**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

A red and yellow sign

Description automatically generated with low confidence

**BÁO CÁO ĐỒ ÁN MÔN HỌC**

**Đề tài:** **Lưu trữ, xử lí, phân tích dữ liệu phim điện ảnh**

**Giảng viên:** TS. Trần Việt Trung

**Nhóm 15**

|  |  |
| --- | --- |
| **Họ và tên** | **MSSV** |
| Đỗ Trọng Đạt | 20190164 |
| Đỗ Ngọc Tuân | 20194398 |
| Trần Văn Kiên | 20194311 |
| Nguyễn Thành Bắc | 20194229 |

*Hà Nội, 01/2023*

Mục lục

[I. Giới thiệu 3](#_Toc126860617)

[II. Hệ thống 5](#_Toc126860618)

[2.1 Tổng quan hệ thống 5](#_Toc126860619)

[2.2. Chi tiết hệ thống 5](#_Toc126860620)

[2.2.1 Hadoop Cluster 6](#_Toc126860621)

[2.2.2 Spark 7](#_Toc126860622)

[2.2.3 MongoDB 9](#_Toc126860623)

[III. Thực nghiệm 10](#_Toc126860624)

[3.1 Luồng dữ liệu của hệ thống 10](#_Toc126860625)

[3.2 Quy trình 10](#_Toc126860626)

[3.2.1 Thu thập dữ liệu 10](#_Toc126860627)

[3.2.2 Khởi chạy HDFS 12](#_Toc126860628)

[3.2.3 Cài đặt và triển khai spark 13](#_Toc126860629)

[3.2.4 Lưu trữ dữ liệu vào MongoDB 16](#_Toc126860630)

[3.2.5 Biểu diễn dữ liệu trên MongoDB Charts 17](#_Toc126860631)

[3.2.6 Kết quả thu được 19](#_Toc126860632)

[3.3 Đánh giá khả năng chịu lỗi và mở rộng của Hadoop 23](#_Toc126860633)

[IV. Đánh giá và kết luận 24](#_Toc126860634)

1. **Giới thiệu**

Hiện nay, các nguồn dữ liệu trên internet ngày càng lớn, và đa số là dữ liệu thô. Dữ liệu ở dạng thô không đem lại nhiều giá trị hữu ích đối với tổ chức/doanh nghiệp hay các cá nhân. Xử lý dữ liệu là phương pháp thu thập dữ liệu thô và chuyển nó thành thông tin có thể sử dụng được. Sau khi thu thập, dữ liệu lần lượt trải qua các bước lọc, sắp xếp, xử lý, phân tích, lưu trữ và sau đó được trình bày ở định dạng có thể đọc được. Trước tiên, ta cần lưu trữ toàn bộ dữ liệu cần khai thác vào cùng một hệ thống. Việc mở rộng hệ thống lưu trữ bằng cách tăng phần cứng lại không đảm bảo được các yêu cầu. Khi đó, chúng ta có thể sử dụng các hệ thống lưu trữ phân tán, ta sẽ đưa các trung tâm lưu trữ mới vào hệ thống để mở rộng không gian lưu trữ dữ liệu. Đối với các nguồn dữ liệu nhỏ, có thể sử dụng một máy tính với tài nguyên giới hạn cùng các công cụ hỗ trợ để phân tích, cũng như khi lưu trữ nguồn dữ liệu lớn, phân tích dữ liệu lớn cũng đòi hỏi nhiều tài nguyên hơn mà một máy tính không thể đáp ứng.

Thông tin dữ liệu phim ảnh là một trong những thông tin liên quan đến lĩnh vực giải trí được nhiều người quan tâm. Việc khai thác được thông tin đánh giá phim ảnh có thể giúp cho mọi người tìm được cho mình những thể loại phim hay những bộ phim phù hợp với sở thích của bản thân, các công ty làm phim hoặc các rạp chiếu phim có thể cân nhắc điều chỉnh theo nhu cầu và sở thích của người dùng. Một giải pháp đơn giản mà hiệu quả là thực hiện đánh giá,thống kê những đánh giá, bình luận được người dùng để lại qua các phim đã công chiếu. Các công đoạn khi thực hiện giải pháp này cơ bản sẽ bao gồm thu thập dữ liệu, lọc dữ liệu và biểu diễn, thống kê dữ liệu.

