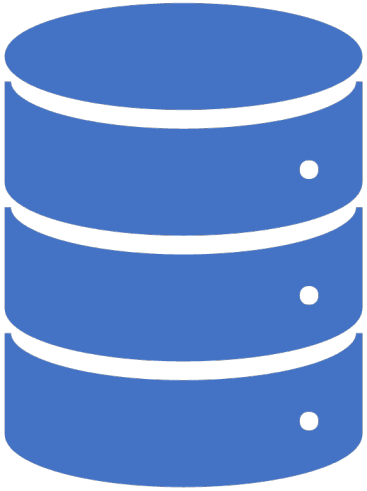
Intro to Bigdata & HDFS

Topics

- What is Big Data?
- Characteristics of Big Data
- Traditional system vs Distributed
- What is Hadoop
- Component of Hadoop.
- Features of Hadoop.
- Hadoop Ecosystem
- HDFS Architecture

TIMESPRO



What will you understand?

- How to identify Big data? - **Characteristics**
- Monolithic way of handling data ? - **Traditional Systems**
- How to store data in efficient way? - **Distributed system (HDFS)**
- How to deal/process this kind of data? - **Hadoop/Hive/spark**

What is Big Data ?

Big Data

- **Big Data** refers to extremely large and complex datasets.
- IBM formal Def - **"Data that is characterized by 3v's/5v's is Big data"**
- These are inadequate to handle by traditional data processing tools and applications. (RDBMS,BI tools, Excel)
- Big Data encompasses
  - a wide variety of Data types,
  - Ranging from structured to semi-structured to unstructured data

Big Data Examples

- **Facebook:**
  - 300 Petabytes of data stored, 600 Terabytes processed daily.
  - 1 billion monthly users, 2.7 billion daily likes.
  - 300 million photos uploaded daily.
- **NSA:**
  - 5 Exabytes of data stored, 30 Petabytes processed.
  - Monitors 1.6% of internet traffic daily.
- **Google:**
  - 15 Exabytes of data stored, 100 Petabytes processed.
  - 60 trillion indexed web pages, 1 billion users served, 2.3 million searches per second.

Note:

1. Terabyte (TB):

- 1 TB = 1,024 gigabytes (GB).
- Commonly used to measure storage on hard drives and servers.

2. Petabyte (PB):

- 1 PB = 1,024 terabytes (TB).
- Used by large organizations for storing massive datasets, like data centers or large databases.

3. Exabyte (EB):

- 1 EB = 1,024 petabytes (PB).
- Used to describe global data storage, such as the total amount of data on the internet.

Characteristics of Big Data

: identify Big data

- **Volume**
- **Velocity**
- **Veracity**
- **Variety**
- **Value**

TIMESPRO

Volume:

EX:Google serach enginer data.

Refers to the amount of data generated every second. With the advent of the Internet, social media, and IoT devices, the amount of data generated is growing exponentially.

Variety:

EX:Google serach enginer data.

Big Data comes in multiple formats: structured (databases), semi-structured (XML, JSON), and unstructured (videos, images, text).

Velocity:

EX:Google serach enginer data.

The speed at which data is generated and processed. With real-time data streams from social media, sensors, and financial markets, the need for rapid data processing and analysis is critical.

Characteristics of Big Data

TIMESPRO

Veracity:

The accuracy and trustworthiness of the data. With the large amounts of data collected, ensuring data quality and accuracy becomes a significant challenge.

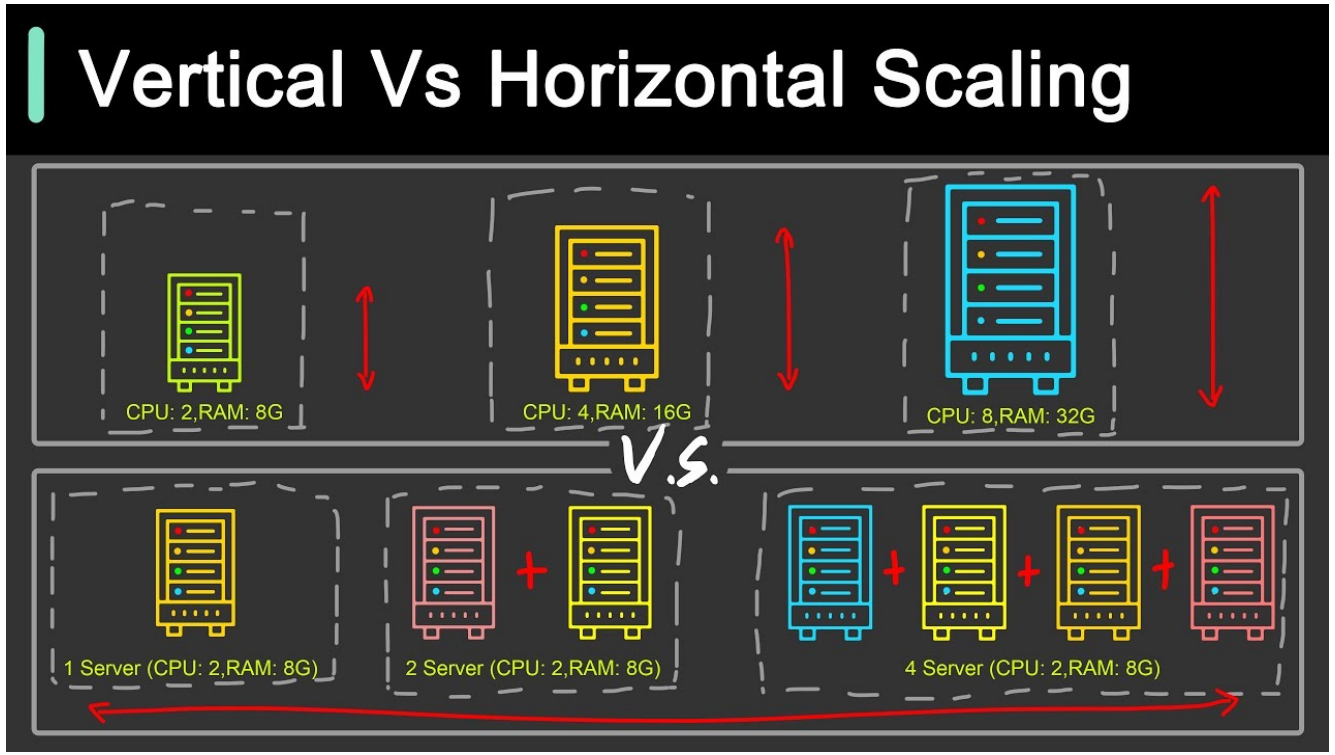
Value:

The potential insights and business value that can be derived from analyzing Big Data. The ability to extract meaningful information and drive decision-making processes is what makes Big Data valuable.

Traditional (Monolythic) Systems

- Mono - One
- One big system holding all power and data.
- X resources ---gives---> y performance
- It leads:
  - 2x increase resource --> lead Costly
  - 2x resource --> not 2x performance but Less
  - Hence, Not truly scalable system and vertical scaling

Horizontal Vs Vertical Scaling



TIMESPRO

Shortcomings of Traditional Systems

Lack of Scalability

Limited Flexibility

Maintenance Complexity

Performance Bottlenecks

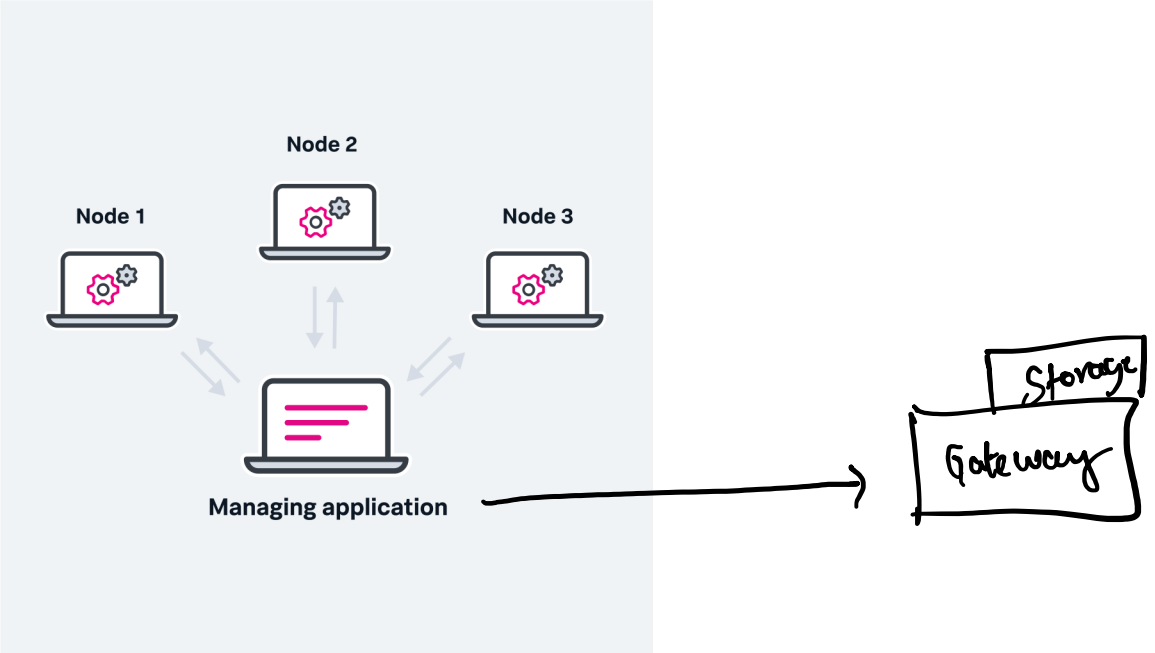
Distributed System

TIMESPRO

- Cluster of system(nodes) grouped with each holding own resources.
- Here computing power is sum of all nodes compute in cluster.
- Follows horizonal scaling. 2x resource—gives--> 2x performance
  - Its true scalability
- Low Maintenance, flexible, more performance,
- This gives distributed storage, computation, language and scalability

TIMESPRO

Distributed System



things to Considerations for Designing a Big Data System

1. **Storage:** Use distributed storage (e.g., HDFS, Amazon S3) for handling massive data volumes across multiple nodes.
   - **Scalability:** Ensure the system scales horizontally by adding nodes as data and processing demands increase.
2. **Processing:** Employ distributed processing frameworks (e.g., Apache Spark, Hadoop MapReduce) for parallel computation across the cluster.
3. **Security :** Data Encryption, Access Control

Quiz Time

The Story of Distributed Systems and Hadoop

- **Introduction to Distributed Systems**
  - **Challenge**: Scaling web search with traditional software was inadequate.
  - **Google's Response**: Development of proprietary software to manage data and processes across hundreds of thousands of machines.
- **Google's Proprietary Software**
  - **Google File System (GFS)**:
    - **Purpose**: Distributed data storage across multiple machines.
    - **Innovation**: Files split into chunks and stored across clusters for redundancy and scalability.
  - **MapReduce**:
    - **Purpose**: Parallel data processing across distributed systems.
    - **Innovation**: Simplified the process of writing distributed applications by abstracting the complexities of data distribution, fault tolerance, and load balancing.

