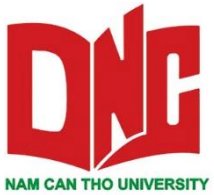
BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC NAM CẦN THƠ



ĐIỆN TOÁN ĐÁM MÂY

Chương 3:

LƯU TRỮ VÀ XỬ LÝ DỮ LIỆU

Giảng viên: Nguyễn Trung Kiên

Hệ thống lưu trữ phân tán và đồng nhất bộ nhớ NFS, AFS

- **NFS** (Network File System): là kiến trúc hệ thống tập tin phân tán mà một máy chủ trong hệ thống đóng vai trò là máy chủ lưu trữ, cung cấp năng lực lưu trữ của các ổ đĩa cứng cục bộ, hệ thống RAID cho các máy tính khác qua giao thức mạng.
- NFS là kiến trúc hệ quản lý tập tin phân tán rất phổ biến, được hỗ trợ bởi hầu hết các nền tảng hệ điều hành như Windows, Unix.
- Ưu điểm của NFS là tính trong suốt cho người dùng cuối về cách thức truy cập tập tin hay vị trí nơi tập tin được lưu trữ.
- Nhược điểm của NFS là tính khả mở rộng thấp do mọi thao tác đọc ghi dữ liệu đều thực hiện qua kết nối mạng với máy chủ lưu trữ NFS.

Hệ thống lưu trữ phân tán và đồng nhất bộ nhớ NFS, AFS (tt)

- **AFS** (Andrew File System): cũng là một hệ thống tập tin phân tán nhằm mục đích chia sẻ tập tin cho một lượng lớn người dùng mạng.
- AFS có tính khả mở rộng cao, đáp ứng được số lượng người dùng lớn hơn.
- Khi truy cập tập tin, toàn bộ tập tin sẽ được sao chép về phía máy người sử dụng và các thao tác đọc ghi được thực hiện trên tập tin đó. Khi tập tin được đóng, nội dung tập tin sẽ được cập nhật về phía máy chủ lưu trữ.

Hệ thống lưu trữ HDFS, GFS

- Hệ thống tệp phân tán giúp lưu trữ và quản lý các tệp tin lớn trên nhiều nút mạng, đảm bảo tính sẵn sàng và độ tin cậy.
- HDFS và GFS là hai trong số các hệ thống lưu trữ phổ biến nhất, hỗ trợ xử lý khối lượng dữ liệu lớn trong môi trường phân tán.

Hệ thống lưu trữ HDFS, GFS (tt)

HDFS (Hadoop Distributed File System)

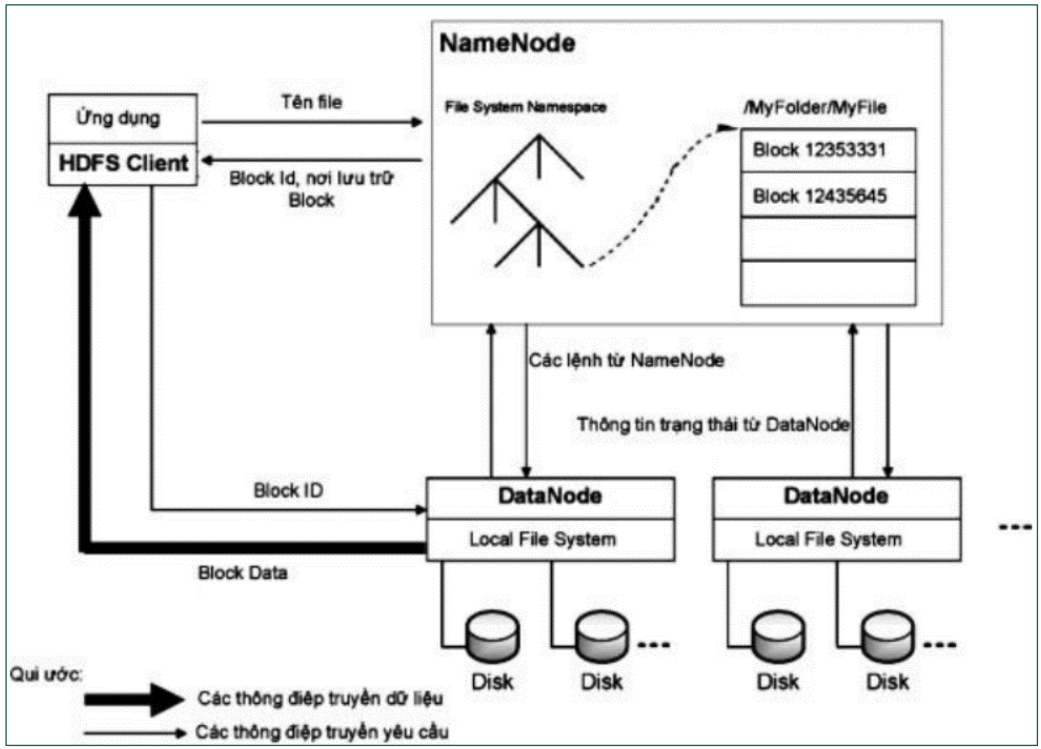
- Nguồn gốc: HDFS là một phần của Hadoop, được phát triển bởi Apache, xuất phát từ nhu cầu lưu trữ dữ liệu của dự án Nutch.
- Chức năng: HDFS được thiết kế để lưu trữ các tập dữ liệu lớn, tối ưu hóa cho việc xử lý các tệp có kích thước hàng GB đến TB trên các cụm (cluster) gồm hàng trăm đến hàng nghìn nút.

