



MÔ HÌNH PHÂN TÍCH DỮ LIỆU LỚN

Giảng viên: Nguyễn Tu Trung, Trần Mạnh Tuấn
BM HTTT, Khoa CNTT, Trường ĐH Thủy Lợi

Hà Nội, 2019

Nội dung

- ❖ Giới thiệu Framework Hadoop
- ❖ Hadoop Distributed File System (HDFS)
- ❖ Mô hình Map/Reduce
- ❖ Lập trình MapReduce trên Hadoop

Giới thiệu Framework Hadoop

- ❖ Hadoop là gì?
- ❖ Lịch sử Hadoop
- ❖ Các thành phần của Hadoop
- ❖ Ứng dụng của Hadoop trong một số công ty lớn
- ❖ Nguyên tắc hoạt động của Hadoop MapReduce
- ❖ Các trình nền của Hadoop

Hadoop là gì?

❖ Theo Apache Hadoop:

- ❖ «Apache Hadoop là một framework dùng để chạy những ứng dụng trên 1 cluster lớn được xây dựng trên những phần cứng thông thường 1. Hadoop hiện thực mô hình Map/Reduce, đây là mô hình mà ứng dụng sẽ được chia nhỏ ra thành nhiều phân đoạn khác nhau, và các phần này sẽ được chạy song song trên nhiều node khác nhau. Thêm vào đó, Hadoop cung cấp 1 hệ thống file phân tán (HDFS) cho phép lưu trữ dữ liệu lên trên nhiều node. Cả Map/Reduce và HDFS đều được thiết kế sao cho framework sẽ tự động quản lý được các lỗi, các hư hỏng về phần cứng của các node.»

Hadoop là gì?

❖ Theo Wikipedia:

- ❖ «Hadoop là một framework nguồn mở viết bằng Java cho phép phát triển các ứng dụng phân tán có cường độ dữ liệu lớn một cách miễn phí. Nó cho phép các ứng dụng có thể làm việc với hàng ngàn node khác nhau và hàng petabyte dữ liệu. Hadoop lấy được phát triển dựa trên ý tưởng từ các công bố của Google về mô hình MapReduce và hệ thống file phân tán Google File System (GFS).»

Hadoop là gì?

❖ Kết luận về Hadoop:

- ❖ Là một framework cho phép phát triển các ứng dụng phân tán
- ❖ Viết bằng Java
- ❖ 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. HDFS che giấu tất cả các thành phần phân tán, các nhà phát triển ứng dụng phân tán sẽ chỉ nhìn thấy HDFS như một hệ thống file cục bộ bình thường
- ❖ Giúp các nhà phát triển ứng dụng phân tán tập trung tối đa vào phần logic của ứng dụng, bỏ qua được một số phần chi tiết kỹ thuật phân tán bên dưới (phần này do Hadoop tự động quản lý)

Lịch sử Hadoop

- ❖ Hadoop được tạo ra bởi Doug Cutting, người sáng tạo ra Apache Lucene – bộ thư viện tạo chỉ mục tìm kiếm trên text được sử dụng rộng rãi
- ❖ Hadoop bắt nguồn từ Nutch, một ứng dụng search engine nguồn mở
- ❖ Nutch, một ứng dụng search engine nguồn mở, được khởi xướng từ năm 2002, gồm crawler và tìm kiếm
- ❖ Nutch chỉ có thể crawl tối đa 100 triệu trang => không thể mở rộng ra để có thể thực hiện vai trò searcher engine của mình trên tập dữ liệu hàng tỷ trang web
- ❖ Nguyên nhân: Nutch lúc này chỉ chạy trên một máy đơn (stand alone) nên gặp phải các khuyết điểm: Khả năng lưu trữ bị giới hạn (ứng với 1 máy), Tốc độ truy xuất chậm (ứng với 1 máy)

