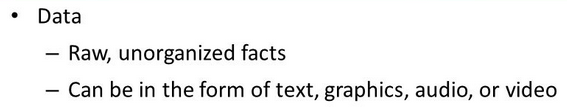
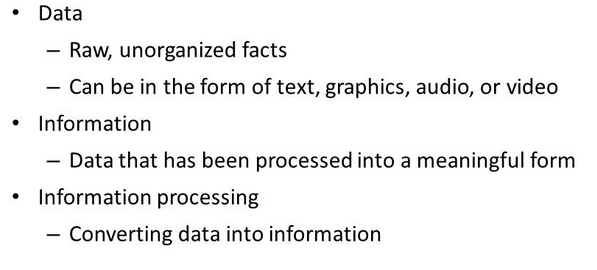
# What is Data?





Data Vs Information



Small Data Vs Big Data –

* Small data is data whose volume and format can be managed, accessible, informative and actionable

Managed – Means can be saved in local PC, Local filesystem, Databases like MySQL, Oracle (Less Space taken than Big Data).

Informative – Easy to extract information because data is mostly in organized format.

Accessible – Since the data can be saved in Local PC, Local File System so we can easy access the data without external tools. But to access the data from databases we need credentials and should have proper permission to access it.

Actionable – Easily transform or analyse the data because of less volume, organized structure.

* Big Data – Refers to the combination of structured and unstructured data that may be measured in petabytes and exabytes. In Big Data world, data is often characterized by 4 V’s .



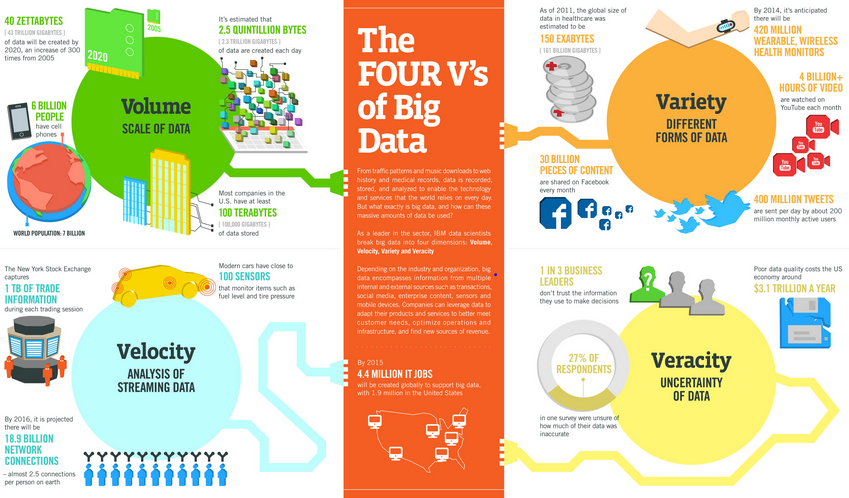
For V’s: Volume, Variety, Velocity, Veracity.

Volume – The size(scale) of data is so huge (like petabytes, zettabytes …)

Variety – Different forms of data (like data generated from disks, from smart watches, from machine, social platform… etc)

Velocity – The speed of data generation. It can be said in streaming analysis of data.

Veracity – Uncertainty in data which means the correct measurement of data quality of analysed.



