**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO BỘ MÔN:**

**NHẬP MÔN BIGDATA**

TP. Hồ Chí Minh, tháng 6 năm 2021

**NHẬN XÉT**

Sinh viên thực hiện: Nhóm 10 MSSV

Ngô Xuân Thắng 18110368

Nguyễn Ngọc Trung Hiếu 18110283

Trần Ngọc Hoàng 18110288

Giảng viên hướng dẫn:

**Lê Thị Minh Châu**

**BÁO CÁO CUỐI KÌ:**

**TÌM HIỂU MÔ HÌNH MAP REDUCE TRÊN HADOOP**

**Ký tên**

**Lê Thị Minh Châu**

**Mục lục**

[Chương 1. Giới thiệu về hadoop 1](#_Toc74785644)

[1.1. Các khái niệm của hadoop 1](#_Toc74785645)

[1.1.1. Hadoop giải quyết những vấn đề 1](#_Toc74785646)

[-Xử lý và làm việc khối lượng dữ liệu khổng lồ tính bằng Petabyte. 1](#_Toc74785647)

[1.1.2. Kiến trúc của Hadoop 1](#_Toc74785648)

[1.1.3. Hadoop hoạt động như thế nào? 3](#_Toc74785649)

[1.1.4. Lợi ích của việc dùng hadoop 4](#_Toc74785650)

[1.2. Cài đặt cụm hadoop nhiều node 4](#_Toc74785651)

[Chương 2. Giới thiệu về Map Reduce 5](#_Toc74785652)

[2.1. Khái niệm MapReduce 5](#_Toc74785653)

[2.2. Các hàm chính của MapReduce 5](#_Toc74785654)

[2.3. Cách thức hoạt động của MapReduce 6](#_Toc74785655)

[2.3.1. Nguyên tắc hoạt động 6](#_Toc74785656)

[2.3.2. Các bước hoạt động của MapReduce 6](#_Toc74785657)

[2.3.3. Luồng dữ liệu nền tảng của Mapreduce 7](#_Toc74785658)

[2.4. Các ưu điểm nổi bật của MapReduce 8](#_Toc74785659)

[Chương 3. Demo MapReduce trên Hadoop 9](#_Toc74785660)

[3.1. Demo1 9](#_Toc74785661)

[3.2. Demo2 9](#_Toc74785662)

[3.3. Demo3 9](#_Toc74785663)

[Chương 4. KẾT LUẬN 10](#_Toc74785664)

[Chương 5. TÀI LIỆU THAM KHẢO 11](#_Toc74785665)

# GIỚI THIỆU VỀ HADOOP

## Các khái niệm của hadoop

Hadoop là một Apache framework mã nguồn mở cho phép phát triển các ứng dụng phân tán (distributed processing) để lưu trữ và quản lý các tập dữ liệu lớn. Hadoop hiện thực mô hình MapReduce, mô hình mà ứng dụng sẽ được chia nhỏ ra thành nhiều phân đoạn khác nhau được chạy song song trên nhiều node khác nhau. Hadoop được viết bằng Java tuy nhiên vẫn hỗ trợ C++, Python, Perl bằng cơ chế streaming.

### Hadoop giải quyết những vấn đề

### -Xử lý và làm việc khối lượng dữ liệu khổng lồ tính bằng Petabyte.

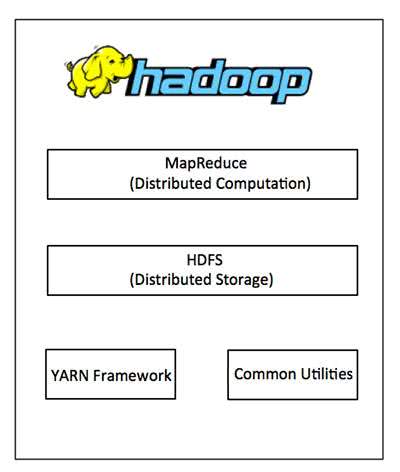
-Xử lý trong môi trường phân tán, dữ liệu lưu trữ ở nhiều phần cứng khác nhau, yêu cầu xử lý đồng bộ

-Các lỗi xuất hiện thường xuyên.

-Băng thông giữa các phần cứng vật lý chứa dữ liệu phân tán có giới hạn.

### Kiến trúc của Hadoop

Một cụm Hadoop nhỏ gồm 1 master node và nhiều worker/slave node. Toàn bộ cụm chứa 2 lớp, một lớp MapReduce Layer và lớp kia là HDFS Layer. Mỗi lớp có các thành phần liên quan riêng. Master node gồm JobTracker, TaskTracker, NameNode, và DataNode. Slave/worker node gồm DataNode, và TaskTracker. Cũng có thể slave/worker node chỉ là dữ liệu hoặc node để tính toán.