Hệ thống lưu trữ HDFS, GFS (tt)

- Các giả định thiết kế của HDFS:
  - Phần cứng lỗi thường xuyên: HDFS phải đối mặt với việc các lỗi phần cứng sẽ xảy ra thường xuyên, do đó khả năng chịu lỗi và phục hồi là rất quan trọng.
  - Tối ưu cho tệp lớn: HDFS tối ưu hóa cho việc lưu trữ các tệp lớn, chia nhỏ dữ liệu thành các khối lớn (thường là 64MB) và phân tán trên các nút lưu trữ.
  - Chỉ đọc: các tệp tin trong HDFS thường chỉ đọc (read-only) sau khi được tạo ra, với các thay đổi chỉ được thực hiện bằng cách thêm dữ liệu vào cuối tệp.

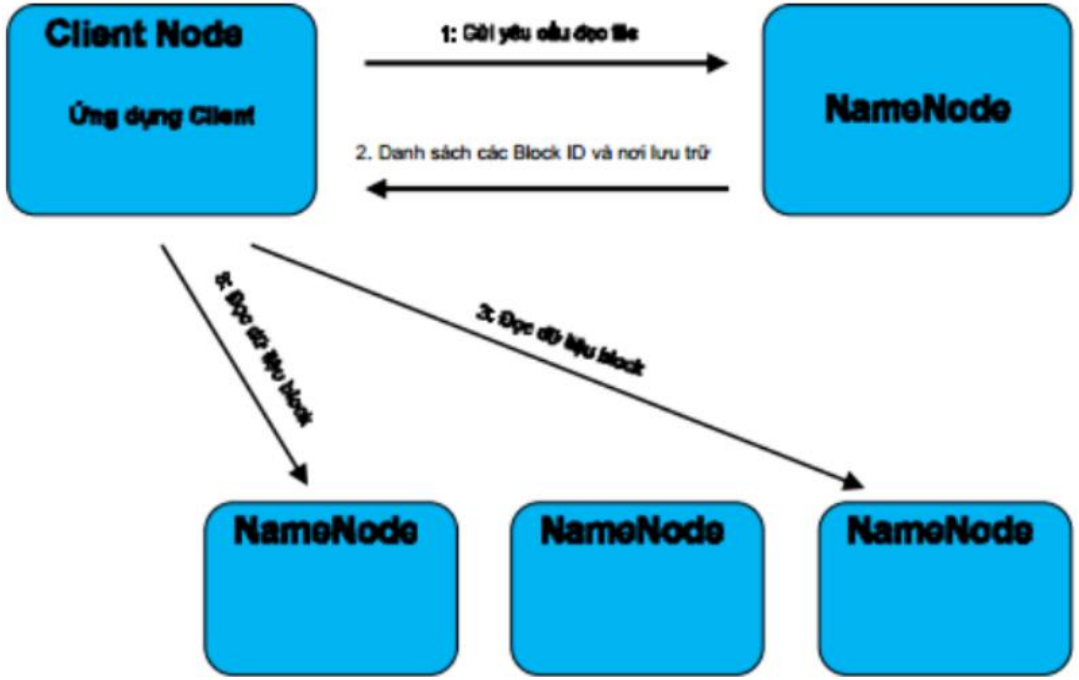
Hệ thống lưu trữ HDFS, GFS (tt)

