



# Hadoop overview

▼ Category	Theory
🔗 Files	<a href="https://www.techtarget.com/searchdatamanagement/definition/Hadoop-Distributed-File-System-HDFS">https://www.techtarget.com/searchdatamanagement/definition/Hadoop-Distributed-File-System-HDFS</a> <a href="https://cloudgeeks.net/tim-hieu-ve-kien-truc-hadoop-ecosystem/">https://cloudgeeks.net/tim-hieu-ve-kien-truc-hadoop-ecosystem/</a> <a href="https://devopsvn.xyz/2019/08/31/cai-dat-cum-hadoop-cluster-3-node-how-to-install-and-set-up-a-3-node-hadoop-cluster/">https://devopsvn.xyz/2019/08/31/cai-dat-cum-hadoop-cluster-3-node-how-to-install-and-set-up-a-3-node-hadoop-cluster/</a>
🕒 Created	@June 27, 2022 5:46 PM
📅 Reminder	
▼ Status	Open
🔗 URL	
🕒 Updated	@June 30, 2022 11:11 AM

Hadoop là gì?

- Là apache framework open source giúp phát triển các ứng dụng phân tán.

- Dùng để lưu trữ, xử lý dữ liệu lớn.

Ưu điểm:

- Có khả năng scale, tổng dung lượng lưu trữ của máy bằng tổng dung lượng của các node trong cụm nên ta có thể dễ dàng bổ sung thêm node vào cụm khi hết dung lượng lưu trữ.
- Có khả năng chịu tải, lỗi.
- Cho phép lưu trữ phân tán, khi một file lưu trữ được đẩy vào HDFS thì file đó sẽ được chia nhỏ và lưu trữ tại các node trong cụm, ngoài ra các file đó cũng có thể được nhân bản nhằm đề phòng có một node trong cụm hadoop gặp sự cố.
- Mở rộng hệ thống theo chiều ngang.

Storage/Lưu trữ: Hadoop distributed file system (HDFS).

Processing: Map reduce.

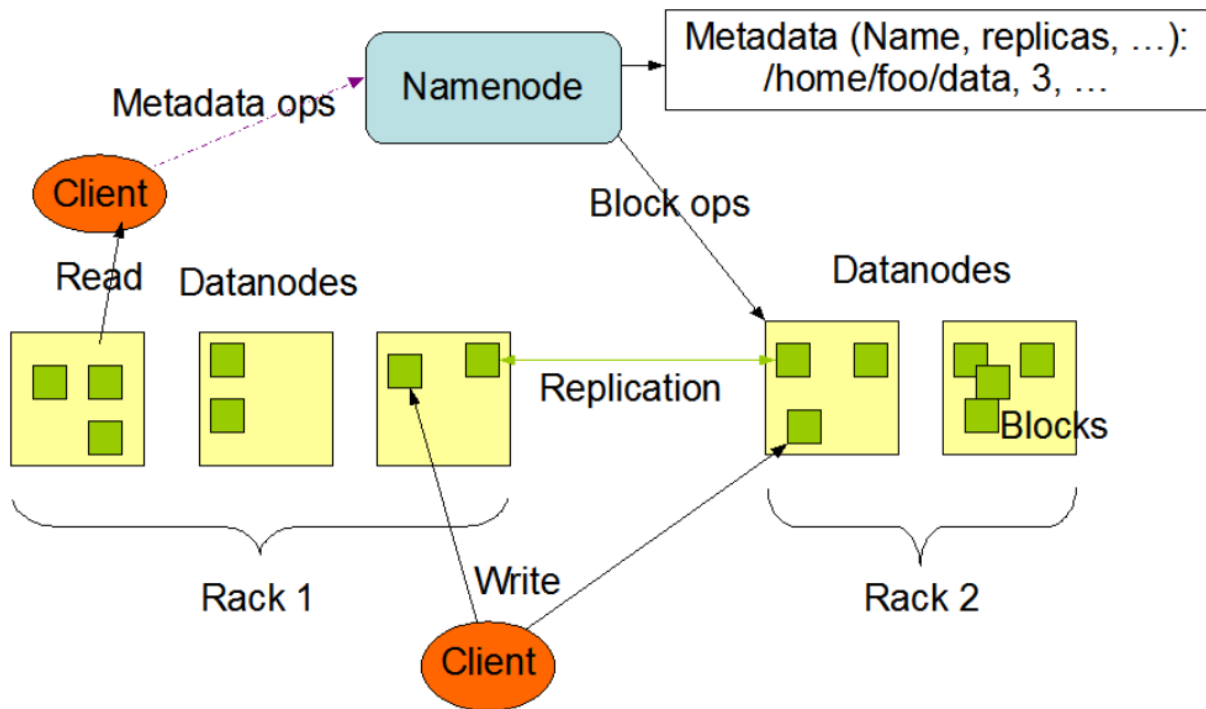
Utilities:

- Hadoop Common.
- Hadoop YARN.

