**BỘ GIÁO DỤC & ĐÀO TẠO**

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-HCM

KHOA CÔNG NGHỆ THÔNG TIN

🙡&🙣

A blue and white logo

Description automatically generated

Hadoop

Môn học: Xử lý phân tích dữ liệu trực tuyến

Bộ môn: Hệ thống thông tin

Mã lớp: CQ2022/1

GVHD: Phạm Minh Tú

Họ và tên: Nguyễn Đăng Trí

MSSV: 22120383

*Thành phố Hồ Chí Minh, ngày 10 tháng 5 năm 2025*

Mục lục

[1. Yêu cầu cơ bản 1](#_Toc197812012)

[1.1. Tạo folder 1](#_Toc197812013)

[1.2. Copy file 2](#_Toc197812014)

[1.3. Xoá file 3](#_Toc197812015)

[2. Yêu cầu nâng cao 4](#_Toc197812016)

[2.1. Tạo folder /logs/2025/03 trên HDFS. 4](#_Toc197812017)

[2.2. Upload các file log giả lập vào đó. 4](#_Toc197812018)

[2.3. Kiểm tra dung lượng đã sử dụng trên HDFS. 4](#_Toc197812019)

[2.4. Kiểm tra quyền truy cập của các file (xem ai có quyền đọc, ghi, thực thi). 4](#_Toc197812020)

[2.5. Sử dụng lệnh du và dfsadmin -report để kiểm tra dung lượng từng thư mục và tình trạng của cụm Hadoop. 4](#_Toc197812021)

[2.6. Giả lập tình huống có một file rất lớn (trên 1GB) 5](#_Toc197812022)

[2.6.1. Yêu cầu: 5](#_Toc197812023)

[2.6.2. Chia nhỏ file lớn (split): 5](#_Toc197812024)

[2.6.3. Upload từng phần lên HDFS 5](#_Toc197812025)

[2.6.4. Hợp nhất các phần lại trong HDFS 5](#_Toc197812026)

[2.6.5. Kiểm tra phân phối block của file 5](#_Toc197812027)

[2.7. Giả sử một DataNode bị lỗi, hãy tìm hiểu cách Hadoop đảm bảo dữ liệu không bị mất 5](#_Toc197812028)

[3. Video thực hành 6](#_Toc197812029)

# Yêu cầu cơ bản

## Tạo folder

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## Copy file

A computer screen shot of a black screen

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## Xoá file

A screenshot of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

# Yêu cầu nâng cao

## Tạo folder /logs/2025/03 trên HDFS.

hdfs dfs -mkdir -p /logs/2025/03

Tham số -p là viết tắt của parents. Khi sử dụng tham số này, HDFS sẽ tự động tạo các thư mục cha cần thiết nếu chúng chưa tồn tại.

## Upload các file log giả lập vào đó.

hdfs dfs -put ~/fake-logs/\* /logs/2025/03/

## Kiểm tra dung lượng đã sử dụng trên HDFS.

hdfs dfs -du -h /logs/2025/03

Tham số -du là viết tắt của disk usage giúp hiển thị dung lượng sử dụng.

Tham số -h là viết tắt của human-readable, nghĩa là dung lượng theo định dạng dễ đọc hơn (KB, MB, GB...) thay vì số byte thuần túy.

## Kiểm tra quyền truy cập của các file (xem ai có quyền đọc, ghi, thực thi).

hdfs dfs -ls /logs/2025/03

Lệnh này sẽ hiển thị quyền truy cập (rwx), chủ sở hữu (owner), nhóm (group).

**Ví dụ đầu ra:**

-rw-r--r-- 3 hadoop hadoopusr 1048576 2025-05-10 10:00 /logs/2025/03/log1.txt

## Sử dụng lệnh du và dfsadmin -report để kiểm tra dung lượng từng thư mục và tình trạng của cụm Hadoop.

**Dùng lệnh du:**

hdfs dfs -du -h /logs/2025

**Dùng lệnh dfsadmin -report:**

hdfs dfsadmin -report

Lệnh này cung cấp thông tin chi tiết về dung lượng còn trống, đã dùng, số lượng DataNode đang online/offline,...

## Giả lập tình huống có một file rất lớn (trên 1GB)

### Yêu cầu:

* Chia file thành các phần nhỏ (split).
* Tải từng phần lên HDFS và hợp nhất lại.
* Kiểm tra sự phân phối các block của file trên các DataNode bằng lệnh hdfs fsck.

### Chia nhỏ file lớn (split):

Giả sử file big\_log.txt lớn hơn 1GB:

split -b 256M big\_log.txt part\_

File sẽ được chia thành các phần part\_aa, part\_ab,...

### Upload từng phần lên HDFS

hdfs dfs -mkdir /logs/2025/03/bigfile\_parts

hdfs dfs -put part\_\* /logs/2025/03/bigfile\_parts/

### Hợp nhất các phần lại trong HDFS

hdfs dfs -cat /logs/2025/03/bigfile\_parts/part\_\* | hdfs dfs -put - /logs/2025/03/big\_log\_combined.txt

### Kiểm tra phân phối block của file

hdfs fsck /logs/2025/03/big\_log\_combined.txt -files -blocks -locations

Lệnh trên hiển thị các block của file và DataNode tương ứng đang lưu trữ block đó.

## Giả sử một DataNode bị lỗi, hãy tìm hiểu cách Hadoop đảm bảo dữ liệu không bị mất

Nếu một DataNode bị lỗi, Hadoop đảm bảo an toàn dữ liệu như thế nào?

Hadoop sử dụng cơ chế replication: Mỗi block của file sẽ được sao chép lên n bản trên các DataNode khác nhau.

Khi một DataNode bị lỗi:

* NameNode phát hiện lỗi qua việc không nhận heartbeat.
* Hadoop sẽ tự động replicate các block bị mất sang DataNode còn sống để đảm bảo số lượng bản sao luôn đủ.
* Dữ liệu không bị mất miễn là có ít nhất 1 bản sao còn tồn tại.

Kiểm tra replication của file bằng:

hdfs fsck /logs/2025/03/big\_log\_combined.txt -files -blocks -locations

Kết quả:

Status: HEALTHY

Total size: 1200000000 B

Total dirs: 0

Total files: 1

Total symlinks: 0

Total blocks (validated): 9 (avg. block size 133333333 B)

Minimally replicated blocks: 9 (100.0 %)

Over-replicated blocks: 0 (0.0 %)

Under-replicated blocks: 0 (0.0 %)

Mis-replicated blocks: 0 (0.0 %)

Default replication factor: 1

Average block replication: 1.0

Corrupt blocks: 0

Missing replicas: 0 (0.0 %)

Number of data-nodes: 1

Number of racks: 1

FSCK ended at Sat May 10 21:26:00 ICT 2025 in 1 milliseconds

Ở đây, file big\_log\_file\_combined.txt chỉ có 1 bản sao   
(Default replication factor: 1), do đó nếu DataNode duy nhất bị lỗi, dữ liệu sẽ bị mất

Để tăng số replication factor, ta chỉnh cấu hình HDFS (hdfs-site.xml)

<property>

<name>dfs.replication</name>

<value>3</value>

</property>

Hoặc có thể thay đổi trên 1 file cụ thể bằng lệnh

hdfs dfs -setrep -R 3 <file\_path>

# Video thực hành

[Quá trình thực hành phần bài tập](https://www.youtube.com/watch?v=dw2D6b7Ost0)