- Kiến trúc của HDFS:
  - NameNode: nút quản lý trung tâm, chịu trách nhiệm lưu trữ thông tin về cấu trúc thư mục, siêu dữ liệu (metadata) của hệ thống tệp.
  - DataNode: các nút lưu trữ dữ liệu thực tế, mỗi khối dữ liệu được lưu trữ dưới dạng các tệp riêng biệt trên các DataNode.
  - Block và Replication: mỗi tệp được chia thành các khối (block) và mỗi khối được sao chép trên nhiều DataNode để đảm bảo an toàn dữ liệu.



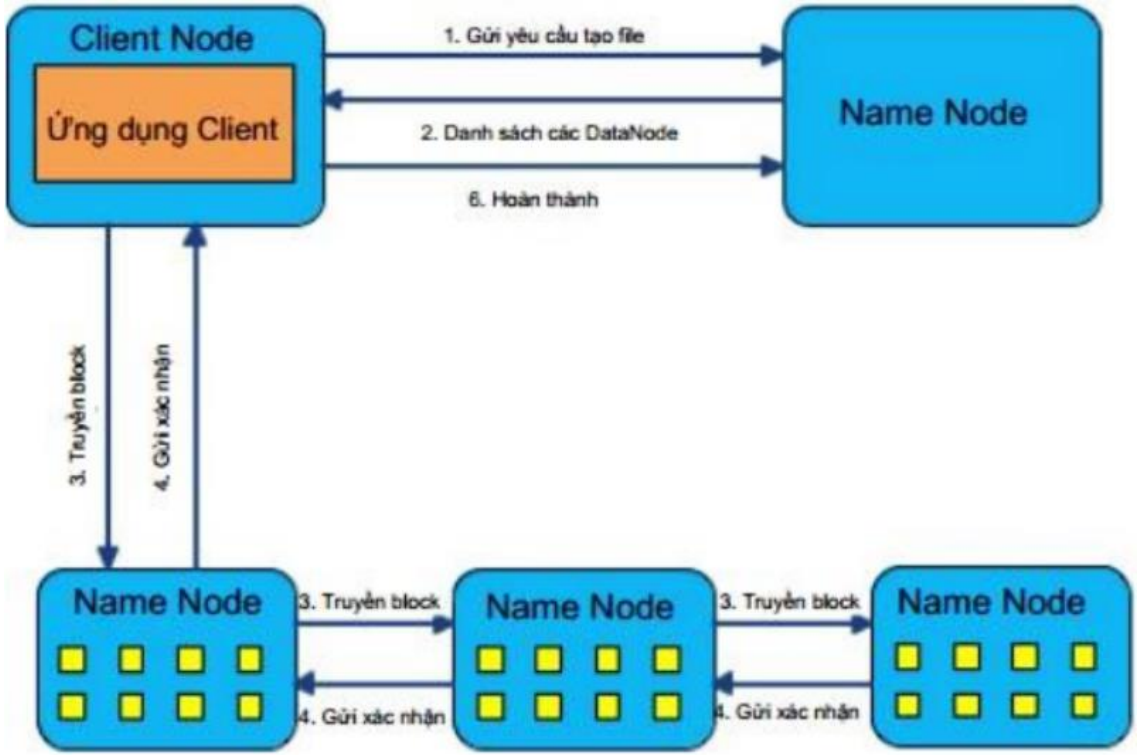
Hệ thống lưu trữ HDFS, GFS (tt)

- Quá trình đọc tập tin trên HDFS:
  - Client gửi yêu cầu đến NameNode để nhận thông tin về vị trí các khối dữ liệu.
  - Client truy cập trực tiếp các DataNode để đọc các khối dữ liệu.



Hệ thống lưu trữ HDFS, GFS (tt)

- Quá trình ghi tập tin trên HDFS:
  - Client gửi yêu cầu tạo tệp mới đến NameNode.
  - NameNode quyết định các DataNode sẽ lưu trữ các khối dữ liệu và gửi lại cho client.
  - Client ghi dữ liệu trực tiếp lên các DataNode.



Hệ thống lưu trữ HDFS, GFS (tt)

GFS (Google file system)

- Là hệ thống tập tin phân tán phát triển bởi Google và ra đời trước HDFS. GFS có kiến trúc tương tự HDFS, là hình mẫu để cộng đồng phát triển nên HDFS.
- GFS là nền tảng phát triển các hệ thống khác của Google như BigTable hay Pregel.