The Story of Distributed Systems and Hadoop

- From Google to the Open-Source World
  - **Knowledge Sharing**: Google published papers detailing GFS and MapReduce.
  - **Nutch Project**: An open-source search engine project that adopted Google's methods.
  - **Result**: Development of Hadoop, modeled on GFS and MapReduce principles.

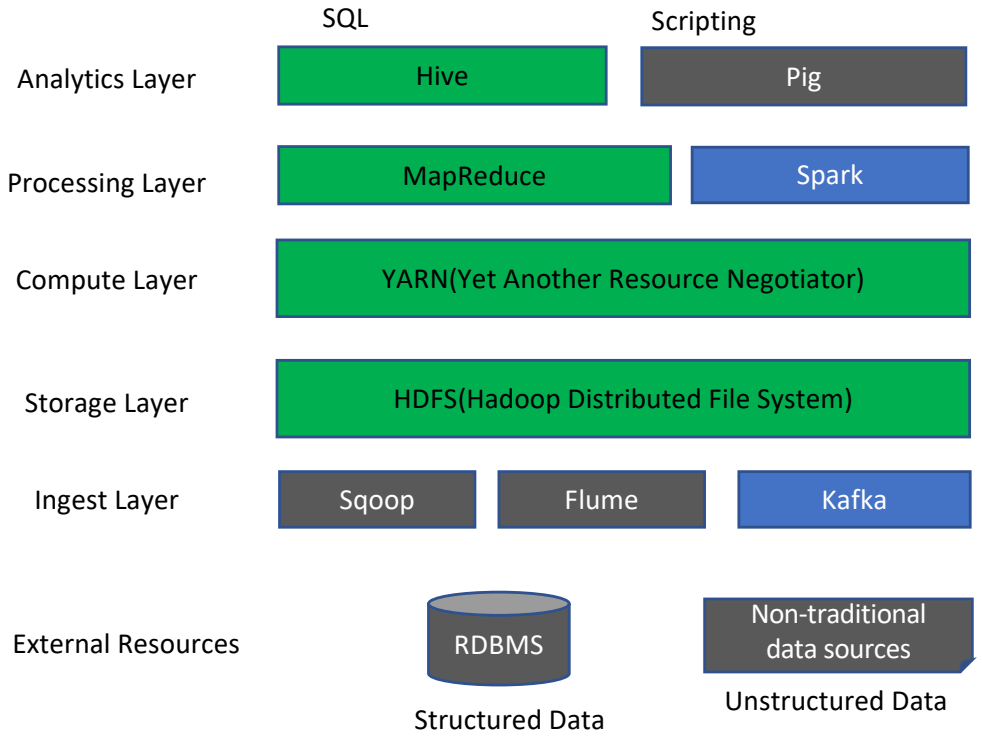
What is Hadoop?

- Hadoop was the first framework designed to solve Big Data problems (processing big data + having many tools to deal).
- It is a framework because, It is not just a single tool but a combination / ecosystem of several tools(HDFS, mapreduce, YARN) and technologies to solve Big Data problems.
- Developed as open source by Apache Software Foundation.
- Enables processing of large data sets across distributed computing environments.

Features of Hadoop

- Handles huge volume of data
- Highly Scalable
- Flexible
- Data Locality
- Highly Reliable
- Integrated
- Cost Effective
- Fault Tolerant

Hadoop Ecosystem Database Layer



TIMESPRO

Managed Hadoop Services on Cloud

- AWS – EMR
- Azure – HDInsight
- GCP - Dataproc

MapReduce

Alternative Hive, Spark

HDFS Architecture

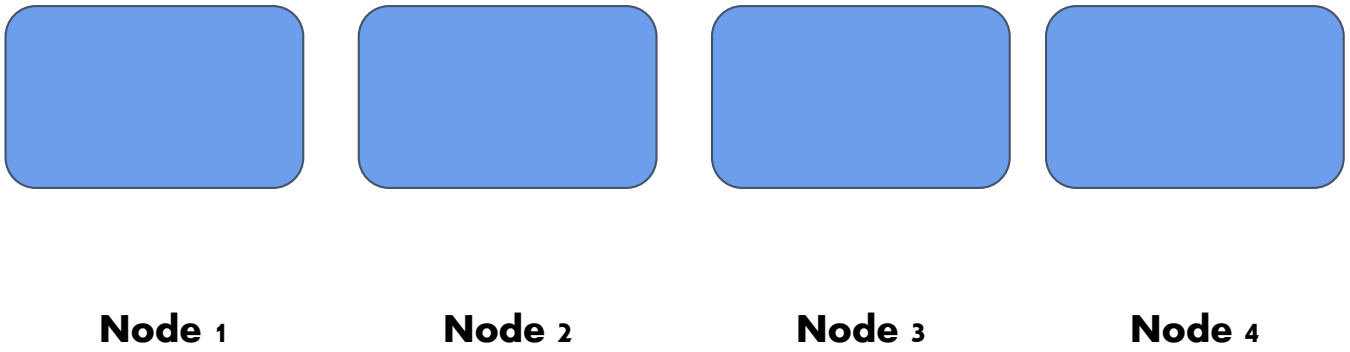
HDFS

Hadoop Distributed File System

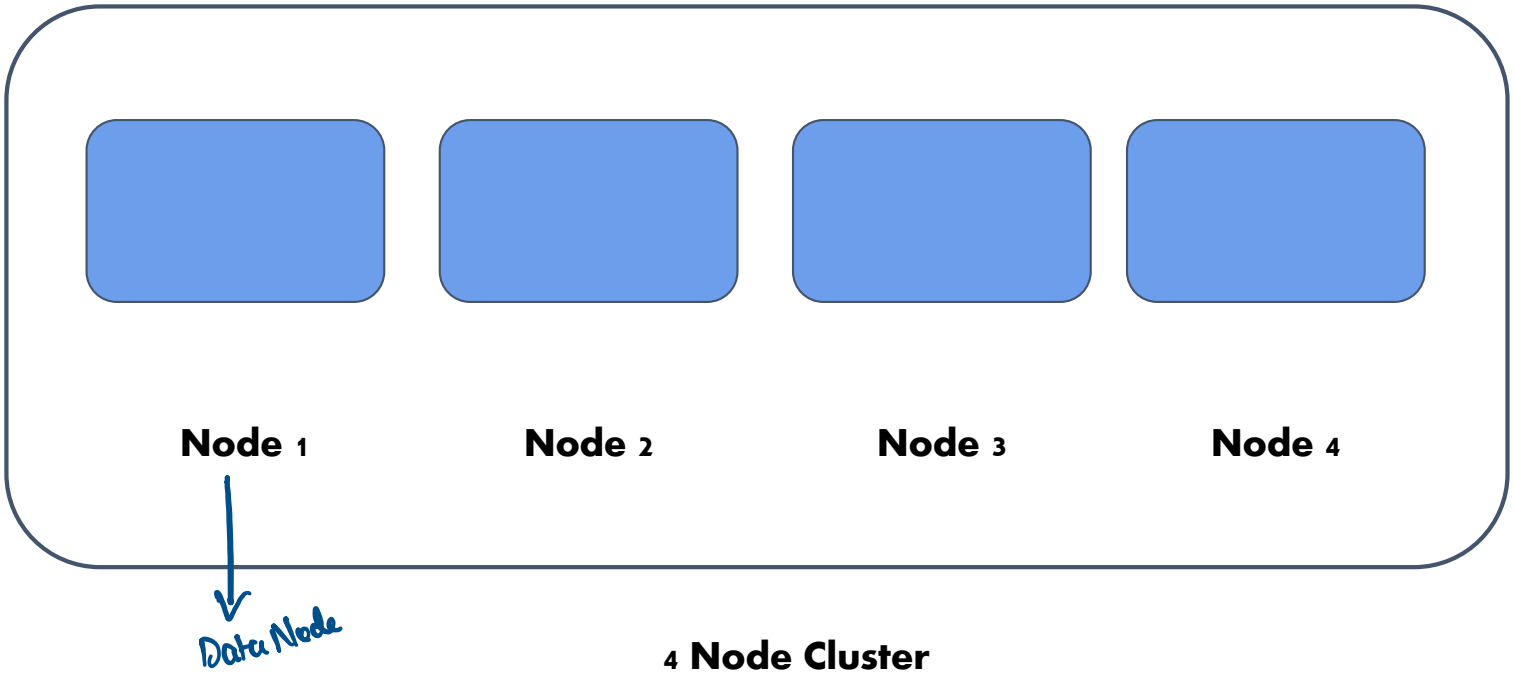
Hadoop Distributed File System

- HDFS - Hadoop distributed file system
- Distributes the data blocks into various nodes
- Replicates the blocks for high availability
- Follows master slave architecture.
- Components:
  - Name node
  - Data node

