ĐẠI HỌC BÁCH KHOA HÀ NỘI

## TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



**BÁO CÁO BÀI TẬP LỚN**

**Đề tài: Lưu trữ, xử lý và phân tích dữ liệu phim lẻ**

Lớp : 154050

Học phần : Lưu trữ và xử lý dữ liệu lớn

Mã học phần : IT4931

Giảng viên hướng dẫn :TS. TrầnViệt Trung

Danh sách thành viên nhóm19:

|  |  |
| --- | --- |
| Họ và tên | MSSV |
| Nguyễn Phi Hùng | 20200263 |
| Lê Nguyên Khang | 20204836 |
| Đỗ Huy Hòa | 20204747 |
| Đới Sỹ Bình | 20204715 |
| Nguyễn Lê Hưng | 20215398 |

**Hà Nội, tháng 12 năm 2024**

# MỤC LỤC

Mục lục

[TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG 1](#_Toc184751400)

[MỤC LỤC 2](#_Toc184751401)

[LỜI NÓI ĐẦU 3](#_Toc184751402)

[CHƯƠNG 1: TỔNG QUAN XÂY DỰNG HỆ THỐNG 4](#_Toc184751403)

[1.1. Tổng quan hệ thống 4](#_Toc184751404)

[1.2. Chi tiết về thành phần hệ thống 5](#_Toc184751405)

[CHƯƠNG 2: XÂY DỰNG CHƯƠNG TRÌNH VÀ HỆ THỐNG 10](#_Toc184751406)

[2.1. Luồng dữ liệu của hệ thống 10](#_Toc184751407)

[2.2. Khởi động hệ thống HDFS 11](#_Toc184751408)

[2.3. Các trải nghiệm khi xây dựng chương trình và hệ thống 12](#_Toc184751409)

2.4. Kinh nghiệm 12

[CHƯƠNG 3: NHẬN XÉT, ĐÁNH GIÁ VÀ HƯỚNG PHÁT TRIỂN 20](#_Toc184751410)

[3.1. Nhận xét, đánh giá 20](#_Toc184751411)

[3.2. Hướng phát triển 20](#_Toc184751412)

[DANH MỤC TÀI LIỆU THAM KHẢO 21](#_Toc184751413)

# LỜI NÓI ĐẦU

Trong quá khứ, khi Internet chưa phát triển, lượng dữ liệu được sinh ra là khá nhỏ và thưa thớt. Nhìn chung, con người có thể xử lý lượng dữ liệu này bằng tay hoặc bằng máy tính một cách dễ dàng. Tuy nhiên, trong kỷ nguyên số, sự bùng nổ của công nghệ truyền thông đã dẫn đến sự gia tăng đột ngột về lượng và đa dạng của dữ liệu người dùng. Điều này đặt ra yêu cầu về một hệ thống mạnh mẽ để có thể phân tích và xử lý hiệu quả những lượng dữ liệu lớn này.

Khái niệm Big Data thường liên quan đến ba khía cạnh chính: tốc độ sinh dữ liệu (velocity), lượng dữ liệu (volume), và độ đa dạng (variety). Lượng dữ liệu này có thể đến từ nhiều nguồn khác nhau như Google, Facebook, Twitter, hay từ các cảm biến và thiết bị IoT trong cuộc sống hàng ngày. Sự thật là doanh nghiệp nào có khả năng kiểm soát và tận dụng thông tin từ những lượng dữ liệu này sẽ có lợi thế lớn trong sự cạnh tranh.

Trong những năm gần đây, nhu cầu giải trí, đặc biệt là việc xem phim, đã trở thành một phần không thể thiếu trong cuộc sống của chúng ta. Với sự phát triển mạnh mẽ của các nền tảng chiếu phim trực tuyến và ngành công nghiệp điện ảnh, mỗi ngày có hàng triệu người trên khắp thế giới thưởng thức các bộ phim từ nhiều thể loại và phong cách khác nhau.

Không chỉ là một hình thức giải trí, phim ảnh còn phản ánh văn hóa, xã hội, và thị hiếu của khán giả. Việc hiểu rõ sở thích và hành vi của người xem đã trở thành một yếu tố quan trọng, không chỉ đối với các nhà làm phim mà còn với các nền tảng chiếu phim và các doanh nghiệp liên quan.