Hệ thống lưu trữ HDFS, GFS (tt)

- GFS thường được cấu hình với MasterNode và các shadow Master nhằm mục đích chịu lỗi.
- Trong quá trình hoạt động, nếu MasterNode gặp sự cố, một shadow Master sẽ được lựa chọn thay thế MasterNode. Quá trình này hoàn toàn trong suốt với client và người sử dụng.
- Ngoài ra, trong quá trình lưu trữ các block, GFS sử dụng các kỹ thuật kiểm tra lỗi lưu trữ như checksum nhằm phát hiện và khôi phục block bị lỗi một cách nhanh chóng.

Cơ sở dữ liệu NoSQL

Lịch sử phát triển

- RDBMS (hệ quản trị cơ sở dữ liệu quan hệ) đã thống trị trong nhiều thập kỷ.
- Sự phát triển của dữ liệu lớn và nhu cầu về tính khả mở đã dẫn đến sự ra đời của NoSQL.
- NoSQL ra đời vào năm 2005 như một phản ứng với hạn chế của RDBMS trong môi trường dữ liệu phân tán.

Cơ sở dữ liệu NoSQL (tt)

Đặc điểm chính của NoSQL

- Mô hình dữ liệu linh hoạt
  - NoSQL không sử dụng cấu trúc bảng và quan hệ như trong RDBMS.
  - Dữ liệu được tổ chức theo nhiều mô hình khác nhau, tùy thuộc vào loại NoSQL.

Cơ sở dữ liệu NoSQL (tt)

Đặc điểm chính của NoSQL (tt)

- Mô hình dữ liệu linh hoạt (tt)
  - Các mô hình phổ biến:
    - Key/Value Stores: dữ liệu được lưu trữ dưới dạng các cặp key (khóa) và value (giá trị), đơn giản và hiệu quả cho việc truy xuất dữ liệu nhanh.
    - Document Stores: dữ liệu được lưu trữ dưới dạng các tài liệu (documents) với cấu trúc linh hoạt, cho phép lưu trữ dữ liệu có cấu trúc phức tạp mà không cần tuân theo cấu trúc bảng cố định.
    - Column-family Stores: dữ liệu được lưu trữ theo cột, phù hợp cho việc lưu trữ và truy vấn dữ liệu có cấu trúc phức tạp và linh hoạt.
    - Graph Databases: tối ưu cho việc lưu trữ và truy vấn dữ liệu có quan hệ phức tạp, chẳng hạn như mạng xã hội hoặc các hệ thống đề xuất.

Cơ sở dữ liệu NoSQL (tt)

Đặc điểm chính của NoSQL (tt)

- Khả năng mở rộng
  - Thay vì nâng cấp phần cứng của một máy chủ đơn lẻ (mở rộng theo chiều dọc), NoSQL cho phép thêm nhiều máy chủ phổ thông vào hệ thống để xử lý dữ liệu (mở rộng theo chiều ngang).
  - Lợi ích:
    - Giảm thiểu chi phí nâng cấp phần cứng.
    - Tăng khả năng xử lý dữ liệu lớn và cải thiện hiệu suất truy vấn.

Cơ sở dữ liệu NoSQL (tt)

Đặc điểm chính của NoSQL (tt)

- Tính nhất quán và sẵn sàng cao
  - Dữ liệu sẽ dần dần trở nên nhất quán giữa các bản sao sau một khoảng thời gian nhất định, thay vì đảm bảo tính nhất quán tức thời như trong RDBMS.
    - Cải thiện khả năng mở rộng và hiệu suất trong môi trường phân tán.
    - Giảm thiểu độ trễ khi ghi và đọc dữ liệu.
  - NoSQL được thiết kế để đảm bảo hệ thống vẫn hoạt động ngay cả khi một số nút bị lỗi.

Cơ sở dữ liệu NoSQL (tt)

Đặc điểm chính của NoSQL (tt)

- Hiệu suất cao
  - NoSQL được tối ưu hóa cho việc truy vấn dữ liệu nhanh chóng, đặc biệt trong các ứng dụng yêu cầu xử lý dữ liệu lớn và phức tạp.
  - Không bị ràng buộc bởi cấu trúc dữ liệu cố định, NoSQL linh hoạt trong việc lưu trữ và truy vấn các loại dữ liệu phi cấu trúc hoặc nửa cấu trúc, như dữ liệu JSON, XML. Phù hợp với các ứng dụng hiện đại như phân tích dữ liệu lớn, xử lý dữ liệu từ mạng xã hội,...