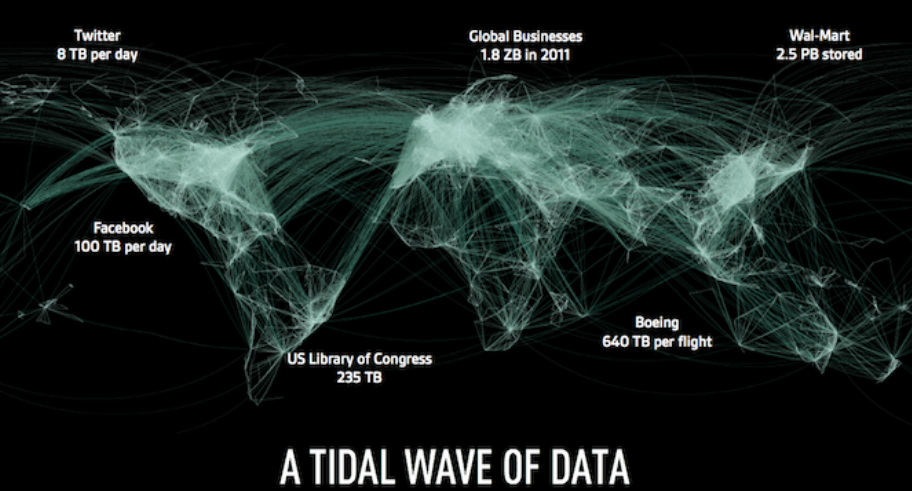
Examples of Big Data in the Real Wold —

• Facebook: has 40 PB of data and captures 100 TB / day

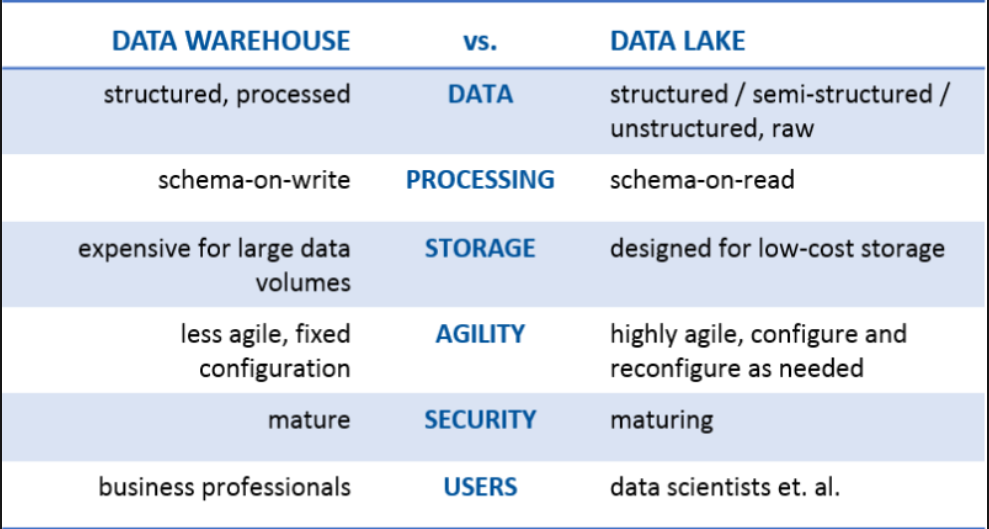
• Yahoo: 60 PB of data

• Twitter: 8 TB / day

• EBay: 40 PB of data, captures 50 TB / day



Data Warehouse Vs Data Lake



Introduction of Hadoop –

Apache Hadoop is an open source framework written in Java. It is designed to process large amount of data on a cluster of commodity hardware.

Large Amount of Data: It has the capability to process huge amount of data.

Commodity Hardware: Low end hardware, they are cheap devices which are economic in nature.

Cluster – A group of low-end devices called as cluster.

Hadoop can be configured on single machine (pseudo distributed mode) but when we configure this on cluster of machines then the real power of Hadoop can be measured (distributed processing).

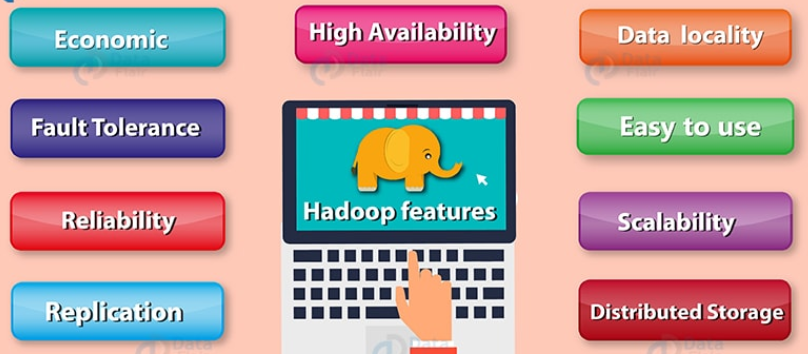
Hadoop consists of three key parts:

Hadoop Distributed File System (HDFS) – It is the storage layer of Hadoop.

Map-Reduce – It is the processing layer of Hadoop.

Yarn – It is the management layer of Hadoop.

Features of Hadoop –



Open Source: It is an open source project which means the source code can be modified according to business requirements.

Distributed Processing: Processing will happen in distributed manner (For Example: Suppose any random user wants to access some information with some specific requirements so Hadoop will help us to process the information from different end points ( in distributed manner).

Fault Tolerance: Since we are working with low-end commodity hardware (which are economic in nature) but we also cannot afford the loss of data so we would replicate the data 3 nodes (configurable parameter) so if any single went down, we can able to process the request through other machines.

Reliability: Since the data is replicated on different machines so data is reliably stored on cluster of machines despite of machine failures.

