

Week - 1

Understanding Big Data  
The Big Picture

# Big Data

- Volume
- Variety
- Velocity
- Veracity
- Value

Need for a new technology stack ?

Monolithic vs Distributed

One powerful system      Cluster of systems

Compute,  
Memory,  
Storage }      2x resources  
                        f  
                        2x performance

(Commodity  
machines)

Vertical Scaling

Horizontal Scaling  
(True Scaling)

Design a good Big Data System? Consider

- Storage (distributed storage)
- Process (distributed processing)
- Scalability ( $\uparrow$  number of nodes)

Hadoop → first framework designed to solve Big Data problems

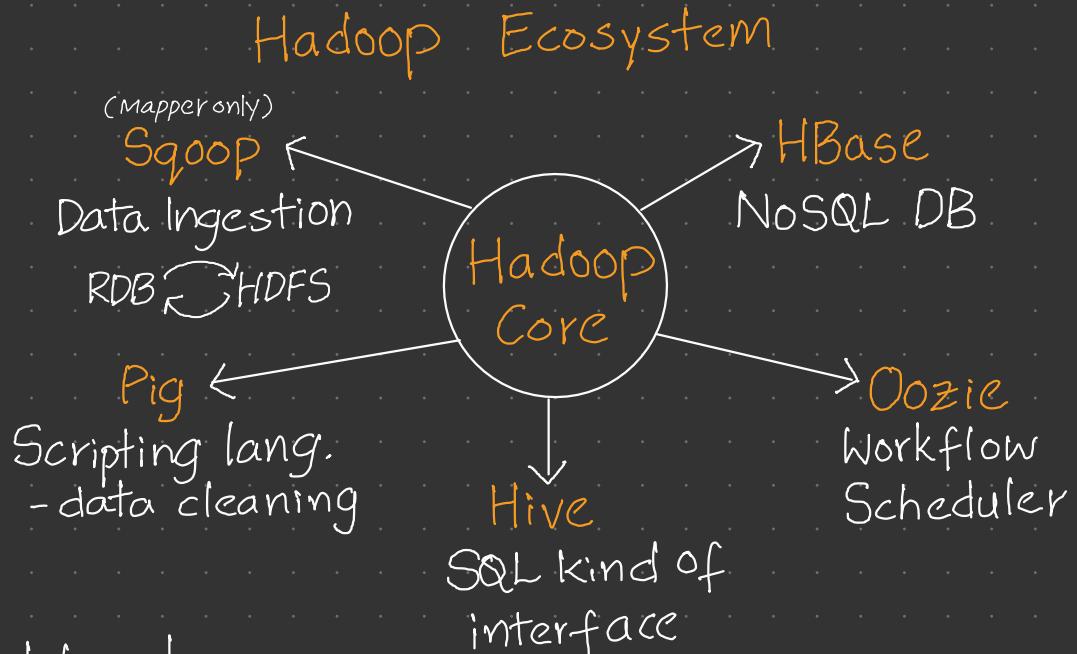
2007      2012      Now  
1.0 → 2.0 → 3.0

## Core Components

- HDFS : Storage
- Map Reduce : Process
- YARN : Resource Manager  
(added in 2.0)

## Challenges

- MR is very slow and hard
- Need to learn different components for different tasks
  - ↓  
under the hood, they convert to MapReduce



## Advantages of Cloud

- Scalable
- Minimum OpEx
- Agility
- Geo Distribution
- Disaster Recovery

## On-Premise Vs Cloud

- |  |   |
|--|---|
| <ul style="list-style-type: none"><li>- Buy the needed Infra, like Space to hold the servers</li><li>- Buy Servers</li><li>- Setup a cooling system</li><li>- Hire a technical team to install and maintain s/w</li><li>- Huge upfront cost / Capex &amp; Opex</li><li>- Not very scalable</li></ul> | <ul style="list-style-type: none"><li>- Infra is taken care by Cloud providers</li><li>- No need to buy servers</li><li>- Setup the cluster with a click of a button</li><li>- Low maintenance cost</li><li>- No upfront cost / Capex</li><li>- Highly Scalable</li></ul> |
|--|---|

## Cloud Types

- Public : AWS, Azure, GCP
- Private: when the data is very confidential
- Hybrid

Spark → general-purpose in-memory compute engine

- Spark can act as a replacement for MapReduce (not for Hadoop)
  - It is a plug and play compute engine
  - Needs storage and resource manager to function.
    - not bound to HDFS, and YARN, only Amazon S3, Mesos, ADLS Gen2, Kubernetes GIC Storage, Local Storage
- Spark is 10X - 100X faster than MR
- Polyglot
- Provides high-level APIs in Java, Scala, Python and R
  - Spark is written in Scala