Cơ sở dữ liệu NoSQL (tt)

Đặc điểm chính của NoSQL (tt)

- Khả năng chịu lỗi và tự phục hồi
  - NoSQL được thiết kế để xử lý các lỗi phần cứng hoặc mạng một cách tự động mà không làm gián đoạn hoạt động của hệ thống.
  - Hệ thống NoSQL có thể tự động khôi phục dữ liệu bị mất hoặc bị lỗi thông qua cơ chế sao lưu và phục hồi.

Điện toán đám mây với dữ liệu lớn

- Điện toán đám mây cung cấp môi trường linh hoạt, phân tán để lưu trữ và xử lý dữ liệu lớn.
- Dữ liệu lớn yêu cầu cơ sở hạ tầng mạnh mẽ, có khả năng mở rộng, và điện toán đám mây là giải pháp lý tưởng.
- Kho lưu trữ dữ liệu phân tán trong điện toán đám mây là thành phần không thể thiếu, cung cấp khả năng lưu trữ dữ liệu tập trung cho người dùng. Một số kho lưu trữ phổ biến: Openstack Swift, Amazon S3,…

Điện toán đám mây với dữ liệu lớn (tt)

Amazon S3

- Ra mắt vào năm 2006, Amazon S3 là một trong những dịch vụ lưu trữ đám mây đầu tiên và phổ biến nhất.
- Cung cấp giao diện API đơn giản để lưu trữ và truy cập dữ liệu từ bất kỳ đâu có kết nối Internet.
- Thanh toán theo dung lượng lưu trữ và băng thông sử dụng, không cần chi phí cài đặt.

Điện toán đám mây với dữ liệu lớn (tt)

Amazon S3 (tt)

- Các tính năng nổi bật của Amazon S3:
  - Khả năng mở rộng: hỗ trợ lưu trữ một lượng lớn dữ liệu với tính khả mở cao.
  - An toàn: dữ liệu được bảo vệ và có thể truy cập từ mọi nơi.
  - Tính sẵn sàng: được sử dụng rộng rãi bởi các tổ chức lớn như Amazon, NASA,...

Điện toán đám mây với dữ liệu lớn (tt)

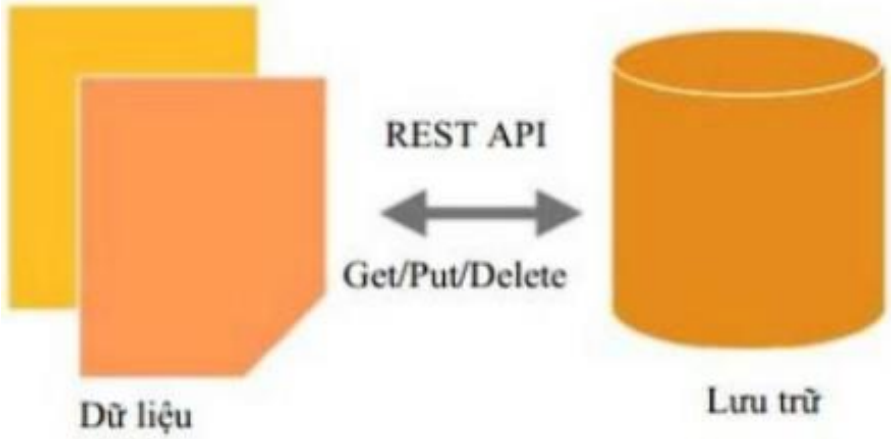
OpenStack Swift

- Là một hệ thống lưu trữ đối tượng được thiết kế để lưu trữ và truy xuất dữ liệu ở quy mô lớn trong môi trường đám mây.
- Được thiết kế để lưu trữ an toàn, khả năng mở rộng, và phân tán các đối tượng dữ liệu như file, hình ảnh, video, và bản sao lưu.

Điện toán đám mây với dữ liệu lớn (tt)

OpenStack Swift (tt)

- Được xây dựng trên một kiến trúc phân tán, giúp loại bỏ các điểm nghẽn và tăng tính sẵn sàng của dữ liệu.
- Hỗ trợ giao tiếp qua HTTP API và tích hợp với các ngôn ngữ lập trình phổ biến như Java, Python, Ruby, C#,...