Để làm được điều này, phân tích dữ liệu phim từ các trang web đóng vai trò như một chiếc chìa khóa, giúp chúng ta giải mã xu hướng, dự đoán nhu cầu, và đưa ra những quyết định chính xác hơn. Trong bài thuyết trình hôm nay, em sẽ giới thiệu về cách chúng ta thu thập và phân tích dữ liệu này, cùng với những phát hiện thú vị có thể ứng dụng vào thực tế

# CHƯƠNG 1: TỔNG QUAN XÂY DỰNG HỆ THỐNG

## Tổng quan hệ thống

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Phông chữ

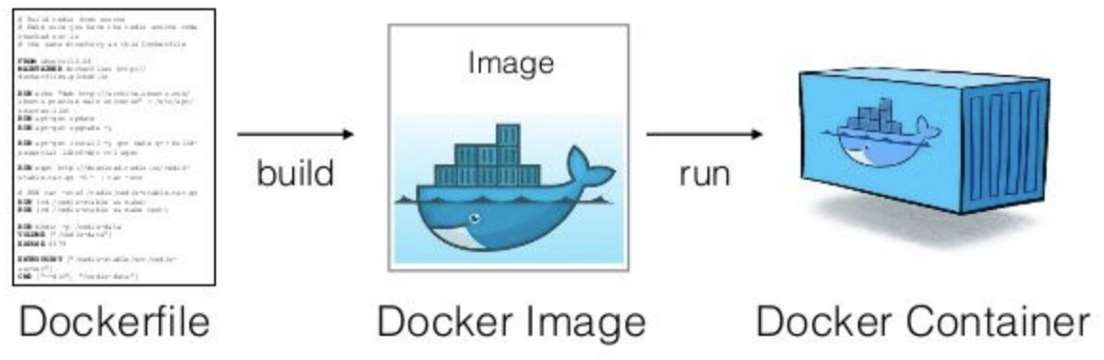
Mô tả được tạo tự động

Hệ thống được xây dựng bao gồm 4 thành phần chính, mỗi thành phần đảm nhận một vai trò quan trọng trong việc thu thập, xử lý, lưu trữ và trực quan hóa dữ liệu tuyển dụng từ các nguồn thông tin trên trang web. Dưới đây là mô tả chi tiết về các thành phần của hệ thống:Bộ phần thu thập dữ liệu: sử dụng BeautifulSoup4, là một thư viện để phân tích cú pháp các văn bảng dạng HTML và XML, chuyên dụng trong việc thu thập dữ liệu từ các trang web.

1. Bộ phận Thu thập Dữ liệu: Sử dụng thư viện BeautifulSoup4, chuyên dụng trong việc phân tích cú pháp HTML và XML để thu thập thông tin từ các trang web tuyển dụng.
2. Bộ phận Lưu trữ: Dữ liệu được lưu trữ vào Hadoop sử dụng HDFS File System (HDFS). Hadoop cung cấp khả năng lưu trữ phân tán, mở rộng, sao lưu và đảm bảo khả năng truy cập khi có sự mất kết nối từ một số máy.
3. Bộ phận Xử lý Dữ liệu: Sử dụng Apache Spark để xử lý dữ liệu đã được lưu trữ trong Hadoop. Spark thực hiện các nhiệm vụ như làm sạch dữ liệu, thực hiện truy vấn và biểu diễn dữ liệu một cách hiệu quả. Dữ liệu sau khi được xử lý lại được lưu về Hadoop và Elasticsearch.
4. Bộ phận Biểu diễn Dữ liệu: Dữ liệu sau khi được xử lý bởi Spark được đưa vào Elasticsearch thông qua thư viện mã nguồn mở Elasticsearch for Apache Hadoop. Điều này giúp tạo ra một cơ sở dữ liệu có khả năng tìm kiếm mạnh mẽ và dễ truy cập cho việc trực quan hóa thông tin tuyển dụng.

Hệ thống này được thiết kế để tối ưu hóa quá trình thu thập và xử lý dữ liệu từ các nguồn thông tin trực tuyến, đồng thời đảm bảo khả năng mở rộng và ổn định trong việc lưu trữ và truy xuất thông tin.

## Chi tiết về thành phần hệ thống

* + 1. Docker

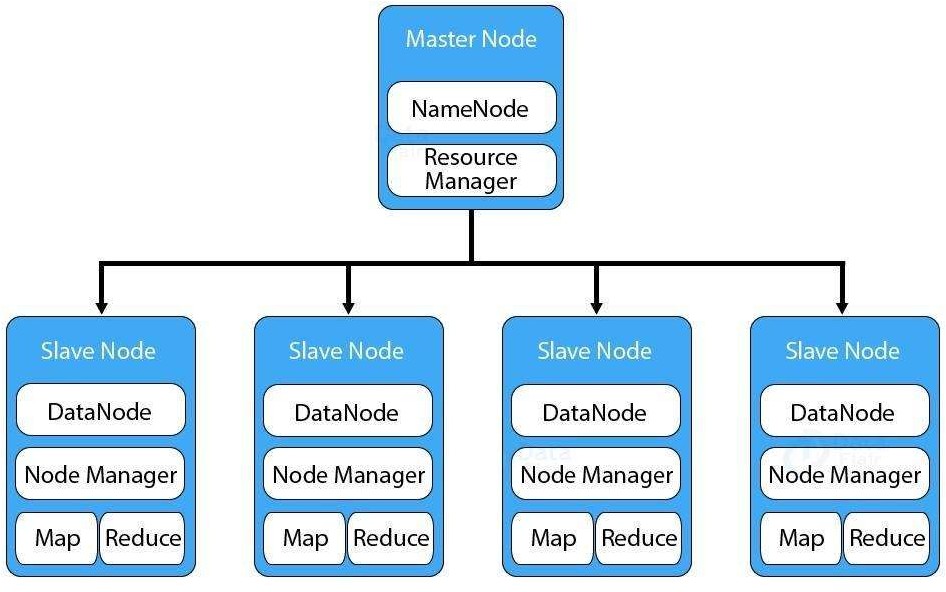
Docker, một nền tảng ảo hóa hóa mức độ hệ thống, cung cấp một giải pháp hiệu quả cho việc triển khai ứng dụng và môi trường mà không cần lo lắng về sự không tương thích giữa hệ điều hành và các phụ thuộc. Docker giúp đơn giản hóa quá trình xây dựng, đóng gói, và chia sẻ ứng dụng trong các môi trường container hóa.

