PAGE No.
DATE / /

What is Big Data

- Any Data which can be charateized by 3V's is consider to be a Big Data as per IBM.

3V's of Big Data

- Volume
- Variety
- Velocity

Volume

- Scale of Data.
- 2.5 quintillion (2,500,000,000,000,000,000) bytes of data are created each day.

Variety

- Different forms of data
- Structured data - RDBMS Databases (Oracle & MySQL)
- Semi-Structured data - CSV, XML, JSON
- Unstructed data - Audio, Video, Image, log files

Velocity

- Speed of Data
- 900 Million photos on Facebooks.
- 600 Million tweets on Twitter
- 0.5 Million hours of Video on Youtube.
- 3.5 Billion searches on Googles.

PAGE No.
DATE / /

Why Big Data.

Process
- To process huge amount of data which traditional system are not capable of processing.

Store
- To process huge amount of data we 1st need to store it.
- Are our traditional systems capable to store such massive amount of data?
- Traditional system are NOT fit to store such huge amount of data.

Big Data System Requirements.

Store - Are our traditional system capable to store such massive amount of Data?
- Traditional system are NOT fit to store such huge amount of data

Process - Process huge amount of data in a efficient and timely manner
- Traditional system are NOT capable to handle.

Scale - Scale easily to accomodate growing requirement
- Traditional system have serious Limitation

Store massive amount of Data	Process it in a timely manner	Scale easily as data grows
Store	Process	Scale

PAGE No.
DATE / /

Two ways to build a system

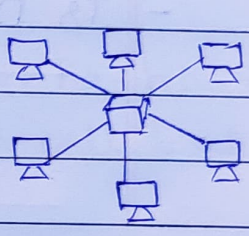
Monolithic	Distributed

Monolithic

One powerful system with lot of resources

Distributed

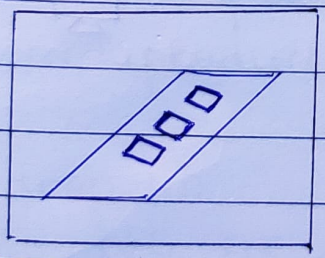
Many smaller system come together



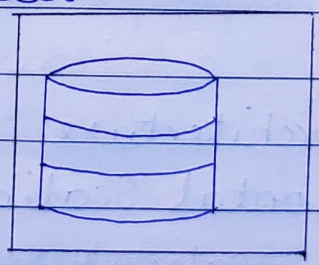
Monolithic

- A single powerful server
- Hard to add resources after a certain limit

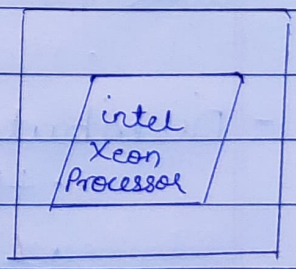
Resources



RAM 8 GB (Memory)



Hard Disk 1 TB (Storage)

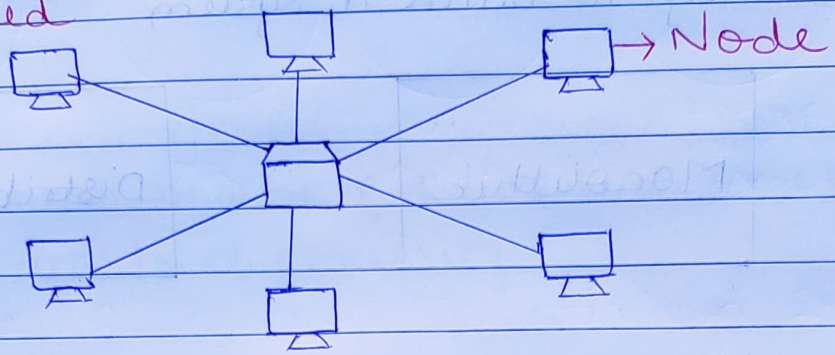


CPU Quad Core (Compute)

