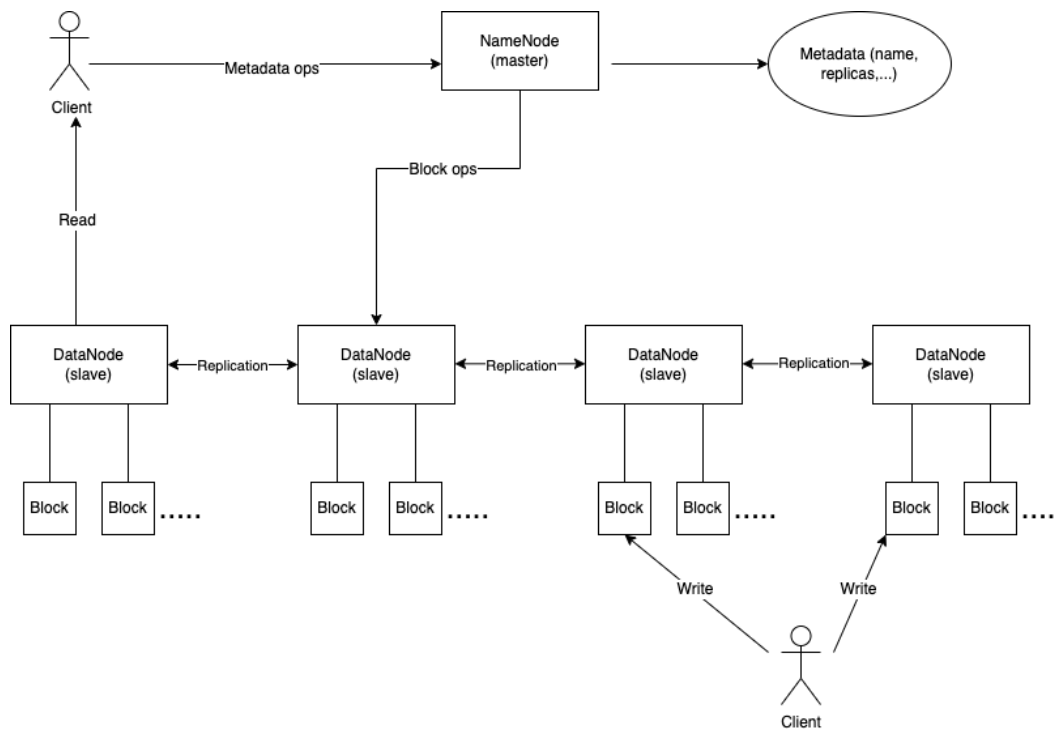


Báo cáo bài tập tuần 4 – Hadoop

Họ và tên: Lã Quốc Anh

Username: anh1q36

1. Kiến trúc hệ thống tập tin phân tán Hadoop (Hadoop Distributed File System Architecture)



HDFS Architecture

Các thành phần trong kiến trúc 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.
 - Các metadata mà Namenode lưu trữ gồm có: File System Namespace, thông tin để ánh xạ từ tên file thành danh sách các block, nơi các block được lưu trữ.
 - Namenode có nhiệm vụ chính là quản lý và duy trì các Datanode, kiểm soát thông tin và tình trạng các Datanode, lắng nghe đảm bảo các Datanode còn sống, điều khiển sự cân bằng về lưu lượng lưu trữ và lượng truy cập tới Datanode
- ➔ Nếu Name node chết cả cụm hadoop sẽ chết.

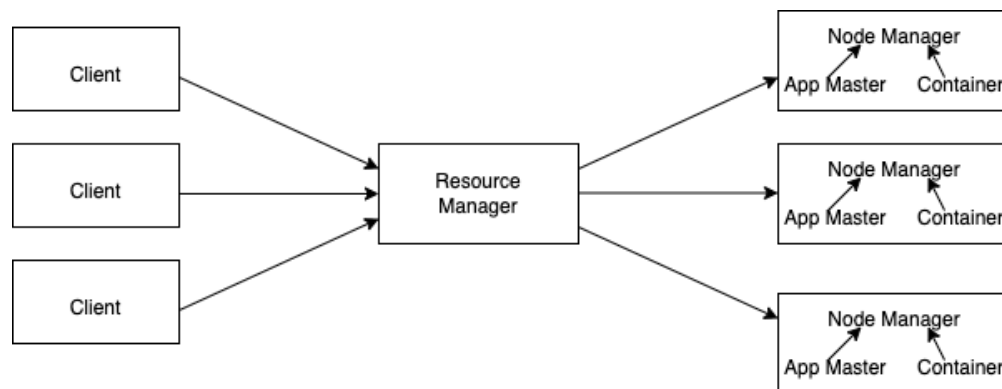
Datanode:

- Đóng vai trò là các slave, một cụm Hadoop có thể có nhiều slave tuy nhiên chỉ có một master. Datanode lưu trữ và truy xuất thông tin về các block khi nhận được yêu cầu (từ namenode hoặc người dùng)
- Là nơi chạy các job xử lý dữ liệu
- Định kỳ gửi “heartbeat” về Namenode (mặc định là 3 giây) nếu Namenode không nhận được thông tin trong khoảng thời gian 10 phút Datanode được coi như đã chết.