Lệnh docker-compose up là một công cụ quan trọng trong việc quản lý và triển khai ứng dụng với Docker. Nó cho phép người dùng định nghĩa và chạy đồng thời nhiều container từ một tập tin cấu hình (thường là docker- compose.yml). Các container có thể liên kết và tương tác với nhau, tạo ra môi trường phát triển hoặc triển khai đồng nhất.

Việc sử dụng Docker và docker-compose up có thể thay thế cho cơ chế SSH trong việc triển khai và quản lý các thành phần của Hadoop, giúp đơn giản hóa quá trình triển khai và giảm thiểu sự phức tạp trong việc tạo và quản lý các container, đặc biệt là khi làm việc trong môi trường phân tán.

* + 1. Hadoop Cluster

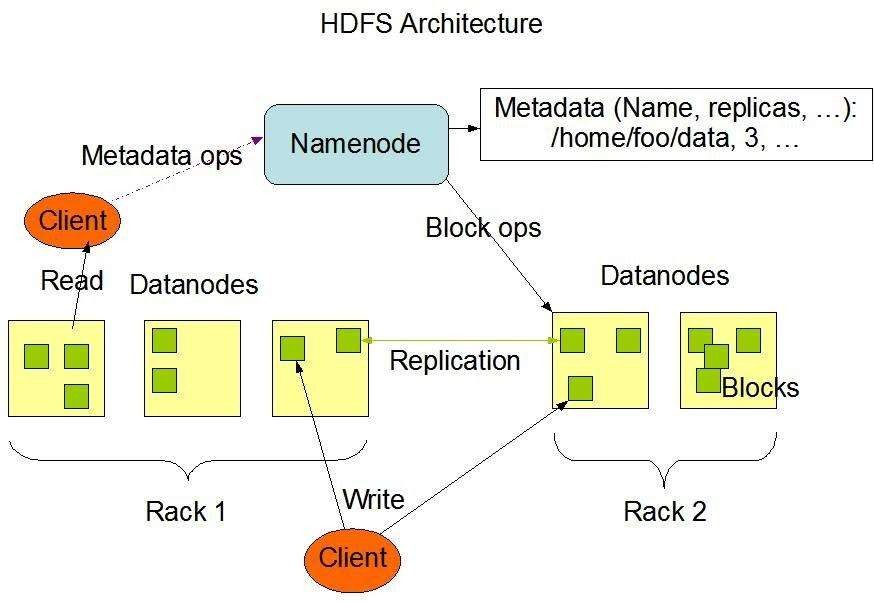
Hadoop Cluster là hệ thống file phân tán, cung cấp khả năng lưu trữ dữ liệu khổng lồ và tính năng tối ưu hoá việc sử dụng băng thông giữa các node.



Hadoop được cài đặt trên các máy tính trong hệ thống phân tán theo kiến trúc master – slave. Hadoop có thể hoạt động trên một máy (giống như 1 team chỉ có 1 member) hoặc mở rộng tới hàng ngàn máy, với mỗi máy đều có thể sử dụng để lưu trữ hoặc tính toán dữ liệu. Khi lưu trữ trên Hadoop, file dữ liệuđược chia thành các chunk và được lưu thành nhiều bản sao, giúp cho cụm Hadoop có khả năng chịu lỗi.

HDFS là nơi lưu dữ liệu của Hadoop, HDFS chia chia nhỏ dữ liệu thành các đơn vị dữ liệu nhỏ hơn gọi là các blocks và lưu trữ chúng phân tán trong các node của cụm Hadoop. HDFS sử dụng kiến trúc master/slave, trong

đó master gồm một Name Node để quản lý hệ thống file metadata v và một hay nhiều slave Data Nodes để lưu trữ dữ liệu.