Trong phạm vi của project này, nhóm thực hiện tạo một hệ thống thu thập dữ liệu phim ảnh, sau đó vận dụng các kiến thức về lưu trữ và dữ liệu lớn để khai thác. Các công nghệ mà nhóm sử dụng trong hệ thống phân tích thông tin tuyển dụng này bao gồm Hadoop File System, Spark, MongoDB và MongoDB Charts.

1. **Hệ thống**
   1. **. Tổng quan hệ thống**

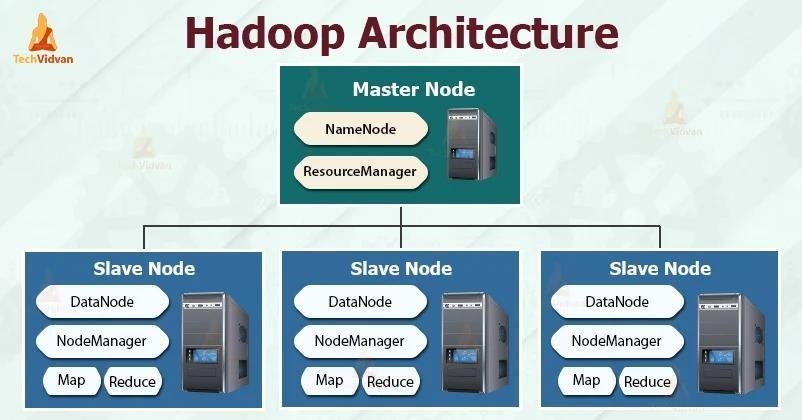
Hệ thống được xây dựng gồm 3 phần với các chức năng nhằm thu thập, xử lý, lưu trữ và trực quan hoá dữ liệu phim ảnh. Các thành phần của hệ thống bao gồm:

* + 1. Bộ phận lưu trữ: hệ thống lưu trữ dữ liệu vào Hadoop dưới dạng HDFS File System (HDFS) để có thể lưu dữ liệu phân tán và có chức năng mở rộng, sao lưu, đảm bảo truy cập được khi một số máy mất kết nối.
    2. Bộ phận xử lý dữ liệu: từ dữ liệu đã được lưu trong Hadoop, Spark được sử dụng để xử lý, làm sạch dữ liệu và thực hiện các truy vấn, giúp cho việc biểu diễn dữ liệu đơn giản hơn. Dữ liệu sau khi được làm sạch được lại được lưu về Hadoop và MongoDB.
    3. Bộ phận biểu diễn dữ liệu: dữ liệu sau khi được xử lý bởi Spark

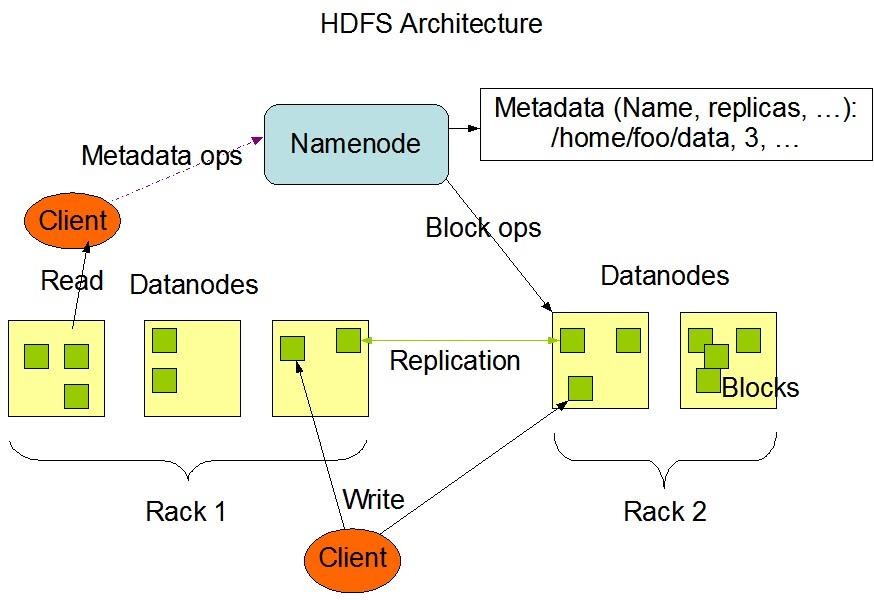
được đưa vào MongoDB và biểu diễn trên MongoDB Charts