HDFS Architecture

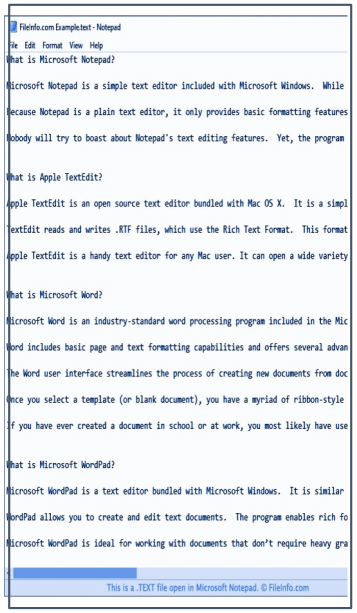


HDFS Architecture

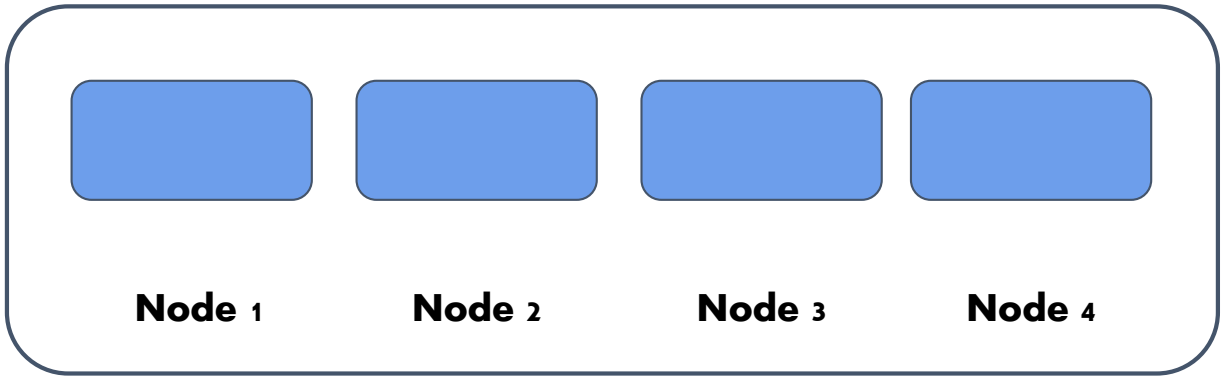


HDFS Architecture

Client Node

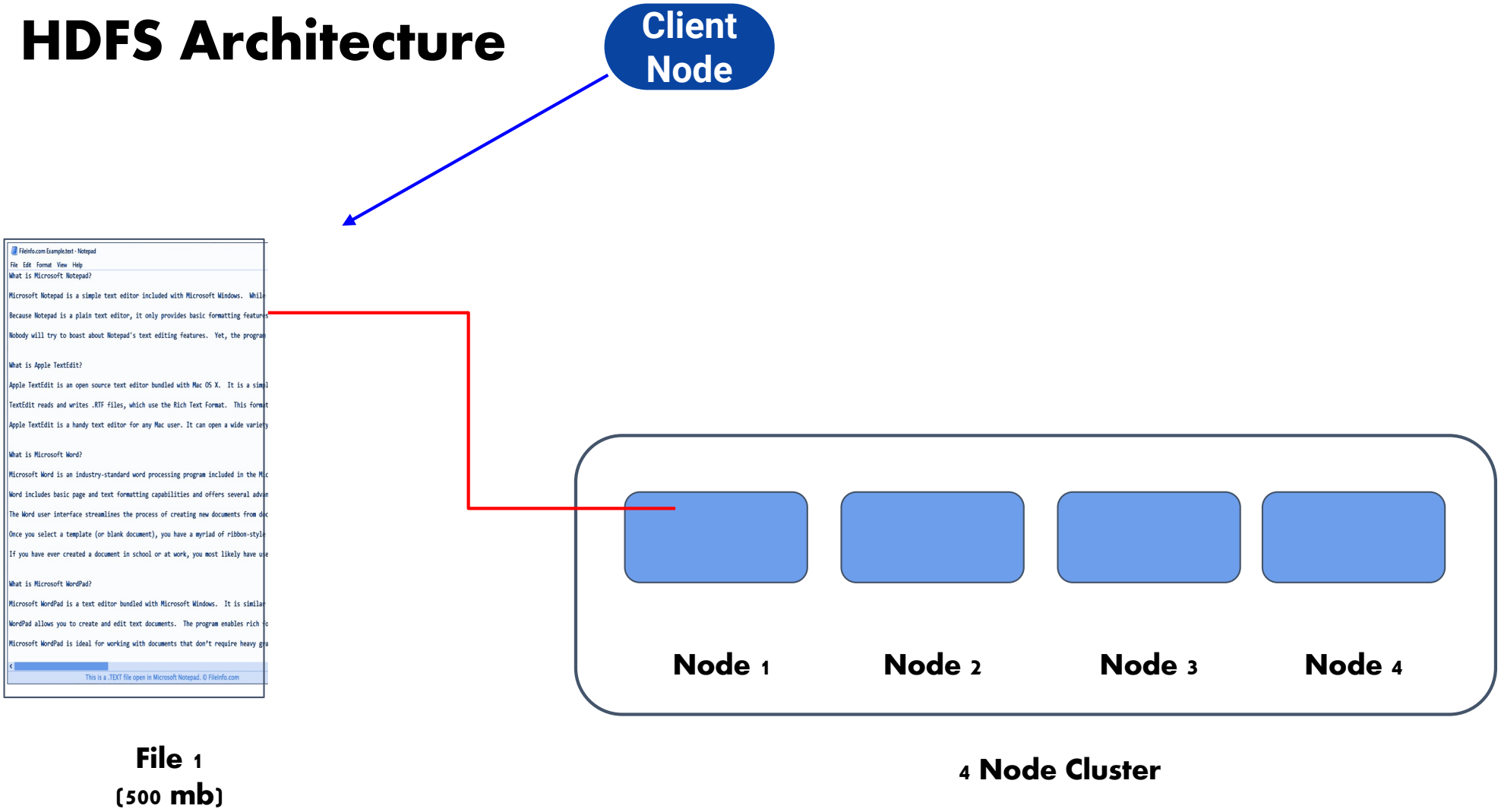


File 1
(500 mb)



4 Node Cluster

HDFS Architecture

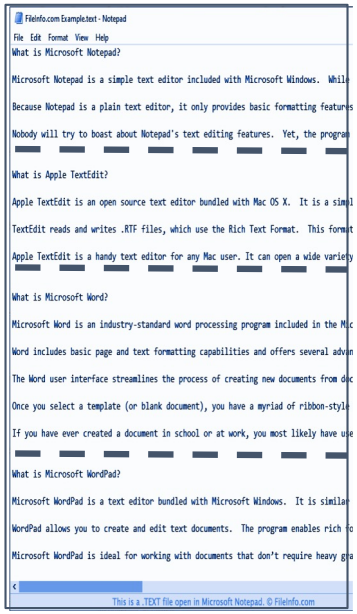


File 1
(500 mb)

4 Node Cluster

HDFS Architecture

Client Node



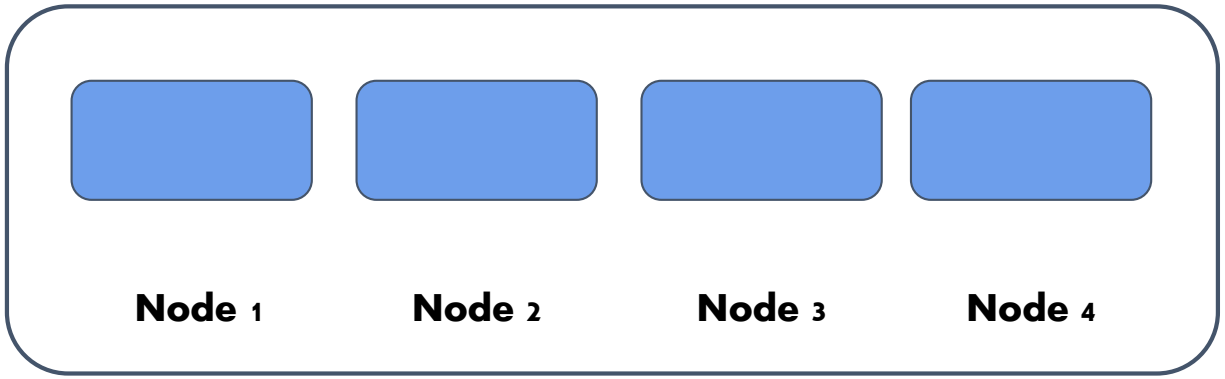
Block 1

Block 2

Block 3

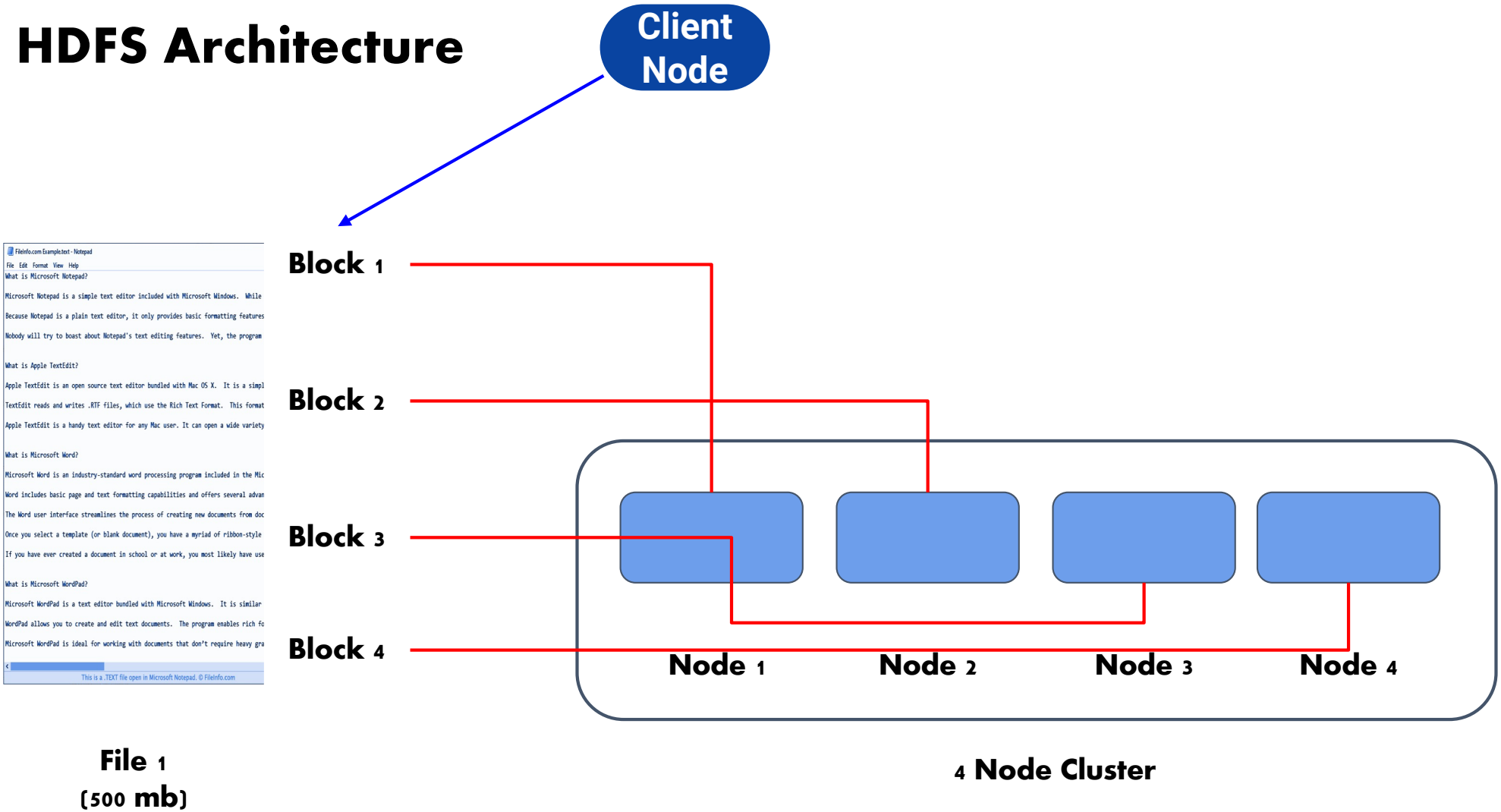
Block 4

File 1
(500 mb)