## **Kiến trúc của HDFS.**

HDFS tuân theo cơ chế master-slave. Theo đó trong một cụm máy sẽ chỉ bao gồm một master gọi là Name node và nhiều slave gọi là data node.

## HDFS Architecture



Để lưu trữ dữ liệu phân tán, dữ liệu sẽ được chia thành các blocks và phân phối tới các node để xử lý. Mỗi block sẽ được replicas tới các server khác trong cụm cluster.

- Namenode:
  - Quản lý metadata(name, replicas ..) của các datanode, vị trí của các block và file.
  - Nơi quản lý truy cập, thực hiện thao tác với file từ phía client.
- Datanode:
  - Lưu trữ data.
  - Chạy các tiến trình xử lý dữ liệu.
- 2nd Namenode:
  - Đọc file, metadata được lưu trên RAM của data node và ghi vào ổ cứng.
  - Kiểm tra tính chính xác của tệp tin được ghi vào data node.

- **Blocks:** Dữ liệu đưa vào HDFS sẽ được chia thành nhiều mảnh và nằm rải rác trong data node, mỗi mảnh dữ liệu như vậy được gọi là blocks. Số lượng và vị trí của các blocks được lưu tại metadata của name node.
  - Dung lượng tối đa của mỗi block là 128M, nếu dữ liệu đầu vào lớn hơn con số này thì dữ liệu đó sẽ được chia nhỏ ra sao cho các mảnh có dung lượng  $\leq 128M$ .
  - Ngoài việc chia dữ liệu thành các block thì HDFS còn thực hiện nhân bản các block tại các data node khác nhau, việc này nhằm giảm thiểu tối đa việc mất mát dữ liệu khi một data node nào đó gặp sự cố (ta có thể cấu hình việc nhân bản dữ liệu này trong file `hdfs-site.xml`).

## Hadoop Mapreduce

Là kỹ thuật , mô hình xử lý tính toán phân tán để triển khai và xử lý dữ liệu lớn dựa trên mô hình map reduce của Google.

Ưu điểm:

- Hỗ trợ xử lý và tính toán song song: công việc được chia thành các công việc nhỏ hơn, công việc được phân chia giữa nhiều node và mỗi node hoạt động đồng thời với một phần công việc.
- Khả năng mở rộng: có khả năng mở rộng tới hàng nghìn node mà không gây ảnh hưởng lớn tới hiệu năng, lỗi.
- Tính sẵn sàng và chịu lỗi: Việc nhân bản dữ liệu tại các node giúp cho dữ liệu luôn sẵn sàng bất cứ khi nào nó được yêu cầu sử dụng. Ngoài ra, khi một node gặp sự cố thì name node sẽ chỉ định node có chứa bản sao của dữ liệu đó thực hiện tiếp công việc tránh gây ảnh hưởng tới hệ thống.