* 1. **Chi tiết hệ thống**

* + 1. **Hadoop Cluster**

Một hệ thống dữ liệu lớn cần có một cụm các máy để lưu trữ dữ liệu phân tán. Cụm các máy này cần phải có khả năng chịu lỗi (ví dụ như một vài máy bị hỏng, mất kết nối, bị tắt hoặc cần bảo dưỡng), có khả năng mở rộng và thực hiện các tác vụ cơ bản lên dữ liệu.

Hadoop Cluster ra đời để giải quyết nhu cầu đó. Một cụm Hadoop bao gồm nhiều các máy tính khác nhau, tuân theo kiến trúc Master-Slave, cho phép lưu trữ và phân tích hàng petabytes dữ liệu. Khi lưu trữ trên Hadoop, file dữ liệu được chia thành các chunk và được lưu thành nhiều bản sao, giúp cho cụm Hadoop có khả năng chịu lỗi.



Kiến trúc Master-Slave của Hadoop

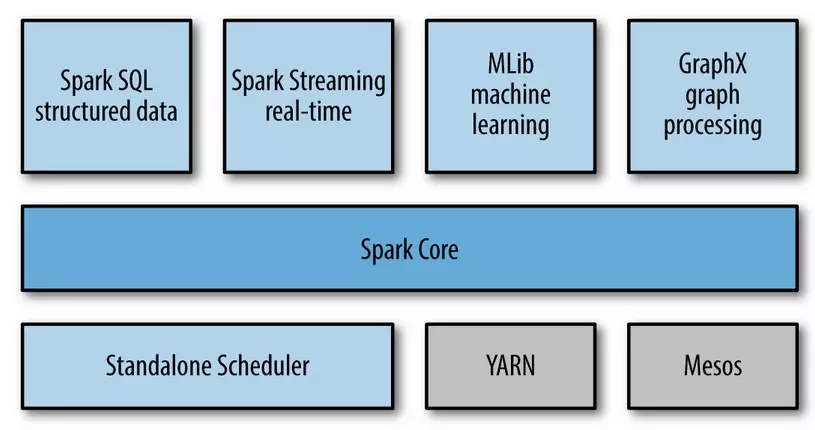
Đối với hệ thống, dữ liệu thu thập được sẽ được lưu trên cụm Hadoop.

Cụm Hadoop bao gồm:

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**

Mô hình cấu trúc Spark chi tiết



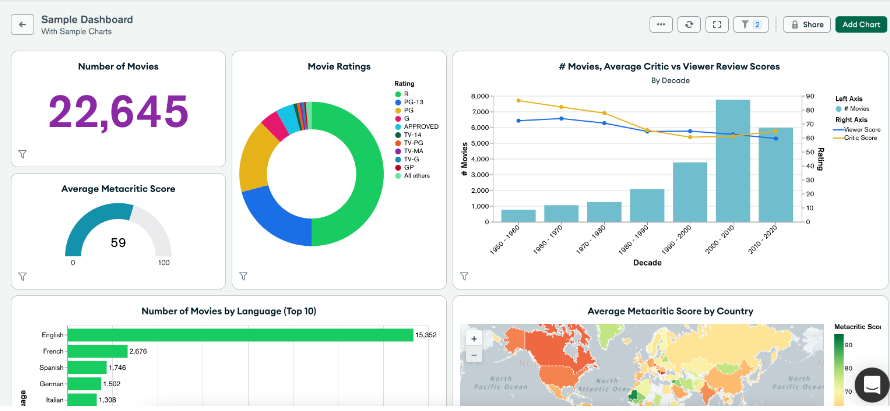
Spark có thể chạy trên nhiều loại Cluster Managers nhưu Hadoop YARN, Apache Mesos hoặc trên chính cluster manager được cung cấp bới Spark được gọi là Standalone Scheduler

