

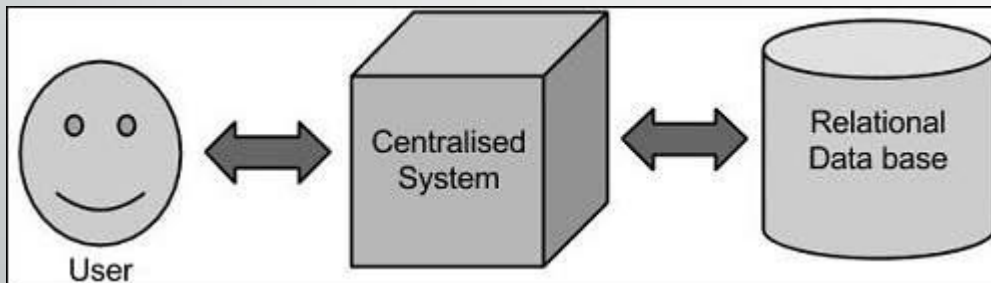
Hadoop



January 2021

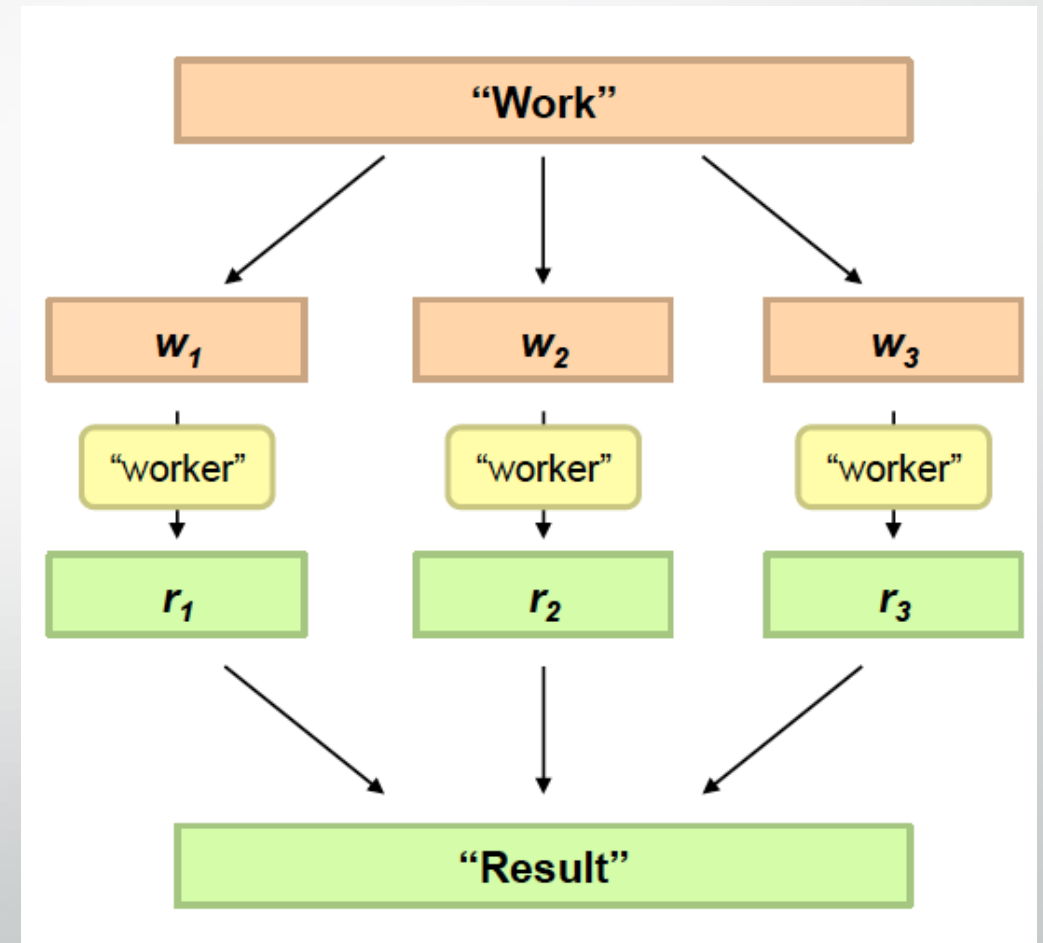
Phương pháp xử lý truyền thống

- Dữ liệu được quản lý & xử lý tập trung
- Sử dụng công cụ quản trị dữ liệu từ các nhà cung cấp: Oracle, IBM, Microsoft, ...
- Người dùng truy cập tới cơ sở dữ liệu thông qua ứng dụng quản trị CSDL tập trung
- Hạn chế
 - Chỉ hoạt động tốt với dữ liệu có dung lượng nhỏ có thể xử lý được bởi các máy chủ CSDL thông thường
 - Để xử lý big data, các công cụ truyền thống (RDBMS) cần nâng cấp CPU, bộ nhớ cho DBMS
 - Phần lớn dữ liệu ngày nay ở dạng không cấu trúc hoặc bán cấu trúc (social media, audio, video, texts, and emails)