	Database	Data Warehouse	Data Lake
Workloads	- OLTP	- OLAP	- OLAP
Data type	- Structured or - Semi-Structured	- Structured and/or - Semi-Structured	- Structured, - Semi-Structured and/or unstructured
Schema	- Schema-on-write	- Schema-on-write	- Schema-on-read
Freshness	- Operational - Current data	- Current and - historical data	- Current and - historical data
Prior Processing	- ETL	- ETL	- ELT
Cost	- High	- High	- Low
Examples	- PostgreSQL, - Oracle, MySQL	- Teradata, - AWS Redshift	- Amazon S3, - ADLS Gen 2
Users	- Application developers	- Business Analysts	- Data Scientists

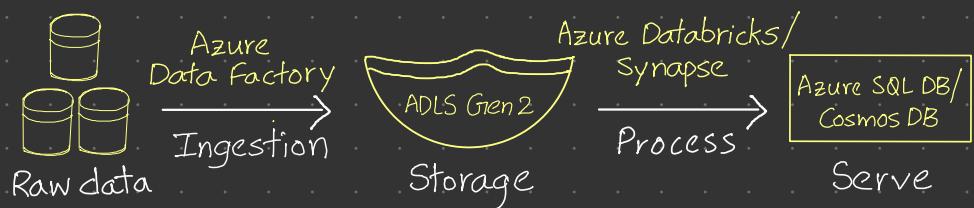
Data Engineering Flow }



Data Pipeline Hadoop }



Data Pipeline Azure }



Data Pipeline AWS }



## Serverless Computing

- Resources are not dedicated  
↓  
No guaranteed performance
- Less expensive
- Good for scheduled jobs  
↓  
that are not particular about the time in which they get finished
- Eg:- Athena, Synapse serverless

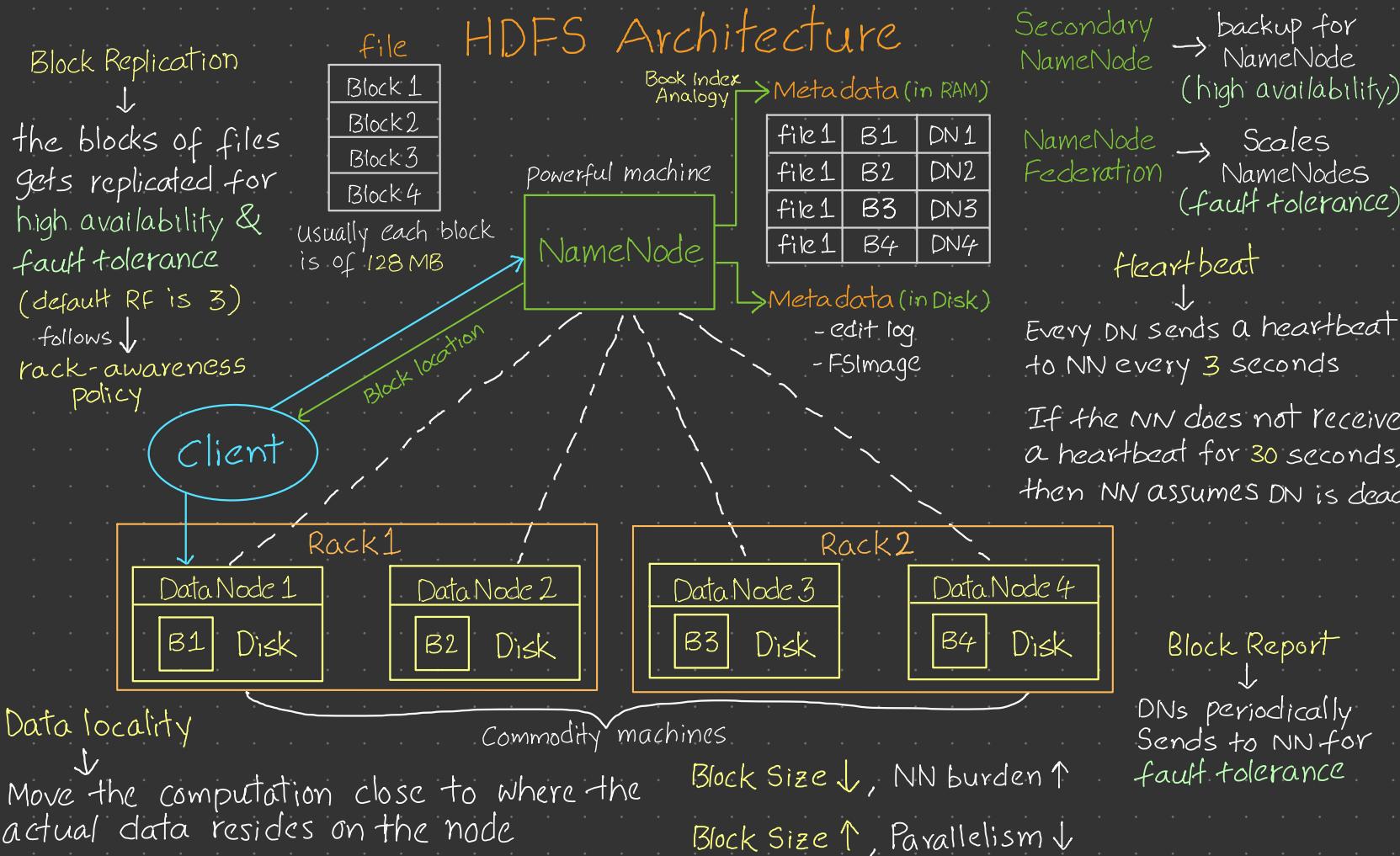
## Serverful Computing

- Resources are dedicated  
↓  
guaranteed performance
- More expensive
- Good for ad-hoc jobs  
↓  
that require immediate execution
- Eg:- Redshift, Synapse dedicated pool