Lịch sử Hadoop

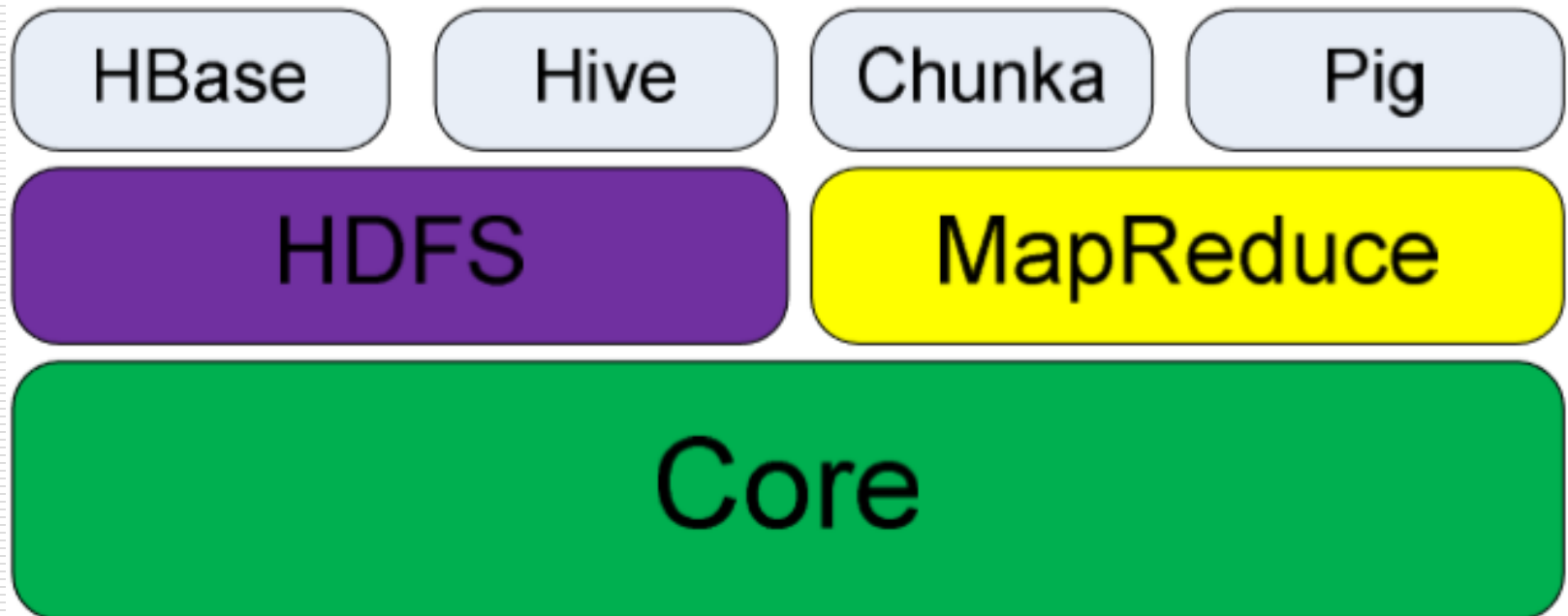
- ❖ Năm 2003, Google công bố kiến trúc của hệ thống file phân tán GFS (viết tắt từ Google File System) => Các nhà kiến trúc sư của Nutch thấy rằng GFS sẽ giải quyết được nhu cầu lưu trữ các file rất lớn từ quá trình crawl và index
- ❖ Năm 2004, họ bắt tay vào việc ứng dụng kiến trúc của GFS vào cài đặt một hệ thống file phân tán nguồn mở có tên Nutch Distributed File System (NDFS)
- ❖ Năm 2004, Google lại công bố bài báo giới thiệu MapReduce
- ❖ Đầu năm 2005, các nhà phát triển Nutch xây dựng được phiên bản MapReduce trên Nutch
- ❖ Giữa năm 2005, tất cả các thuật toán chính của Nutch đều được cải tiến lại để chạy trên nền NDFS và MapReduce
- ❖ Tháng hai 2006 Doug Cutting đã tách riêng NDFS và MapReduce ra để hình thành một dự án độc lập có tên Hadoop. Cùng thời gian này, Doug Cutting gia nhập vào Yahoo!.