Đối với hệ thống phân tích thông tin tuyển dụng dữ liệu thu thập được trên Recruitment Platform sẽ được lưu trên cụm Hadoop. Cụm Hadoop của RecruitmentAnalys bao gồm một Namenode/SecondaryNamenode và 2 Datanode. Khi lượng dữ liệu tăng lên, kiến trúc này có thể mở rộng thêm bằng cách bổ sung các Datanode để tăng cường dung lượng lưu trữ của hệ thống.

* + 1. Spark Cluster

Apache Spark là một framework xử lý dữ liệu mã nguồn mở trên quy mô lớn. Spark cung cấp một giao diện để lập trình các cụm tính toán song song với khả năng chịu lỗi.

Tốc độ xử lý của Spark có được do việc tính toán được thực hiện cùng lúc trên nhiều máy khác nhau. Đồng thời việc tính toán được thực hiện hoàn toàn trên RAM.

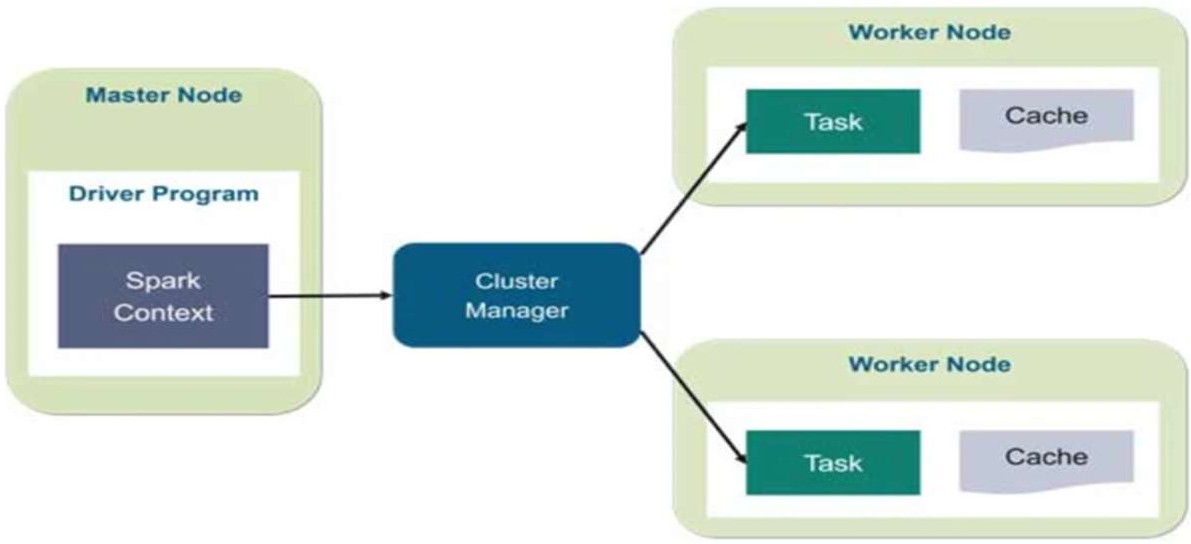
Spark cho phép xử lý dữ liệu theo thời gian thực, vừa nhận dữ liệu từ các nguồn khác nhau đồng thời thực hiện ngay việc xử lý trên dữ liệu vừa nhận được.

Những điểm nổi bật của Spark:

* Xử lý dữ liệu: Spark xử lý dữ liệu theo lô và theo thời gian thực.
* Tính tương thích: Có thể tích hợp với tất cả nguồn dữ liệu và định dạng tệp được hỗ trợ bởi cụm Hadoop.
* Hỗ trợ ngôn ngữ: Java, Python, Scala, R.
* Phân tích thời gian thực.

Kiến trúc của Spark bao gồm hai thành phần chính: trình điều khiển (driver) và trình thực thi (executors). Trình điều khiển dùng để chuyển đổi mã của người dùng thành nhiều tác vụ (tasks) có thể được phân phối trên các nút xử lý (worker nodes). Khi thực thi, trình điều khiển Driver tạo ra 1 SparkContext, sau đó giao tiếp với Cluster Manager để tính toán tài nguyên và phân chia các tác vụ đến cho các worker nodes.

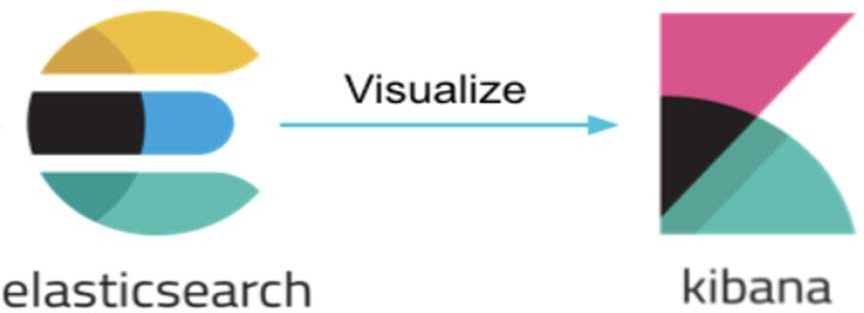
Apache Spark xây dựng các lệnh xử lý dữ liệu của người dùng thành Đồ thị vòng có hướng hoặc DAG. DAG là lớp lập lịch của Apache Spark; nó xác định những tác vụ nào được thực thi trên những nút nào và theo trình tự nào.