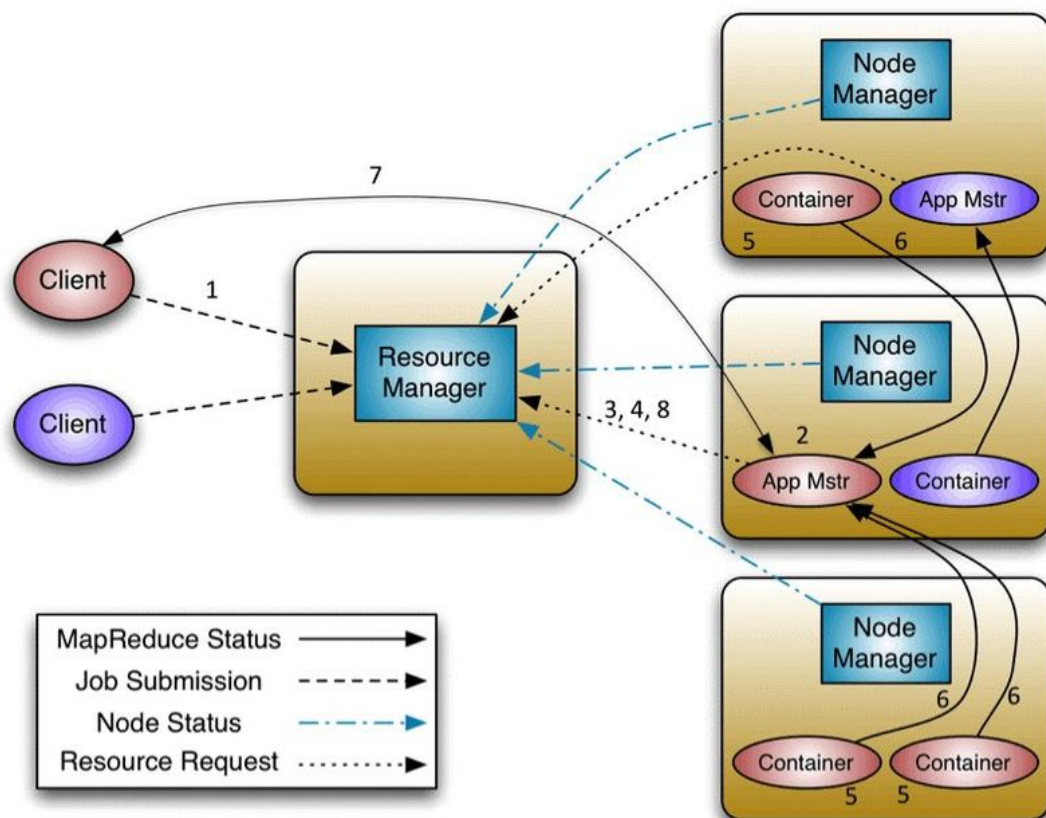
Các thành phần của hadoop mapreduce.

- **Mapper:** Gồm hai bước:
  - **Splitting:** Tập dữ liệu được chia thành các đơn vị bằng nhau(chunks).
  - **Mapping**

- Shuffle and sorting: Gồm hai bước:
  - Sorting: Các cặp key-value được sắp xếp bằng cách sử dụng các keys.
  - Merging:
- Reducer: Lấy đầu vào là đầu ra từ bước trước đó (dạng tuples: key/ value) và giảm các giá trị trung gian. Đầu ra của pha này được lưu trữ tại HDFS.

Khi công việc được hoàn thành, dữ liệu sau khi xử lý sẽ được nhân bản tại các node.

## Kiến trúc YARN



YARN là viết tắt của cụm từ “Yet another resource negotiator” là một ứng dụng hỗ trợ phát triển phân tán. Công việc của YARN là xử lý, lập lịch sử dụng tài nguyên

tính toán và giám sát quá trình thực thi các ứng dụng. Resource manager và application master đảm nhiệm thực hiện các nhiệm vụ trên. YARN bao gồm 3 thành phần chính: Resource manager, application master và node manager.

Resource manager có nhiệm vụ quản lý tài nguyên trong cluster, quản lý các node manager. Các thành phần của resource manager gồm:

- Scheduler: cung cấp tài nguyên cho các ứng dụng.
- Application master:
  - tiếp nhận các job từ phía client
  - cấp phát tài nguyên để chạy application master cho từng ứng dụng.
  - restart application master.

Node manager được cài đặt trên node có nhiệm vụ quản lý tài nguyên như RAM và các job trên một node. Ngoài ra các node manager còn quản lý các container mà resource manager giao cho.

Container là nơi lưu trữ các tài nguyên của cluster, mỗi container có một id khác nhau.

Các bước thực thi job với hadoop YARN

1. Client gửi resource request của job tới Resource manager(RM).
2. Sau đó, RM tiến hành cấp phát tài nguyên cho việc khởi chạy Application master(AM).
3. AM đăng ký với RM cho phép Client giao tiếp với AM.
4. AM thực hiện resource request với RM.
5. Khởi tạo container trên node manager(NM) từ tài nguyên mà AM request từ RM. Các container có thể giao tiếp với AM.

6. Khi thực thi chương trình, client giao tiếp trực tiếp với AM để cập nhật tiến độ thực hiện task.
7. Khi hoàn thành các task, AM ngắt kết nối với RM và đồng thời tắt các container.