Điện toán đám mây với dữ liệu lớn (tt)

Dữ liệu lớn trong bối cảnh Điện toán đám mây

- Khái niệm 4V:
  - Volume: khối lượng dữ liệu khổng lồ.
  - Velocity: tốc độ xử lý dữ liệu nhanh và liên tục.
  - Variety: sự đa dạng của các loại dữ liệu (cấu trúc và phi cấu trúc).
  - Value: giá trị quý báu khi khai thác dữ liệu lớn.
- Vai trò của điện toán đám mây: cung cấp nền tảng cơ bản để lưu trữ và xử lý dữ liệu lớn một cách hiệu quả.

Xử lý dữ liệu lớn MapReduce/Hadoop

- MapReduce: một mô hình lập trình và công cụ do Google giới thiệu vào năm 2004, hỗ trợ xử lý lượng dữ liệu lớn trên môi trường phân tán.
- Hadoop: một framework mã nguồn mở được phát triển bởi Apache để hiện thực hóa mô hình MapReduce, cung cấp khả năng mở rộng và xử lý dữ liệu lớn một cách hiệu quả.

Xử lý dữ liệu lớn MapReduce/Hadoop (tt)

Nguyên tắc hoạt động của MapReduce

- Hai bước chính:
  - Map: nhận đầu vào là một cặp khóa/giá trị và xuất ra một tập các cặp khóa/giá trị trung gian.
  - Reduce: tập hợp các cặp khóa/giá trị trung gian có cùng khóa và tính toán trên các cặp này để tạo ra kết quả đầu ra cuối cùng.
- Điểm nổi bật: tính toán được phân tán và thực hiện song song trên nhiều nút, giúp xử lý dữ liệu lớn một cách nhanh chóng.

Xử lý dữ liệu lớn MapReduce/Hadoop (tt)

Ưu điểm của MapReduce

- Xử lý dữ liệu lớn: giải quyết hiệu quả các bài toán liên quan đến phân tích và tính toán phức tạp trên khối lượng dữ liệu lớn.
- Khả năng chịu lỗi cao: khi có lỗi xảy ra tại một nút tính toán, quá trình tính toán sẽ được thực hiện lại mà không ảnh hưởng đến các nút khác.
- Phân tán dữ liệu: thay vì di chuyển dữ liệu tới các nút tính toán, mã chương trình được sao chép tới các nút lưu trữ để thực thi, giúp tiết kiệm băng thông và tài nguyên hệ thống.

Xử lý dữ liệu lớn MapReduce/Hadoop (tt)

Kiến trúc Hadoop MapReduce

- Thành phần:
  - Cluster: bao gồm một nút Master và nhiều nút Worker.
  - Nút Master: quản lý và điều phối các tác vụ Map và Reduce.
  - Nút Worker: thực hiện các tác vụ Map hoặc Reduce dựa trên dữ liệu lưu trữ cục bộ.
- Quy trình thực thi:
  - Mã chương trình được gửi tới các Worker chứa dữ liệu đầu vào.
  - Worker thực hiện tác vụ Map, sinh ra các cặp khóa/giá trị trung gian.
  - Các cặp này được phân phối tới các Worker thực hiện tác vụ Reduce, tạo ra kết quả đầu ra.

Xử lý dữ liệu lớn MapReduce/Hadoop (tt)

Tính toán MapReduce trong thực tiễn

- Môi trường triển khai: triển khai trên cluster gồm hàng trăm tới hàng ngàn máy chủ phổ thông kết nối qua mạng LAN (100 Mbs hoặc 1 Gbs).
- Phần cứng: sử dụng các ổ đĩa cứng thông thường, không đòi hỏi hiệu năng cao.

Xử lý dữ liệu lớn MapReduce/Hadoop (tt)

Tính toán MapReduce trong thực tiễn (tt)

- Cấu trúc Cluster:
  - Nút Master: quản lý và điều tiết các nút Worker.
  - Nút Worker: thực hiện các tác vụ Map và Reduce.
- Luồng công việc:
  - Gói công việc (Job): bao gồm nhiều tác vụ (Tasks) được phân phối tới các Worker.
  - Tác vụ Map: phân tán trên các nút lưu trữ, sinh ra cặp khóa/giá trị trung gian.
  - Tác vụ Reduce: nhóm và xử lý các cặp khóa/giá trị trung gian theo khóa.

