CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

1. Giới thiệu về Hadoop

- Hadoop là một framework giúp phát triển các ứng dụng phân tán
- Hadoop cung cấp một phương tiện lưu trữ dữ liệu phân tán trên nhiều node, hỗ trợ tối ưu hoá lưu lượng mạng, đó là HDFS
- Đa nền tảng

CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

1. Giới thiệu về Hadoop

Ưu điểm của Apache Hadoop

- Apache Hadoop là một framework cho phép các xử lý phân tán các tập dữ liệu lớn trên các cụm nhiều máy tính.
- Apache Hadoop thiết kế để tự nó phát hiện và xử lý lỗi ở tầng ứng dụng

CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

2. Các thành phần của Hadoop

Apache Hadoop bao gồm các thành phần:

1. **Hadoop Common**: Đây là các thư viện 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

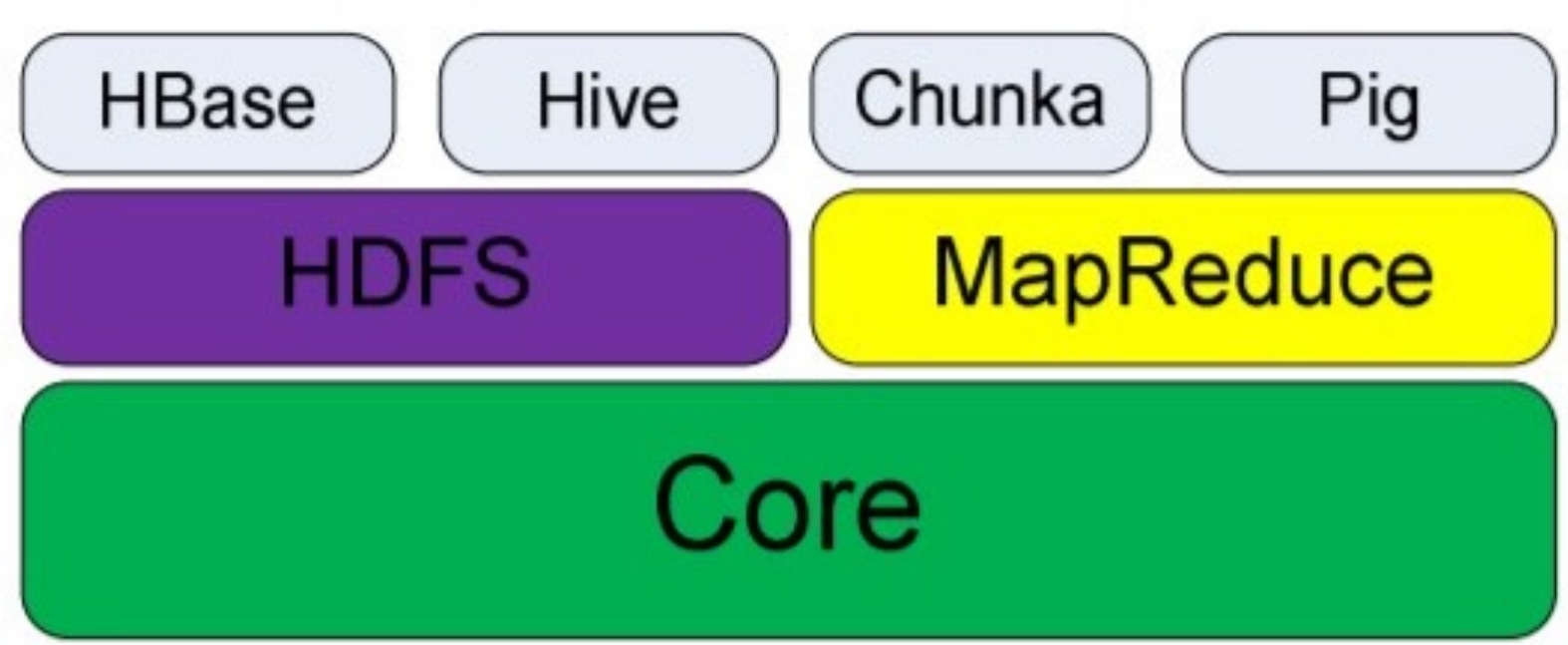
2. **Hadoop YARN:** Đây là framework để quản lý tiến trình và tài nguyên của các cluster

3. **Hadoop Distributed File System (HDFS):** là một hệ thống file phân tán cung cấp truy cập băng thông cao vào dữ liệu ứng dụng.

4. **Hadoop MapReduce:** Là một mô hình phần mềm cho việc xử lý phân tán các tập dữ liệu lớn trên ác cụm máy tính.

CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

2. Các thành phần của Hadoop



CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

2. Các thành phần của Hadoop

- Core: cung cấp các công cụ và giao diện cho hệ thống phân tán và các tiện ích I/O. Đây là phần lõi để xây dựng nên HDFS và MapReduce.

- MapReduce (MapReduce Engine): một framework giúp phát triển các ứng dụng phân tán theo mô hình MapReduce một cách dễ dàng và mạnh mẽ, ứng dụng phân tán MapReduce có thể chạy trên một cluster lớn với nhiều node.

- HDFS: hệ thống file phân tán, cung cấp khả năng lưu trữ dữ liệu khổng lồ và tính năng tối ưu hoá việc sử dụng băng thông giữa các node. HDFS có thể được sử dụng để chạy trên một cluster lớn với hàng chục ngàn node.

CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

2. Các thành phần của Hadoop

- HBase: một cơ sở dữ liệu phân tán, theo hướng cột (colunm-oriented). HBase sử dụng HDFS làm hạ tầng cho việc lưu trữ dữ liệu bên dưới, và cung cấp khả năng tính toán song song dựa trên MapReduce.

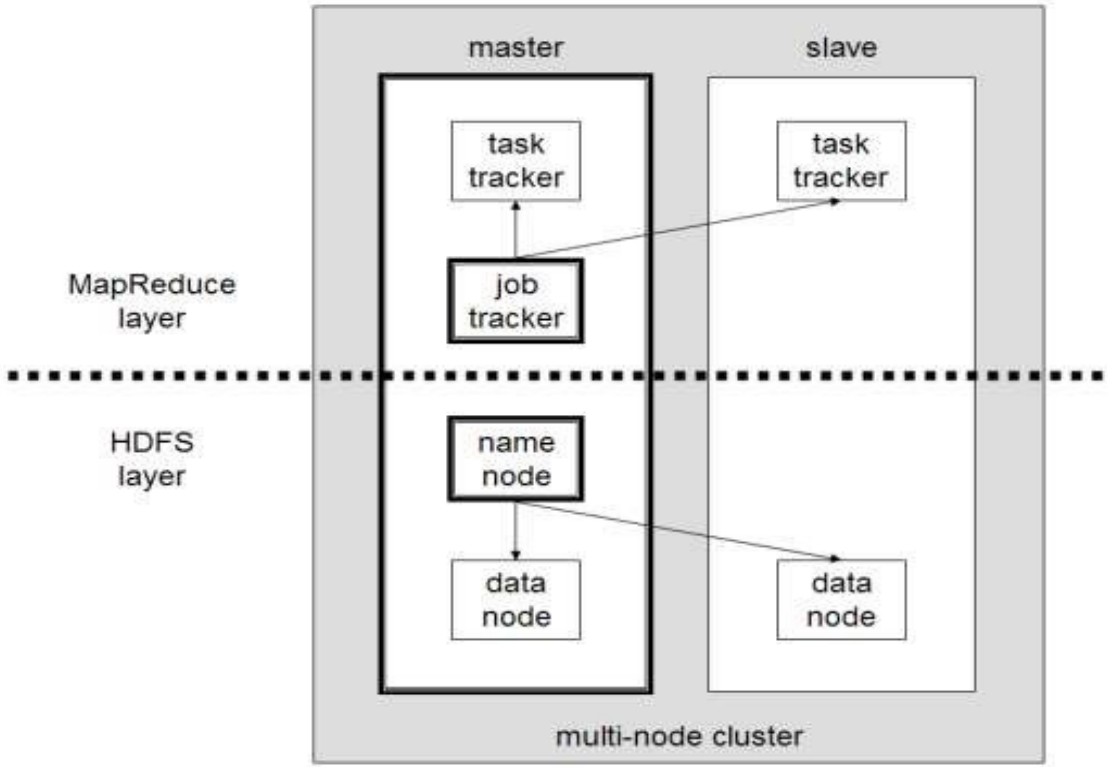
- Hive: một data warehouse phân tán. Hive quản lý dữ liệu được lưu trữ trên HDFS và cung cấp một ngôn ngữ truy vấn dựa trên SQL.

- Chukwa: một hệ thống tập hợp và phân tích dữ liệu. Chukwa chạy các collector (các chương trình tập hợp dữ liệu), các collector này lưu trữ dữ liệu trên HDFS và sử dụng MapReduce để phát sinh các báo cáo.

- Pig: ngôn ngữ luồng dữ liệu cấp cao và framework thực thi dùng cho tính toán song song.

CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

2. Các thành phần của Hadoop



Hadoop là kiến trúc master-slave, và cả hai thành phần HDFS và MapReduce đều tuân theo kiến trúc master-slave này.

CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

3. Cài đặt Hadoop cluter

- Ubuntu 18.04
- Java JDK
- Apache Hadoop

CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

4. Hệ thống tập tin phân tán

Hệ thống tệp tin hỗ trợ việc truy cập vào các tập tin từ nhiều máy tính (host) chia sẻ dữ liệu với nhau thông qua một mạng máy tính.

Hệ thống cho phép nhiều người dùng trên nhiều máy khác nhau có thể chia sẻ các tập tin và các tài nguyên lưu trữ.