* **Spark Core**: Thành phần cốt lõi, các thành phần muốn hoạt động đều cần thông qua Spark Core. Cung cấp những chức năng cơ bản như lập lịch cho các tác vụ, quản lý bộ nhớ, fault recovery, tương tác với các hệ thông lưu trữ… Nó còn cung cấp API để định nghĩa RDD
* **Spark SQL:** Nó tập trung vào việc xử lý dữ liệu có cấu trúc, sử dụng phương pháp tiếp cận khung dữ liệu được mượn từ các ngôn ngữ R và Python. Nó còn cung cấp một giao diện tiêu chuẩn để đọc và ghi vào các kho dữ liệu khác bao gồm JSON, HDFS, Apache Hive, JDBC, Apache ORC và Apache Parquet. Các cơ sở dữ liệu phổ biến khác như Apache Cassandra, MongoDB, Apache Hbase,.. cung được hỗ trợ thông qua trình kết nối riêng biệt từ hệ sinh thái Spark Packages
* **Spark Streaming:** Giúp đáp ứng các yêu cầu xử lý thời gian thực (realtime) hoặc gần như thời gian thực. Nó chia nhỏ luồng xử lý thành một chuỗi liên tục bao gồm các microbatch và sau đó có thể được thao tác bằng API Apache Spark
  + 1. **MongoDB**

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.

MongoDB và MongoDB Charts là những ứng dụng phù hợp để đảm nhận vai trò này. MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở thuộc noSQL - một dạng CSDL phi quan hệ. NoSQL được phát triển cũng trên Javascript Framework, lưu trữ dữ liệu kiểu JSON với dạng dữ liệu tổ chức theo key-value, mục đích hướng đến đối tượng. Điều này cho phép MongoDB linh hoạt trong việc insert dữ liệu dữ liệu, bởi đữ liệu không có sự ràng buộc lẫn nhau, không có join như trong RDBMS nên khi insert, xóa hay update nó không, cần phải mắt thời gian kiểm tra xem có thỏa mãn các ràng buộc dữ liệu như trong RDBMS.. Mô hình này giải quyết khá nhiều hạn chế tồn đọng của mô hình CSDL quan hệ RDBMS (Relation Database Managemrnt System) về tốc độ, tính mở rộng,…

MongoDB Charts là một công cụ để tạo các minh họa trực quan cho dữ liệu MongoDB của bạn. Trực quan hóa dữ liệu là một thành phần quan trọng để cung cấp sự hiểu biết rõ ràng về dữ liệu của bạn, làm nổi bật mối tương quan giữa các biến và giúp bạn dễ dàng phân biệt các mẫu và xu hướng trong bộ dữ liệu của mình. Biểu đồ MongoDB giúp cho việc truyền tải dữ liệu của bạn trở nên đơn giản hơn bằng cách cung cấp các công cụ có sẵn để dễ dàng chia sẻ và cộng tác trên các hình ảnh trực quan.

1. **Thực nghiệm**
   1. **Luồng dữ liệu của hệ thống**

Luồng dữ liệu của hệ thống gồm 4 giai đoạn. Giai đoạn thứ nhất là thu thập dữ liệu, sau đó được lưu vào Hadoop. Ứng dụng Spark sẽ làm sạch dữ liệu lưu trên Hadoop và lưu thành 2 bản, 1 bản lưu vào MongoDB để biểu diễn, bản còn lại lưu về Hadoop. Dữ liệu trên MongoDB được biểu diễn dưới dạng bảng biểu, đồ thị bằng MongoDB Charts:



* 1. **Quy trình**
     1. **Thu thập dữ liệu**

Dữ liệu của hệ thống là dữ liệu phim ảnh liên quan đến lĩnh vực phần mềm, có thể thu thập được trên Internet. Mỗi thông tin phim ảnh sẽ được lưu thành một đối tượng (một bản ghi).