Giải pháp của Google

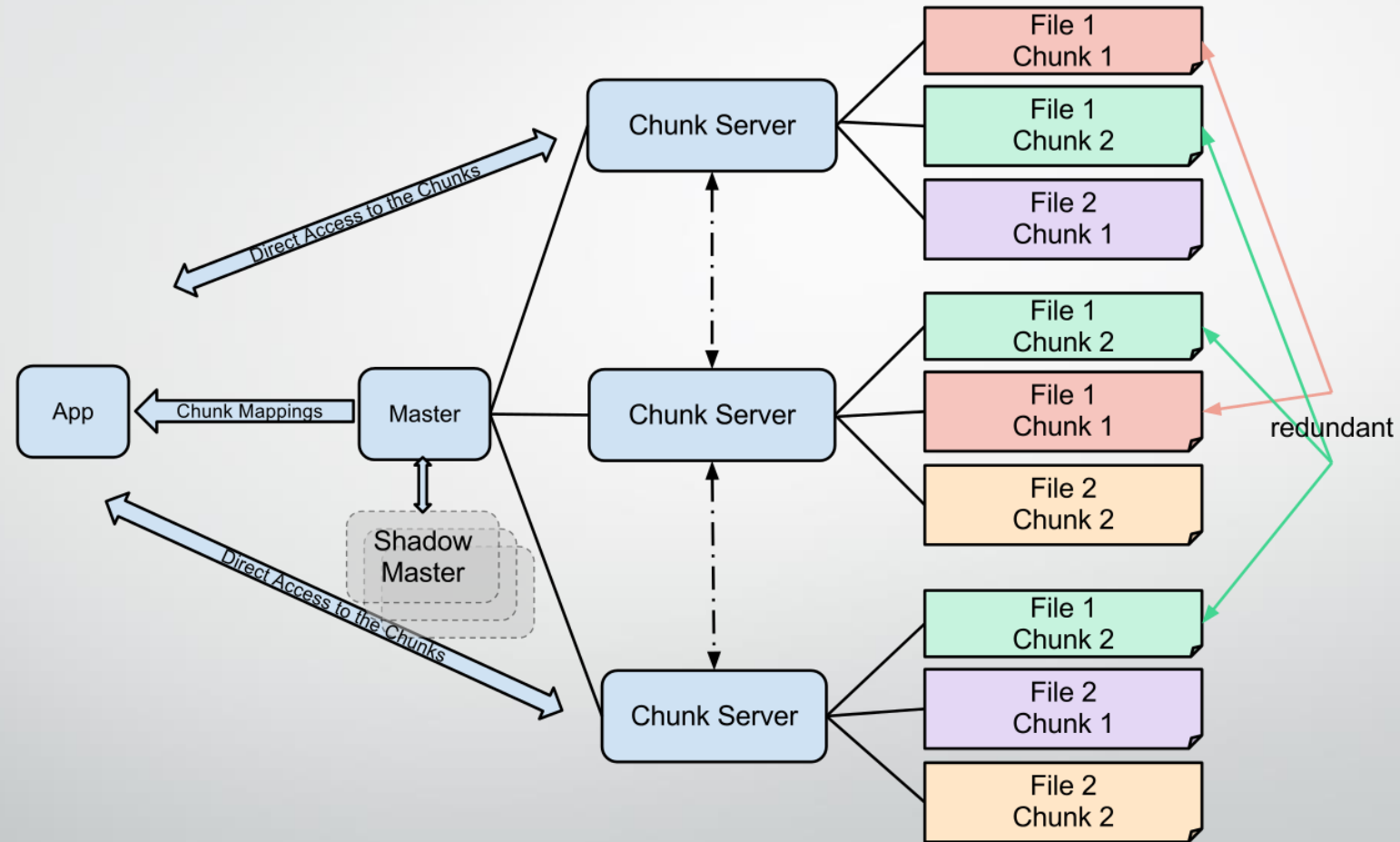
- Để xử lý dữ liệu lớn, Google đưa ra thuật toán gọi là MapReduce
- Giải pháp này chia tác vụ thành nhiều phần nhỏ giao cho nhiều máy tính xử lý, sau đó tổng hợp kết quả từ các máy trạm để cho ra kết quả xử lý tổng thể



Google File System

- Các đặc điểm của xử lý dữ liệu lớn
 - Dung lượng dữ liệu rất lớn (hàng GB) đòi hỏi hệ thống quản lý tập tin phân tán có khả năng mở rộng
 - Khi xử lý phân tán, có khả năng một số trạm xử lý bị lỗi, đòi hỏi hệ thống có khả năng chịu lỗi cao (fault-tolerance)
 - Hầu hết việc cập nhật dữ liệu được xử lý ở cuối các files.
 - Yêu cầu duy trì băng thông (bandwidth) cao
- ➔ • Giải pháp: Google File System (GFS)
 - Hiệu quả, tin cậy
 - Khả năng mở rộng và sẵn sàng cao