Hadoop Distributed File System (HDFS)

CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

4. Hệ thống tập tin phân tán

Hadoop Distributed File System (HDFS)

Lỗi về phần cứng sẽ thường xuyên xảy ra

Kích thước file sẽ lớn hơn so với các chuẩn truyền thống

Khả năng phục hồi dữ liệu

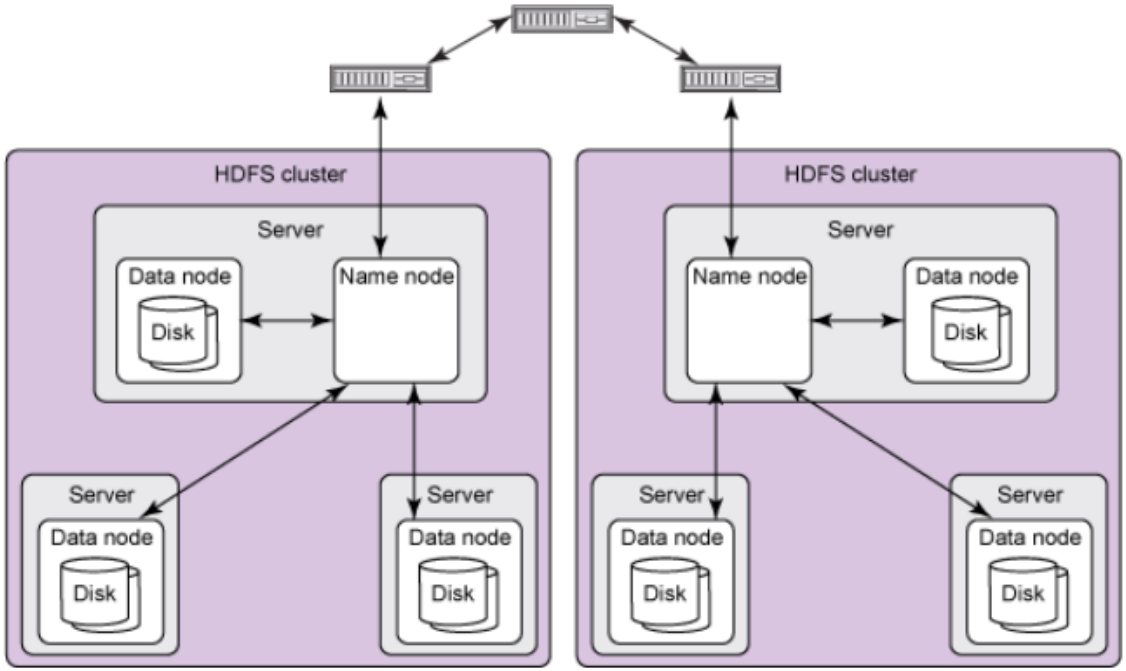
CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

4. Hệ thống tập tin phân tán

Hadoop Distributed File System (HDFS)

HDFS có một kiến trúc master/slave. Trên một cluster chạy HDFS, có hai loại node là Namenode và Datanode.

Một cluster có duy nhất một Namenode và có một hay nhiều Datanode



CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

4. Hệ thống tập tin phân tán

Hadoop Distributed File System (HDFS)

Namenode đóng vai trò là master, chịu trách nhiệm duy trì thông tin về cấu trúc cây phân cấp các file, thư mục của hệ thống file và các metadata khác của hệ thống file. Cụ thể, các Metadata mà Namenode lưu trữ gồm có:

CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

4. Hệ thống tập tin phân tán

Hadoop Distributed File System (HDFS)

Các Metadata mà Namenode lưu trữ gồm có:

- File System Namespace
- Thông tin để ánh xạ từ tên file ra thành danh sách các block
- Nơi lưu trữ các block

CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

4. Hệ thống tập tin phân tán

Hadoop Distributed File System (HDFS)

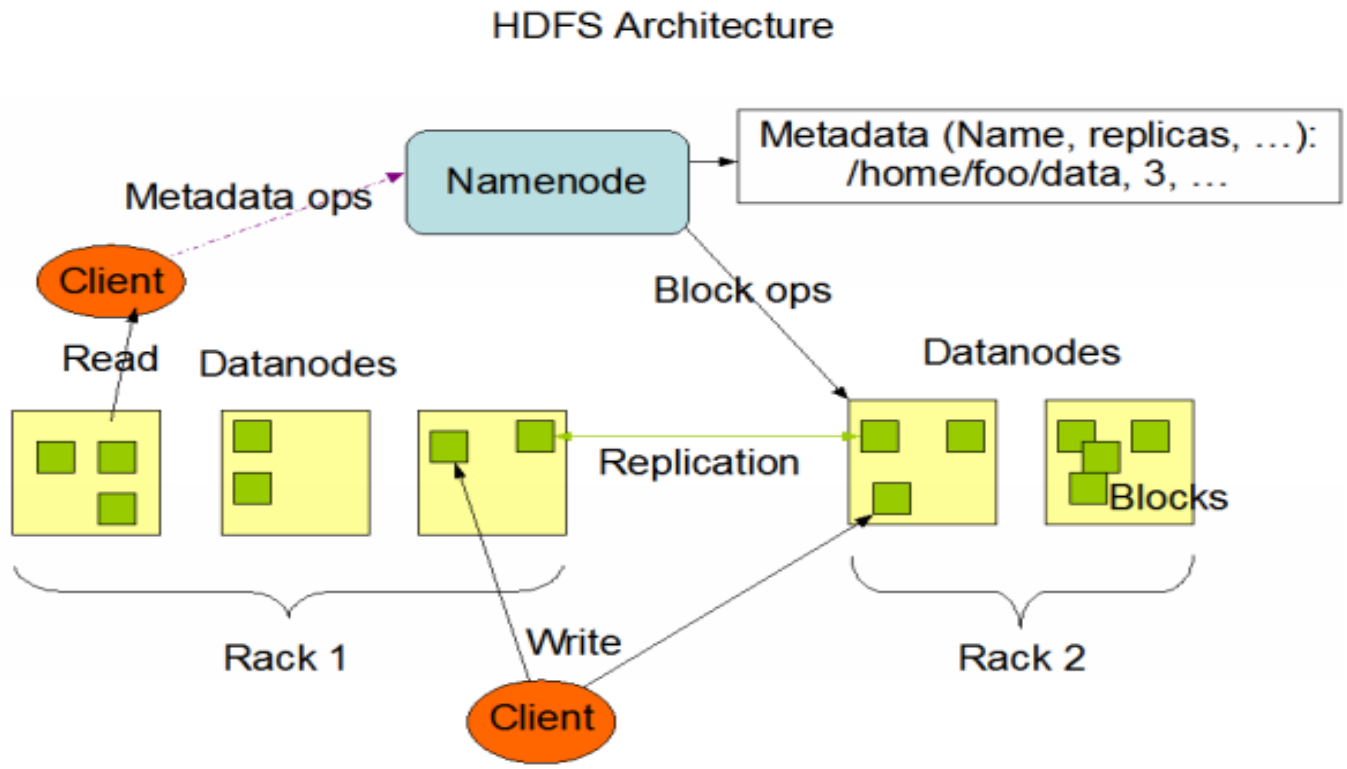
Datanode

- Sẽ chịu trách nhiệm lưu trữ các block thật sự của từng file của hệ thống file phân tán lên hệ thống file cục bộ của nó.
- Mỗi 1 block sẽ được lưu trữ như là 1 file riêng biệt trên hệ thống file cục bộ của DataNode

CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

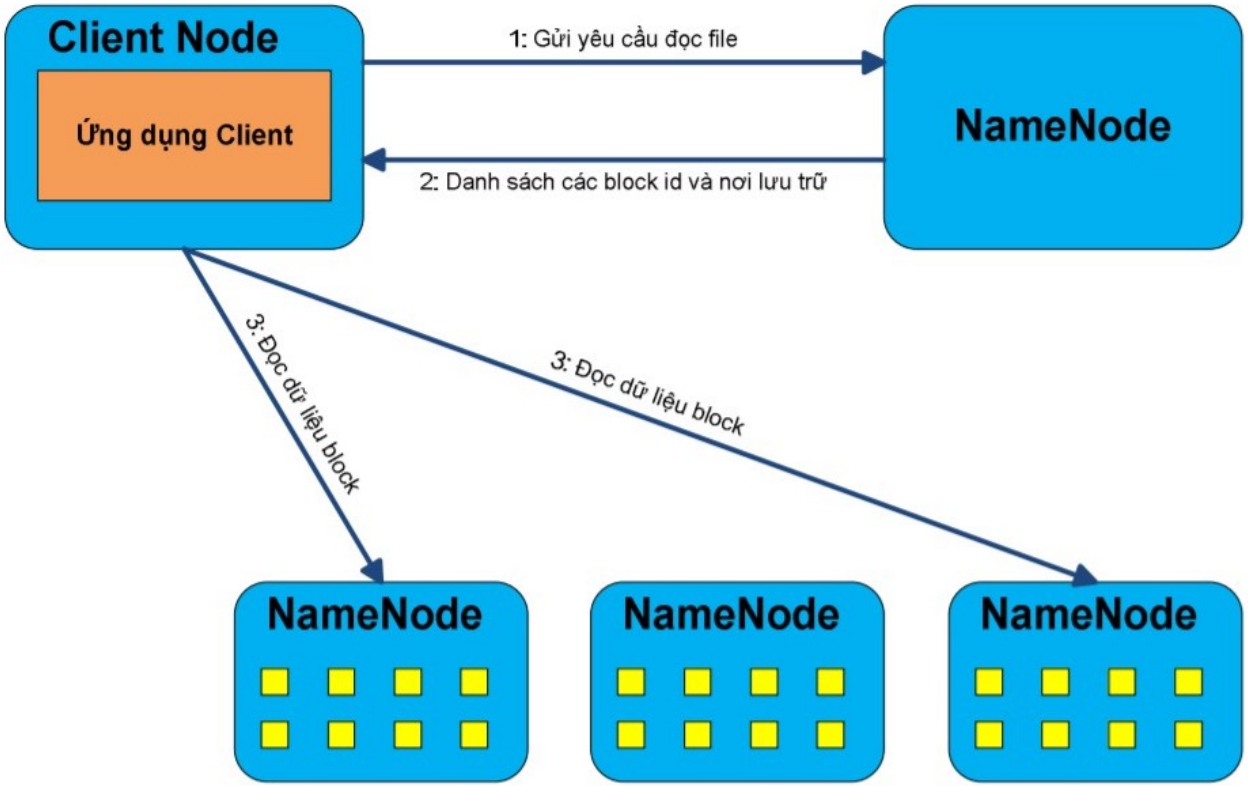
4. Hệ thống tập tin phân tán

Kiến trúc



CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

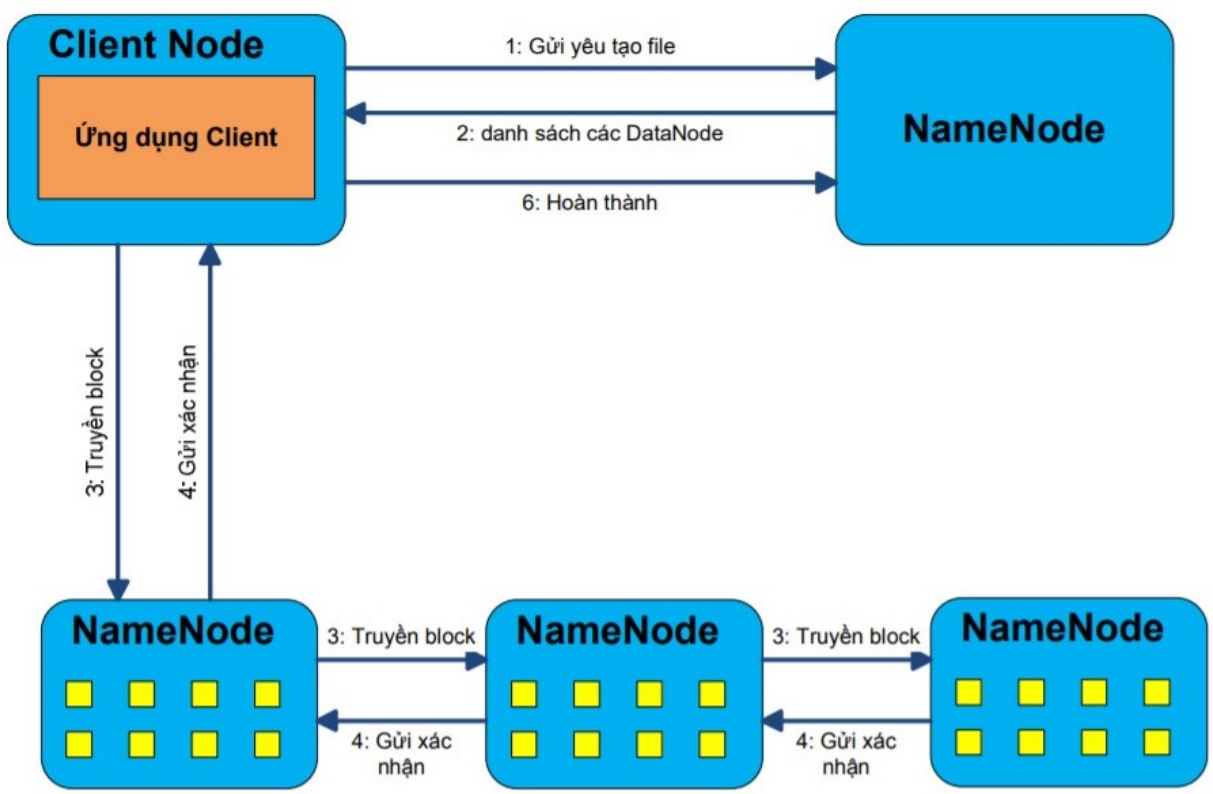
4. Hadoop Distributed File System (HDFS)



Sơ đồ sau miêu tả rõ quá trình client đọc một file trên HDFS

CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

4. Hadoop Distributed File System (HDFS)



Quá trình tạo và ghi dữ liệu lên file trên HDFS

CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

4. Hadoop Distributed File System (HDFS)

Block size

Block size của đĩa cứng sẽ quy định lượng dữ liệu nhỏ nhất mà ta có thể đọc/ghi lên đĩa.

HDFS cũng chia file ra thành các block, mỗi block này sẽ được lưu trữ trên

Datanode thành một file riêng biệt trên hệ thống file local của nó. Đây cũng sẽ là đơn vị trao đổi dữ liệu nhỏ nhất giữa HDFS và client của nó.

Block size mặc định của HDFS là **64MB**

CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

4. Hadoop Distributed File System (HDFS)

Vị trí lưu các block

NameNode sẽ không lưu trữ bền vững thông tin về nơi lưu trữ các bản sao của các block.

Nó chỉ hỏi các DataNode các thông tin đó lúc DataNode khởi động. NameNode sẽ giữ cho thông tin nơi lưu trữ các block đươc cập nhật.

- NameNode điều khiển tất cả các thao tác sắp đặt các bản sao của các block lên các.
- DataNode và quản lý tình trạng các DataNode bằng thông điệp HearBeat

CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

4. Hadoop Distributed File System (HDFS)

Nhật ký thao tác

EditLog chứa tất cả nhật ký các thao tác làm thay đổi tình trạng của metadata.

EditLog được lưu trữ như một file trên hệ thống file cục bộ của NameNode.

EditLog sẽ được dùng trong quá trình phục hồi hệ thống với SecondaryNameNode

CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

4. Hadoop Distributed File System (HDFS)

Khả năng chịu lỗi và chẩn đoán lỗi của HDFS

Khả năng phục hồi nhanh chóng giữa Namenode và Datanode.

Nhân bản các block: Người dùng (hoặc ứng dụng) có thể gán các chỉ số mức độ nhân bản (replication level).

Toàn vẹn dữ liệu trên HDFS: HDSF đảm bảo tính toàn vẹn của dữ liệu bằng cách thực hiện tạo checksum5 tất cả dữ liệu ghi lên nó và sẽ kiểm tra lại checksum mỗi khi đọc dữ liệu

CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

4. Hadoop Distributed File System (HDFS)

Các giao diện tương tác

Giao diện command line: **hdfs://<namenode>/<path>**

Giao diện java (API)

Giao diện web: **http://<namenode>:50070/**

CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

4. Hadoop Distributed File System (HDFS)

Quản trị HDFS

Permission: HDFS có một mô hình phân quyền tập tin. Có ba loại quyền truy cập: quyền được phép đọc (r), quyền ghi (w), và quyền thực thi (x).

Khi truy cập vào HDFS, client được nhận diện người dùng (username) và nhóm (group) của tiến trình trên client.

Trên HDFS còn có một người dùng đặc biệt, đó là super-user. Đây chính là user đại diện cho các tiến trình trên NameNode. User này có quyền hạn toàn cục và sẽ không bị kiểm tra quyền truy cập

CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

4. Hadoop Distributed File System (HDFS)

Quản trị HDFS

Quản lý hạn ngạch (quotas): HDFS cho phép người quản trị có thể thiết lập hạn ngạch (quotas) cho số lượng tên (file/thư mục) sử dụng và dung lượng sử dụng cho các thư mục

CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

4. HDFS

Tạo thư mục trong HDFS

```
hdfs dfs -mkdir <path>
```

Ví dụ:

```
hdfs dfs -mkdir test
```

CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

4. HDFS

Xem nội dung thư mục trong HDFS

```
hdfs dfs –ls <path>
```

Ví dụ:

```
hdfs dfs -ls /test
hdfs dfs -ls /user/test
```

CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

4. HDFS

Copy một hoặc nhiều file vào HDFS

```
hdfs dfs -put <local_path> … <HDFS_path>
```

Ví dụ:

```
hdfs dfs -put test.txt test
```

CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

4. HDFS

Download file trong HDFS về local

```
hdfs dfs -get <HDFS_path> <local_path>
```

Ví dụ:

```
hdfs dfs -get test/test.txt /root/lediep
```

CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

4. HDFS

Xóa file/thư mục trong HDFS

```
hdfs dfs -rm -r -f <args>
```

Ví dụ:

```
hdfs dfs -rm -r -f test
```

CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

5. Map Reduce

Mapreduce là một mô hình lập trình, thực hiện quá tình xử lý tập dữ liệu lớn. Mapreduce gồm 2 pha : map và reduce.

Hàm Map: Các xử lý một cặp (key, value) để sinh ra một cặp (key, value) - key và value trung gian. Dữ liệu này input vào hàm Reduce.

Hàm Reduce: Tiếp nhận các (key, value) và trộn các cặp (key, value) trung gian , lấy ra các valuel có cùng key.

CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

5. Map Reduce

Ý tưởng

Chia vấn đề cần xử lý thành các phần nhỏ để xử lý.

Xử lý các phần nhỏ đó một cách song song và độc lập trên các máy tính phân tán.

Tổng hợp các kết quả thu được để dưa ra kết quả cuối cùng.

CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

5. Map Reduce

MapReduce được tóm tắt như sau:

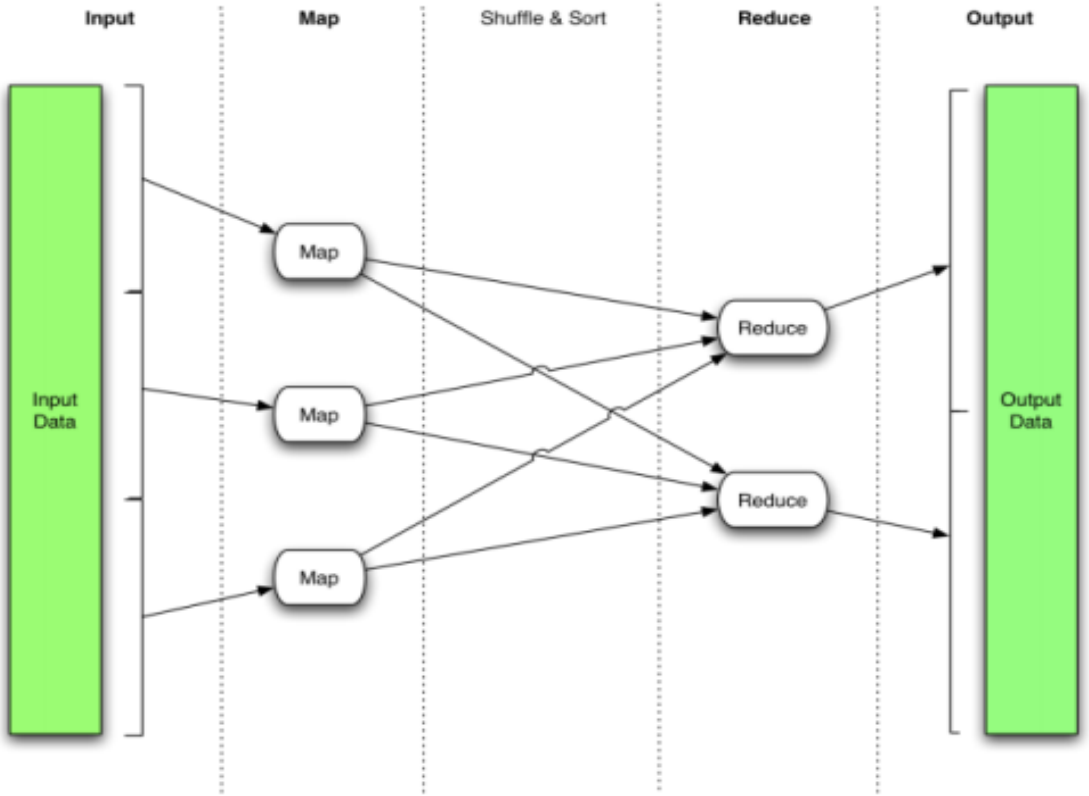
Đọc dữ liệu đầu vào

Xử lý dữ liệu đầu vào (thực hiện hàm map)

Sắp xếp và trộn các kết quả thu được từ các máy tính phân tán thích hợp nhất.