Lịch sử Hadoop

- ❖ Tháng 2 năm 2008 Yahoo đã công bố sản phẩm search engine của họ được xây dựng trên một Hadoop cluster có kích thước 10.000 nhân vi xử lý
- ❖ Năm 2008, Apache đã đưa Hadoop lên thành dự án ở top-level Apache Software Foundation, nhằm xác nhận sự thành công và các áp dụng rộng rãi của Hadoop, thậm chí nhiều công ty ngoài Yahoo! như Last.fm, Facebook, New York Times
- ❖ Năm 2008, Hadoop đã phá kỷ lục thế giới về sắp xếp một terabyte dữ liệu: Chạy trên một cluster gồm 910 node, Hadoop đã sắp xếp một terabyte dữ liệu trong vòng 209 giây, phá kỷ lục cũ là 297 giây
- ❖ Sau đó, Google công bố ứng dụng chạy trên MapReduce của họ đã sắp xếp được một terabyte dữ liệu trong 68 giây
- ❖ Tháng 5 năm 2009, một đội các nhà phát triển của Yahoo! đã dùng Hadoop để sắp xếp một terabyte dữ liệu trong vòng 62 giây

Các thành phần của Hadoop

- ❖ Hiện nay, ngoài NDFS (đã được đổi tên lại thành HDFS – Hadoop Distributed File System) và MapReduce, đội ngũ phát triển Hadoop đã phát triển các dự án con dựa trên HDFS và MapReduce
- ❖ Hiện tại, Hadoop gồm có các dự án con sau:



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
- ❖ Là phần lõi để xây dựng nên HDFS và 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
- ❖ Có thể được sử dụng để chạy trên một cluster lớn với hàng chục ngàn node

Các thành phần của Hadoop

❖ HBase:

- ❖ Một cơ sở dữ liệu phân tán, theo hướng cột (column-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
- ❖ Quản lý dữ liệu được lưu trữ trên HDFS và cung cấp một ngôn ngữ truy vấn dựa trên SQL

❖ Pig:

- ❖ Ngôn ngữ luồng dữ liệu cấp cao và framework thực thi dùng cho tính toán song song

Các thành phần của Hadoop

- ❖ Chukwa:
 - ❖ Một hệ thống tập hợp và phân tích dữ liệu
 - ❖ 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
- ❖ Tập trung vào hai phần quan trọng nhất của Hadoop
 - ❖ HDFS
 - ❖ MapReduce