Hình: kiến trúc của hadoop

#### Hadoop Distributed File System (HDFS)

Đây là hệ thống file phân tán cung cấp truy cập thông lượng cao cho ứng dụng khai thác dữ liệu. Hadoop Distributed File System (HDFS) là hệ thống tập tin ảo. Khi chúng ta di chuyển 1 tập tin trên HDFS, nó tự động chia thành nhiều mảnh nhỏ. Các đoạn nhỏ của tập tin sẽ được nhân rộng và lưu trữ trên nhiều máy chủ khác để tăng sức chịu lỗi và tính sẵn sàng cao.

HDFS sử dụng kiến trúc master/slave, trong đó master gồm một NameNode để quản lý hệ thống file metadata và một hay nhiều slave DataNodes để lưu trữ dữ liệu thực tại.

Một tập tin với định dạng HDFS được chia thành nhiều khối và những khối này được lưu trữ trong một tập các DataNodes. NameNode định nghĩa ánh xạ từ các khối đến các DataNode. Các DataNode điều hành các tác vụ đọc và ghi dữ liệu lên hệ thống file. Chúng cũng quản lý việc tạo, huỷ, và nhân rộng các khối thông qua các chỉ thị từ NameNode.

#### Hadoop MapReduce

Đây là hệ thống dựa trên YARN dùng để xử lý song song các tập dữ liệu lớn. Là cách chia một vấn đề dữ liệu lớn hơn thành các đoạn nhỏ hơn và phân tán nó trên nhiều máy chủ. Mỗi máy chủ có 1 tập tài nguyên riêng và máy chủ xử lý dữ liệu trên cục bộ. Khi máy chủ xử lý xong dữ liệu, chúng sẽ gởi trở về máy chủ chính.

MapReduce gồm một single master (máy chủ) JobTracker và các slave (máy trạm) TaskTracker trên mỗi cluster-node. Master có nhiệm vụ quản lý tài nguyên, theo dõi quá trình tiêu thụ tài nguyên và lập lịch quản lý các tác vụ trên các máy trạm, theo dõi chúng và thực thi lại các tác vụ bị lỗi. Những máy slave TaskTracker thực thi các tác vụ được master chỉ định và cung cấp thông tin trạng thái tác vụ (task-status) để master theo dõi.

JobTracker là một điểm yếu của Hadoop Mapreduce. Nếu JobTracker bị lỗi thì mọi công việc liên quan sẽ bị ngắt quãng.

#### Hadoop Common

Đây là các thư viện và tiện ích cần thiết của Java để các module khác sử dụng. Những thư viện này cung cấp hệ thống file và lớp OS trừu tượng, đồng thời chứa các mã lệnh Java để khởi động Hadoop.

#### Hadoop YARN

Quản lý tài nguyên của các hệ thống lưu trữ dữ liệu và chạy phân tích.

### Hadoop hoạt động như thế nào?

**Giai đoạn 1:**

Một user hay một ứng dụng có thể submit một job lên Hadoop (hadoop job client) với yêu cầu xử lý cùng các thông tin cơ bản:

+Nơi lưu (location) dữ liệu input, output trên hệ thống dữ liệu phân tán.

+Các java class ở định dạng jar chứa các dòng lệnh thực thi các hàm map và reduce.

+Các thiết lập cụ thể liên quan đến job thông qua các thông số truyền vào.

**Giai đoạn 2:**

Hadoop job client submit job (file jar, file thực thi) và các thiết lập cho JobTracker. Sau đó, master sẽ phân phối tác vụ đến các máy slave để theo dõi và quản lý tiến trình các máy này, đồng thời cung cấp thông tin về tình trạng và chẩn đoán liên quan đến job-client.

**Giai đoạn 3:**

TaskTrackers trên các node khác nhau thực thi tác vụ MapReduce và trả về kết quả output được lưu trong hệ thống file.

Khi “chạy Hadoop” có nghĩa là chạy một tập các trình nền – daemon, hoặc các chương trình thường trú, trên các máy chủ khác nhau trên mạng của bạn. Những trình nền có vai trò cụ thể, một số chỉ tồn tại trên một máy chủ, một số có thể tồn tại trên nhiều máy chủ.

Các daemon bao gồm:

+NameNode

+DataNode

+SecondaryNameNode

+JobTracker

+TaskTracker

### Lợi ích của việc dùng hadoop

Các điểm thuận lợi khi dùng Hadoop:

-Robus and Scalable – Có thể thêm node mới và thay đổi chúng khi cần.

-Affordable and Cost Effective – Không cần phần cứng đặc biệt để chạy Hadoop.

-Adaptive and Flexible – Hadoop được xây dựng với tiêu chí xử lý dữ liệu có cấu trúc và không cấu trúc.

