HDFS Fundamentals

Hadoop Distributed File System

Professor Anis Koubaa

SE 446
Alfaisal University

Spring 2026

Outline

What is HDFS?

Definition

HDFS (Hadoop Distributed File System) is a distributed storage system designed to store very large files across multiple machines.

Key Features:

- **Distributed**: Files split across many machines
- **Fault-tolerant**: Data replicated for reliability
- **Scalable**: Add more machines as needed
- **Cost-effective**: Uses commodity hardware

HDFS Design Philosophy

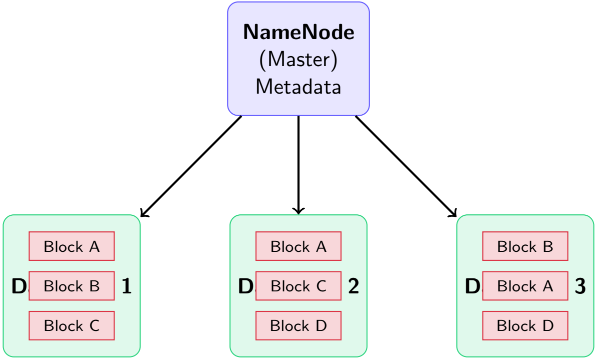
Optimized For

- Very large files (GB to PB)
- Streaming data access
- Write once, read many
- Commodity hardware

NOT Optimized For

- Low-latency access
- Many small files
- Random writes
- Interactive queries

HDFS Architecture: Master-Slave



NameNode
(Master)
Metadata

Block A
D Block B **1**
Block C

Block A
D Block C **2**
Block D

Block B
D Block A **3**
Block D

Stores metadata

Store actual data

NameNode: The Master

Responsibilities

- Manages file system **namespace** (directory tree)
- Stores **metadata**: file names, permissions, block locations
- Handles **client requests** for file operations
- Tracks **health** of DataNodes via heartbeats

Critical Component!

If NameNode fails, **the entire cluster is unavailable**.
Solutions: Secondary NameNode, HA (High Availability) mode

DataNode: The Workers

Responsibilities

- Store actual **data blocks** on local disk
- Serve **read/write requests** from clients
- Perform **block replication** as instructed by NameNode
- Send **heartbeats** to NameNode every 3 seconds

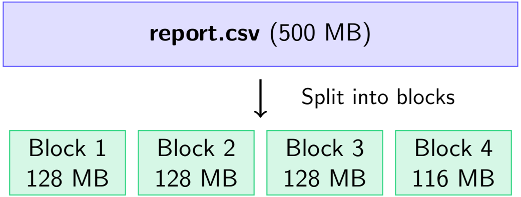
Commodity Hardware

DataNodes are designed to run on **cheap, commodity servers**.
Failures are expected and handled automatically.

Data Blocks

What is a Block?

Files in HDFS are split into fixed-size **blocks** (default: 128 MB).



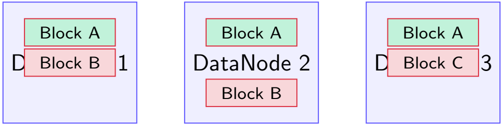
Why 128 MB? Minimize seek time, maximize throughput for large files.

Replication Factor

What is Replication?

Each block is copied to **multiple DataNodes** for fault tolerance.

Default replication factor: **3**



Block A is stored on 3 different nodes

Trade-off: More replicas = more fault tolerance, but 3× storage cost.

Storage Calculation

Example

File size: 500 MB

Replication factor: 3

Total storage used: $500 \times 3 = 1,500$ MB $=$ **1.5 GB**

Formula

$$\text{Total Storage} = \text{File Size} \times \text{Replication Factor}$$

Question: If you store 10 TB of data with replication factor 3, how much disk space do you need?

File Formats Comparison

Format	Type	Compression	Best For
CSV	Row-based	None	Simple exchange
JSON	Row-based	None	APIs, configs
Parquet	**Column-based**	**Yes**	**Analytics**
Avro	Row-based	Yes	Streaming
ORC	Column-based	Yes	Hive

For Big Data Analytics

Use **Parquet** — columnar storage with excellent compression.

Why Columnar Storage?

Row-Based (CSV)

ID	Name	Age	Salary
1	Alice	30	75000
2	Bob	25	65000

Column-Based (Parquet)

ID	Name	Age	Salary
1	Alice	30	75000
2	Bob	25	65000

Query: SELECT AVG(Salary)
Only reads Salary column!

Parquet Benefits

1. **Read only needed columns**
   Faster queries, less I/O
2. **Better compression**
   Similar values in a column compress well (10x smaller)
3. **Schema embedded**
   No need for external schema files
4. **Predicate pushdown**
   Filter data at storage level, before loading

Common HDFS Commands

```
# List files in a directory
hdfs dfs -ls /user/data/

# Create a directory
hdfs dfs -mkdir /user/mydata/

# Upload a file from local to HDFS
hdfs dfs -put local_file.csv /user/mydata/

# Download a file from HDFS to local
hdfs dfs -get /user/mydata/file.csv ./local/

# View file contents
hdfs dfs -cat /user/mydata/file.txt

# Delete a file
hdfs dfs -rm /user/mydata/old_file.csv

# Check disk usage
```

Summary: Key Takeaways

1. **HDFS** = Distributed file system for Big Data
2. **NameNode** (master) stores metadata; **DataNode** (workers) store data
3. Files split into **blocks** (128 MB default)
4. **Replication factor 3** for fault tolerance
5. Use **Parquet** for Big Data analytics (columnar, compressed)

Next Week

MapReduce: Distributed data processing paradigm

ExamGPT Quiz Topics

Be prepared to answer questions about:

- Role of NameNode vs DataNode
- Storage calculations with replication
- Block size and why it matters
- CSV vs Parquet comparison
- Basic HDFS commands

Quiz in 15 minutes!

Open ExamGPT and complete the Week 2B quiz.

Questions?

Prof. Anis Koubaa
akoubaa@alfaisal.edu