Ứng dụng của Hadoop trong một số công ty

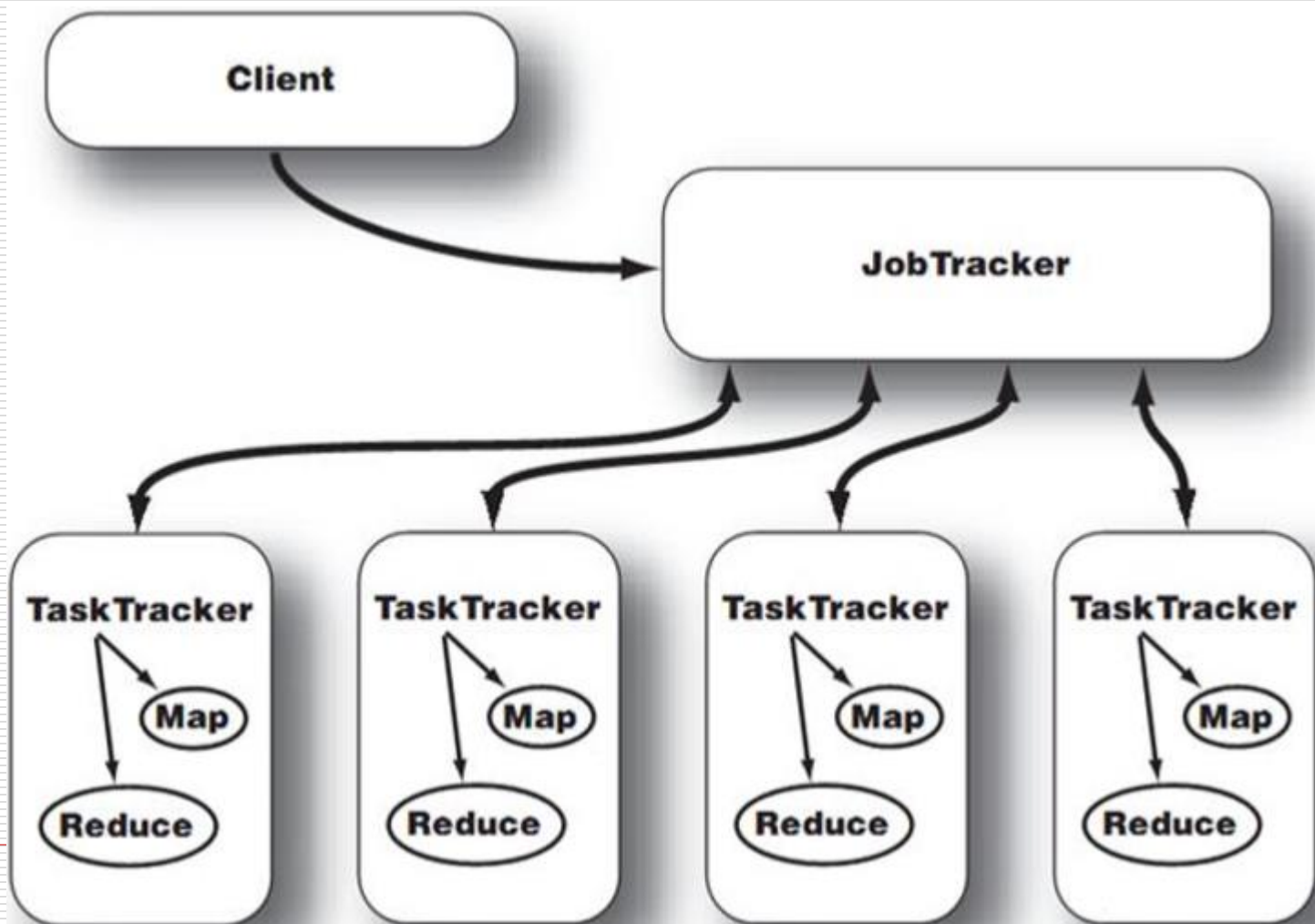
- ❖ Ngoài Yahoo!, nhiều công ty sử dụng Hadoop như công cụ để lưu trữ và phân tích dữ liệu trên các khối dữ liệu lớn
- ❖ Twitter: sử dụng Hadoop để xử lý tweets (các bài viết văn bản lên đến 140 ký tự hiển thị trên profile của tác giả), logs và các nguồn dữ liệu phát sinh trong quá trình hoạt động của Twitter
- ❖ Facebook: Sử dụng Hadoop để lưu trữ các log nội bộ và kích thước của nguồn dữ liệu
 - ❖ Các dữ liệu này được dùng làm nguồn cho các báo cáo phân tích và máy học
 - ❖ Facebook có 2 Hadoop cluster chính: một cluster 1100 máy với 8800 nhân và 12 Petabyte ổ cứng lưu trữ
- ❖ A9.com – Amazon: Sử dụng Hadoop để đánh giá chỉ số tìm kiếm sản phẩm trên Amazon
 - ❖ Xử lý đến hàng triệu Session mỗi ngày
 - ❖ Các cluster của A9.com có độ lớn từ 1-100 node
- ❖ Và còn rất nhiều công ty khác

Nguyên tắc hoạt động của Hadoop

- ❖ Chia dữ liệu đầu vào thành các mảnh (piece) có kích thước cố định gọi là các input split hoặc là các split => Có rất nhiều các split
- ❖ Tạo ra một task map cho mỗi split => điều này có nghĩa là thời gian xử lý mỗi split (thời gian một task map) nhỏ hơn so với thời gian xử lý toàn bộ đầu vào
- ❖ Xử lý các split một cách song song trên một hoặc nhiều máy tính (cụm máy tính)
- ❖ Khi “chạy Hadoop” có nghĩa là chạy một tập các trình nền – daemon, bao gồm: NameNode, DataNode, Secondary NameNode, JobTracker, TaskTracker
- ❖ Các trình nền có vai trò cụ thể, tồn tại trên một máy chủ hoặc nhiều máy chủ