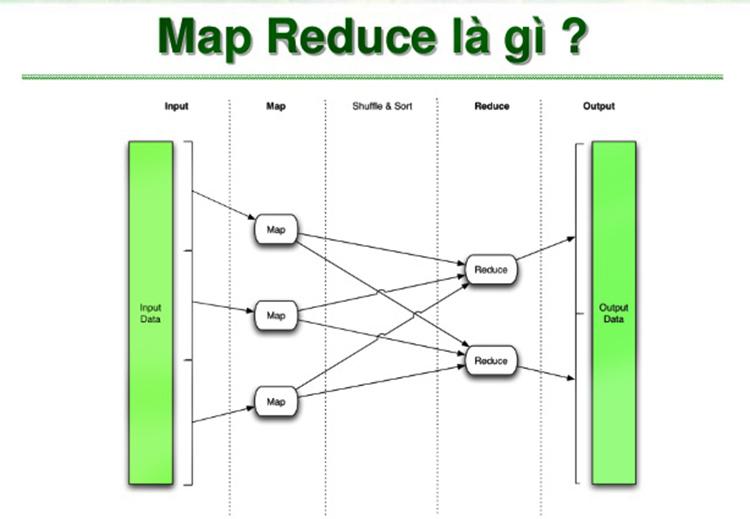
-Highly Available and Fault Tolerant – Khi 1 node lỗi, nền tảng Hadoop tự động chuyển sang node khác.

## Cài đặt cụm hadoop nhiều node

# GIỚI THIỆU VỀ MAP REDUCE

## Khái niệm MapReduce

MapReduce là mô hình được thiết kế độc quyền bởi Google, nó có khả năng lập trình xử lý các tập dữ liệu lớn song song và phân tán thuật toán trên 1 cụm máy tính. MapReduce trở thành một trong những thành ngữ tổng quát hóa trong thời gian gần đây.



**Định nghĩa:**MapReduce sẽ  bao gồm những thủ tục sau: thủ tục 1 Map() và 1 Reduce(). Thủ tục Map() bao gồm lọc (filter) và phân loại (sort) trên dữ liệu khi thủ tục khi thủ tục Reduce() thực hiện quá trình tổng hợp dữ liệu. Đây là mô hình dựa vào các khái niệm biển đối của bản đồ và reduce những chức năng lập trình theo hướng chức năng. Thư viện của thủ tục Map() và Reduce() sẽ được viết bằng nhiều loại ngôn ngữ khác nhau. Thủ tục được cài đặt miễn phí và được sử dụng phổ biến nhất là là Apache Hadoop.

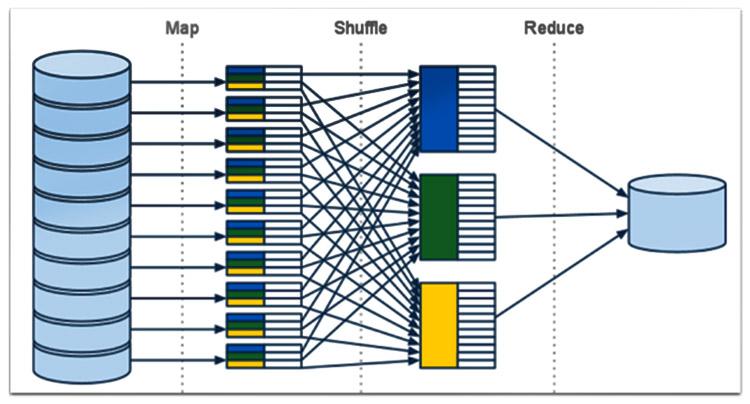
## Các hàm chính của MapReduce

MapReduce có 2 hàm chính là Map() và Reduce(), đây là 2 hàm đã được định nghĩa bởi người dùng và nó cũng chính là 2 giai đoạn liên tiếp trong quá trình xử lý dữ liệu của MapReduce. Nhiệm vụ cụ thể của từng hàm như sau:

- **Hàm Map():** có nhiệm vụ nhận Input cho các cặp giá trị/ khóa và output chính là tập những cặp giá trị/khóa trung gian. Sau đó, chỉ cần ghi xuống đĩa cứng và tiến hành thông báo cho các hàm Reduce() để trực tiếp nhận dữ liệu.

**- Hàm Reduce():**có nhiệm vụ tiếp nhận từ khóa trung gian và những giá trị tương ứng với lượng từ khóa đó. Sau đó, tiến hành ghép chúng lại để có thể tạo thành một tập khóa khác nhau. Các cặp khóa/giá trị này thường sẽ thông qua một con trỏ vị trí để đưa vào các hàm reduce. Quá trình này sẽ giúp cho lập trình viên quản lý dễ dàng hơn một lượng danh sách cũng như phân bổ giá trị sao cho phù hợp nhất với bộ nhớ hệ thống.

- Ở giữa Map và Reduce thì còn 1 bước trung gian đó chính là **Shuffle**. Sau khi Map hoàn thành xong công việc của mình thì Shuffle sẽ làm nhiệm vụ chính là thu thập cũng như tổng hợp từ khóa/giá trị trung gian đã được map sinh ra trước đó rồi chuyển qua cho Reduce tiếp tục xử lý.



Hình: Các hàm của Mapreduce

## Cách thức hoạt động của MapReduce

### Nguyên tắc hoạt động

Mapreduce hoạt động dựa vào nguyên tắc chính là “Chia để trị”, như sau:

- Phân chia các dữ liệu cần xử lý thành nhiều phần nhỏ trước khi thực hiện.