High Availability: Data is highly available and accessible despite hardware failure due to multiple copies of data (which means accessible from different path)

Scalability: Means nodes can be added anytime (this feature also called horizontal scalability ) without downtime.

Economic: Apache Hadoop is not very expensive as it runs on a cluster of commodity hardware. We do not need any specialized machine for it. Hadoop also provides huge cost saving also as it is very easy to add more nodes on the fly here. So if requirement increases, then you can increase nodes as well without any downtime and without requiring much of pre-planning.

Easy to Use: No need to external tool or client, to perform distributed processing of data. Hadoop can take care of this easily.

Data Locality: Which means the computation moves to the datanode instead data to computation. When a client submits the MapReduce algorithm, this algorithm is moved to data in the cluster rather than bringing data to the location where the algorithm is submitted and then processing it.

## Hadoop Assumptions

Hadoop is written with large clusters of computers in mind and is built around the following hadoop assumptions:

* Hardware may fail, (as commodity hardware can be used)
* Processing will be run in batches. Thus there is an emphasis on high throughput as opposed to low latency.
* Applications that run on HDFS have large data sets. A typical file in HDFS is gigabytes to terabytes in size.
* Applications need a **write-once-read-many** access model.
* Moving Computation is Cheaper than Moving Data.

Traditional Vs Distributed File System:

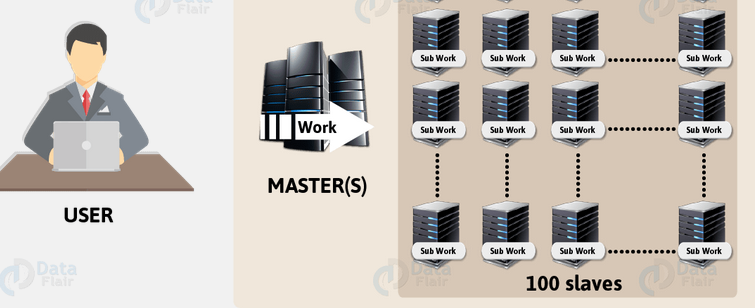
|  |  |
| --- | --- |
| **Traditional File System** | **Distributed File System** |
| Data is maintained in Single system. If machine is down, we cannot able to get the data and failover chances are more | Data is replicated in different nodes. clients able to read the data, if any node is failed. Failover is less. |
| The time taken to read the data in this less as there are only read call to hardisk and the local processing time | The time taken to read the data in this is more as we have network remote call and local data read to disc and coordinating the data from multiple systems |

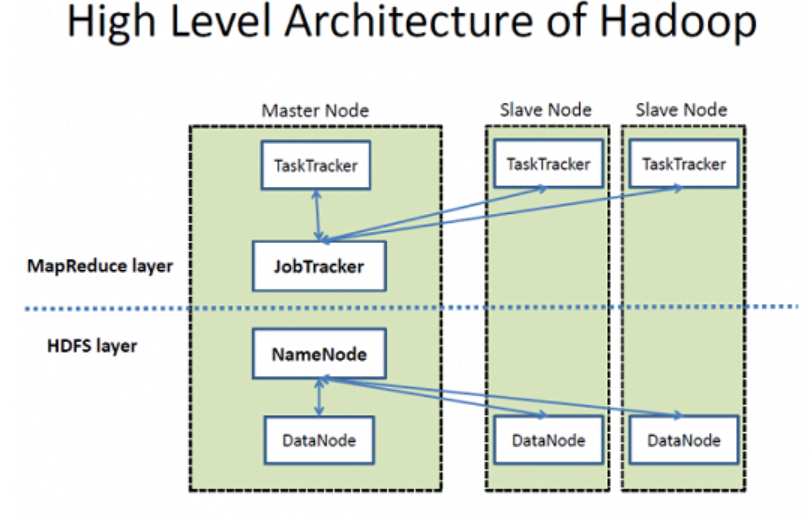
Hadoop Architecture:

* Master-Slave architecture –
  1. Master is responsible for to monitor, manage and maintains the slave nodes in the

architecture .

* 1. Slave nodes(machines) are the machines where actual data is being stored and perform complex computations.





HDFS – is the distributed storage layer for Hadoop.

Map-Reduce – is the processing layer of Hadoop .

NameNode – is the node(or machine) which stores the metadata of your cluster so this machine should be highly configured. Is also the masternode for data storage using HDFS.

JobTracker – is the service in Hadoop that sends out MapReduce tasks on different slave nodes for parallel processing of data.