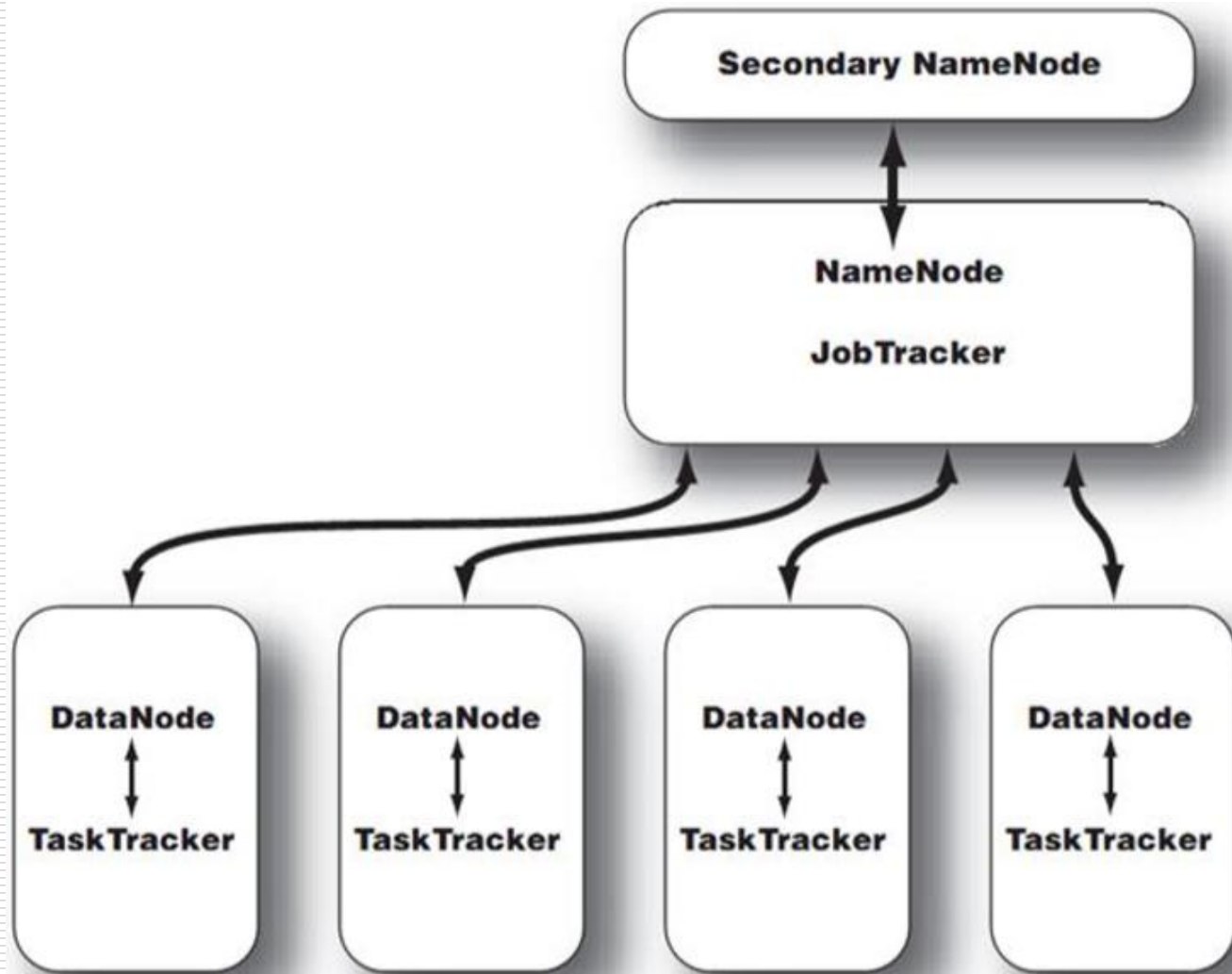
Nguyên tắc hoạt động của Hadoop

- ❖ Client gọi JobTracker bắt đầu công việc xử lý dữ liệu, JobTracker làm việc và giao các nhiệm vụ cho mỗi TaskTracker trong cluster



Nguyên tắc hoạt động của Hadoop

- ❖ Trong mỗi cluster, kiến trúc của Hadoop là master-slave (chủ-tớ): NameNode và JobTracker là Master và DataNode & TaskTracker là slave