Kiến trúc GFS



GFS

- Mỗi GFS cluster gồm 1 server chính (master server) và nhiều máy trạm (chunkservers).
- Mỗi máy trạm thường là 1 máy chủ Linux, xử lý các yêu cầu của người dùng
- Các files được chia thành nhiều chùm (chunks) có kích thước cố định. Mỗi chùm được quản lý bởi master server.
- Để tăng độ tin cậy mỗi chunk dữ liệu được sao lưu lên một số máy trạm khác nhau.
- Máy server chính quản lý metadata của hệ thống, bao gồm: không gian tên (namespace), thông tin điều khiển truy cập, sơ đồ phân chia dữ liệu ra các máy trạm (mapping), vị trí hiện tại của các chunk.
- Server chính định kỳ liên lạc với từng máy trạm để ghi nhận trạng thái và gửi chỉ thị.

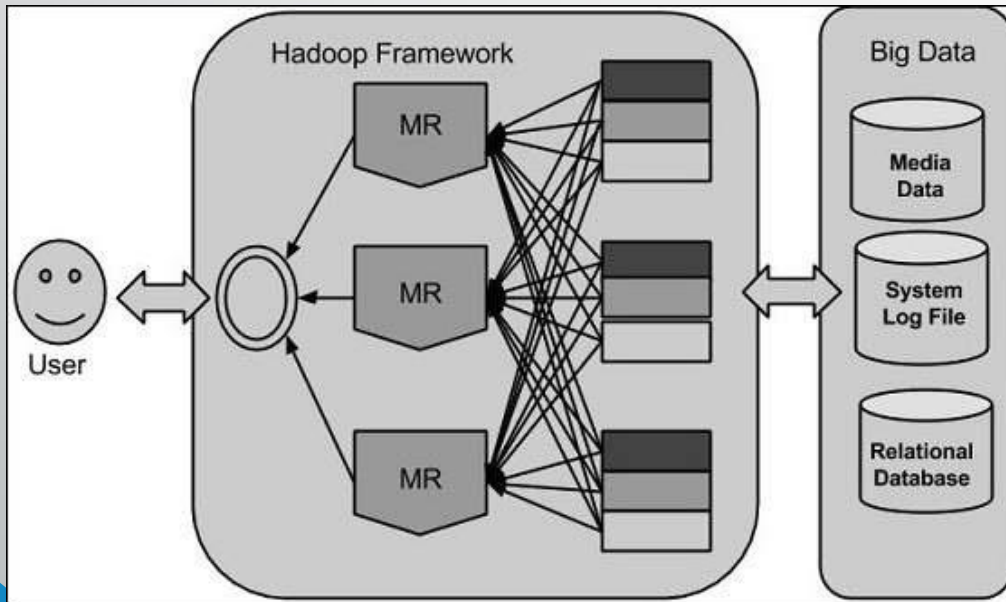


Hadoop là gì?

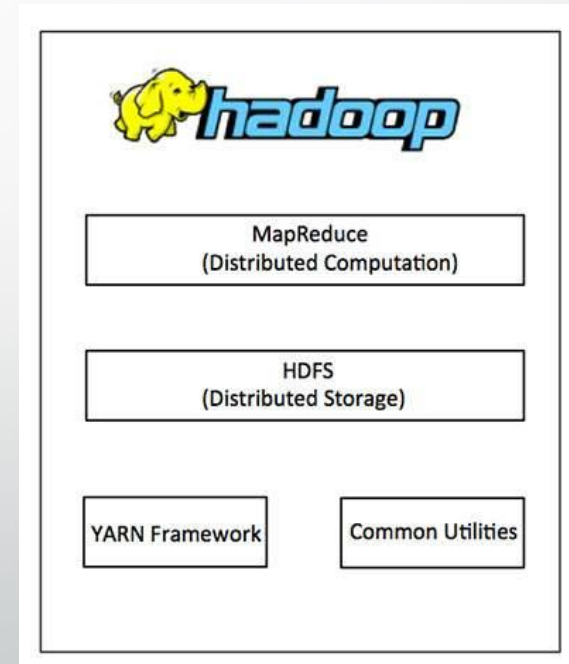
- Hadoop là công cụ được phát triển nhằm đáp ứng xử lý dữ liệu lớn trong giới hạn chi phí và thời gian hợp lý.
- 2005, **Doug Cutting** và cộng sự phát triển phần mềm mã nguồn mở, nhằm tạo ra nền tảng tính toán phân tán, tin cậy và dễ mở rộng. Sản phẩm được đặt tên là Hadoop.
- Hadoop hỗ trợ xử lý phân tán trên dữ liệu lớn sử dụng mô hình MapReduce của Google.
- Có thể triển khai Hadoop trên các máy đơn lẻ hoặc trên cụm hàng ngàn máy trạm. Mỗi máy trạm có bộ xử lý và lưu trữ riêng.
- 2006, Yahoo! chuyển Hadoop cho Apache Software Foundation
- Để xử lý dữ liệu lớn, thay vì sử dụng một vài máy tính cấu hình siêu mạnh, Hadoop kết nối nhiều máy tính thông thường thành một cụm (cluster) cùng xử lý.
- Các cụm máy tính đọc dữ liệu song song và cho thông lượng nhanh hơn, chi phí thấp hơn các siêu máy tính.

