

**SAMSUNG**

# Samsung Innovation Campus

| Khóa học Big Data

Together for Tomorrow!  
**Enabling People**

Education for Future Generations

Chương 2.

# Những nguyên tắc cơ bản của Big Data

Khóa học Big Data

# Mô tả chương

---

## ● Mục tiêu:

- ✓ Chúng ta sẽ tìm hiểu cách Hadoop và các công cụ Hệ sinh thái của nó kết hợp với nhau để giải quyết thách thức Big Data
  - Hadoop giải quyết vấn đề lưu trữ và xử lý lượng dữ liệu khổng lồ theo cách tiết kiệm chi phí, có thể mở rộng và chịu lỗi thông qua sự chuyển đổi mô hình từ phụ thuộc phần cứng sang phụ thuộc phần mềm.
  - Về cốt lõi, Hadoop cung cấp HDFS để lưu trữ dữ liệu và Yarn/MapReduce để xử lý dữ liệu.
  - Hệ sinh thái Hadoop cung cấp một bộ công cụ phong phú để nhập dữ liệu từ nhiều nguồn dữ liệu khác nhau, xử lý trước và chuyển đổi dữ liệu để chuẩn bị cho việc truy vấn, thực hiện cả truy vấn hàng loạt và tương tác trên hàng petabyte dữ liệu, tạo mô hình AI và suy luận câu trả lời, và cuối cùng là tạo trực quan hóa và bảng điều khiển trực quan và thân thiện với người dùng để giúp hiểu kết quả.
- ✓ Big Data không ngừng phát triển. Chúng ta sẽ khám phá Dịch vụ đám mây đang đóng vai trò quan trọng như thế nào và cách ngành đang áp dụng Đám mây như một giải pháp thay thế cho Hadoop.

## ● Nội dung của chương

1. Xử lý Bì Data
2. Tổng quan về hệ thống Hadoop Core & Eco
3. Kiến trúc Hadoop cho Big Data

Bài 1.

# Xử lý Big Data

Những nguyên tắc cơ bản Big Data

Bài 1.

# Xử lý Big Data

| 1.1. Ứng dụng và xử lý Big Data

| 1.2. Big Data trên Đám mây công cộng

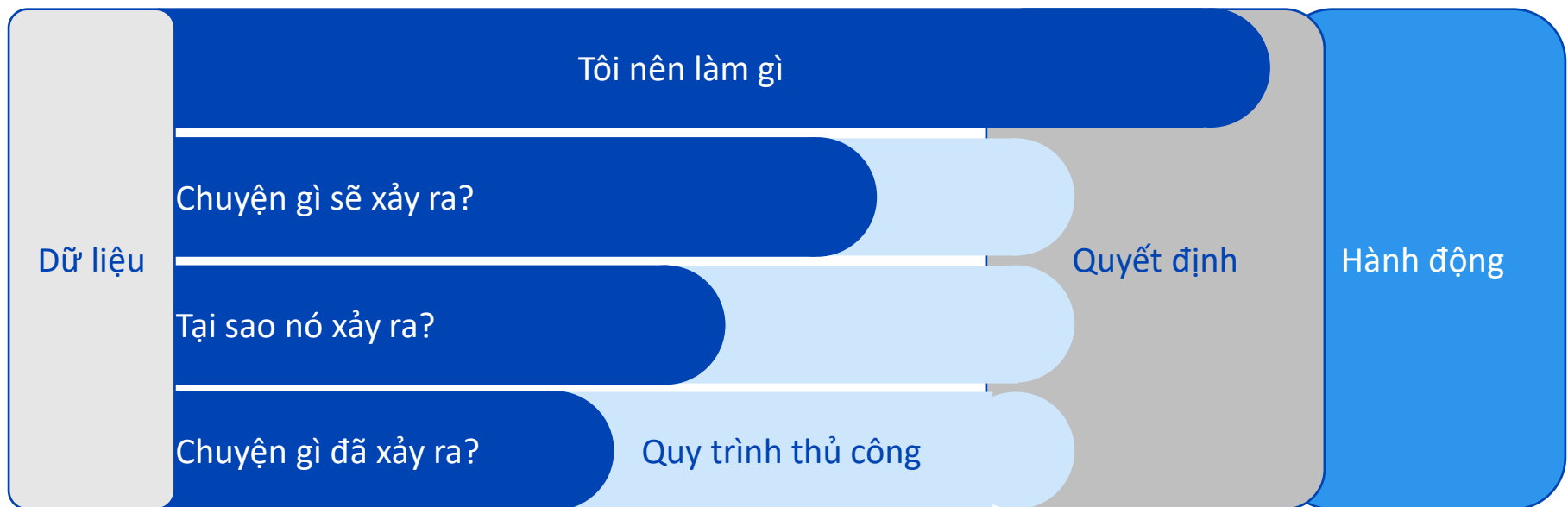
# Mục tiêu của xử lý và phân tích Big Data

I So sánh Predictive Analytics (Phân tích dự đoán) và Prescriptive Analytics (Phân tích theo quy định)

Phân tích dự đoán	Phân tích theo quy định
<ul style="list-style-type: none"><li>✓ Học hỏi từ quá khứ</li><li>✓ Xem xét dữ liệu trong quá khứ và thiết kế các mô hình thống kê hoặc mô hình máy học để dự đoán tương lai</li></ul>	<ul style="list-style-type: none"><li>✓ Thực hiện các hành động dựa trên quá khứ</li><li>✓ Đưa những dự đoán về tương lai và biến chúng thành hành động hoặc chính sách</li></ul>

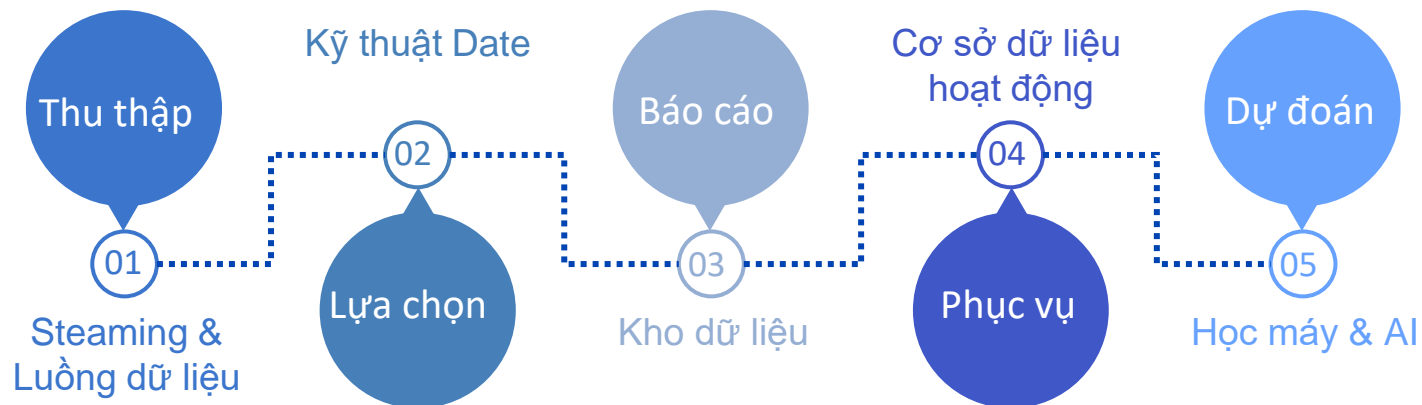
# Mục tiêu của xử lý và phân tích Big Data

I Từ Phân tích dự đoán đến Phân tích theo quy định



# Đường ống Big Data

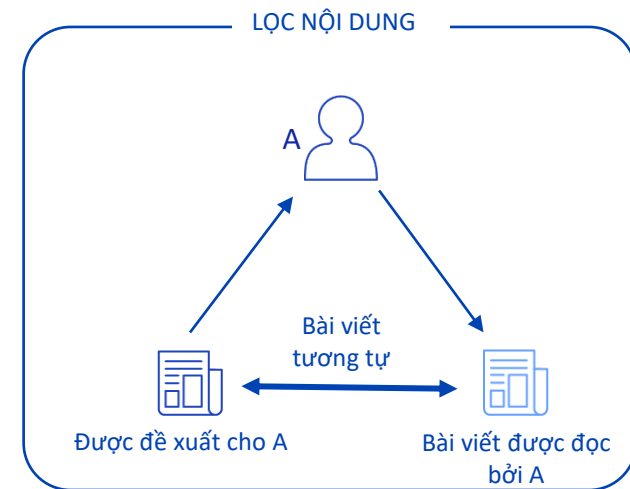
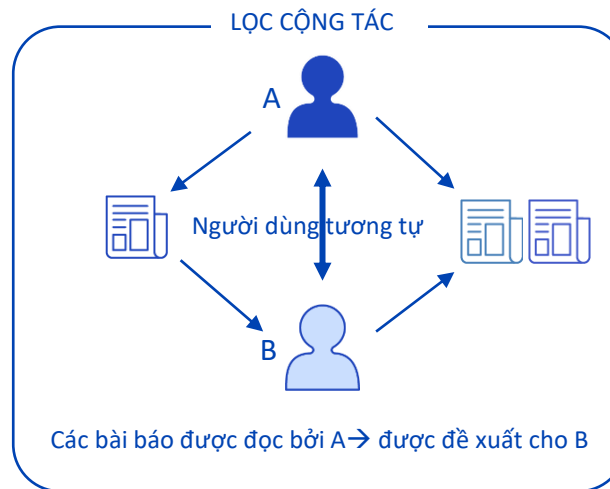
- | Các dịch vụ được tùy chỉnh cho các bước cụ thể trong vòng đời dữ liệu
- | Nhấn mạnh năng suất và tính dễ sử dụng
- | Tự động thay đổi quy mô tài nguyên điện toán để phù hợp với nhu cầu thay đổi
- | Cô lập tài nguyên điện toán để duy trì hiệu suất khối lượng công việc





# Trường hợp sử dụng Big Data - Đề xuất

- ▮ Dự đoán sở thích của người dùng và đưa ra các đề xuất tùy chỉnh
- ▮ Có tính ứng dụng cao trong nhiều ngành
  - mua sắm trực tuyến
  - Dịch vụ phát trực tuyến
- ▮ Collaborative Filtering (Lọc cộng tác) và Content Filtering (Lọc nội dung) là các kỹ thuật chính được triển khai
- ▮ Các công ty áp dụng
  - Netflix
  - Amazon



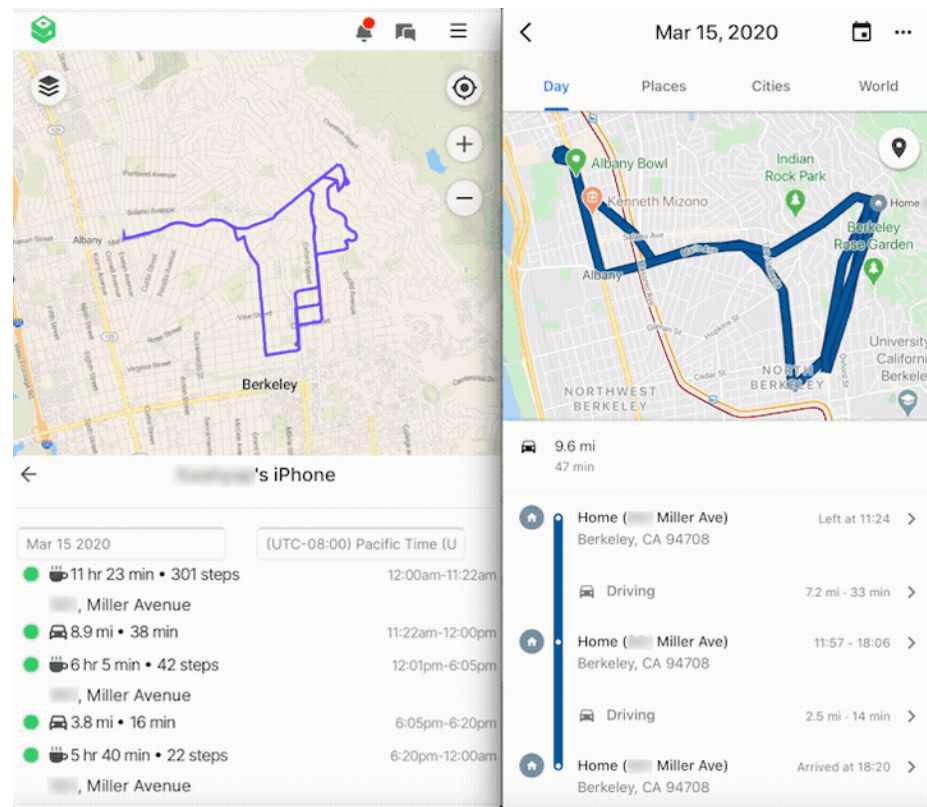
# Trường hợp sử dụng Big Data - Dịch vụ tùy chỉnh

I Cung cấp dịch vụ tùy chỉnh dựa trên góc nhìn 360 độ của khách hàng

- ▶ Vị trí
- ▶ Giới tính
- ▶ Sự giàu có
- ▶ Sở thích

I Các Dịch vụ áp Dụng

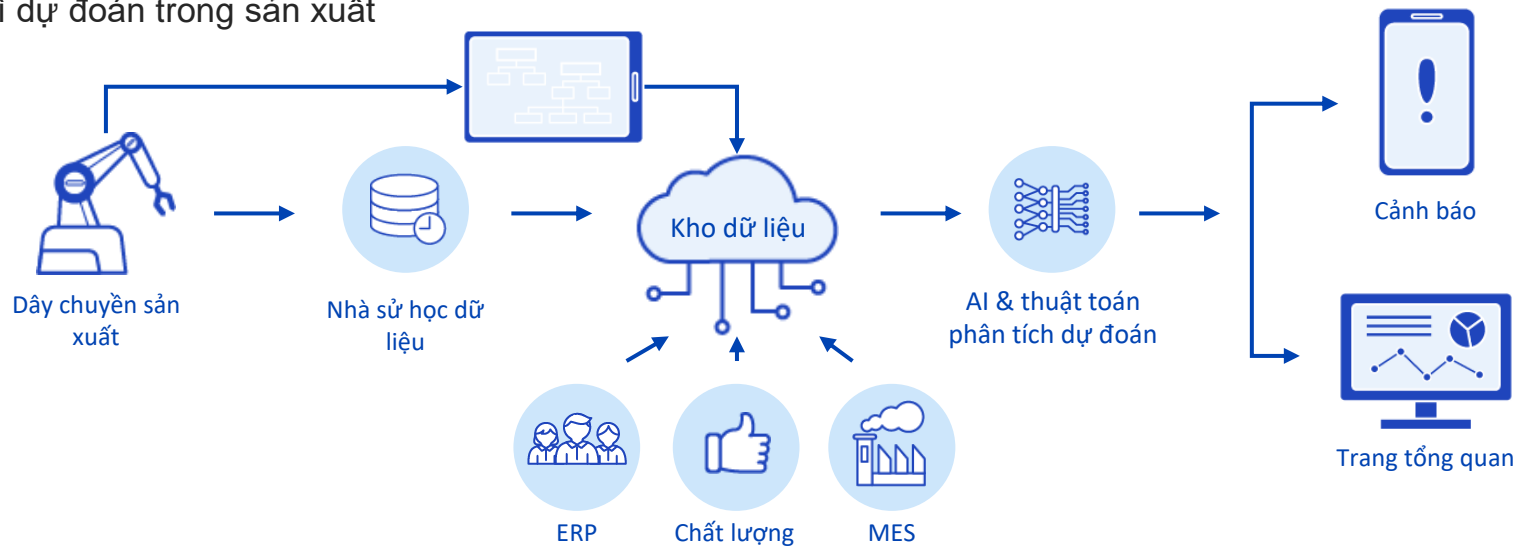
- ▶ Dòng thời gian của Google Maps
- ▶ Quảng cáo Google và YouTube



<https://hypertrack.com/blog/content/images/2020/04/Comaprison.gif>

# Trường hợp sử dụng Big Data - Dự đoán

- I Dự đoán lợi nhuận hoặc xu hướng trong tương lai bằng cách xác định các mẫu cụ thể từ dữ liệu tích lũy có liên quan
- I Được sử dụng trong các ngành công nghiệp khác nhau
  - ▶ Phát triển sản phẩm
  - ▶ Quản lý rủi ro và bảo mật
  - ▶ Bảo trì dự đoán trong sản xuất



# Trường hợp sử dụng Big Data - Phân tích liên kết

### I Sự kết hợp

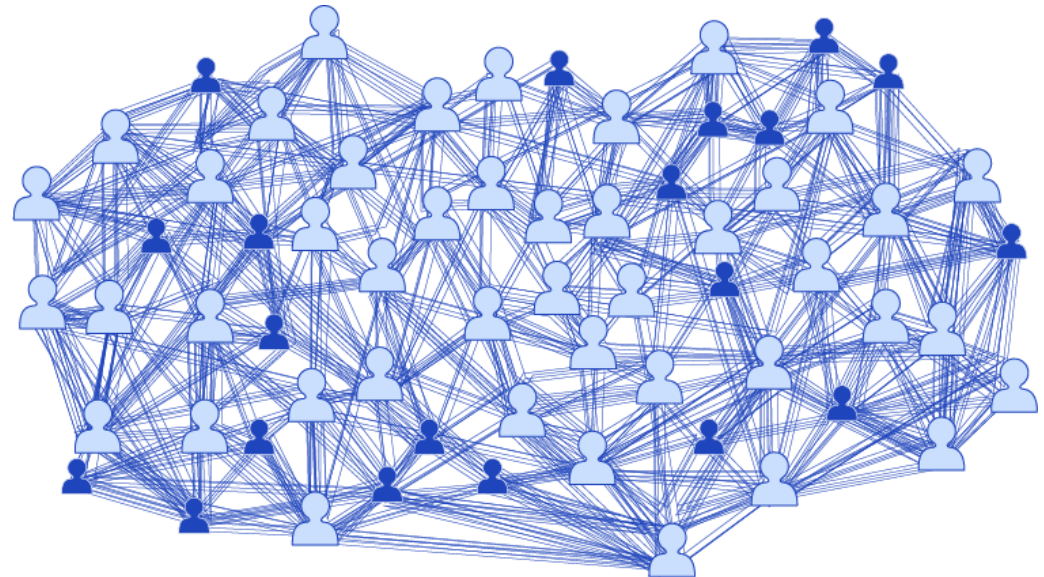
- ▶ Tạo các mô hình học máy để phân tích các sự kiện xảy ra cùng nhau
- ▶ Walmart - Khách hàng mua tã trẻ em vào cuối tuần cũng có xu hướng mua bia
- ▶ Đầu tư - Các nhà đầu tư mua cổ phiếu ở A và B cũng có xu hướng mua C

### I Phân tích đồ thị

- ▶ Tìm kết nối giữa các nút và cạnh
- ▶ Các nút và người dùng, các cạnh là quan hệ

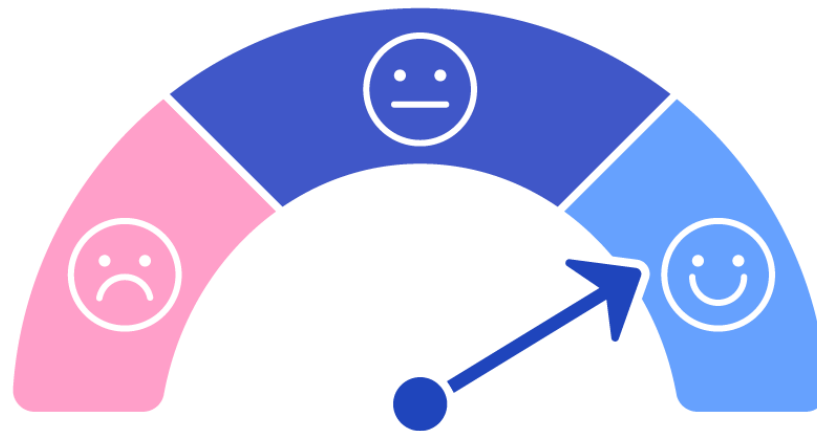
### I Các ngành áp dụng

- ▶ Mua sắm trực tuyến và ngoại tuyến
- ▶ Dịch vụ mạng xã hội



# Trường hợp sử dụng Big Data - Phân tích tình cảm

- | Xử lý ngôn ngữ tự nhiên
- | Voice of Customer (VoC): Tiếng nói của khách hàng
  - ▶ Phân tích VoC có thể được lưu trữ ở nhiều định dạng khác nhau như văn bản, âm thanh, v.v.
  - ▶ Phân tích ấn tượng chủ quan, cảm xúc, thái độ và ý kiến cá nhân về một chủ đề từ dữ liệu phi cấu trúc như SNS và đánh giá sản phẩm
- | Các trường hợp sử dụng có thể áp dụng
  - ▶ Marketing
  - ▶ Dịch vụ khách hàng
  - ▶ Dịch vụ lâm sàng



# Trường hợp sử dụng Big Data – Phân loại / phân cụm

### I Phân loại

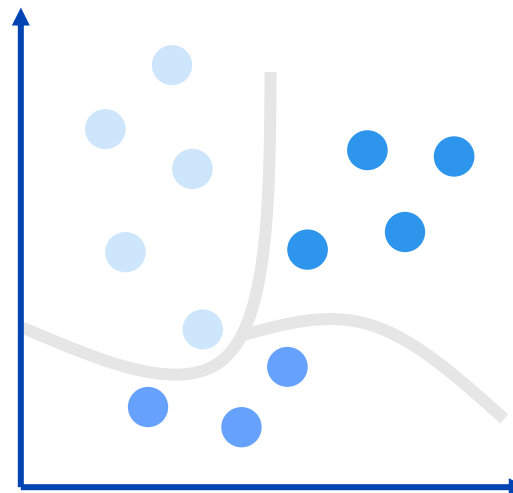
- ▶ Phân loại một tập dữ liệu nhất định thành một số danh mục được xác định trước

### I Phân cụm

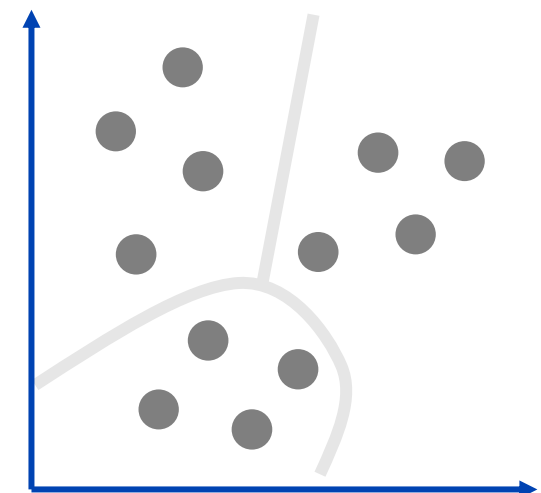
- ▶ Xác định các đặc điểm phân phối dựa trên sự giống nhau của dữ liệu đầu vào và chia nó thành nhiều nhóm tùy ý.

### I Các trường hợp sử dụng được áp dụng

- ▶ Xử phạt nội dung không phù hợp bằng AI của Facebook
- ▶ Công ty dược phẩm



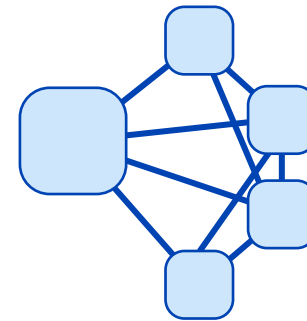
Phân loại (Dán nhãn)



Phân cụm  
(không dán nhãn)

# MapReduce: Công cụ tính toán Big Data cốt lõi

- | MapReduce làm mô hình lập trình cho Dữ liệu lớn
- | Xử lý dữ liệu theo định hướng bản ghi (khóa và giá trị)
- | Bao gồm hai giai đoạn do nhà phát triển tạo ra
  - ▶ Bản đồ (Map)
  - ▶ Giảm bớt (Reduce)
- | Ở giữa Map và Reduce là xáo trộn và sắp xếp
  - ▶ Gửi dữ liệu từ Mapper đến Reducers



# Các tính năng của MapReduce

- | MapReduce là một phương pháp xử lý song song phân tán.
  - ▶ Phân phối công việc trên nhiều nút phụ
- | Chạy trên các nút có sẵn dữ liệu
- | Một môi trường được cung cấp để các nhà phát triển có thể tập trung vào mã chương trình của họ
  - ▶ Phân phối dữ liệu tự động
  - ▶ Có thể xử lý dữ liệu cục bộ
  - ▶ Di chuyển dữ liệu theo yêu cầu
  - ▶ Các ngôn ngữ được hỗ trợ - Java và các ngôn ngữ khác



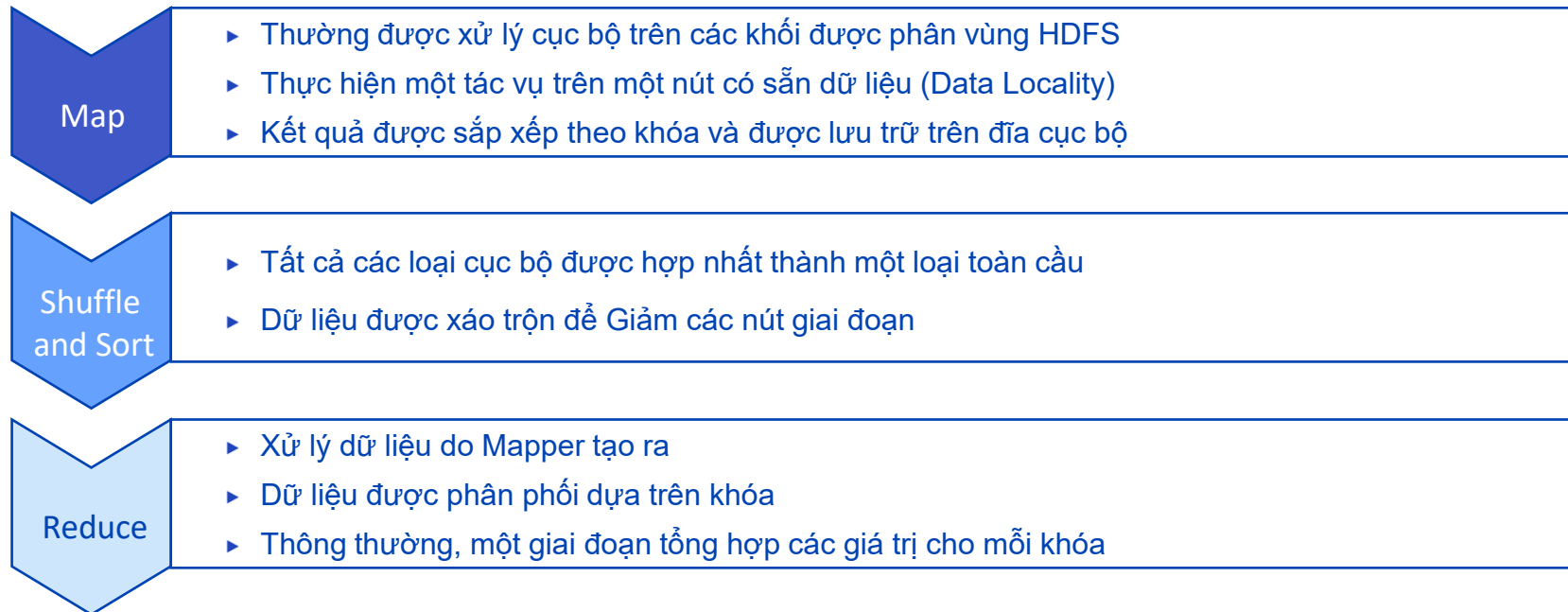


# Ưu điểm của MapReduce

- | Tự động song song hóa và phân phối
- | Khả năng chịu lỗi
- | Công cụ giám sát và trạng thái
- | Một sự trừu tượng rõ ràng cho các lập trình viên
- | MapReduce trừu tượng hóa tất cả công việc 'dọn phòng' khỏi lập trình viên

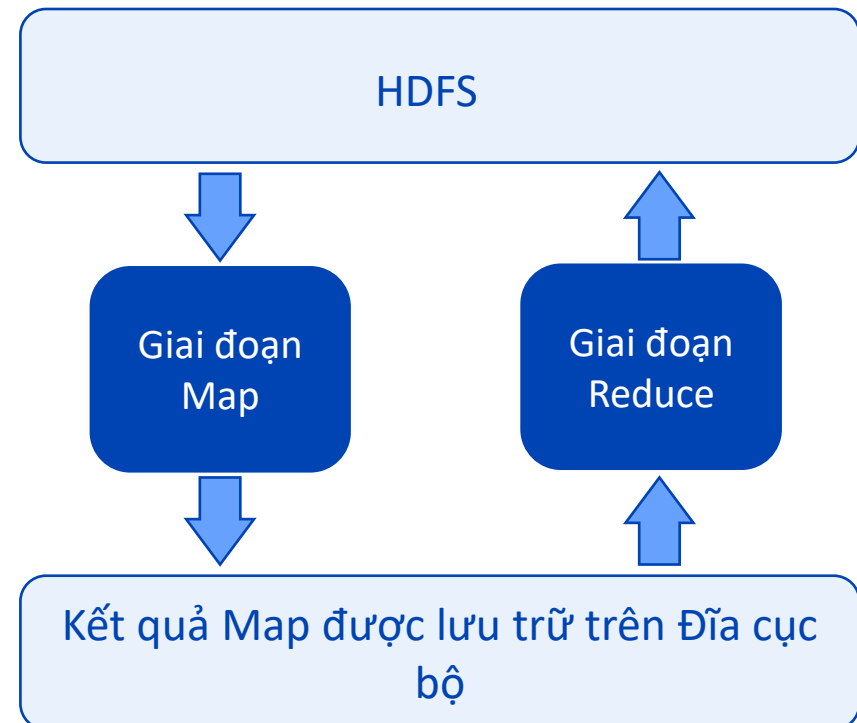


# Ba giai đoạn của MapReduce



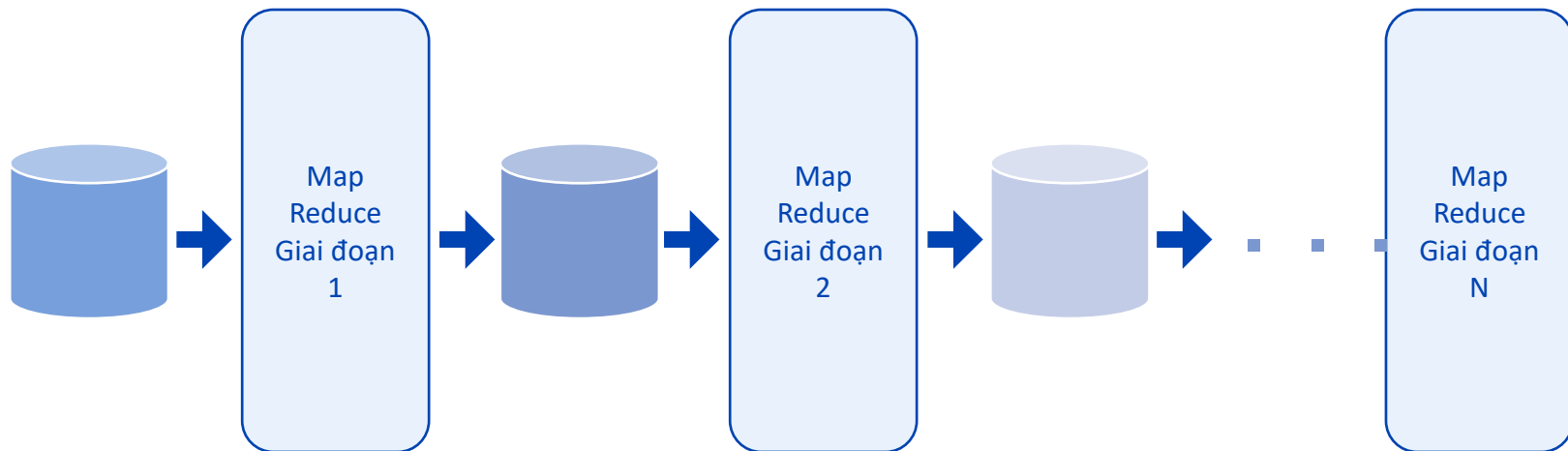
# Mẫu truy cập đĩa MapReduce

- | Mappers đọc dữ liệu từ đĩa và thực hiện tính toán
- | Sau khi tất cả những người lập bản đồ đã hoàn thành, dữ liệu được xáo trộn, sắp xếp và ghi vào đĩa
- | Các Reducers đọc dữ liệu, thực hiện rút gọn và ghi kết quả vào đĩa



# Nhiều chu kỳ MapReduce

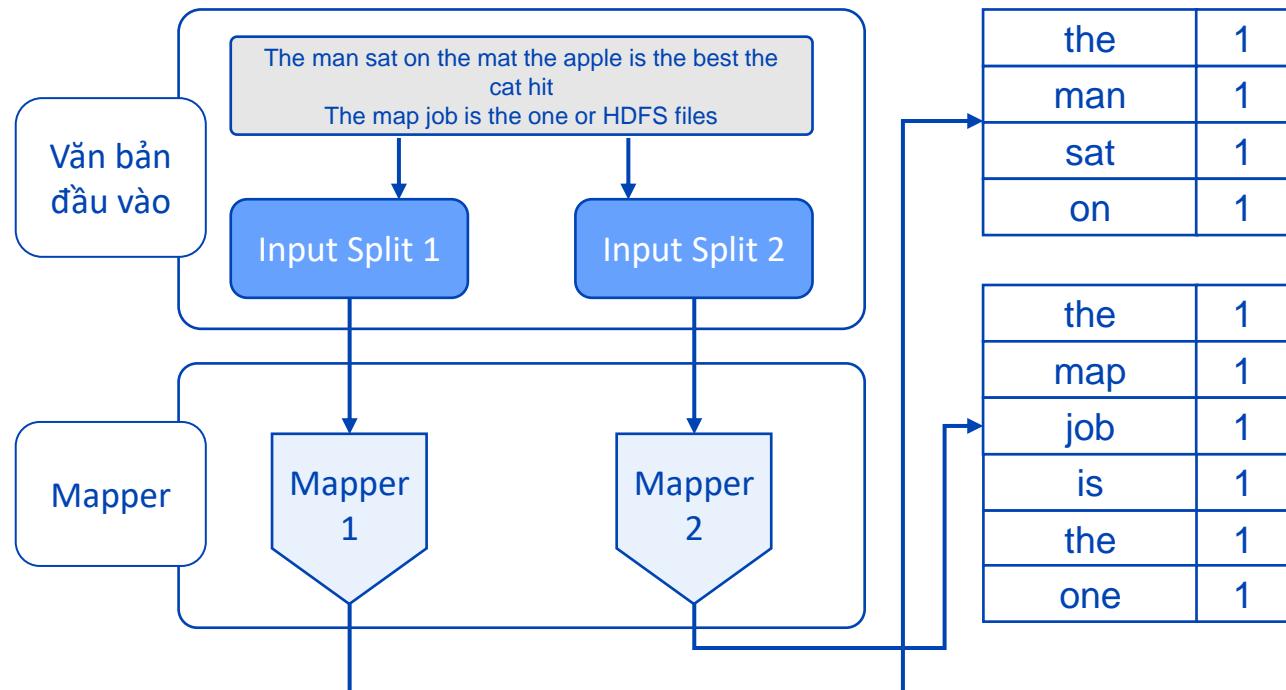
- | Thông thường để có được bất kỳ kết quả thú vị nào, một công việc phải trải qua nhiều chu kỳ Map và Reduce
- | Trong mỗi giai đoạn thu nhỏ Map, chúng tôi đã thấy đĩa đọc và ghi giữa trình mapper và trình reducer
- | Ở giữa các giai đoạn, mỗi kết quả ngay lập tức được ghi vào đĩa và đọc ghi lại ở giai đoạn tiếp theo
- | Đĩa I/O rất Đắt



# MapReduce - Word Count

## I Giai đoạn Map

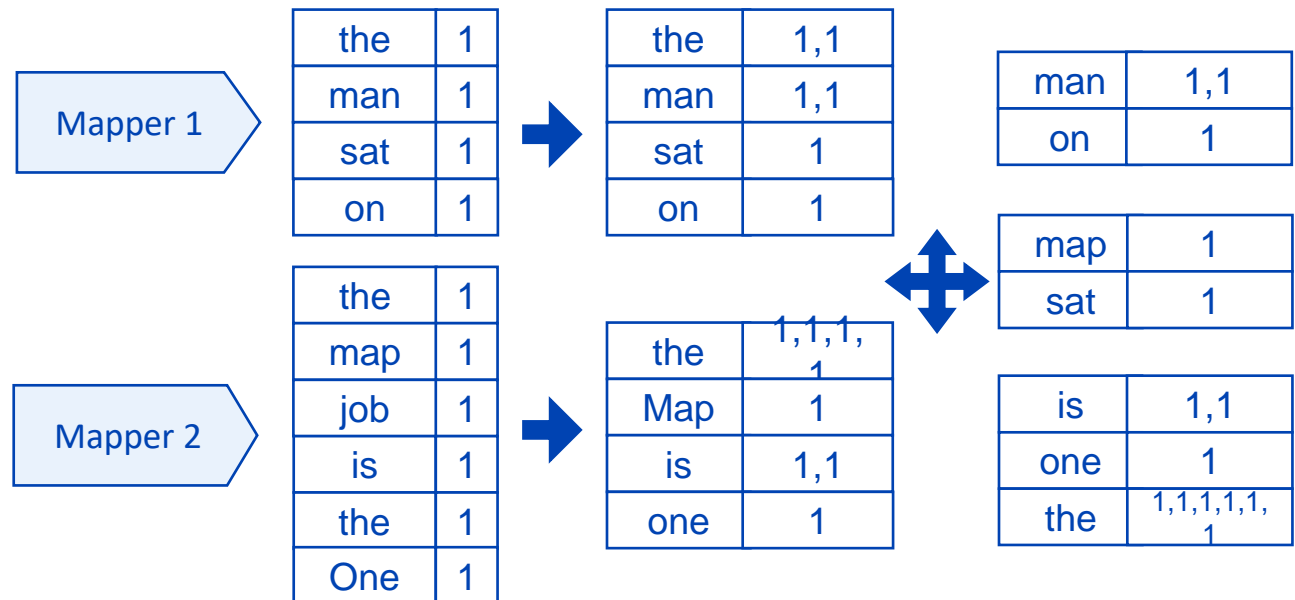
- ▶ Đọc văn bản từ nguồn đầu vào
- ▶ Tạo khóa: cặp giá trị
- ▶ Khóa là mỗi từ
- ▶ Giá trị không đổi 1



# MapReduce – Số từ

## I Giai đoạn Shuffle Sort

- ▶ Kết quả của mỗi Mapper được sắp xếp theo khóa và được lưu trữ trên đĩa cục bộ
- ▶ Theo tùy chọn, để có hiệu suất tốt hơn, khi các phím được sắp xếp, chúng có thể được tổng hợp



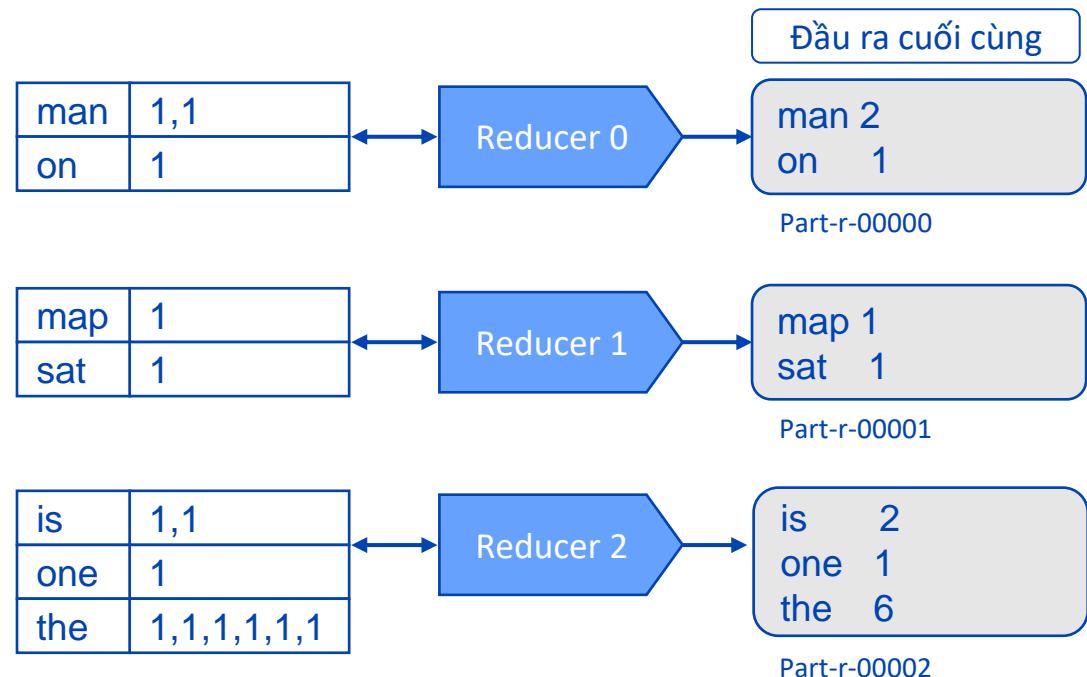
# MapReduce – Số từ

## I Giai đoạn Reducer

- ▶ Kết hợp các giá trị trung gian cho mỗi khóa thành một hoặc nhiều giá trị cuối cùng
- ▶ Mỗi bộ giảm tốc có thể chạy song song,
- ▶ Mỗi bộ giảm tốc hoạt động trên một tập hợp con khác nhau của khóa: cặp, được phân vùng theo khóa

## I Bottleneck

- ▶ Bước reduce chỉ có thể tiến hành sau khi tất cả những người lập bản đồ đã hoàn thành
- ▶ Người lập bản đồ chậm nhất quyết định tốc độ



Bài 1.