* + 1. ElasticSearch và Kibana

Dữ liệu sau khi được làm sạch bởi Spark cần được biểu diễn dưới dạng bảng biểu, đồ thị để mang đến cho người dùng góc nhìn trực quan nhất.

Elasticsearch và Kibana là những ứng dụng phù hợp để đảm nhận vai trò này. Là một công cụ tìm kiếm (với tốc độ gần thời gian thực) và phân tích dữ liệu phân tán, Elasticsearch có thể lưu trữ và phân tích nhiều loại dữ liệu khác nhau như: giữ liệu có cấu trúc, giữ liệu phi cấu trúc, giữ liệu số, dữ liệu về không gian địa lý, đánh chỉ mục dữ liệu một cách hiệu quả nhằm hỗ trợ quá trình tìm kiếm được thực hiện nhanh chóng. Các truy vấn trên Elasticsearch được thực hiện thông qua API, curl, python, hoặc qua Kibana. Kibana cung cấp giao diện đồ hoạ để người dùng dễ dàng hơn trong việc khai phá, biểu diễn trực quan dữ liệu được lưu trên Elasticsearch.



# CHƯƠNG 2: XÂY DỰNG CHƯƠNG TRÌNH VÀ HỆ THỐNG

## Luồng dữ liệu của hệ thống

**Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Phông chữ

Mô tả được tạo tự động**

Luồng dữ liệu của hệ thống chúng em xây dựng gồm 4 quá trình:

* + 1. Thu thập dữ liệu trên website phim
    2. Lưu dữ liệu vào Hadoop.
    3. Lọc, làm sạch dữ liệu trên Hadoop bằng Spark. Sau đó lưu thành 2 bản: 1 bản lưu trả về Hadoop, 1 bản gửi lưu vào Elasticsearch.
    4. Biểu diễn dữ liệu trên Elasticsearch dưới dạng biểu đồ, đồ thị, danh sách bảng sử dụng Kibana.

## Khởi động hệ thống HDFS

*docker-compose up*

*Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, số

Mô tả được tạo tự động*

Khởi động spark master: master.sh

Ảnh có chứa văn bản, ảnh chụp màn hình, số, hàng

Mô tả được tạo tự động

Khởi động Elasticsearch:

Ảnh có chứa văn bản, Phông chữ, số, ảnh chụp màn hình

Mô tả được tạo tự động

## Các trải nghiệm khi xây dựng chương trình và hệ thống

* + 1. Thu thập dữ liệu

Dữ liệu của hệ thống là dữ liệu về thông tin các bộ phim có thể được thu thập tại website MotPhim. Hệ thống sẽ truy cập vào từng link và thu thập thông tin theo các thẻ. Mỗi bộ phim sẽ được lưu thành một đối tượng json (một bản ghi), trong đó tên của các thẻ trong html và nội dung của các thẻ tương ứng sẽ tạo thành các cặp key-value.

Mỗi bản ghi có các trường sau:

* name: Tên bộ phim
* link: đường dẫn đến bộ phim
* type: loại phim
* năm phát hành
* Trạng thái
* Số tập
* Tình trạng
* Thể loại
* Đại diễn
* Diễn viên
* Đánh giá

Chương trình thu thập dữ liệu của hệ thống được lưu ở file **crawler.py**, sử dụng thư viện BeautifulSoup. BeautifulSoup là một thư viện Python dùng để lấy dữ liệu ra khỏi các file HTML và XML. Nó hoạt động cùng với các parser (trình phân

tích cú pháp) cung cấp cho bạn các cách để điều hướng, tìm kiếm và chỉnh sửa trong parse tree (cây phân tích được tạo từ parser).

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

* + 1. Lưu dữ liệu vào Hadoop

Dữ liệu sau khi được thu thập sẽ được đẩy lên Hadoop và lưu vào HDFS:

Ảnh có chứa văn bản, ảnh chụp màn hình, số, phần mềm

Mô tả được tạo tự động

Dữ liệu được lưu trên datanode là node02

Ảnh có chứa văn bản, đồ điện tử, ảnh chụp màn hình, phần mềm

Mô tả được tạo tự động

* + 1. Lọc dữ liệu bằng Spark

Dữ liệu vừa được đẩy lên HDFS mới chỉ là dữ liệu thô, ta cần trích xuất, tiền xử lý để mang loại bỏ thông tin dư thừa giúp tối ưu khả năng lưu trữ cũng như mang lại những tri thức, những góc nhìn có ý nghĩa về dữ liệu đối với người dùng.