Tổng hợp các kết quả trung gian thu được (thực hiện hàm reduce)

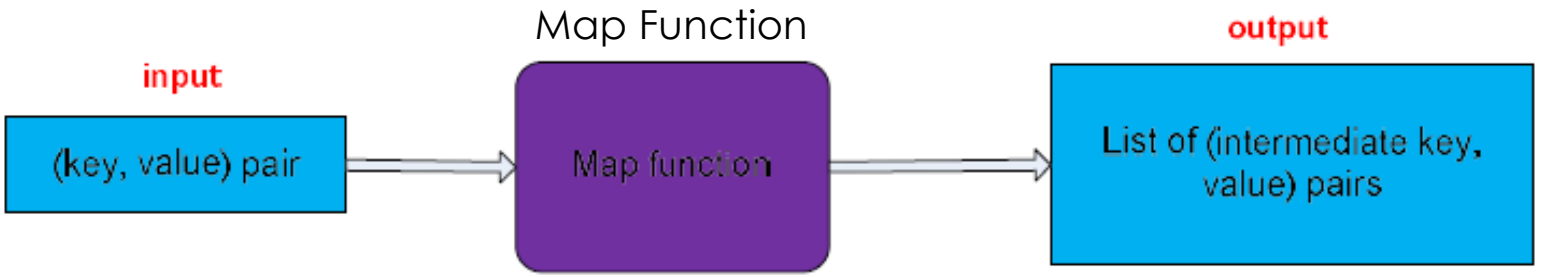
Đưa ra kết quả cuối cùng.



CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

5. Map Reduce

Hàm Map



Mỗi phần tử của dữ liệu đầu vào sẽ được truyền cho hàm Map dưới dạng cặp <key,value>

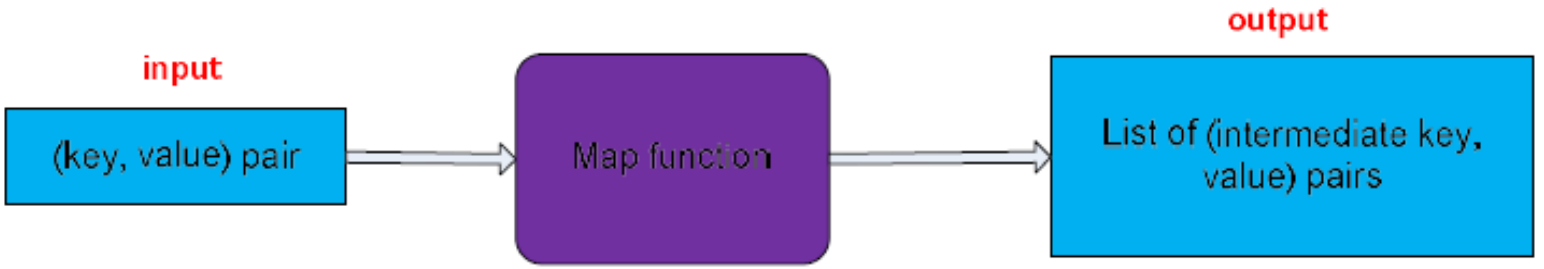
Hàm Map xuất ra một hoặc nhiều cặp <key,value>

Sau quá trình Map, các giá trị trung gian được tập hợp thành các danh sách theo từng khóa

CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

5. Map Reduce

Hàm Reduce



Kết hợp, xử lý, biến đổi các value: Hệ thống sẽ gom nhóm tất cả value key từ các output của hàm map, để tạo thành tập các cặp dự liệu với cấu trúc là (key, tập các value cùng key). Dữ liệu input của hàm reduce là từng cặp dữ liệu được gom nhóm ở trên.

Đầu ra là một cặp <key,value> đã được xử lý

CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

5. Map Reduce

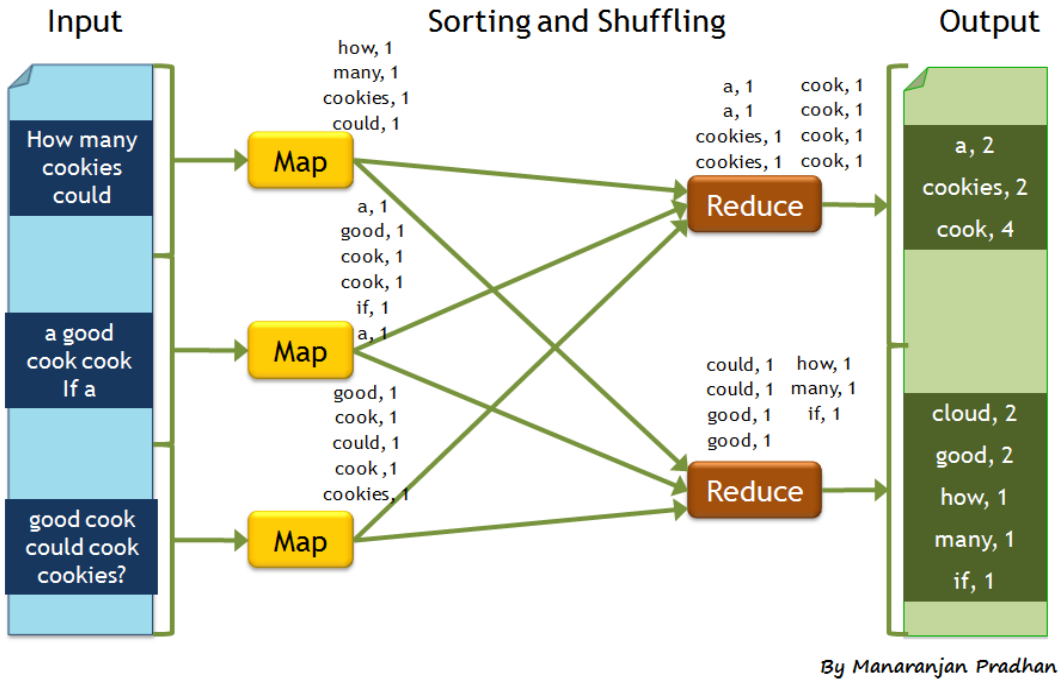
Ví dụ bài toán đếm từ (WordCount)

Mapper

+ Đầu vào : Một dòng của văn bản
+ Đầu ra : key : từ, value : 1

Reducer

+ Đầu vào : key: từ, values: tập hợp các giá trị đếm được của mỗi từ
+ Đầu ra : key: từ, value: tổng



CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

5. Map Reduce

Tính toán song song

Hàm Map chạy song song tạo ra các giá trị trung gian khác nhau từ các tập dữ liệu khác nhau

Hàm Reduce cũng chạy song song, mỗi reducer xử lý một tập khóa khác nhau

Tất cả các giá trị được xử lý độc lập

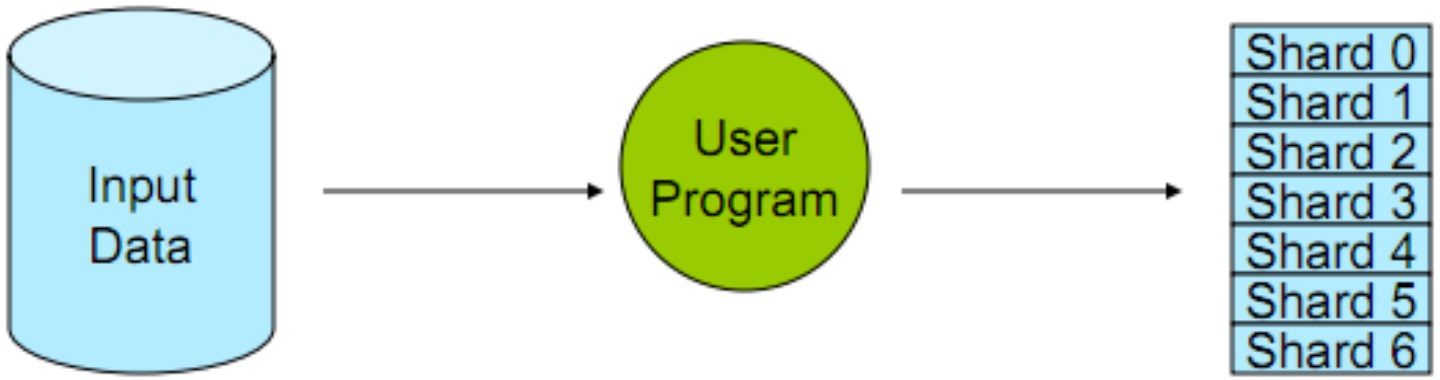
Bottleneck: Giai đoạn Reduce chỉ bắt đầu khi giai đoạn Map kết thúc

CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

5. Map Reduce

Thực thi

Chương trình (user program), thông qua thư viện MapReduce phân mảnh dữ liệu đầu vào