# Xử lý Big Data

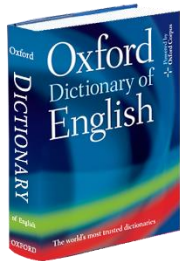
| 1.1. Ứng dụng và xử lý Big Data

| 1.2. Big Data trên Đám mây công cộng



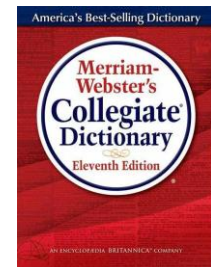
# Điện toán đám mây là gì?

I Định nghĩa từ điển “Điện toán đám mây”



“Việc sử dụng một mạng máy chủ từ xa được lưu trữ trên Internet để lưu trữ, quản lý và xử lý dữ liệu, thay vì máy chủ cục bộ hoặc máy tính cá nhân.”

“Việc lưu trữ dữ liệu máy tính được sử dụng thường xuyên trên nhiều máy chủ có thể được truy cập thông qua Internet.”



## Điện toán đám mây là gì?

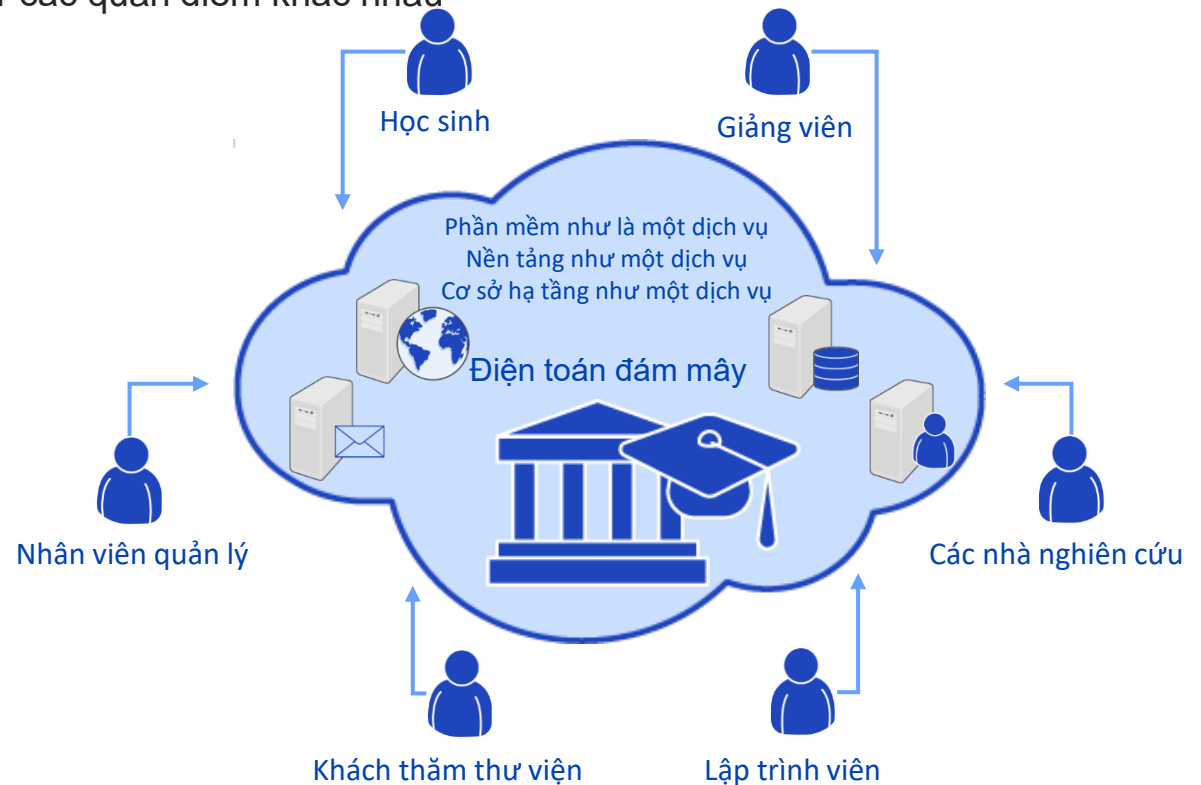
[Thảo luận]



# Quan điểm về điện toán đám mây

I Điện toán đám mây có ý nghĩa gì từ các quan điểm khác nhau

- ▶ Người dùng cuối
- ▶ Người lập trình ứng dụng
- ▶ Nhà cung cấp dịch vụ
- ▶ Quản lý cơ sở hạ tầng CNTT
- ▶ CIO
- ▶ CFO



# Sự phát triển của điện toán đám mây

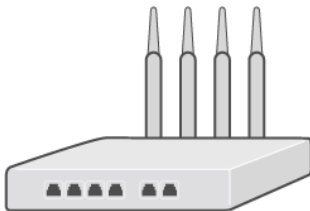
## I Những tiến bộ và thay đổi trong công nghệ đám mây

Đang phát triển	Giai đoạn	Đặc trưng
	Điện toán lưới	Giải quyết các vấn đề lớn với điện toán song song Được Liên minh toàn cầu chủ đạo
	Điện toán tiện ích	Tài nguyên máy tính được cung cấp dưới dạng dịch vụ có đồng hồ đo vào cuối những năm 1990
	Phần mềm dưới dạng Dịch vụ	Phần mềm dựa trên đăng ký được truy cập qua Internet đã đạt được động lực sau năm 2001
	Điện toán đám mây	Trung tâm dữ liệu thế hệ tiếp theo với công nghệ ảo hóa toàn bộ dịch vụ - IaaS, PaaS, & SaaS

## Các công nghệ hỗ trợ chính

- Mạng diện rộng nhanh phổ biến
- Máy chủ mạnh mẽ và giá cả phải chăng
- Công nghệ ảo hóa hiệu năng cao

WAN băng thông rộng



Máy chủ



Ảo hóa



Máy siêu giám sát



## Các đặc điểm chính của dịch vụ đám mây

- | Dịch vụ tự phục vụ theo yêu cầu
- | Truy cập mạng phổ biến
- | Tập hợp tài nguyên không phụ thuộc vào vị trí
- | Tính đàn hồi nhanh
- | Thanh toán cho mỗi lần sử dụng



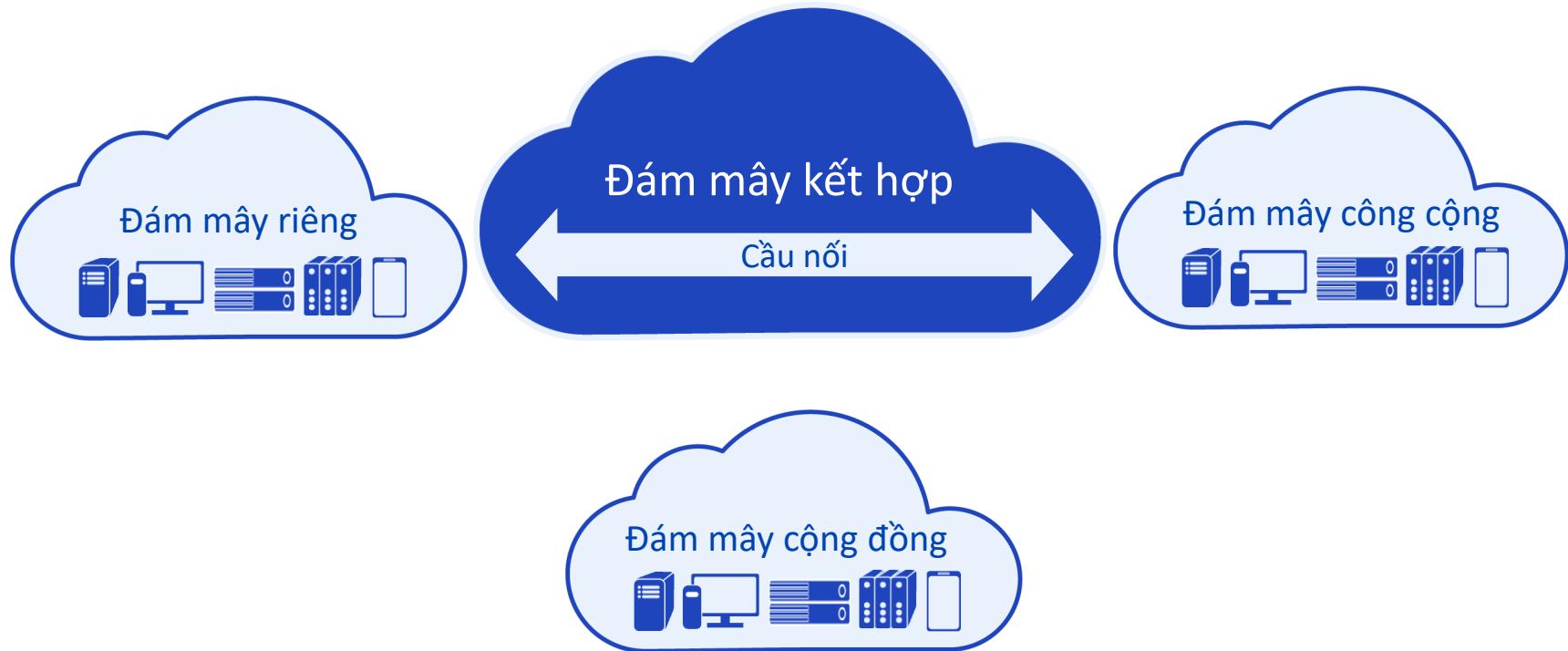
# Dịch vụ đám mây và mô hình tiêu thụ

## I So sánh SaaS, PaaS và IaaS

Mô hình dịch vụ	Mô hình tiêu thụ	Mô tả
Software as a Service (SaaS) (Phần mềm dưới dạng dịch vụ)	Trực tiếp tiêu dùng dịch vụ	Ứng dụng người dùng cuối được phân phối dưới dạng dịch vụ
Platform as a Service (PaaS) (Nền tảng dưới dạng Dịch vụ)	Phát triển và xây dựng trên nền tảng	Nền tảng ứng dụng hoặc phần mềm trung gian được cung cấp dưới dạng dịch vụ
Infrastructure as a Service (IaaS) (Cơ sở hạ tầng như một dịch vụ)	Cơ sở hạ tầng theo yêu cầu	Máy tính, lưu trữ hoặc cơ sở hạ tầng CNTT khác được cung cấp dưới dạng dịch vụ

# Mô hình triển khai đám mây

## I Các loại đám mây





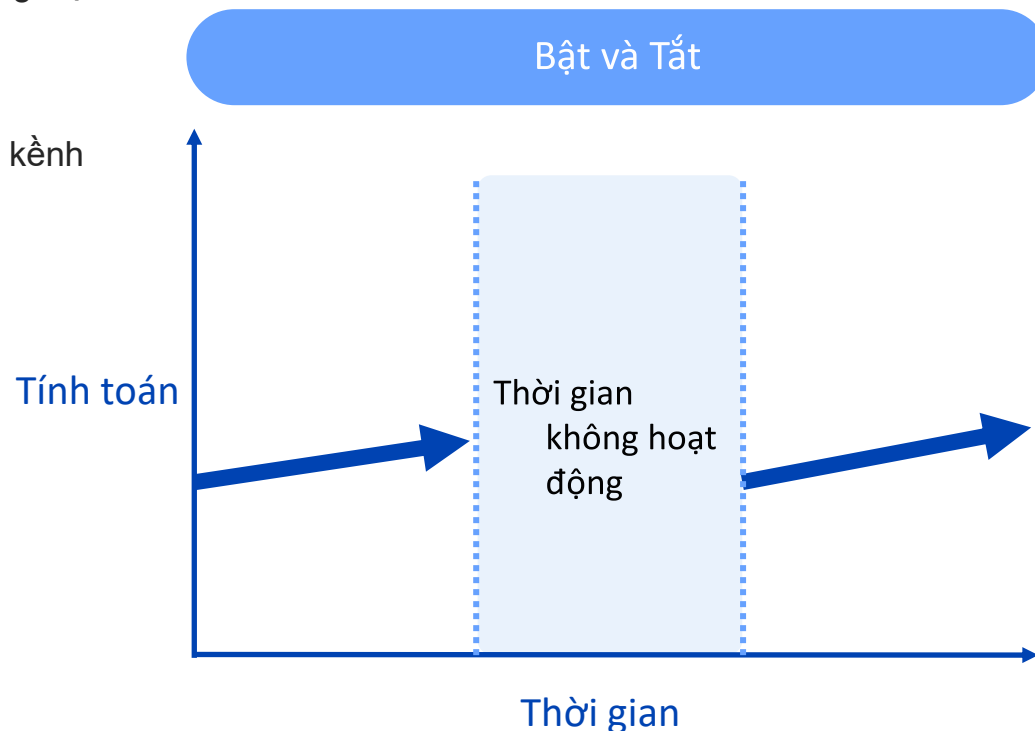
## Tại sao lại là Điện toán đám mây?



# Các mẫu khối lượng công việc trong điện toán đám mây

## I Mô hình tính toán điển hình

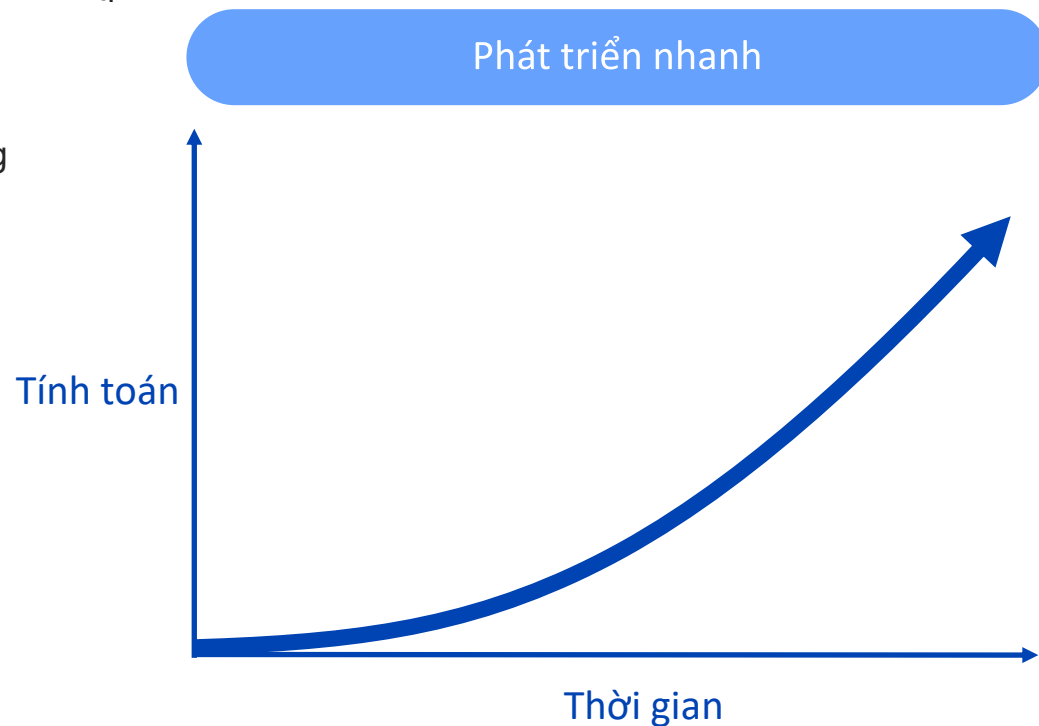
- ▶ Khối lượng công việc bật và tắt (ví dụ: công việc hàng loạt)
- ▶ Công suất lãng phí
- ▶ Thời gian để thị trường có thể được công kênh



# Các mẫu khối lượng công việc trong điện toán đám mây

## I Công ty phát triển nhanh chóng

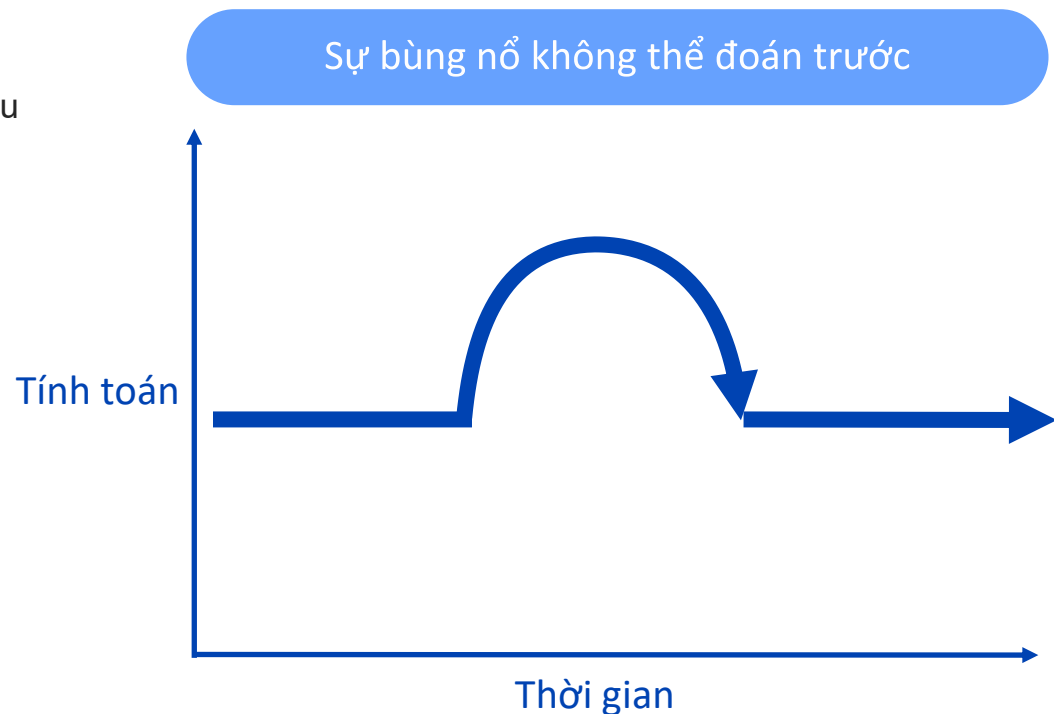
- ▶ Thách thức lớn đối với bộ phận CNTT để theo kịp với sự tăng trưởng
- ▶ Khả năng mất cơ hội kinh doanh
- ▶ Các vấn đề dịch vụ khách hàng tiềm năng



# Các mẫu khối lượng công việc trong điện toán đám mây

## I Sự kiện bất ngờ

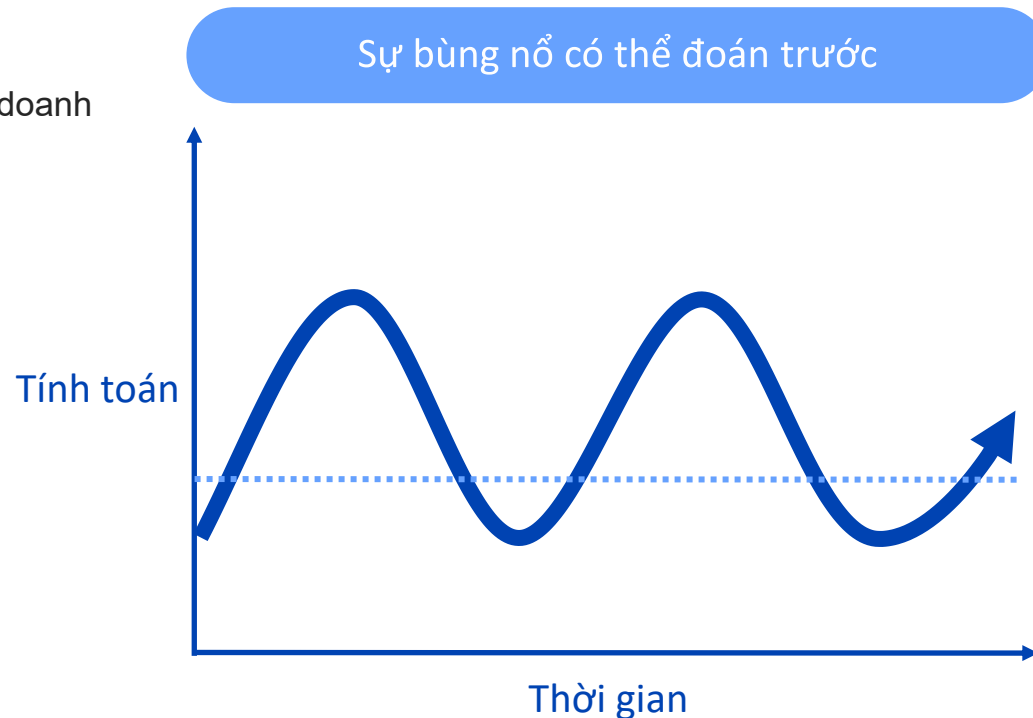
- ▶ Nhu cầu cao bất ngờ
- ▶ Mất cơ hội kinh doanh
- ▶ Công suất bị lãng phí nếu nhu cầu suy yếu



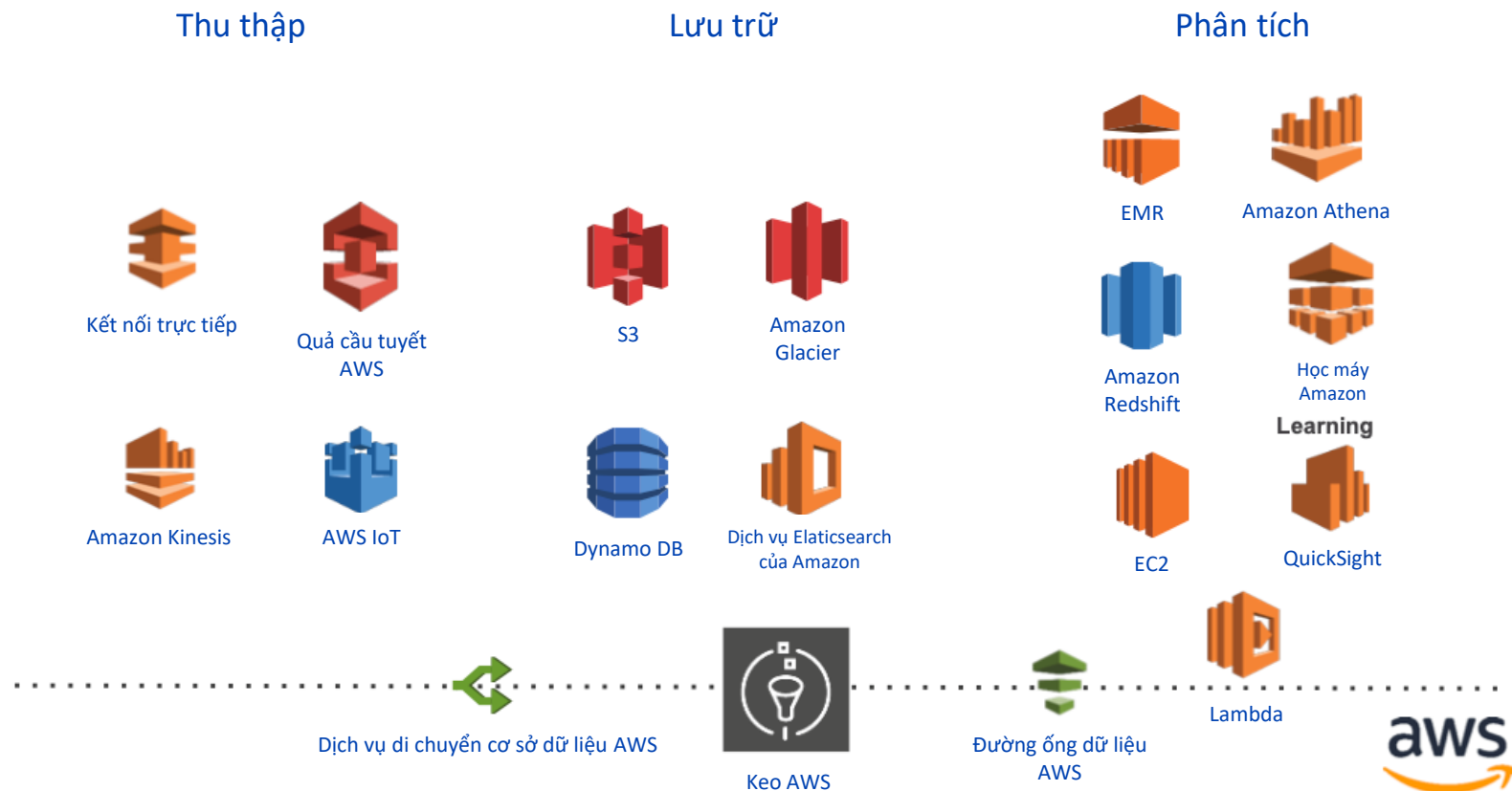
# Các mẫu khối lượng công việc trong điện toán đám mây

## I Điện toán đỉnh theo mùa

- ▶ Đỉnh và đáy theo mùa
- ▶ Dự phòng tiến thoái lưỡng nan
- ▶ Công suất bị lãng phí hoặc Mất việc kinh doanh

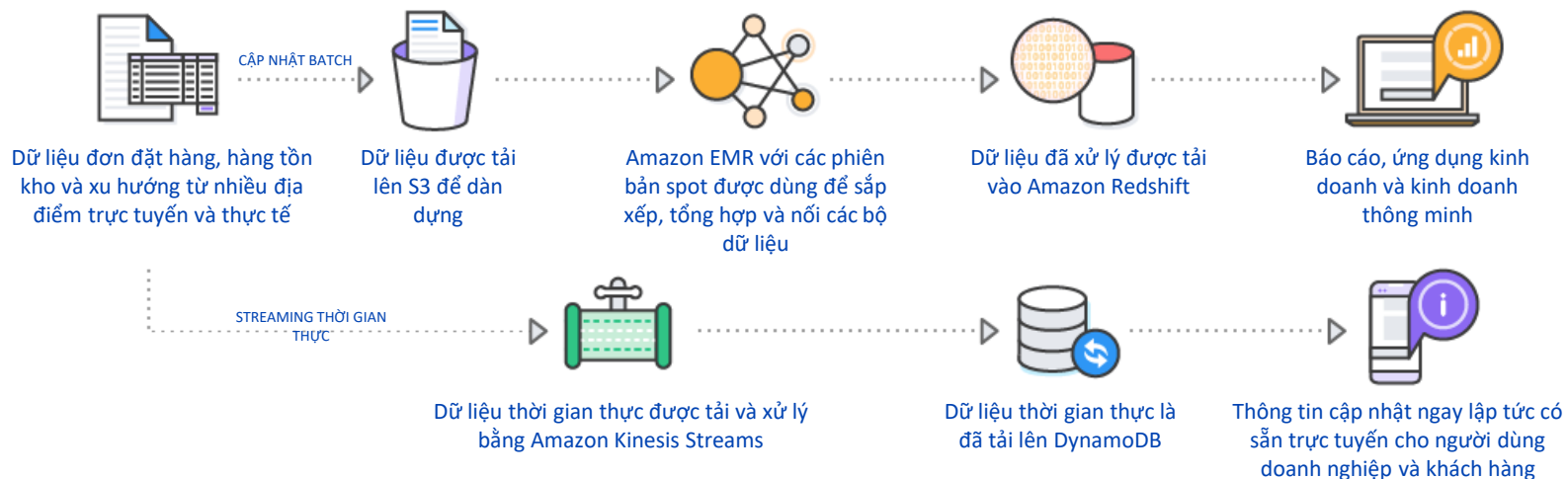


# Dịch vụ Big Data trên Amazon AWS



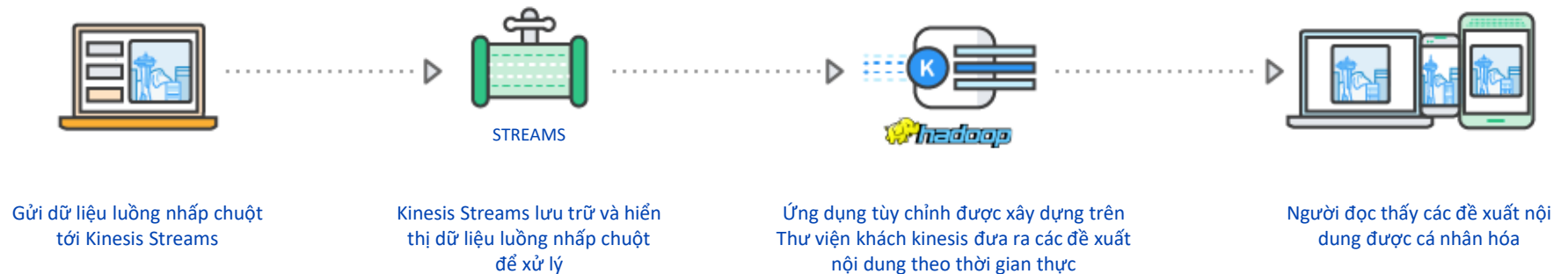
# Big Data trên Amazon AWS

## I Phân tích Big Data theo yêu cầu



# Big Data trên Amazon AWS

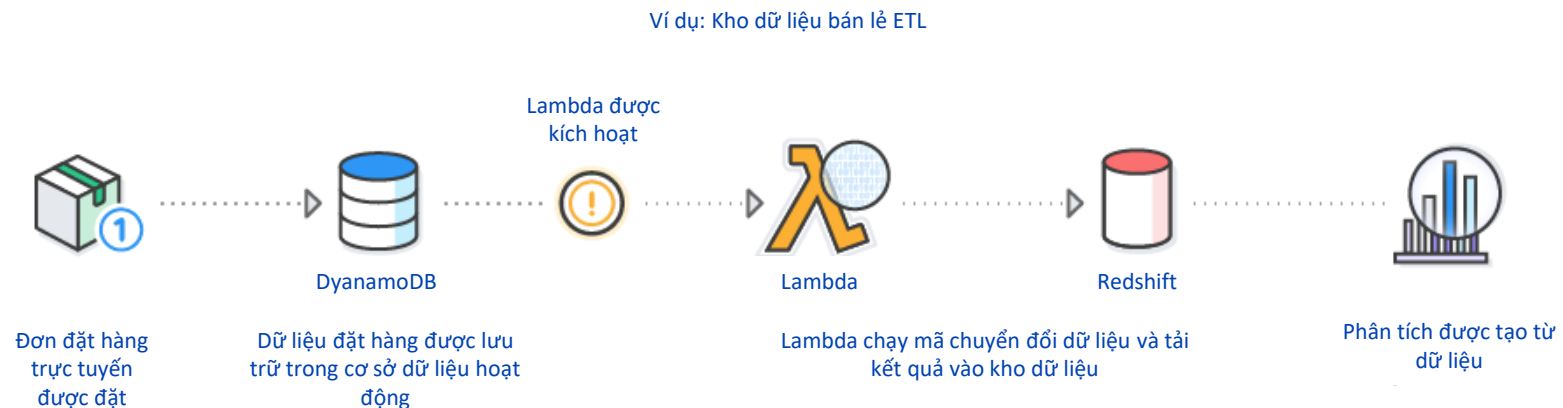
## I Phân tích luồng nhấp chuột





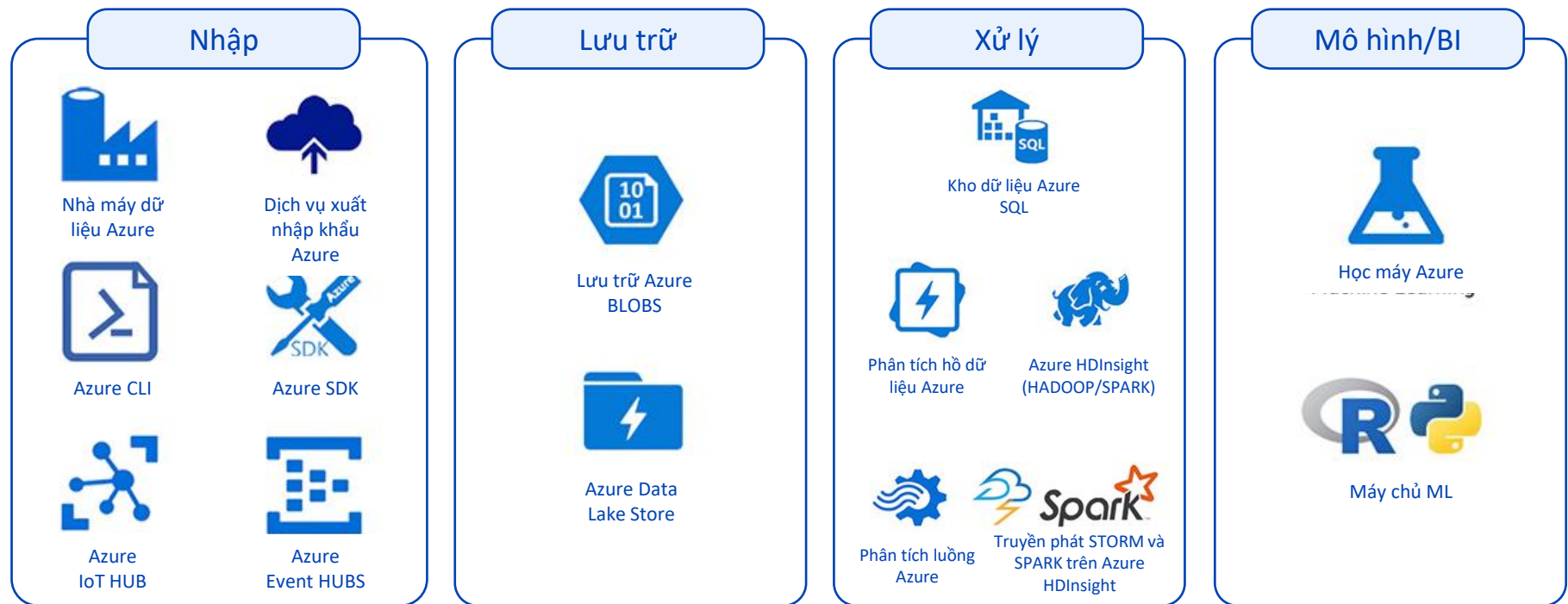
# Big Data trên Amazon AWS

I Trích xuất, chuyển đổi và tải theo hướng sự kiện



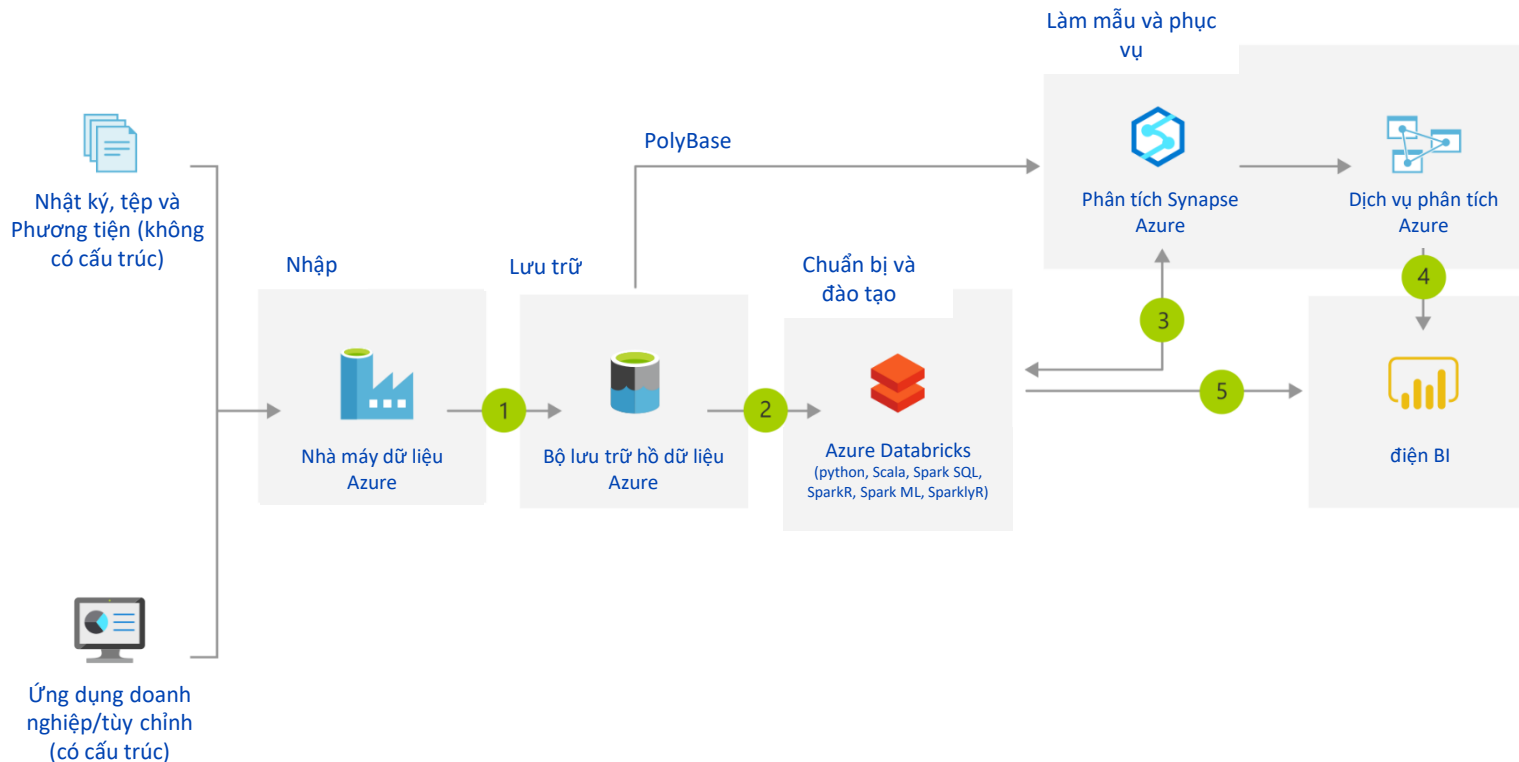
# Big Data trên Microsoft Azure

## I Công cụ Big Data trong Microsoft Azure



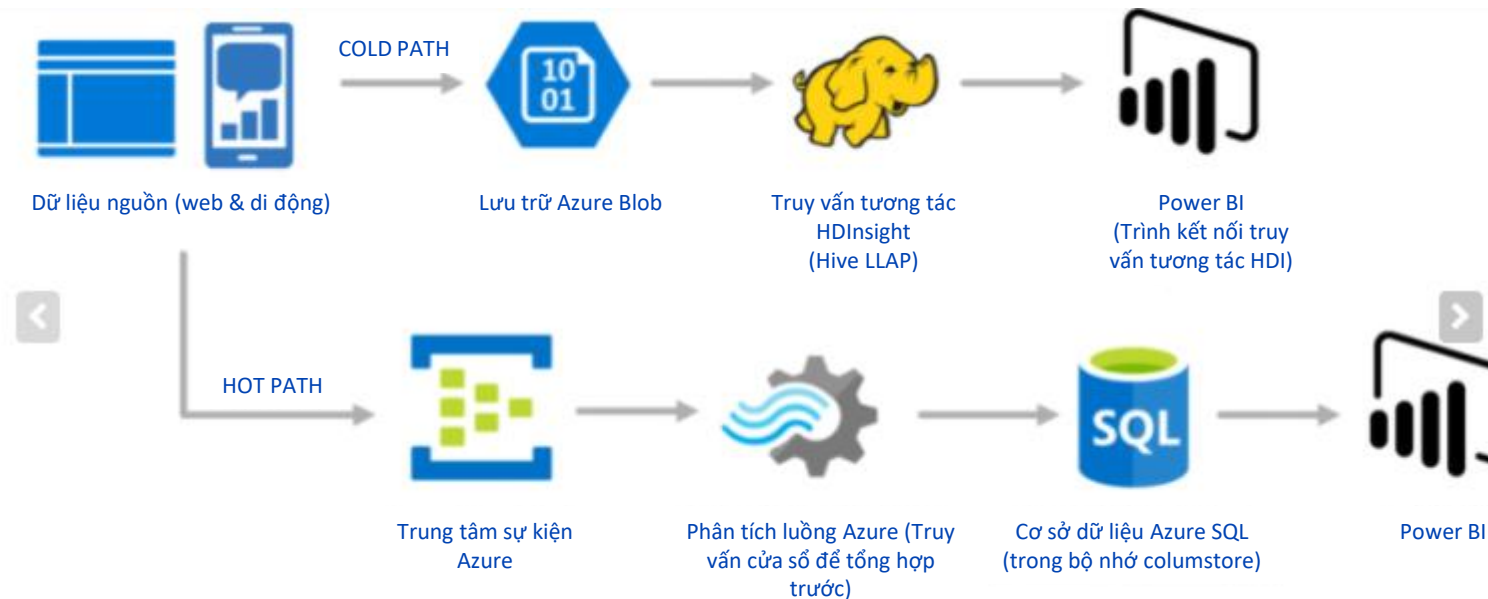
# Big Data trên Microsoft Azure

## I Tiền xử lý và Phân tích dữ liệu



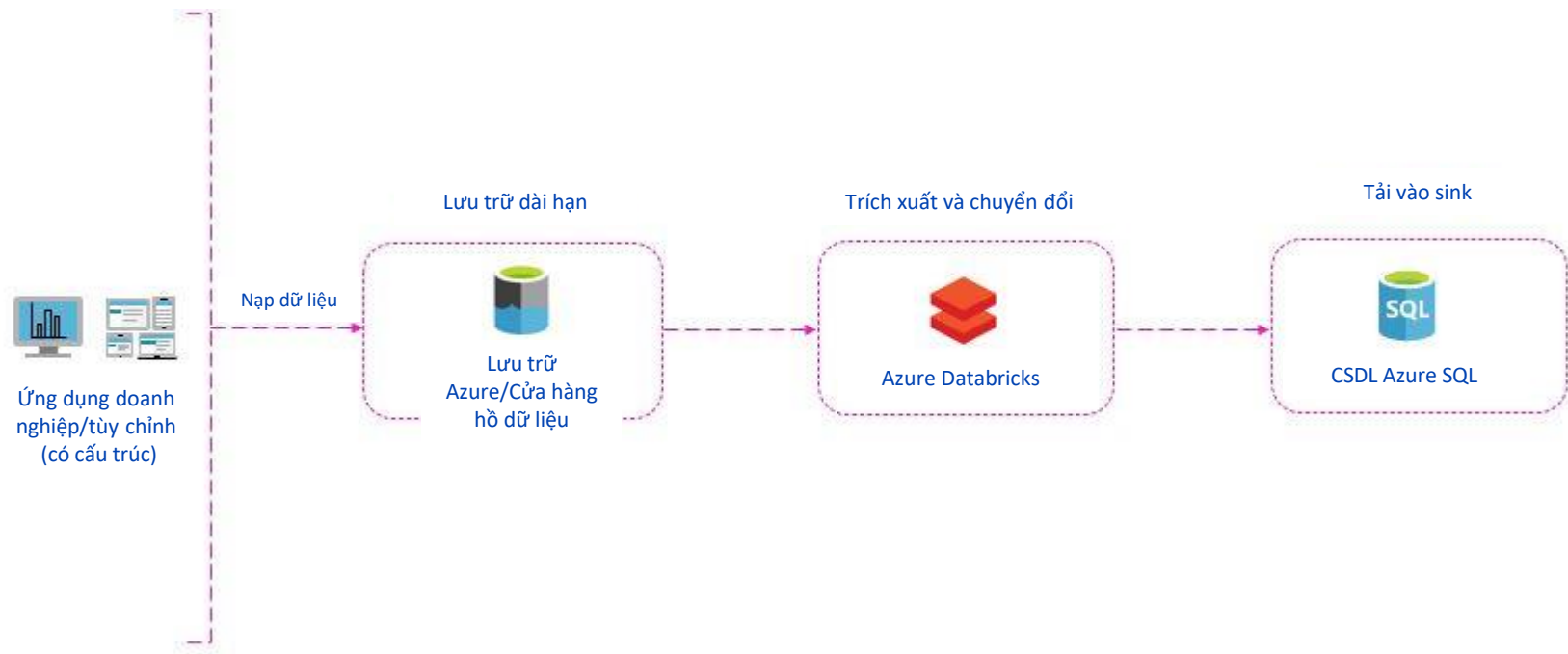
# Big Data trên Microsoft Azure

I Đồng thời xử lý theo thời gian thực và hàng loạt



# Big Data trên Microsoft Azure

## I Trích xuất chuyển đổi và nạp trên MS Azure



# [Thử nghiệm] Thử nghiệm AWS của giảng viên



Bài 2.

# Tổng quan về hệ thống Hadoop Core & Eco

Những nguyên tắc cơ bản Big Data

Bài 2.

# Tổng quan về hệ thống Hadoop Core & Eco

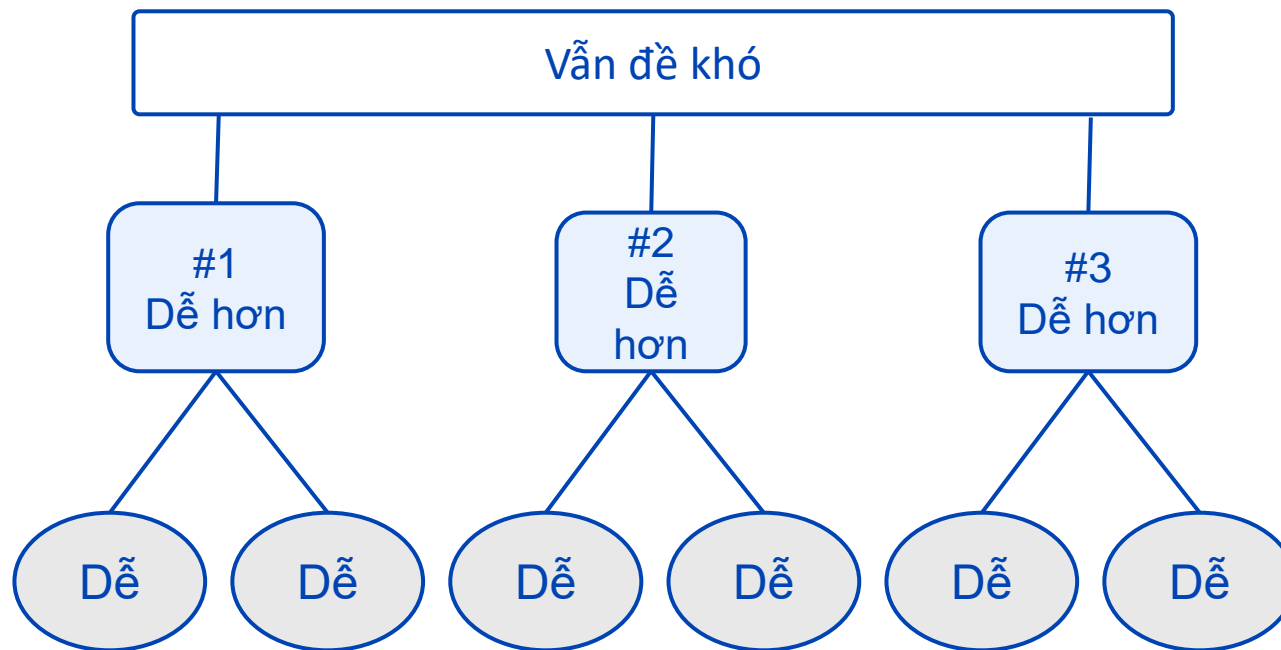
| 2.1 Tổng quan về nền tảng Apache Hadoop & Spark

| 2.2. Tổng quan về đường ống Big Data



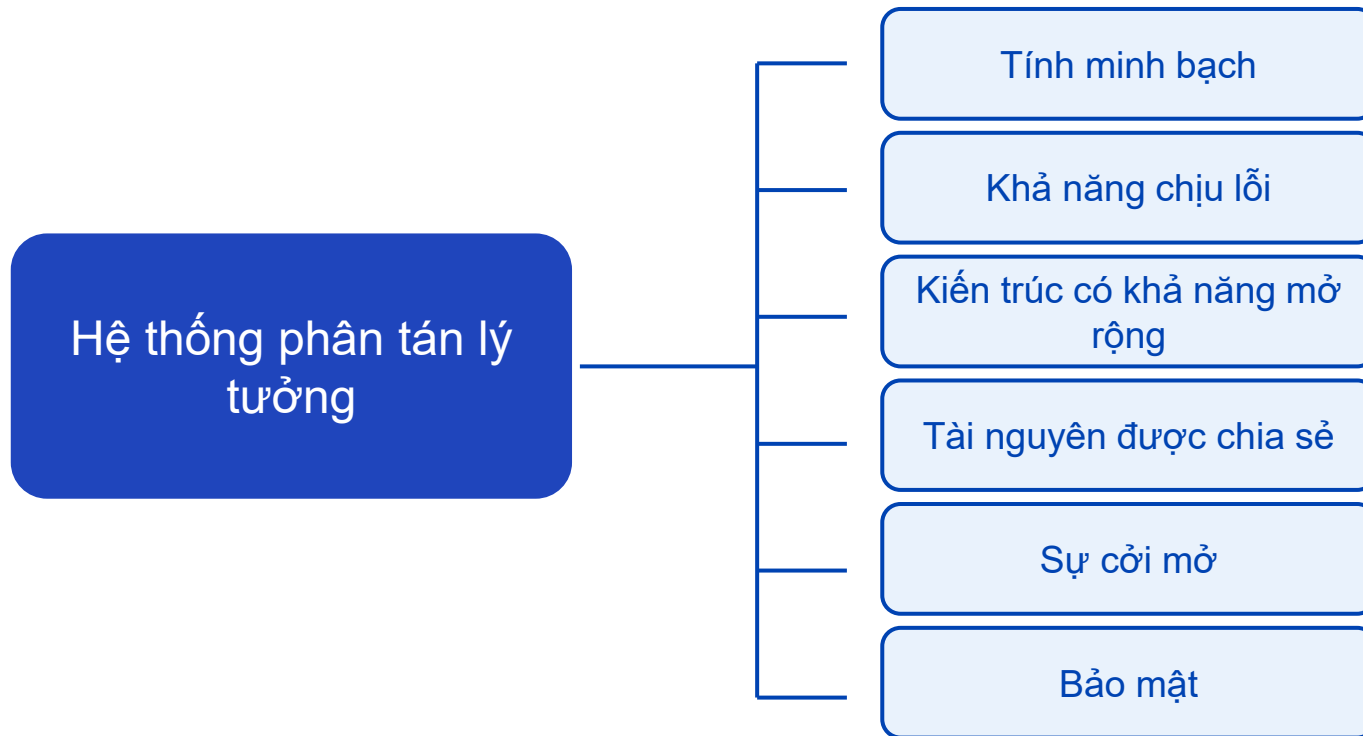
# Xử lý song song phân tán cho Big Data

- Giải quyết các vấn đề lớn bằng cách chia nhỏ chúng thành các vấn đề nhỏ hơn



# Những thách thức của hệ thống phân tán

I Những đặc điểm nào mà một Hệ thống phân tán lý tưởng phải có



# The Disk Bottleneck

### I Hiệu suất đĩa bị đình trệ

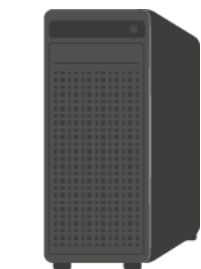
- ▶ Thông lượng đĩa năm 1997->2015: 16,6 MB/giây -> 210 MB/giây
- ▶ Trong khi giá đĩa đã giảm theo cấp số nhân, hiệu suất đĩa đã cải thiện rất ít
- ▶ Đọc 1 terabyte ở tốc độ trung bình 100 MB/giây sẽ mất khoảng 3 giờ để đọc



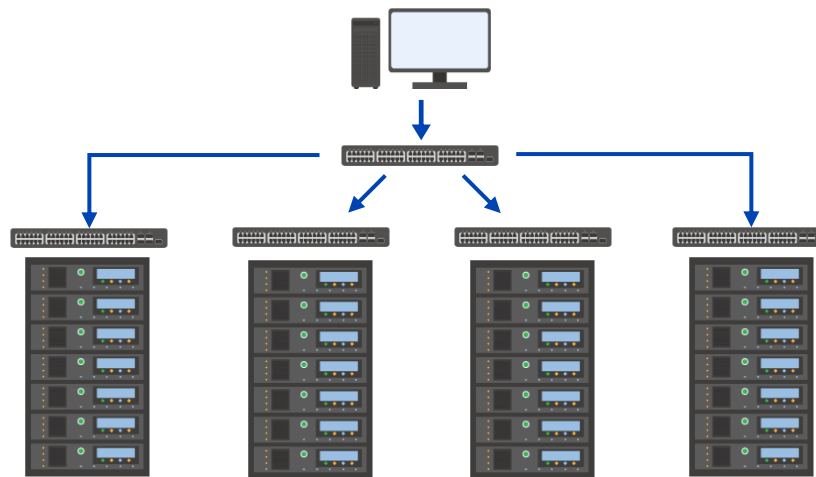
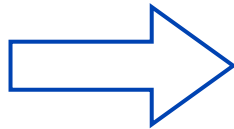
# Đọc đĩa song song tối ưu

### I Từ kiến trúc đĩa đơn sang đĩa song song

- ▶ Tốc độ truyền của một đĩa là 210 MB/giây -> 4 giờ cho 3TB
- ▶ 1000 đĩa trong cụm song song có thể truyền 3TB trong 15 giây
- ▶ Trong trường hợp này, 100 nút với 10 đĩa trên mỗi nút hoạt động song song.



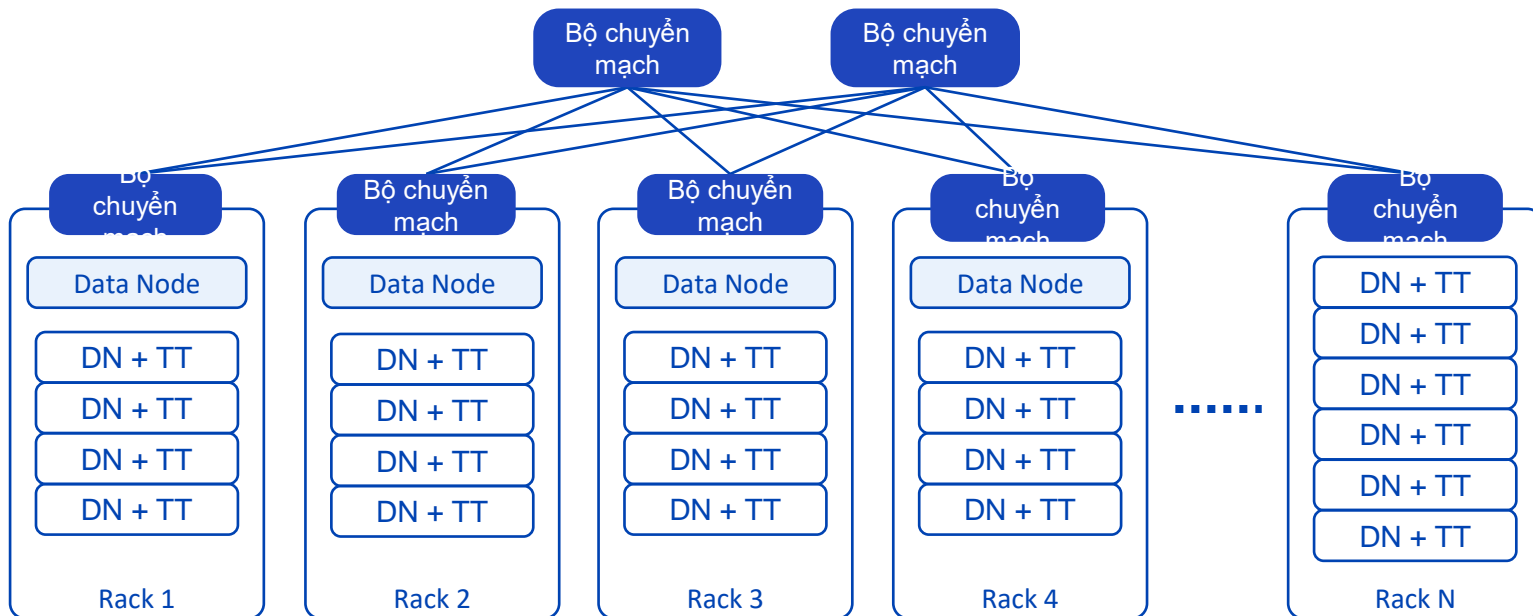
210 MB/s,  
3.96H



Cụm 100 nút với 10 đĩa/nút 14,28 giây

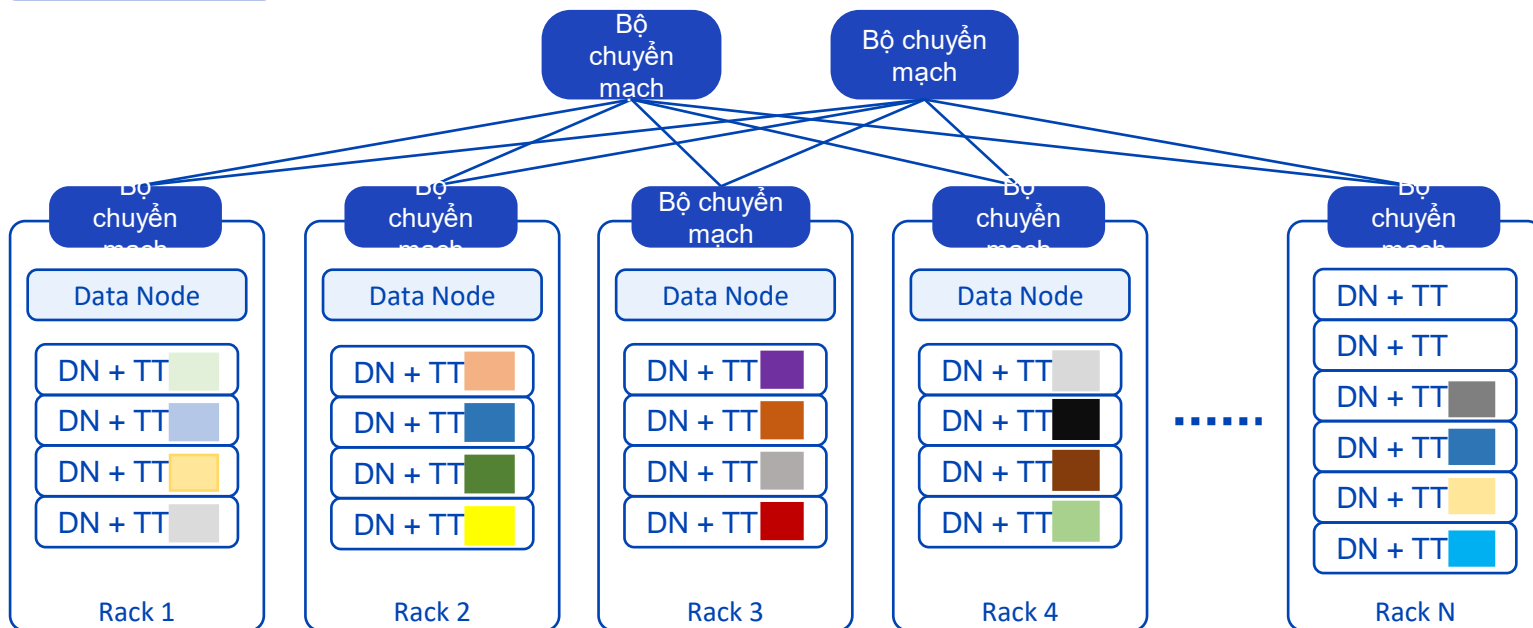
# Mẫu truy cập đĩa lý tưởng .vs. thực tế

- ▶ Một viễn cảnh thực tế hơn về phân phối khối được phân vùng
  - ▶ Các khối dữ liệu được phân vùng không bao giờ được phân phối theo cách lý tưởng
  - ▶ Việc đọc toàn bộ đồng thời chỉ có thể xảy ra nếu mỗi khối dữ liệu nằm trên máy chủ riêng biệt

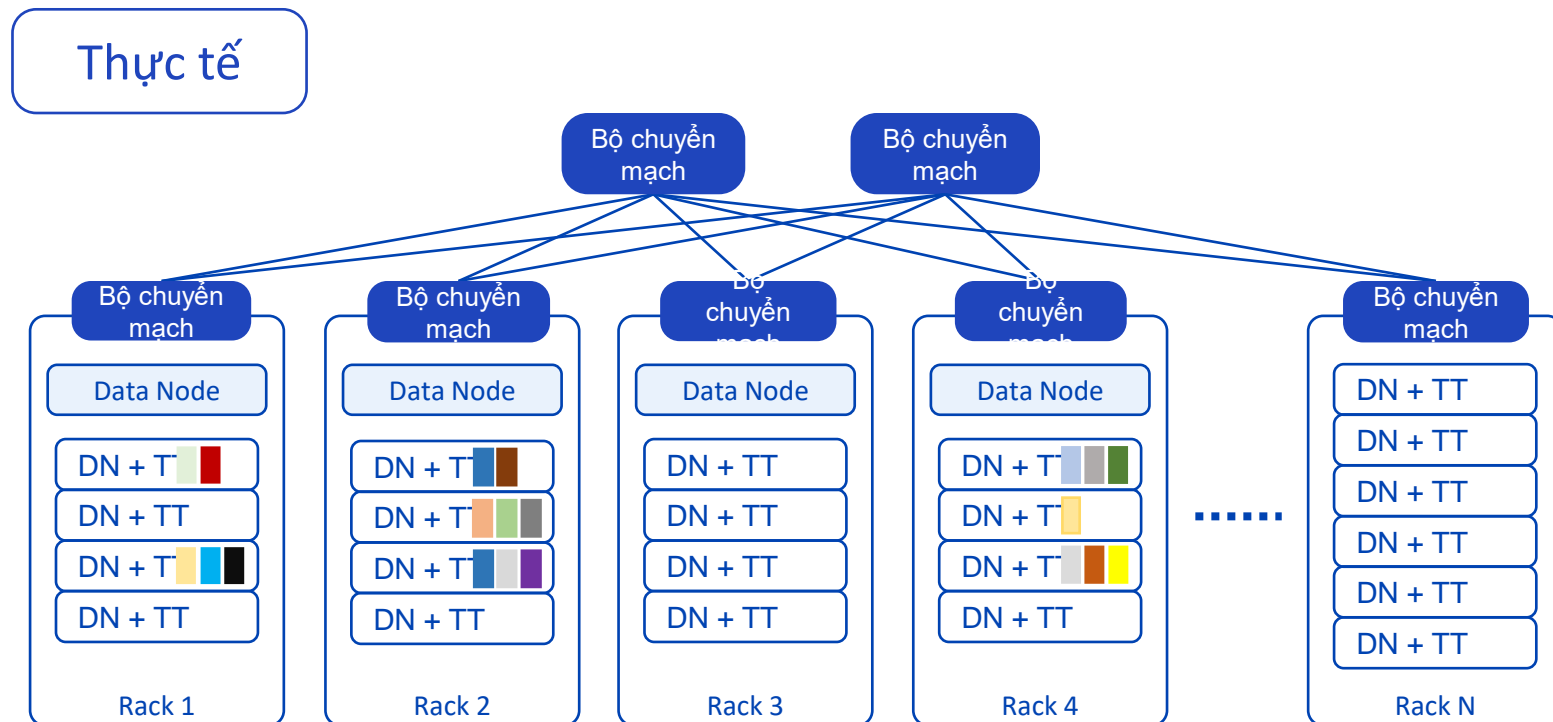


# Mẫu truy cập đĩa lý tưởng .vs. thực tế

Lý tưởng



# Mẫu truy cập đĩa lý tưởng .vs. thực tế



# Hadoop giải quyết các thách thức

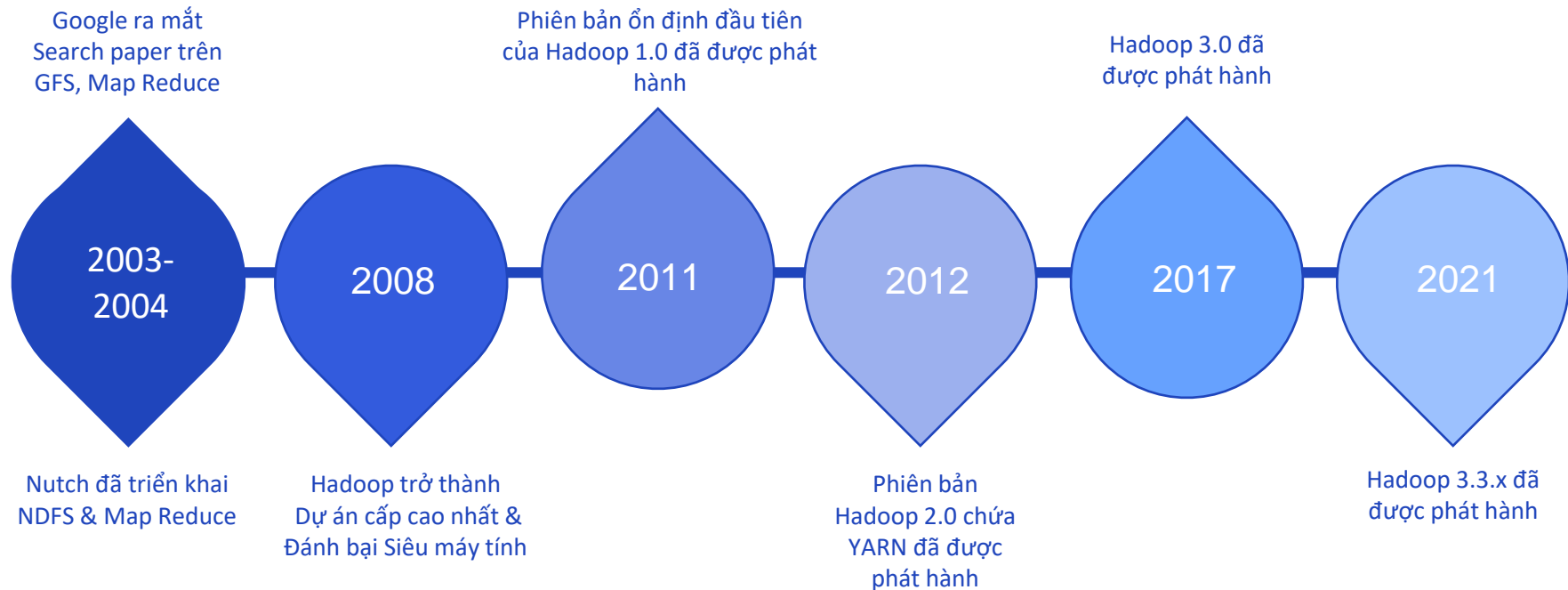
### I Một mô hình điện toán phân tán mới

- ▶ Xử lý dữ liệu nơi nó được lưu trữ
- ▶ Sử dụng daemon dựa trên phần mềm thay vì phần cứng để giải quyết các thách thức của điện toán phân tán
- ▶ Kiến trúc Master-Slave nơi các nô lệ có thể được thêm vào để mở rộng nền tảng
- ▶ Cung cấp một nền tảng nơi các nhà phát triển và người dùng có thể tập trung vào việc xử lý dữ liệu mà không phải lo lắng về các nhiệm vụ quản lý liên quan đến nền tảng



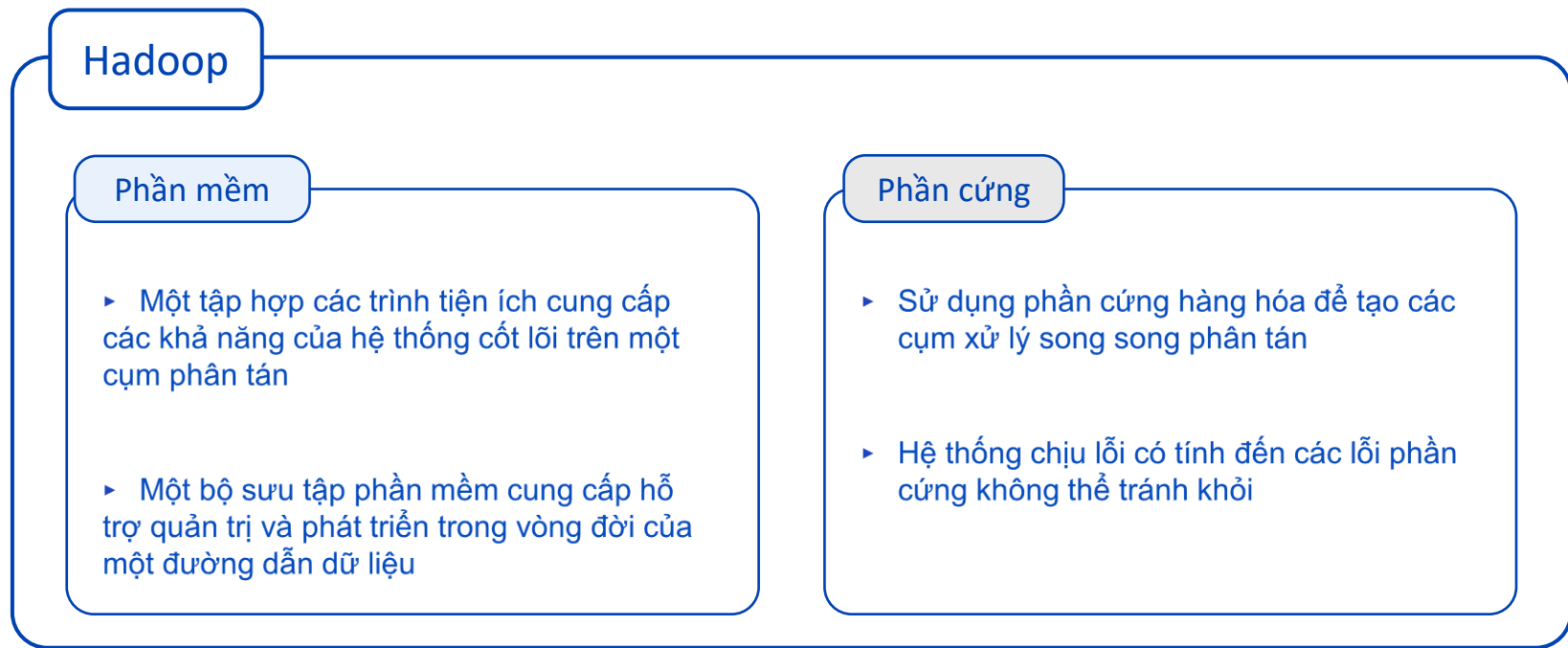
# Nguồn gốc của Hadoop

I Lịch sử của Hadoop trong những điểm ngắn gọn.