Kiến trúc tổng quát của Hadoop

Mô hình xử lý



Kiến trúc tổng quát



Kiến trúc tổng quát của Hadoop (cont.)

- Lớp xử lý/tính toán (Processing/Computation layer) (**MapReduce - MR**)
 - MR là mô hình tính toán song song dùng cho các ứng dụng phân tán, được đề xuất bởi Google để xử lý hiệu quả dữ liệu lớn.
 - Có thể thực thi chương trình MR trên hệ thống Hadoop với hàng nghìn trạm xử lý (mỗi trạm gọi là node).
 - Độ tin cậy, khả năng chịu lỗi cao.

Kiến trúc tổng quát của Hadoop (cont.)

- Lớp quản lý lưu trữ (Storage layer) (**Hadoop Distributed File System - HDFS**)
 - HDFS dựa trên GSF.
 - Là hệ thống quản lý tập tin phân tán.
 - Tốc độ xử lý dữ liệu cao.
 - Khả năng chịu lỗi cao.
 - Thiết kế để chạy trên phần cứng giá rẻ.

Kiến trúc tổng quát của Hadoop (cont.)

- **Hadoop Common Utilities**
 - Thư viện Java và các tiện ích phục vụ cho các modules Hadoop khác.
- **Hadoop YARN**
 - Là nền tảng quản lý tài nguyên và lập lịch cho các tác vụ.

Trình tự xử lý của Hadoop

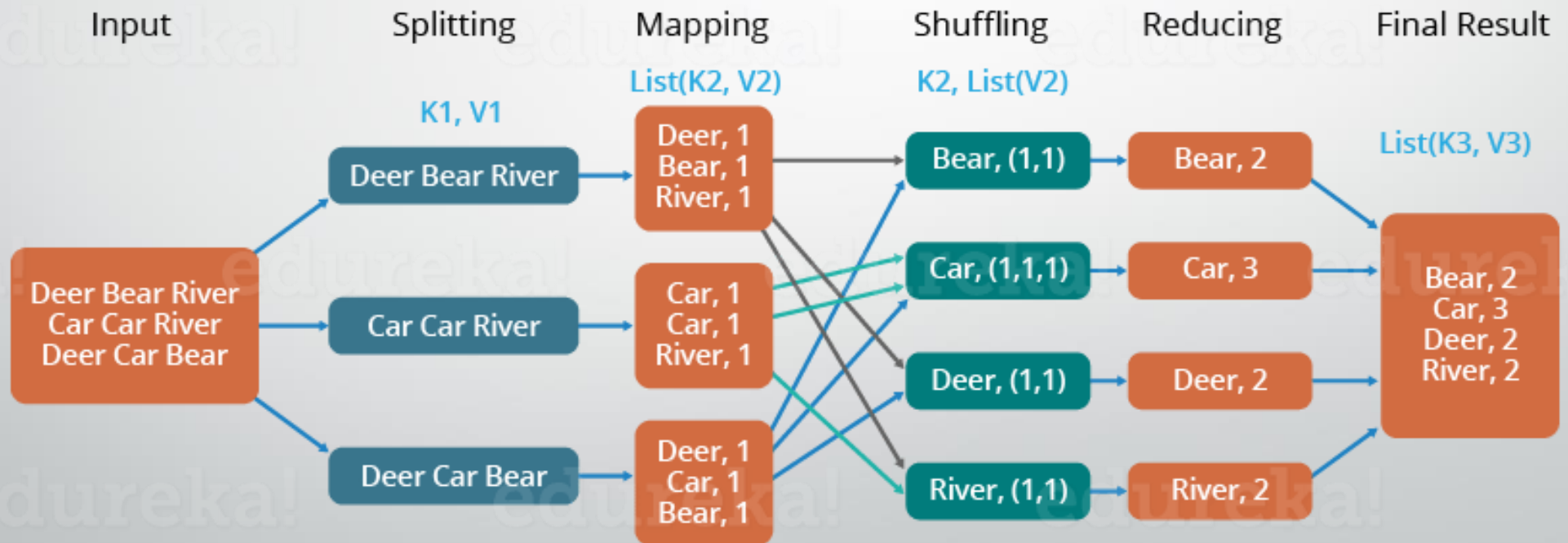
1. Dữ liệu được chia thành nhiều thư mục và files. Các files được chia thành các khối có cùng kích thước (64MB, 128MB).
2. Các khối dữ liệu được phân tán cho các trạm (nodes) để xử lý.
3. HDFS giám sát quá trình xử lý.
4. Để đề phòng lỗi phần cứng, các khối dữ liệu được sao chép lên một vài trạm khác.
5. Kiểm tra việc mã lệnh đã thực thi thành công.
6. Tiến hành việc sắp xếp kết quả, diễn ra giữa 2 giai đoạn map và reduce.
7. Gửi kết quả đã sắp xếp.
8. Ghi lại quá trình cho mỗi tác vụ.