TaskTracker – is the daemon in the cluster (part of Slave node) that accepts the tasks (map , reduce and shuffle operations) from JobTracker.

Components of Hadoop:

* + Hadoop Common
  + HDFS
  + MapReduce Framework

Hadoop Common-

* A set of utilities that support the Hadoop subprojects
* Provides access to the filesystems supported by Hadoop
* Hadoop Common package contains:

1. The JAR files and scripts necessary to start Hadoop
2. Source code, documentation and a contribution section with projects from the Hadoop Community

HDFS - Hadoop Distributed File System

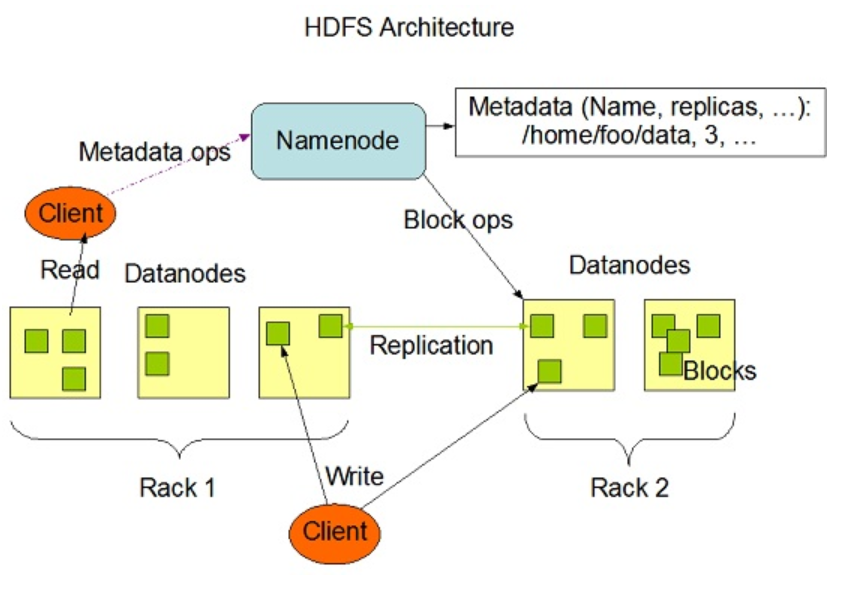
* Primary storage system for Hadoop
* Distributed, portable, scalable file system written in Java
* Files are broken down and stored in multiple machines
* Designed for large scale distributed data processing
* Follows master/slave architecture
* In HDFS data is divided into blocks and is stored on the nodes present in HDFS cluster

HDFS consists of :

* NameNode stores the metadata of the system.
* DataNodes, where the data blocks are stored
* Secondary NameNode takes snapshots of NameNode's memory structure.

Advantages of HDFS:

* Data awareness between JobTracker and TaskTracker
* Highly fault-tolerant
* Designed to deploy on low-cost hardware
* Suitable for applications that have large datasets(Giga to TeraBytes)
* Enables streaming access to file system data.
* Reliable, Distributed, Scalable, Highly Available.



NameNode—

This stores the metadata of the filesystem in memory.

All the files and directories in HDFS are represented in the form of Inodes which contains information like permission, accesstime, modification timestamp etc.

Two file are used for persistence during restarts:

FsImage: this contains the entire filesystem image which means the inodes and the list of blocks at any point of time.

Editlogs – contains the modification happened on the filesystem after the restart of the namenode.

When the NameNode starts, fsimage file is loaded and then the contents of the edits file are applied to recover the latest state of the file system. The only problem with this is that over the time the edits file grows and consumes all the disk space resulting in slowing down the restart process. If the hadoop cluster has not been restarted for months together then there will be a huge downtime as the size of the edits file will be increase. This is when Secondary NameNode comes to the rescue. Secondary NameNode gets the fsimage and edits log from the primary NameNode at regular intervals and loads both the fsimage and edit logs file to the main memory by applying each operation from edits log file to fsimage. Secondary NameNode copies the new fsimage file to the primary NameNode and also will update the modified time of the fsimage file to fstime file to track when then fsimage file has been updated.

On startup every DataNode connects to the NameNode and performs a handshake to verify the namespace ID and the software version of the DataNode. If either of them does not match then the DataNode shuts down automatically. A DataNode verifies the block replicas in its ownership by sending a block report to the NameNode. As soon as the DataNode registers, the first block report is sent. DataNode sends heartbeat to the NameNode every 3 seconds to confirm that the DataNode is operating and the block replicas it hosts are available.

High Availability:

