**BỘ TÀI NGUYÊN VÀ MÔI TRƯỜNG**

**TRƯỜNG ĐH TÀI NGUYÊN VÀ MÔI TRƯỜNG TP. HCM**

**KHOA HỆ THỐNG THÔNG TIN VÀ VIỄN THÁM**



**BÁO CÁO ĐỒ ÁN MÔN HQT CSDL**

***Survey of Big Data Storage Technology***

**Thành viên nhóm I:**

|  |  |
| --- | --- |
| Trịnh Hoàng Việt | 0950080152 |
| Đặng Văn Tiên | 0950080142 |
| Nguyễn Trường Vi | 0950080151 |

Lớp: K09\_CNTT04

***TP. Hồ Chí Minh, tháng 12 năm 2022***

**MỤC LỤC**

[I. Giới thiệu 3](#_Toc121057598)

[II. Phân tích đề tài 4](#_Toc121057599)

[1. Giới thiệu 4](#_Toc121057600)

[2. Công nghệ lưu trữ dữ liệu lớn 5](#_Toc121057601)

[2.1 Hệ thống tệp phân tán 5](#_Toc121057602)

[2.2. Cơ sở dữ liệu NOSQL (phi quan hệ). 9](#_Toc121057603)

[2.3 Máy tính cở sở dữ liệu All-in-one 12](#_Toc121057604)

[2.4 Cụm cơ sở dữ liệu mới của kiến trúc MPP 14](#_Toc121057605)

[III. Triển vọng của lưu trữ dữ liệu lớn Công nghệ 14](#_Toc121057606)

[IV. Demo: Hadoop trên Windows 15](#_Toc121057607)

[1. Giới thiệu Hadoop 15](#_Toc121057608)

[2. Cài đặt JDK bản 1.8 (bắt buộc) 19](#_Toc121057609)

[3. Thiết lập biến môi trường cho Java JDK 21](#_Toc121057610)

[4 Tải Hadoop và giải nén vào ổ C 25](#_Toc121057611)

[5 Thiết lập biến môi trường cho Hadoop 26](#_Toc121057612)

[6 Cấu hình các tập tin cho Hadoop 27](#_Toc121057613)

[7 WordCount Program trên Hadoop 34](#_Toc121057613)

[V. Hướng phát triển của Big DATA 39](#_Toc121057614)

[Lời cảm ơn 42](#_Toc121057614)

[Nguồn tham khảo 43](#_Toc121057614)

# Giới thiệu

**Big Data** là các tập dữ liệu có khối lượng lớn và phức tạp. Độ lớn đến mức các phần mềm xử lý dữ liệu truyền thống không có khả năng thu thập, quản lý và xử lý dữ liệu trong một khoảng thời gian hợp lý.

Những tập dữ liệu lớn này có thể bao gồm các dữ liệu có cấu trúc, không có cấu trúc và bán cấu trúc, mỗi tập có thể được khai thác để tìm hiểu insights.

Các giải pháp Big Data cung cấp các công cụ, phương pháp và công nghệ được sử dụng để nắm bắt, lưu trữ, tìm kiếm và phân tích dữ liệu trong vài giây để tìm mối quan hệ và hiểu biết về cải tiến và lợi ích cạnh tranh mà trước đây không có.



**Big Data là các tập dữ liệu có khối lượng lớn và phức tạp**

Ngày nay, 80% dữ liệu là dữ liệu không có cấu trúc và không thể được xử lý bởi các công nghệ truyền thống. Trước đó, một lượng dữ liệu được tạo ra không cao. Chúng ta tiếp tục lưu trữ dữ liệu vì chỉ cần phân tích lịch sử dữ liệu. Nhưng ngày nay việc tạo dữ liệu tính bằng petabyte và không thể lưu trữ dữ liệu nhiều lần và lấy lại khi cần.

# Phân tích đề tài

**Trừu tượng:** Lưu trữ nguồn dữ liệu lớn là nền tảng của xử lý và phân tích nguồn dữ liệu lớn. Bằng cách nghiên cứu và tóm tắt cốt lõi chính của công nghệ xử lý lưu trữ dữ liệu, bài báo này lần lượt nghiên cứu và phân tích bốn khía cạnh sau: hệ thống tập tin phân tán, cơ sở dữ liệu NoSQL (phi quan hệ), thiết bị cơ sở dữ liệu và công nghệ lưu trữ dữ liệu kiểu mới của kiến trúc MPP. Ngoài ra, bài báo này đưa ra một số đề nghị áp dụng cho các môi trường khác nhau có lợi cho việc nắm bắt các sự phát triển của dữ liệu Công nghệ lưu trữ từ các góc độ khác nhau. Bài báo tóm tắt phân khúc tệp, nội dung thích hợp, giá trị và lỗi của hệ thống tập tin phân tán, chủ yếu phân tích và tóm tắt các lý thuyết và nội dung phù hợp trong số bốn mô hình lưu trữ dữ liệu của cơ sở dữ liệu NoSQL (phi quan hệ). Hơn nữa, bài báo nghiên cứu và kết luận những phát triển và tính năng của thiết bị cơ sở dữ liệu từng phút. Đồng thời, phác thảo kiến trúc MPP (Xử lý song ồ ạt), một công nghệ lưu trữ dữ liệu mới. Cuối cùng, các xu hướng nghiên cứu của công nghệ lưu trữ được triển vọng, cung cấp tài liệu tham khảo cho lập trình viên về công nghệ lưu trữ dữ liệu lớn.

**Từ khoá:** Lưu trữ dữ liệu lớn, NoSql (phi quan hệ), Hệ thống tệp phân tán, Máy tính cơ sở dữ liệu All-in-one, Kiến trúc MPP

## Giới thiệu

Trong vài thập kỷ qua, với sự mở rộng lớn của quy mô ứng dụng, dịch vụ web phát triển từ một hình thức duy nhất thành dạng đa phương tiện. Dẫn đến cấu trúc dữ liệu đa dạng, biểu mẫu và tăng trưởng dữ liệu theo cấp số nhân. Tập đoàn Dữ liệu Quốc tế (IDC) dự đoán rằng trong tương lai kích thước dữ liệu sẽ tăng gấp đôi sau mỗi hai năm [1]. Công ty tiên phong trong nghiên cứu dữ liệu lớn (Big Data) — McKinsey & Company, một công ty tư vấn ở Hoa Kỳ, định nghĩa dữ liệu lớn là: tập dữ liệu có quy mô vượt quá khả năng thu thập, lưu trữ, quản lý và phân tích các công cụ cơ sở dữ liệu thông thường [2]. Hệ thống lưu trữ dữ liệu truyền thống đã gặp phải tình trạng tắc nghẽn và không thể hoàn thành quá trình xử lý dữ liệu kịp thời. Dữ liệu lớn có các tính năng như dung lượng cao, nhiều loại dữ liệu khác nhau, mật độ giá trị thấp, tốc độ xử lý cao, mối quan hệ động phức tạp giữa các dữ liệu. Và yêu cầu về tính sẵn sàng, khả năng mở rộng và độ tin cậy cao [3], đặt ra những thách thức đối với công nghệ lưu trữ dữ liệu truyền thống.

Bài báo đã nghiên cứu và tóm tắt các loại công nghệ lưu trữ dữ liệu mới, cho các vấn đề về mở rộng quy mô ứng dụng, tăng trưởng dữ liệu nhanh chóng, nhiều loại dữ liệu. Đầu tiên, nghiên cứu và phân tích tình trạng nghiên cứu của hệ thống tệp phân tán. Bên cạnh đó, hãy phân tích và so sánh các đặc điểm của hệ thống tệp phân tán chính,

chẳng hạn như GFS [4], HDFS [5], GlusterFS [6], GridFS [7], TFS [8], Lustre [9], FastDFS [10]. Ngoài ra, bằng cách nghiên cứu bốn mô hình dữ liệu của cơ sở dữ liệu NoSQL(phi quan hệ), tương ứng tương phản công nghệ lưu trữ của mô hình giá trị chính, mô hình loại cột, mô hình tài liệu và mô hình đồ họa. Và nghiên cứu khả năng ứng dụng và tính năng kiến trúc của máy All-in-one dữ liệu lớn hiện nay. Cuối cùng, giới thiệu định nghĩa và kịch bản của các loại cụm cơ sở dữ liệu mới bằng MPP (Xử lý song ồ ạt) [11] khuôn khổ và dự báo các xu hướng nghiên cứu có thể xảy ra của công nghệ lưu trữ trong tương lai.

## Công nghệ lưu trữ dữ liệu lớn

Bài báo nghiên cứu và phân tích công nghệ lưu trữ dữ liệu lớn từ bốn khía cạnh sau: hệ thống tệp phân tán, cơ sở dữ liệu NoSQL (phi quan hệ), công nghệ lưu trữ dữ liệu kiểu mới của kiến trúc MPP và máy tính cơ sở dữ liệu All-in-one bài báo này đưa ra một số đề nghị áp dụng cho các môi trường khác nhau có lợi cho việc nắm bắt các sự phát triển của dữ liệu Công nghệ lưu trữ từ các góc độ khác nhau

### 2.1 Hệ thống tệp phân tán

Hệ thống tệp là cơ bản của chương trình ứng dụng. Tuy nhiên, với sự phát triển của ứng dụng mạng, dữ liệu phát triển nhanh chóng. Vì vậy, công nghệ lưu trữ dữ liệu lớn đã trở thành nhiệm vụ chính của các doanh nghiệp và tổ chức nghiên cứu. Do dung lượng lưu trữ hạn chế nên các hệ thống lưu trữ truyền thống khó có thể giải quyết được bài toán lưu trữ dữ liệu lớn. Vì vậy, chúng tôi sử dụng hệ thống tệp phân tán để chuyển tải hệ thống sang nhiều nút. Hệ thống tệp phân tán cung cấp dung lượng lưu trữ polymer và băng thông I/O, để có thể dễ dàng mở rộng quy mô của hệ thống [12].

Nói chung, việc xác định xem hệ thống tệp phân tán có thành công hay không phụ thuộc vào ba yếu tố sau: chế độ lưu trữ dữ liệu, tốc độ đọc, cơ chế bảo mật. Tuy nhiên, vẫn có những cải tiến được thực hiện trong hệ thống tệp phân tán. Ví dụ: GFS và HDFS được thiết kế cho các tệp lớn không thể đáp ứng các yêu cầu lưu trữ của nhiều tệp nhỏ. Tần suất truy cập của tệp nhỏ cao, dẫn đến tần suất truy cập đĩa cứng cao. Vì vậy, hiệu suất của I/O đã bị giảm [13]. Các tệp nhỏ cũng có thể dẫn đến việc tạo ra một số lượng lớn siêu dữ liệu, điều này sẽ ảnh hưởng đến khả năng quản lý và khôi phục máy chủ siêu dữ liệu, và khả năng phục hồi, và sau đó dẫn đến sự suy giảm của tổng hiệu suất. Hơn nữa, vì tệp tương đối nhỏ, Nó dễ bị tạo ra sự phân mảnh tệp, lãng phí dung lượng đĩa. Tạo liên kết cho mỗi tệp sẽ gây ra độ trễ mạng [14].

*Bảng 1. Phân tích danh sách các hệ thống tệp phân tán phổ biến [15-20].*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tên | Sự phân loại của tệp | Hệ thống sao lưu | Ưu điểm | Khuyết điểm | Ứng dụng |
| GPS | Các tệp được lưu trữ trong GFS được chia thành các khối có kích thước cố định. | Các khối được sao chép đến nhiều đoạn máy chủ, nhanh hơn 3 bước so với mặc định | GFS sẽ không coi lỗi phần cứng là bất thường. Thông thường, một bản cập nhật được xử lý bằng cách thêm dữ liệu mới thay vì thay đổi dữ liệu hiện có. | Không áp dụng được cho các tệp bộ nhớ nhỏ. Các tệp nhỏ bổ sung sẽ làm giảm hiệu suất. | Tập dữ liệu phân tán cỡ lớn. Kích thước dữ liệu từ 4GB ~ 40GB. |
| HDFS | Các tệp lớn được chia thành các khối có kích thước cố định là 64MB. Mỗi khối sẽ được lưu trữ một số bản sao trên nhiều node (nút dữ liệu). | Hỗ trợ sao chép dữ liệu. Sao lưu nhiều bản trên các node khác nhau. | Có khả năng mở rộng rất mạnh. Một HDFS có thể hỗ trợ cho hàng chục triệu tài liệu. Và có khả năng thời gian thực cao. | Không thể sử dụng trong trường hợp truy cập dữ liệu yêu cầu độ trễ thấp | Tệp dữ liệu rất lớn có kích thước từ mức GB ~ TB. |
| GlusterFS | Không hỗ trợ cho tệp phân tán. | Hỗ trợ sao chép dữ liệu và cung cấp không gian tên toàn cầu. Bản sao lưu của các tệp có thể được lưu trữ trong các máy chủ khác nhau. Khi đọc bản sao lưu, hệ thống sẽ chọn bản sao lưu tốt nhất. | Hỗ trợ CIFS, NFS và máy gốc sử dụng máy khách GlusterFS. Hệ thống đa tệp có thể được triển khai trên hệ thống tệp phân tán ảo. Sử dụng bảng điều khiển dành cho quản trị viên, quản lý cập nhật trung tâm một cách dễ dàng. Tất cả các node có thể được sử dụng để lấy dữ liệu. | Chỉ có thể quản lý hệ thống trên một máy chủ, không có dự phòng. Không thể thêm một node mới giữa chừng. Không biết cách thêm nhiều đĩa vào mỗi node; Không có chính sách bảo mật trong GlusterFS. | Hỗ trợ tệp dữ liệu ở mức PB. |
| TFS | Không hỗ trợ tệp phân tán. Một số lượng lớn các tệp nhỏ sẽ gộp thành tệp lớn. | TFS lưu trữ các tệp dữ liệu trong các khối và lưu trữ nhiều bản sao trong trường hợp bảo mật dữ liệu. | Hoạt động đơn giản, với khả năng dễ dàng mở rộng và cân bằng tải. Hỗ trợ chia tuyến tính. Dễ dàng mở rộng đến mức PB. | Khi xử lý đồng thời nhiều tệp và các tệp có kích thước lớn hơn 5 MB, TFS sẽ phát sinh các lỗi nghiêm trọng. Rất hiếm khi hỗ trợ lưu các tệp cỡ lớn. Không hỗ trợ danh mục và sự cho phép người dùng. | Việc lưu trữ và xử lý một lượng lớn dữ liệu phi cấu trúc và hình ảnh khổng lồ trên trang web Taobao. |
| GridFS | Hỗ trợ chia một tệp lớn thành nhiều tệp tập tin tài liệu nhỏ. | GridFS lưu trữ dữ liệu và siêu dữ liệu trong MongoDB. Sao lưu tệp để đối phó với chuyển đổi dự phòng và tích hợp dữ liệu. Và cũng có thể được sử dụng để tiện ích mở rộng, sao lưu ngay lập tức hoặc được sử dụng làm nguồn ngoại tuyến xử lý hàng loạt dữ liệu . | Nó dựa trên mô hình cấu trúc của bộ lưu trữ đối tượng, giảm thêm chi phí truy cập trong thời gian chạy. GridFS được thiết kế để điều chỉnh cơ chế truy cập, nhằm thích ứng với hiệu suất ứng dụng nhanh hơn do chế độ I/O đưa ra. Có thể cung cấp hiệu suất I/O nhanh nhất mà cụm ứng dụng cần. GridFS đảm bảo cân bằng tải trong từng thiết bị lưu trữ, thay vì lấp đầy mộ node. | Tốc độ đọc tệp từ GridFS chậm hơn so với đọc trực tiếp từ hệ thống tệp. Nếu tệp lớn và được lưu trữ dưới dạng nhiều tệp, không thể khóa tất cả các khối tệp khi sửa đổi tệp lớn này. Khi thay đổi tài liệu được lưu trữ trong GridFS, nó chỉ có thể xóa tài liệu cũ trước rồi lưu lại tài liệu. | Thích hợp cho các tệp lớn hiếm khi cần  có được thay đổi. |
| Tên | Sự phân loại của tệp | Hệ thống sao lưu | Ưu điểm | Khuyết điểm | Ứng dụng |
| Lustre | Chia dữ liệu thành một số đối tượng cố định. Mỗi đối tượng chứa một số khối dữ liệu. Khi một khối dữ liệu được ghi vào đối tượng vượt quá dung lượng của nó, lần ghi tiếp theo sẽ được lưu trữ trong đối tượng tiếp theo. Lustre có thể phân phối tệp tối đa thành 160 đối tượng để lưu trữ. | Lustre cung cấp hai công cụ sao lưu. Một cái được sử dụng để quét hệ thống tệp và một cái khác được sử dụng để sao lưu gói và khôi phục dữ liệu. | Có thể cung cấp khả năng chia sẻ dữ liệu và xử lý song song. Khả năng mở rộng là rất mạnh mẽ. Có thể cung cấp công nghệ chuyển đổi dự phòng cho siêu dữ liệu và dữ liệu đích dưới sự quản lý ánh sáng, đạt được quyền truy cập với độ tin cậy cao. Cơ chế quản lý phân tán có thể đạt được kiểm soát đồng thời. Cung cấp quyền truy cập trong nhiều phương thức mạng. | Rất khó để thực hiện phản chiếu dữ liệu. Chuyển đổi dự phòng giữa các nút dựa trên công nghệ heartbeat của bên thứ ba. Chỉ có hai nút quản lý siêu dữ liệu. Nếu quy mô hệ thống đã đạt đến quy mô nhất định, nút quản lý sẽ đạt đến mức quá tải. Lustre kernel chỉ có thể được triển khai trên Linux với một số hạn chế. | Hỗ trợ lưu trữ tệp lớn ở cấp độ PB và phù hợp với cụm máy tính lớn hoặc siêu máy tính. |
| Ceph | Áp dụng mẫu RADIO cho nhiều ổ đĩa cứng. Phân tán dữ liệu liên tục trên nhiều ổ đĩa để truy cập, nhằm thích nghi với việc cân bằng tải. | Hỗ trợ sao lưu dữ liệu. Có nhiều máy chủ siêu dữ liệu. | Lưu trữ dữ liệu và siêu dữ liệu riêng biệt. Siêu dữ liệu được quản lý bằng cách sử dụng phân phối động. Có bộ lưu trữ đối tượng phân tán tự động đáng tin cậy. | Công nghệ chưa hoàn thiệt, chưa áp dụng được vào môi trường sản xuất. | Hỗ trợ lưu trữ tệp lớn ở cấp độ PB. |
| FastDFS | FastDFS không lưu trữ tệp theo khối. Các tệp do máy khách tải lên tương ứng với các tệp được lưu trữ trên máy chủ. | FastDFS áp dụng chế độ lưu trữ theo nhóm. Các máy chủ lưu trữ trong cùng một nhóm sao lưu lẫn nhau. | Máy chủ FastDFS chỉ có hai ký tự, nút theo dõi và lưu trữ. Vì vậy, nó nhẹ. FastDFS áp dụng chế độ lưu trữ theo nhóm, linh hoạt và được kiểm soát chặt chẽ. Trong FastDFS, mỗi nút là một nút chính, có cấu trúc ngang hàng. Có thể thay đổi số lượng trình theo dõi bất kỳ lúc nào, tùy theo áp lực của máy chủ. | FastDFS không lưu trữ tệp theo khối. Vì vậy, nó không phù hợp với kịch bản tính toán phân tán. Dung lượng lưu trữ bị giới hạn bởi một máy chủ lưu trữ duy nhất. | Thích hợp cho dịch vụ có lưu lượng truy cập cao, với tệp là nhà cung cấp dịch vụ, chẳng hạn như trang web album ảnh, trang web video, v.v. Và số lượng lớn tệp nhỏ, 4K ~ 500M. |

### 2.2. Cơ sở dữ liệu NOSQL (phi quan hệ).

Với sự tăng trưởng nhanh chóng về quy mô dữ liệu của người dùng doanh nghiệp và nâng cao nhu cầu của người dùng về mức độ dịch vụ, Cơ sở dữ liệu quan hệ truyền thống có một số hạn chế. Cơ sở dữ liệu truyền thống sử dụng tệp phẳng dựa trên bản ghi có cấu trúc để lưu trữ tất cả dữ liệu ứng dụng, dẫn đến sự không phù hợp giữa ứng dụng và cơ sở dữ liệu. Điều này xảy ra khi ứng dụng được mã hóa bằng ngôn ngữ khai báo. Cấu trúc của nó hoàn toàn khác với các cơ sở dữ liệu này [21]. Trước đây, để cải thiện hiệu suất của hệ thống, các thành phần và tài nguyên là mở rộng theo chiều dọc. Tuy nhiên, vì bộ nhớ và ứng dụng không còn tách biệt nữa, Mỗi lần mở rộng tài nguyên sẽ là gián đoạn dịch vụ và đặt lại ứng dụng. Hầu hết dữ liệu chúng tôi tạo ra là dữ liệu không đồng nhất. Sự tồn tại của một số lượng lớn dữ liệu có cấu trúc và phi cấu trúc gây khó khăn cho việc xác định trước mô hình dữ liệu quan hệ hoàn hảo và thống nhất, và khả năng mở rộng theo chiều ngang của cơ sở dữ liệu quan hệ là xấu [22]. Hầu hết các cơ sở dữ liệu quan hệ không hỗ trợ lưu trữ phân tán quy mô lớn. Đồng thời, khó có thể đáp ứng yêu cầu thời gian thực về tính đồng thời cao và lượng dữ liệu lớn. Vì vậy, công nghệ lưu trữ cơ bản không chỉ nên linh hoạt để cho phép dữ liệu được lưu trữ ở dạng tự nhiên mà còn đáp ứng nhu cầu của giới hạn lưu trữ. So với cơ sở dữ liệu quan hệ, hệ thống lưu trữ cơ sở dữ liệu NoSQL (phi quan hệ) hỗ trợ lưu trữ và quản lý động dữ liệu đại chúng. Nó tránh được sự phức tạp không cần thiết, có thông lượng cao và có thể xử lý tốt tỷ lệ ngang. Và khả năng chịu lỗi cao có thể lưu trữ cấu trúc, dữ liệu bán cấu trúc và phi cấu trúc để tránh ánh xạ quan hệ đối tượng. Ý tưởng thiết kế của cơ sở dữ liệu NoSQL (phi quan hệ)là trích xuất cơ chế lập chỉ mụccủa cơ sở dữ liệu quan hệ, kết hợp chiến lược lưu trữ phân tán và xóa những điều đó không cần thiết đối với một số vấn đề trong hệ thống SQL. Do đó nó đạt được hiệu quả tương đối tốt, Khả năng mở rộng và tính linh hoạt [23].

Cơ sở dữ liệu Nosql (phi quan hệ) chủ yếu được chia thành: lưu trữ khóa-giá trị, lưu trữ dựa trên cột, lưu trữ tài liệu, lưu trữ đồ họa.

Cơ sở dữ liệu khóa-giá trị được thiết kế để hỗ trợ các thao tác truy vấn đơn giản, để lại các hoạt động phức tạp cho lớp ứng dụng. Tập dữ liệu sẽ ánh xạ khóa đến một hoặc một tập hợp các giá trị. Đó là, key là từ khóa duy nhất để tìm từng địa chỉ dữ liệu, điều đó cũng có nghĩa là nó không thể thiếu. Giá trị là nội dung mà dữ liệu thực sự lưu trữ. Lưu trữ khóa-giá trị cung cấp một bảng băm với các cặp khóa-giá trị trên các máy chủ từ xa của một cụm phân tán, để thực hiện ánh xạ từ key đến value. Giá trị băm dựa trên khóa định vị địa chỉ của dữ liệu trực tiếp, đạt được truy vấn đồng thời cao nhanh chóng, và cũng hỗ trợ hoạt động của dữ liệu đại chúng. Lưu trữ khóa-giá trị được chia thành loại khóa-giá trị, loại khóa-tài liệu và loại khóa-cột [24]. Loại khóa-cột là sự mở rộng điển hình của các cặp khóa-giá trị của loại khóa-giá trị. Vì tính đơn giản và khả năng mở rộng linh hoạt của nó, nó cũng là xu hướng chủ đạo của mô hình dữ liệu

Internet of Things and Cloud Computing 2016; 4(3): 28-33

Lưu trữ dựa trên cột tổng quát thay thế các cột bằng họ cột. Ý tưởng của cơ sở dữ liệu quan hệ là lưu trữ tất cả các bảng có một dòng trên đĩa. Nghĩa là, một danh sách các mục nhập được liên kết với cùng một id hàng cụ thể sẽ được lưu trữ cùng nhau [25]. Vì các ngân hàng hoặc tổ chức tài chính cần phải duy trì một số lượng lớn các hồ sơ liên quan, không đảm bảo rằng tất cả các giá trị luôn được lưu trữ một cách liên tục. Trong cơ sở dữ liệu của dữ liệu cột, toàn bộ cột của bảng được lưu trữ cùng nhau, ánh xạ tới một chìa khóa. Bởi vì tất cả các mục được liệt kê đều có index, chúng ta chỉ có thể tìm kiếm một phần của bảng. Một cột cũng có thể có các cột lồng nhau có cấu trúc phân cấp và một trong số đó là siêu cột [26]. Điều này cung cấp truy vấn đơn giản và truy cập nhanh chóng, đồng thời tránh chi phí không cần thiết khi tìm kiếm khóa duy nhất của bản ghi.

Cơ sở dữ liệu tài liệu là một loại cơ sở dữ liệu quan hệ khác, được sử dụng để lưu trữ dữ liệu bán cấu trúc. Có XML (ngôn ngữ đánh dấu có thể mở rộng), JSON (ký hiệu đối tượng JavaScript), hoặc các định dạng tương tự khác [27]. Một tài liệu có thể được xem như một dòng trong cơ sở dữ liệu quan hệ, trong đó chứa tất cả các thông tin liên quan của tài liệu. Một nhóm bao gồm nhiều tài liệu, và mỗi tài liệu có thể có các mẫu khác nhau và số lượng và loại lưu trữ dữ liệu khác nhau [28]. Lưu trữ tin nhắn văn bản là tối ưu hóa đặc biệt. Do các tập dữ liệu liên quan được lưu trữ chuyên sâu, chi phí của hoạt động SQL JOIN được tiết kiệm. Mặc dù cơ sở dữ liệu là thiết kế không có lược đồ, Nó lưu trữ các bản ghi bán cấu trúc và là cấu trúc phân cấp. Cơ sở dữ liệu đồ thị phù hợp nhất với tìm kiếm ứng dụng và tìm kiếm ứng dụng, chẳng hạn như tìm các liên kết liên quan trên LinkedIn, tìm bạn bè trên Facebook [29], v.v. Nó chú ý nhiều hơn đến mối quan hệ giữa các mục dữ liệu thay vì chính dữ liệu. Họ tối ưu hóa cao việc đi qua nhanh chóngvà sử dụng thuật toán đồ thị một cách hiệu quả. Ví dụ: con đường ngắn nhất là đầu tiên để tìm ra sự liên quan giữa thông tin, v.v.

*BẢNG 2: Danh sách phân tích bốn mô hình cơ sở dữ liệu phi quan hệ (NoSQL)*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Loại cơ sở dữ liệu | Ưu điểm | Khuyết điểm | Mô hình dữ liệu | Ứng dụng | Trường hợp |
| Khóa - Lưu trữ giá trị | Có hiệu suất đọc và viết đồng thời rất cao. Dữ liệu được lập chỉ mục và phân đoạn theo giá trị khóa. Tìm kiếm nhanh chóng và mô hình dữ liệu rất đơn giản. | Dữ liệu không có cấu trúc và không hỗ trợ hoạt động dữ liệu logic phức tạp. | ánh xạ khóa-giá trị giữa khóa và giá trị | Bộ nhớ đệm nội dung. Chủ yếu được sử dụng cho hệ thống nhật ký. | Dynamo, Redis, Voldemort |
| Lưu trữ dựa trên Cột | Tìm kiếm nhanh chóng, khả năng mở rộng là tốt và tiết kiệm rất nhiều hoạt động I / O. Nó dễ dàng hơn cho phần mở rộng phân tán. | Chức năng tương đối hạn chế. | Lưu trữ dựa trên cột, trong đó dữ liệu trong cùng một cột được lưu trữ trên cùng một trang. | Hệ thống tập tin phân tán | Bigtable, Cassandra, HBase HyperTable |
| Lưu trữ tài liệu | Không cần xác định cấu trúc dữ liệu trước. Sử dụng tài liệu có định dạng cụ thể thay vì bộ làm đơn vị lưu trữ dữ liệu. | Hiệu quả truy vấn không cao và thiếu cú pháp truy vấn thống nhất. | Giá trị trỏ đến dữ liệu có cấu trúc. | Ứng dụng web | CouchDB, MongoDB XML Database ThruDB |
| Lưu trữ đồ họa | Sử dụng lý thuyết đồ thị và thuật toán liên quan để cải thiện hiệu suất lưu trữ, dữ liệu quản lý và vận hành. | Chức năng tương đối hạn chế. | Cấu trúc đồ thị | Mạng xã hội, biểu đồ mối quan hệ | Neo4j, GraphDB InfoGrdi |

### 2.3 Máy tính cở sở dữ liệu All-in-one

Trong những năm gần đây, đối mặt với việc xử lý và lưu trữ dữ liệu hàng loạt, nhiều nhà sản xuất phần cứng truyền thống đề xuất Giải pháp tích hợp --- cơ sở dữ liệu máy All-in-one, đã trở thành một điểm nóng. Theo hình thức sản phẩm của máy All-in-one, nó là sự đơn giản hóa các sự phức tạp của việc triển khai và quản lý cơ sở hạ tầng của trung tâm dữ liệu, giải quyết vấn đề của tài nguyên phần cứng cơ bản ở thời đại dữ liệu lớn, yêu cầu của máy All-in-one và chi phí lưu trữ dữ liệu hàng loạt. Các nhà sản xuất quốc tế, chẳng hạn như IBM, Oracle, EMC, ra mắt các sản phẩm và giải pháp tích hợp cho dữ liệu lớn [35]. Theo họ, các nhà sản xuất Trung Quốc cũng phát triển cơ sở dữ liệu riêng máy All-in-one. Ví dụ, máy cơ sở dữ liệu All-in-one của Huawei sử dụng lợi thế kiến trúc phần cứng của máy tính, lưu trữ và hội tụ mạng, và tính năng thông lượng cao và IOPS cao, tích hợp các đặc tính tuyệt vời của card mạng thông minh, SSD và các phần cứng khác, giải quyết nút thắt cổ chai hiệu suất giữa máy tính và lưu trữ. Bộ xử lý dữ liệu lớn XData của Shuguang tách đơn vị lưu trữ dữ liệu và đơn vị xử lý. Bằng cách xây dựng phần mềm trung gian dịch vụ hiệu quả, trùng hợp nút lưu trữ dữ liệu cơ bản áp dụng cấu trúc không chia sẻ gì thành một hình ảnh hệ thống xử lý dữ liệu duy nhất. Langchao đám mây máy All-in-one dữ liệu lớn bao gồm các phiên kỹ thuật như lưu trữ dữ liệu, xử lý dữ liệu, trình bày dữ liệu, v.v. . Và cũng có dữ liệu Yunchuang Storage máy All-in-one điện toán đám mây cube, máy All-in-one dữ liệu lớn Zhongzhiheda, Dữ liệu lớn dựa trên Zhiyitu Hadoop Máy All-in-one [36].

Máy cơ sở dữ liệu All-in-one thường phù hợp với mô hình dữ liệu của các mối quan hệ lưu trữ phức tạp. Đồng thời, Máy tính cần tính giao dịch và tính nhất quán cao. Nói chung, cấu hình máy chủ công cụ cơ sở dữ liệu phụ thuộc vào nhu cầu đồng thời, và cấu hình máy chủ của các nút lưu trữ cơ sở dữ liệu phụ thuộc vào nhu cầu về kích thước dữ liệu [37]. Cơ sở dữ liệu máy All-in-one áp dụng đầy đủ Kiến trúc xử lý dữ liệu lớn phân tán, tích hợp phần cứng và phần mềm trong một hệ thống. Với sự tăng trưởng của dữ liệu người dùng và mở rộng kinh doanh, nó có thể được cải thiện bằng cách mở rộng chiều dài phần cứng, và cũng có thể đạt được tuyến tính mở rộng quy mô bằng cách thêm các nút theo chiều rộng, đảm bảo hiệu suất của độ trễ thấp, thông lượng cao và tính liên tục của doanh nghiệp [38]. Máy all in one là một sự kết hợp giữa phần mềm và phần cứng, được thiết kế hoàn toàn để xử lý lưu trữ dữ liệu hàng loạt. Và nó được tạo thành từ một bộ máy chủ tích hợp, thiết bị lưu trữ, hệ điều hành, hệ thống quản lý cơ sở dữ liệu và phần mềm cài đặt sẵn cho quản lý dữ liệu. Nó cung cấp giải pháp lưu trữ dữ liệu lớn, chủ yếu dành cho thị trường kho dữ liệu lớn. Và nó cao khả năng thông lượng tạo điều kiện giải quyết nút thắt cổ chai I/O vấn đề. Người dùng có thể chọn các loạt sản phẩm khác nhau theo yêu cầu, tùy chỉnh theo yêu cầu.

Tuy nhiên, máy cơ sở dữ liệu All-in-one cũng phải đối mặt với Thách thức. Trong thời đại dữ liệu lớn, lượng dữ liệu đang tăng lên một cách đáng kinh ngạc. Do đó nếu người dùng cần mở rộng Máy All-in-one, họ chỉ có thể thêm một tủ thiết bị, dẫn đến việc mở rộng không linh hoạt. Và bởi vì All-in-one Phần mềm được tích hợp cao, rất khó để triển khai trong môi trường khác.

Trong một số ngành công nghiệp, nhu cầu thay đổi nhanh hơn. Như vậy Mô hình kinh doanh sẽ thay đổi rất nhanh với nó. Sử dụng máy All-in-one sẽ hạn chế hành động của doanh nghiệp ngược lại. Nhưng trong một số ứng dụng tương đối trưởng thành và ổn định, máy All-in-one thể hiện giá trị của việc đơn giản hóa CNTT.

### 2.4 Cụm cơ sở dữ liệu mới của kiến trúc MPP

MPP là hệ thống xử lý song song quy mô lớn, là một loại phương pháp để mở rộng tài nguyên hệ thống, chủ yếu là xử lý song song. Điều này có nghĩa là một máy tính duy nhất có nhiều bộ xử lý mạng [39]

Mở rộng theo chiều ngang là mục tiêu thiết kế chính của cơ sở dữ liệu kiến trúc MPP. Nó được liên kết bởi nhiều máy chủ SMP thông qua internet của các nút cố định, cộng tác cho các nhiệm vụ thông thường. Từ cấp độ của người dùng, nó là một hệ thống máy chủ, hỗ trợ mô hình quan hệ dữ liệu nghiêm ngặt. Đặc điểm lớn nhất là mỗi nút chỉ có thể truy cập tài nguyên địa phương của riêng họ, không có chia sẻ.

Cụm cơ sở dữ liệu sử dụng kiến trúc MPP có thể hỗ trợ hiệu quả việc lưu trữ dữ liệu có cấu trúc hàng loạt ở mức PB. Nó dựa trên kiến trúc Shared Nothing. Bằng công nghệ xử lý dữ liệu lớn của lưu trữ cột, các chỉ số hạt thô, v.v. và kết hợp mô hình điện toán phân tán của nó với hiệu suất cao, Nó hoàn thành hỗ trợ kỹ thuật của bộ nhớ ứng dụng của lớp phân tích. Môi trường hoạt động chủ yếu là PC giá rẻ, và nó có những ưu điểm là cao Hiệu suất và khả năng mở rộng cao [40]. Nó có thể cải thiện hiệu suất xử lý dữ liệu, cải thiện tải quy trình dữ liệu, nâng cao hiệu quả xử lý dữ liệu hàng loạt và giảm chi phí xử lý tổng thể cho mỗi TB. Do đó, nó đã được sử dụng rộng rãi trong kho dữ liệu doanh nghiệp thuộc lĩnh vực phân tích dữ liệu thế hệ mới và có cấu trúc.

# Triển vọng của lưu trữ dữ liệu lớn Công nghệ

Do số lượng lớn không có cấu trúc và dữ liệu bán cấu trúc, cơ sở dữ liệu quan hệ truyền thống đã được bất lực. Tuy nhiên, công nghệ lưu trữ mới, chẳng hạn như cơ sở dữ liệu NoSQL(phi quan hệ) và hệ thống tệp phân tán, vượt trội hơn so với lưu trữ truyền thống, bất kể khả năng chịu lỗi, khả năng mở rộng và tính di động của dữ liệu. Và nó phù hợp để lưu trữ dữ liệu liên tục và quản lý lưu trữ dữ liệu khối lượng lớn [41].Nhưng đối với hiệu suất xử lý dữ liệu theo thời gian thực, có một khoảng cách nhất định giữa công nghệ lưu trữ mới và cơ sở dữ liệu quan hệ. Vì vậy, mỗi người đều có mặt tốt của nó. Hiện tại, chiếc sự kết hợp giữa cơ sở dữ liệu quan hệ và hệ thống xử lý song phân tán có thể nâng cao hiệu quả lưu trữ, Tốc độ xử lý và tốc độ phân tích [42]. Cách tiếp cận này cũng đang là xu hướng hot trong tương lai. Vấn đề cốt lõi của lưu trữ dữ liệu lớn công nghệ là hiệu suất. Một kỹ thuật và nền tảng duy nhất không còn có thể đáp ứng nhu cầu tăng trưởng bùng nổ dữ liệu và yêu cầu phân tích và lưu trữ dữ liệu từ các nhà khai thác. Trong quá trình phát triển tiếp theo, cơ sở dữ liệu kiểu mới sẽ dần được trộn lẫn với hệ sinh thái Hadoop hoặc hệ sinh thái Spark [43], cung cấp SQL và hỗ trợ giao dịch cho ứng dụng. Sử dụng Hadoop hoặc Spark để đạt được quá trình xử lý dữ liệu bán cấu trúc, phi cấu trúc. Vì vậy, trong tương lai, lưu trữ cũng sẽ được phát triển theo hướng kết hợp giữa cụm cơ sở dữ liệu song MPP và cụm Hadoop/Spark. Ngoài ra, với sự phát triển bùng nổ của dữ liệu doanh nghiệp, máy tính All-in-one dữ liệu lớn chắc chắn sẽ trở thành một công nghệ hot, và được sử dụng rộng rãi.

Bằng cách nghiên cứu công nghệ lưu trữ dữ liệu mới, bài báo này tóm tắt từng chút một và đối chiếu hệ thống tệp phân tán, cơ sở dữ liệu NoSQL (phi quan hệ), máy tính cở sở dữ liệu All-in-one và loại mới cụm cơ sở dữ liệu của kiến trúc MPP từ các góc độ khác nhau. Và xu hướng nghiên cứu trong tương lai đã được đưa ra. Lưu trữ dữ liệu lớn vẫn đang trong giai đoạn phát triển nhanh chóng. Không gian phát triển rất lớn, và cần các nhà nghiên cứu khám phá liên tục.

# Demo: Hadoop trên Windows

Big Data đang trở thành một phần thế mạnh và là tài sản rất lớn của mỗi công ty, tổ chức, cá nhân…, và Hadoop là một trong các công nghệ cốt lõi cho việc lưu trữ và truy cập dữ liệu lớn, đặc biệt là kiến trúc phân tán.

## ****1. Giới thiệu Hadoop****

**1.1 Hadoop là gì?**

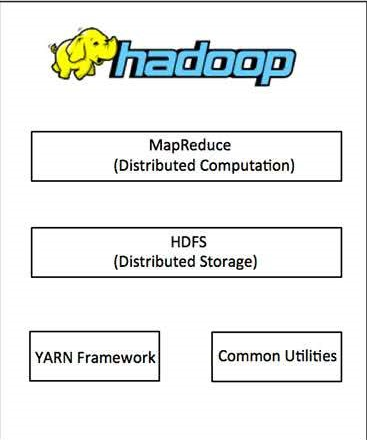
Hadoop là một Apache framework mã nguồn mở cho phép phát triển các ứng dụng phân tán để lưu trữ và quản lý các tập dữ liệu lớn. Hadoop hiện thực mô hình MapReduce, mô hình mà ứng dụng sẽ được chia nhỏ ra thành nhiều phân đoạn khác nhau được chạy song trên nhiều node khác nhau.

1.2 Chức năng nhiệm vụ của Hadoop

* Xử lý và làm việc khối lượng dữ liệu khổng lồ tính bằng Petabyte.
* Xử lý trong môi trường phân tán, dữ liệu lưu trữ ở nhiều phần cứng khác nhau, yêu cầu xử lý đồng bộ
* Các lỗi xuất hiện thường xuyên.
* Băng thông giữa các phần cứng vật lý chứa dữ liệu phân tán có giới hạn.

1.3 Kiến trúc Hadoop

Một cụm Hadoop nhỏ gồm 1 master node và nhiều worker/slave node. Toàn bộ cụm chứa 2 lớp, một lớp MapReduce Layer và lớp kia là HDFS Layer. Mỗi lớp có các thành phần liên quan riêng. Master node gồm JobTracker, TaskTracker, NameNode, và DataNode. Slave/worker node gồm DataNode, và TaskTracker. Cũng có thể slave/worker node chỉ là dữ liệu hoặc node để tính toán.



**Hadoop framework gồm 4 module:**

**Module 1: Hadoop Distributed File System (HDFS)**

Đây là hệ thống file phân tán cung cấp truy cập thông lượng cao cho ứng dụng khai thác dữ liệu. **Hadoop Distributed File System (HDFS)** là hệ thống tập tin ảo. Khi chúng ta di chuyển 1 tập tin trên HDFS, nó tự động chia thành nhiều mảnh nhỏ. Các đoạn nhỏ của tập tin sẽ được nhân rộng và lưu trữ trên nhiều máy chủ khác để tăng sức chịu lỗi và tính sẵn sàng cao.

HDFS sử dụng kiến trúc master/slave, trong đó master gồm một NameNode để quản lý hệ thống file metadata và một hay nhiều slave DataNodes để lưu trữ dữ liệu thực tại.

Một tập tin với định dạng HDFS được chia thành nhiều khối và những khối này được lưu trữ trong một tập các DataNodes. NameNode định nghĩa ánh xạ từ các khối đến các DataNode. Các DataNode điều hành các tác vụ đọc và ghi dữ liệu lên hệ thống file. Chúng cũng quản lý việc tạo, huỷ, và nhân rộng các khối thông qua các chỉ thị từ NameNode.

**Module 2: Hadoop MapReduce**

Đây là hệ thống dựa trên YARN dùng để xử lý song các tập dữ liệu lớn. Là cách chia một vấn đề dữ liệu lớn hơn thành các đoạn nhỏ hơn và phân tán nó trên nhiều máy chủ. Mỗi máy chủ có 1 tập tài nguyên riêng và máy chủ xử lý dữ liệu trên cục bộ. Khi máy chủ xử lý xong dữ liệu, chúng sẽ gởi trở về máy chủ chính.

MapReduce gồm một single master (máy chủ) JobTracker và các slave (máy trạm) TaskTracker trên mỗi cluster-node. Master có nhiệm vụ quản lý tài nguyên, theo dõi quá trình tiêu thụ tài nguyên và lập lịch quản lý các tác vụ trên các máy trạm, theo dõi chúng và thực thi lại các tác vụ bị lỗi. Những máy slave TaskTracker thực thi các tác vụ được master chỉ định và cung cấp thông tin trạng thái tác vụ (task-status) để master theo dõi.

JobTracker là một điểm yếu của Hadoop Mapreduce. Nếu JobTracker bị lỗi thì mọi công việc liên quan sẽ bị ngắt quãng.

**Module 3: Hadoop Common**

Đây là các thư viện và tiện ích cần thiết của Java để các module khác sử dụng. Những thư viện này cung cấp hệ thống file và lớp OS trừu tượng, đồng thời chứa các mã lệnh Java để khởi động Hadoop.

**Module 4: Hadoop YARN**

Quản lý tài nguyên của các hệ thống lưu trữ dữ liệu và chạy phân tích.

1.4 Cơ chế hoạt động của Hadoop

**Giai đoạn 1:**

Một user hay một ứng dụng có thể submit một job lên Hadoop (hadoop job client) với yêu cầu xử lý cùng các thông tin cơ bản:

Nơi lưu (location) dữ liệu input, output trên hệ thống dữ liệu phân tán.

Các java class ở định dạng jar chứa các dòng lệnh thực thi các hàm map và reduce.

Các thiết lập cụ thể liên quan đến job thông qua các thông số truyền vào.

**Giai đoạn 2:**

Hadoop job client submit job (file jar, file thực thi) và các thiết lập cho JobTracker. Sau đó, master sẽ phân phối tác vụ đến các máy slave để theo dõi và quản lý tiến trình các máy này, đồng thời cung cấp thông tin về tình trạng và chẩn đoán liên quan đến job-client.

**Giai đoạn 3:**

TaskTrackers trên các node khác nhau thực thi tác vụ MapReduce và trả về kết quả output được lưu trong hệ thống file.

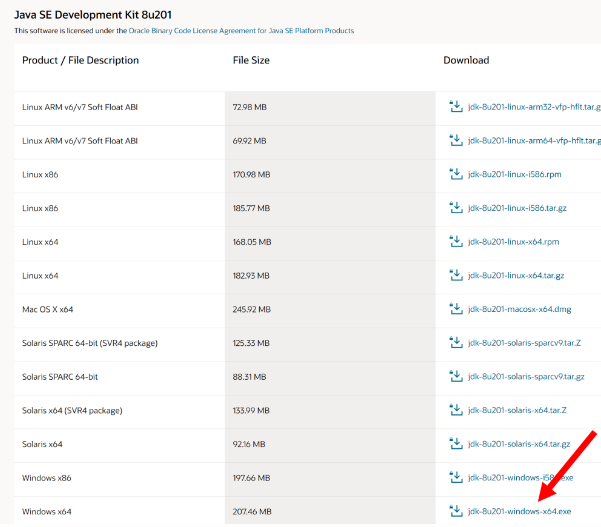
Khi “chạy Hadoop” có nghĩa là chạy một tập các trình nền – daemon, hoặc các chương trình thường trú, trên các máy chủ khác nhau trên mạng của bạn. Những trình nền có vai trò cụ thể, một số chỉ tồn tại trên một máy chủ, một số có thể tồn tại trên nhiều máy chủ.

Các daemon bao gồm:

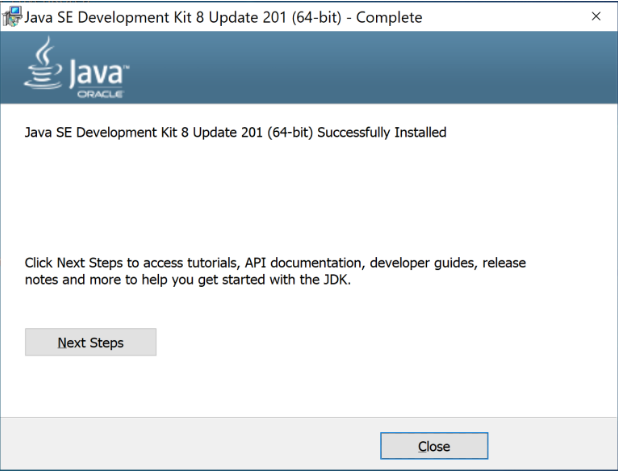
* NameNode
* DataNode
* SecondaryNameNode
* JobTracker
* TaskTracker

## 2. Cài đặt JDK bản 1.8 (bắt buộc)

Hadoop sử dụng JDK 1.8



Tiến hành cài đặt



Hoàn thành cài đặt

## 3. Thiết lập biến môi trường cho Java JDK

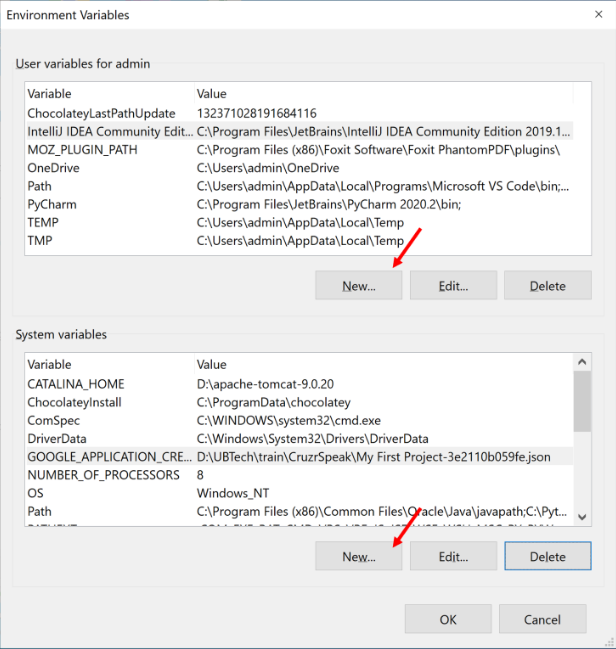
Cần cấu hình biến môi trường JAVA\_HOME cho Java JDK

Bấm chuột phải vào Computer / chọn Properties

**Chọn Advanced System Settings**

**Chọn Environment Variables…**

Màn hình Environment Variables sẽ xuất hiện như dưới đây:

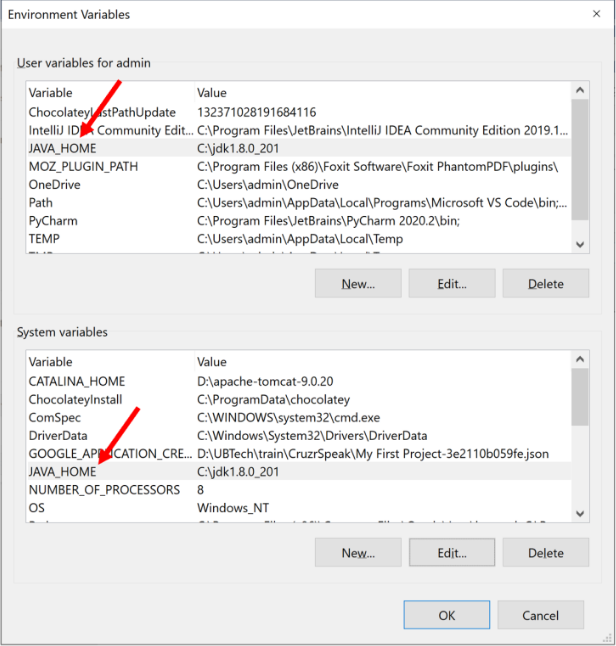


**Trong mục user và system variables ta cấu hình JAVA\_HOME trỏ tới nơi cài đặt JDK (bằng cách nhấn vào nút New…)**

Variable name: JAVA\_HOME

Variable value: C:\jdk1.8.0\_201

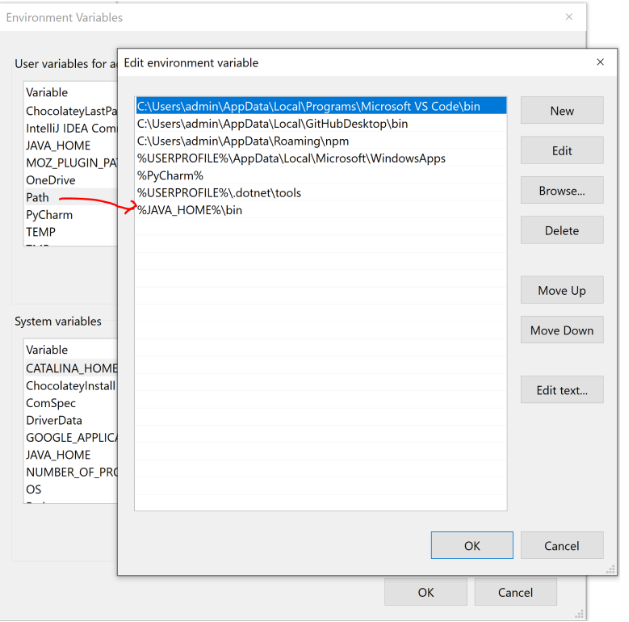
Kết quả:



Sau đó bấm OK liên tục để đóng các cửa sổ cũng như xác nhận sự thay đổi

Tiếp theo cấu hình Path (cho cả user và system variable). Tìm tới biến Path, nhấn Edit:

Thêm lệnh: **%JAVA\_HOME%\bin**

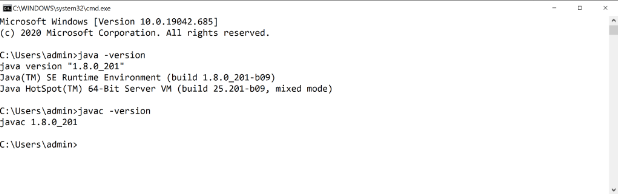


Trong màn hình command line lên gõ các lệnh trên để thấy kết quả:

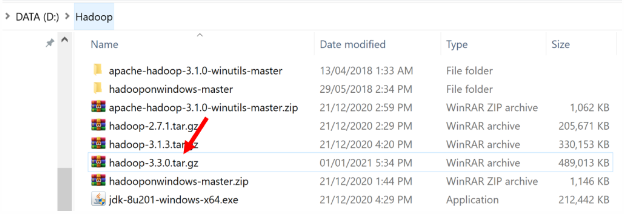
**java -version**

**javac -version**

Kết quả:

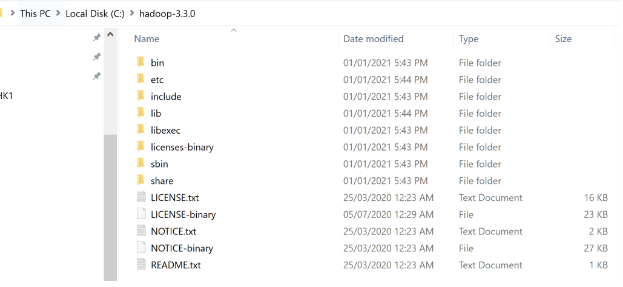


## 4 Tải Hadoop và giải nén vào ổ C



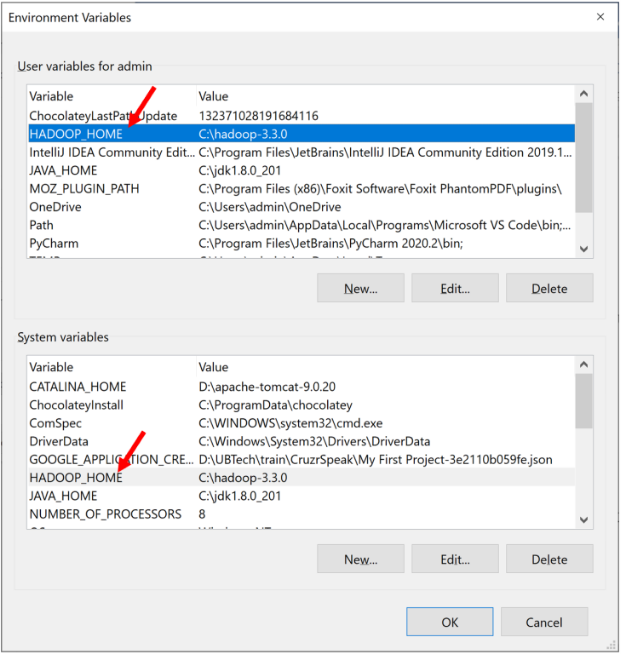
Giải nén: chọn Extract files…

Sau khi giải nén



## 5 Thiết lập biến môi trường cho Hadoop

Tương tự như JAVA JDK, ta cần cấu hình biến môi trường cho Hadoop (**HADOOP\_HOME)**



%HADOOP\_HOME%\bin

%HADOOP\_HOME%\sbin

Nhấn Ok để đóng tất cả các cửa sổ

Mở CMD để test lại:

[](https://duythanhcse.files.wordpress.com/2021/01/hadoop-31.png)

Gõ lệnh: hadoop version

Ta thấy kết quả là hadoop có version 3.3.0, như vậy cấu hình biến môi trường đã xong.

## 6 Cấu hình các tập tin cho Hadoop

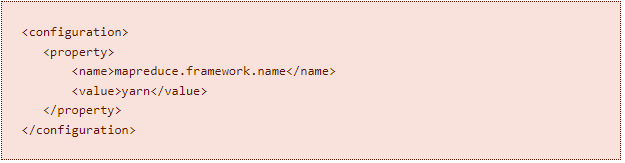
Trong thư mục **C:/Hadoop-3.3.0/etc/hadoop** lần lượt chỉnh sửa các file:

* core-site.xml
* mapred-site.xml
* hdfs-site.xml
* yarn-site.xml
* hadoop-env.cmd

Cấu hình **core-site.xml**như dưới đây:



Cấu hình **mapred-site.xml**như dưới đây:

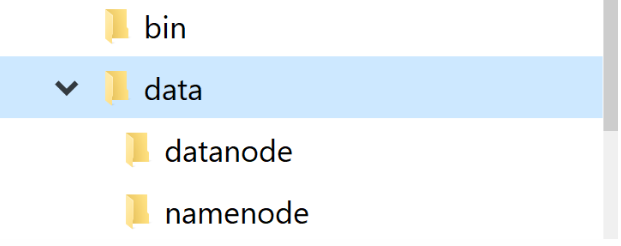


Cấu hình **hdfs-site.xml** như dưới đây:

Tạo thư mục “data” trong “C:/Hadoop-3.3.0”

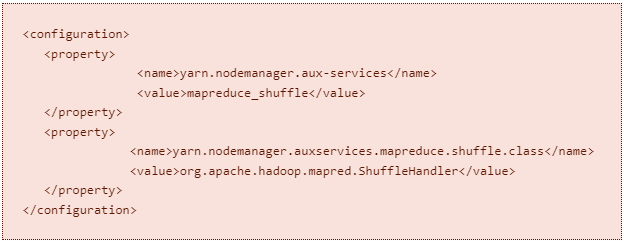
Tạo thư mục con “datanode” trong “C:/Hadoop-3.3.0/data”

Tạo thư mục con “namenode” trong “C:/Hadoop-3.3.0/data”



Sau đó cấu hình **hdfs-site.xml**như sau: 

Cấu hình **yarn-site.xml**như dưới đây:



Cấu hình **hadoop-env.cmd**:

Mở file này lên và tìm tới lệnh:

* set JAVA\_HOME=%JAVA\_HOME%

sửa %JAVA\_HOME% thành đường dẫn cài JDK trong ổ C:

* set JAVA\_HOME= C:/jdk1.8.0\_201

7) Cập nhật các Hadoop Configurations

Tải về giải nén ra thấy thư mục **bin** ở bên trong

Chép đè thư mục bin này trong thư mục bin của C:\hadoop-3.3.0\bin

Sau đó format lại **namenode và datanode**: mở command line lên, gõ lệnh sau:

* **hdfs namenode –format**
* **hdfs datanode -format**

[](https://duythanhcse.files.wordpress.com/2021/01/hadoop-33.png)

Bước format này chỉ cần làm 1 lần.

**\*Tiếp theo sao chép file:**

“C:/hadoop-3.3.0/share/hadoop/yarn/timelineservice/ hadoop-yarn-server-timelineservice-3.3.0.jar”

vào “C:/hadoop-3.3.0/share/hadoop/yarn/hadoop-yarn-server-timelineservice-3.3.0.jar”

8) Hoàn thành cài đặt Hadoop và test thử nghiệm với start-all.cmd

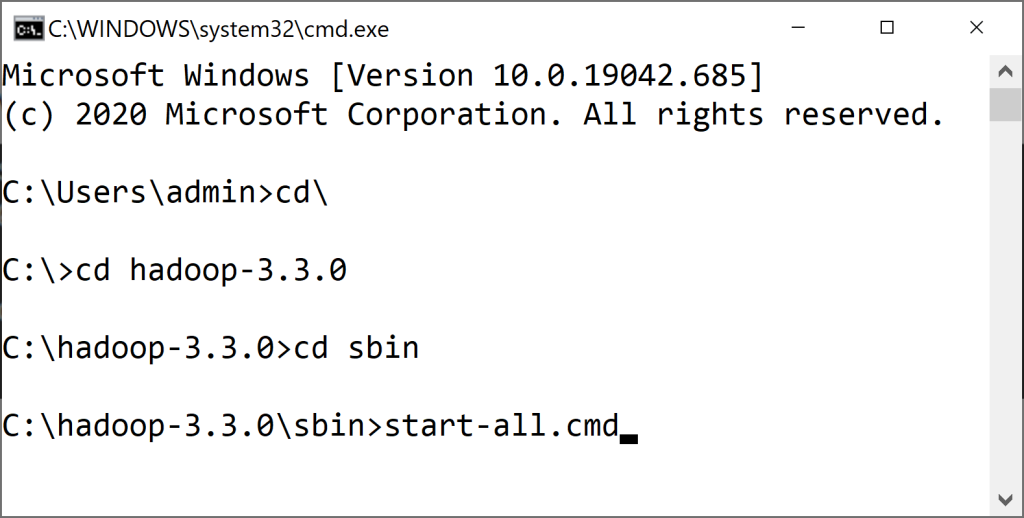
Để test Hadoop, ta mở command line và di chuyển tới thư mục

C:/hadoop-3.3.0/sbin

Sau đó gõ lệnh:

* Start-all.cmd

Chi tiết xem hình các lệnh dưới đây:

[](https://duythanhcse.files.wordpress.com/2021/01/hadoop-34.png)

Sau khi gõ lệnh trên, hệ thống sẽ chạy Hadoop

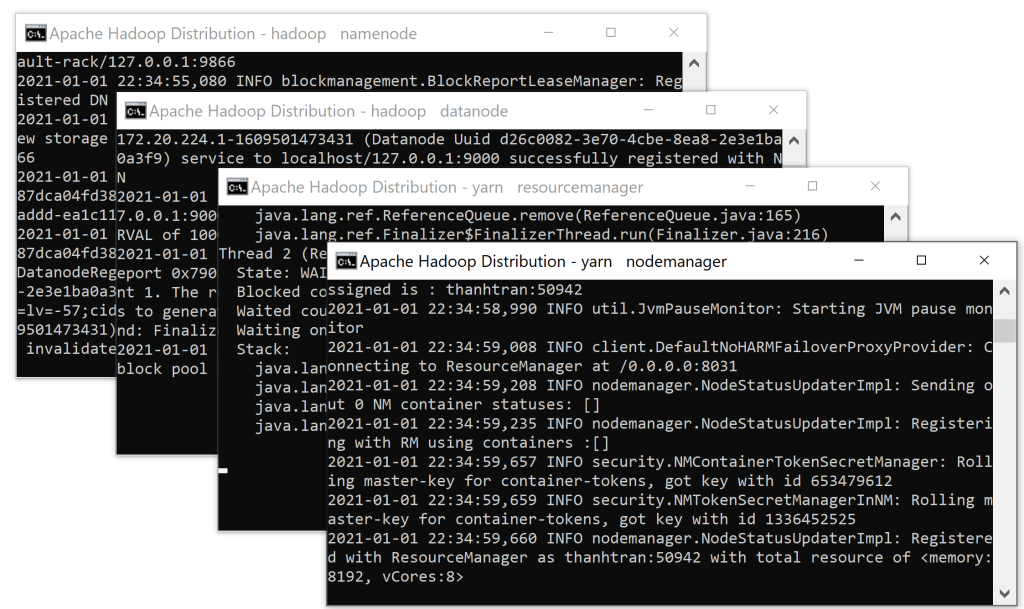
Phải đảm bảo các ứng dụng sau được chạy:

– Hadoop Namenode

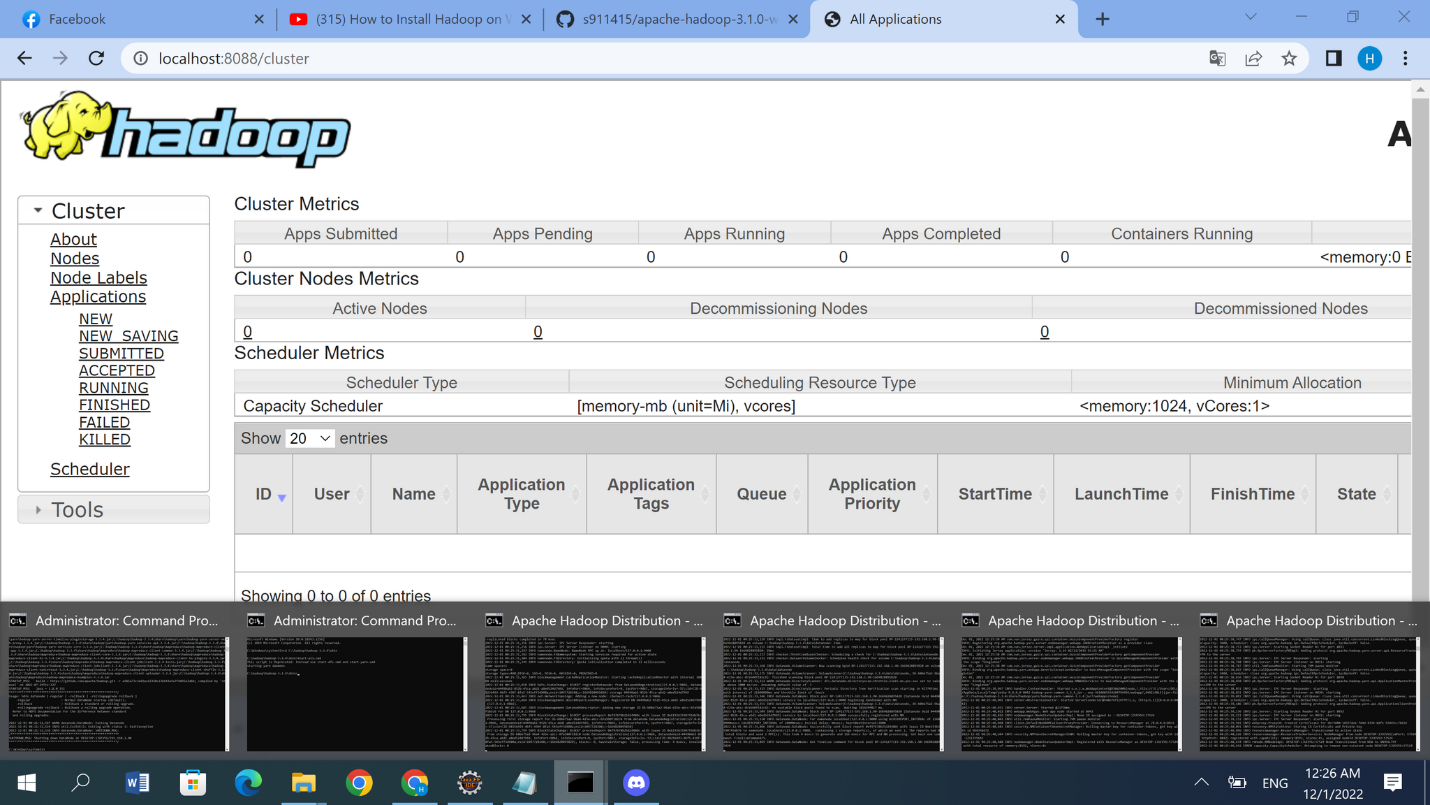
– Hadoop datanode

– YARN Resource Manager

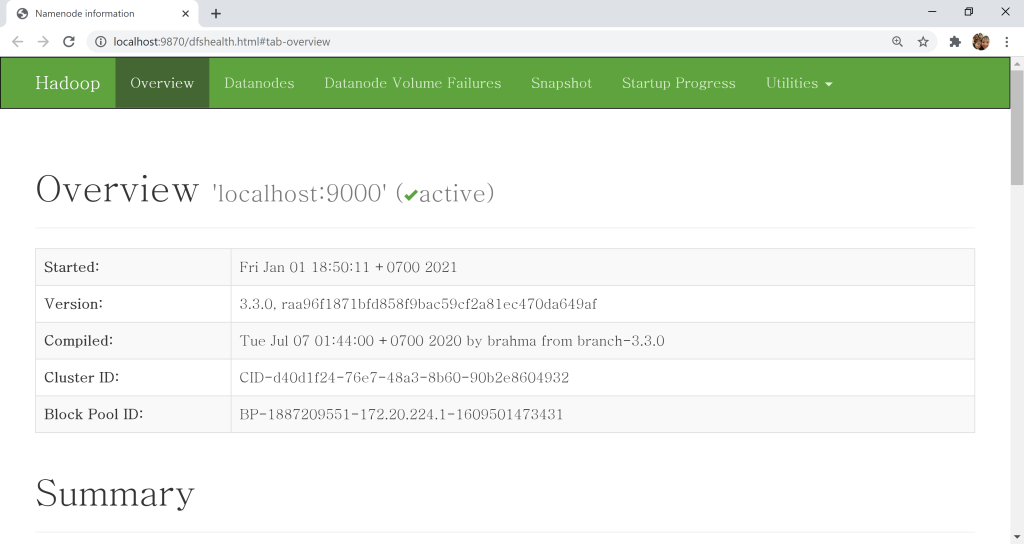
– YARN Node Manager

[](https://duythanhcse.files.wordpress.com/2021/01/hadoop-35.png)

Như vậy ta đã khởi động thành công: vào [http://localhost:8088](http://localhost:8088/)



vào [http://localhost:9870](http://localhost:9870/)

[](https://duythanhcse.files.wordpress.com/2021/01/hadoop-37.png)

Ngoài ra ta có thể tách chạy 2 lệnh:

– Khởi động namenode và datanode :

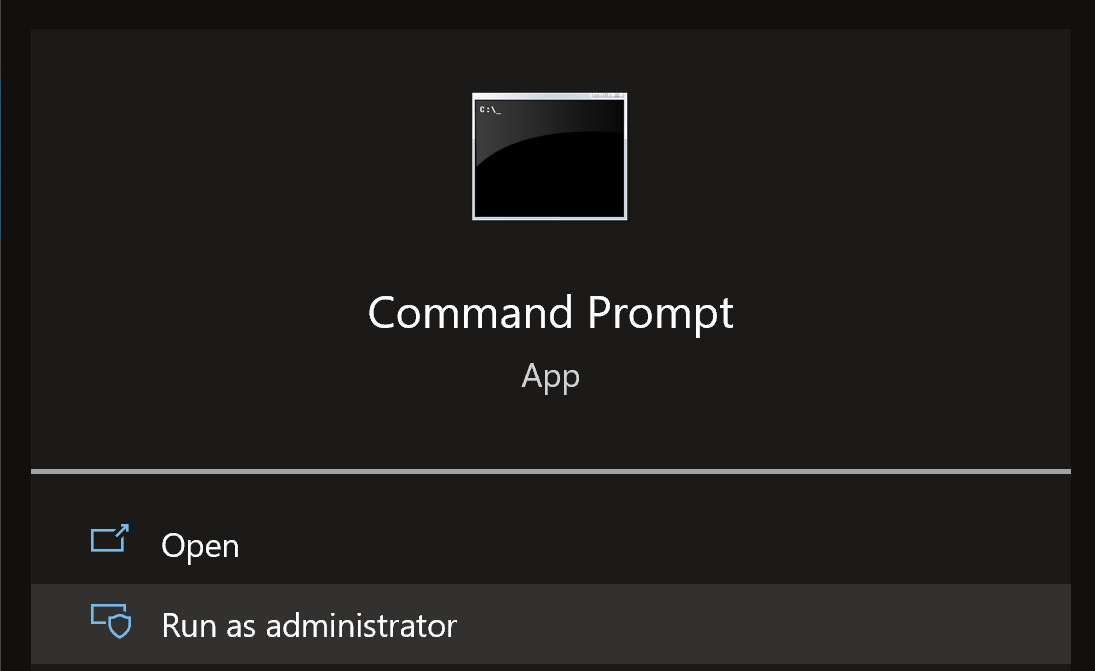
start-dfs.cmd

– Khởi động yarn bằng lệnh:

start-yarn.cmd

**Chạy Wordcount Program on Hadoop**

**Khởi động cmd bằng quyền Admin**

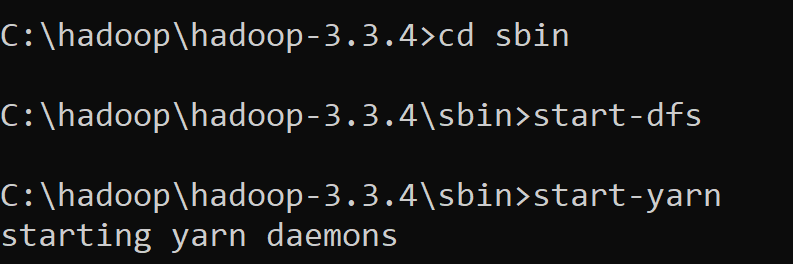
****

Trỏ đến thư mục sbin trong Hadoop

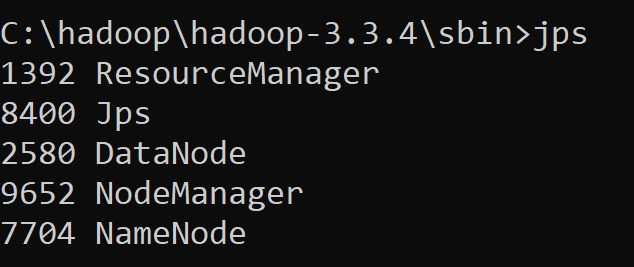
C:\>cd C:\hadoop\hadoop-3.3.4\bin

C:\hadoop\hadoop-3.3.4>cd sbin

Chạy 2 module dfs và yarn



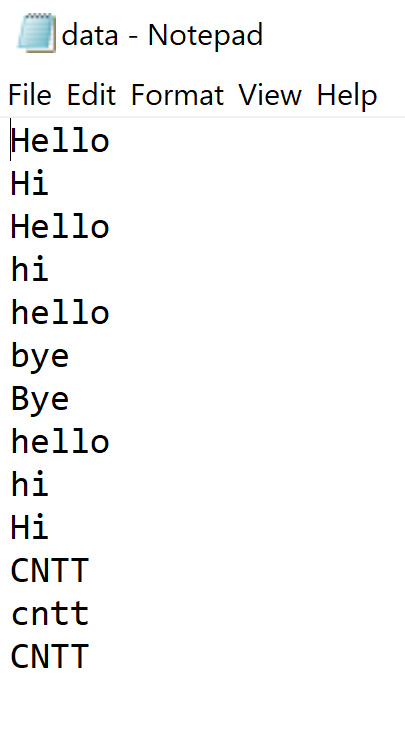
Khởi động jps



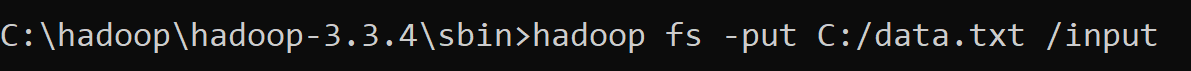
Tạo input



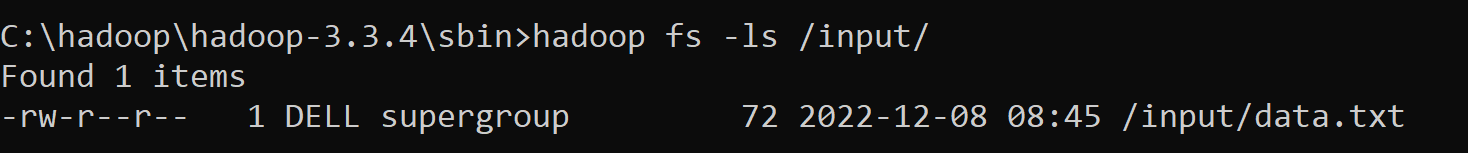
Tạo một file data.txt để count số chữ



Khai báo data.txt



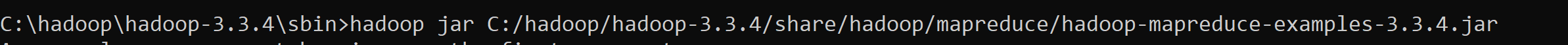
Kiểm tra khai báo thành công



Kiểm tra nội dung

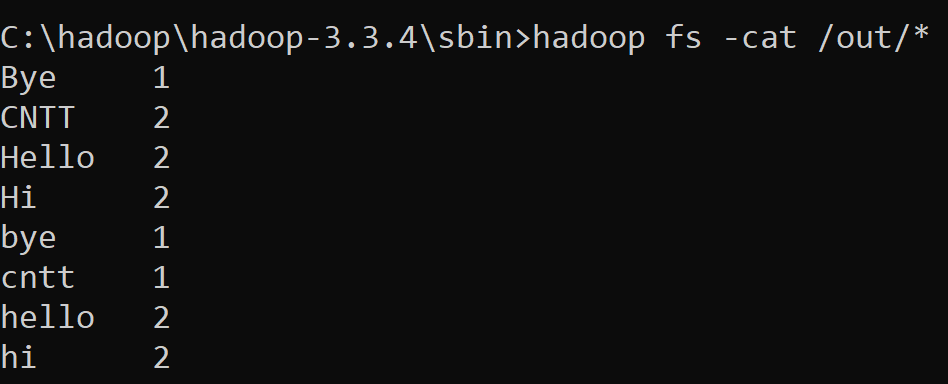


Khởi động mole Mapreduce





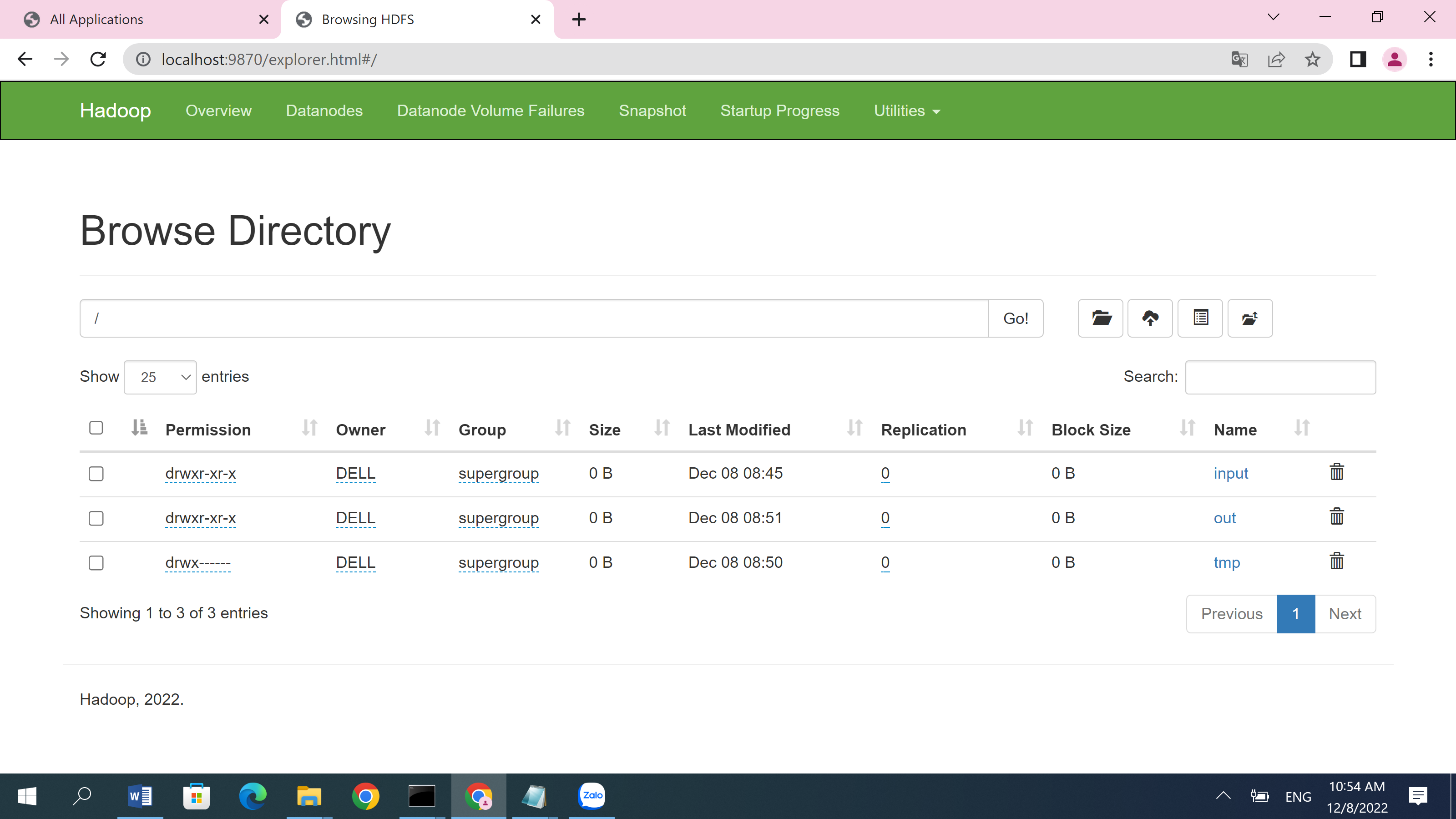
Đếm số chữ trong file data.txt



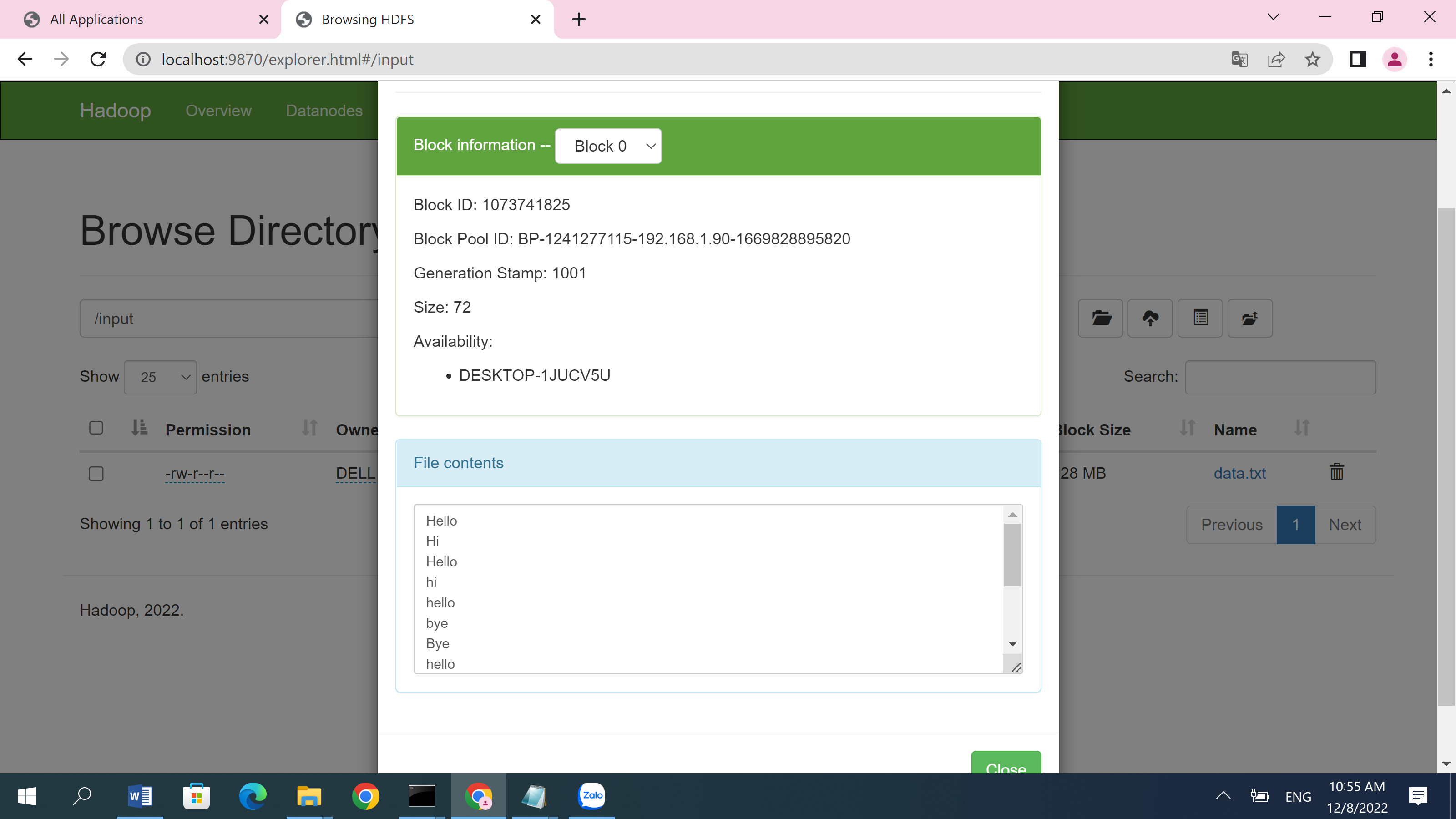
Truy cập localhost:8088 để thấy Application mình vừa tạo



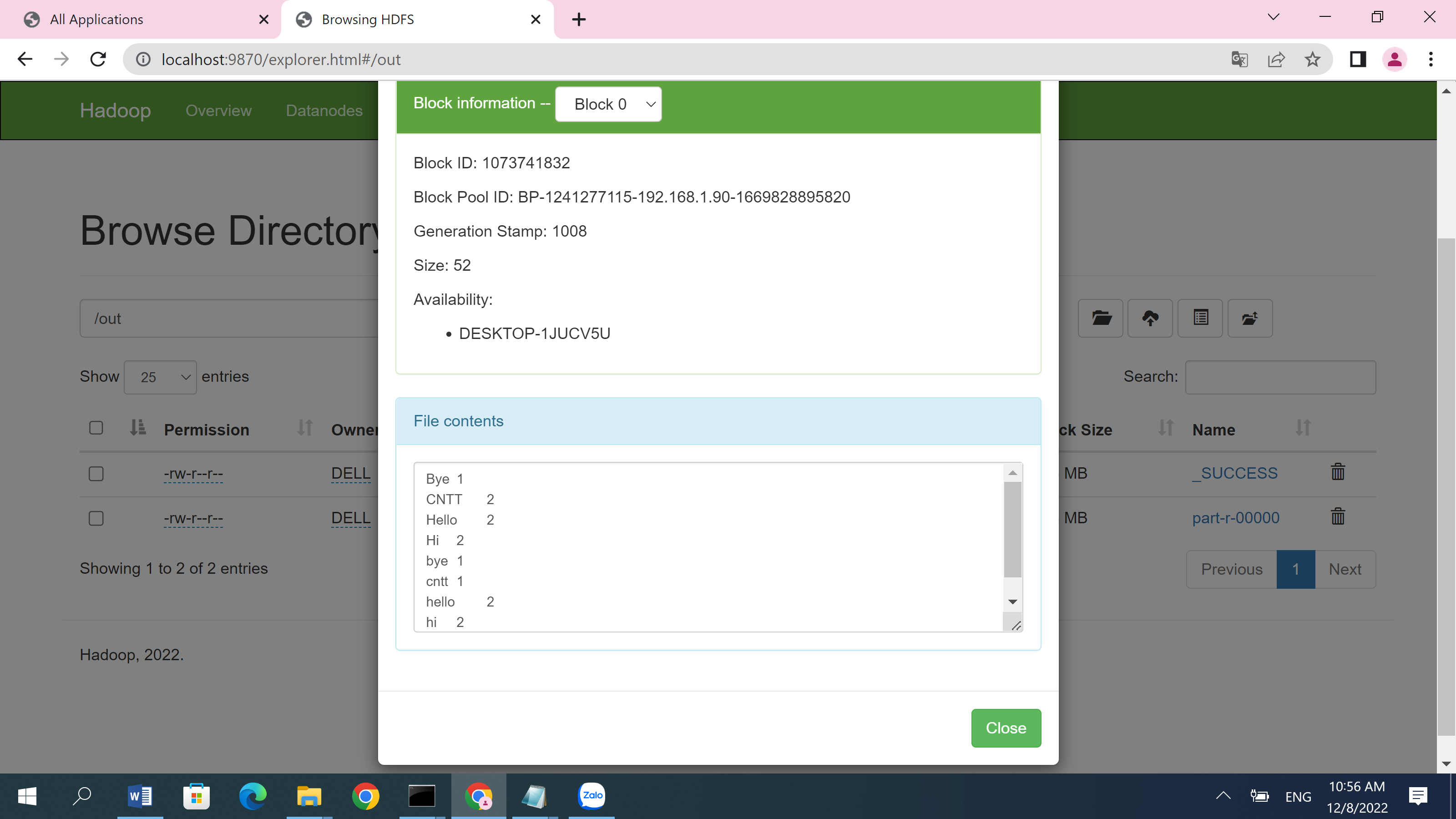
Truy cập localhost:9870



Trong phần /input sẽ thấy được thông tin và dữ liệu file data.txt



Và ở phần /out sẽ thấy file data.txt sau khi đã đếm theo Mapreduce



# Hướng phát triển của Big DATA

**Những lĩnh vực ứng dụng Big Data trên toàn cầu**

Big Data được sử dụng ở nhiều lĩnh vực với đa dạng mục đích trên toàn cầu. Trên thực tế, việc áp dụng Big Data trong kinh doanh cũng đã tạo ra nhiều chuyển biến ấn tượng trong ngành, giúp hiểu rõ hơn tệp khách hàng và góp phần làm tăng hiệu quả, năng suất của doanh nghiệp. Ứng dụng rộng rãi của Big Data vào các ngành có thể kể đến như:

**Ngành bán lẻ**

Từ những dữ liệu thu thập được tại thị trường cạnh tranh, sau quá trình phân tích và liên kết các mỗi quan hệ trong từng hành vi trải nghiệm của khách hàng. Doanh nghiệp đã dễ dàng đoán được xu hướng mua sắm trong tương lai và biết được mức độ hài lòng của họ. Từ đó đưa ra những chính sách tối ưu hơn nhằm thu hút khách hàng. Giúp cải thiện hiệu suất và nâng cao hiệu quả bán hàng.

**Ngành y tế**

Big Data được các nước phát triển và đang phát triển ứng dụng vào y tế để theo dõi, quản lý và dự đoán tình hình sức khỏe của bệnh nhân. Điều này không những giúp bác sĩ xác định được phương hướng điều trị mà còn giúp cải thiện quá trình chăm sóc sức khỏe, giảm thiểu thời gian làm thủ tục xét nghiệm rườm rà truyền thống.

**Ngành tài chính**

Đây là ngành được áp dụng hiệu quả nhất của Big Data, từ việc thu tiền mặt đến quản lý tài chính của từng khách hàng, giúp phân tích và xác định địa điểm nên đặt chi nhánh theo nhu cầu tập trung, tiên đoán lượng tiền cần cung ứng cho từng chi nhánh và cây ATM, phát hiện hoạt động gian lận. Đặt biệt quan trọng hơn cả là Big Data giúp hỗ trợ xử lý, lưu trữ và phân tích nguồn dữ liệu khổng lồ đến từ các hoạt động hàng ngày.

**Ngành thương mại điện tử**

Để đạt được mục tiêu kinh doanh, ngành thương mại điện tử phải ứng dụng Big Data trong việc thu thập dữ liệu và yêu cầu của khách hàng để phân tích, nhằm tạo ra một mô hình tiếp thị với hiệu suất cao hơn. Nâng cao lợi thế cạnh tranh cho doanh nghiệp bằng cách hiểu được khách hàng và đoán trước xu hướng tiêu dùng trong tương lai gần.

**Tương lai và xu hướng của Big Data**

**Phát triển Internet kết hợp vạn vật (IoT) và điện toán đám mây**

Với khả năng cho phép truy cập và sao lưu dữ liệu trên nhiều thiết bị khác nhau, không mấy ngạc nhiên khi Big Data được dự đoán sẽ phát triển kết hợp với IoT và điện toán đám mây. Giúp lấp đầy khoảng trống của việc thu thập dữ liệu và cho phép người dùng dễ dàng giao tiếp, nhập dữ liệu từ nhiều nguồn và nhiều thiết bị. Việc này cũng tạo ra một cuộc cách mạng về sự đa kết nối của các công nghệ 4.0.

**Bị chi phối bởi các phần mềm nguồn mở (Open Source Applications)**

Phần mềm nguồn mở cho phép bất cứ ai đều có thể nghiên cứu, thay đổi và cải tiến khi sử dụng. Forrester Research, xu hướng này vẫn đang tiếp tục tăng với tỉ lệ là 32,9% mỗi năm. Vì thế, tương lai của Big Data cũng sẽ bị chi phối bởi những phần mềm mã nguồn mở, điển hình như Apache Hadoop, Spark,….

**Kết hợp mạnh mẽ với phân tích dự đoán (Predictive Analytics)**

Việc khai thác dữ liệu từ Big Data cũng được kết hợp với phân tích dự đoán trong tương lai. Phân tích dự đoán gồm nhiều kỹ thuật thống kê khác nhau, giúp phân tích các sự kiện trong quá khứ và hiện tại nhằm đưa ra dự đoán cho các sự kiện/xu hướng diễn ra trong tương lai. Mang lại tầm nhìn cụ thể cho doanh nghiệp, giúp họ dễ dàng đưa ra các quyết định chính xác hơn trong quá trình quản trị.

Không phải ngẫu nhiên mà Big Data được xem là xu hướng và là một phần không thể thiếu trong công nghệ ở tương lai. Big Data sẽ ngày càng phát triển mạnh mẽ với quy mô lớn hơn nữa, hứa hẹn mang đến nhiều điều bất ngờ cho nhân loại. Mong rằng bài viết này hữu ích với thông tin về Big Data mà bạn đang tìm kiếm.

**LỜI CẢM ƠN**

Để hoàn thành tiểu luận này, em xin gửi lời cảm ơn chân thành đến:

Xin cảm ơn giảng viên bộ môn - Thầy Phạm Trọng Huynh đã giảng dạy tận tình, chi tiết để em có đủ kiến thức và vận dụng chúng vào bài tiểu luận này.

Do chưa có nhiều kinh nghiệm làm để tài cũng như những hạn chế về kiến thức, trong bài tiểu luận chắc chắn sẽ không tránh khỏi những thiếu sót. Rất mong nhận được sự nhận xét, ý kiến đóng góp, phê bình từ phía Thầy để bài tiểu luận được hoàn thiện hơn.

Lời cuối cùng, em xin kính chúc thầy nhiều sức khỏe, thành công và hạnh phúc.

**Nguồn tham khảo**

**References**

[1] Zhang X, Xu F. Survey of Research on Big Data Storage [C] // International Symposium on Distributed Computing and Applications To Business, Engineering & Science. IEEE Computer Society, 2013: 76-80.

[2] Biesdorf S, Court D, Willmott P. Big data: What's your plan? [J]. Mckinsey Quarterly, 2013 (2): 40-51. [3] TU Xinli, Liu Bo, Lin Weiwei. Survey of Big Data [J]. Application Research of Computers, 2014, 31 (6): 1612-1616.

[4] Garcia H, Ludu A. The Google file system [C] // Acm Sigops Operating Systems Review. ACM, 2003: 29-43.

[5] Tong Ming. Research and application of distributed storage based on HDFS [D]. Huazhong University of Science and Technology, 2012.

[6] Davies A, Orsaria A. Scale out with GlusterFS [J]. Linux Journal, 2013, 2013 (235): 1.

[7] Hows D, Membrey P, Plugge E, et al. GridFS [M]. Apress, 2013.

[8] Zhao Yang. Depth Profiles of TaoBao TFS [J]. Digital Users, 2013 (3). Internet of Things and Cloud Computing 2016; 4(3): 28-33 33

[9] Wang Bo, Li Xianguo, Zhang Xiao. Research on performance optimization of Lustre file system [J]. Microcomputer Applications, 2011, 27 (5): 31-33.

[10] Yu Qing. Analyses of Distributed File System FastDFS Architecture [J]. Programmer, 2010 (11): 63-65.

[11] Golov N, Rönnbäck L. Big Data Normalization for Massively Parallel Processing Databases [C] // International Workshop on Modeling & Management of Big Data. 2015.

[12] Thereska E, Gunawardena D S, Scott J W, et al. Distributed File System: US, US20120254116 [P]. 2012.

[13] Li Hongqi, Zhu Liping, Sun Guoyu, et al. Design and Implementation of Distributed Storage System Facing Vast Small Files [J]. Computer Engineering and Design, 2016 (1): 86-92.

[14] Qi Ying. Research on Low Latency Access Technology of Distributed File System with Vast Small Files [D]. University of Chinese Academy of Sciences, 2013.

[15] Weil S A, Brandt S A, Miller E L, et al. Ceph: A Scalable, High-Performance Distributed File System [C] // 7th Symposium on Operating Systems Design and Implementation (OSDI '06), November 6-8, Seattle, WA, USA. 2006: 307--320.

[16] Gpfs B. A Shared-Disk File System for Large Computing Clusters [C] // of the First Conference on File and Storage Technologies. 2010.

[17] Xu Chunling, Zhang Guangquan. Comparison and Analysis of Distributed File System Hadoop HDFS and Traditional File System Linux FS [J]. Journal of Soochow University (Engineering Science Edition), 2010, 30 (4): 5-9.

[18] Xiong Wen, Yu Zhibin, Xu Chengzhong. Feature Analysis and Performance Comparison of Several Common Distributed File System [J]. Journal of Integration Technology, 2012, 1 (4): 58-63.

[19] Sawicki A, Nowak T. NETWORK DISTRIBUTED FILE SYSTEM:, US20080320097[P]. 2008.

[20] Shi Xiaodong. Research on High Availability of Distributed File System [D]. Institute of Computing Technology, Chinese Academy of Sciences, 2002.

[21] Qin Xiongpai, Wang Huiju, Du Xiaoyong, et al. big data analytics --Competition and Coexistence of RDBMS and MapReduce [J]. Journal of Software, 2012, 23 (1): 32-45.

[22] Shen Derong, Yu Ge, Wang Xite, et al. Survey of Research on NoSQL System Supporting Big Data Management [J]. Journal of Software, 2013 (8): 1786-1803.

[23] Curé O, Kerdjoudj F, Faye D, et al. On The Potential Integration of an Ontology-Based Data Access Approach in NoSQL Stores [J]. International Journal of Distributed Systems & Technologies, 2012, 4 (3): 166-173.

[24] Wang Jieping, Li Haibo, Song Jie, et al. Research on Cloud Data Storage and Management Standardization [J]. Information Technology and Standardization, 2011 (9): 28-31.

[25] Liu Y, Zhu L, Jiang W. Column caching mechanism for column based database:, EP2743839 [P]. 2014. [26] Bhogal J, Choksi I. Handling Big Data Using NoSQL [C]// IEEE International Conference on Advanced Information NETWORKING and Applications Workshops. IEEE, 2015: 393-398.

[27] Amirian P, Basiri A, Winstanley A. Efficient Online Sharing of Geospatial Big Data Using NoSQL XML Databases [C] // Fourth International Conference on Computing for Geospatial Research and Application. IEEE, 2013: 152-152.

[28] Deka G C. A Survey of Cloud Database Systems [J]. It Professional, 2014, 16 (2): 50-57.

[29] Castelltort A, Laurent A. Fuzzy Historical Graph Pattern Matching A NoSQL Graph Database Approach for Fraud Ring Resolution [M] // Artificial Intelligence Applications and Innovations. Springer International Publishing, 2015.

[30] Dong-Hai L U, Xian-Bo H E. The Analysis of NoSQL Database [J]. Science & Technology of West China, 2011.

[31] Srivastava P P, Goyal S, Kumar A. Analysis of various NoSql database [C] // International Conference on Green Computing and Internet of Things. IEEE, 2015: 539-544.

[32] Chandra D G. BASE analysis of NoSQL database [J]. Future Generation Computer Systems, 2015, 52: 13–21.

[33] Gu Y, Wang X, Shen S, et al. Analysis of data replication mechanism in NoSQL database MongoDB [C] // IEEE International Conference on Consumer Electronics - Taiwan. IEEE, 2015.

[34] Han J, Haihong E, Le G, et al. Survey on NoSQL database [C] // Pervasive Computing and Applications (ICPCA), 2011 6th International Conference on. IEEE, 2011: 363-366.

[35] Hinshaw F D, Meyers D L, Zane B M. Programmable streaming data processor for database appliance having multiple processing unit groups: US, US7577667[P]. 2009.

[36] Zhang Dong, Qi Kaiyuan, Wu Nan, et al. Architecture and Key Technology of Yunhai Big Data All-in-one Machine [J]. Computer Research and Development, 2016, 53 (2): 374-389.

[37] Yue Junfeng, Zhao Junfeng, Zhao Wei, et al. Analysis of Database All-in-one Machine Technical Architecture [J]. Electric Power Information and Communication Technology, 2013, 11 (4): 60-64.

[38] Pu Siyu, Bai qionghua. A Kind of New-type Cloud Storage All-in-one Machine Backing Up Data Information Automatically:, CN204305087U[P]. 2015.

[39] Li C, Yang J, Han J, et al. The Distributed Storage System Based on MPP for Mass Data [C] // Proceedings of the 2012 IEEE Asia-Pacific Services Computing Conference). IEEE Computer Society, 2012: 384-387.

[40] Chen Z, Song L. A solution based on the MPP'S to storage mass data [C] // Mechatronic Science, Electric Engineering and Computer (MEC), 2011 International Conference on. IEEE, 2011: 868-871.

[41] Cheng Lianjuan. Application Practice and Beneficial Reference of Big Data Promotion in the U. S.: An Analysis from the Perspective of Library [J]. Information & Documentation Services, 2013, 34 (5): 110-112.

[42] Li G. Research Status and Scientific Thinking of Big Data [J]. Bulletin of Chinese Academy of Sciences, 2012.

[43] Chen Jirong, Yue Jiajin. Survey of Big Data Solution Based on Hadoop Ecosystem [J]. Computer Engineering and Science, 2013, 35 (10): 25-35.

Link trang web tham khảo:

* brain-mentors.com
* mastercode.vn
* bigdataviet.wordpress.com
* topdev.vn
* iboss.vn