02 Hadoop Intro

> *Hadoop is an open-source distributed computing framework that enables the processing of large datasets across clusters of computers using simple programming models.*

Hadoop is an opensource project comprising a broad set of tools to store and process big data in a distributed environment across a cluster of commodity servers using a simple programming model.
It is designed to scale out from single node to thousands of nodes, each offering its processing power and storage capacity.
The challenge is not about storing massive data, but it is about the processing speed and developing distributed algorithms.
Hadoop is highly distributed,
horizontally scalable,
fault-tolerant,
high throughput,
flexible, and cost-effective software solution for big data processing.

BigData

History of Hadoop:

For many years, users stored data in a database and processed via SQL queries for analysis.
The DWH was used to store structured historical data and process them using OLAP for knowledge extraction.
Data in DWH (before Hadoop) is pre-processed/sampled before the analysis.

Google faced problems in processing huge data.
Google wanted to download the whole Internet and index to support search queries with MySQL database.

Google indexed 1 million pages in 1998, one billion in 2008, and over a trillion pages every day after 2010.
So, Google implemented Google File System (GFS), Google Map Reduce (GMR), and Big Table in C++ for large-scale index data processing.

Later, **Doug Cutting** and **Mike Cafarella** implemented HDFS and MapReduce in java based on GFS and GMR.

2003 – GFS white paper was released.
2004 – Nutch distributed file system was designed based on GFS.
2004 – GMR white paper was released.
2005 – Nutch MapReduce was designed based on GMR.
2006 – Nutch distributed file system + Nutch MapReduce together renamed as Hadoop (as a

subproject of Nutch) and separated from the Nutch project. 2008 – Hadoop was handed over to Apache software foundation, renamed as Apache Hadoop and opensourced.
2009 – Apache Spark
2011 – Apache Storm
2012 – Yet Another Resource Manager (YARN)

HADOOP FEATURES

Hadoop transforms a cluster of commodity servers into a service that stores and processes PBs of data reliably in a cost-effective way.
Hadoop is capable of handling volume and various problems in big data. It is also offered as a cloud service (HDInsight from Microsoft Azure, Elastic MapReduce (EMR) from Amazon, etc.).

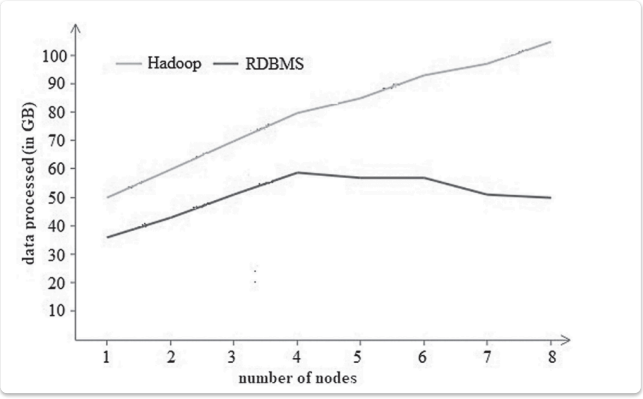
1. **Highly distributed**
   Hadoop works on a distributed file system and distributed computing. So, data and program can be moved around in a compute cluster.
2. **Horizontally scalable**
   Hadoop framework supports horizontal scalability for compute cluster and facilitates to write scalable algorithms using MapReduce. Hadoop supports AP in CAP theorem for a distributed system. Losing consistency in CAP theorem, shared nothing architecture, moving computation to data, no blocking and synchronization among tasks allow Hadoop more horizontally scalable. Theoretically, there is no maximum limit in scaling out. Moreover, the application program need not be rewritten according to scaling.
3. **Fault-tolerance** Cheap nodes fail, especially if you have many in the cluster. Meantime between failures for a server is three years. Meantime between failures for 1000 servers is about one day. So, the failure of servers in a cluster is more likely. Hence, data and computation loss are apparent.

> *Fault-tolerance is the ability of a system to recover from failure automatically and remain functional. Hadoop self-corrects (autonomic computing) when data/process gets lost.*

Hadoop has been designed to detect and handle failures.

4. **High throughput**
   Throughput is the amount of data processed per unit of time. As you increase the number

of nodes in the Hadoop cluster, throughput increases linearly (which is not true in RDBMS).



5. **Flexible software**
   When a Hadoop cluster is up and running, you can dynamically add/remove nodes on the fly without disturbing/shutting down the cluster.
6. **Commodity hardware**
   Hadoop is designed to run on commodity servers. This means that you are not tied to expensive, proprietary hardware from a single vendor. You can use standardized, commonly available hardware from any vendor to build your Hadoop cluster. The commodity does not mean our laptop/desktop machines, which are cheap and has a higher failure rate. Nonetheless, Hadoop can be deployed in our personal computers/laptop too, but running them 24/7 is not possible.
7. **Rack-aware/topology-aware**
   Hadoop is fed the network topology (arrangement of nodes in a cluster) via a configuration file, using which Hadoop can decide where to place data to minimize network flow and improve fault-tolerance
8. **Shared nothing cluster** One server in a cluster cannot access the resources (memory, storage) of another server. Therefore, each server in Hadoop cluster autonomously functions. Finally, Hadoop application developers need not worry about networking, file IO, distributing program and data, parallelization and load balancing, task/ data/node failure (fault-tolerance). Hadoop takes care of these complexities and lets programers concentrate on writing algorithms to process the data. Hadoop completely hides system-level complexities from programmers.