4 Node Cluster

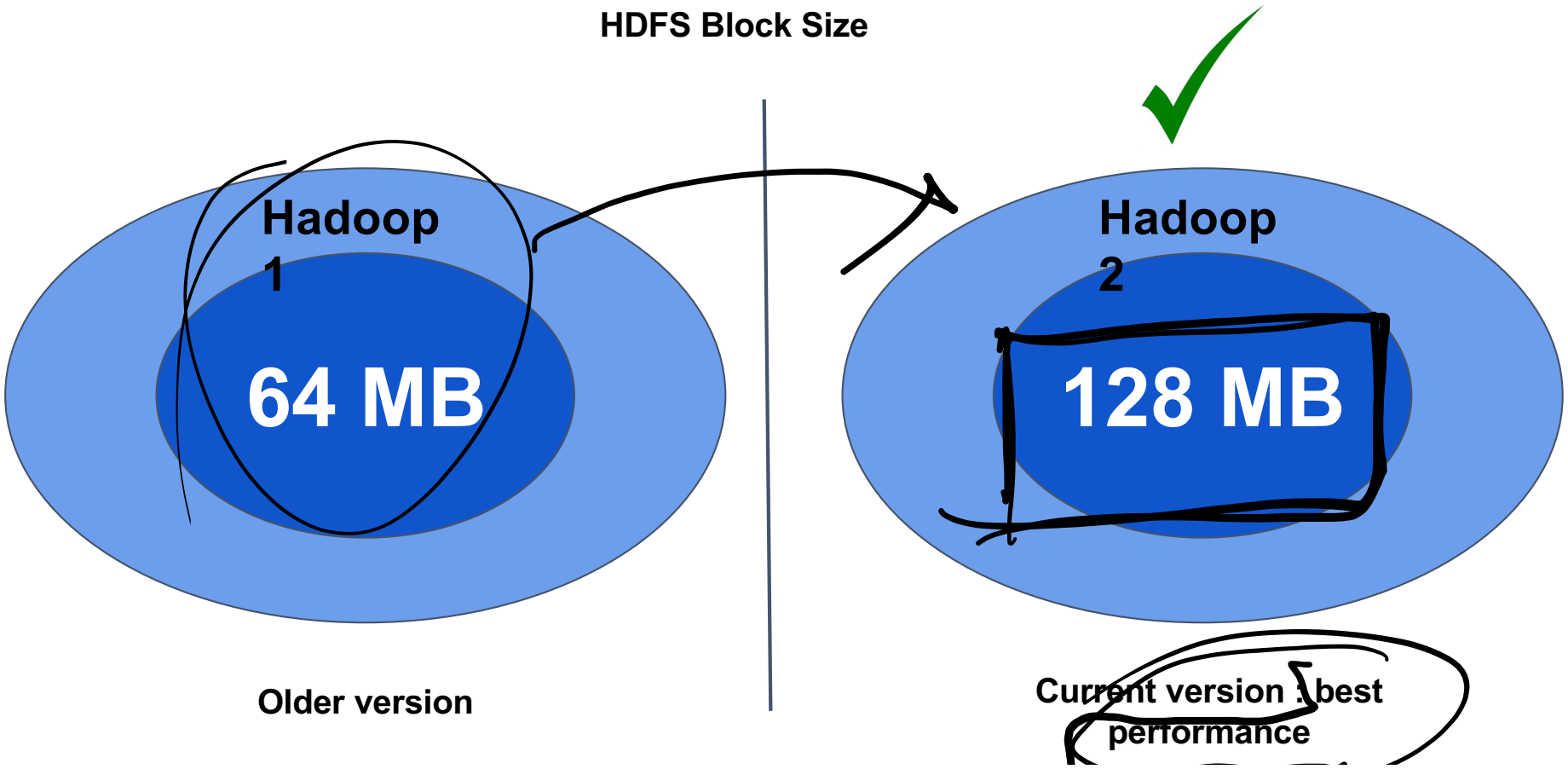
HDFS Architecture



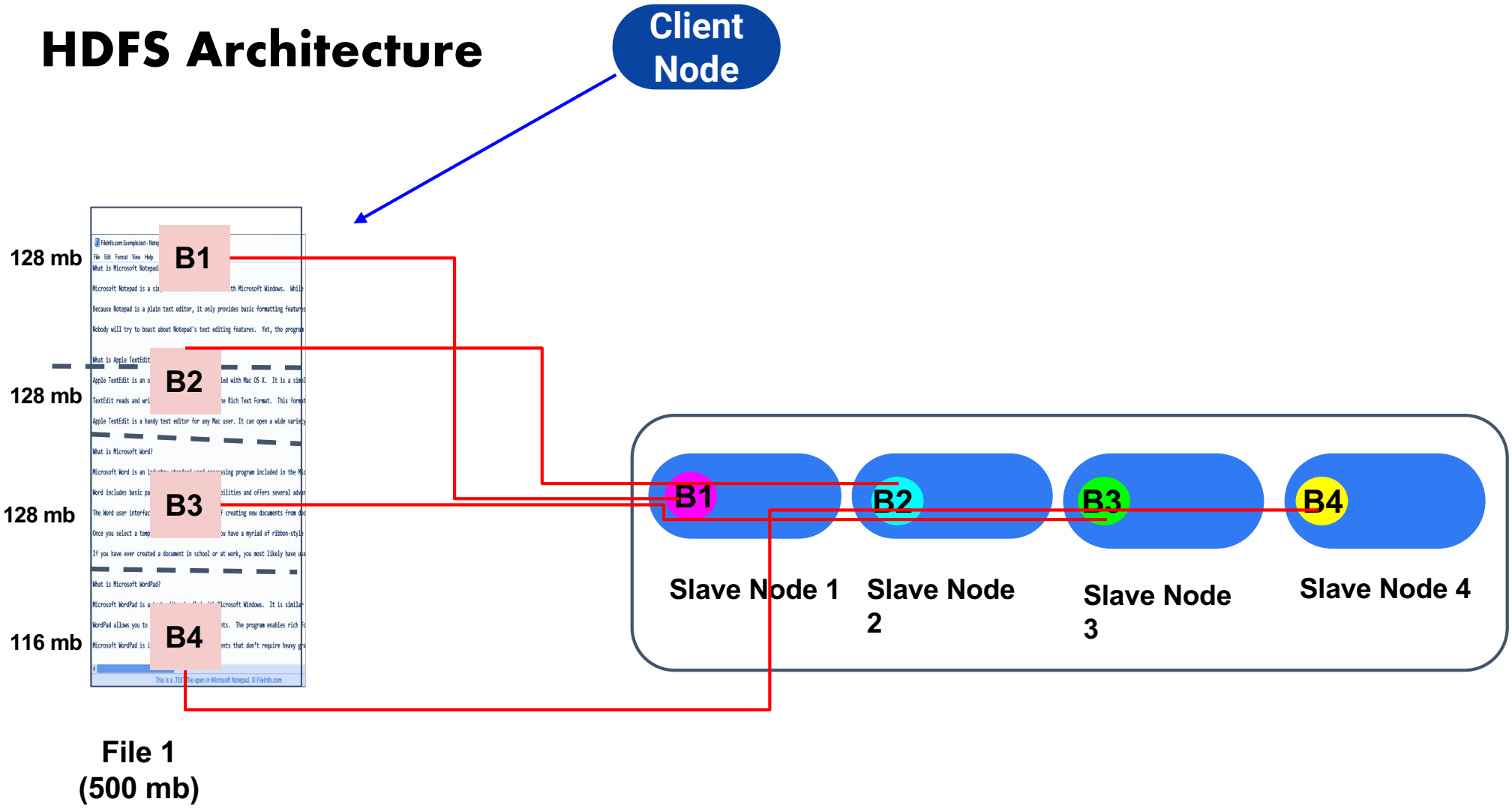
File 1
(500 mb)

4 Node Cluster

HDFS Architecture



HDFS Architecture



HDFS Architecture

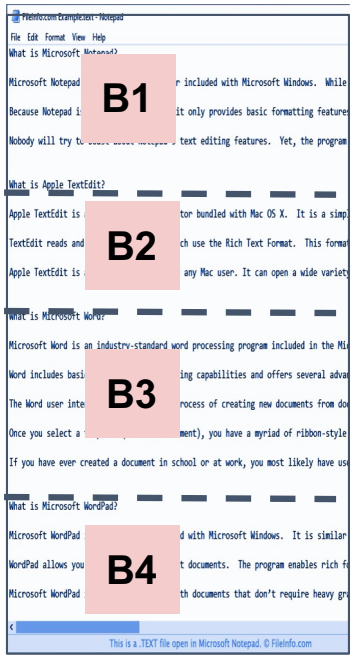
Client Node

128 mb

128 mb

128 mb

116 mb



File 1
(500 mb)

Master Node

(Name Node)