Xử lý dữ liệu lớn MapReduce/Hadoop (tt)

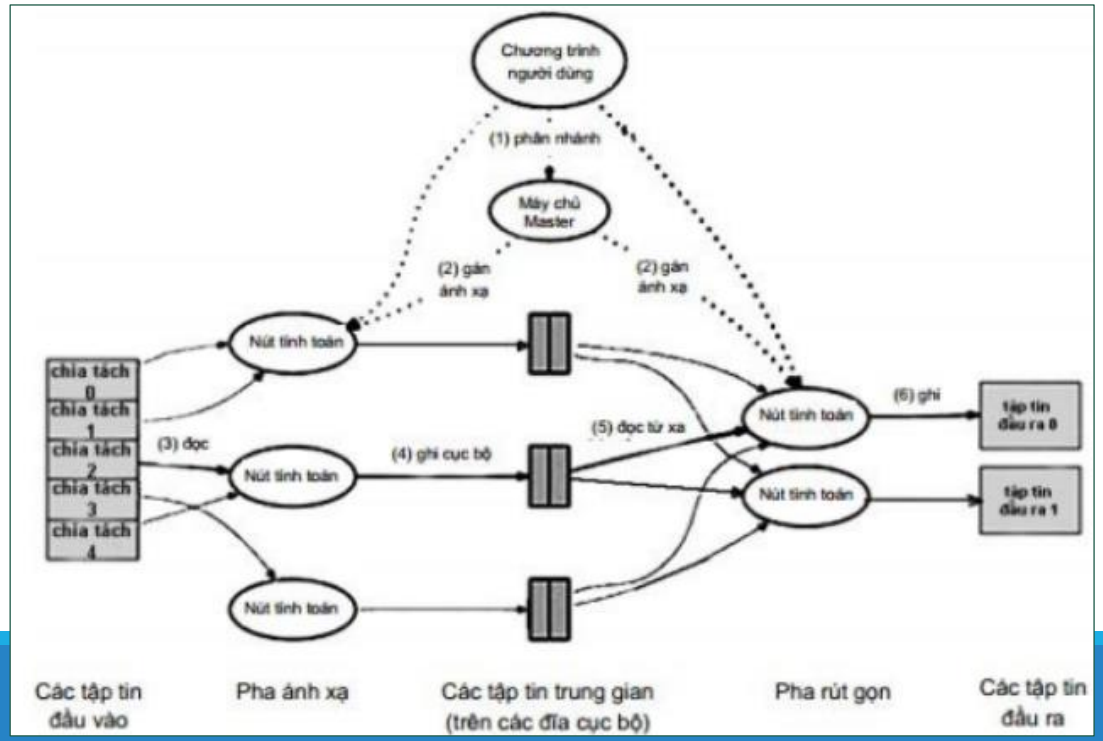
Tính toán MapReduce trong thực tiễn (tt)

- Chi tiết quá trình thực thi
  - Thư viện MapReduce: gửi mã tác vụ Map tới các nút chứa dữ liệu đầu vào.
  - Worker: đọc dữ liệu, thực hiện Map, và lưu trữ cặp khóa/giá trị trung gian.
  - Thông báo vị trí: Master thông báo vị trí của cặp khóa/giá trị trung gian cho các Worker thực hiện Reduce.
  - Tác vụ Reduce: đọc, sắp xếp và tính toán trên dữ liệu, lưu kết quả vào tập tin đầu ra.

Xử lý dữ liệu lớn MapReduce/Hadoop (tt)

Tính toán MapReduce trong thực tiễn (tt)

- Kết thúc tác vụ MapReduce
  - Khi tất cả các tác vụ Map và Reduce kết thúc, Master thông báo kết quả cho chương trình của người sử dụng.



Mô tả quá trình thực thi gói công việc

Xử lý dữ liệu lớn MapReduce/Hadoop (tt)

Ứng dụng của MapReduce/Hadoop

- Phân tích dữ liệu lớn: Hadoop được sử dụng rộng rãi trong các công ty như Yahoo, Facebook, Amazon để phân tích và xử lý khối lượng dữ liệu khổng lồ.
- Xử lý log: phân tích log server, mạng xã hội, và các dữ liệu từ cảm biến.
- Tìm kiếm và sắp xếp: sử dụng trong các công cụ tìm kiếm và xếp hạng dữ liệu.