- Xử lý các vấn đề nhỏ theo phương thức song song trên các máy tính rồi phân tán hoạt động theo hướng độc lập.

- Tiến hành tổng hợp những kết quả thu được để đề ra được kết quả sau cùng.

### Các bước hoạt động của MapReduce

- Bước 1: Tiến hành chuẩn bị các dữ liệu đầu vào để cho Map() có thể xử lý.

- Bước 2: Lập trình viên thực thi các mã Map() để xử lý.

- Bước 3: Tiến hành trộn lẫn các dữ liệu được xuất ra bởi Map() vào trong Reduce Processor - Bước 4: Tiến hành thực thi tiếp mã Reduce() để có thể xử lý tiếp các dữ liệu cần thiết. - Bước 5: Thực hiện tạo các dữ liệu xuất ra cuối cùng.

### Luồng dữ liệu nền tảng của Mapreduce

- Input Reader

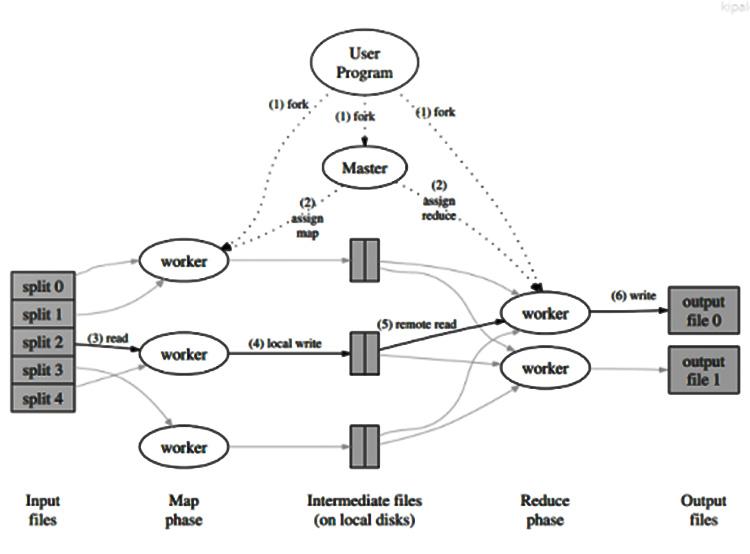
- Map Function

- Partition Function

- Compare Function

- Reduce Function

- Output Writer



## Các ưu điểm nổi bật của MapReduce

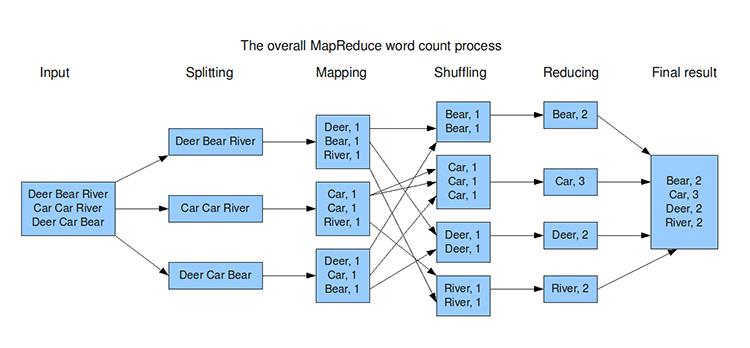
Mapreduce được ưa chuộng sử dụng như vậy bởi nó sở hữu nhiều ưu điểm vượt trội như sau:

- MapReduce có khả năng xử lý dễ dàng mọi bài toán có lượng dữ liệu lớn nhờ khả năng tác vụ phân tích và tính toán phức tạp. Nó có thể xử lý nhanh chóng cho ra kết quả dễ dàng chỉ trong khoảng thời gian ngắn.

- Mapreduce có khả năng chạy song song trên các máy có sự phân tán khác nhau. Với khả năng hoạt động độc lập kết hợp phân tán, xử lý các lỗi kỹ thuật để mang lại nhiều hiệu quả cho toàn hệ thống.

- MapRedue có khả năng thực hiện trên nhiều nguồn ngôn ngữ lập trình khác nhau như: Java ,C/ C++, Python, Perl, Ruby,… tương ứng với nó là những thư viện hỗ trợ. –

Như bạn đã biết, mã độc trên internet ngày càng nhiều hơn nên việc xử lý những đoạn mã độc này cũng trở nên rất phức tạp và tốn kém nhiều thời gian. Chính vì vậy, các ứng dụng MapReduce dần hướng đến quan tâm nhiều hơn cho việc phát hiện các mã độc để có thể xử lý chúng. Nhờ vậy, hệ thống mới có thể vận hành trơn tru và được bảo mật nhất.