Week - 2

Distributed Storage

# HDFS Architecture



# Rack Awareness Policy

Data Replication → High Availability  
→ Fault Tolerance

Rack awareness policy says :

- Not more than 1 replica be placed on 1 DN
- Not more than 2 replicas are placed on the same rack
- The #racks used in a cluster must be < #replicas

Helps to maximize the Network Bandwidth

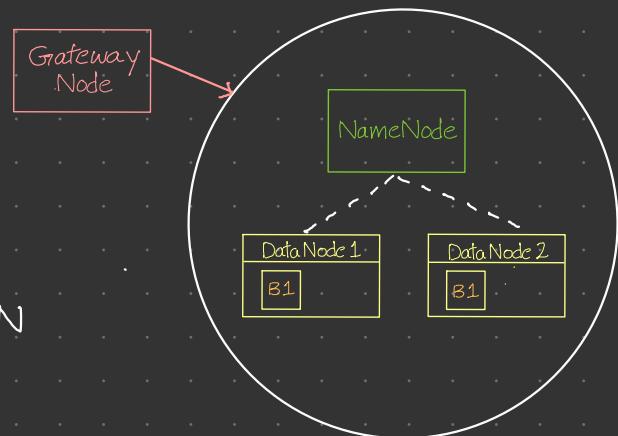


Hadoop mainly works on 3 different modes

Standalone Mode	Pseudo-distributed	Fully-distributed
- Default mode	- Single-node cluster	- Production mode
- Uses local file system	- Uses HDFS	- Uses HDFS
- Used for debugging	- Used for testing	
- No need to change configuration files	- Need to change configuration files	- Need to change configuration files

## Gateway or Edge node

- Connects to the distributed cluster, but does not run any of the daemons.
- Prevent users from a direct connection to critical components such as NN or DN



# Linux Commands

pwd show present working dir

ls list files and directories

whoami show user name

ls-l long listing format

cd to change a particular dir

ls-t sort by last modified

cd or cd ~ navigate to home dir

ls-r reverse order by dict

cd / navigate to root dir

ls-a list hidden files as well

cd . current dir

ls-R recursive list

cd .. move one level up

ls-S sort by file size

cd - navigate to previous dir

ls-h human readable size

Absolute Path Starts from the root /

Touch creates an empty file

Relative Path Starts relative to the current location

Vi edit or create a file

**chmod** change the permissions

		chmod 644 file-name		
r	read	4	4+2	4 4
w	Write	2	RW-	R--
x	Execute	1	Owner	Group Other

**chmod -R** change the permissions  
of a dir recursively

**Cat** view, create, concatenate

Cat file-name view

Cat > new-file-name create

Cat file1 >> file2 concatenate

Cat file1 file2 > merged-file-name

**mkdir** make new dir

**rmdir** removes an empty dir

**rm** removes files

**rm -R** removes a dir recursively

**cp** copy files and dir's

**mv** move/rename files & dir's

**head** show first 10 lines

**tail** show last 10 lines

**grep** global search for regex  
and print out

**du** disk space used

**du -h** human readable

# HDFS Commands - linux based

To execute a linux command on a distributed environment, just prefix it with : hadoop fs - or hdfs dfs -

## HDFS Specific Commands

Copying files/dir's from local to hdfs :

-put  
hadoop fs or <local file path> <hdfs file path>  
-copyFromLocal

Copying files/dir's from hdfs to local :

-get  
hadoop fs or <hdfs file path> <local file path>  
-copyToLocal

File system check - used to check the health of HDFS

hadoop fsck <hdfs file path> -files -blocks -locations

## HDFS

- Distributed file system that stores data as blocks
- HDFS is not persistent  
i.e; data is lost when the Hadoop cluster is terminated  
(newer versions have storage mgmt. features & functions consistent with persisting data)

\* Storage is tightly coupled with Compute (I/O Performance benefits)

i.e; Increasing storage capacity entails increasing Compute & Memory

- The data in HDFS is tied to the cluster it is stored on  
i.e; cannot access it from another cluster

## Cloud Data Lake

- Object-based storage that stores data as objects  
Eg; Amazon S3, ADLS Gen2
- { Unique id, value, metadata }
- Amazon S3 and ADLS Gen2 are persistent

- Storage is decoupled from Compute

i.e; we only need to pay for the storage we use

- Any number of clusters can access the same data