# Hadoop là gì?

### I Tập hợp phần mềm và phần cứng để xử lý Big Data



# Hadoop cung cấp giá trị gì

### I Nền tảng lưu trữ và xử lý dữ liệu phân tán

- ▶ Hỗ trợ ra quyết định tốt hơn bằng cách cung cấp giám sát thời gian thực các điểm dữ liệu kinh doanh
- ▶ Cung cấp phân tích dữ liệu nâng cao
- ▶ Các tổ chức hoàn toàn có thể tận dụng dữ liệu của họ
- ▶ Chạy trên hàng hóa thay vì kiến trúc tùy chỉnh

# Các tính năng chính của Hadoop



Tính linh hoạt



Khả năng chịu lỗi



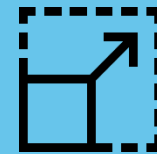
Mạnh mẽ



Nhanh



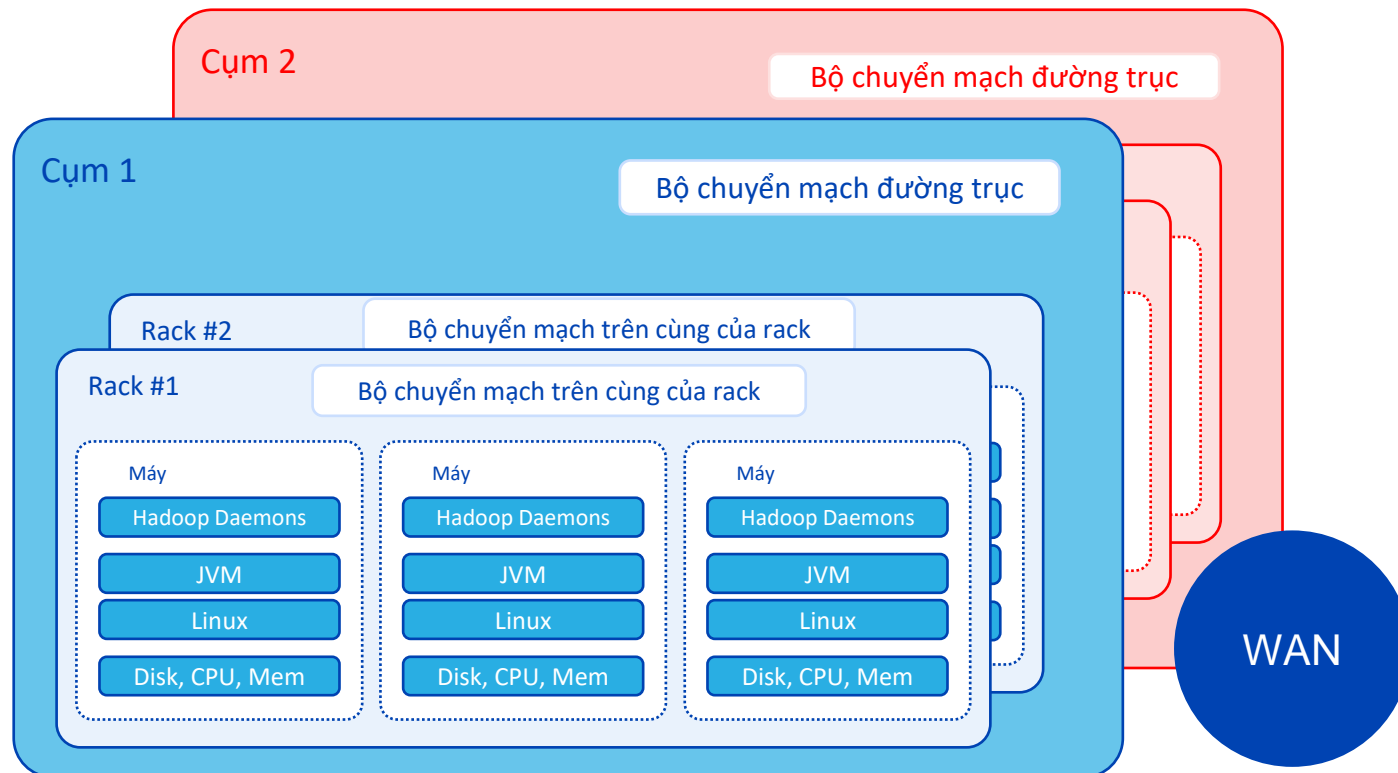
Chi phí hiệu quả



Khả năng mở rộng

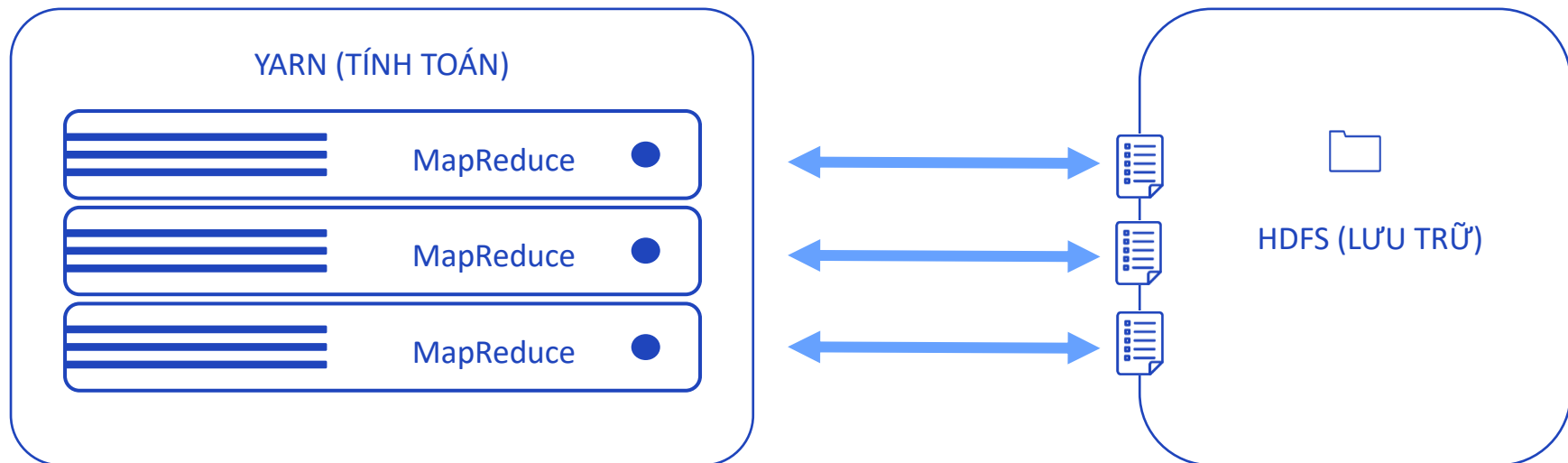
# Cụm Hadoop

- Một tập hợp phần mềm và phần cứng để xử lý Big Data



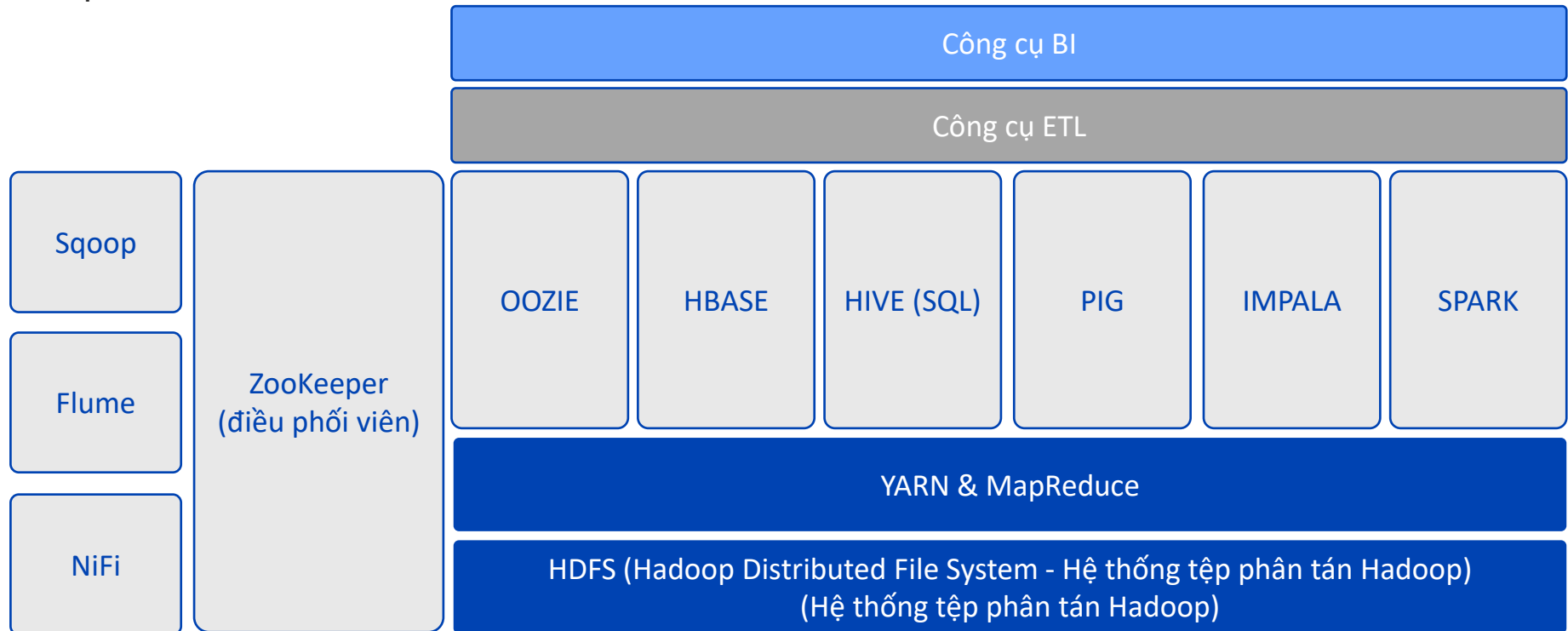
# Thành phần cốt lõi của Hadoop

- I Nền tảng lưu trữ và xử lý dữ liệu phân tán
  - ▶ Thành phần lưu trữ - HDFS
  - ▶ Thành phần tính toán - MapReduce hoặc Yarn



# Hệ sinh thái Hadoop

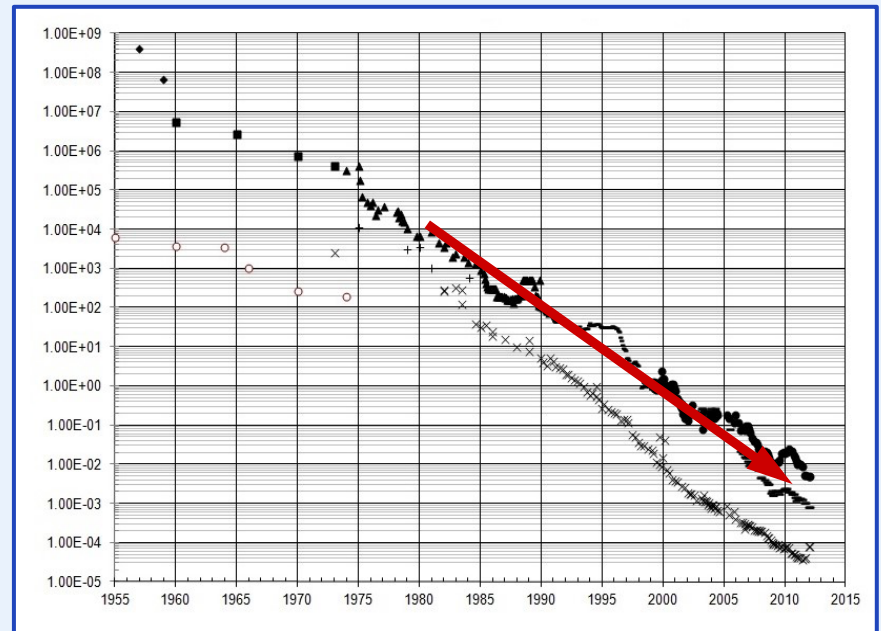
I Bạn biết bao nhiêu?



# Nền tảng Spark - Đã nắm bắt cơ hội

- | Chi phí bộ nhớ tiếp tục giảm
- | Thiết kế kiến trúc mới hiện có thể sử dụng nhiều bộ nhớ
- | Điện toán trong bộ nhớ được sinh ra
  - ▶ Cơ sở dữ liệu trong bộ nhớ
  - ▶ Các nền tảng trong bộ nhớ như Apache Spark
- | Đơn đặt hàng cường độ hiệu suất tốt hơn
- | Hệ điều hành 64 bit để truy cập nhiều bộ nhớ hơn

Giá lưu trữ (\$/MB) Tỷ lệ lôgarit

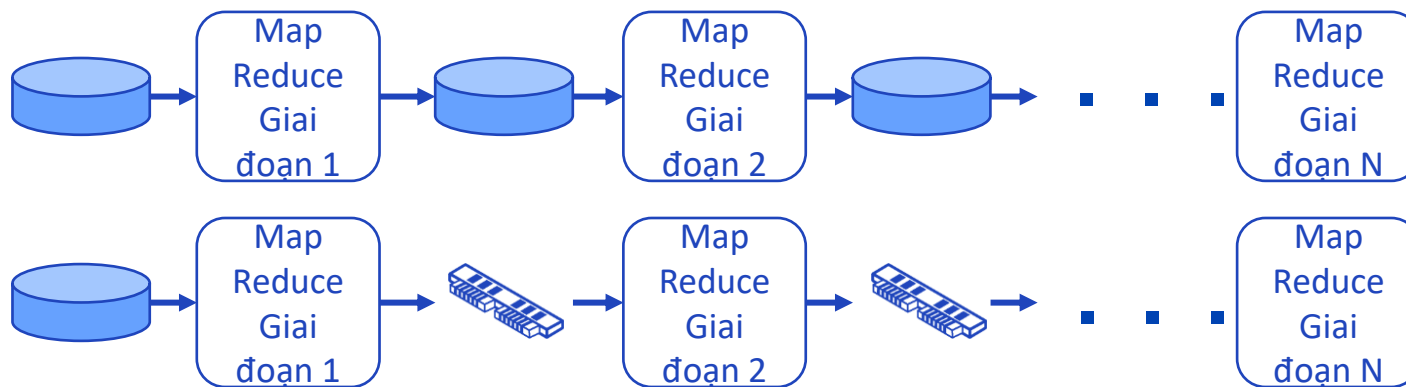


Năm



# MapReduce .vs. Apache Spark

- | Đọc và ghi vào bộ nhớ thay vì đĩa cứng
- | Chỉ thực hiện Disk I/O khi bắt đầu tính toán để tải dữ liệu vào bộ nhớ
- | Phân phối dữ liệu trên tất cả các tài nguyên bộ nhớ trong cụm
- | Thông thường nhanh hơn 10 ~ 100 lần



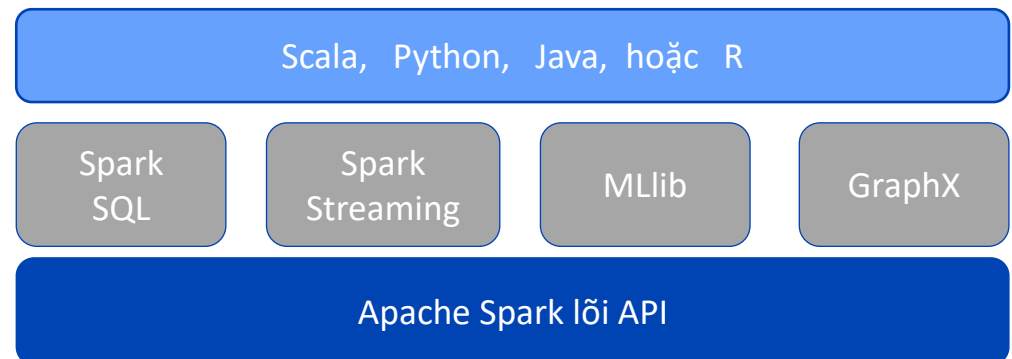
# Điện toán thế hệ tiếp theo cho Hadoop

- I Điện toán dữ liệu trong bộ nhớ
  - ▶ Nhanh hơn 10 đến 100 lần so với Hadoop MapReduce
- I Có thể tích hợp với Hadoop và các hệ sinh thái của nó
  - ▶ HDFS
  - ▶ Amazon S3, HBase, Hive, Cassandra
- I API lập trình dễ dàng hơn
  - ▶ API cấp cao rất mạnh mẽ
  - ▶ tiền xử lý dữ liệu
  - ▶ Tốt trong các ứng dụng nhiều giai đoạn phức tạp
  - ▶ Học máy
  - ▶ Truy vấn tương tác
- I Cung cấp xử lý dữ liệu thời gian thực

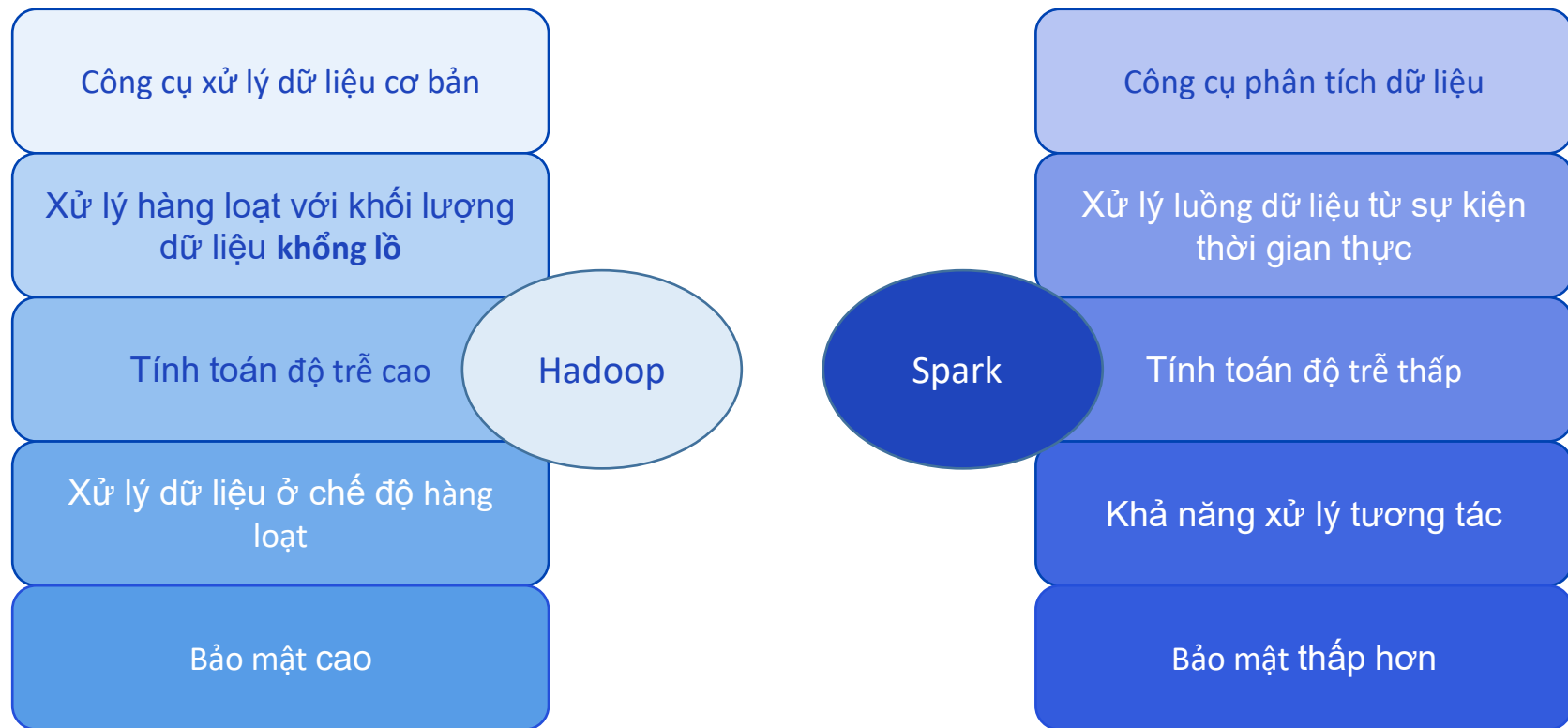


# Apache Spark Framework

- I Apache Spark cung cấp nhiều API lập trình giải quyết các lĩnh vực khác nhau
- I Lỗi API
  - ▶ Tốt cho xử lý dữ liệu phi cấu trúc và bán cấu trúc
- I Spark SQL – Khung dữ liệu API
  - ▶ Tốt cho việc xử lý dữ liệu có cấu trúc
- I Spark Streaming - API truyền trực tuyến có cấu trúc và Dstream
  - ▶ Tốt cho việc xử lý dữ liệu thời gian thực
- I ML và Mllib API
  - ▶ Học máy Spark
- I GraphX API
  - ▶ Xử lý các nút và cạnh



# Ưu và nhược điểm của Hadoop và Spark



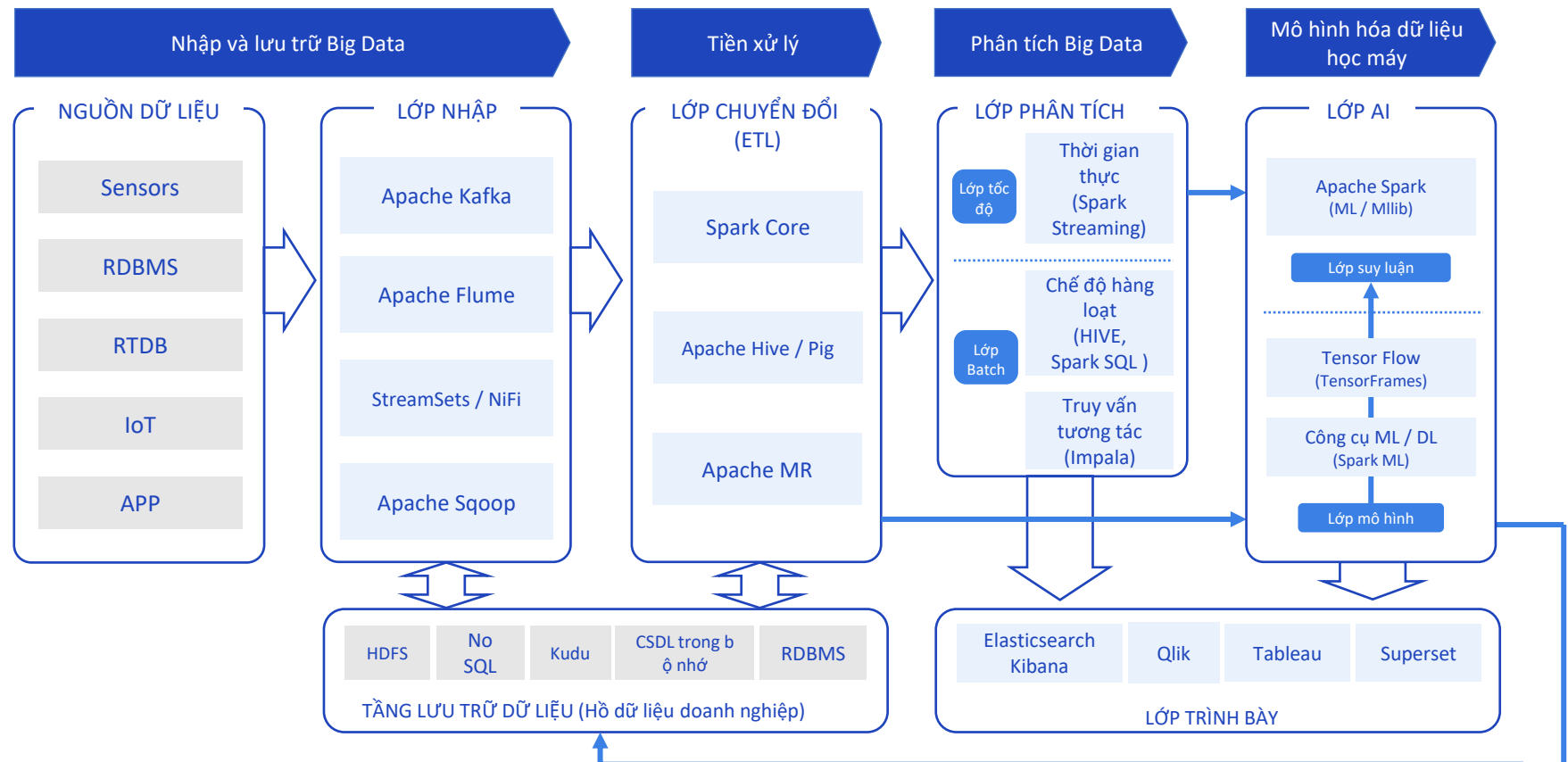
Bài 2.

# Tổng quan về hệ thống Hadoop Core & Eco

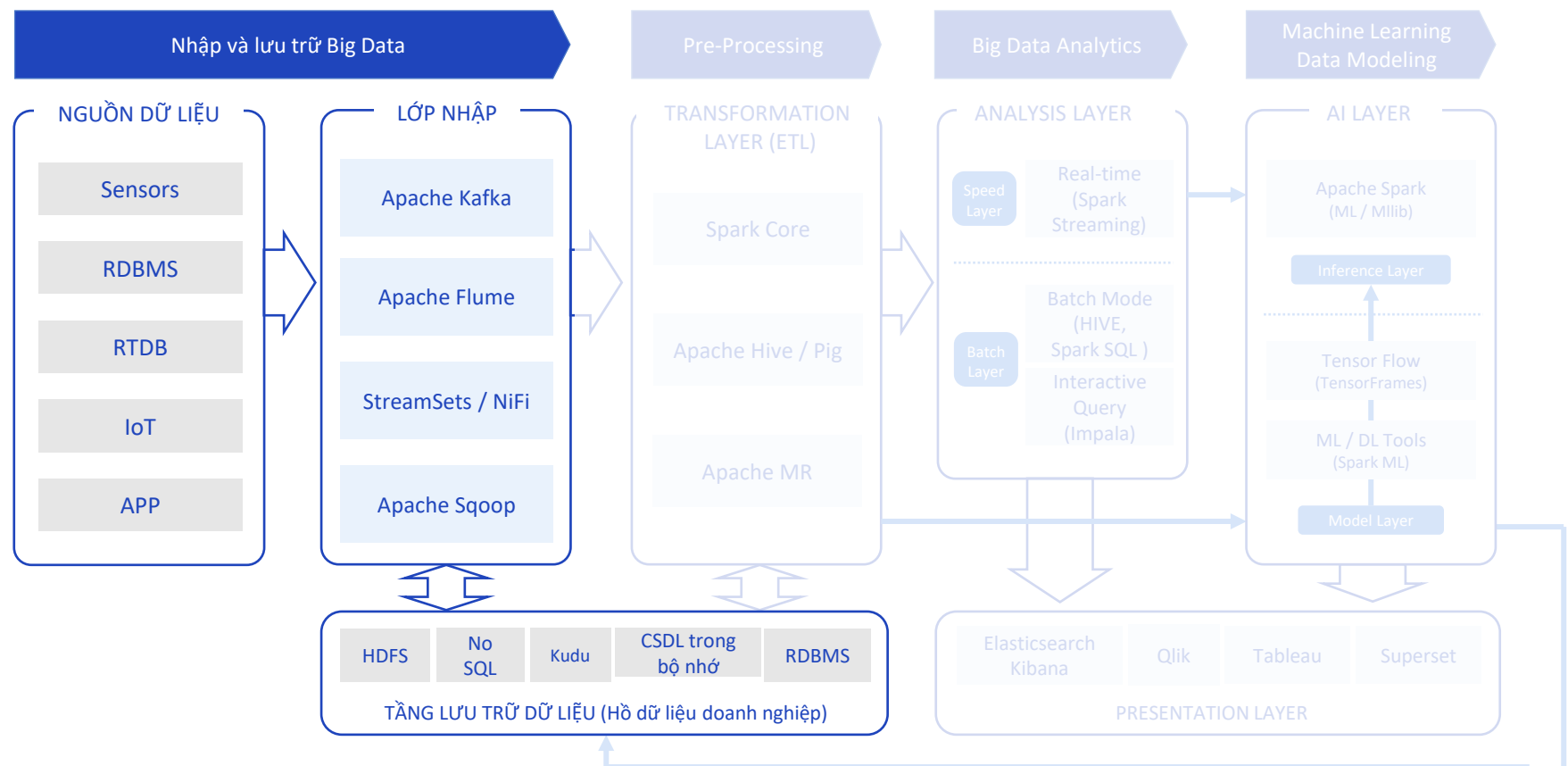
| 2.1 Tổng quan về nền tảng Apache Hadoop & Spark