Hình: Mapreduce có khả năng xử lý nhanh chóng lượng dữ liệu lớn

# DEMO MAPREDUCE TRÊN HADOOP

## Demo1

1. Kiểm tra hadoop, java version

- kiểm tra hadoop, java version.

hadoop version

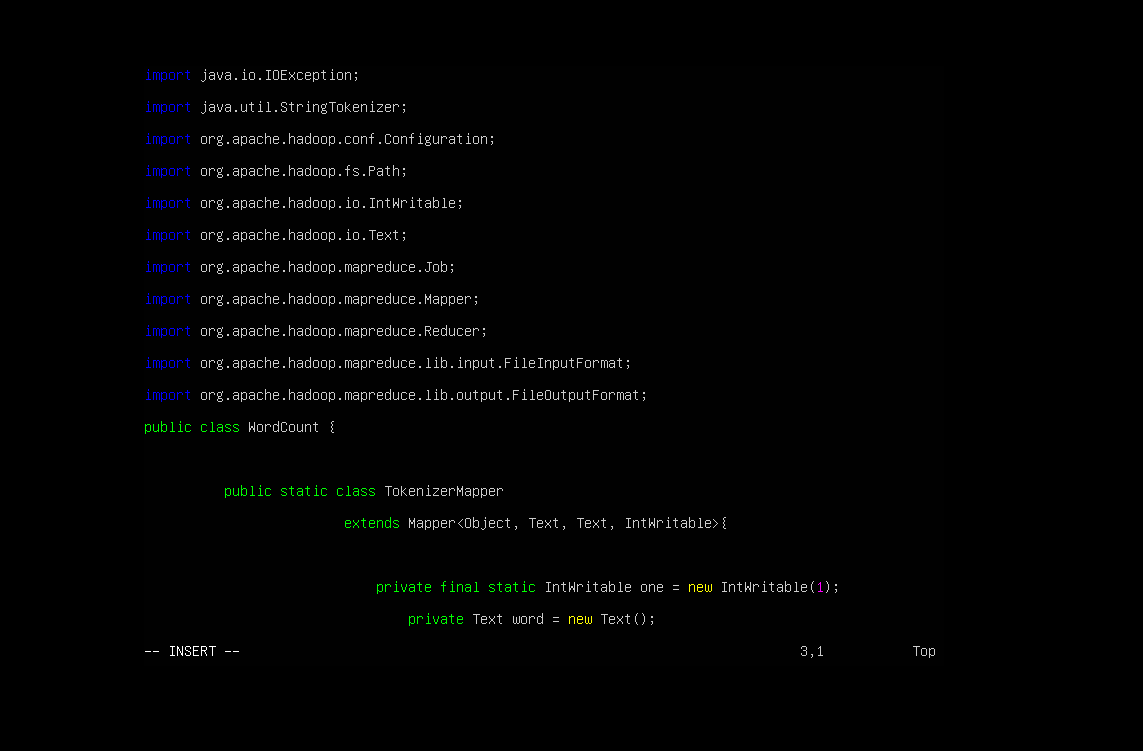


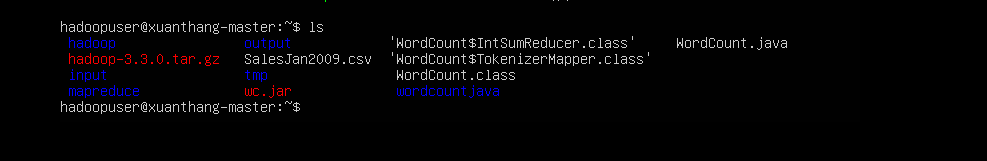
javac -version



2. Tạo file code java.

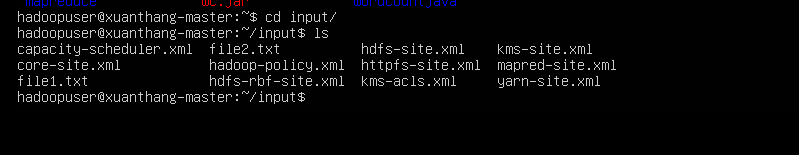
vim WordCount.java



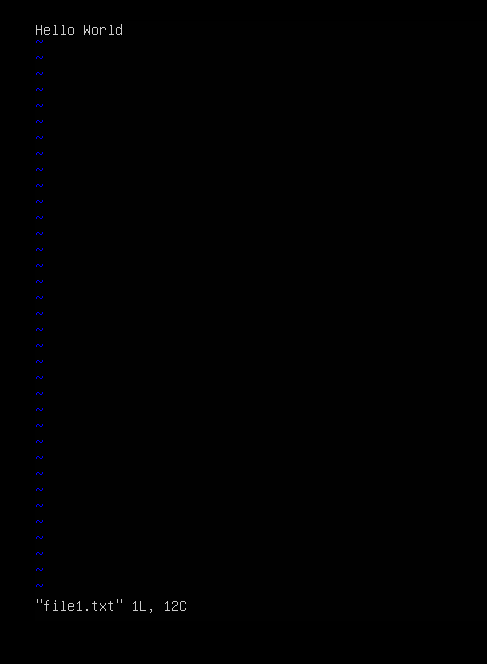


4. Tạo thư mục chứa file input và tạo file text

cd input



vim text1.txt



Vim text2.txt



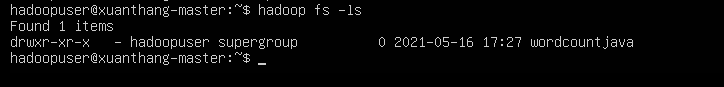
5. chạy cụm hadoop.