Trình tự xử lý của Hadoop (cont.)

- Ví dụ: Chương trình đếm từ

The Overall MapReduce Word Count Process

edureka!



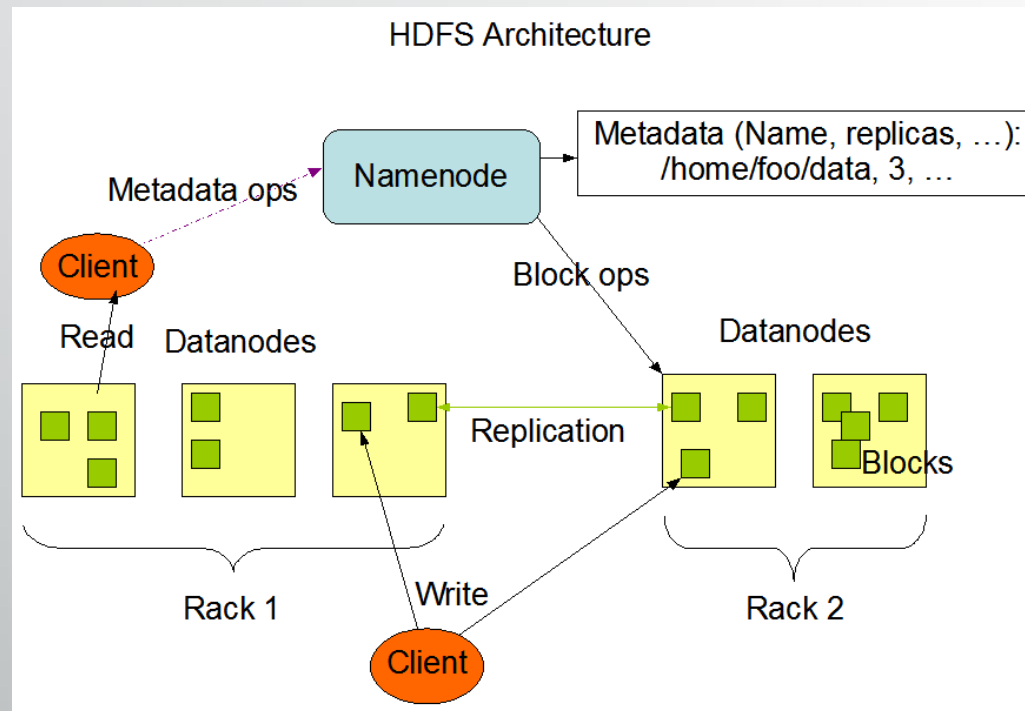
Ưu điểm của Hadoop

- Cho phép tạo ứng dụng xử lý phân tán nhanh chóng, hiệu quả.
- Việc xử lý phân tán hoàn toàn tự động, giúp tối ưu tài nguyên phần cứng.
- Khả năng chịu lỗi và sẵn sàng cao mà không phụ thuộc phần cứng.
- Bảo đảm hoạt động liên tục khi thêm/bớt các trạm xử lý.
- Mã nguồn mở, tương thích với mọi nền tảng (do được phát triển bằng Java)

HDFS (Hadoop Distributed File System)

- Thiết kế cho lưu trữ và xử lý phân tán
- Cung cấp môi trường giao tiếp dòng lệnh (command interface) với HDFS
- Thông qua các máy chủ thiết lập sẵn (built-in servers): namenode và datanode, người dùng dễ dàng kiểm tra, quản lý tình trạng của cụm máy tính (cluster).
- Hỗ trợ truy xuất liên tục đến dữ liệu (streaming access)
- Có chức năng phân quyền, xác thực tập tin.

