type in Hadoop, thus making running machine-learning and deep learning workloads easier for customers.

3.Offers data protection method known as **Erasure coding.** It eliminates the need to store 3 full copies of each piece of data across clusters and allows more efficient data replication.

4. Disaster recovery is made easy by **NameNode federation.** If one node goes down, cluster can continue to operate since it supports multiple standby NameNodes.



References:

- https://www.wisdomjobs.com/e-university/hadoop-tutorial-484/a-brief-history-of-hadoop-14745.html
- https://opensource.com/life/14/8/intro-apache-hadoop-big-data
- https://www.mssqltips.com/sqlservertip/3140/big-data-basics--part-3--overview-of-hadoop/
- https://www.tutorialscampus.com/tutorials/hadoop/characteristics.htm
- https://www.edureka.co/blog/5-reasons-when-to-use-and-not-to-use-hadoop/
- https://www.guru99.com/introduction-to-mapreduce.html
- https://hortonworks.com/products/data-platforms/hdp/
- https://hortonworks.com/datasheet/hortonworks-data-platform-3-0-datasheet/
- https://www.cloudera.com/documentation/enterprise/5-9-x/topics/cdh_intro.html
- https://intellipaat.com/blog/top-6-hadoop-vendors-providing-big-data-solutions-in-open-data-platform/
- https://www.tutorialspoint.com/apache_pig/apache_pig_architecture.htm
- https://www.ibm.com/analytics/hadoop/hbase
- https://mapr.com/datasheets/mapr-converged-data-platform/
- Ashish Thusoo, J. S. (2010). Hive – A Petabyte Scale Data Warehouse Using. IEEE, 10.
- Apache Hive Architecture & Components BY DATAFLAIR TEAM · PUBLISHED SEPTEMBER 2, 2017 · UPDATED NOVEMBER 17, 2018