start-all.sh

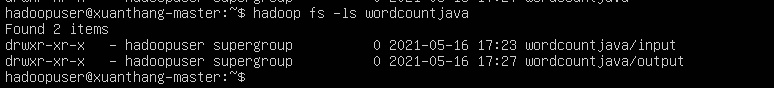


6. tạo folder chứa input trong hdfs

hadoop fs -mkdir wordcountjava

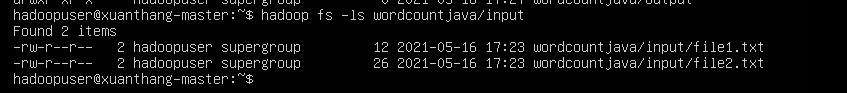


hadoop fs -mkdir wordcountjava/input



7. Đặt input file đã tạo ở ngoài đặt vào hdfs.

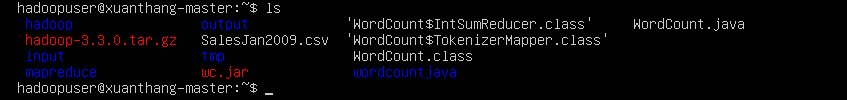
Hadoop fs -put ./input/\*.text ./wordcountjava/input



8. Biên dịch java code.

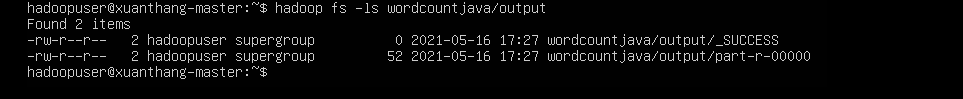
hadoop com.sum.tools.javac.Main WordCount.java => chuyen thanh file.class

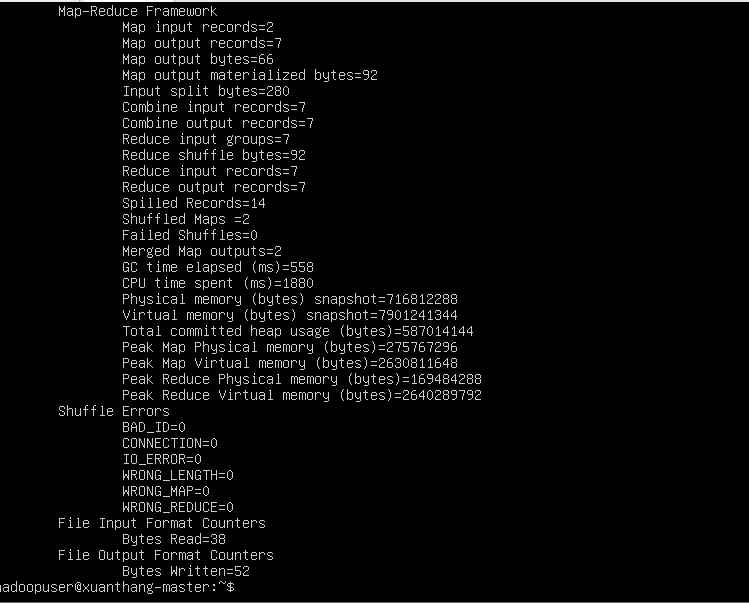
jar cf wc.jar WordCount\*.class => tao file wc.jar bo vao thu muc hien tai



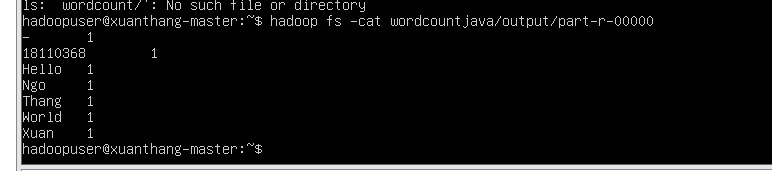
9. run ứng dụng.