**Bảng 3.1** Thiết kế chi tiết dữ liệu của collection movies

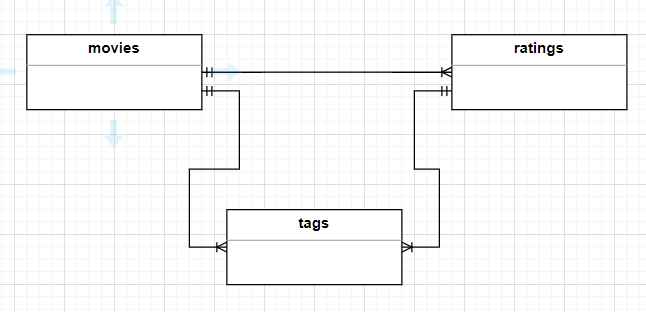
|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Giải thích** |
| movieId | String | Mã định danh của phim |
| title | String | Tiêu đề của phim |
| genres | String | Thể loại của phim |

**Bảng 3.2** Thiết kế chi tiết dữ liệu của collection ratings

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Giải thích** |
| movied | String | Mã định danh của phim |
| userId | String | Mã định danh của người dùng |
| tag | String | Tag mà người dùng gán |
| timestamp | Date | Thời điểm người dùng gắn tag |

**Bảng 3.3** Thiết kế chi tiết dữ liệu của collection ratings

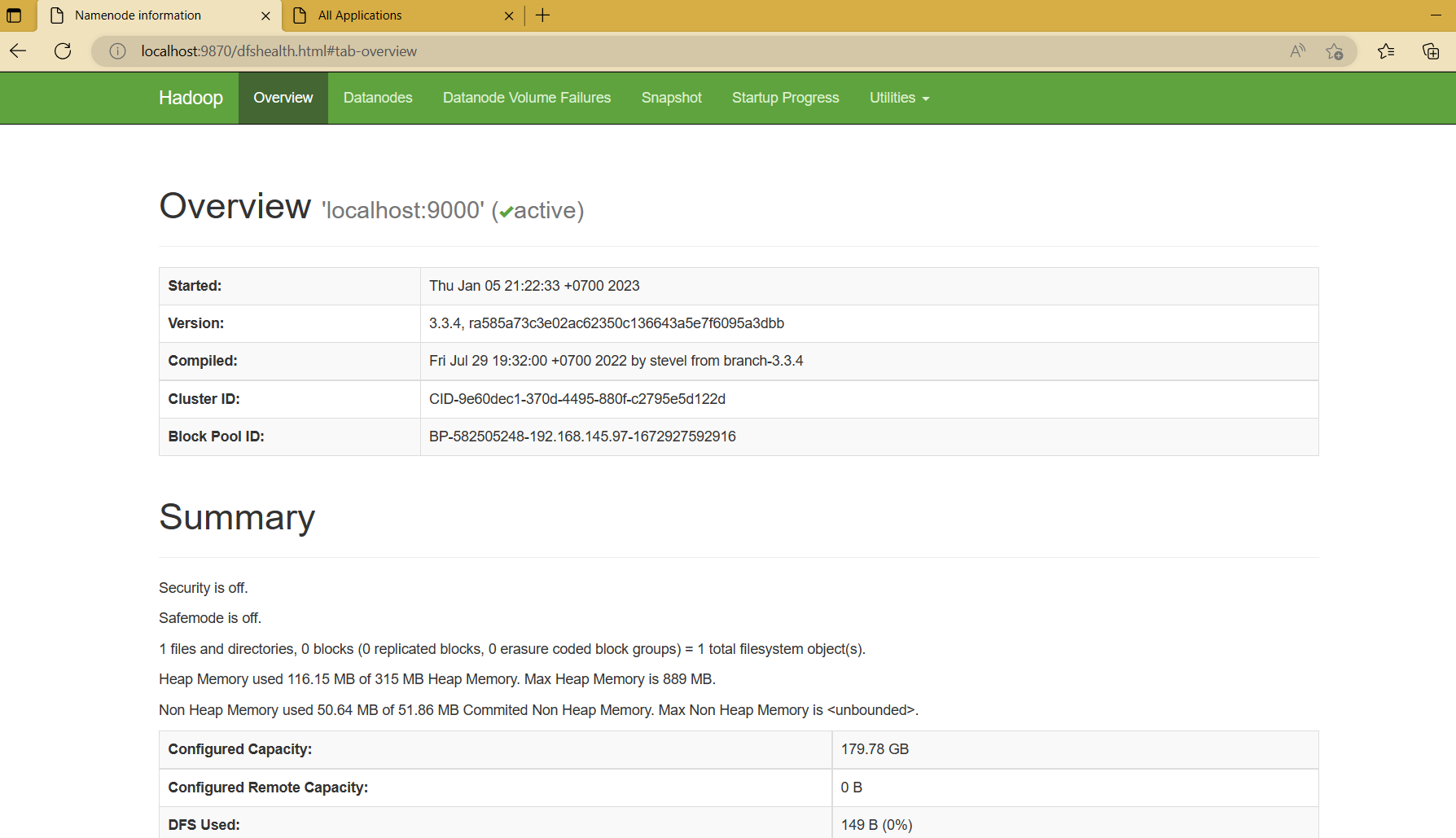
|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Giải thích** |
| movied | String | Mã định danh của phim |
| userId | String | Mã định danh của người dùng |
| rating | Number | Điểm đánh giá của người dùng về phim |
| timestamp | Date | Thời điểm đánh giá |



