

Directed Works TD 1 : Hadoop Systems & MapReduce Paradigm

Learning Objectives:

- Explain the scaling limits of traditional RDBMS and justify Hadoop's architecture using the data locality principle
- Analyze the Master/Worker topology of Hadoop clusters, evaluating block storage, replication, and fault tolerance in HDFS and YARN
- Trace key-value pair transformations across all 5 stages of the MapReduce pipeline for real-world batch processing tasks

Exercise 1: Limitations of Traditional Systems & The Data Locality Imperative

Scenario: A legacy financial enterprise stores a **50 Terabyte (TB)** central audit dataset on a single high-end Relational Database Management System (RDBMS) server. The analytics team needs to execute a nightly batch job that reads the entire dataset to compute compliance metrics.

Questions:

1. **Transfer Bottleneck Calculation:** Assuming the data resides on a dedicated storage node and must be transferred over a standard **1 Gbps (Gigabit per second)** network link to a separate processing server, calculate the exact theoretical network transfer time (in hours and days) required just to move the dataset before computation begins.
(Note: Use $1 \text{ TB} = 10^{12} \text{ bytes} = 8 \times 10^{12} \text{ bits}$).
2. **Single-Machine Bottlenecks:** Identify three physical hardware bottlenecks (CPU, Disk I/O, RAM) and one organizational risk associated with **vertical scaling** (scale-up) for petabyte-scale workloads.
3. **Data Locality Principle:** Explain how Hadoop's "**Data Locality**" paradigm (*Code* → *Data*) fundamentally resolves the network bottleneck calculated in Question 1 compared to the traditional architecture (*Data* → *Code*).

Exercise 2: Hadoop Cluster Topology, HDFS Mechanics, and Fault Tolerance

Scenario: An e-commerce platform generates a **1 TB (1,024 GB)** web log file. The system administrator uploads this file to a Hadoop cluster configured with:

- Default HDFS Block Size = **128 MB**
- Default Replication Factor = **3**

Questions:

1. **Storage Calculations:**
 - **a)** Calculate the exact number of HDFS blocks created for this 1 TB file.
 - **b)** Calculate the total physical storage space consumed across the DataNodes in the cluster to store this file with full replication.

2. **Cluster Topology & Node Roles:** Complete the matrix below by matching each Hadoop component with its primary role and node type:

Component	Node Type (Master / Worker)	Primary Responsibility
NameNode		Stores file namespace metadata & block mapping
DataNode		Stores actual data blocks & executes local reads/writes
ResourceManager (YARN)		Allocates cluster compute resources to applications
NodeManager (YARN)		Manages execution containers on individual worker nodes

3. **Fault Tolerance Mechanism:** Suppose *DataNode #4* unexpectedly crashes while holding Replica 1 of *Block #512*. Trace step-by-step how the **NameNode** detects this failure and automatically restores the configured replication factor without data loss.

Exercise 3: MapReduce Workflow & Paradigm Trace (Case Study: Server Access Logs)

Scenario: A system security team wants to monitor a web server cluster by counting the occurrences of **HTTP Error Status Codes** (e.g., 404, 500, 403) from server log streams.

You are provided with two input splits processed by two parallel mappers:

- **Input Split 1 (Mapper 1):**
 - Line 1: 192.168.1.10 - GET /index.html 200
 - Line 2: 10.0.0.5 - GET /login.php 404
 - Line 3: 192.168.1.12 - POST /checkout 500
- **Input Split 2 (Mapper 2):**
 - Line 4: 10.0.0.5 - GET /admin 403
 - Line 5: 172.16.0.2 - GET /images/logo.png 404
 - Line 6: 192.168.1.10 - GET /cart 404

Questions:

1. **Map Phase:** Define the signature of the Map function in terms of (InputKey, InputValue) -> List(IntermediateKey, IntermediateValue). Write down the exact key-value pairs emitted by Mapper 1 and Mapper 2.
2. **Shuffle & Sort Phase:** Detail the output of the Shuffle & Sort phase by showing how intermediate key-value pairs are grouped into (Key, List(Values)) lists prior to reaching the Reducer tasks.
3. **Reduce Phase:** Define the Reduce function logic and list the final output key-value pairs written back to HDFS.
4. **Optimization Concept:** Explain how introducing a local **Combiner** (mini-reducer) on each mapper node optimizes cluster bandwidth during the Shuffle phase.