Định nghĩa 1 schema để đọc tại Spark khi Hadoop tạo 1 dataframe:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Một dataframe raw\_recruit\_df với schema đã được định nghĩa như trên được tạo ra từ dữ liệu lưu trong các file json đã được lưu trong Hadoop. Nhưng mà raw\_recruit\_df vẫn chỉ là 1 dataframe với dữ liệu thô. Từ raw\_recruit\_df, Spark sẽ trích xuất thông tin để tạo ra một dataframe với các trường dữ liệu bao gồm :

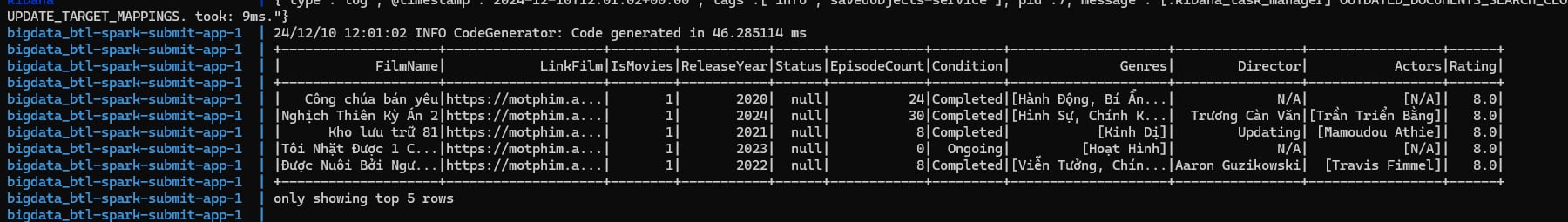
* FilmName: Tên bộ phim
* LinkFilm: Đường dẫn đến bộ phim
* IsMovies: Phân loại phim (phim bộ hay phim lẻ)
* ReleaseYear: Năm phát hành
* Status: Trạng thái
* EpisodeCount: Số tập
* Condition: tình trạng
* Genres: Thể loại
* Director: Đạo diễn
* Actors: Diễn viên
* Rating: Đánh giá
* Với các user define function được định nghĩa, một dataframe mới, extracted\_recruit\_df, được lọc từ raw\_recruit\_df

Tạo dataframe với dữ liệu được lọc từ dataframe ban đầu:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Các dòng đầu của dataframe lọc từ dataframe ban đầu:



Dữ liệu lúc này đã sẵn sàng để lưu về Hadoop và Elasticsearch

Để Spark và Elasticsearch tương tác với nhau cần sử dụng thư viện Elasticsearch for Apache Hadoop.

Spark-master sẽ tiến hành phân chia tác vụ và tài nguyên cho các spark- worker:

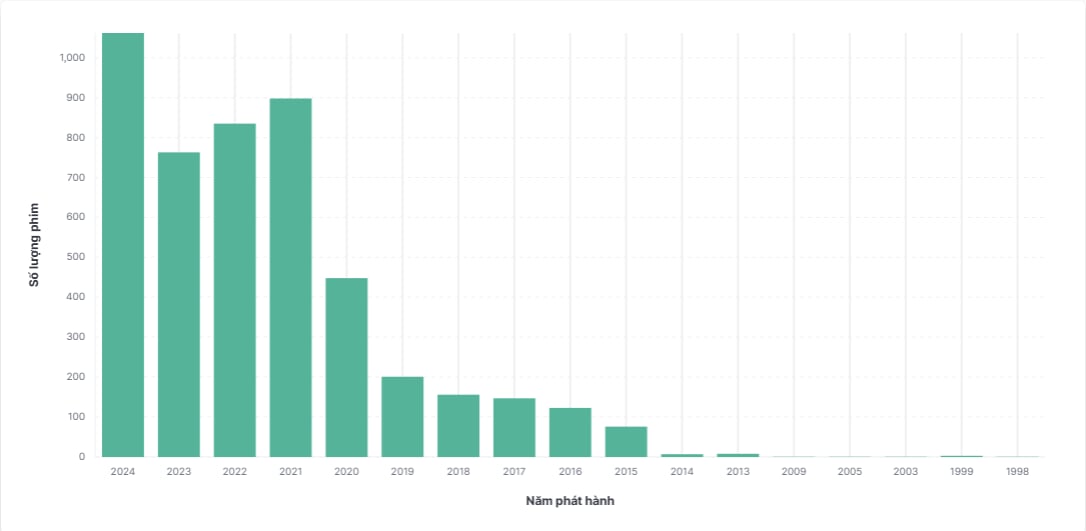
Ảnh có chứa văn bản, ảnh chụp màn hình, số, hàng

Mô tả được tạo tự động

* + 1. Biểu diễn dữ liệu bằng Kibana

Dữ liệu lưu tại Elasticsearch sẽ được dùng Kibana để biểu diễn Ví dụ:

* + - 1. Số lượng phim theo năm phát hành



* + - 1. Các thể loại phim

Ảnh có chứa đồ điện tử, văn bản, đĩa nén, vòng tròn

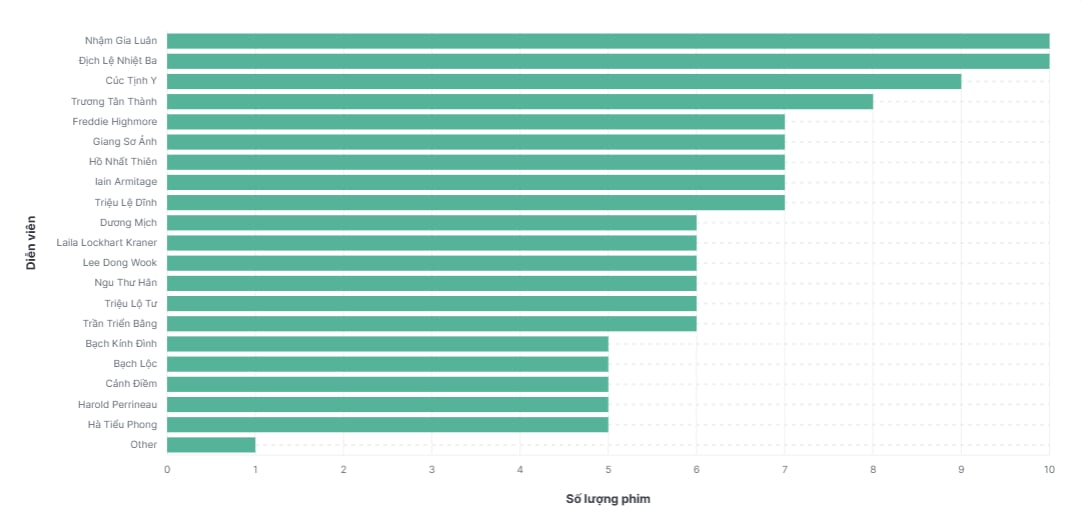
Mô tả được tạo tự động

* + - 1. Tỉ lệ phim bộ và phim lẻ

Ảnh có chứa vòng tròn, ảnh chụp màn hình, biểu đồ, thiết kế

Mô tả được tạo tự động

* + - 1. Số lượng bộ phim của các diễn viên đóng



* + - 1. Số lượng bộ phim theo đánh giá trung bình từ 1 – 10

Ảnh có chứa văn bản, Sơ đồ, hàng, ảnh chụp màn hình

Mô tả được tạo tự động

## Kinh nghiệm

* + 1. **BeautifulSoup**
* **Tính năng**:
  + Dễ dàng trích xuất dữ liệu từ HTML hoặc XML (web scraping).
  + Hỗ trợ xử lý DOM, tìm kiếm và lọc các phần tử cụ thể.
* **Vấn đề gặp phải**:
  + Chặn bot: Một số trang web triển khai CAPTCHA hoặc giới hạn tần suất truy cập (rate-limiting).
  + Tương thích: BeautifulSoup thiếu hiệu năng với khối lượng lớn dữ liệu so với các framework khác như Scrapy.

**2.4.2 HDFS (Hadoop Distributed File System)**

* **Tính năng**:
  + Lưu trữ dữ liệu phân tán trên nhiều node.
  + Tự động sao chép dữ liệu để đảm bảo tính toàn vẹn.
  + Thử nghiệm xóa DataNode container: hệ thống vẫn hoạt động, nhưng thời gian truy xuất tăng do dữ liệu cần khôi phục từ bản sao.
* **Vấn đề gặp phải**:
  + **Kết nối giữa NameNode và DataNode**:
    - Docker yêu cầu cấu hình chính xác các địa chỉ trong core-site.xml và hdfs-site.xml.
    - Khó khăn trong việc kiểm tra giao tiếp khi không sử dụng Docker logs.
  + **Lưu trữ dữ liệu**:
    - **Thiết lập replication nhỏ sẽ dẫn đến việc rơi vào safe mode và không lưu được dữ liệu khi chạy spark**
    - Không thiết lập persistent volume ban đầu, dẫn đến mất dữ liệu khi container tắt.

**2.4.3 Apache Spark**

* **Tính năng**:
  + Chạy các tác vụ xử lý dữ liệu lớn trên các worker node.
  + Cơ chế **checkpointing** và **recovery** để xử lý lỗi trong pipeline.
  + Checkpointing lưu trạng thái dữ liệu trong pipeline xử lý (ví dụ: trạng thái RDD).
  + Thử nghiệm tắt Spark Worker giữa chừng: Sau khi khởi động lại, Spark tiếp tục pipeline từ checkpoint gần nhất.
* **Vấn đề gặp phải**:
  + **Cấu hình Spark Cluster**:
    - Phải chỉ định SPARK\_MASTER\_HOST và Docker network để các worker kết nối với master.
  + **Hiệu năng**:
    - Giới hạn tài nguyên (RAM/CPU) của container ảnh hưởng đến tốc độ xử lý dữ liệu.