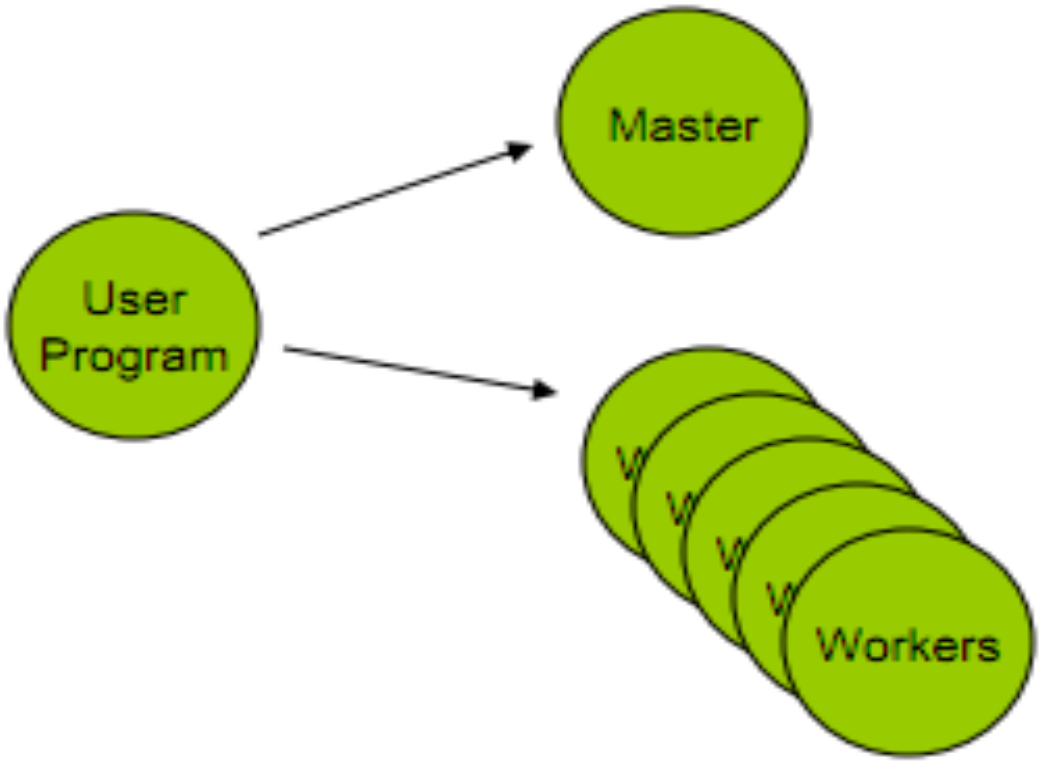


CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

5. Map Reduce

Thực thi

Map Reduce sao chép chương trình này vào các máy cluster (master và các worker)



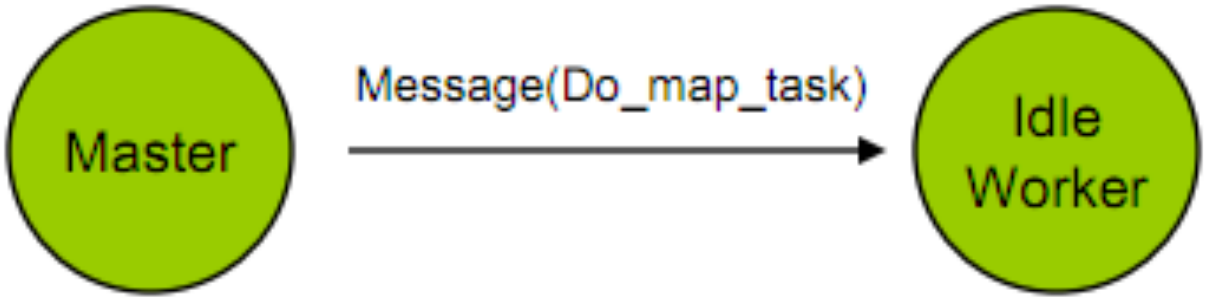
CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

5. Map Reduce

Thực thi

Master phân phối M tác vụ Map và R tác vụ Reduce vào các worker rảnh rỗi

Master phân phối các tác vụ dựa trên vị trí của dữ liệu



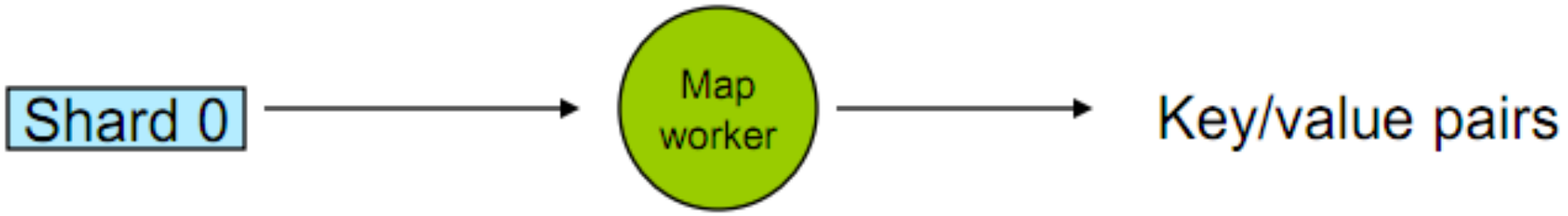
CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

5. Map Reduce

Thực thi

Mỗi map-task worker đọc dữ liệu từ phân vùng dữ liệu được gán cho nó và xuất ra những cặp <key,value> trung gian

Dữ liệu này được ghi tạm trên RAM

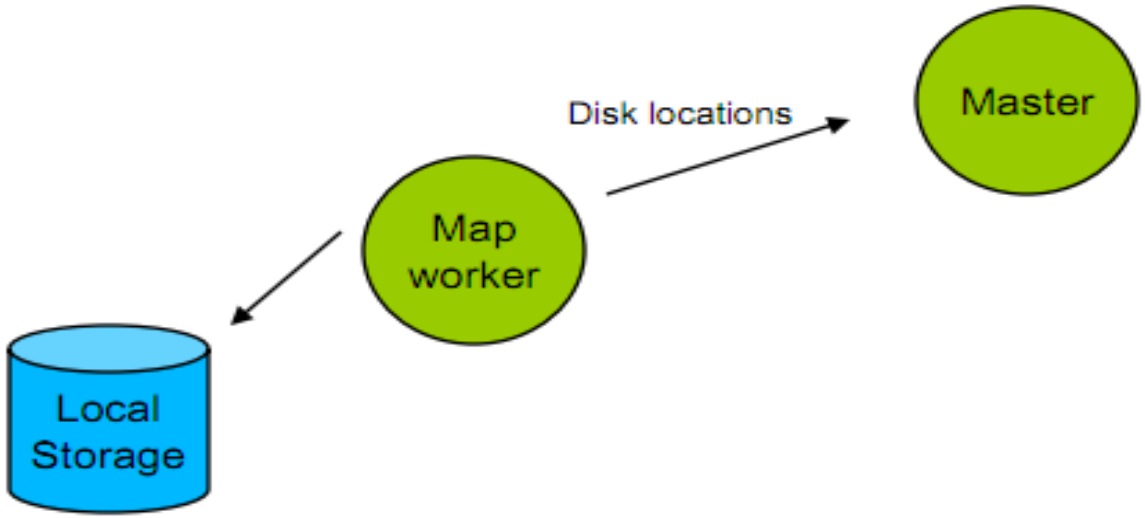


CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

5. Map Reduce

Thực thi

Mỗi worker phân chia dữ liệu trung gian thành R vùng, lưu xuống đĩa, xóa dữ liệu trên bộ đệm và thông báo cho Master

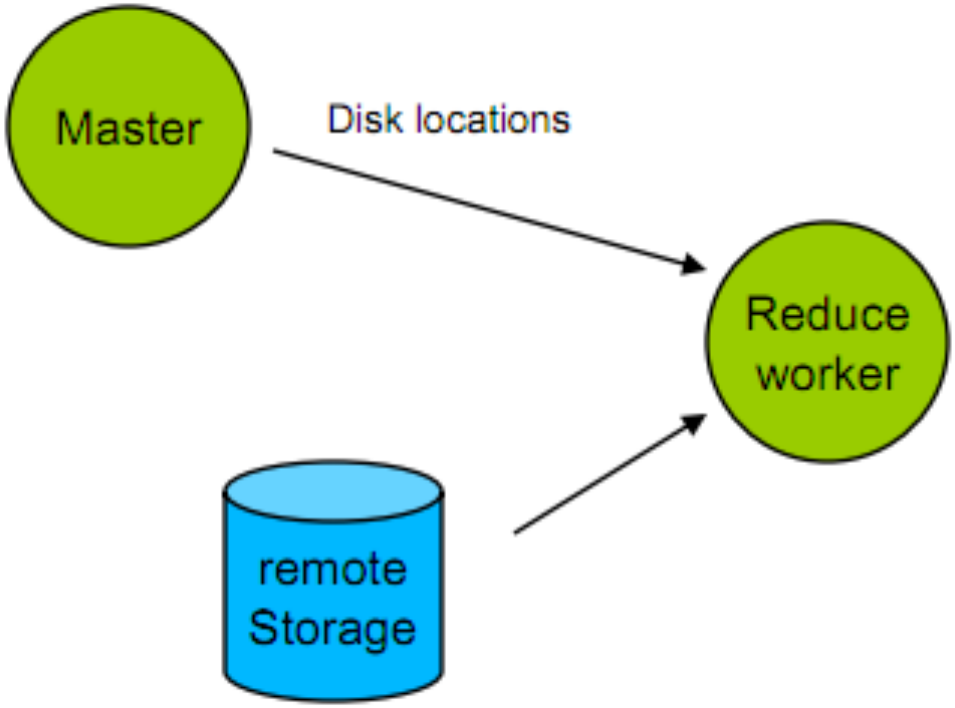


CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

5. Map Reduce

Thực thi

Master gán các dữ liệu trung gian và chỉ ra vị trí của dữ liệu cho các reduce-task