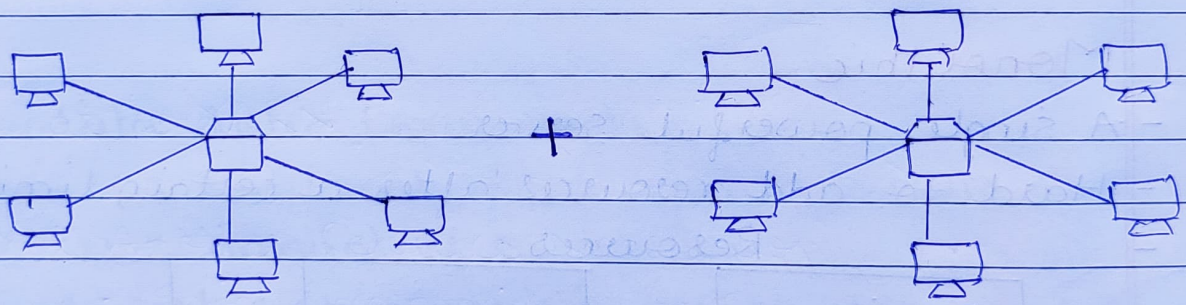
- Is Monolithic scalable?
  - No
    2x resources ≠ 2x performance
- Monolithic Architecture
  - vertical scaling
  - (Not true scaling)

PAGE No.
DATE / /

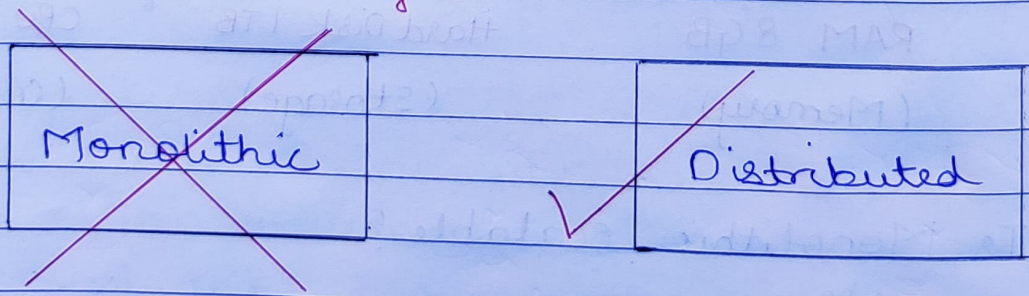
Distributed



- It is 6 Node cluster.
- Many small and cheap computer comes together to acts as a single entity
- Is Distributed system scalable?
  - Yes
  - Distributed system are linearly scable.
  - 2X resource = 2X speed



- Distributed Architecture
  - Horizontal Scaling
  - (True Scaling)



That is why all good bigdata system are based on Distributed architecture.

PAGE No.
DATE / /

What is Hadoop.

- Hadoop is a framework to solve Bigdata problems.

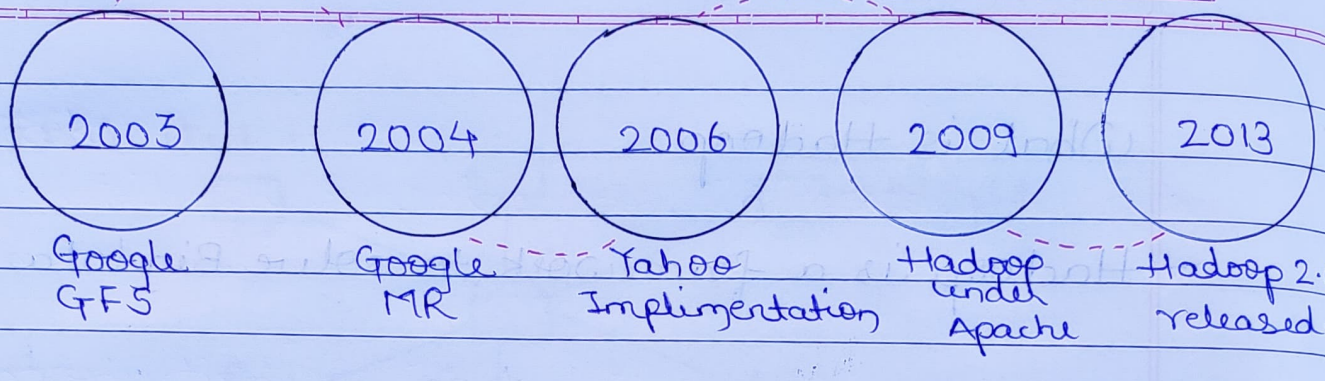
Hadoop Evolution

- In 2003, Google released a paper to describe how to store large datasets. This paper was called as GFS (Google File System).
- In 2004, Google released another paper to describe how to process large datasets. This paper was called as MapReduce.
- In 2006, Yahoo took these papers and implemented it.
  - The implementation of GFS was named as HDFS (Hadoop Distributed File System).
  - The implementation of MapReduce was named as MapReduce (unchanged).