Kiến trúc HDFS



- **HDFS** có mô hình master/stave
- Một HDFS có 1 máy chủ chính (master server) gọi là **NameNode**
- Có nhiều máy chủ lưu trữ dữ liệu, gọi là **DataNodes**
- **NameNode**
 - Xử lý mở, đóng file/thư mục
 - Phân phối các khối dữ liệu tới các **DataNodes**
- **DataNodes**
 - Chịu trách nhiệm xử lý các yêu cầu đọc/ghi dữ liệu
 - Thực hiện tạo/xóa các khối dữ liệu, nhân bản dữ liệu theo chỉ thị từ máy chủ **NameNode**

Sao lưu dữ liệu trong HDFS

- HDFS lưu mỗi file thành một dãy các khối (block) có cùng kích thước (ngoại trừ block cuối).
- Để tăng độ tin cậy, các khối dữ liệu được sao lưu.
- Cho phép thiết lập kích thước block và số bản sao (replicas).
- Máy chủ chính (NameNode):
 - Quản lý việc sao lưu các khối dữ liệu.
 - Định kỳ nhận trạng thái (heartbeat) và báo cáo (blockreport) từ mỗi máy trạm (DataNode)
 - Heartbeat cho biết máy trạm có hoạt động bình thường hay không
 - Blockreport chứa danh sách tất cả các khối trên máy trạm

Truy cập HDFS

- Hadoop cung cấp [FileSystem Java API](#) cho các ứng dụng để truy xuất HDFS. use.
- Có thể truy cập các files của HDFS thông qua trình duyệt Web
- Có thể gắn HDFS với hệ thống quản lý file của máy tính cục bộ thông qua công cụ [NFS gateway](#)
- **FS Shell**
 - HDFS cho phép tổ chức dữ liệu thành các file và thư mục.
 - FS shell - giao diện dòng lệnh cho phép người dùng thao tác với dữ liệu trên HDFS.
 - FS shell có cú pháp tương tự các tập lệnh trên Linux (bash, csh)

FS Shell

Hành động	Lệnh
Tạo thư mục có tên là /mydata	bin/hadoop dfs -mkdir /mydata
Xóa thư mục /mydata	bin/hadoop fs -rm -R /mydata
Hiển thị nội dung file văn bản	bin/hadoop dfs -cat /mydata/myfile.txt
Sao chép dữ liệu lên HDFS	hadoop fs -copyFromLocal <localsrc>

Các lệnh thường dùng trên HDFS

Cú pháp	Diễn giải
<code>-ls <path></code>	Tác dụng như lệnh ls (list) trên Linux.
<code>-lsr <path></code>	Tác dụng tương tự lệnh -ls nhưng hiển thị cả nội dung các thư mục con.
<code>-du <path></code>	Tác dụng như lệnh du (disk usage) trên Linux.
<code>-mv <src> <dest></code>	Tác dụng như lệnh mv (move) trên Linux.
<code>-cp <src> <dest></code>	Tác dụng như lệnh cp (copy) trên Linux.
<code>-rm <path></code>	Tác dụng như lệnh rm (remove) trên Linux.
<code>-put <localSrc> <dest></code>	
<code>-copyFromLocal <localSrc> <dest></code>	Tác dụng như lệnh -put
<code>-moveFromLocal <localSrc> <dest></code>	Di chuyển file/thư mục từ ổ đĩa máy cục bộ lên HDFS.
<code>-get [-crc] <src> <localDest></code>	Copy file/thư mục từ HDFS sang ổ đĩa cục bộ.

Các lệnh thường dùng trên HDFS (cont.)

Cú pháp	Diễn giải
<code>-cat <filename></code>	Hiển thị nội dung của file.
<code>-copyToLocal <src> <localDest></code>	Tác dụng như -get
<code>-moveToLocal <src> <localDest></code>	Di chuyển file/thư mục từ HDFS sang ổ đĩa cục bộ
<code>-mkdir <path></code>	Như lệnh mkdir (make directory) của Linux.
<code>-chmod [-R] mode,mode,... <path>...</code>	Như lệnh chmod của Linux.
<code>-chown [-R] [owner][:[group]] <path>...</code>	Như lệnh chown của Linux.
<code>-help <cmd-name></code>	Hiện hướng dẫn sử dụng cho một lệnh. Ví dụ, xem hướng dẫn lệnh -ls : hdfs dfs -help ls

HDFS (Hadoop Distributed Filesystem)

- Để lưu trữ dữ liệu lớn - vượt quá khả năng lưu trữ của một máy tính đơn lẻ - một giải pháp là phân chia chúng làm nhiều phần và lưu trữ trên nhiều máy tính.
- Hệ thống quản lý lưu trữ trên một mạng nhiều máy tính được gọi là **hệ thống tập tin phân tán** (distributed filesystems).
- Hệ thống quản lý tập tin phân tán hoạt động trên mạng máy tính -> phức tạp hơn hệ thống file thông thường. Chẳng hạn, một thách thức là bảo đảm hệ thống hoạt động ổn định khi có một số máy trạm bị lỗi.
- Hệ thống quản lý file phân tán của Hadoop gọi là HDFS.