| 2.2. Tổng quan về đường ống Big Data

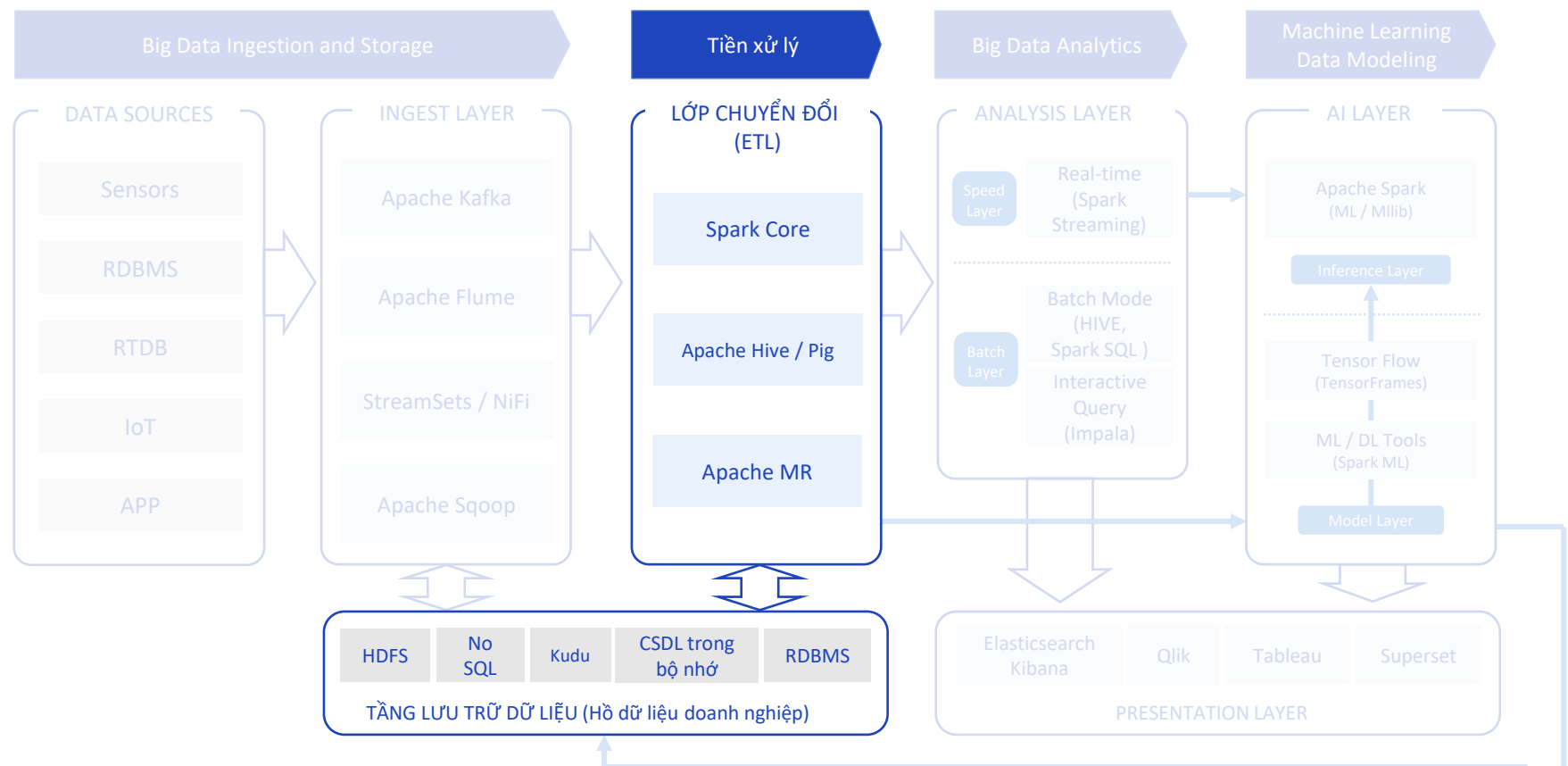
# Đường ống dữ liệu cho Big Data



# Thu thập và lưu trữ dữ liệu

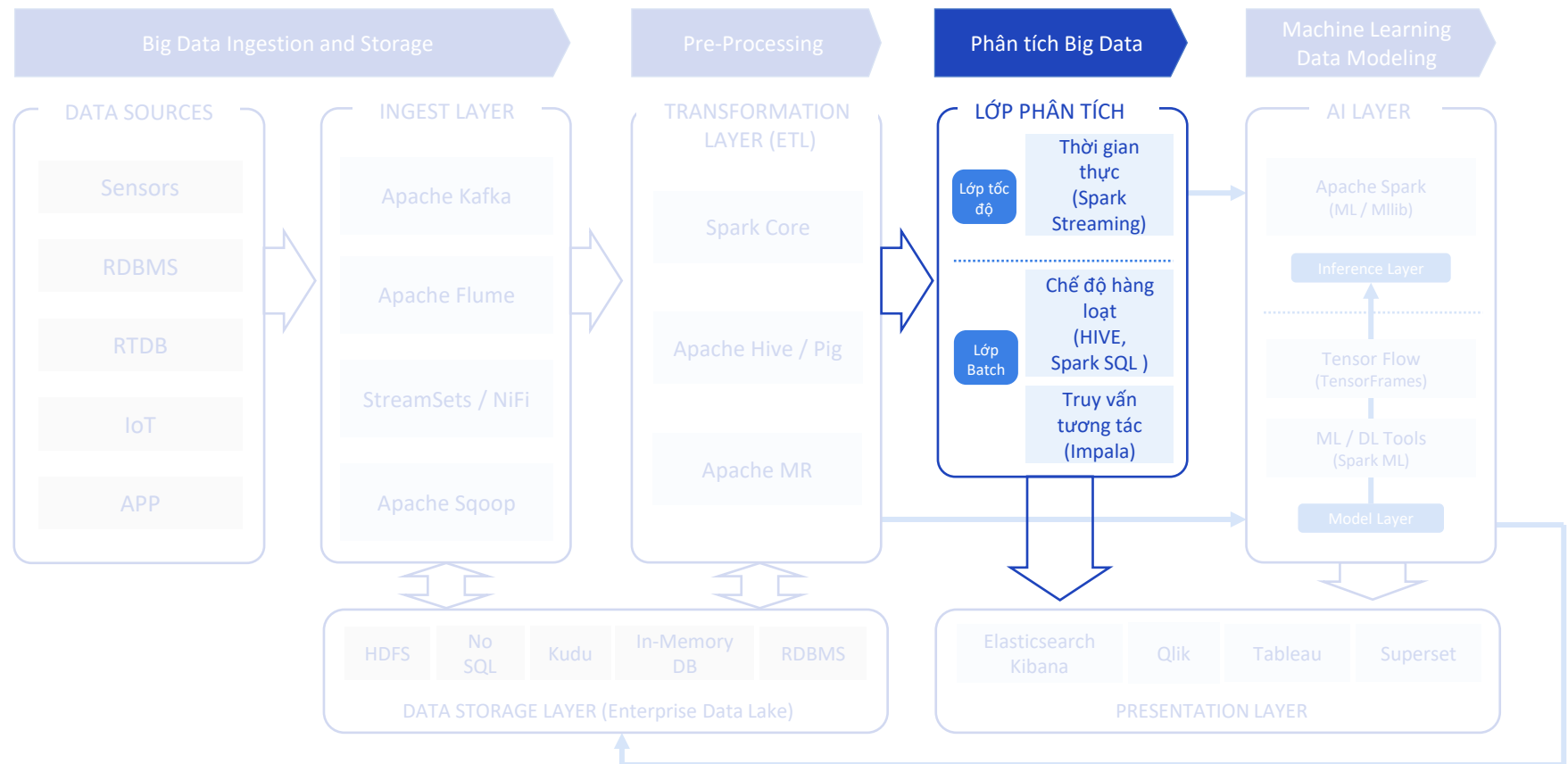


# Chuyển đổi dữ liệu của bạn

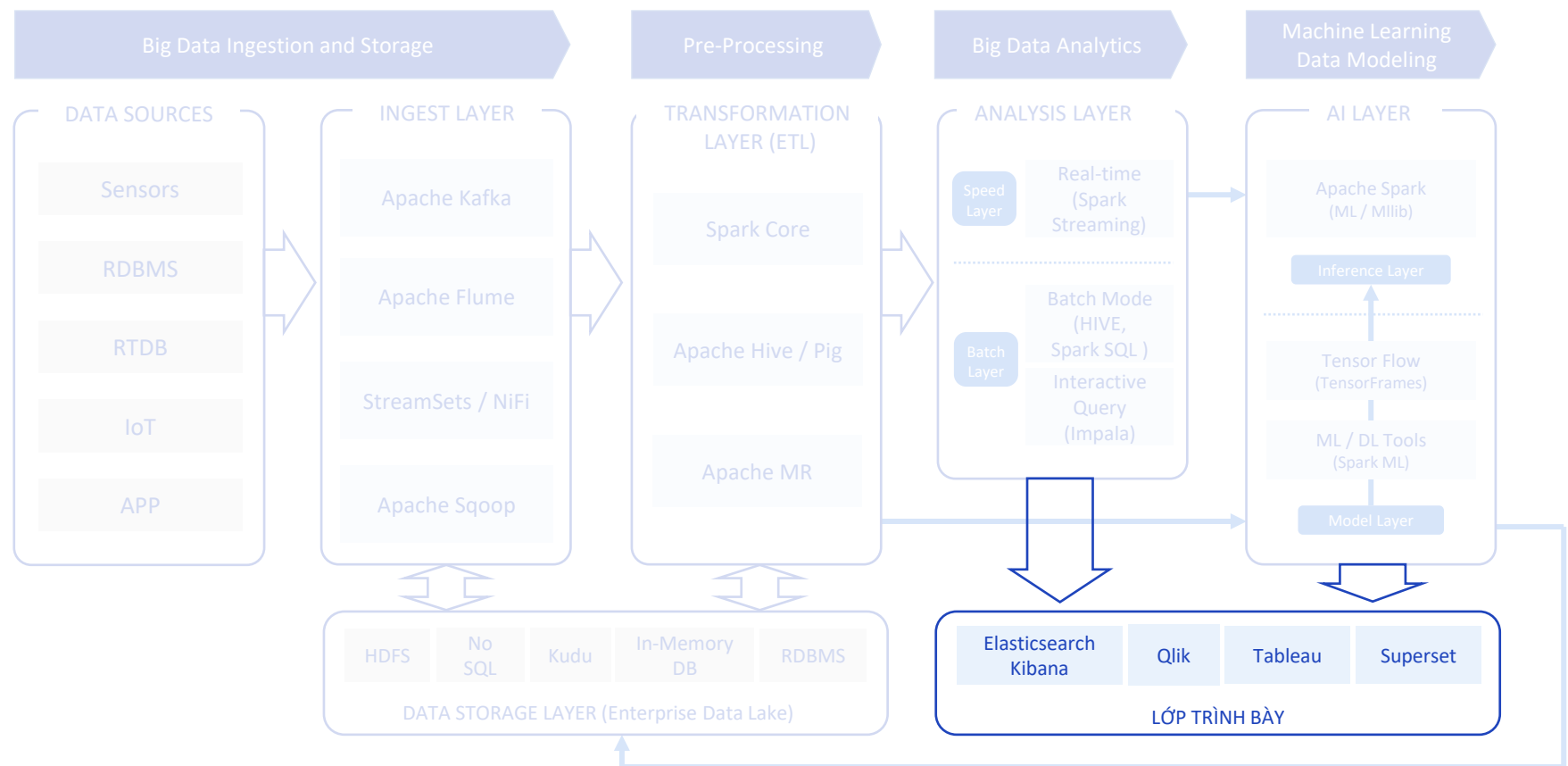




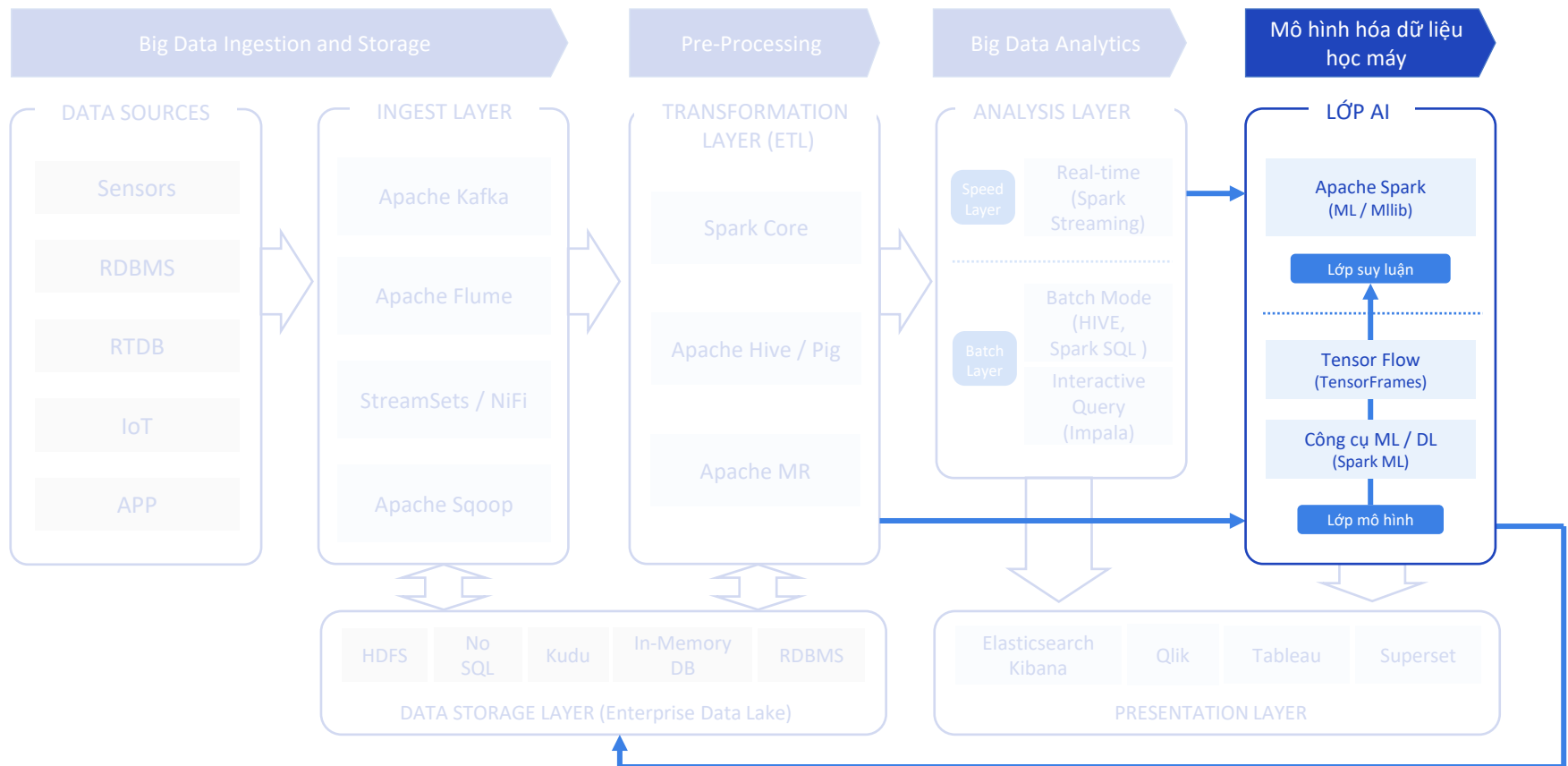
# Truy vấn dữ liệu của bạn



# Trình bày dữ liệu của bạn



# Lập mô hình dữ liệu của bạn với AI



# Câu hỏi ôn tập (1/2)

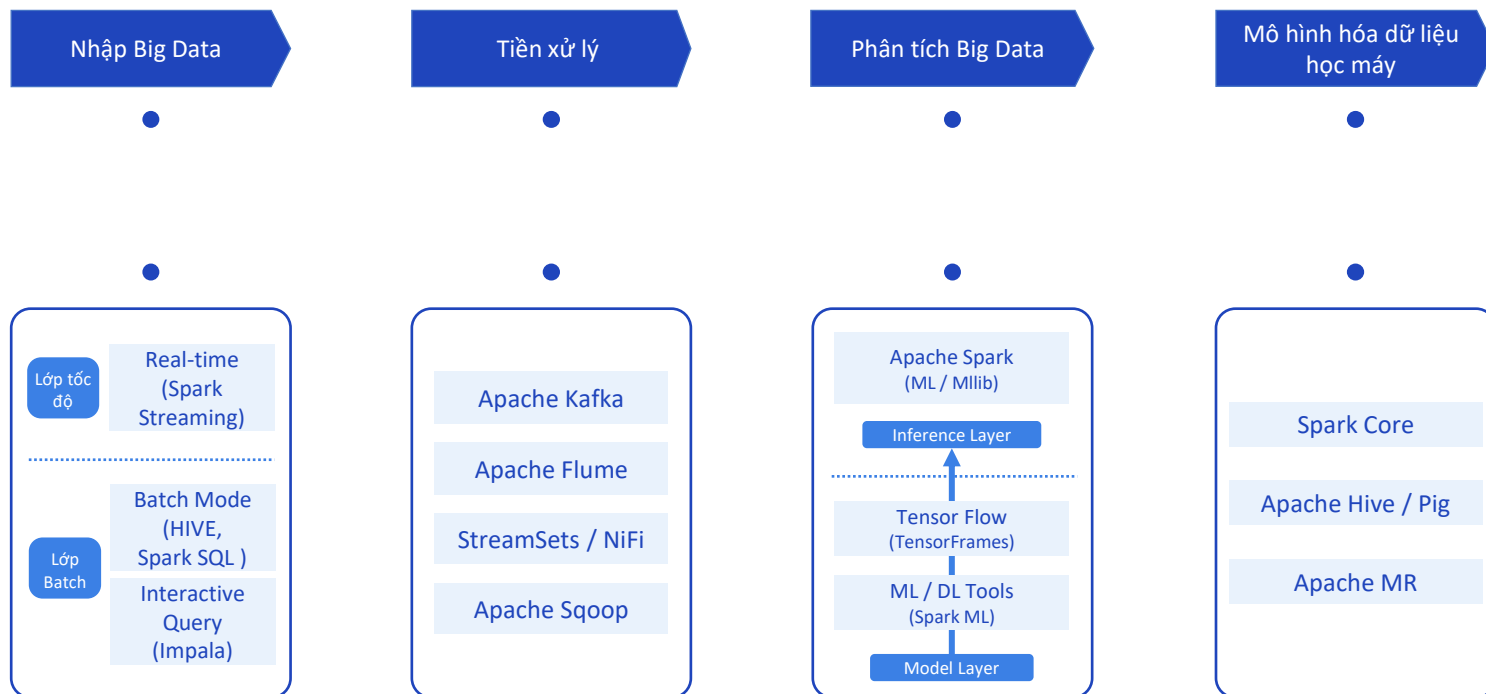
1. Là nền tảng đầu tiên cho dữ liệu lớn, tên của bộ sưu tập phần mềm và phần cứng xử lý dữ liệu lớn là gì?
2. Tên của hệ thống tập phân tán xử lý các tập dữ liệu lớn chạy trên phần cứng hàng hóa là gì?

# Câu hỏi ôn tập (1/2)

1. Là nền tảng đầu tiên cho dữ liệu lớn, tên của bộ sưu tập phần mềm và phần cứng xử lý dữ liệu lớn là gì?  
**Hadoop**
2. Tên của hệ thống tệp phân tán xử lý các tập dữ liệu lớn chạy trên phần cứng hàng hóa là gì?  
**HDFS (Hadoop Distributed File System)**

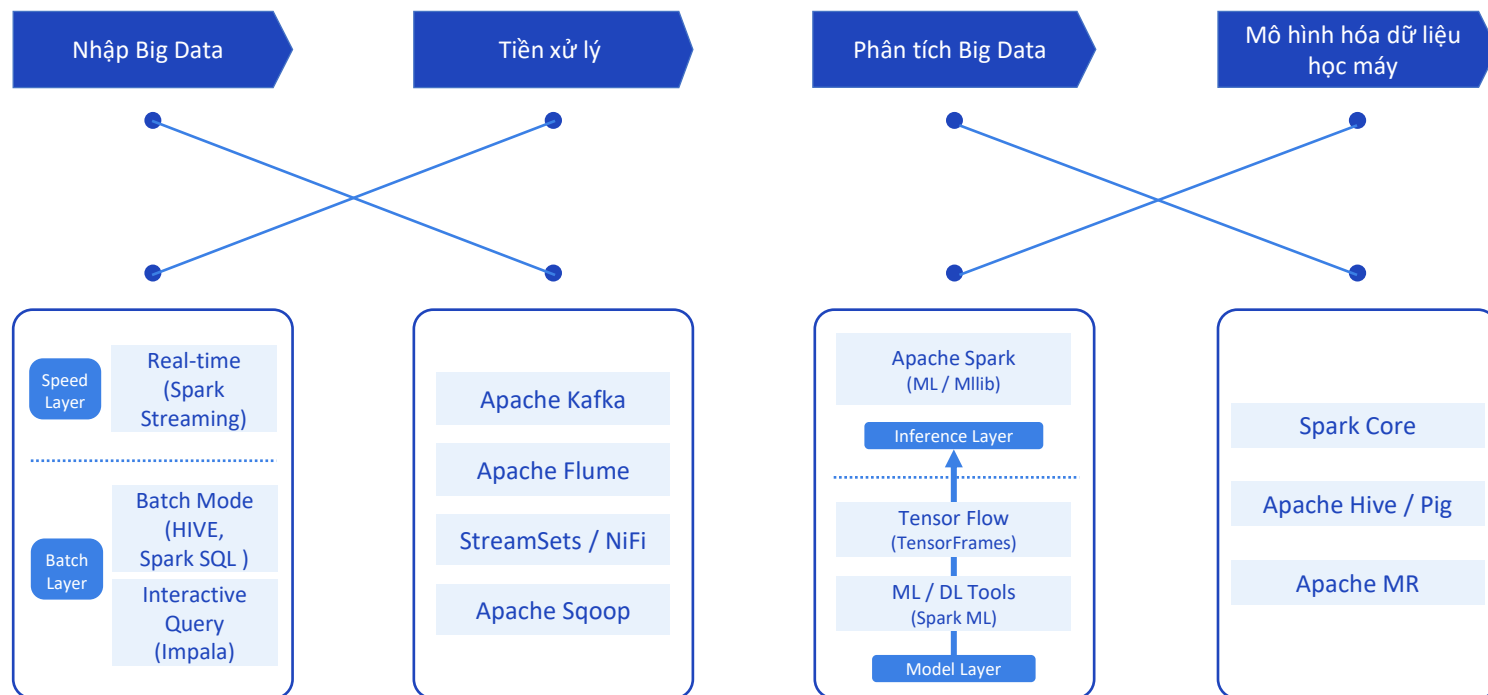
# Câu hỏi ôn tập (2/2)

## 3. Kết nối các công cụ tương ứng với đường dẫn Big Data bên dưới



# Câu hỏi ôn tập (2/2)

## 3. Kết nối các công cụ tương ứng với đường dẫn Big Data bên dưới



Bài 3.

# Kiến trúc Hadoop cho Big Data

Những nguyên tắc cơ bản Big Data



Bài 3

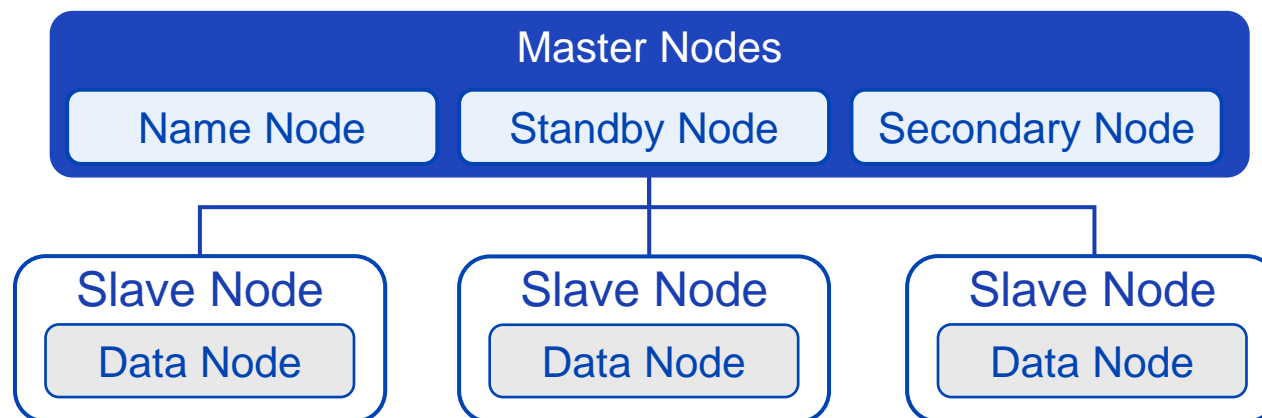
# Kiến trúc Hadoop cho Big Data

| 3.1. Lưu trữ hệ thống tệp phân tán Hadoop

| 3.2. Trình quản lý tài nguyên sori và kiến trúc máy tính

# Tổng quan về kiến trúc HDFS

- I HDFS tuân theo kiến trúc Master – Slave
- I Master Nodes
  - ▶ Chịu trách nhiệm quản lý công việc và lưu giữ hồ sơ siêu dữ liệu
- I Worker / Slave Nodes
  - ▶ Thực hiện đọc và ghi dữ liệu thực tế

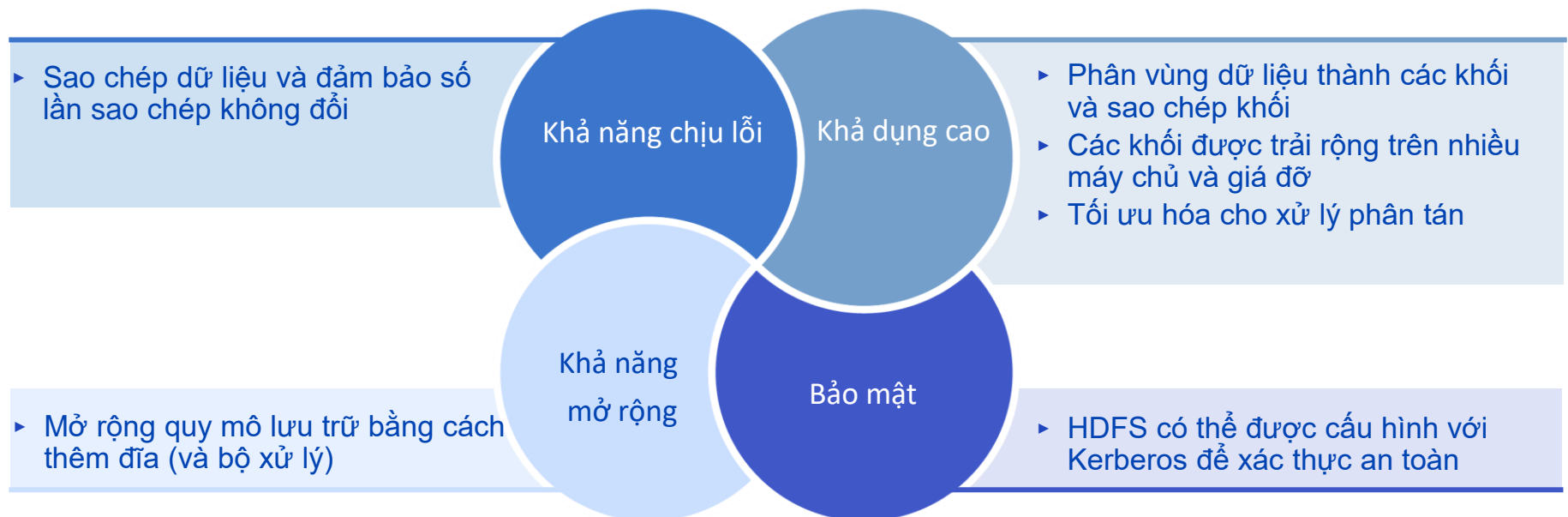


# Các khái niệm cơ bản của HDFS

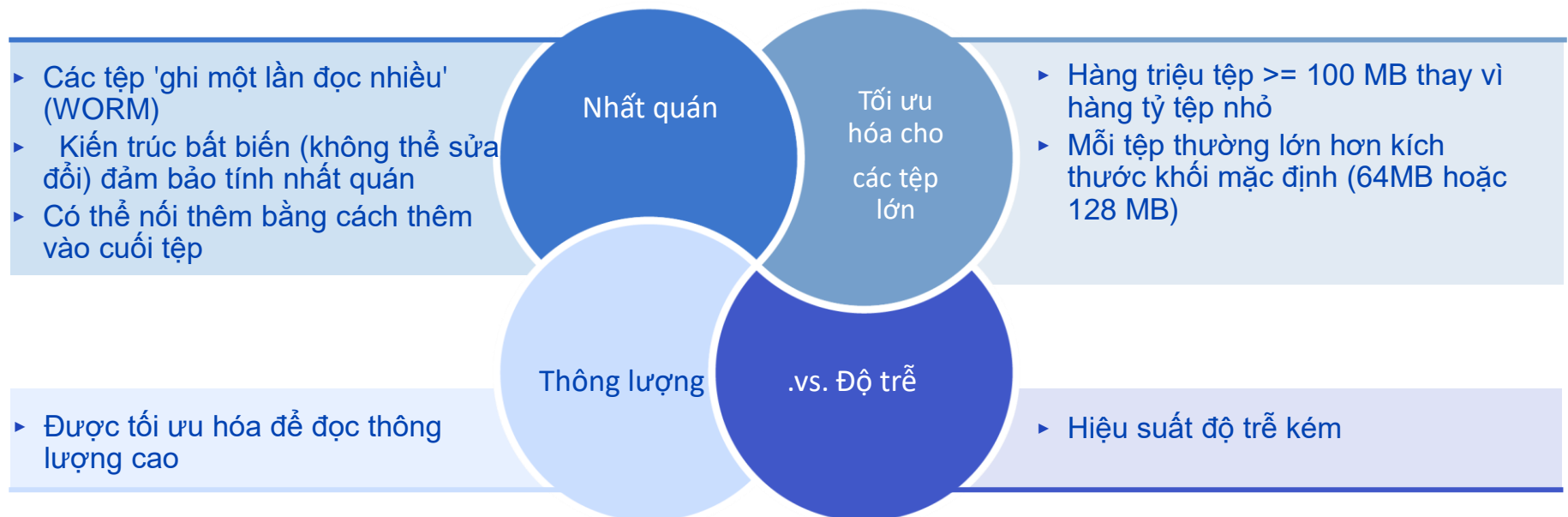
- | Hoạt động trên hệ thống tệp hệ thống (thường là Linux)
  - ▶ Linux ext3, ext4, hoặc xfs
- | Mô phỏng hệ thống tệp Linux với các tệp và thư mục
  - ▶ Có ACL (danh sách kiểm soát truy cập) tương tự như Linux
- | Hoạt động thông qua các ứng dụng Java (daemon)
- | Dựa trên Hệ thống tệp của Google (GFS)
- | Sử dụng phần cứng tiêu chuẩn công nghiệp có thể mở rộng
- | Dữ liệu được phân vùng thành các khối và phân phối trên nhiều máy chủ trong quá trình ghi



# Các tính năng của HDFS (1/2)

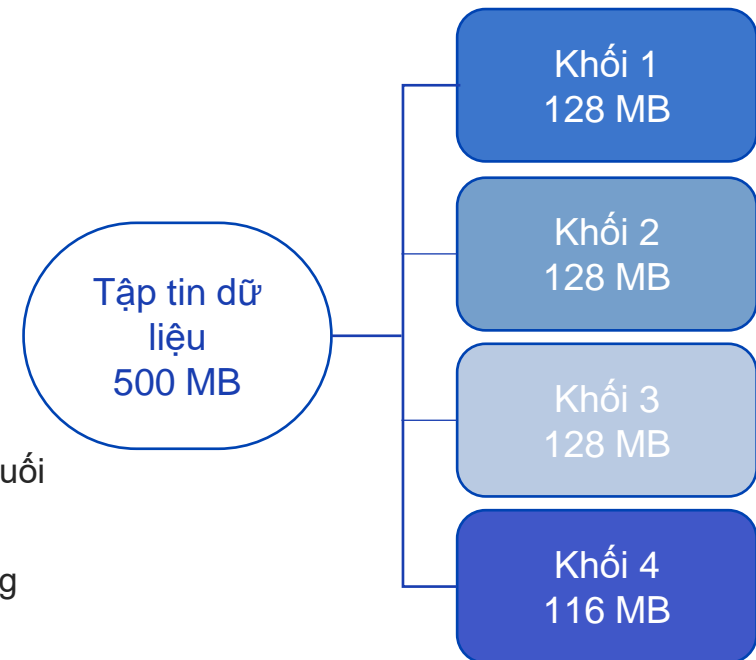


# Các tính năng của HDFS (2/2)



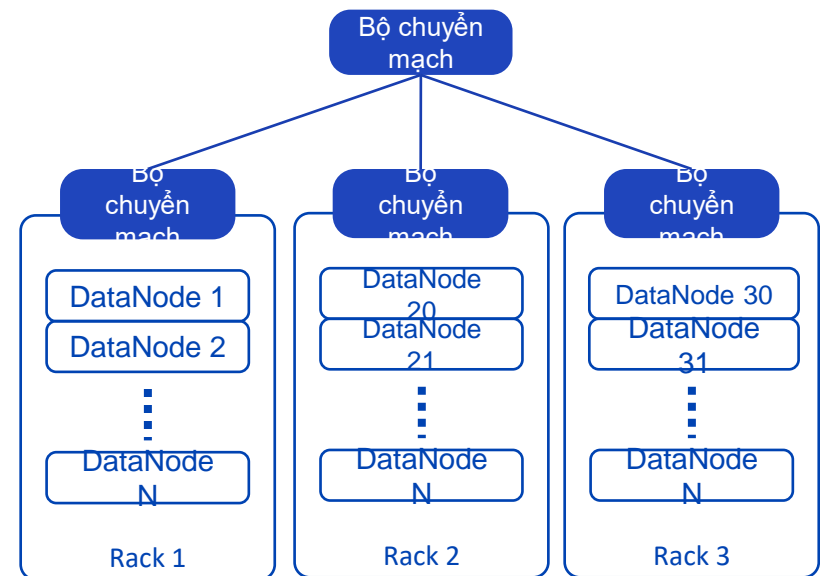
# Phân vùng tệp lớn

- I Các tệp lớn được phân vùng thành các khối nhỏ hơn và được nhân rộng
  - ▶ Tăng tính khả dụng
  - ▶ Truy cập đồng thời để tăng thông lượng
- I Kích thước khối có thể định cấu hình
  - ▶ Vanilla Hadoop → 64 MB
  - ▶ Cloudera Hadoop → 128 MB
- I Các khối không phải là các khe có kích thước cố định
  - ▶ Tất cả các khối trong một tệp có cùng kích thước, ngoại trừ khối cuối cùng
  - ▶ Như có thể thấy trong Khối 4, HDFS chỉ sử dụng nhiều dung lượng đĩa cần thiết



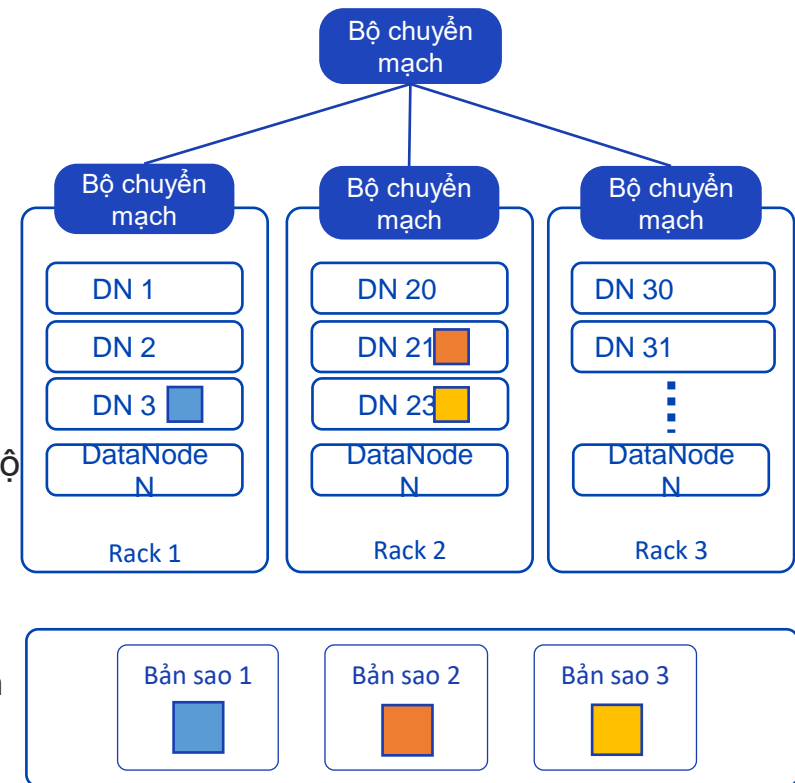
## HDFS is Rack-Aware

- | HDFS có thể được cấu hình để nhận biết cấu trúc liên kết giá cụm
- | Nếu được định cấu hình như vậy, HDFS sẽ biết các máy chủ “gần” với nhau như thế nào
  - ▶ Closest: sẽ ở trên cùng một máy chủ
  - ▶ Closer : Trên cùng một giá đỡ
- | Máy khách đọc các khối dữ liệu từ máy chủ "gần nhất" bất cứ khi nào có thể
- | HDFS sử dụng cấu trúc liên kết giá đỡ trong các thao tác ghi để tăng khả năng chịu lỗi



# Nhân rộng và Rack-Awareness

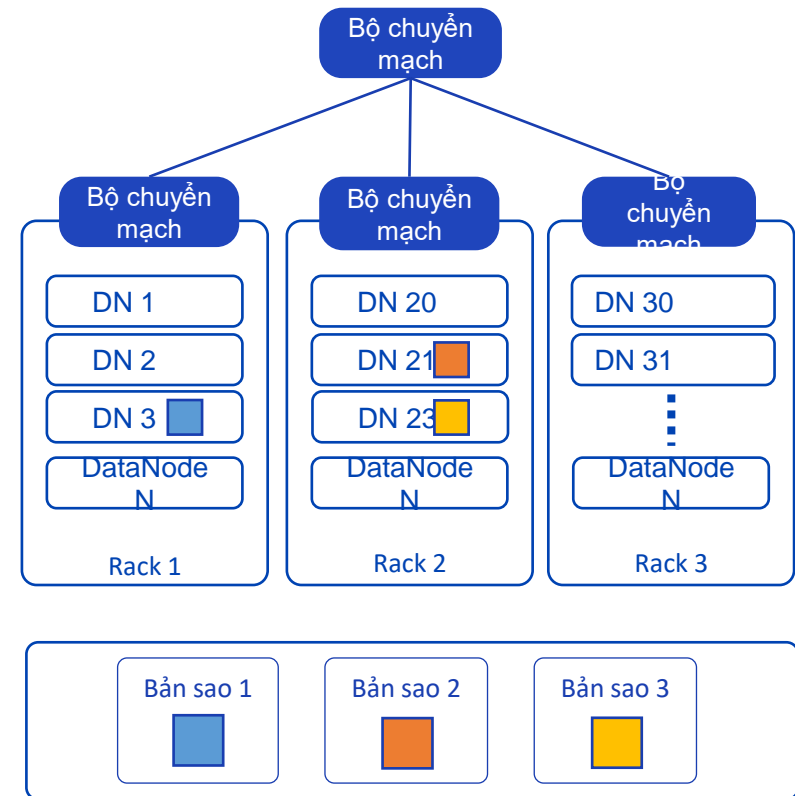
- HDFS phân vùng các tệp lớn thành các khối và sao chép các khối này
  - Số lượng khối được sao chép có thể định cấu hình - hệ số sao chép
  - Hệ số sao chép mặc định là 3
  - Tăng tính khả dụng - 3 vị trí để bắt đầu đọc
  - Tăng độ tin cậy – Có thể bị tổn thất 66% và vẫn phục hồi
- Vị trí của các khối được sao chép, ảnh hưởng lớn đến mức độ chịu lỗi
  - Tất cả các khối trên cùng một đĩa – thực sự không tăng độ tin cậy
  - Tất cả các khối trên cùng một giá – lỗi công tắc giá có thể khiến tất cả các bản sao không thể truy cập được





# Chính sách Rack-Awareness

- | Không có nhiều hơn một bản sao được đặt trên một nút
- | Không quá hai bản sao được đặt trên cùng một giá đỡ
- | Số lượng giá đỡ được sử dụng để sao chép khối phải ít hơn số lần sao chép
- | Ví dụ: Hệ số nhân rộng = 3
  - ▶ Sử dụng ít hơn 3 rack– nên 1 hoặc 2 rack
  - ▶ Không quá 2 bản sao trên cùng một rack nên không thể trên 1 rack
  - ▶ Đặt một bản sao trên 1 rack
  - ▶ Đặt 2 bản sao còn lại trên rack thứ hai



# Thành phần HDFS

## I Trình nền Master / Slave

- ▶ Trình nền là các quy trình tồn tại lâu dài thường được kích hoạt khi bật máy chủ

## I Name Node

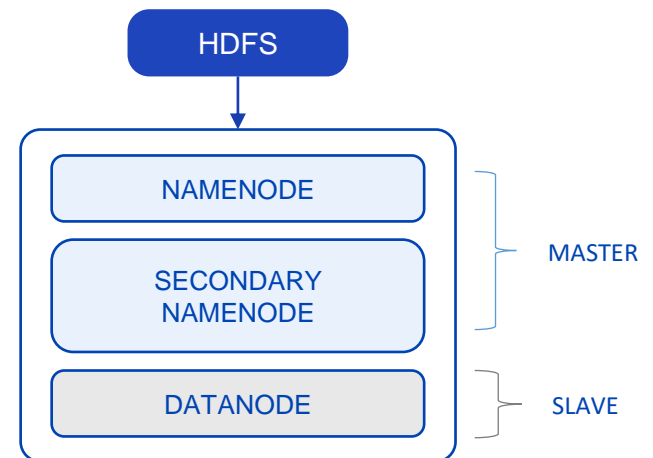
- ▶ Trình nền chủ đang hoạt động
- ▶ Xác định và duy trì cách các khối dữ liệu được phân phối trên các DataNodes
- ▶ Không tham gia vào hoạt động đọc/ghi thực tế

## I Secondary / Standby Node

- ▶ Trình nền chủ thụ động
- ▶ Chịu trách nhiệm duy trì độ bền, siêu dữ liệu mà Name Node lưu trữ trong bộ nhớ