Block

- File trong HDFS được chia nhỏ thành nhiều mảnh gọi là block, kích thước mặc định của một block là 128MB. Tuy nhiên một tệp có size nhỏ hơn một khối sẽ không chiếm dung lượng của toàn bộ khối.
- Các block được replicate và lưu trữ tại các Datanode, khi có Datanode chết dữ liệu có thể được phục hồi từ bản các bản replicate. Đây là một đặc trưng về độ tin cậy và khả năng chịu lỗi của HDFS.

2. Kiến trúc Hadoop Yarn



Hadoop YARN Architecture

YARN (Yet Another Resource Negotiator) được giới thiệu từ Hadoop 2.0, kiến trúc của Yarn tách riêng resource management layer và processing layer.

Resource Manager:

- Chịu trách nhiệm phân công và quản lý tài nguyên giữa các ứng dụng, khi nhận được yêu cầu xử lý nó sẽ chuyển tiếp đến Node manager tương ứng và phân bổ tài nguyên để hoàn thành yêu cầu.
- Bao gồm 2 thành phần chính:
 - o Scheduler: thực hiện lập lịch dựa trên ứng dụng được phân bổ và tài nguyên có sẵn.
 - o Application Manager: chấp nhận các ứng dụng và điều phối container đầu tiên từ resource manager.

Node Manager:

- Quản lý mỗi node trên cụm Hadoop và các ứng dụng trên mỗi node. Nó đăng ký với Resource Manager và gửi health status của node.
- Giám sát tài nguyên sử dụng, quản lý log và kill container dựa trên chỉ dẫn từ Resource Manager

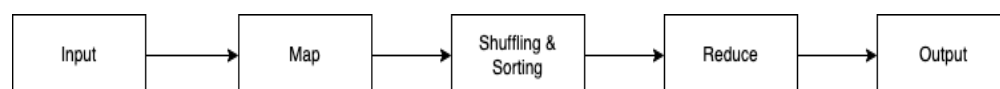
Application Master

- Quản lý vòng đời của các ứng dụng cùng với Resource Manager, khi ứng dụng bắt đầu chạy nó gửi báo cáo tình trạng cho Resource Manager theo thời gian

Container:

- Quản lý các tài nguyên vật lý như RAM, CPU trên mỗi node

3. Hadoop MapReduce



Mapping Phase:

- Bao gồm 2 bước: splitting và mapping

- Dữ liệu được chia nhỏ thành các chunks ở bước split. Hadoop bao gồm RecordReader sử dụng TextInputFormat để biến đổi thành các cặp key-value.
- Các cặp key-value sau đó được sử dụng làm đầu vào cho bước mapping, đầu ra tạo ra các cặp key-value mới.

Shuffling Phase:

- Bao gồm 2 bước chính: sorting và merging
- Trong bước sorting các cặp key-value được sắp xếp bằng các khoá, merging đảm bảo các cặp key-value được kết hợp
- Shuffling phase nhóm các key giống nhau, đầu ra cũng là các cặp key-value tương tự như map phase.

Reducer Phase:

- Tổng hợp các value dựa trên các key đã được map tương ứng, tạo ra cặp key-value mới, là kết quả của MapReduce job và được lưu trữ mặc định trong HDFS
- Các task reduce được thi hành đồng thời và độc lập, có thể không có bước Reduce.

4. Các câu lệnh thao tác:

- Xem các thư mục trong path: `hdfs dfs -ls path`
- Tạo thư mục: `hdfs dfs -mkdir /foldername`
- Tạo file: `hdfs dfs -touch /filename`
- Xoá thư mục: `hdfs dfs -rm -r /path`
- Xem file: `hdfs dfs -cat /pathToFile`
- Đưa file lên hdfs: `hdfs dfs -put filename hdfsFolder`
- Lấy file từ hdfs về: `hdfs dfs -get hdfsfile`
- Start Hadoop & Yarn: `start-all.sh`
- Stop Hadoop & Yarn: `stop-all.sh`
- Monitor Yarn: `yarn node -list`
- Monitor HDFS cluster: `hdfs dfsadmin -report`
- Monitor system status: `hdfs fsck /`
- Format namenode: `hdfs namenode -format` (chỉ chạy 1 lần nếu format nhiều lần clusterID bị thay đổi sẽ gây ra lỗi)