Thiết kế của HDFS

- HDFS được thiết kế với mục tiêu:
 - Lưu trữ những file dung lượng lớn: Lên đến terabytes, petabytes.
 - Hỗ trợ cơ chế truy cập streaming: HDFS giả định trường hợp xử lý dữ liệu lớn chủ yếu là ghi một lần (write-once), đọc nhiều lần (read-many-times).
 - Hoạt động tin cậy, ổn định trên phần cứng phổ thông (có khả năng xảy ra lỗi cao).
- Nhược điểm của HDFS
 - Có độ trễ trong truy xuất dữ liệu
 - Những ứng dụng yêu cầu truy cập dữ liệu thấp (hàng chục mili giây) không hoạt động tốt với HDFS. HDFS cho thông lượng dữ liệu cao, bù lại, cho độ trễ cao.
 - Có quá nhiều file dung lượng bé

Các thuật ngữ trên HDFS

- Blocks
 - Đĩa từ có kích thước khối lưu trữ (block size), là dung lượng nhỏ nhất có thể đọc, ghi. Chẳng hạn, Windows 10 có block size mặc định là 4 KB.
 - Block trên HDFS có dung lượng lớn, mặc định là 128 MB.
 - Nếu 1 file trên HDFS nhỏ hơn kích thước 1 block thì nó không chiếm toàn bộ block như trên Windows. Ví dụ: file 1 MB lưu trên HDFS với kích thước block 128 MB chỉ sử dụng 1 MB.
 - Lý do kích thước HDFS block lớn: giúp tìm kiếm nhanh hơn.

HDFS: Namenodes and Datanodes

HDFS có 2 loại trạm (node) vận hành theo cơ chế master-worker: namenode và datanodes

- Namenode

- Quản lý không gian tên của hệ thống quản lý file
- Chứa cây thư mục và siêu dữ liệu (metadata) của tất cả file và thư mục.

- Datanodes

- Lưu trữ các blocks dữ liệu khi có yêu cầu (bởi clients hoặc namenode)
- Định kỳ gửi namenode danh sách các blocks đang lưu trữ