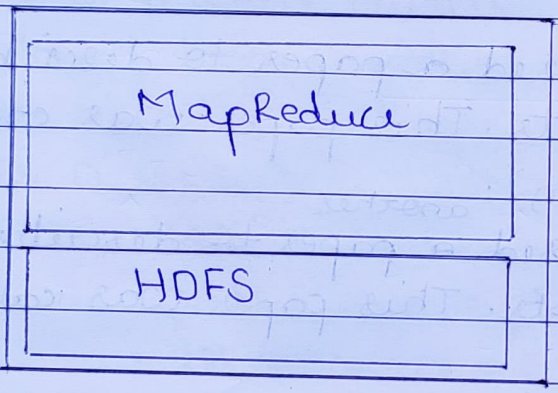
- Hadoop 1.0
  - HDFS for distributed storage.
  - MapReduce for distributed storage.

- In 2009, Hadoop came under Apache Software foundation and become open source.

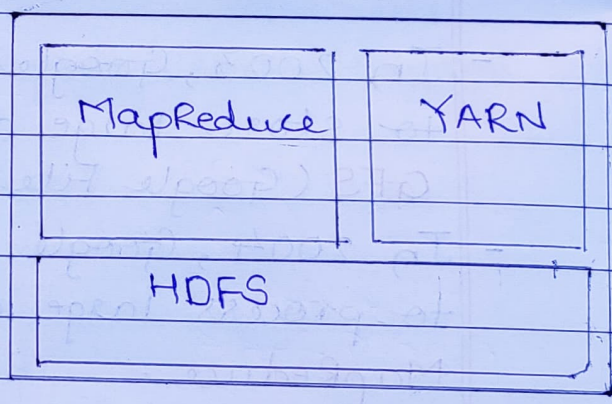
- In 2013, Apache released Hadoop 2.0 to provide major performance enhancements.



Hadoop 1.0



Hadoop 2.0



What is YARN?

Yet Another Resource Negotiator

- Mainly responsible for Resource management.