Cơ sở dữ liệu của hệ thống

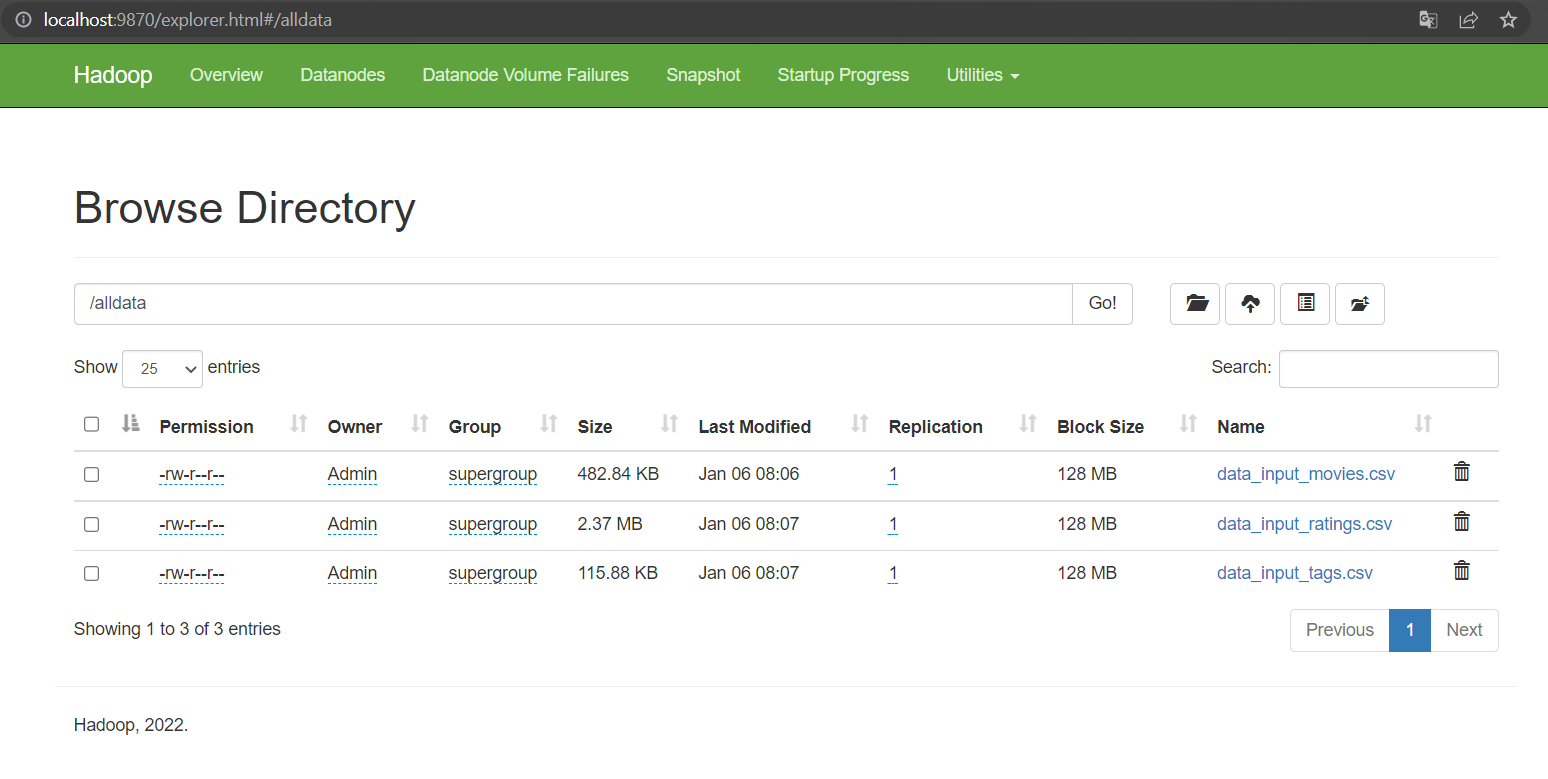
* + 1. **Khởi chạy HDFS**

*hdfs namenode -format start-dfs.sh start-yarn.sh*

Localhost 9870: 

**Đưa dữ liệu lên HDFS**

Dữ liệu sau đó được đẩy lên và lưu trên hdfs với lệnh hdfs dfs -put

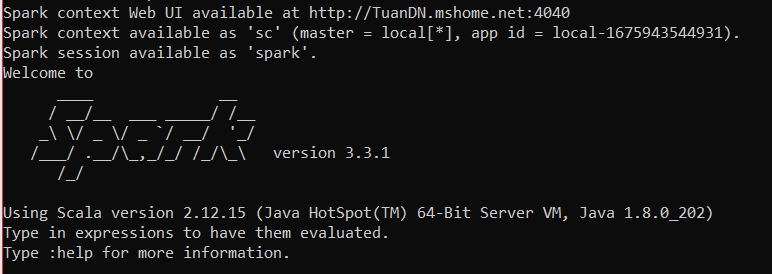


### **3.2.3 Cài đặt và triển khai spark**

**Cài đặt Spark**

* Trước khi cài đặt Apache Spark, chúng ta cần cài đặt Java . Sau đó thiết lập biến môi trường JAVA\_HOME
* Sau khi cài đặt các gói ngôn ngữ cần thiết, chúng ta tiến hành tải và lưu Apache Spark (tải tại [đây](https://spark.apache.org/downloads.html)) vào ổ **C:\** rồi thiết lập biến môi trường SPARK\_HOME và Path tương tự như khi cài đặt Java và Scala.
* Sau khi tải Spark, để sử dụng được trên Windows, chúng ta cần tải thêm Winutils (tại [đây](https://github.com/steveloughran/winutils)). Lựa chọn phiên bản Hadoop phù hợp với phiên bản Spark đã tải, download file winutils.exe lưu vào một thư mục bất kỳ rồi đặt biến môi trường HADOOP\_HOME dẫn tới thư mục đó và lưu một bản copy của Winutils.exe vào thư mục bin của Spark.
* Sau khi hoàn thành các bước trên, sẽ có thư mục tmp/hive được tạo ra trong ổ C. Ta cần phải thay đổi quyền truy cập cho thư mục này nếu không khi chạy Spark sẽ báo lỗi
* Để thực hiện việc này, ta mở một cửa sổ command rồi dùng Winutils để thay đổi permission như sau (chmod 777: cấp quyền read, write và execute):

**winutils.exe chmod 777 C:\tmp\hive**



* Sau khi cài đặt Apache Spark, ta tiếp tục cài đặt các gói thư viện Python: **PySpark** và **FindSpark** sử dụng **pip** (Trình quản lý gói của Python)

**pip install pyspark, pip install findspark**

**Triển khai Spark**

* Tạo Spark Session kết nối với MongoDB. Spark session: Đại diện cho khả năng tương tác với executors trong 1 chương trình. Spark session chính là entry point của mọi chương trình Spark. Từ SparkSession, có thể tạo RDD/ DataFrame/ DataSet, thực thi SQL… từ đó thực thi tính toán phân tán.