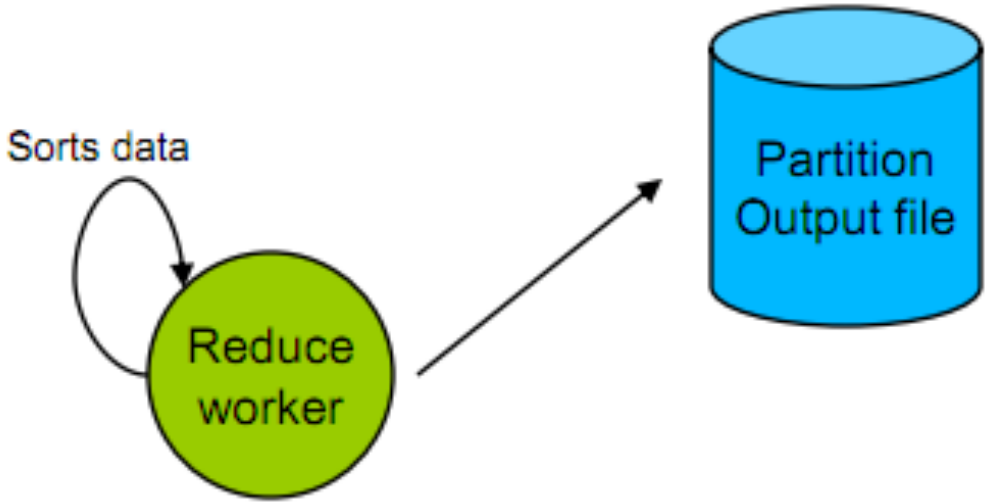


CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

5. Map Reduce

Thực thi

Mỗi reduce-task worker sắp xếp các key, gọi hàm reduce và xuất kết quả đầu ra



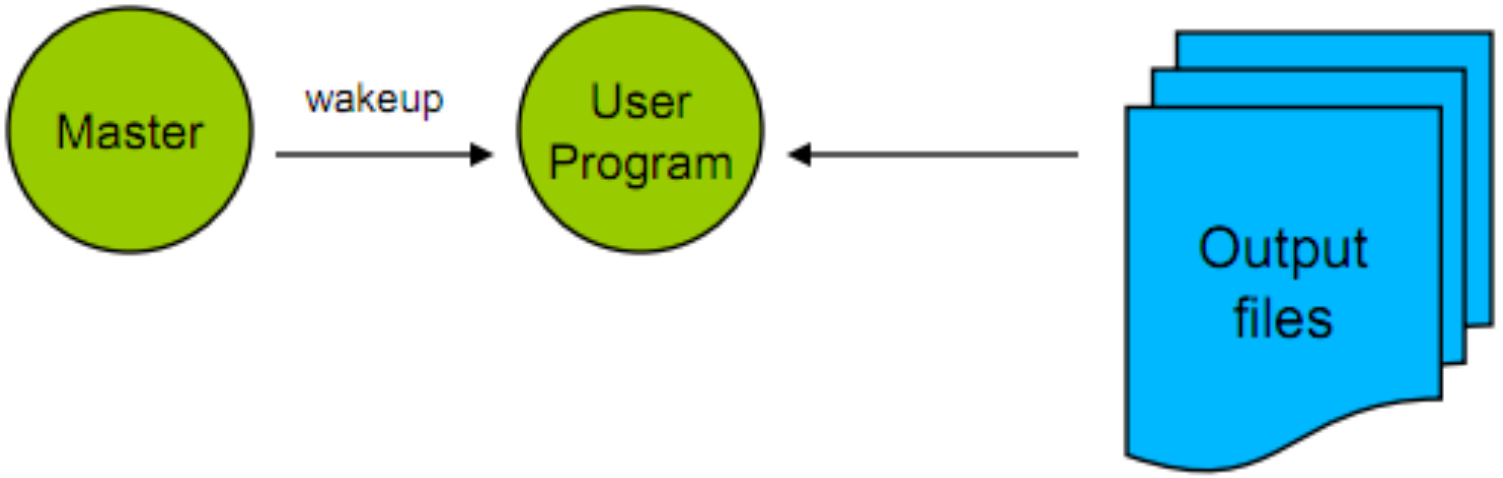
CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

5. Map Reduce

Thực thi

Master kích hoạt (wakes up) chương trình của người dùng thông báo kết quả hoàn thành

Dữ liệu đầu ra được lưu trong R tập tin



CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

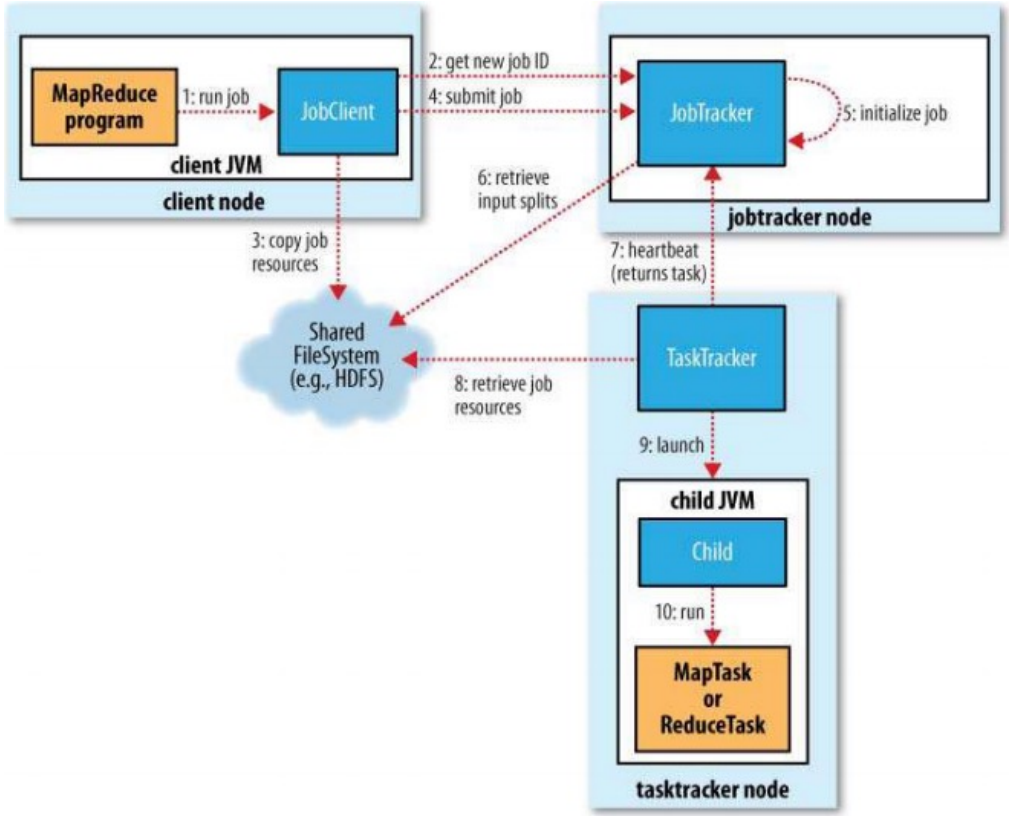
5. Map Reduce

Hadoop Map Redude (master/slave)

Client gửi MapReduce Job

JobTracker điều phối việc thực thi Job

TaskTracker thực thi các task đã được chia ra

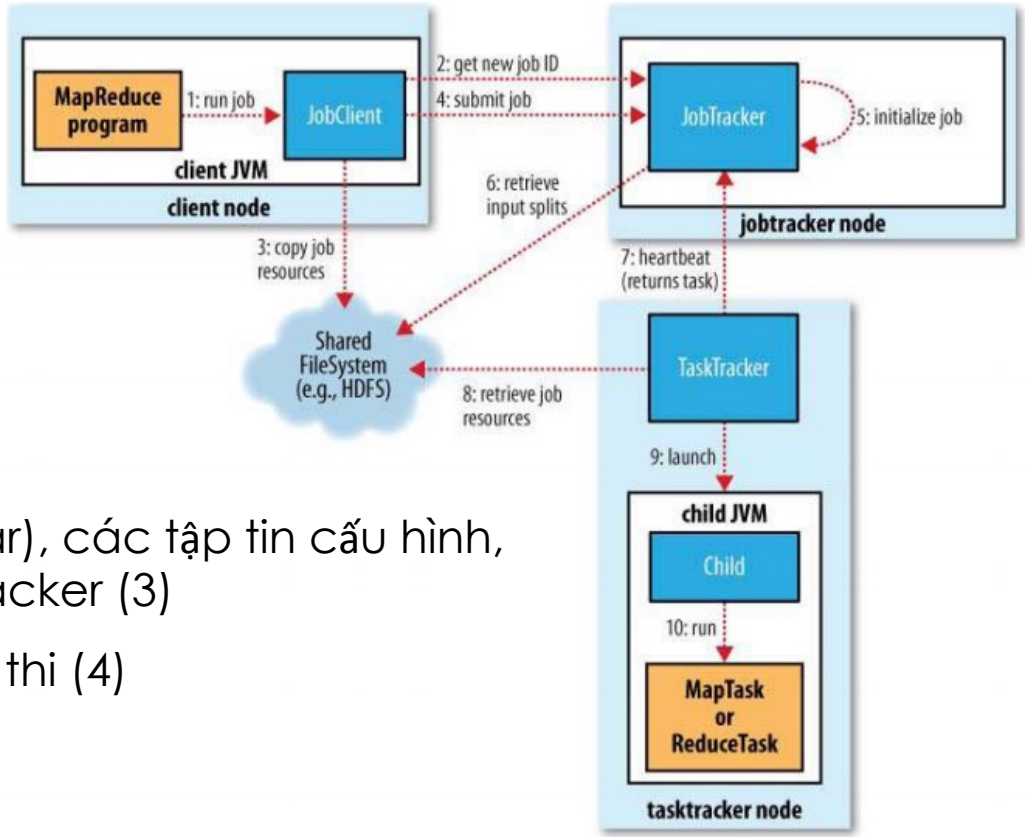


CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

5. Map Reduce

Hadoop Map Redude (master/slave)

Job Submission



Yêu cầu ID cho job mới (1)