B1 B2 B3 B4

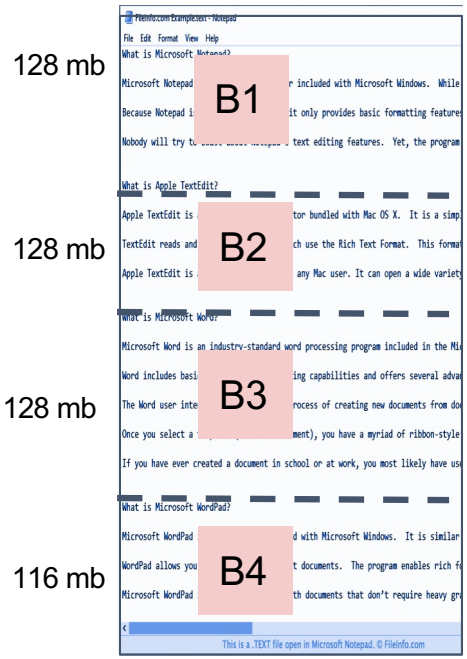
Slave Node 1
(Data Node)

Slave Node 2
(Data Node)

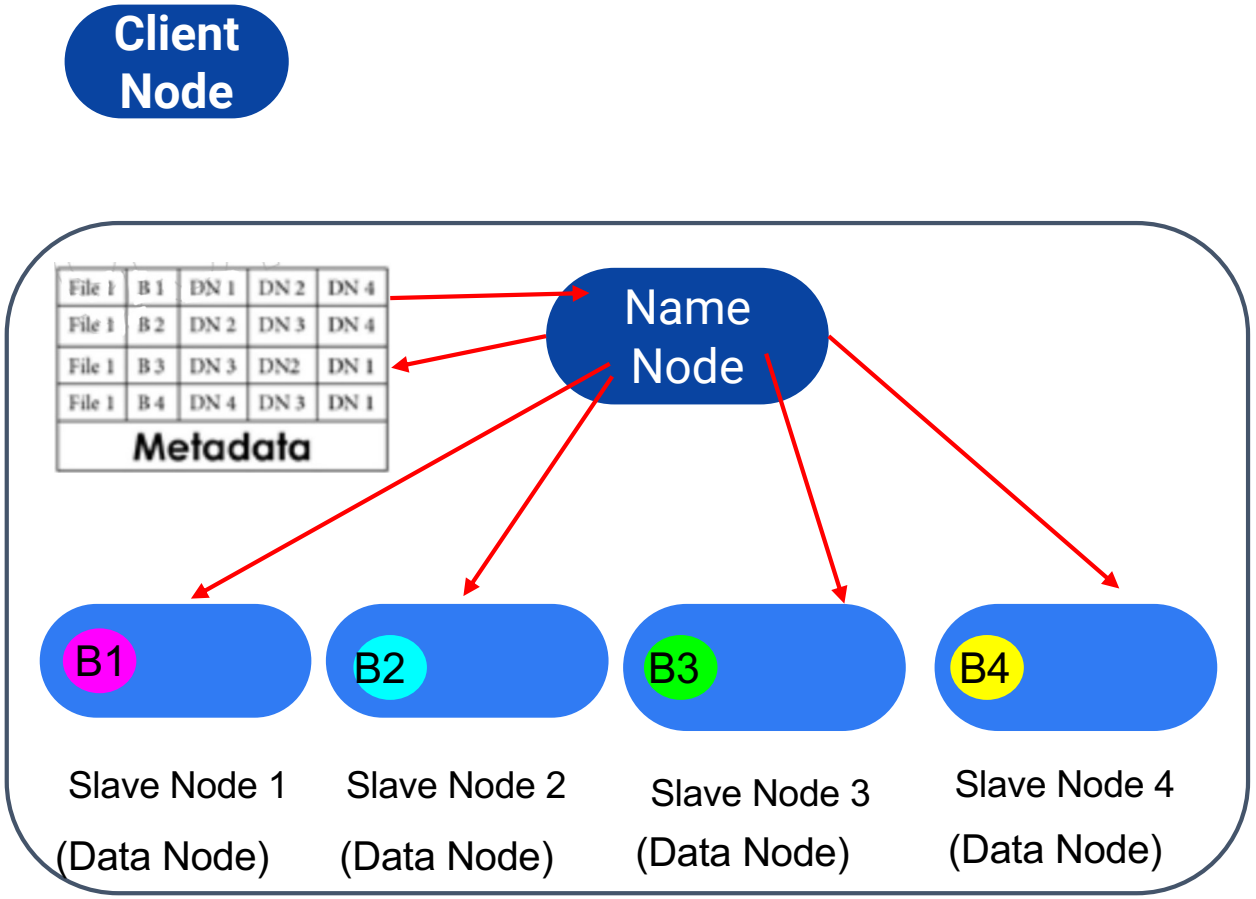
Slave Node 3
(Data Node)

Slave Node 4
(Data Node)

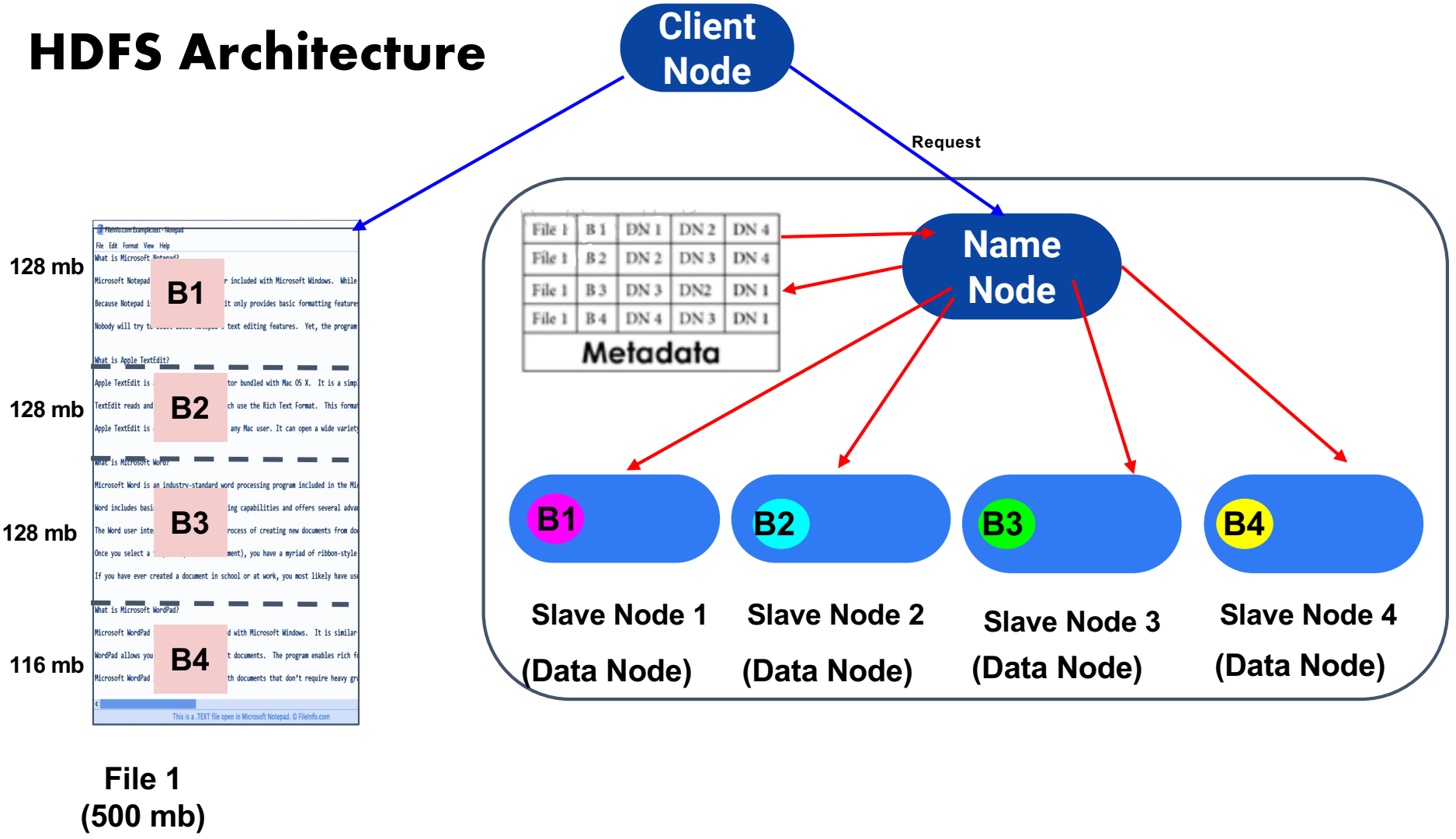
HDFS Architecture



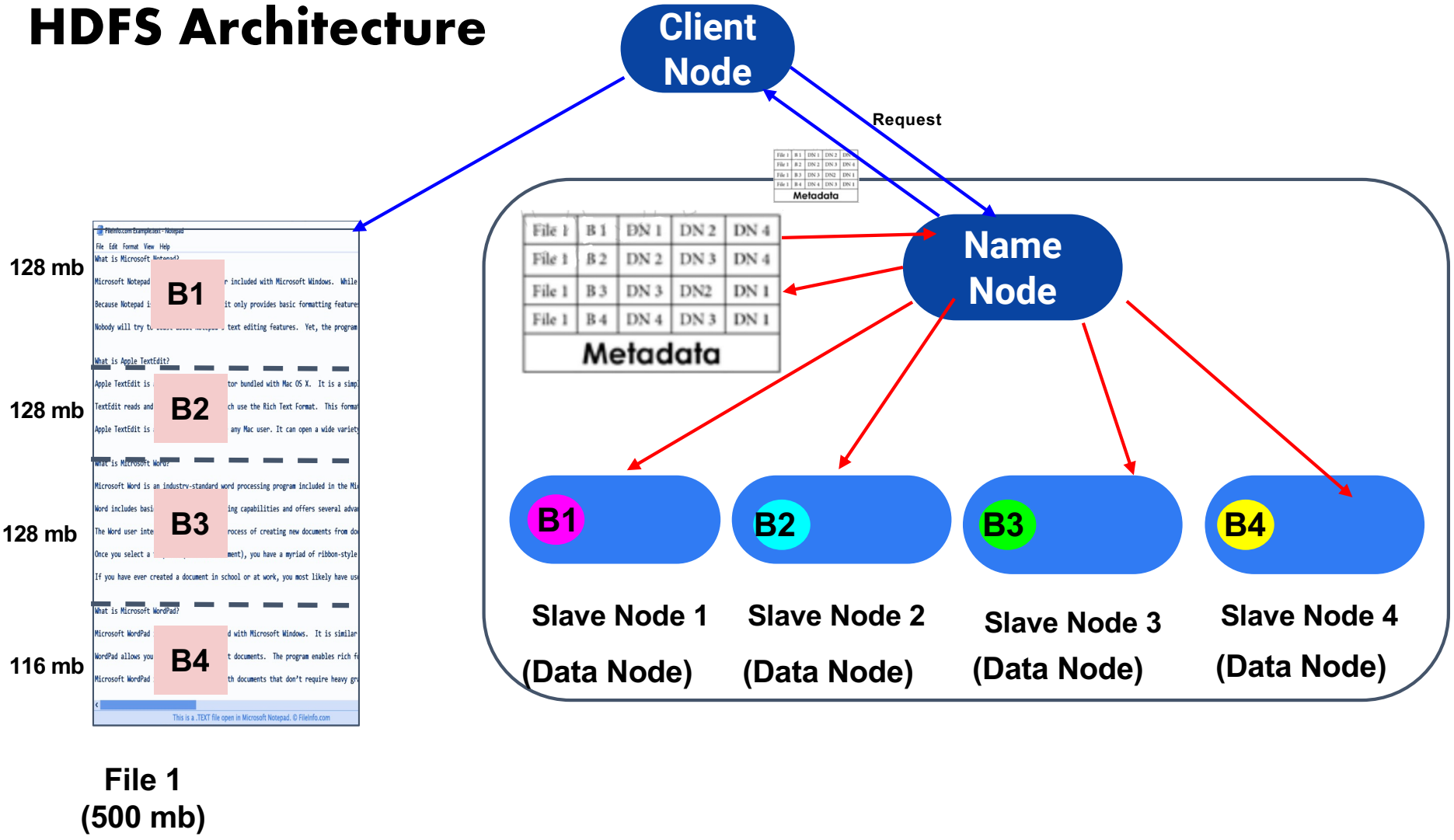
File 1
(500 mb)