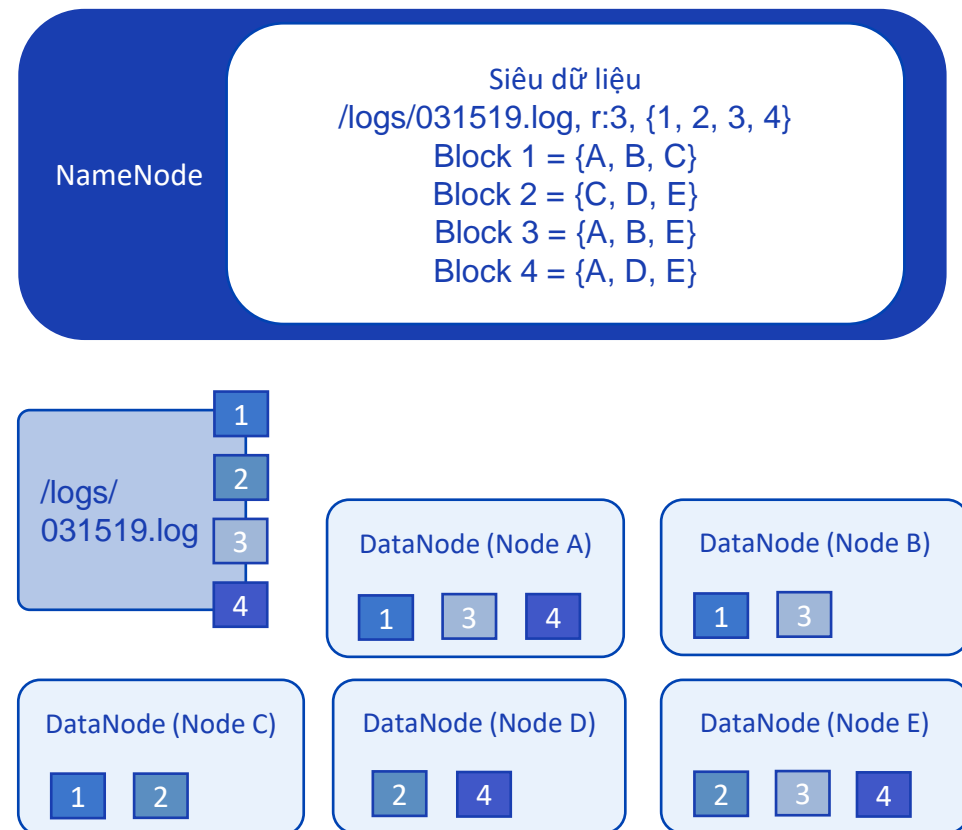
## I DataNode

- ▶ Đọc và ghi các khối dữ liệu
- ▶ Chịu trách nhiệm sao chép các khối trên các DataNode khác



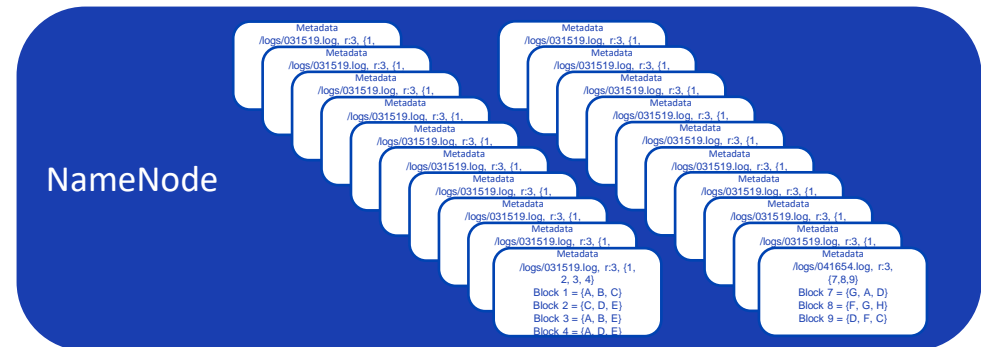
# HDFS NameNode

- I NameNode chịu trách nhiệm giữ siêu dữ liệu namespace
  - ▶ Siêu dữ liệu namespace bao gồm liên kết giữa các tệp, đó là các khối và vị trí của các khối đó
  - ▶ Siêu dữ liệu được lưu trữ trong bộ nhớ để có hiệu suất nhanh hơn
- I Siêu dữ liệu cũng được lưu trữ trên đĩa để đảm bảo độ bền
  - ▶ Được lưu trữ trong tệp fsimage
  - ▶ fsimage không lưu trữ vị trí khối
  - ▶ Các thay đổi đối với siêu dữ liệu được ghi vào tệp nhật ký chỉnh sửa



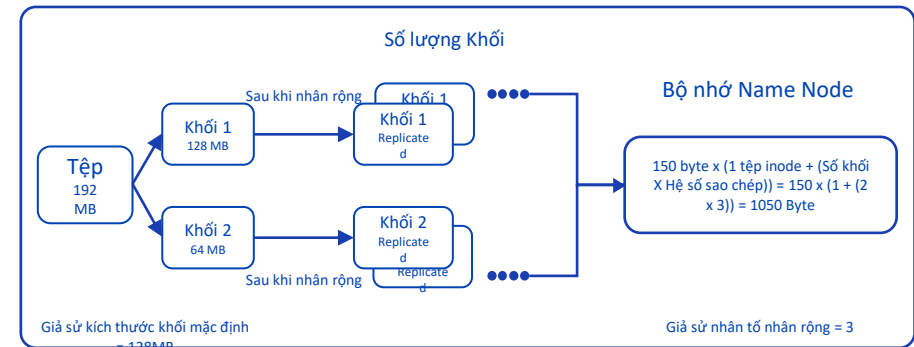
# Cấp phát bộ nhớ NameNode

- Khi NameNode đang chạy, tất cả siêu dữ liệu được giữ trong phần hồi nhanh của RAM
- NameNode có Kích thước Heap Java mặc định là 1 GB
- NameNode lưu trữ siêu dữ liệu sau
  - ▶ Thông tin tệp - tên tệp, quyền sở hữu, quyền, v.v.
  - ▶ Thông tin khối - đối với mỗi khối là một phần của tệp, tên khối và vị trí
  - ▶ Mỗi mục sử dụng từ 150 đến 250 byte bộ nhớ

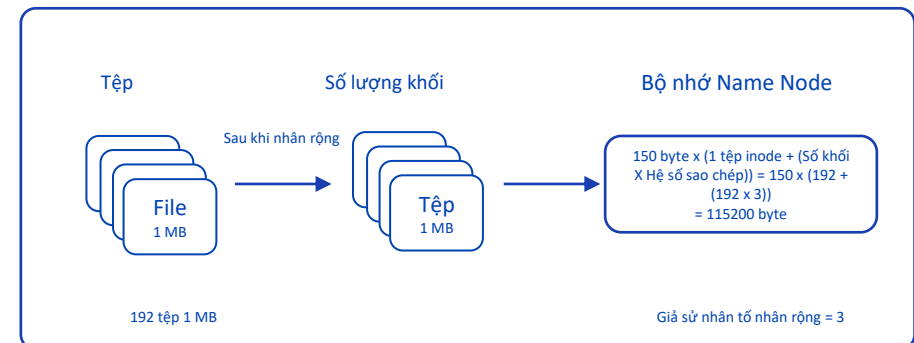


# Sự cố tệp nhỏ NameNode

- I HDFS bị vấn đề về tệp nhỏ
- I Đây là khi một phần lớn các tệp trong HDFS là các tệp nhỏ
  - ▶ HDFS NameNode có thể hết bộ nhớ Java trước khi hết dung lượng đĩa
- I Ví dụ: 1 GB dữ liệu được lưu trữ dưới dạng một tệp .vs. 1000 tệp 1 MB
- I Tệp 1 GB duy nhất có kích thước khối là 128 MB
  - ▶ 1 Thông tin tệp và 8 Thông tin khối
  - ▶ Tổng cộng 8 mục trong bộ nhớ
- I Tệp 1000 x 1 MB với kích thước khối là 128 MB
  - ▶ Thông tin 1000 tệp và 1000 thông tin khối
  - ▶ Tổng số 2000 mục trong bộ nhớ

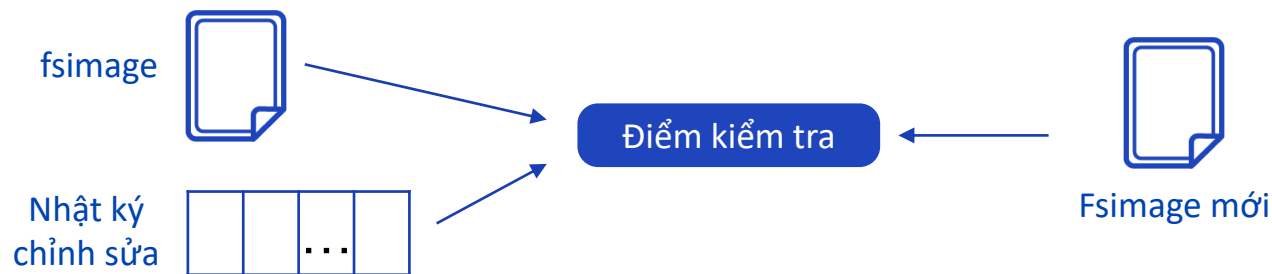


Tình huống 2 (192 tệp nhỏ, mỗi tệp 1MiB):



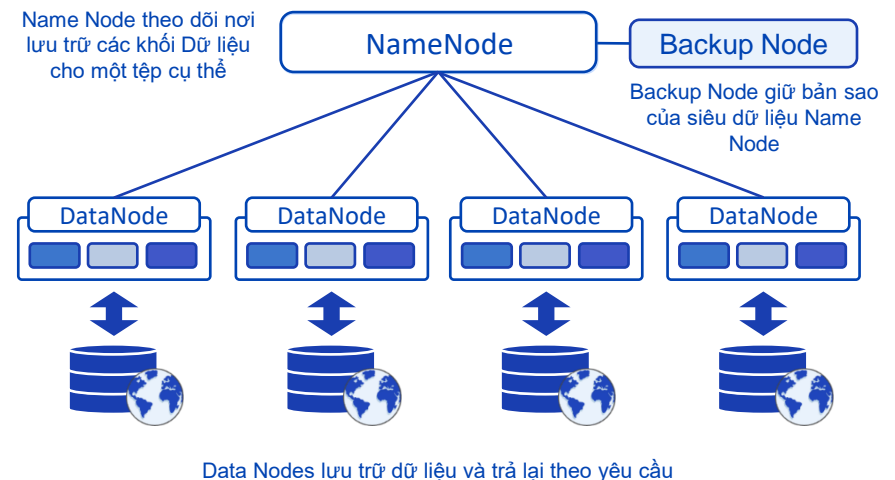
## HDFS Secondary NameNode

- I NameNode phụ, mặc dù đúng với tên của nó, nhưng không phải là daemon chuyển đổi dự phòng cho NameNode
- I Vai trò chính của nó là kiểm tra siêu dữ liệu không gian tên
  - ▶ Nhận một bản sao của fsimage và chỉnh sửa tệp nhật ký mới nhất từ NameNode
  - ▶ Hợp nhất các thay đổi trong tệp nhật ký chỉnh sửa vào fsimage
  - ▶ Gửi bản sao mới của fsimage tới NameNode
  - ▶ NameNode cập nhật fsimage của nó và đặt lại tệp nhật ký chỉnh sửa



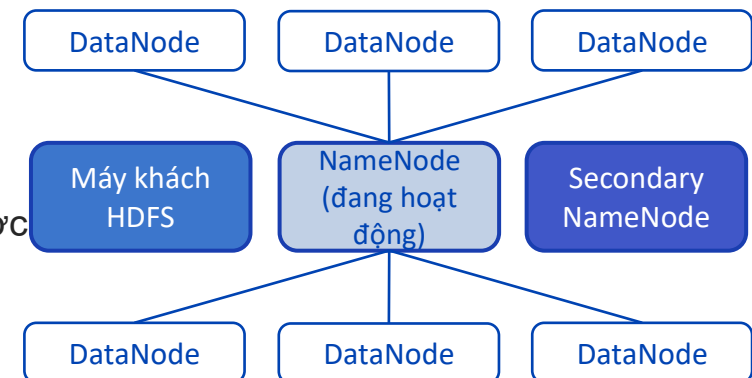
# HDFS DataNodes

- | DataNode là những công nhân thực tế chịu trách nhiệm viết và đọc các khối dữ liệu
- | Khi viết các khối, các DataNode giao tiếp với nhau để sao chép khối theo hệ số sao chép
  - ▶ Các khối được sao chép theo kiểu đường ống
  - ▶ NameNode không phải là một phần của quá trình ghi.
  - ▶ Vai trò duy nhất của nó là ánh xạ các khối tới các nút dữ liệu
- | Các khối được lưu dưới dạng tệp trong hệ thống tệp cơ bản
  - ▶ Vị trí lưu khối có thể được định cấu hình
  - ▶ Các khối được đặt tên là blk\_xxxxxxxx
- | DataNodes không biết khối thuộc về tệp nào



## HDFS ở Chế độ không có tính khả dụng cao

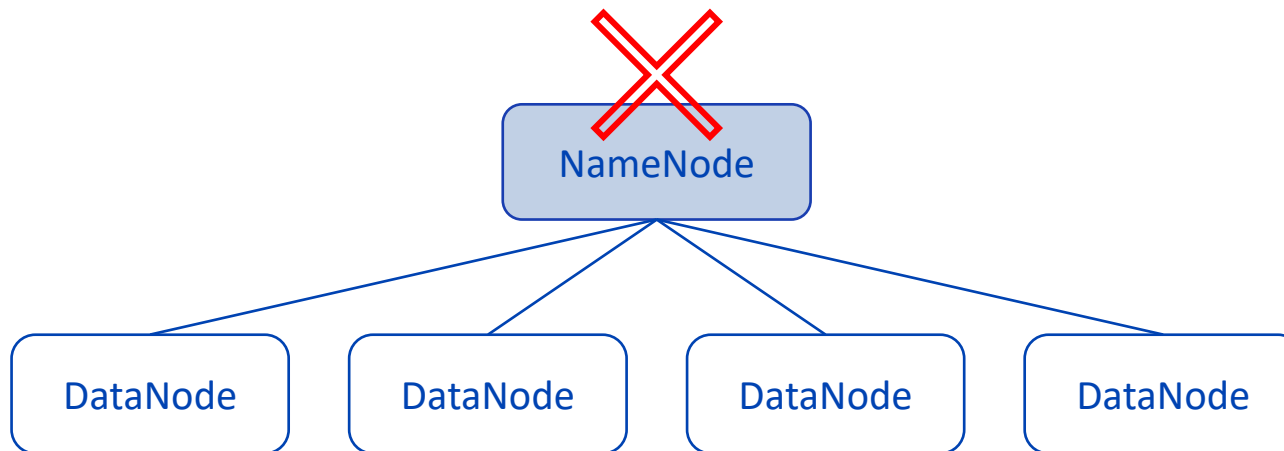
- I HDFS có thể được triển khai ở chế độ High Availability (HA) hay không
- I HDFS có thể được triển khai ở chế độ High Availability (HA) hay không
  - ▶ NameNode (Chủ)
  - ▶ Secondary NameNode (Chủ)
  - ▶ DataNode (Tớ)
- I Ở chế độ không HA, nếu NameNode bị lỗi, hệ thống phải được khởi động lại với NameNode phụ thay thế cho NameNode
  - ▶ Khi khởi động lại, NameNode mới đọc tệp fsimage
  - ▶ Tất cả các datanode gửi thông tin vị trí khối đến NameNode



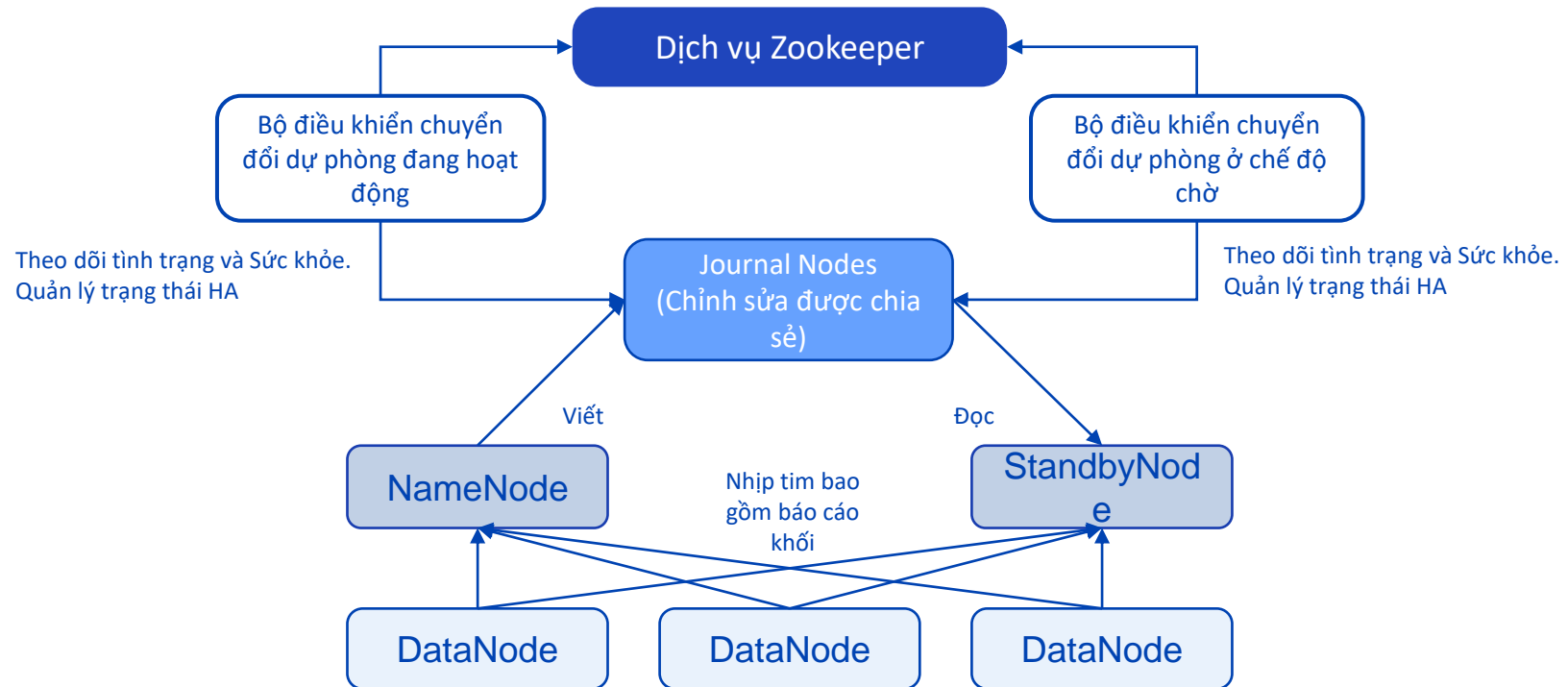


# Single Point of Failure (SPOF): Điểm lỗi duy nhất

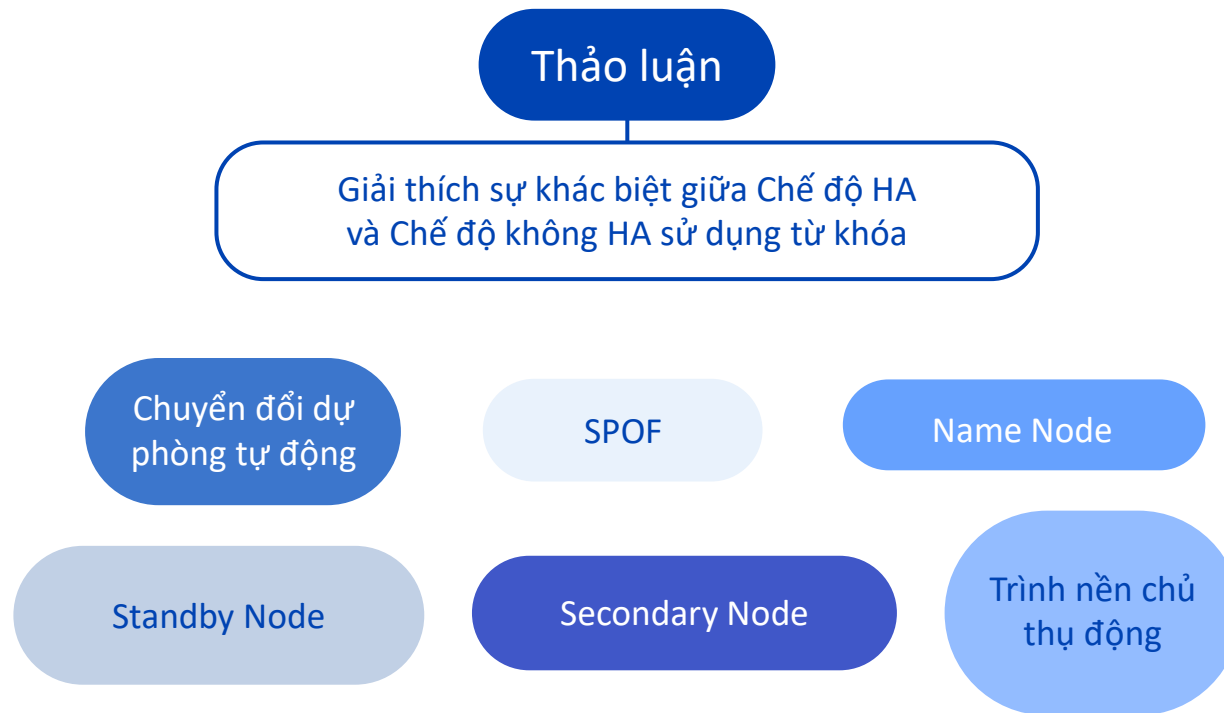
- I Một cụm Hadoop có một NameNode đang hoạt động
  - ▶ Nếu tại bất kỳ thời điểm nào, thực sự có nhiều namenode đang hoạt động, thì cụm đó sẽ bị chia tách nào
- I Nếu NameNode đang hoạt động bị lỗi, toàn bộ dịch vụ HDFS sẽ không thể truy cập được
  - ▶ Đây là một điểm thất bại duy nhất
  - ▶ HDFS sẽ không khả dụng cho đến khi NameNode được khởi động lại



# HDFS in High Availability Mode



## HDFS ở Chế độ sẵn sàng cao



# DataNode Thất bại và phục hồi

- I DataNode gửi nhịp tim đến NameNode theo định kỳ
  - ▶ Tần suất có thể được cấu hình với mặc định là 3 giây
- I Nếu nhịp tim không được nhận từ DataNode, nó sẽ tiến triển qua nhiều giai đoạn và cuối cùng được tuyên bố là đã chết
  - ▶ Ban đầu được tuyên bố là cũ sau 30 giây
  - ▶ Được tuyên bố là đã chết sau 10,5 phút
- I NameNode loại trừ các DataNode cũ tham gia vào quá trình ghi mới
- I Sau khi được tuyên bố là đã chết, tất cả các khối trên DataNode đã chết được coi là bị mất
  - ▶ Điều này sẽ kích hoạt quy trình khôi phục chưa được sao chép và các khối trên nút dữ liệu chết sẽ được sao chép lại trên các nút dữ liệu khác
  - ▶ Nếu một DataNode tham gia lại cụm sau một thời gian chết, NameNode đảm bảo rằng các khối đó không bị sao chép quá mức bằng cách xóa chúng

# Quyền đối với tệp HDFS

- I Các tệp và thư mục HDFS có quyền rất giống với Linux
  - ▶ Linux như quyền đọc (r), viết (w) và thực thi (x) cho chủ sở hữu, nhóm và những người khác
  - ▶ Mỗi quyền được thể hiện dưới dạng bit – 1 để cho phép và 0 để từ chối
  - ▶ HDFS cũng đặt quyền cho chủ sở hữu, nhóm và những người khác
- I Tuy nhiên, trong HDFS, không có quyền thực thi (x) đối với tệp hoặc thư mục
  - ▶ Đối với các thư mục, quyền thực thi (x) cho phép truy cập vào các thư mục con
  - ▶ Đối với tệp, quyền này bị bỏ qua
- I Thay vào đó, HDFS cũng có thể được đặt khi bật ACL
  - ▶ ACL cho phép thực thi với mức độ chi tiết tốt hơn nhiều so với chủ sở hữu, nhóm và những người khác
  - ▶ ACL cho phép kiểm soát quyền truy cập của từng người dùng, nhiều nhóm, v.v.

```
$ hdfs dfs -ls .  
drwxr-xr-x – owner group 0 2016-04-02 22:10 /user/owner/test  
-rw-rw-r-- owner group 110 2016-04-22 22:15 /user/owner/test/t.txt
```

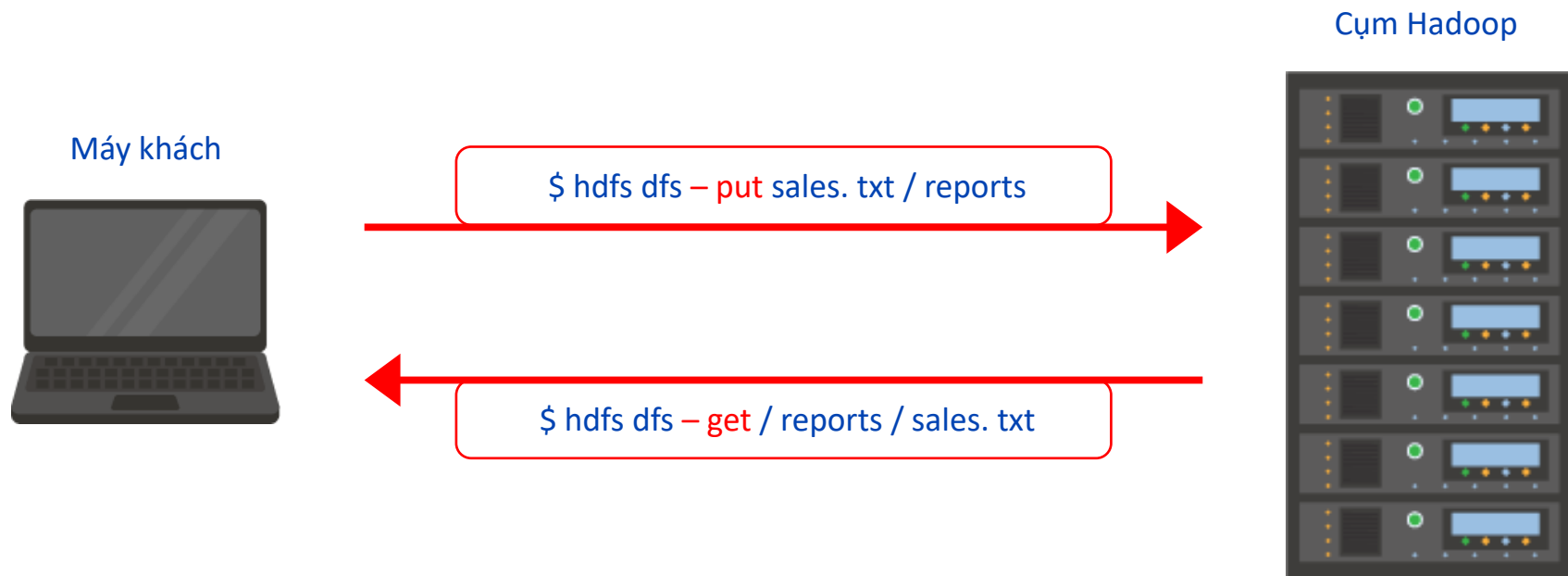
# Truy cập HDFS từ Dòng lệnh

- I Mặc dù các khối dữ liệu cho một tệp được lưu trữ trong hệ thống tệp Linux, nhưng chúng ta không thể truy cập trực tiếp vào tệp đó
- I Từ dòng lệnh, sử dụng `hdfs dfs -<lệnh phụ> <tùy chọn> <tham số>`
- I Danh sách các lệnh con thường được sử dụng
  - ▶ `ls` – liệt kê nội dung của một đạo diễn hdfs
  - ▶ `cp` – sao chép tệp hdfs
  - ▶ `mv` – di chuyển tệp hdfs
  - ▶ `mkdir` – tạo một thư mục hdfs
  - ▶ `put` - sao chép tệp Linux cục bộ sang hdfs
  - ▶ `get` – sao chép tệp hdfs sang tệp Linux

```
$ hdfs dfs -put input.txt input.txt  
$ hdfs dfs -ls /  
$ hdfs dfs -rm /reports/sales.txt
```

# Sao chép dữ liệu giữa HDFS và Linux

I `hdfs dfs <-put> / <-get>`



## Các lệnh Shell của hệ thống tệp tin HDFS

Lệnh	Mô tả
ls	liệt kê nội dung của các thư mục
du	hiển thị việc sử dụng đĩa
count	đếm số lượng thư mục, tệp và byte trong một đường dẫn
chgrp, chown, chmod	thay đổi quyền truy cập tệp tin và thư mục
stat	in số liệu thống kê về một tệp tin hoặc thư mục
cat, text	hiển thị nội dung của các tệp tin
tail	hiển thị 1KB cuối cùng của nội dung tệp
get, copyToLocal	các lệnh giống hệt nhau sao chép tệp từ HDFS sang hệ thống tệp cục bộ
put, copyFromLocal	các lệnh giống hệt nhau sao chép tệp từ hệ thống tệp cục bộ vào HDFS
getmerge	lấy một tập hợp các tệp và hợp nhất chúng thành một tệp duy nhất

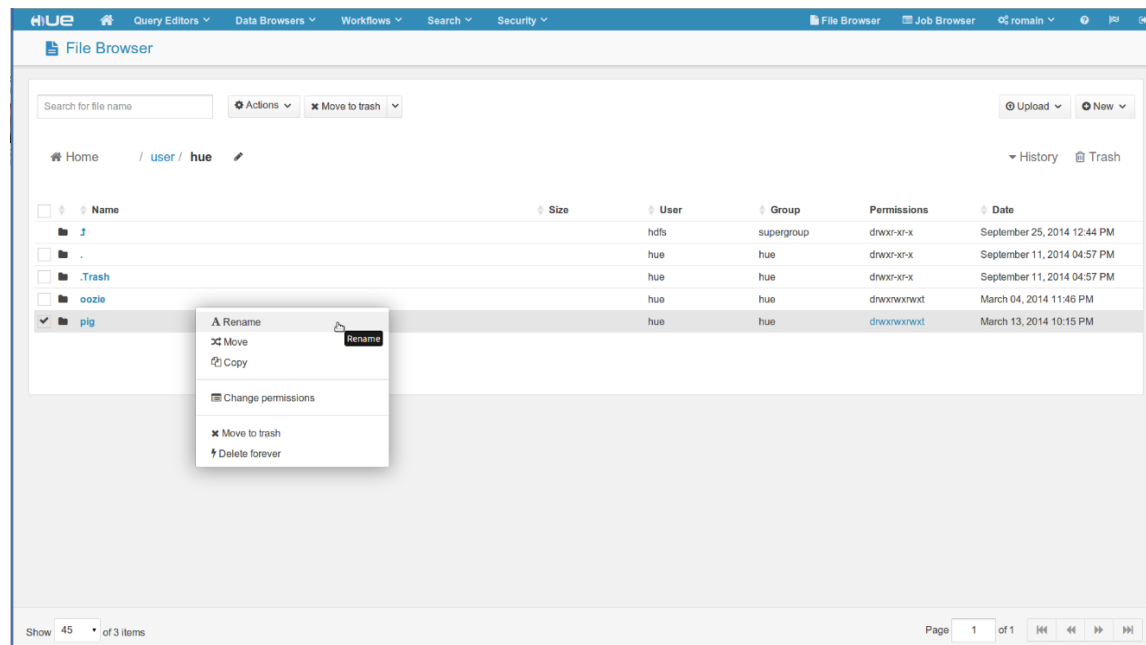


## Các lệnh Shell của hệ thống tệp tin HDFS

Lệnh	Mô tả
mv	di chuyển một tệp trong HDFS
cp	sao chép tệp sang vị trí khác trong HDFS
mkdir	tạo một thư mục mới trong HDFS
rm	xóa một tệp (vào thư mục Thùng rác)
rm -R	xóa các thư mục và xóa đệ quy mọi tệp và thư mục con (vào thư mục Thùng rác)
test	kiểm tra nếu một tệp tin tồn tại
touch	ghi dấu thời gian vào một tệp mới
expunge	làm trống thư mục Thùng rác của người dùng

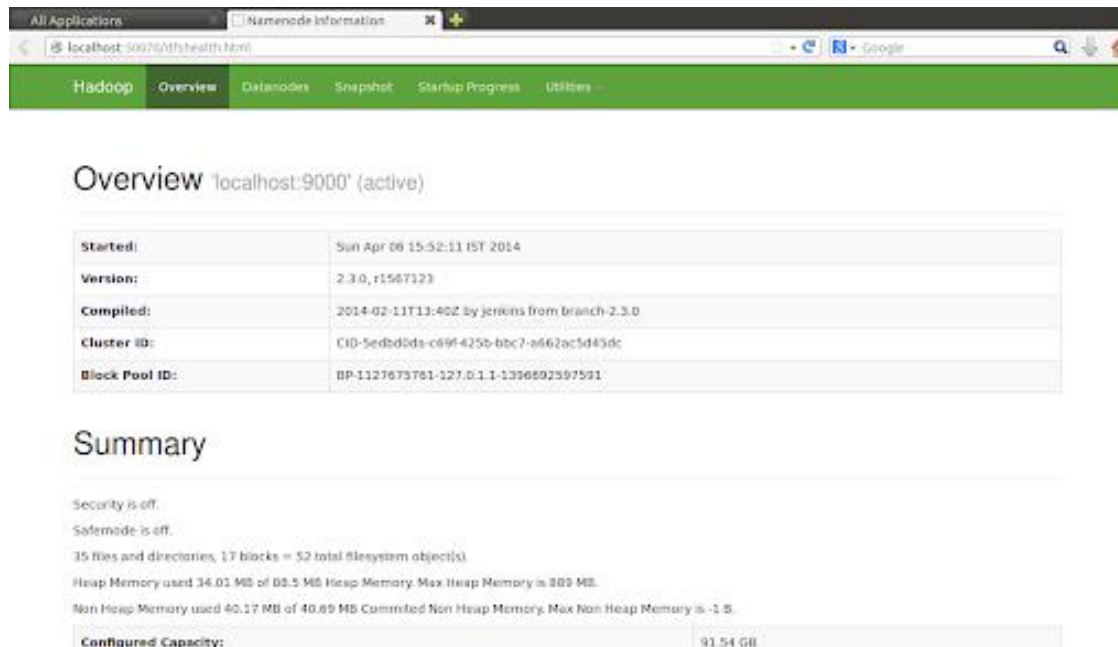
# Truy cập HDFS từ HUE

- Trải nghiệm người dùng Hadoop (HUE) là một công cụ dựa trên GUI để truy cập HDFS
  - Tạo, di chuyển, đổi tên, sửa đổi, tải lên, tải xuống và xóa các thư mục và tệp
  - Xem nội dung tệp văn bản



# Giao diện người dùng web NameNode

- Nhận báo cáo và tình trạng sức khỏe của dịch vụ HDFS
- Dung lượng DataNode và tình trạng sức khỏe
- Thông tin siêu dữ liệu tệp – ID khối, vị trí, v.v.