Kiểm tra các thư mục đầu vào và đầu ra

Chia tách dữ liệu đầu vào

Chép các tài nguyên bao gồi chương trình (Jar), các tập tin cấu hình, các mảnh dữ liệu đầu vào filesystem của jobtracker (3)

Thông báo với jobtracker job sẵn sàng để thực thi (4)

CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

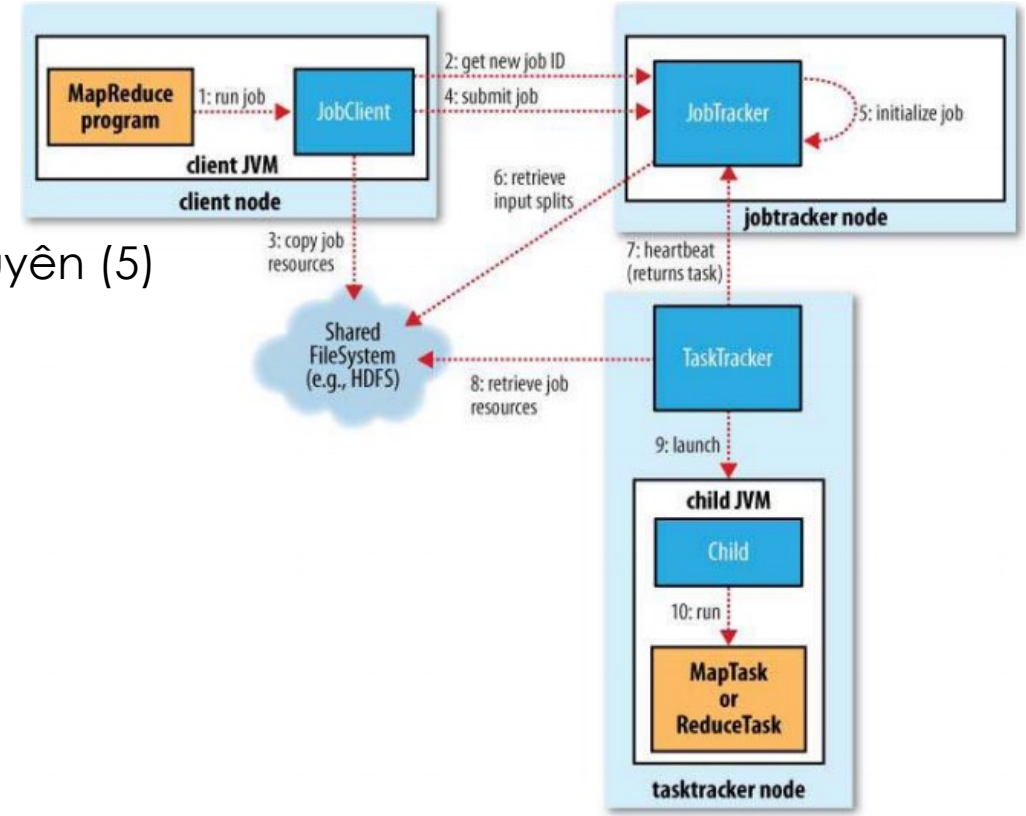
5. Map Reduce

Hadoop Map Redude (master/slave)

Khởi tạo JOB

Thêm job vào hàng đợi & khơi tạo các tài nguyên (5)

Tạo danh sách các tác vụ (task) (6)



CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

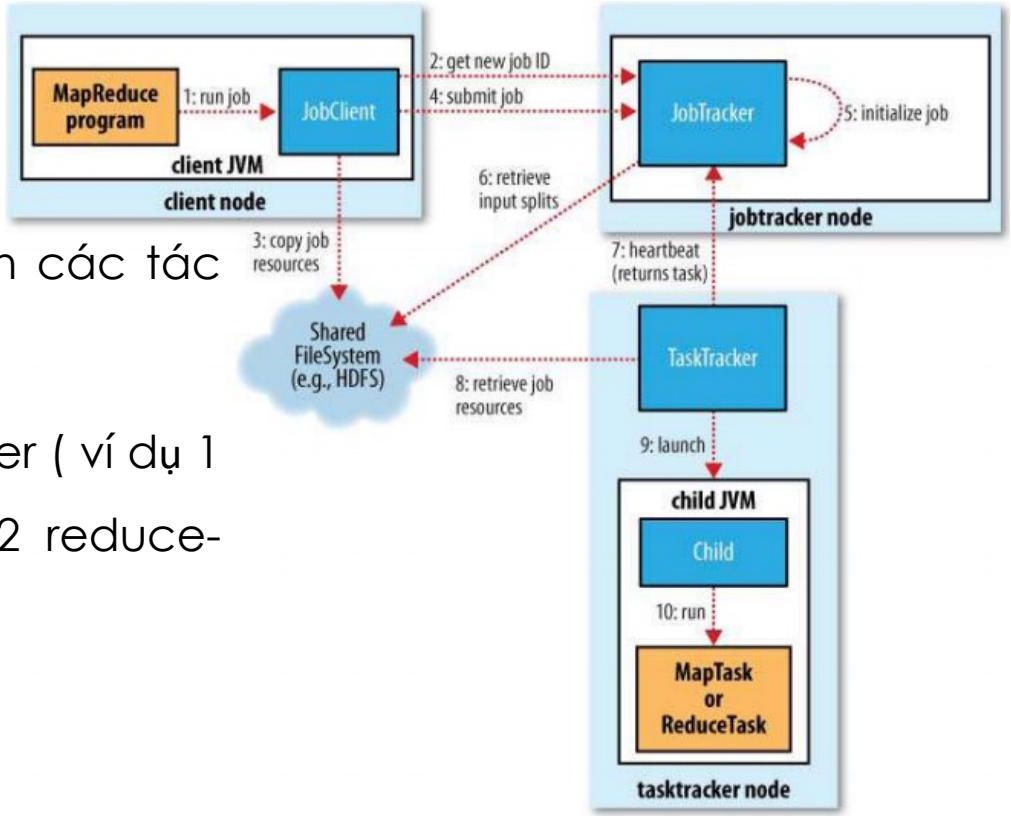
5. Map Reduce

Hadoop Map Redude (master/slave)

Phân phối các tác vụ

TaskTracker định kỳ thông báo sẵn sàng nhận các tác vụ mới (7)

JobTracker giao tác vụ cố định cho TaskTracker (ví dụ 1 TaskTracker chạy đồng thời 2 map-task và 2 reduce-task)



CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

5. Map Reduce

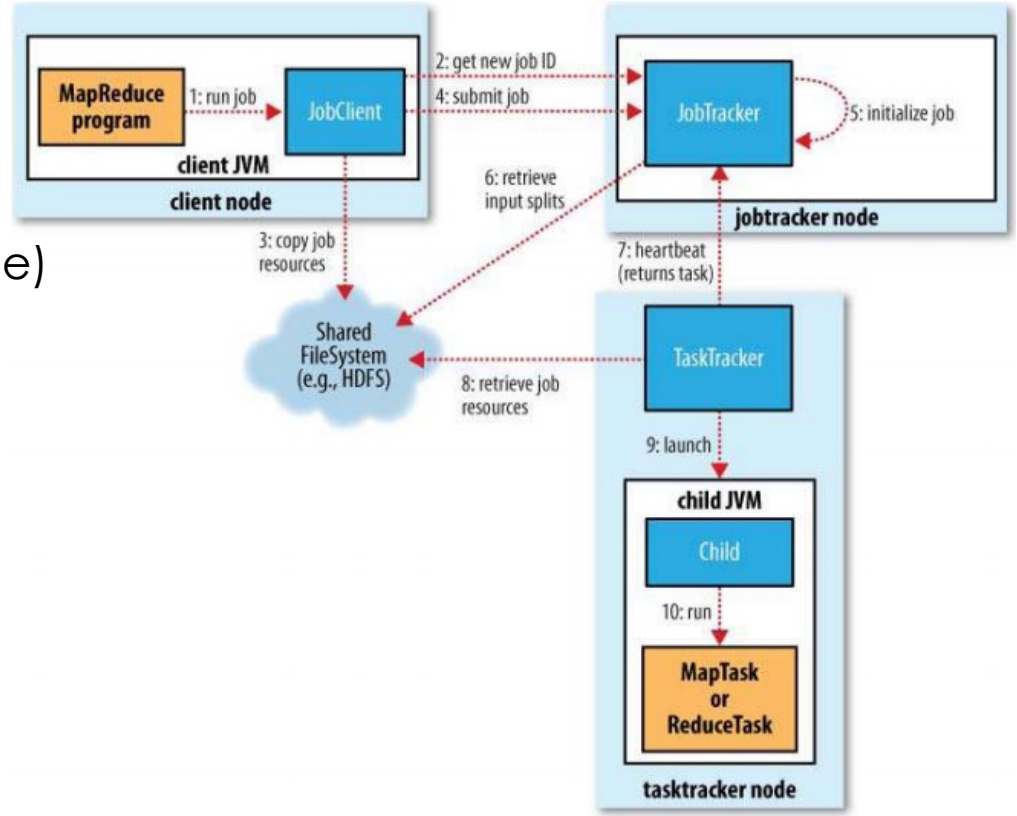
Hadoop Map Redude (master/slave)