hadoop jar wc.jar WordCount ./input\_wordcount/input ./wordcountjava/output





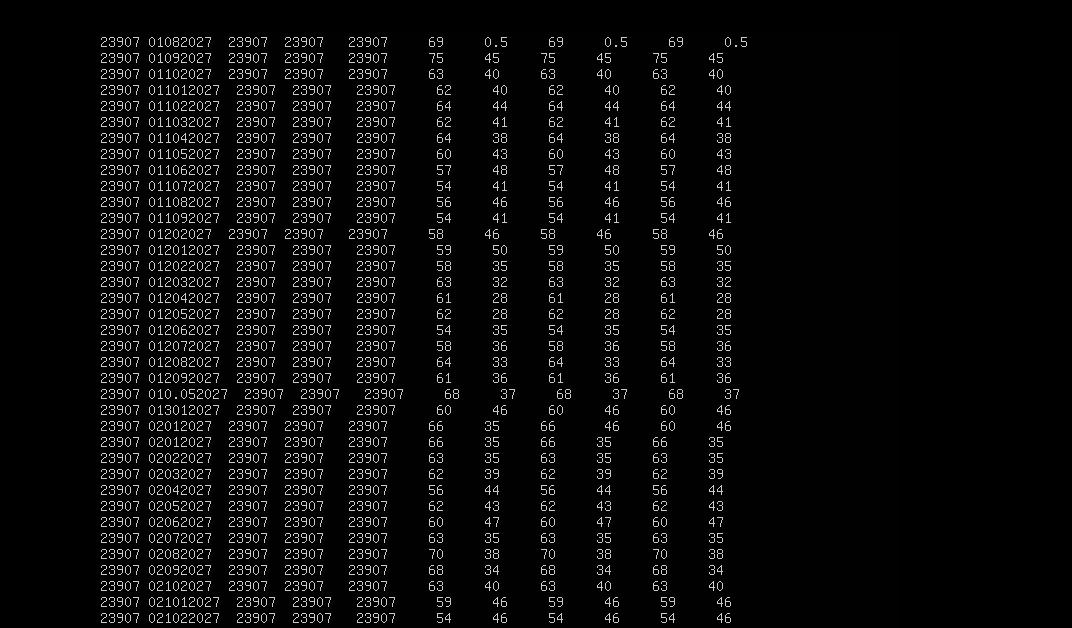
10. Xem kết quả



## Đánh giá ngày nóng hay lạnh

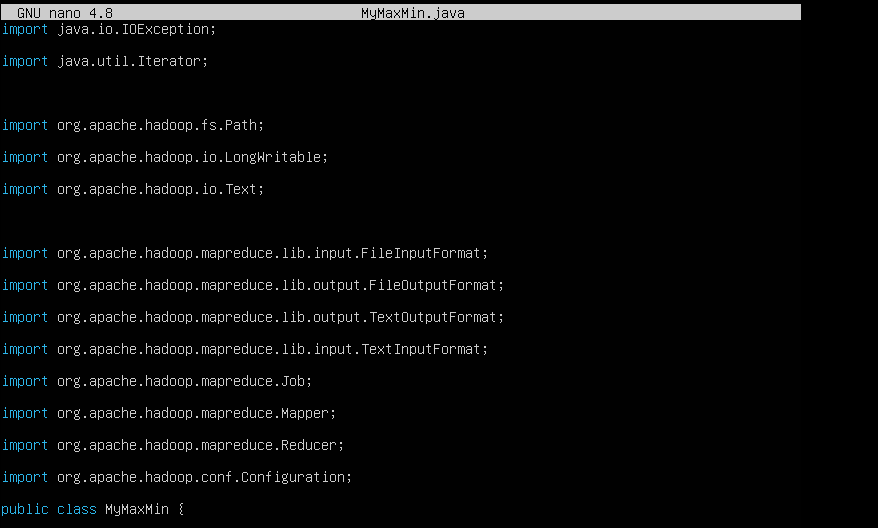
- Chuẩn bị dataset có khoảng hơn 46000 dòng