Bài 3

# Kiến trúc Hadoop cho Big Data

| 3.1. Lưu trữ hệ thống tệp phân tán Hadoop

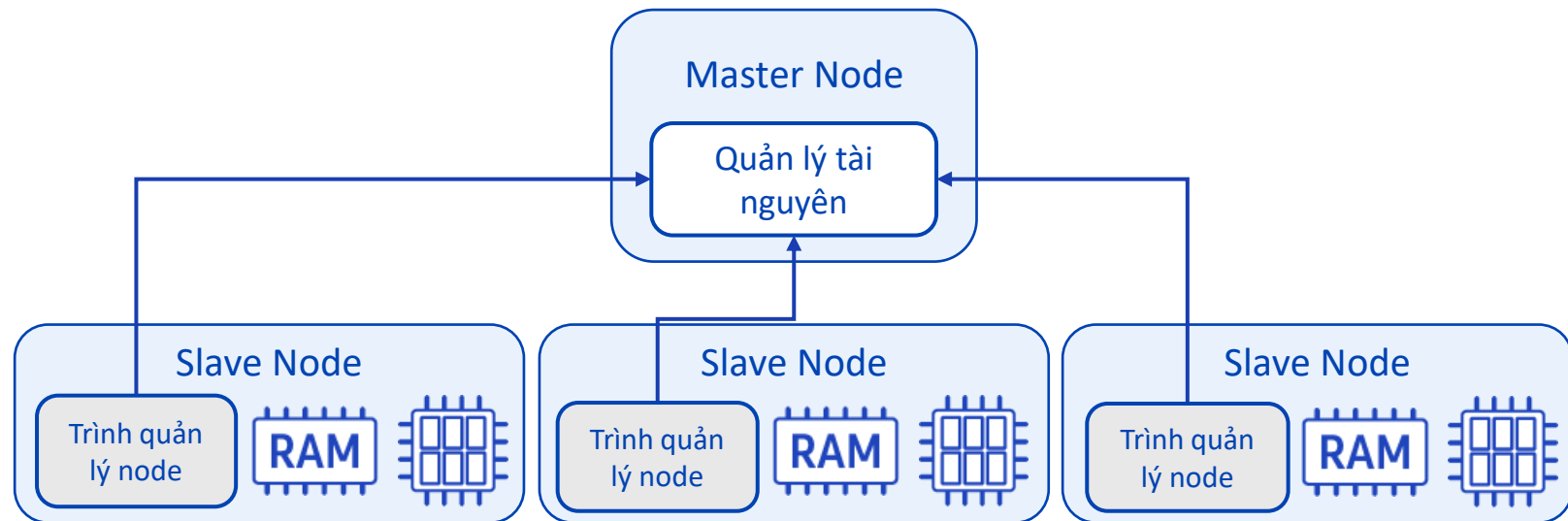
| 3.2. Trình quản lý tài nguyên Yarn và kiến trúc máy tính

# YARN là gì?

- | Một nền tảng để quản lý tài nguyên trong cụm Hadoop
- | Tuy nhiên, một nhà đàm phán tài nguyên khác
- | Kiến trúc Master/Slave
- | Khung cho các ứng dụng xử lý phân tán trên cụm để thực thi bằng tài nguyên cụm
  - ▶ MapReduce v2
  - ▶ Spark
  - ▶ Impala
  - ▶ Tìm kiếm
  - ▶ Truy vấn SQL
  - ▶ Học máy
  - ▶ Xử lý truyền phát

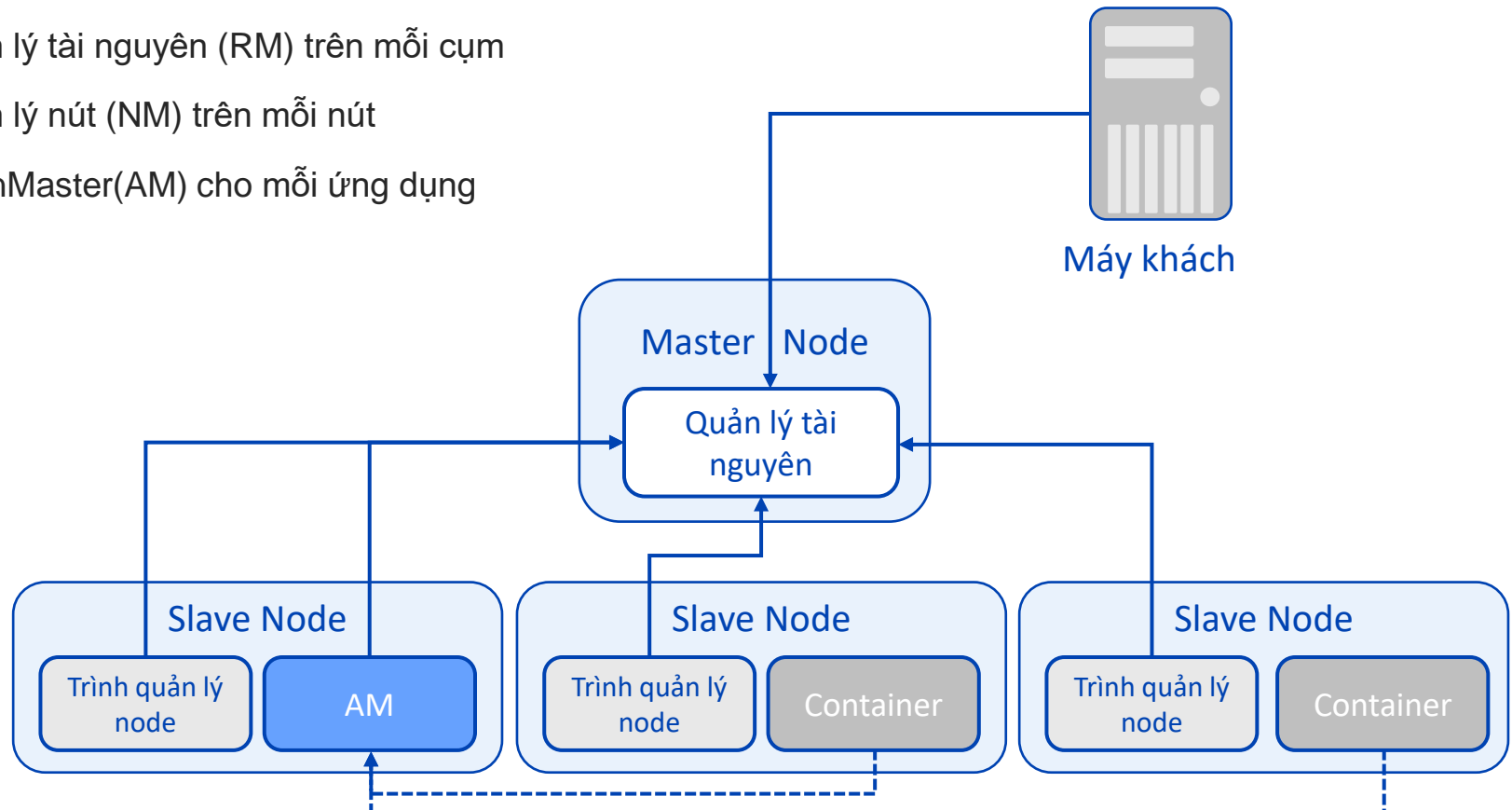
# Tại sao lại là YARN?

- YARN cho phép bạn chạy các khối lượng công việc đa dạng trên cùng một cụm Hadoop
- Cho phép chia sẻ động bộ nhớ cụm và tài nguyên CPU giữa các ứng dụng
- Tăng cường sử dụng cụm



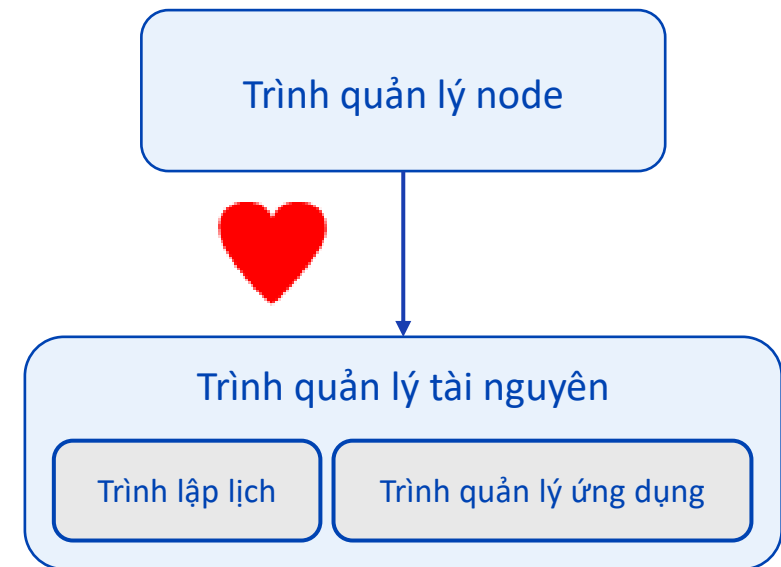
# Trình nền YARN

- | Trình quản lý tài nguyên (RM) trên mỗi cụm
- | Trình quản lý nút (NM) trên mỗi nút
- | ApplicationMaster(AM) cho mỗi ứng dụng



# Trình quản lý tài nguyên (RM) YARN

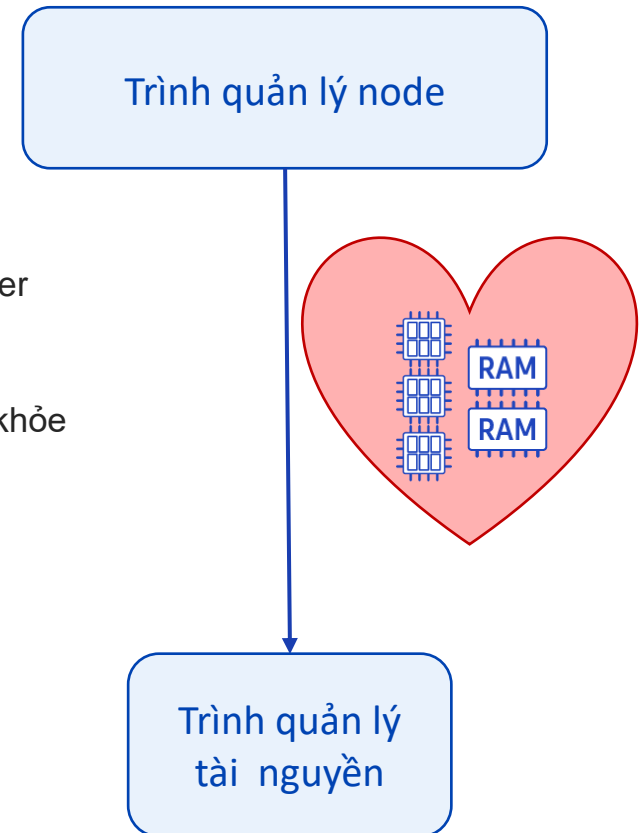
- I Trình nền master
  - I Hai thành phần chính: Scheduler và ApplicationsManager
  - I Scheduler (Trình lập lịch):
    - ▶ Chịu trách nhiệm phân bổ tài nguyên cho các ứng dụng đang chạy khác nhau
    - ▶ Chính sách có thể cấm – CapacityScheduler, FairScheduler
  - I ApplicationsManager (Trình quản lý ứng dụng)
    - ▶ Chịu trách nhiệm tiếp nhận hồ sơ công việc
    - ▶ Đàm phán vùng chứa cho ApplicationMaster trên mỗi ứng dụng và theo dõi nhịp tim của nó
  - I Theo dõi nhịp tim từ Trình quản lý node





# Trình quản lý node YARN

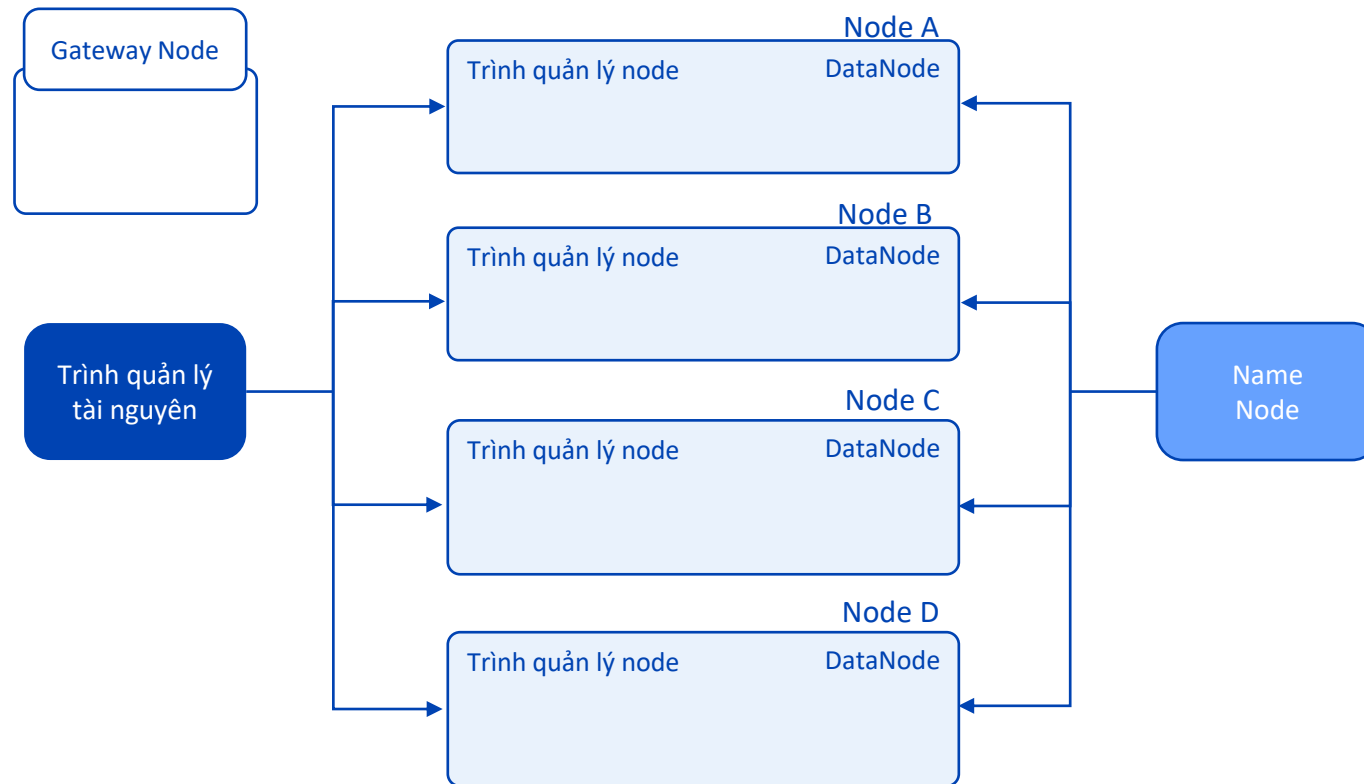
- | Trình nền worker
- | Chạy trên worker node và thường cùng với DataNode
- | Chịu trách nhiệm khởi chạy và quản lý các thùng chứa cho ứng dụng
  - ▶ Các bộ chứa thực thi các tác vụ theo chỉ định của mỗi ứng dụng AppMaster
- | Theo dõi sức khỏe và tài nguyên của node mà nó đang chạy trên đó
  - ▶ Gửi nhịp tim tới Trình quản lý tài nguyên với trạng thái tài nguyên và sức khỏe của node hiện tại



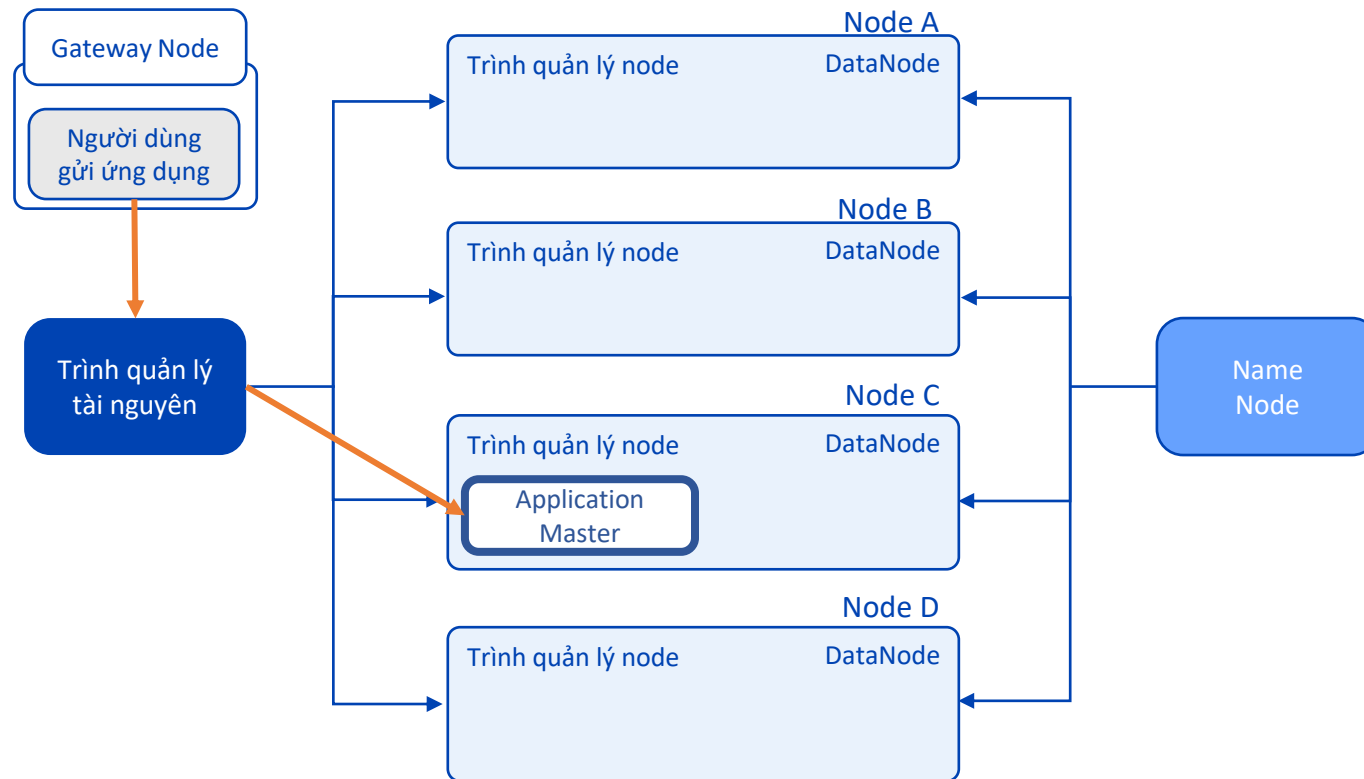
# Khả năng chịu lỗi của YARN

- | Có nhiều trình nền liên quan đến việc chạy một ứng dụng
- | YARN xử lý các trường hợp ngoại lệ như thế nào?
- | Các tác vụ trên Trình quản lý node thoát với ngoại lệ hoặc ngừng gửi nhịp tim
  - ▶ ApplicationMaster thử lại tác vụ trên một vùng chứa mới trên một nút khác
  - ▶ Thử lại tối đa 4 lần
- | ApplicationMaster ngừng gửi nhịp tim tới YARN
  - ▶ Trình quản lý tài nguyên bắt đầu một ApplicationMaster mới và thử lại toàn bộ ứng dụng
  - ▶ Thử lại tối đa 2 lần
- | Trình quản lý tài nguyên chết
  - ▶ Có thể chạy ResourceManager ở chế độ HA
  - ▶ Ở chế độ HA, Trình quản lý tài nguyên dự phòng sẽ tự động chuyển đổi dự phòng