**2.4.4 Elasticsearch:**

* **Trải nghiệm**:
  + Hỗ trợ sharding và replication để đảm bảo dữ liệu có thể mở rộng và chịu lỗi.
* **Vấn đề**:
  + Khi số lượng index tăng quá lớn, hiệu suất tìm kiếm giảm do overhead của quản lý shard.
  + Docker network yêu cầu cấu hình cụ thể để Kibana nhận đúng địa chỉ Elasticsearch.

**2.4.5 Kibana**

* **Tính năng**:
  + Giao diện trực quan để phân tích dữ liệu Elasticsearch.
  + Tích hợp công cụ xây dựng dashboard theo thời gian thực.
* **Vấn đề gặp phải**:
  + Biểu đồ phức tạp: Các thao tác query lớn đôi khi làm giảm hiệu năng dashboard.
  + Định dạng dữ liệu: Yêu cầu dữ liệu được index trong Elasticsearch cần đúng cấu trúc để hiển thị.
  + Dữ liệu lớn từ Spark đẩy vào Elasticsearch gây chậm phản hồi trên Kibana.

# CHƯƠNG 3: NHẬN XÉT, ĐÁNH GIÁ VÀ HƯỚNG PHÁT TRIỂN

## Nhận xét, đánh giá

Hệ thống của nhóm đã cho thấy nhiều lợi ích quan trọng của một hệ thống Big Data, bao gồm khả năng lưu trữ, tìm kiếm, và biểu diễn lượng lớn dữ liệu, cũng như khả năng mở rộng linh hoạt khi lượng tài nguyên hiện tại không đủ, và khả năng chịu lỗi trong môi trường mạng phân tán khi một số thành phần gặp sự cố. Những đặc tính này làm cho hệ thống Big Data trở nên mạnh mẽ hơn so với các hệ thống truyền thống, đặc biệt là khi đối mặt với quy mô và đa dạng của dữ liệu ngày nay.

Tuy nhiên, hệ thống cũng gặp một số nhược điểm. Việc sử dụng Spark không được tận dụng hết tiềm năng của hệ thống, có thể làm giảm hiệu suất toàn bộ quá trình. Lượng dữ liệu thu thập cũng khá ít, có thể xử lý trên một máy tính đơn. Ngoài ra, quy trình thực hiện của hệ thống vẫn có một số bước phải thực hiện bằng cách nhập mã thủ công, chưa được tự động hóa hoàn toàn. Điều này có thể tạo ra sự rời rạc và đôi khi làm giảm hiệu suất và hiệu quả của hệ thống. Để tối ưu hóa và cải thiện hệ thống, việc tự động hóa quy trình và tối ưu hóa việc sử dụng Spark là những điểm mà nhóm em cần cải thiện.

## Hướng phát triển

Do quá trình crawl dữ liệu được thực hiện trên một luồng nên tốc độ có thể được tăng tốc bằng lập trình đa luồng và kết hợp thêm với thư viện Selenium.

Sử dụng Spark Streaming để phân tích và cải thiện tốc độ ghi dữ liệu. Điều này giúp tối ưu hóa quá trình xử lý dữ liệu liên tục, mang lại khả năng phản ứng nhanh hơn đối với dữ liệu đang được sinh ra.

# DANH MỤC TÀI LIỆU THAM KHẢO

* + 1. https://demanejar.github.io/posts/mode-in-spark/
    2. Bài giảng “Lưu trữ và xử lý dữ liệu lớn” – TS. Trần Việt Trung
    3. https://[www.youtube.com/watch?v=dLTI2HN9Ejg](http://www.youtube.com/watch?v=dLTI2HN9Ejg)
    4. https://hadoop.apache.org/docs/r1.2.1/hdfs\_design.html
    5. https://viblo.asia/p/tim-hieu-ve-hadoop-bJzKmOBXl9N
    6. https://viblo.asia/p/tim-hieu-ve-apache-spark-ByEZkQQW5Q0
    7. https://[www.youtube.com/watch?v=mafw2-CVYnA](http://www.youtube.com/watch?v=mafw2-CVYnA)
    8. https://[www.youtube.com/watch?v=hRtInGQhBxs&list=PLJlKGwy-](http://www.youtube.com/watch?v=hRtInGQhBxs&list=PLJlKGwy-) 7Ac6ASmzZPjonzYsV4vPELf0x
    9. https://xuanthulab.net/gioi-thieu-va-cai-dat-elasticsearch-va-kibana-bang- docker.html
    10. Giáo trình “Tổng quan về dữ liệu lớn (Big Data)” – Ks. Nguyễn Công Hoan – Trung Tâm Thông Tin Khoa học thống kê (Viện KHTK)