Các trình nền của Hadoop

- ❖ NameNode
- ❖ DataNode
- ❖ Secondary NameNode
- ❖ JobTracker
- ❖ TaskTracker

NameNode

- ❖ Là một trình nền quan trọng nhất của Hadoop
- ❖ Hadoop sử dụng một kiến trúc master/slave cho cả lưu trữ phân tán và xử lý phân tán
- ❖ Hệ thống lưu trữ phân tán là Hadoop File System (HDFS)
- ❖ NameNode là master của HDFS để chỉ đạo các trình nền DataNode slave để thực hiện các nhiệm vụ I/O mức thấp
- ❖ Đóng vai trò là master của hệ thống HDFS
- ❖ Quản lý các meta-data của hệ thống HDFS như file system space, danh sách các file trên hệ thống và các block id tương ứng của từng file
- ❖ Quản danh sách slave và tình trạng hoạt động của các DataNode (live hay dead)
- ❖ Điều hướng quá trình đọc/ghi dữ liệu từ client lên các DataNode

DataNode

- ❖ Mỗi máy slave trong cluster lưu trữ một trình nền DataNode để thực hiện các công việc nào đó của hệ thống file phân tán: **đọc và ghi các khối HDFS tới các file thực tế trên hệ thống file cục bộ (local filesystem)**
- ❖ Chứa các block dữ liệu thực sự của các file trên HDFS
- ❖ Đáp ứng các yêu cầu tạo/xoá các block dữ liệu từ NameNode
- ❖ Khi đọc hay ghi một file HDFS
 - ❖ File đó được chia nhỏ thành các khối
 - ❖ NameNode sẽ nói cho client nơi các khối nằm trong trình nền DataNode nào
 - ❖ Client liên lạc trực tiếp với các trình nền DataNode để xử lý các file cục bộ tương ứng với các block
- ❖ Một DataNode có thể giao tiếp với các DataNode khác để nhân bản các khối dữ liệu của nó để dự phòng

Secondary NameNode

- ❖ Các Secondary NameNode (SNN) là một trình nền hỗ trợ giám sát trạng thái của các cụm HDFS
- ❖ Giống như NameNode, mỗi cụm có một SNN, và nó thường trú trên một máy của mình
- ❖ Không có các trình nền DataNode hay TaskTracker chạy trên cùng một server
- ❖ SNN khác với NameNode:
 - ❖ Không nhận hoặc ghi lại bất cứ thay đổi thời gian thực tới HDFS
 - ❖ Giao tiếp với các NameNode bằng cách chụp những bức ảnh của siêu dữ liệu HDFS (HDFS metadata) tại những khoảng xác định bởi cấu hình của các cluster
- ❖ Nhiệm vụ: duy trì một bản sao của meta-data trên NameNode và bản sao này sẽ được dùng để phục hồi lại NameNode nếu NameNode bị hư hỏng

JobTracker

- ❖ Quản lý việc thực thi ứng dụng
 - ❖ Tiếp nhận các yêu cầu thực thi các MapReduce job
 - ❖ Phân chia job này thành các task và phân công cho các TaskTracker thực hiện
 - ❖ Quản lý tình trạng thực hiện các task của TaskTracker và phân công lại nếu cần
- ❖ Nếu một nhiệm vụ (task) thất bại (fail), JobTracker sẽ tự động chạy lại nhiệm vụ đó, có thể trên một node khác, cho đến một giới hạn nào đó được định sẵn của việc thử lại này (ví dụ: Số lần thử lại tối đa được xác định trước)
- ❖ Quản lý danh sách các node TaskTracker và tình trạng của từng node
- ❖ Chỉ có một JobTracker trên một cụm Hadoop, thường chạy trên một máy chủ như là một nút master của cluster

TaskTracker

- ❖ JobTracker và TaskTracker chịu trách nhiệm duy trì bộ máy MapReduce
- ❖ Tuân theo kiến trúc master/slave:
 - ❖ JobTracker là giám sát tổng việc thực hiện chung của một công việc MapReduce
 - ❖ Các taskTracker quản lý việc thực hiện các nhiệm vụ riêng trên mỗi node slave
- ❖ Nhận các task từ JobTracker và thực hiện task
- ❖ Một trong những trách nhiệm của các TaskTracker là liên tục liên lạc với JobTracker
 - ❖ Nếu JobTracker không nhận được liên lạc từ một TaskTracker nào đó trong vòng khoảng thời gian xác định, nó sẽ cho rằng TaskTracker đã bị treo (cached) và sẽ gửi lại nhiệm vụ tương ứng cho các TaskTracker khác trong cluster