HDFS: Các thao tác cơ bản

- Tạo thư mục trên HDFS

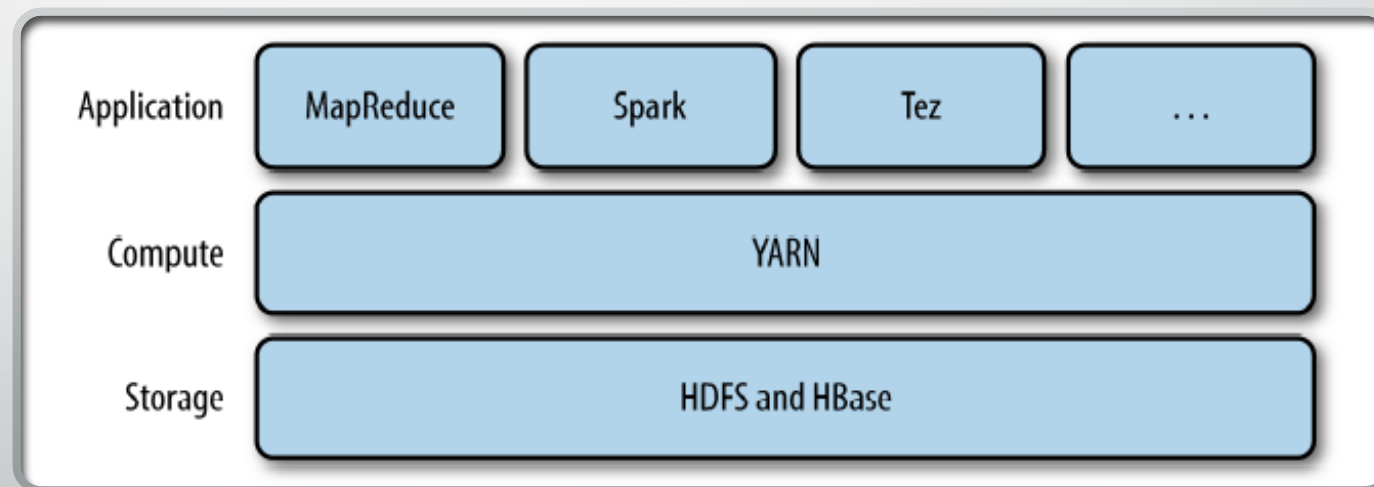
`hadoop fs -mkdir [-p] <dir>`

- Đưa dữ liệu lên HDFS

`hadoop fs -copyFromLocal <localSrc> <hdfs_Dest>`

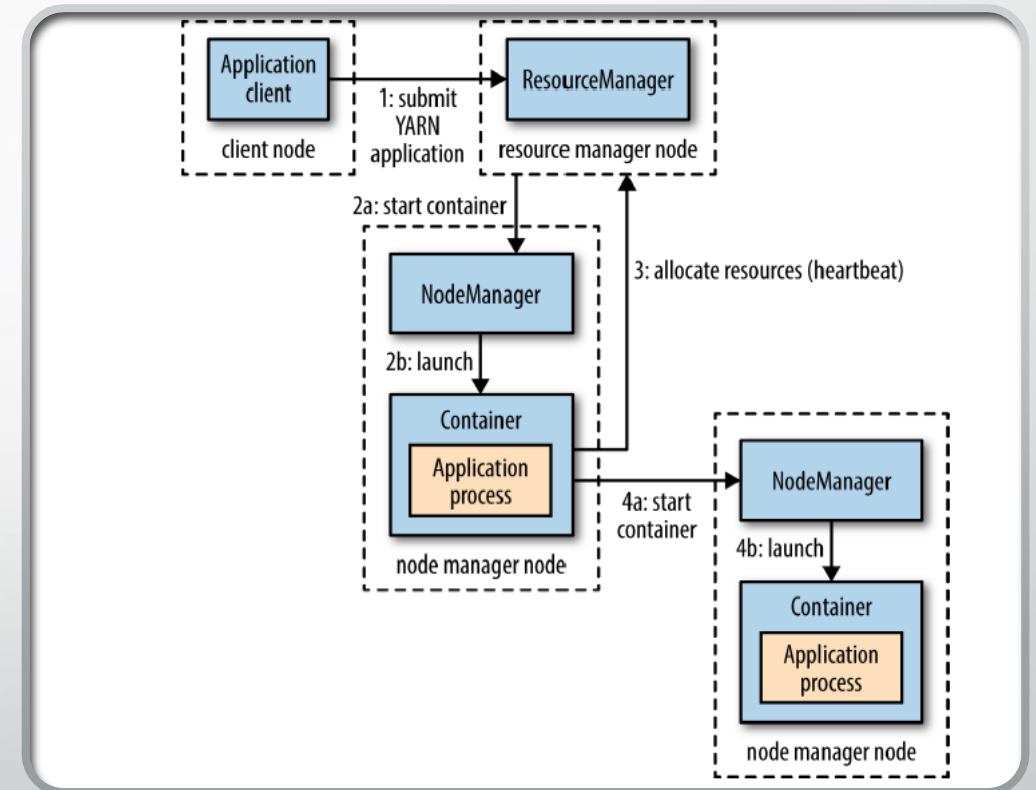
YARN (Yet Another Resource Negotiator)

- YARN là hệ thống quản lý tài nguyên của Hadoop (từ version 2)
- Cung cấp các APIs quản lý tài nguyên. Các APIs này được gọi thông qua các nền tảng xử lý trên nền YARN (MapReduce, Spark).



CÁCH THỨC YARN THỰC THI ỨNG DỤNG

- YARN cung cấp các dịch vụ lõi:
 - *Resource manager* quản lý tài nguyên trên toàn cluster
 - *Node managers* chạy trên tất cả các node để khởi động và giám sát *containers*
 - *Container* chạy tiến trình gắn với ứng dụng với tài nguyên ràng buộc
 - YARN chạy ứng dụng



Tạo ứng dụng YARN

- Viết một ứng dụng YARN khá phức tạp
- Đa số ứng dụng có thể được viết dựa trên nền tảng có sẵn.
- Ví dụ: Để viết ứng dụng xử lý đồ thị có hướng không tuần hoàn (DAG – directed acyclic graph), có thể dùng Spark
- Các nền tảng hỗ trợ tạo ứng dụng YARN:
 - Apache Slider
 - Apache Twill

Hadoop I/O

- Toàn vẹn dữ liệu (Data Integrity)
 - Bảo đảm không xảy ra mất hoặc lỗi dữ liệu
 - Cách thông thường để phát hiện lỗi dữ liệu là tính toán *checksum*. Tính lần đầu khi đưa dữ liệu vào, tính lại khi nó được truyền đi. Nếu checksum không khớp -> khả năng dữ liệu bị lỗi. Phương pháp này chỉ phát hiện mà không sửa lỗi.
 - Ví dụ: Mã CRC-32 (32-bit cyclic redundancy check)
- Kiểm tra toàn vẹn dữ liệu trên HDFS
 - HDFS tạo checksums tất cả dữ liệu ghi lên nó và mặc định đối chiếu checksum khi đọc dữ liệu.
 - Kích thước checksum (bytes) được xác định bởi thuộc tính *dfs.bytes-perchecksum*, mặc định là 512.

Nén dữ liệu trên Hadoop

- Nén dữ liệu có 2 lợi ích: giảm dung lượng bộ nhớ lưu trữ và tăng tốc độ truyền trên mạng.
- Hadoop cung cấp nhiều định dạng nén

Compression format	Tool	Algorithm	Filename extension	Splittable?
DEFLATE ^a	N/A	DEFLATE	<i>.deflate</i>	No
gzip	<i>gzip</i>	DEFLATE	<i>.gz</i>	No
bzip2	<i>bzip2</i>	bzip2	<i>.bz2</i>	Yes
LZO	<i>lzop</i>	LZO	<i>.lzo</i>	No ^b
LZ4	N/A	LZ4	<i>.lz4</i>	No
Snappy	N/A	Snappy	<i>.snappy</i>	No