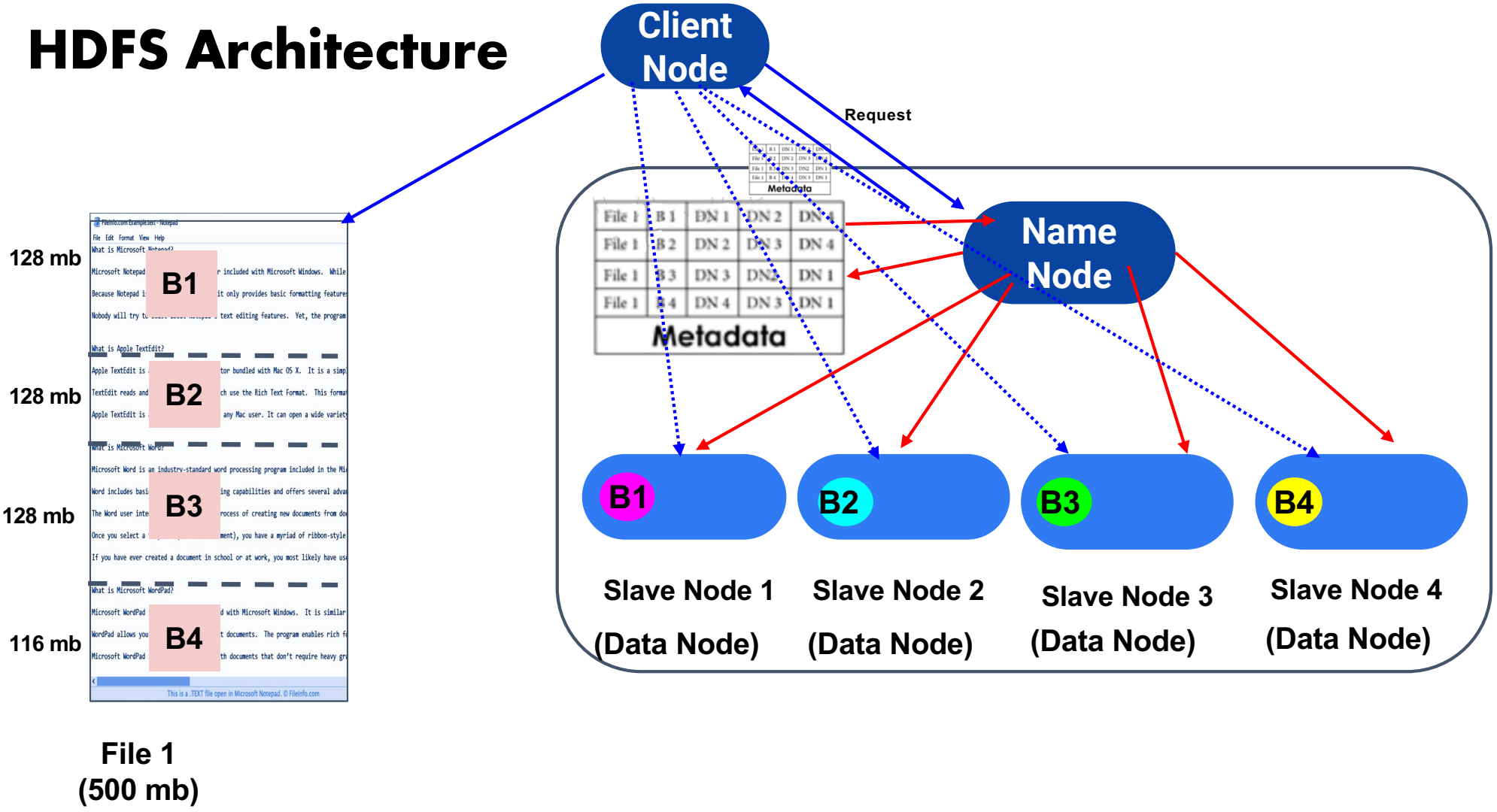
HDFS Architecture



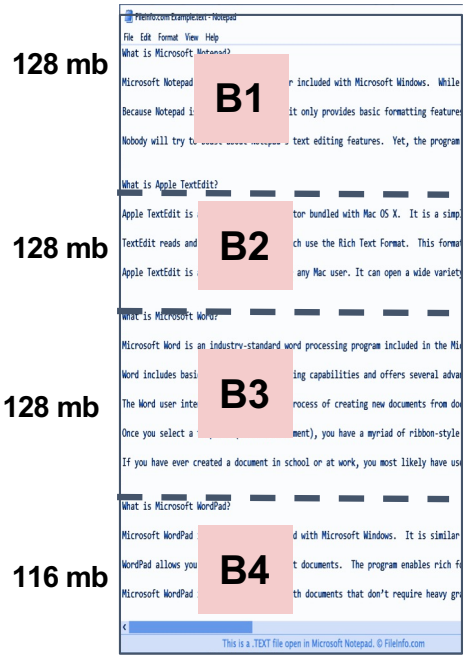
HDFS Architecture



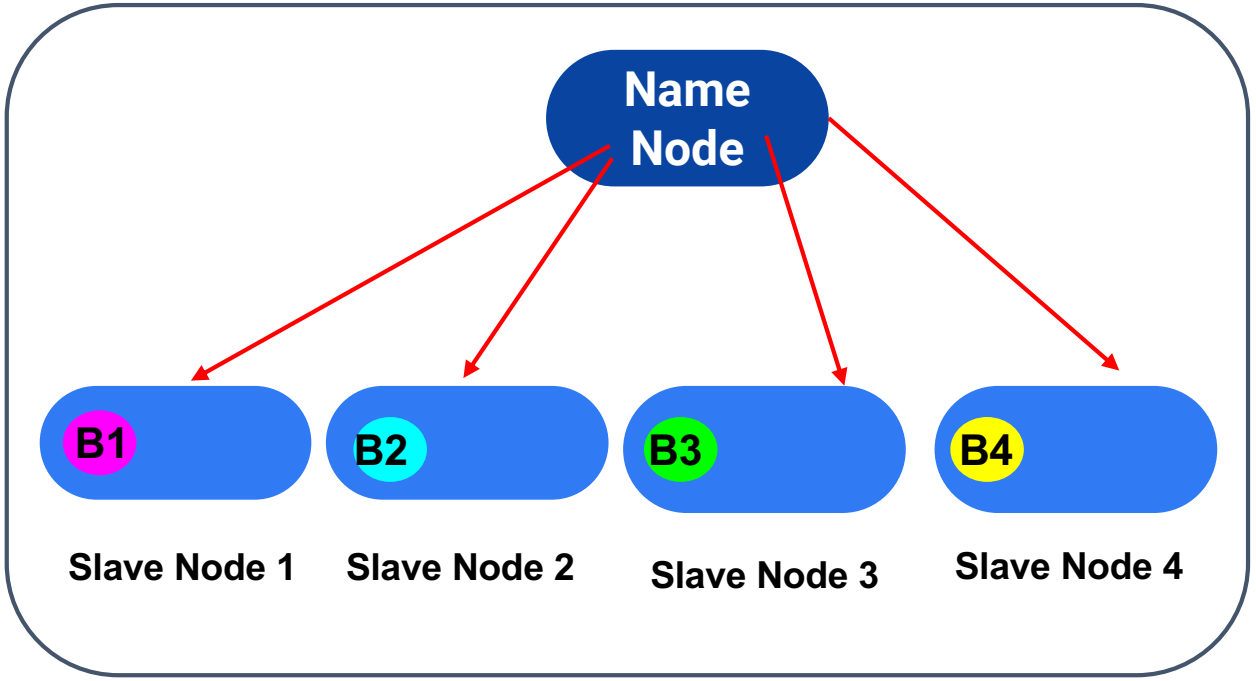
HDFS Architecture



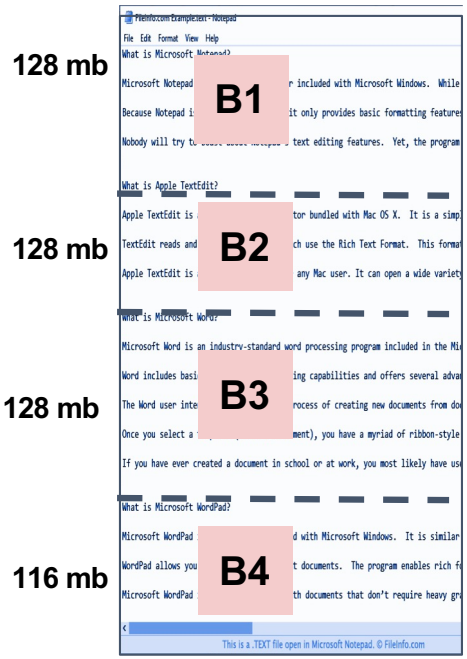
HDFS Architecture



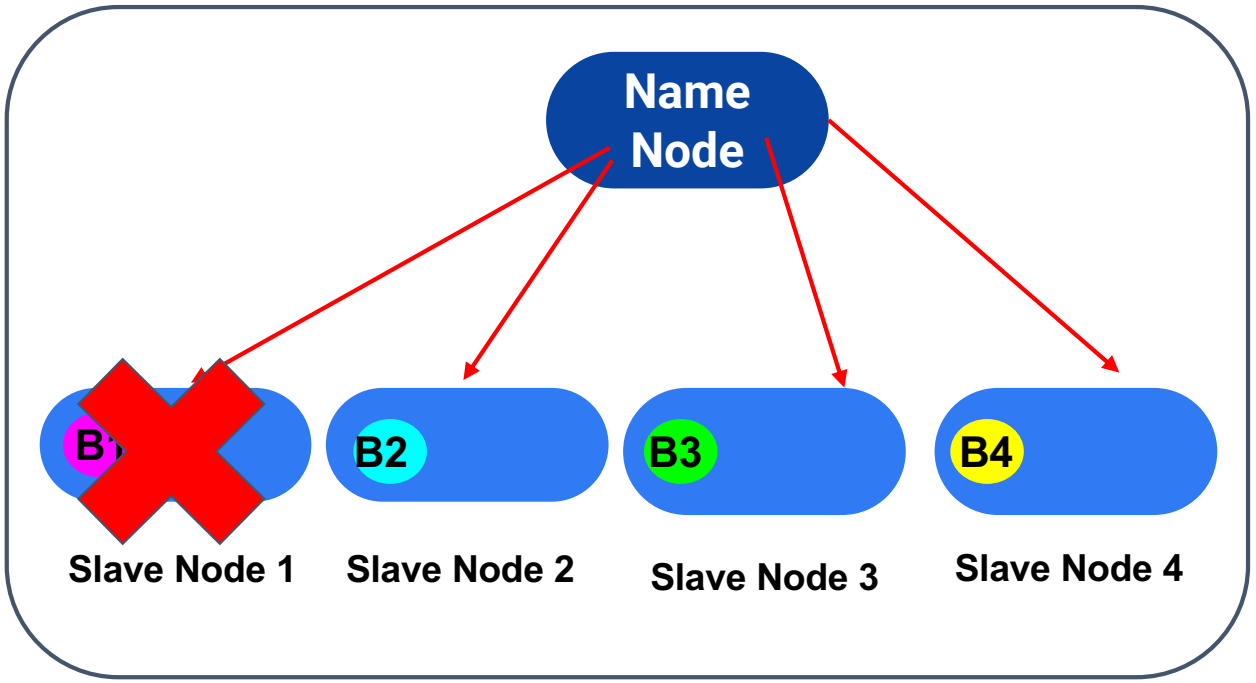
File 1
(500 mb)



HDFS Architecture



File 1
(500 mb)