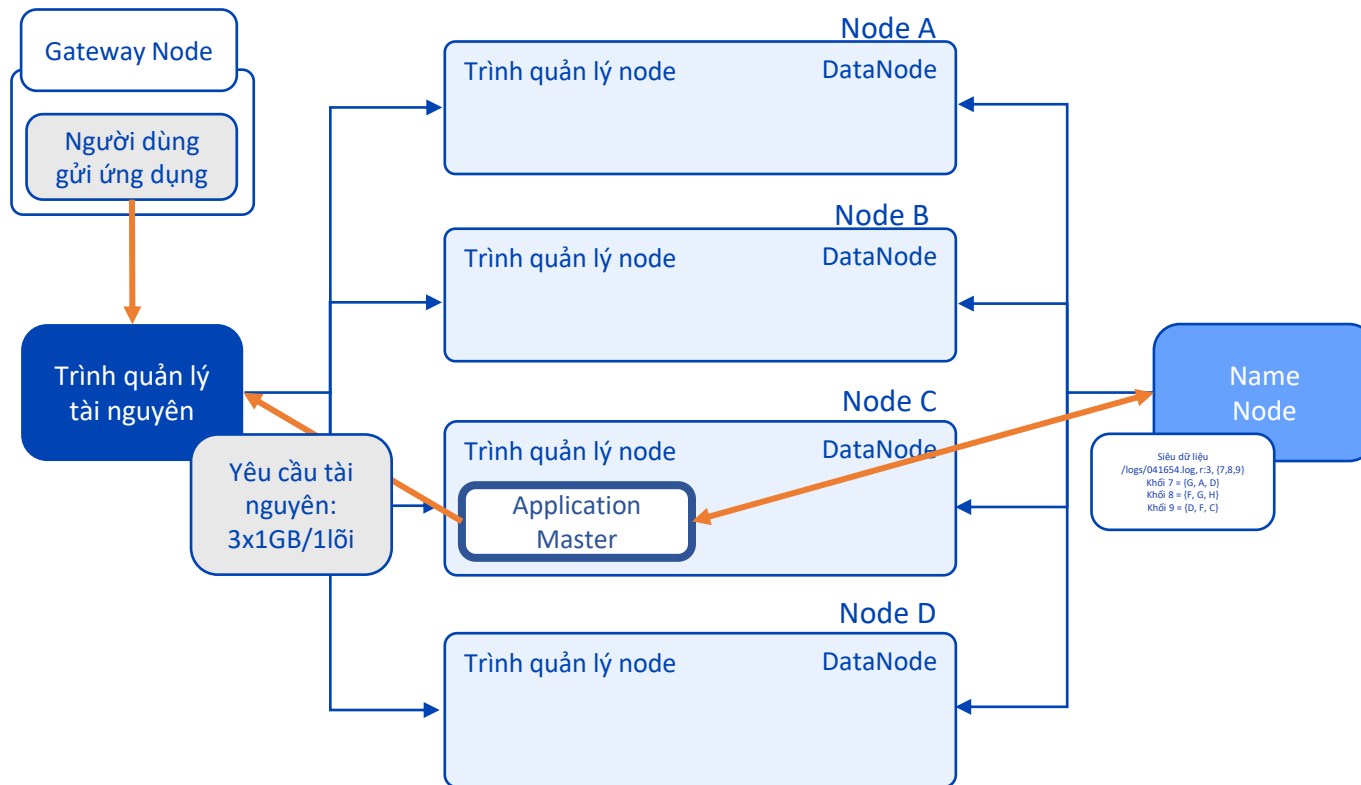
# Chạy ứng dụng trong YARN



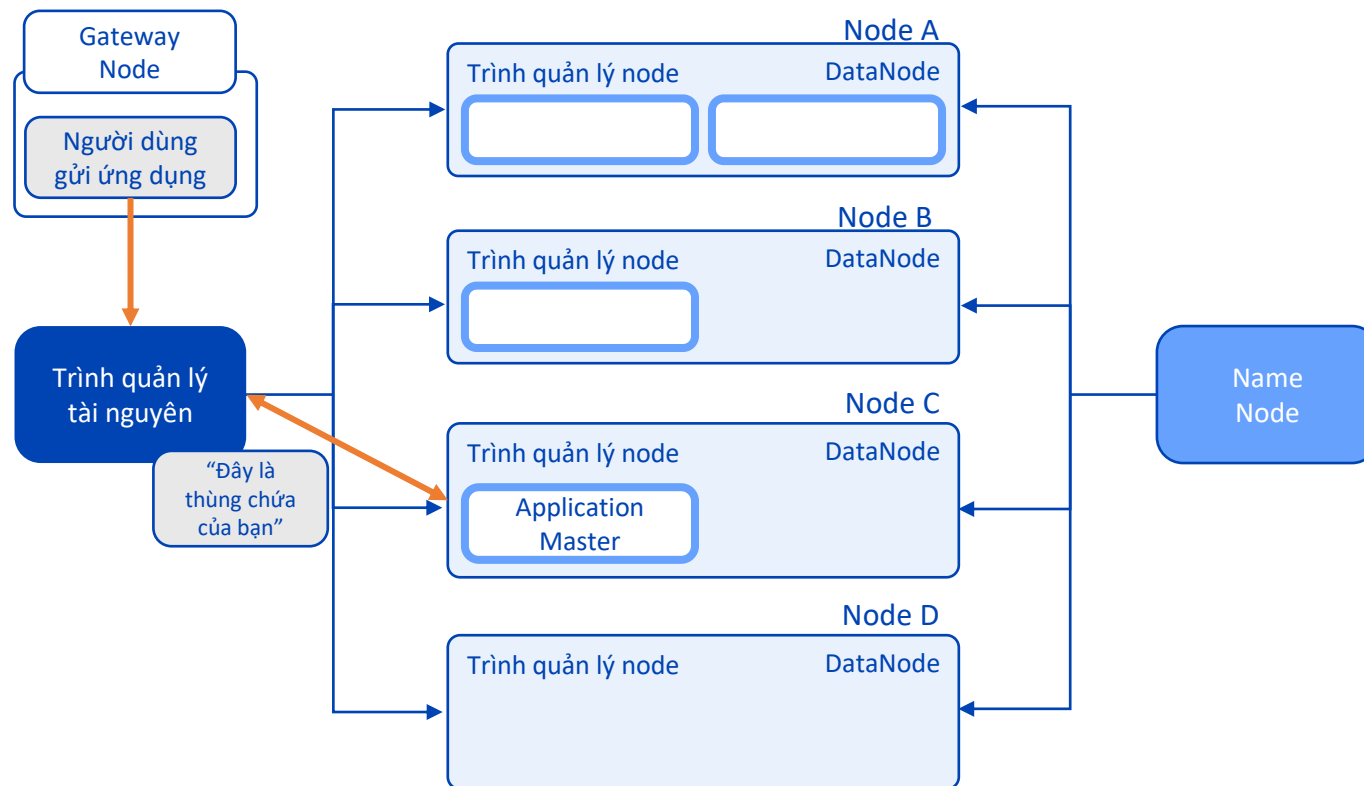
# Chạy ứng dụng trong YARN



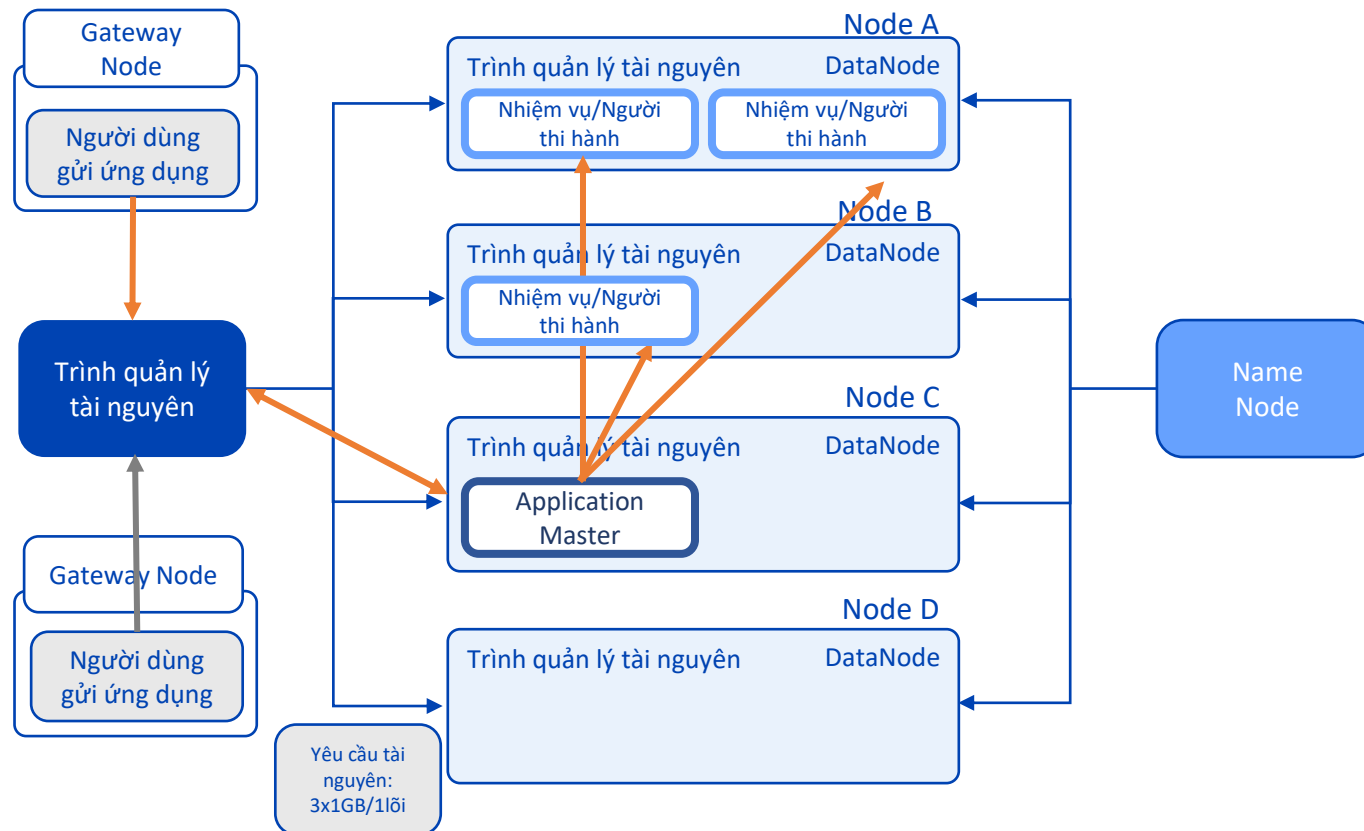
# Chạy ứng dụng trong YARN



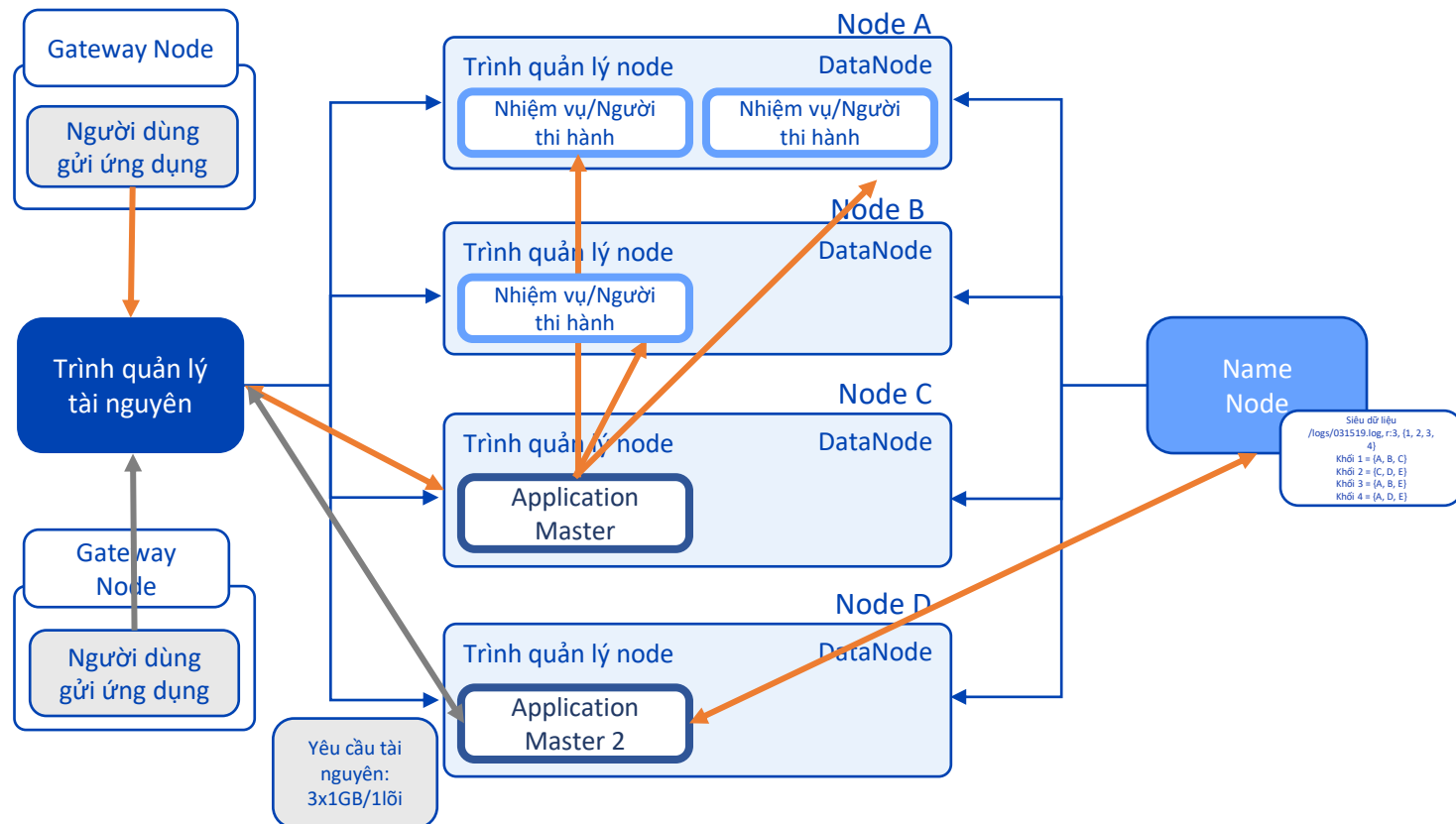
# Chạy ứng dụng trong YARN



# Chạy ứng dụng trong YARN

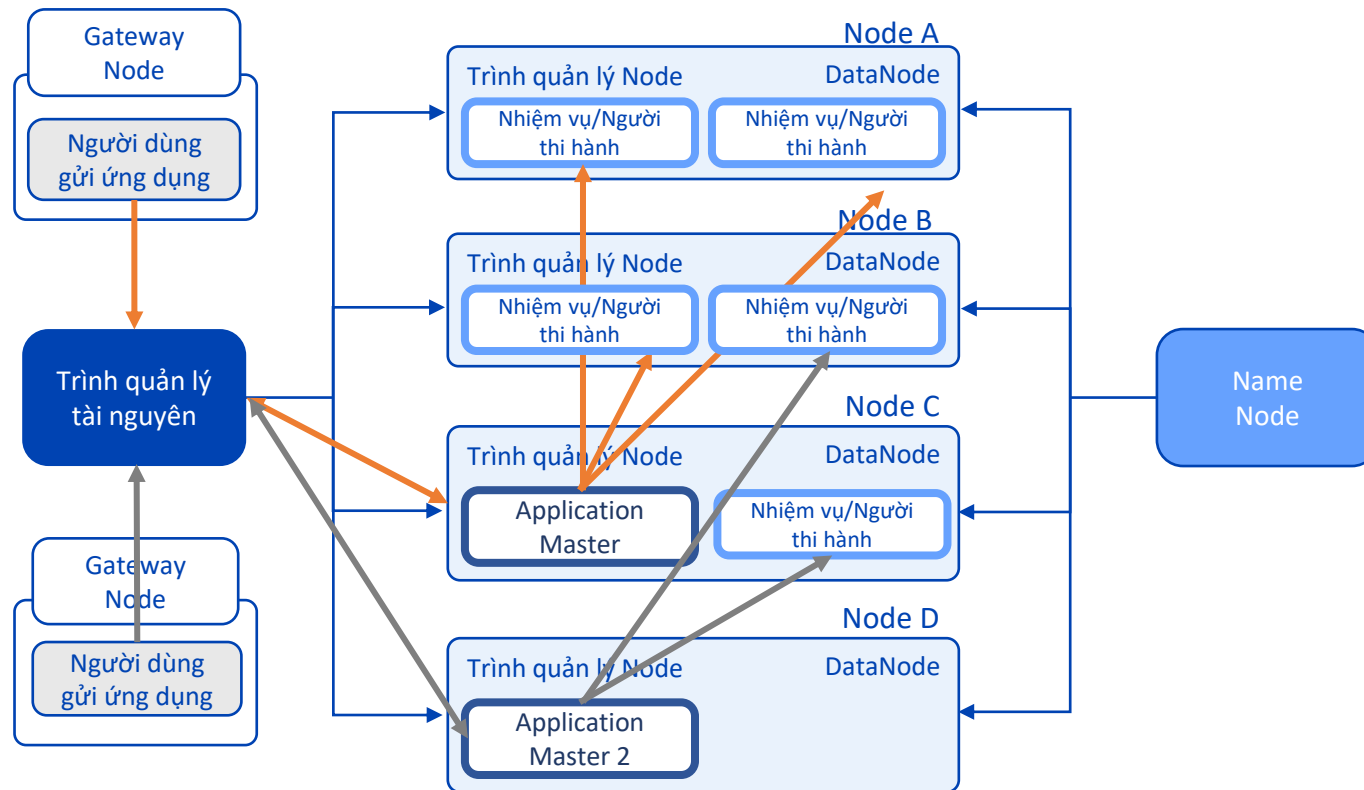


# Chạy ứng dụng trong YARN

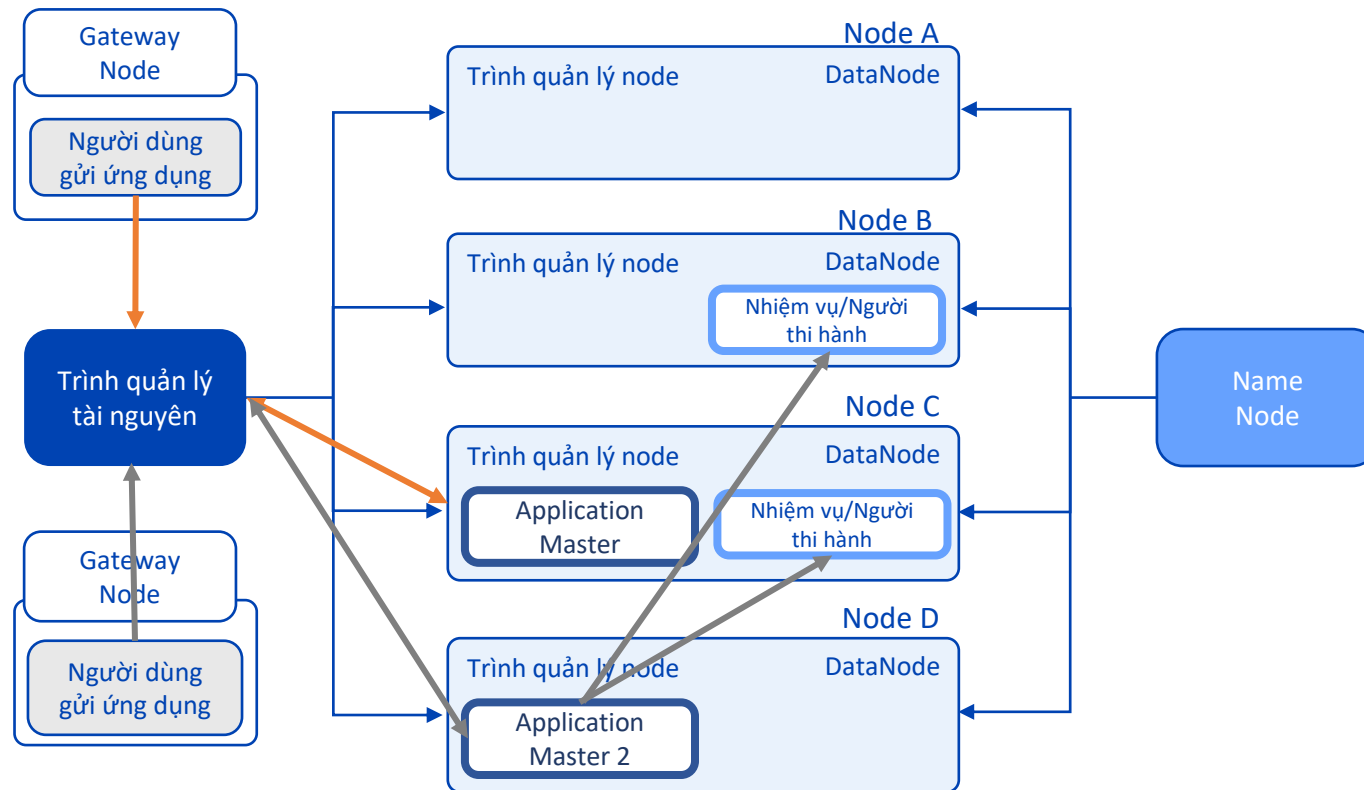




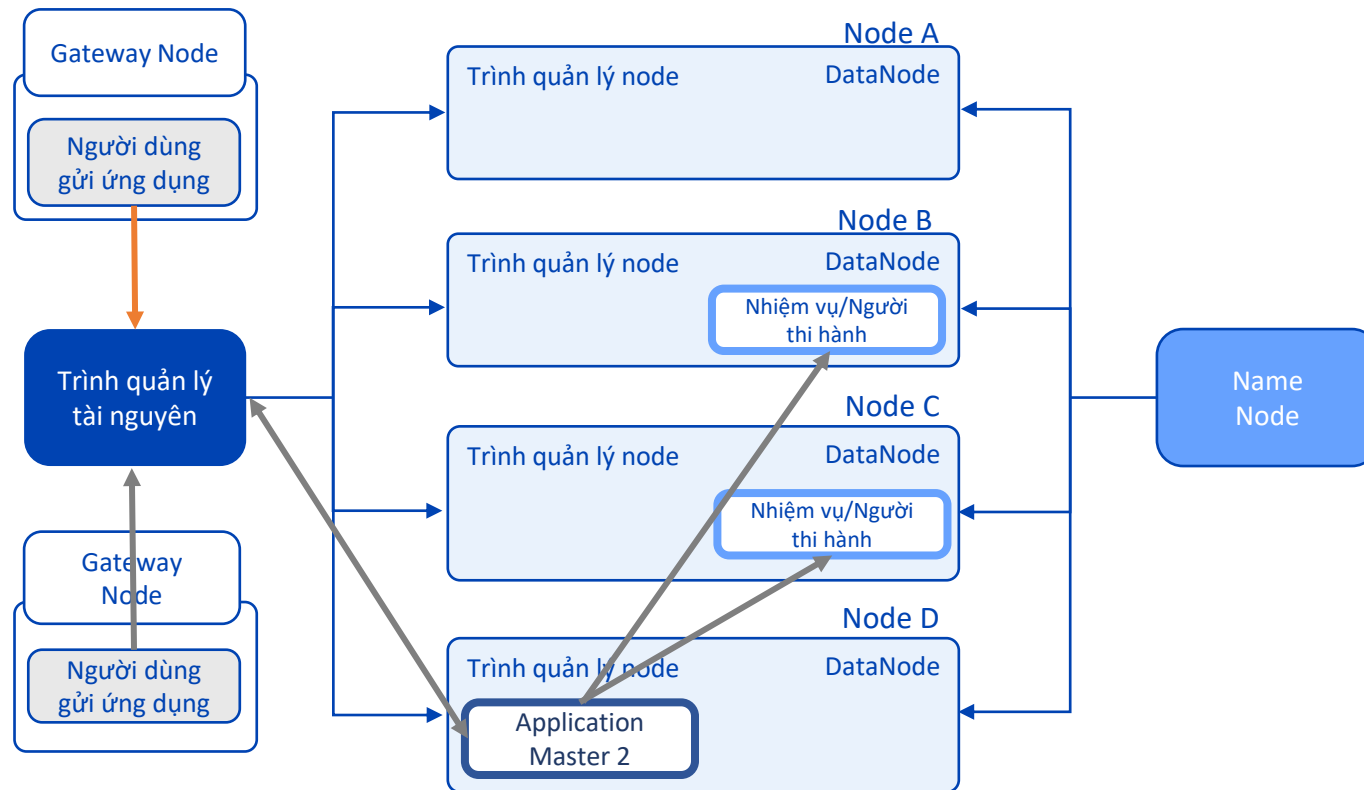
# Chạy ứng dụng trong YARN



# Chạy ứng dụng trong YARN



# Chạy ứng dụng trong YARN

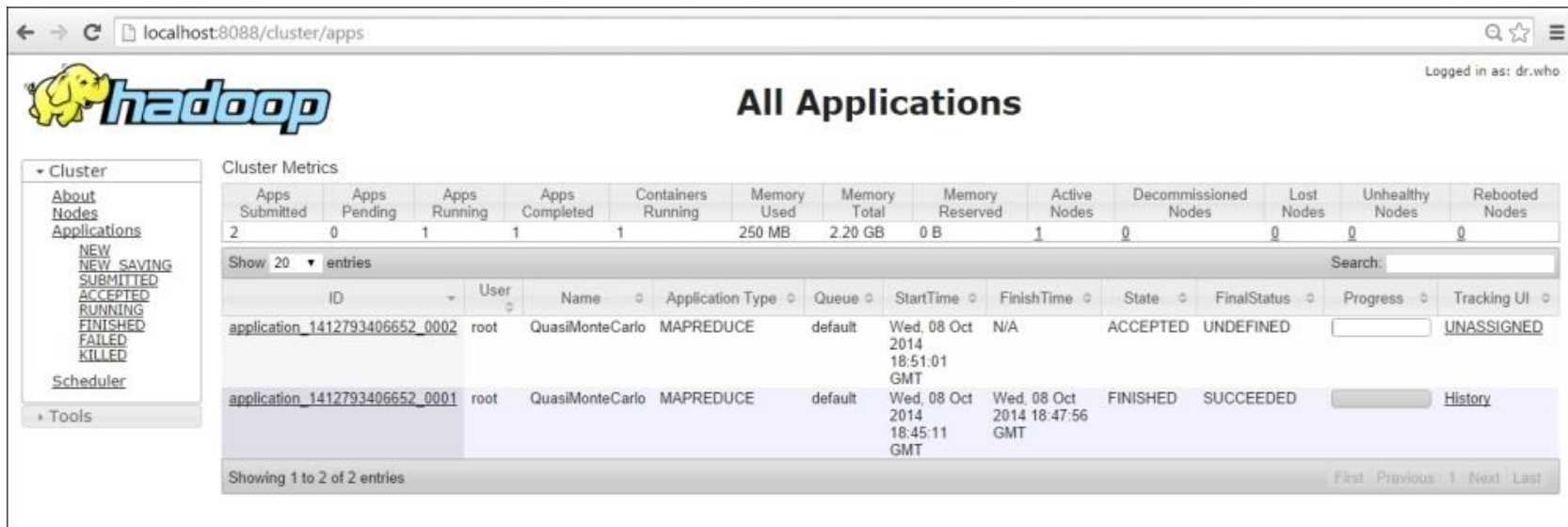


# Dòng lệnh YARN

- | Cú pháp cơ bản: `yarn <lệnh con> <args>`
- | `yarn application -list`
  - ▶ Để xem tất cả các ứng dụng do YARN quản lý hiện đang chạy trên cụm
- | `yarn application -status <app-id>`
  - ▶ Để hiển thị trạng thái của một ứng dụng riêng lẻ
- | `yarn application -kill <app-id>`
  - ▶ Để tắt một ứng dụng đang chạy được xác định bởi id ứng dụng

## Giao diện người dùng web ứng dụng YARN

- Cung cấp GUI dựa trên trình duyệt để xem trạng thái của các ứng dụng đang chạy YARN
- Xem tệp nhật ký



The screenshot displays the Hadoop YARN web interface at the URL `localhost:8088/cluster/apps`. The page is titled "All Applications" and shows the user is logged in as "dr.who".

**Cluster Metrics:**

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
2	0	1	1	1	250 MB	2.20 GB	0 B	1	0	0	0	0

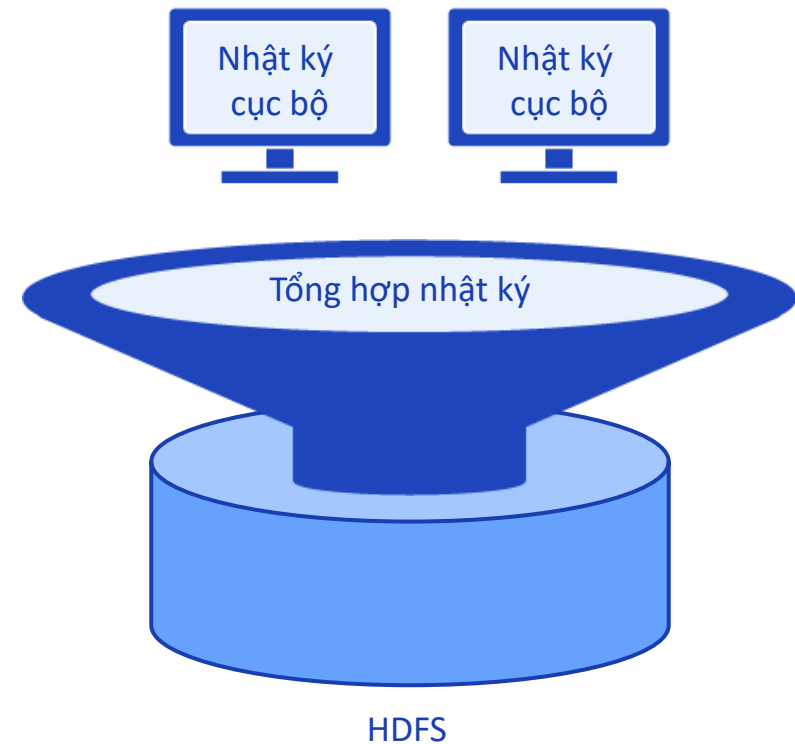
**Applications Table:**

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI
<a href="#">application_1412793406652_0002</a>	root	QuasiMonteCarlo	MAPREDUCE	default	Wed, 08 Oct 2014 18:51:01 GMT	N/A	ACCEPTED	UNDEFINED	<div></div>	<a href="#">UNASSIGNED</a>
<a href="#">application_1412793406652_0001</a>	root	QuasiMonteCarlo	MAPREDUCE	default	Wed, 08 Oct 2014 18:45:11 GMT	Wed, 08 Oct 2014 18:47:56 GMT	FINISHED	SUCCEEDED	<div></div>	<a href="#">History</a>

Showing 1 to 2 of 2 entries

# Tổng hợp nhật ký ứng dụng YARN

- I YARN cung cấp dịch vụ tổng hợp nhật ký ứng dụng
  - ▶ Được đề xuất (được bật theo mặc định trong bản phân phối Cloudera)
- I Khi tính năng tổng hợp nhật ký YARN khả dụng:
  - ▶ Các tệp nhật ký vùng chứa được di chuyển từ /var/log/Hadoop-yarn/container HDFS của máy chủ NodeManager khi ứng dụng hoàn tất
  - ▶ Mặc định HDFS directory /tmp/logs
- I Lưu giữ nhật ký ứng dụng YARN tổng hợp
  - ▶ CM mặc định: 7 ngày



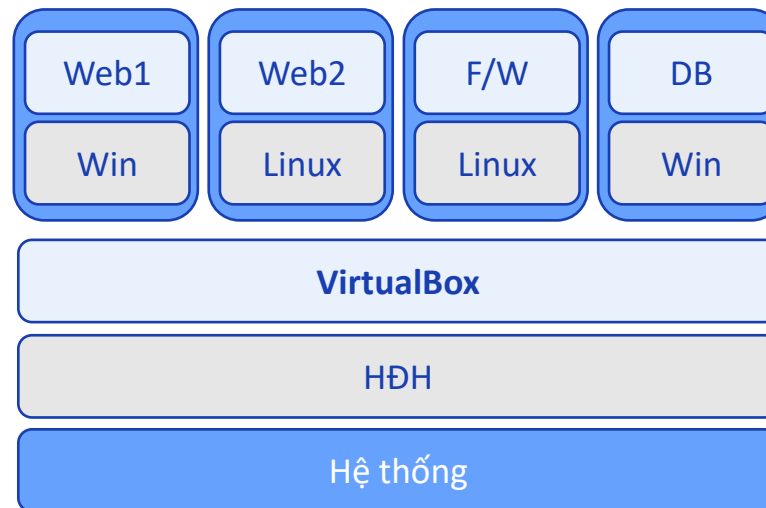
# [Lab1] Khởi động VirtualBox



# VirtualBox là gì?

## I Một ứng dụng ảo hóa đa nền tảng

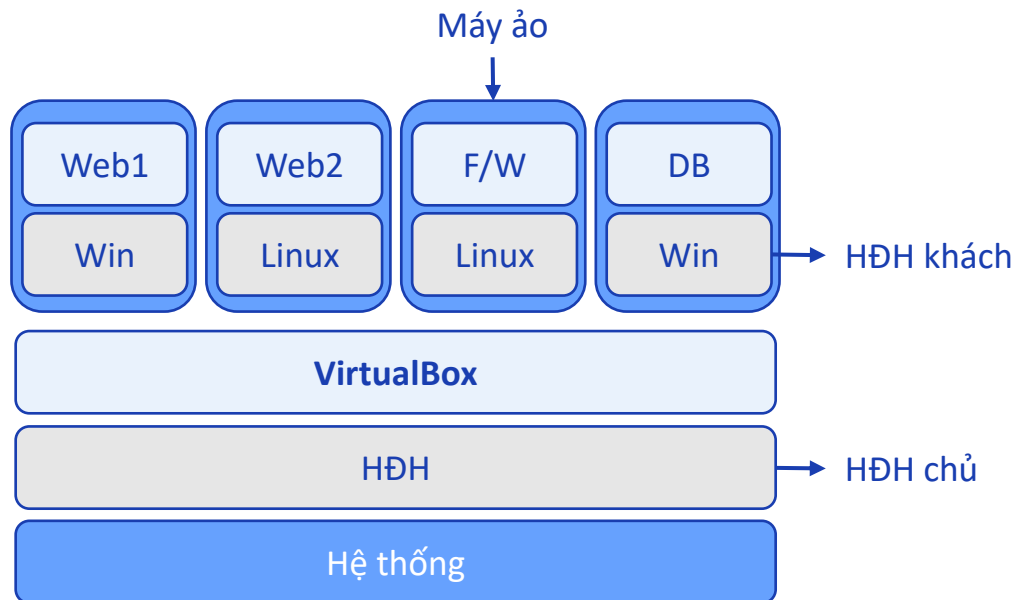
- Chạy trên Windows, Linux, Mac OS X
- Cho phép nhiều máy cùng chạy trên một hệ thống





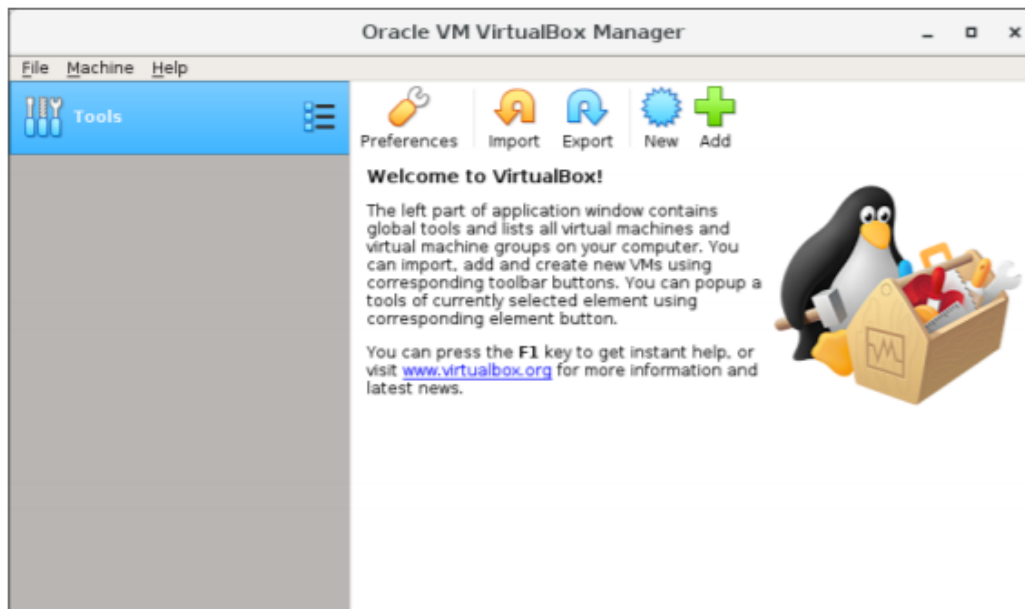
# Thuật ngữ cơ bản

- | HĐH chủ
- | HĐH khách
- | Máy ảo (VM)



# Khởi động VirtualBox

- I Nhấp đúp vào biểu tượng VirtualBox trên màn hình nền Windows
- I Trình quản lý VirtualBox được hiển thị:



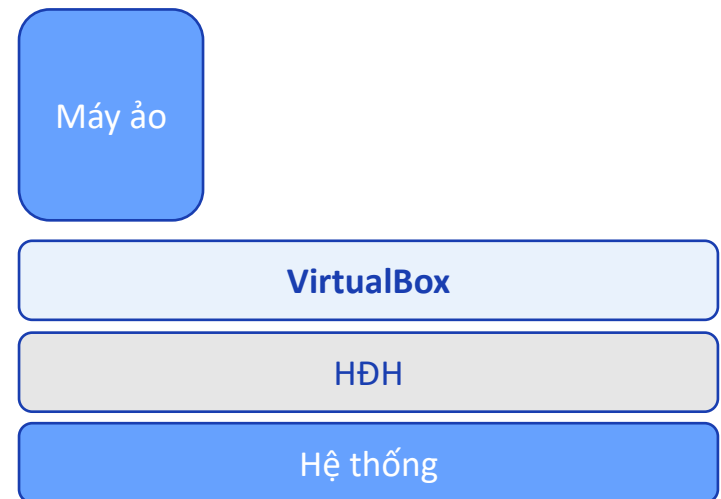
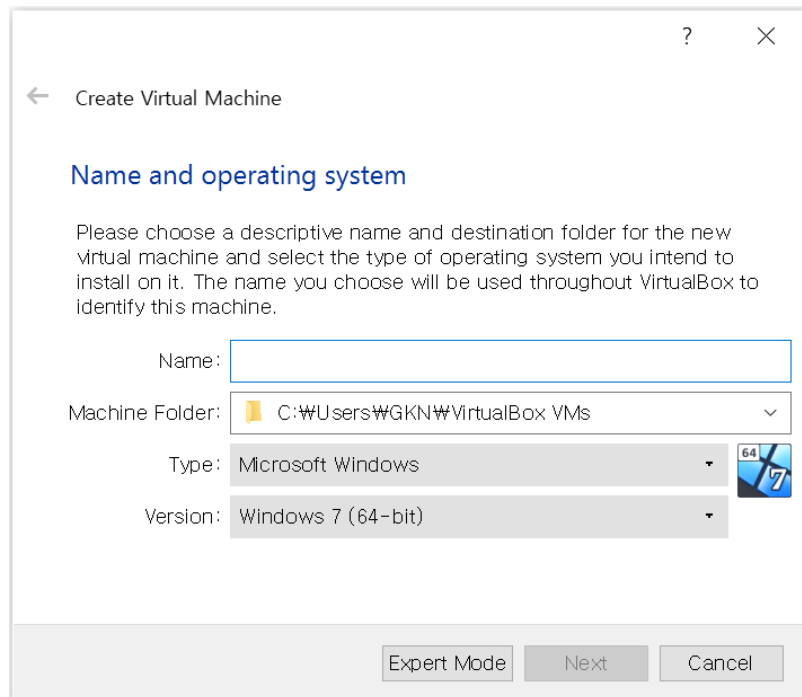
VirtualBox

HDH

Hệ thống

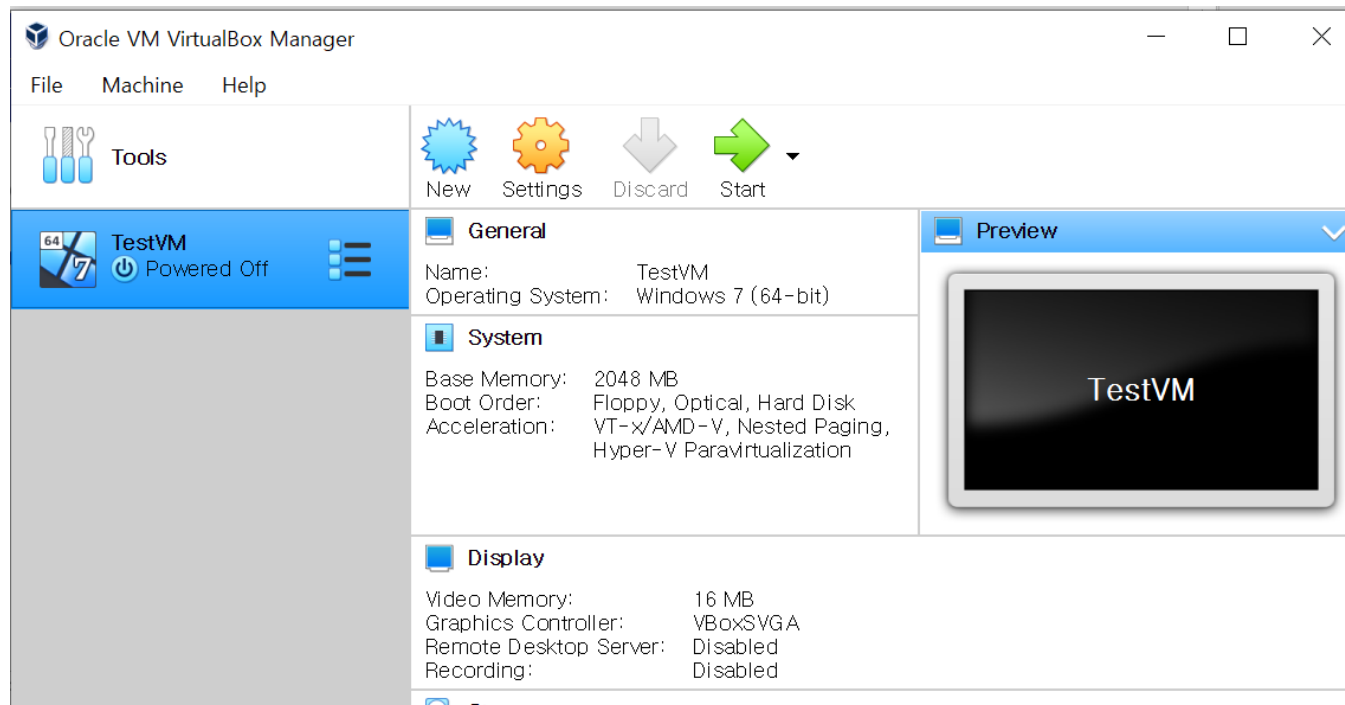
# Tạo Máy ảo

- I Nhấp vào Mới trong Cửa sổ Trình quản lý VirtualBox
- I Một trình hướng dẫn được hiển thị để hướng dẫn bạn cách thiết lập một máy ảo mới



# Kiểm tra thông tin Máy ảo

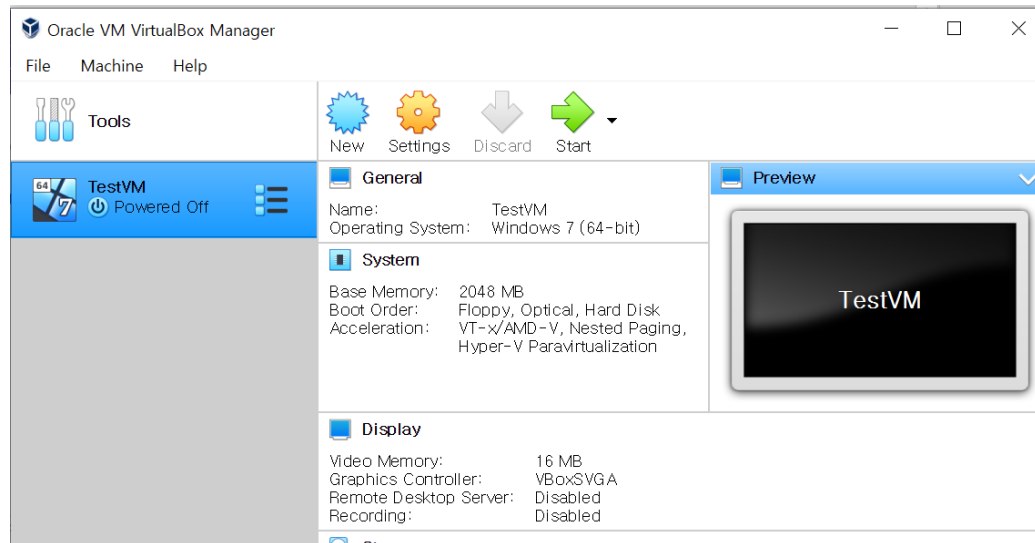
## I Kiểm tra thông tin VM trong Cửa sổ Trình quản lý VirtualBox



# Khởi động Máy ảo

I Một vài lựa chọn:

- ▶ Nhấp đúp vào mục nhập của VM trong danh sách
- ▶ Chọn mục nhập của VM trong danh sách và nhấp vào **Start**



Máy ảo

VirtualBox

HDH

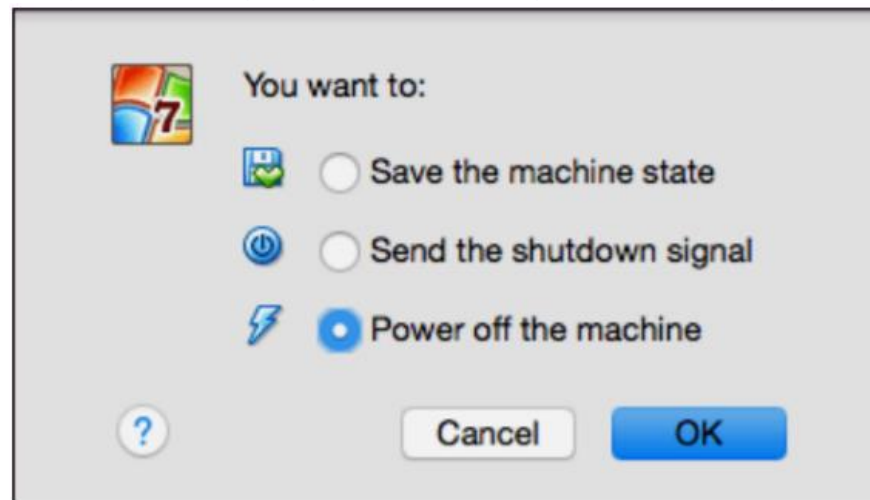
Hệ thống

# Đóng Máy ảo

- Click vào nút Close của cửa sổ máy ảo

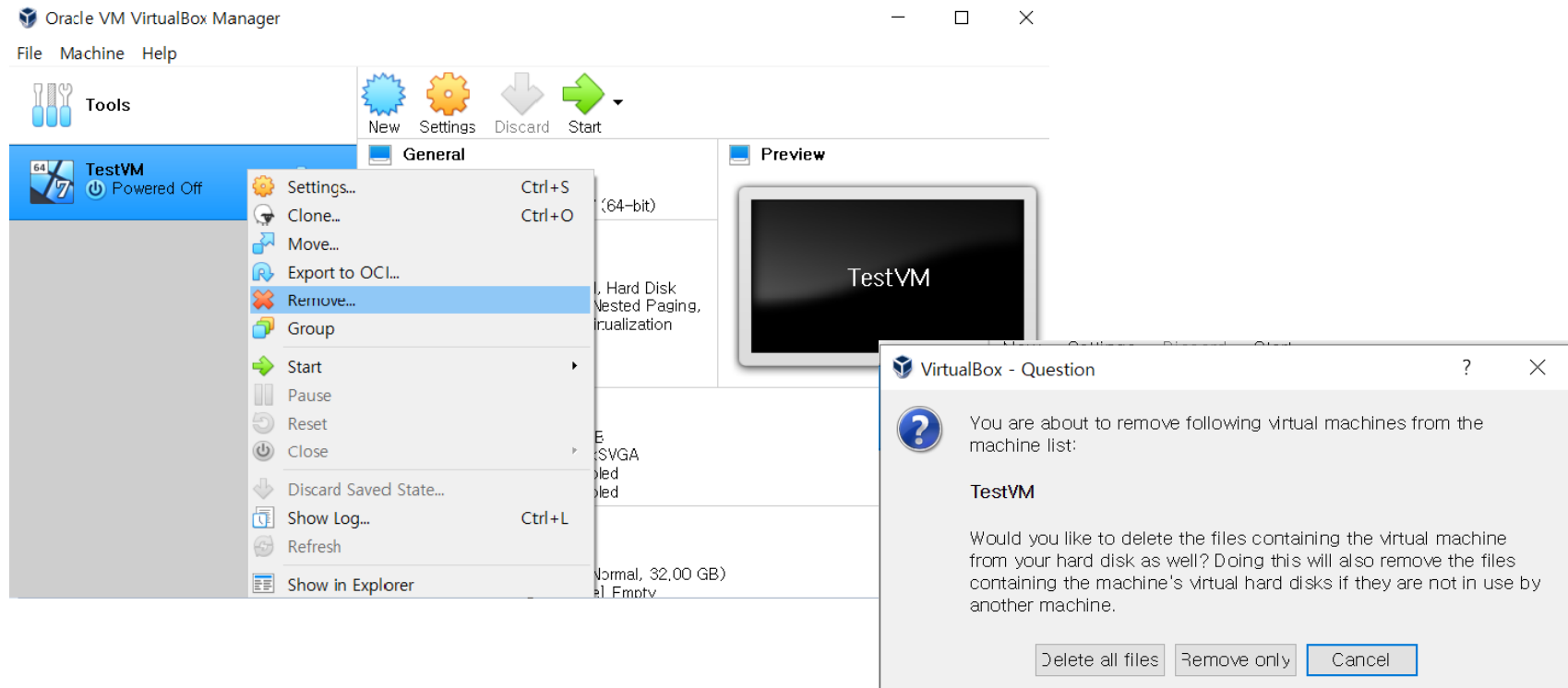


tùy chọn nào giống như khi chúng ta tắt PC thông thường?



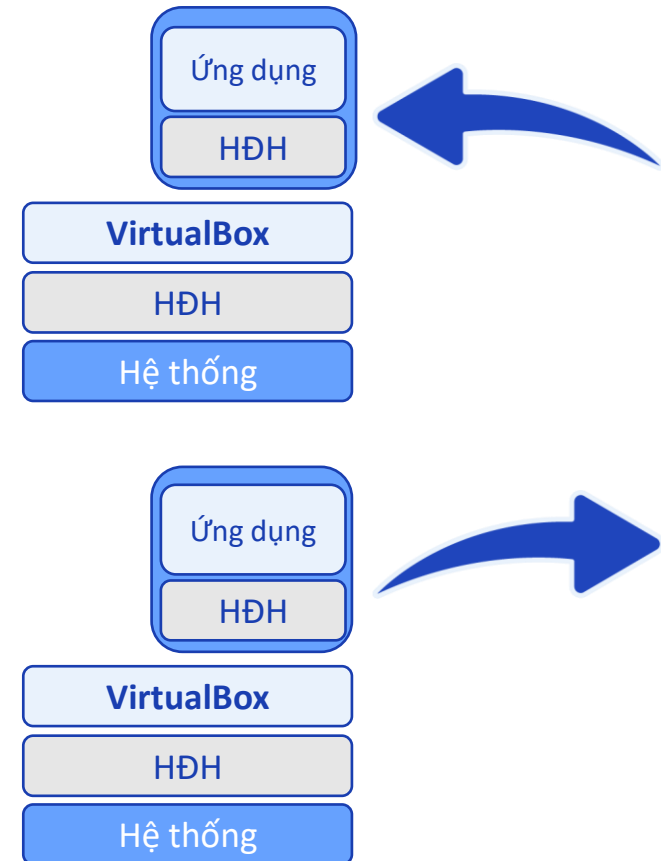
# Xóa Máy ảo

I Nhấp chuột phải vào Máy ảo và chọn **Xóa**



# Nhập & Xuất

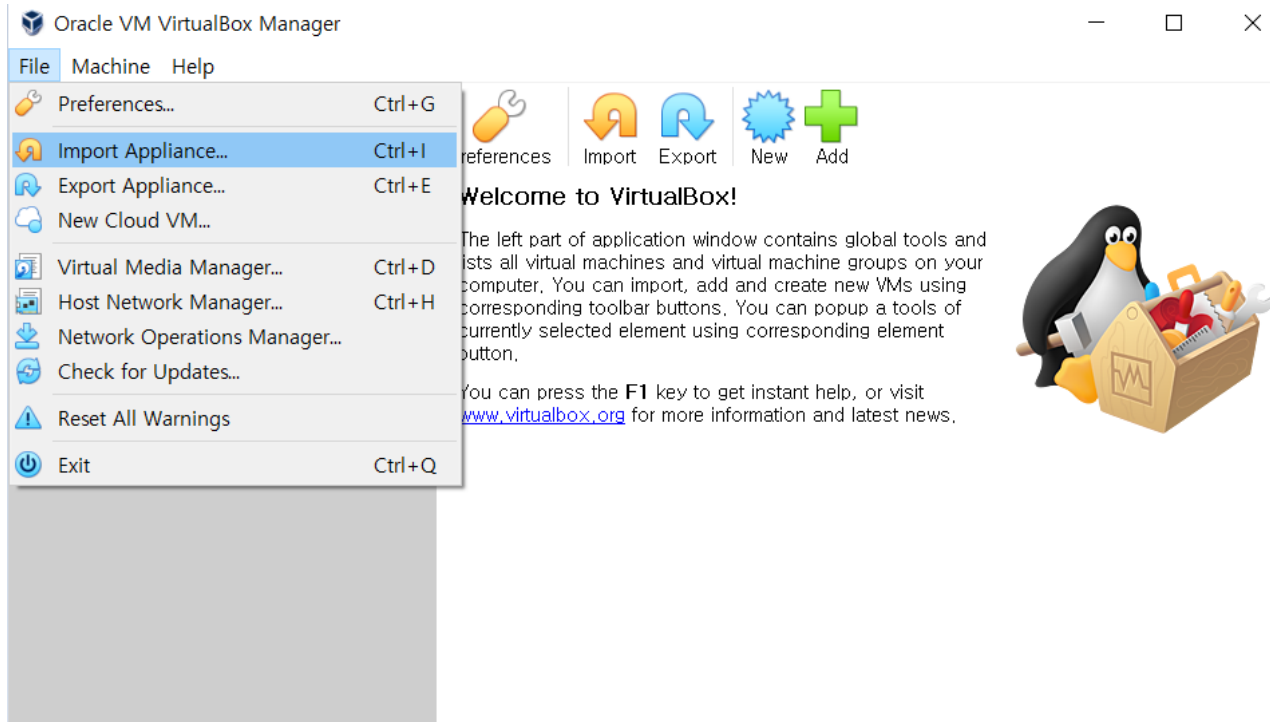
- I Tính năng nhập cho phép bạn dễ dàng nhập hình ảnh máy ảo vào môi trường VirtualBox của mình.
  - ▶ Không cần tạo Máy ảo
  - ▶ Không cần cài đặt hệ điều hành và ứng dụng.
  - ▶ Chỉ cần nhập một máy ảo được tạo tốt
- I Xuất cho phép bạn dễ dàng xuất máy ảo từ môi trường VirtualBox của mình.
  - ▶ Có thể xuất VM của bạn và chuyển sang PC hoặc môi trường khác.





# Nhập & Xuất

## I Nhập và xuất máy ảo



# [Lab2] Làm việc với HDFS



# [Lab3]

## Làm việc với YARN/MapReduce





**SAMSUNG**

Together for Tomorrow!  
**Enabling People**

Education for Future Generations

©2021 SAMSUNG. All rights reserved.

Samsung Electronics Corporate Citizenship Office holds the copyright of book.

This book is a literary property protected by copyright law so reprint and reproduction without permission are prohibited.

To use this book other than the curriculum of Samsung innovation Campus or to use the entire or part of this book, you must receive written consent from copyright holder.