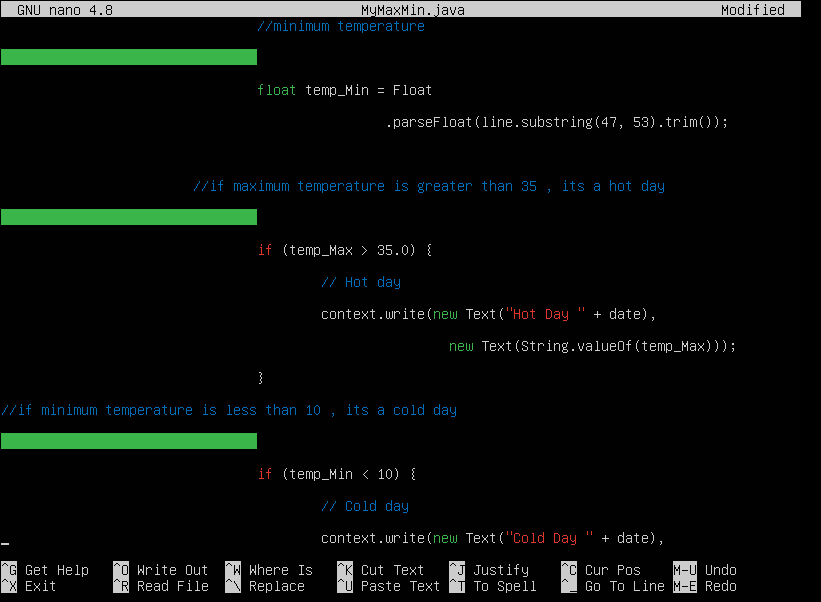




-Tạo file code java

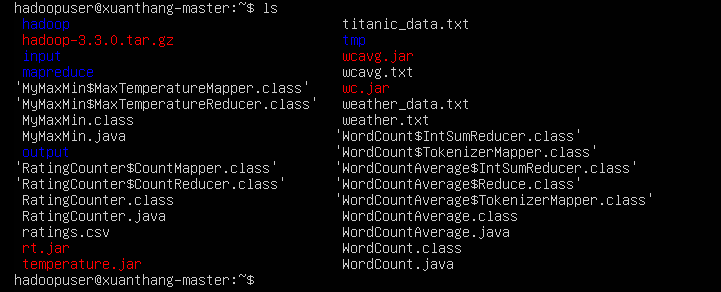






-Biên dịch file java thành class





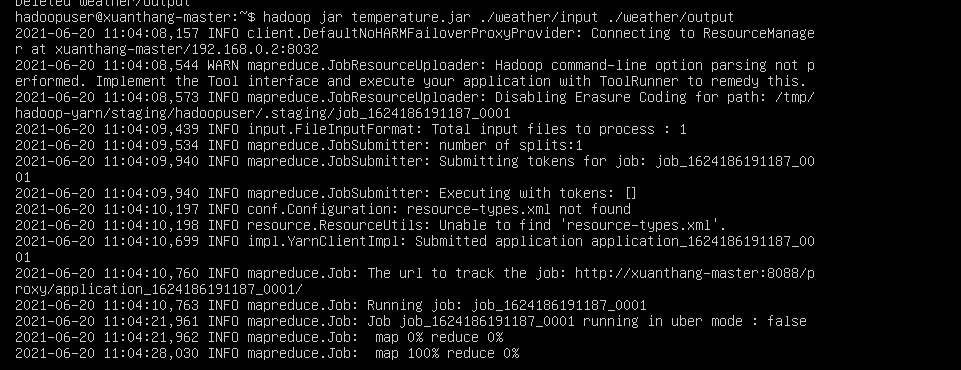
-Nén các file class thành file jar

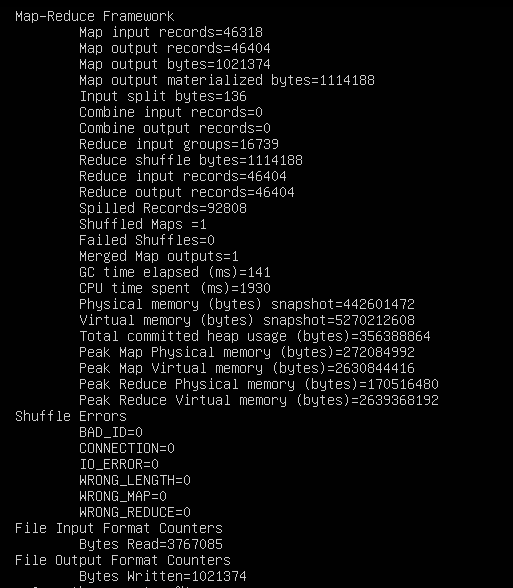




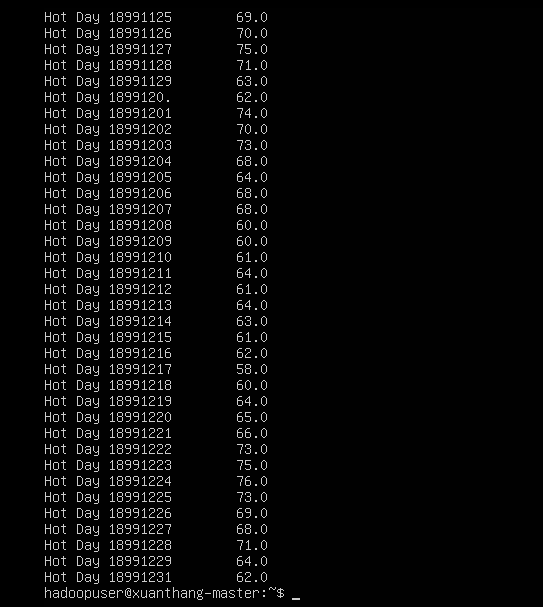
-Chạy chương trình







-Xem kết quả



## Demo3

# KẾT LUẬN

Trong quá trình thực hiện đồ án với sự cố gắn nhóm đã đạt được một số mục tiêu đề ra và hoàn thành trong thời gian qui định.

**-Kết quả đạt được:**

+ Hiểu được mô hình MapReduce.

+ Triển khái cài đặt được mô hình.

+ Thực hiện được các ví dụ demo.

-**Hạn chế:**

+ Chưa hiểu rõ về công nghệ là .

+ Đồ án chưa thể áp dụng vào thực tế.

# TÀI LIỆU THAM KHẢO

[1] <https://topdev.vn/blog/hadoop-la-gi/>

[2]<https://blog.itnavi.com.vn/mapreduce-nhung-uu-diem-va-cach-thuc-hoat-dong-cua-nen-tang-nay/>

[3] Các kiến thức và tài liệu đã học trên lớp của giảng viên Lê Thị Minh Châu