Thực thi tác vụ

TaskTracker Chép chương trình thực thi (Jar File)

và các dữ liệu cần thiết từ hệ thống chia sẻ file

Tạo tiến trình TaskRunner để thực thi tác vụ



CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

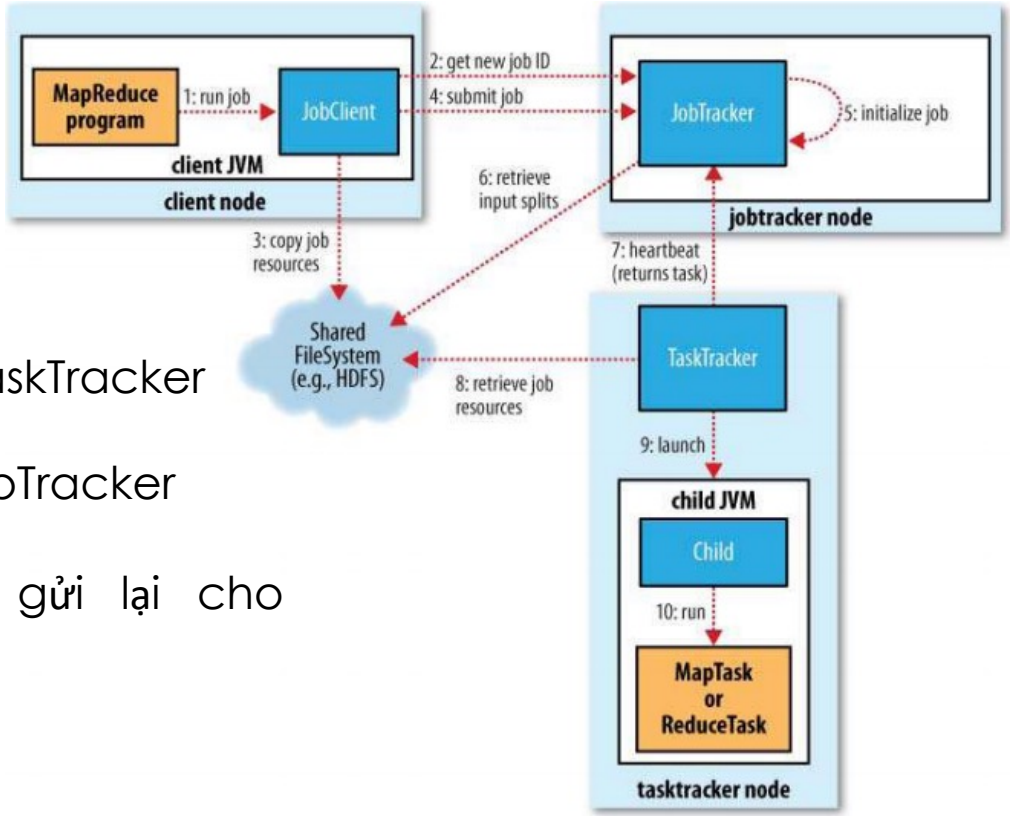
5. Map Reduce

Hadoop Map Redude (master/slave)

Cập nhật trạng thái

Cập nhật trạng thái trong quá trìng thực thi

- Task process gửi báo cáo 3s một lần cho TaskTracker
- TaskTracker gửi báo cáo 5s một lần cho JobTracker
- JobTracker tổng hợp các báo cáo, gửi lại cho JobClient mỗi giây một lần



CHƯƠNG 2. HADOOP, HDFS VÀ MAPREDUCE

5. Map Reduce

Hadoop Map Redude (master/slave)

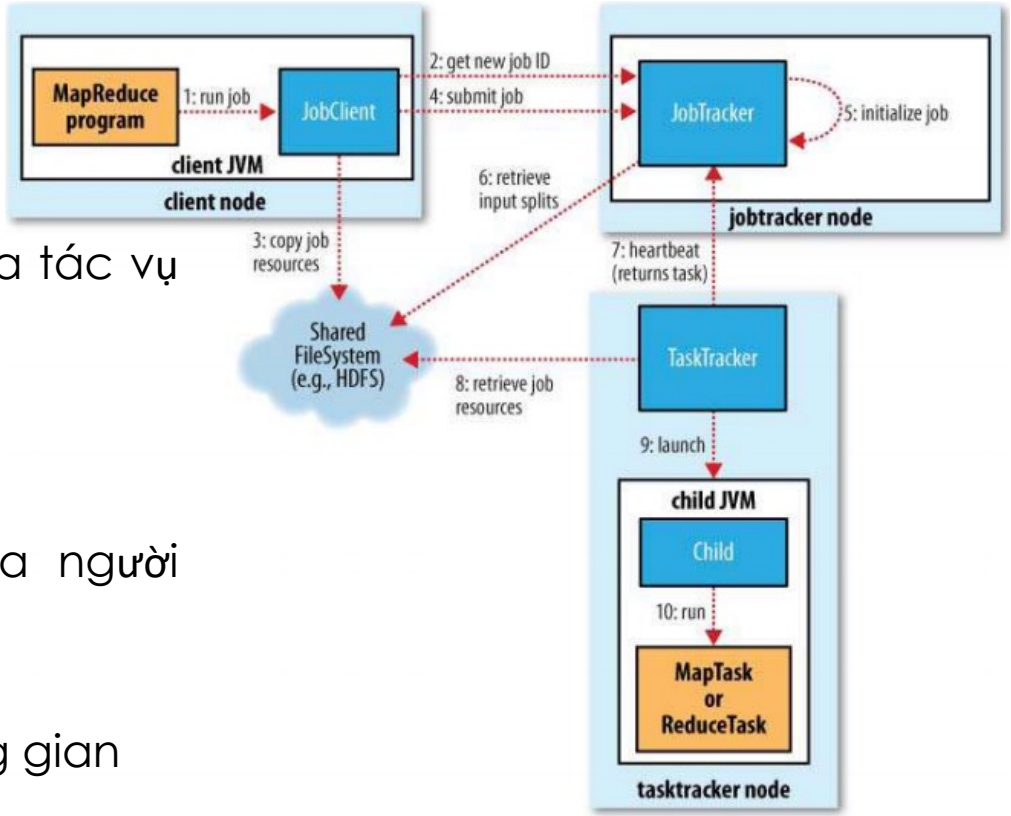
Kết thúc Job

Khi JobTracker nhận được tín hiệu kết thúc của tác vụ cuối cùng

JobTracker gửi tín hiệu success cho JobClient

JobClient thông báo cho chương trình của người dùng

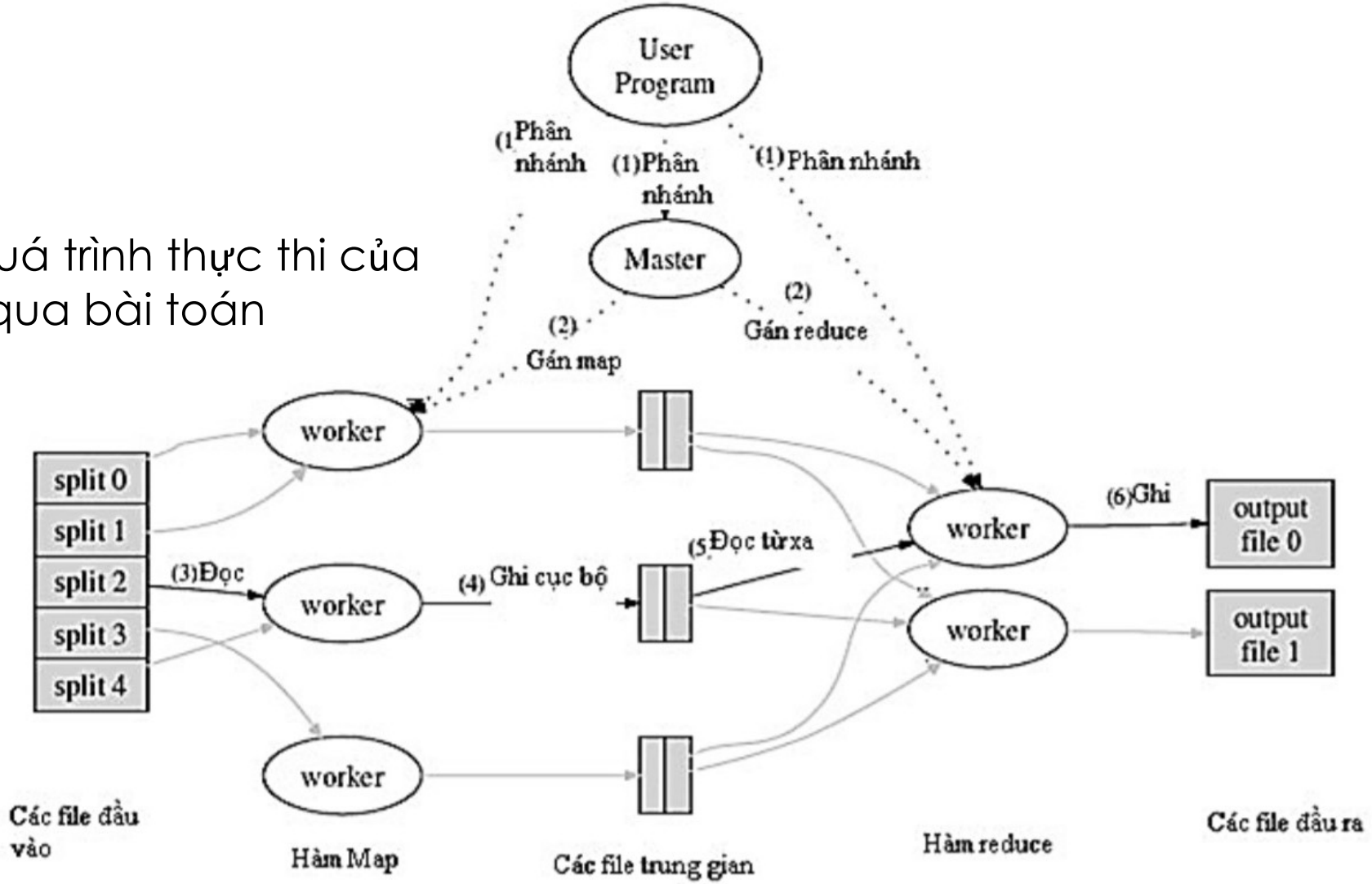
JobTracker thu gom rác, hủy các kết quả trung gian



BÀI TẬP

BÀI TẬP

Hãy miêu tả quá trình thực thi của MapReduce qua bài toán Wordcount ?



BÀI TẬP

BÀI TẬP

Câu 1: Điều gì xảy ra nếu một máy chủ trong mô hình MapReduce bị lỗi

Câu 2: Tại sao MapReduce thường được sử dụng trong các tác vụ xử lý dữ liệu lớn?

Câu 3: Tại sao Hadoop lại được sử dụng để triển khai MapReduce?

Câu 4: Hãy đề xuất một phương pháp để cải thiện hiệu suất của MapReduce.

Câu 5: Tại sao MapReduce không phù hợp cho các tác vụ xử lý thời gian thực