Hadoop Core Components

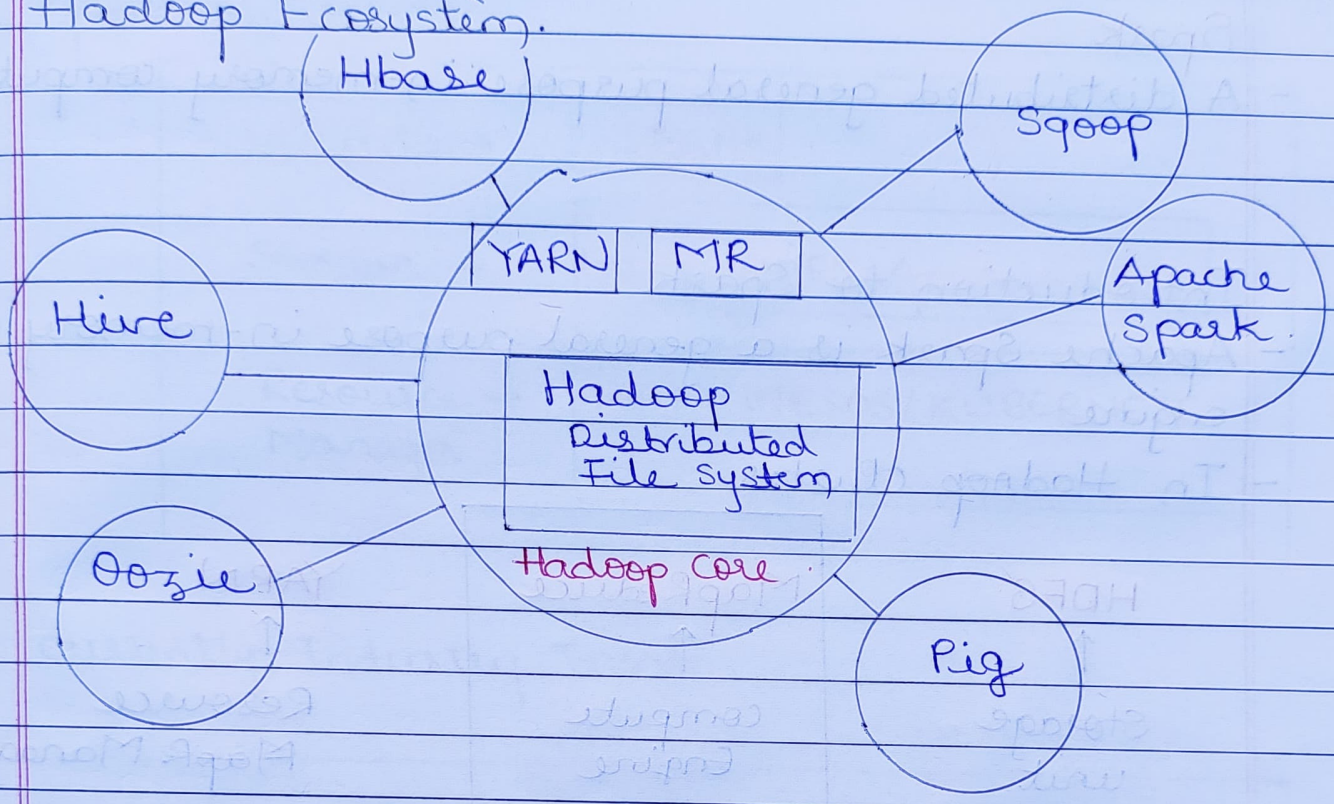
HDFS → for Distributed Storage

MR → for Distributed processing

YARN → for resource management

PAGE No.
DATE / /

Hadoop Ecosystem.



Hive

- Data warehouse tool built on top of Apache Hadoop for providing data query and analysis.

Pig

- A scripting language for data manipulation. Transform unstructure data into structured format.

Sqoop

- A command-line interface application for transfering data between relational databases and Hadoop.

Hbase

- A column-oriented NOSQL database that runs on top of HDFS

Oozie

- A workflow shedular system to manage Apache Hadoop jobs.

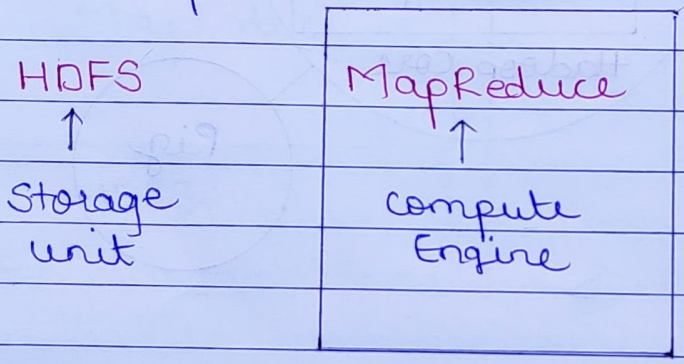
PAGE No.
DATE / /

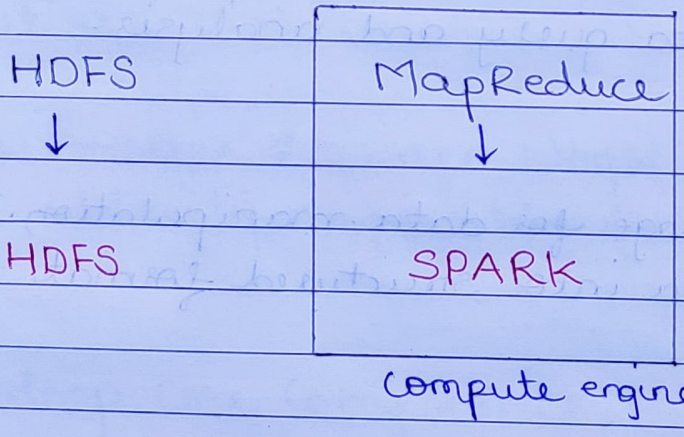
Spark

- A distributed general purpose in-memory compute-engine

Introduction to Spark

- Apache Spark is a general purpose in-memory compute engine.
- In Hadoop cluster