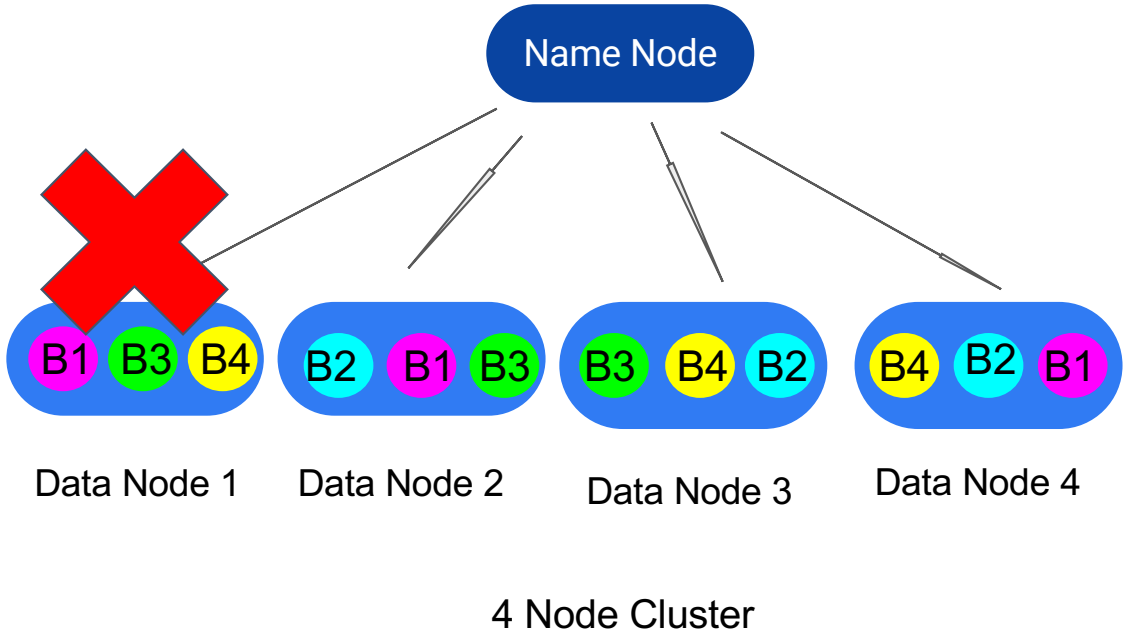


HDFS Architecture

Replication Factor

Default Hadoop Replication Factor:

3

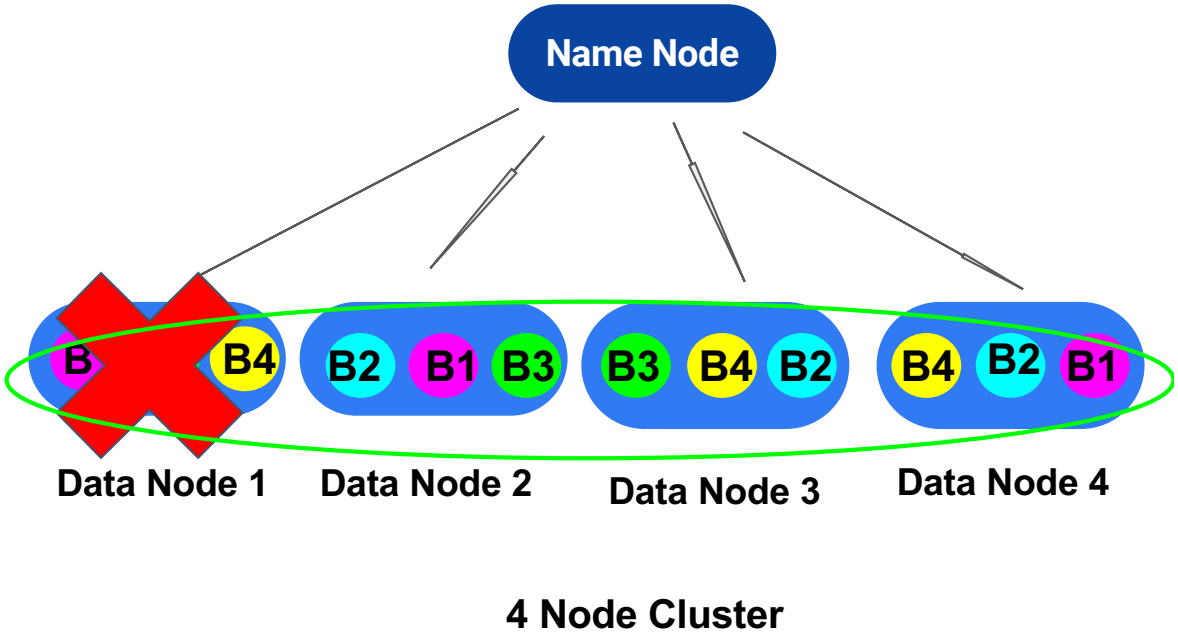


HDFS Architecture

Replication Factor

Default Hadoop Replication Factor:

3

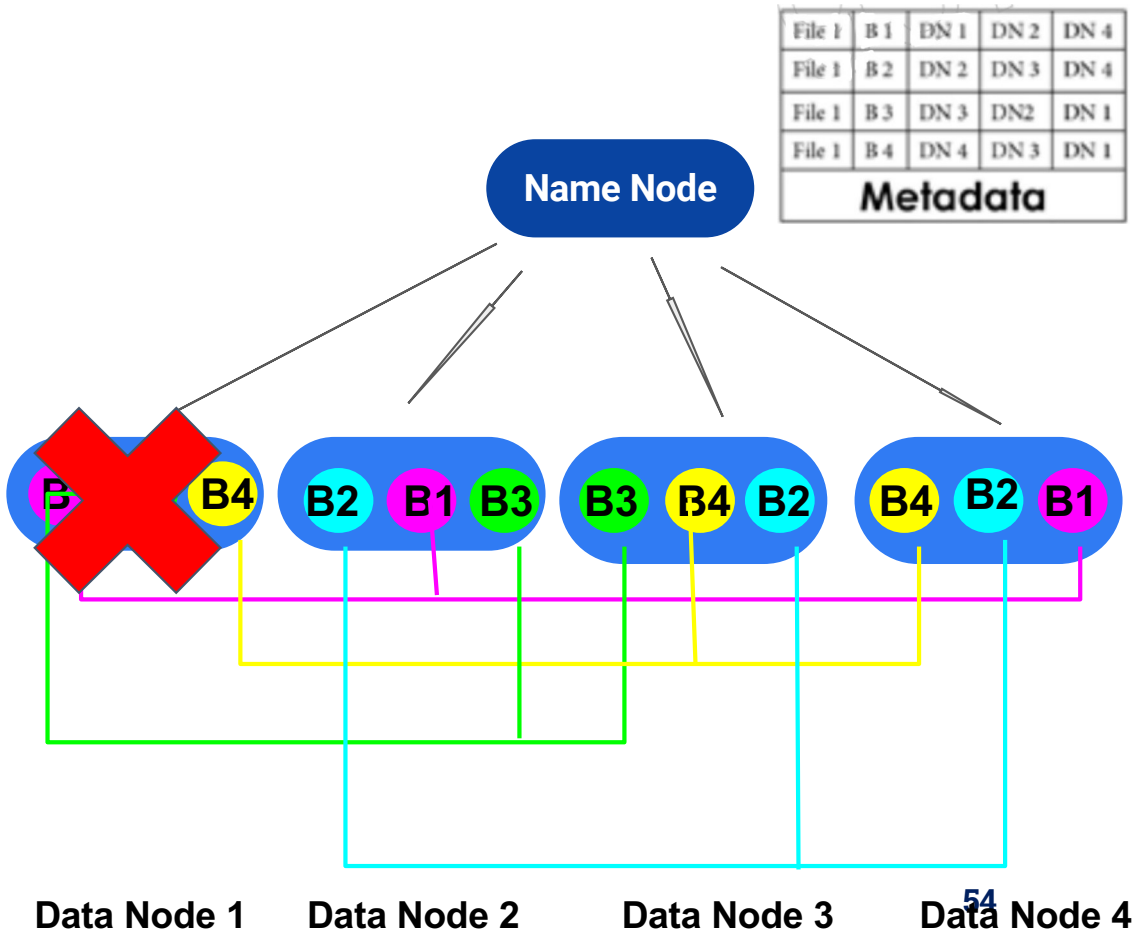


HDFS Architecture

Replication Factor

Default Hadoop Replication Factor:

3

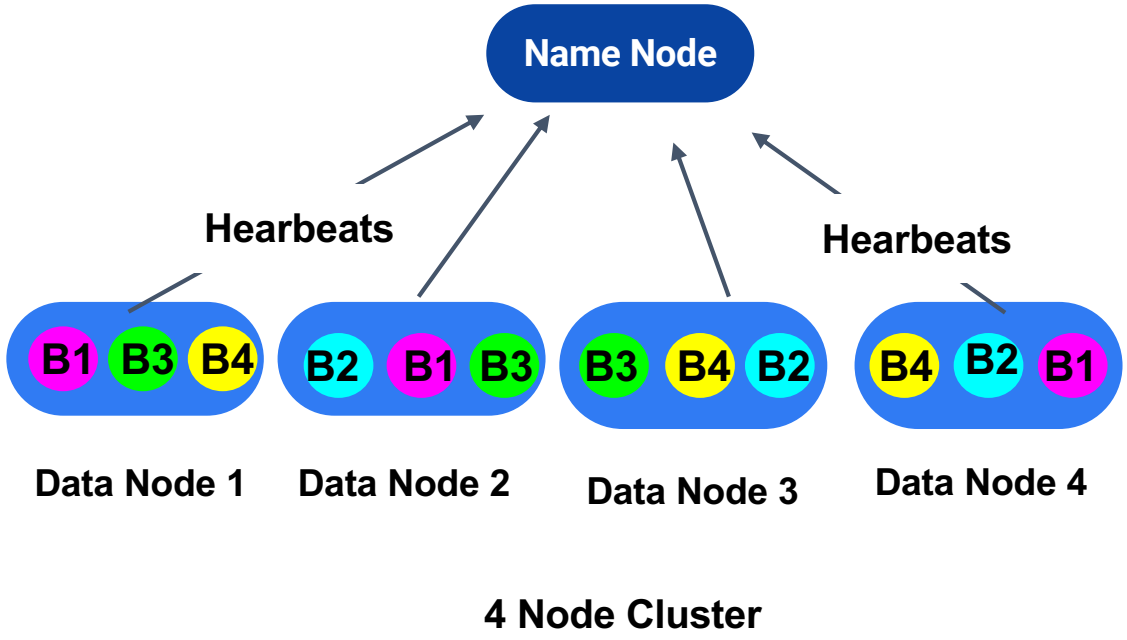


HDFS Architecture

Heart Beat

Each Data Node sends heart beats to Name Node in every 3 seconds

If a Name Node doesn't receive 10 consecutive heart beats, it assumes that the Data Node is dead or running very slow



Demo on HDFS with lab Platform

Accessing HDFS

- Fs shell Commands : fs shell is a command line utility which will allows clients to access HDFS files from the UNIX shell
- ` hadoop fs -<any of the below commands> `
- Commands
  - -cat
  - -mkdir
  - -ls
  - -put or -copyFromLocal
  - -get or –copyToLocal
  - -rm

Zoom quiz

Extra Reads

Useful info

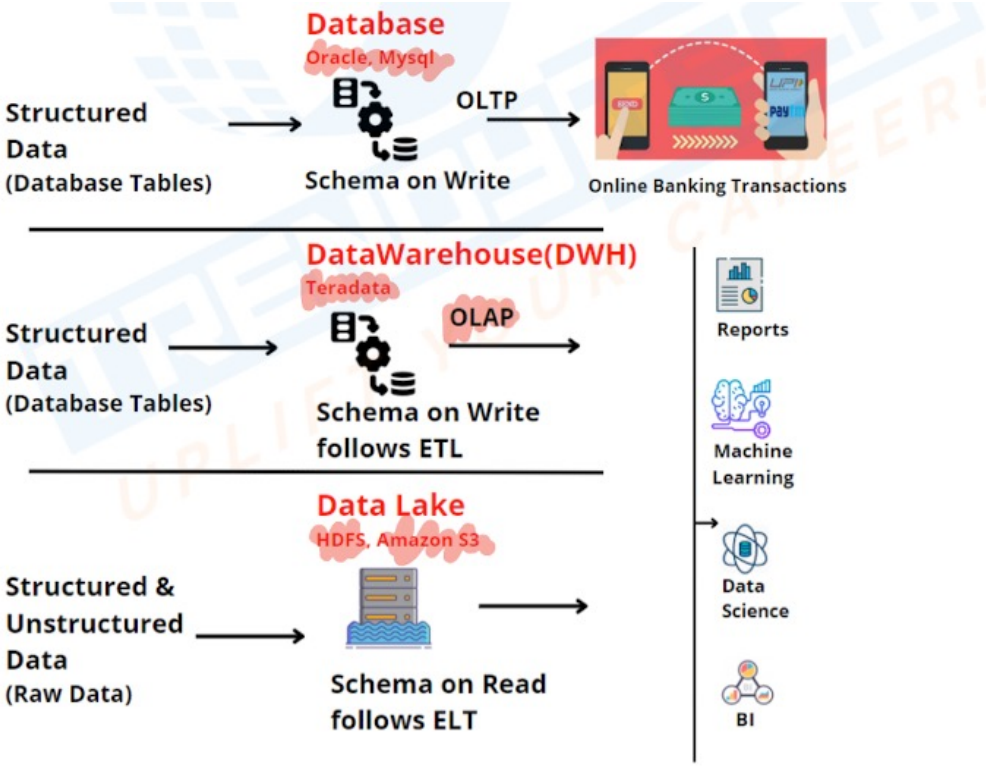
TIMESPRO

On-Premise Vs Cloud

Suppose you are a Startup and want to set-up a 50 node cluster for all the processing requirements.

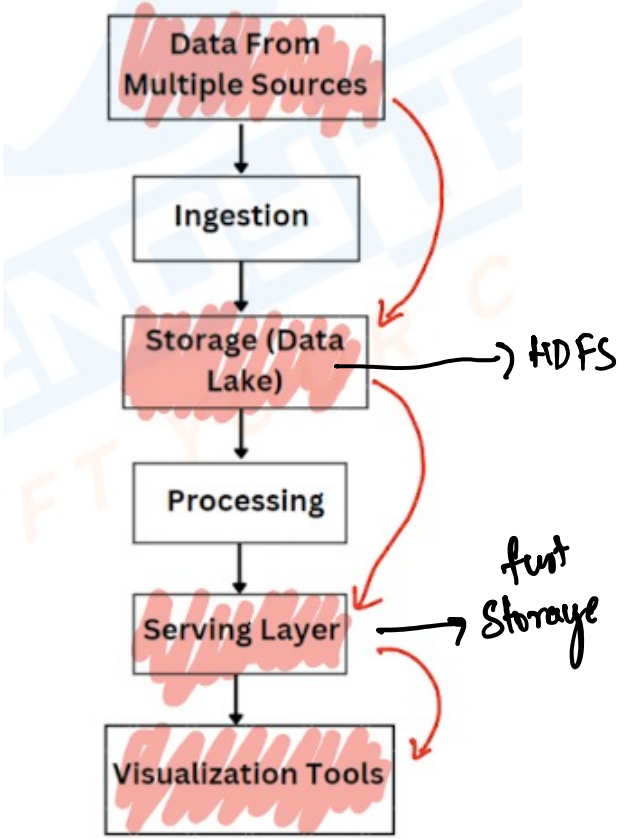
On-Premise Way	Cloud Way
-Buy the Needed Infrastructure, like space to hold the servers -Buy 50 servers -Setup a Cooling System -Hire a technical team to install and maintain needed software -Huge upfront cost / Capex & Opex -On-Premise systems are not very Scalable	-Infrastructure is taken care by Cloud providers -No need to buy servers -Setup the cluster with a Click of a Button -Low Maintenance Cost -No Upfront cost / Capex -Highly Scalable

Database Vs Data Warehouse Vs Data Lake



Data engineering Flow

- **Data Collection**: Gather data from various sources into a **Data Lake** for centralized storage.
- **Data Ingestion**: Use an **Ingestion Framework** to move data from different sources into the Data Lake.
- **Data Processing**: Follow the **ELT process**—load data into the Data Lake, then transform it (e.g., cleaning, aggregation, joins) as needed.
- **Data Serving**: Store processed data in the **Serving Layer** for visualization tools like Tableau and Power BI to display results graphically.



Database Vs Data Warehouse Vs Data Lake

Feature	Database	Data Warehouse (DWH)	Data Lake
Purpose	OLTP (Transactional Processing)	OLAP (Analytical Processing)	Insights from large volumes of data
Data Structure	Structured (Rows/ Columns)	Structured	Raw (Structured & Unstructured)
Data Scope	Recent data for performance	Historical data	Both recent & historical data
Examples	Oracle, MySQL	Teradata	HDFS, Amazon S3
Schema Approach	Schema on Write	Schema on Write	Schema on Read
Process	-	ETL (Extract, Transform, Load)	ELT (Extract, Load, Transform)
Storage Cost	High	High, but less than Database	Cost-effective
Challenges	-	Complex transformations, rigidity	Flexibility, but requires management

Assignment -Blog

Summary

- So far, we have learned:
  - What is Big Data
  - Characteristics if Big Data
  - What is Hadoop along with its Features
  - HDFS Architecture
  - Don’t forget Assignments and Revisit all slides

TIMESPRO

References :

- https://www.youtube.com/watch?v=fCnH6EvxemU&list=PLat4EDcV8F_I8Yyr5mNnhokWQWYYC20zv&index=2

- https://www.youtube.com/watch?v=pOgoLcbeZKk