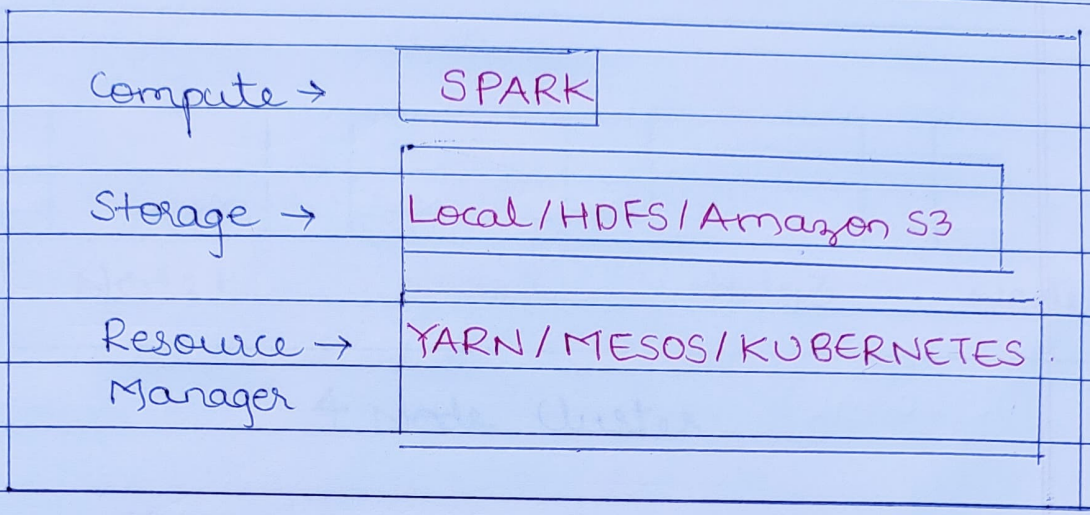




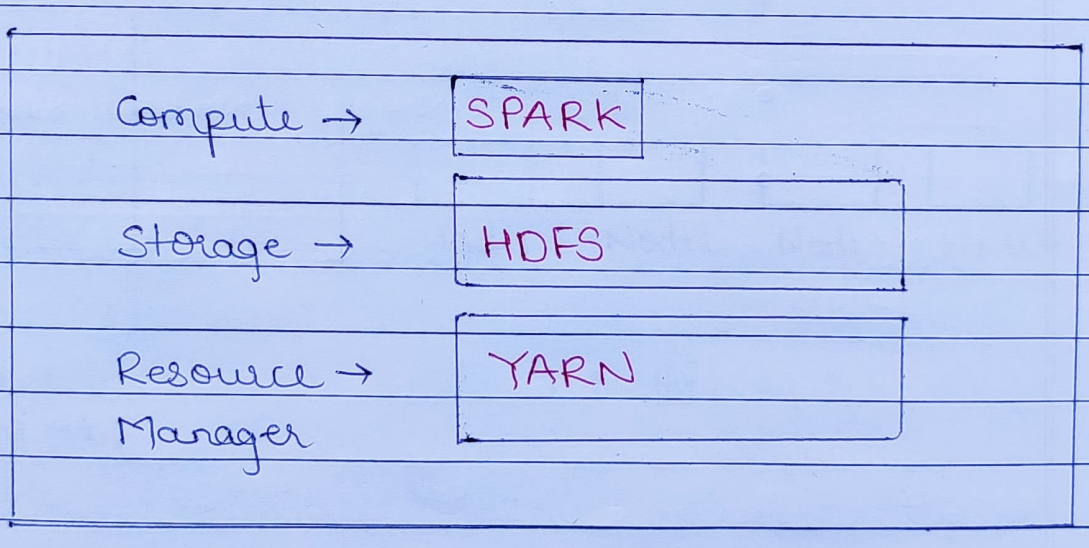
- A plug and play Compute Engine.
- Plug it with any storage system.
  Local Storage / HDFS / Amazon S3
- Plug it with any Resource Manager
  YARN / MESOS / KUBERNETES

PAGE No.
DATE / /

Spark Cluster



currently Industry Trend.



- Spark is written in Scala
- However, spark officially supports Java, Scala, Python and R.