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

* Lấy dữ liệu từ HDFS đổ vào Spark. Spark thực hiện đọc dữ liệu file .csv từ HDFS được format thành DataFrame

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

**-** Khai báo cấu trúc của dữ liệu. Với một số trường hợp dữ liệu không được sạch, bao gồm có những bản ghi sai cấu trúc thì ta tiến hành khai báo cấu trúc để thực hiện loại bỏ những bản ghi đó

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

* Sử dụng Spark SQL để xử lý data bằng truy vấn

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

* + 1. **Lưu trữ dữ liệu vào MongoDB**

Từ SparkSession cấu hình thêm MongoDb URI để Spark có thể kết nối để đọc, ghi dữ liệu sang MongoDb



Ghi data từ Spark sang MongoDB với mode(“append”) để thêm mới các bản ghi vào MongoDB

Ảnh có chứa văn bản

Mô tả được tạo tự động

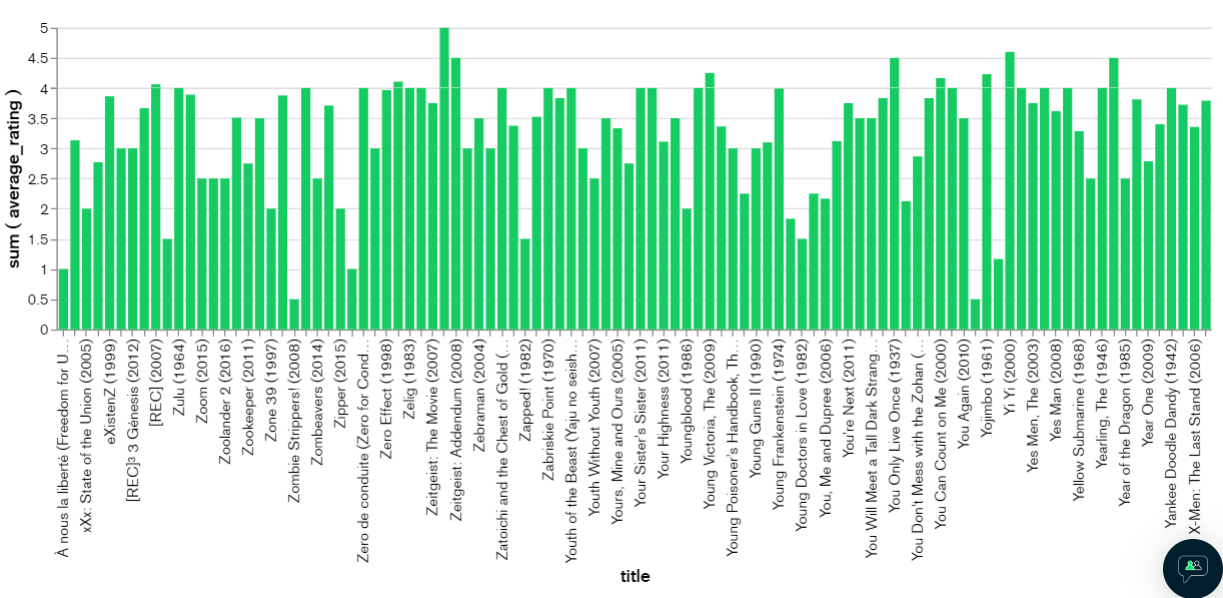
Sau khi ghi dữ liệu sang MongoDB thành công thì dữ liệu sẽ được hiển thị như sau:

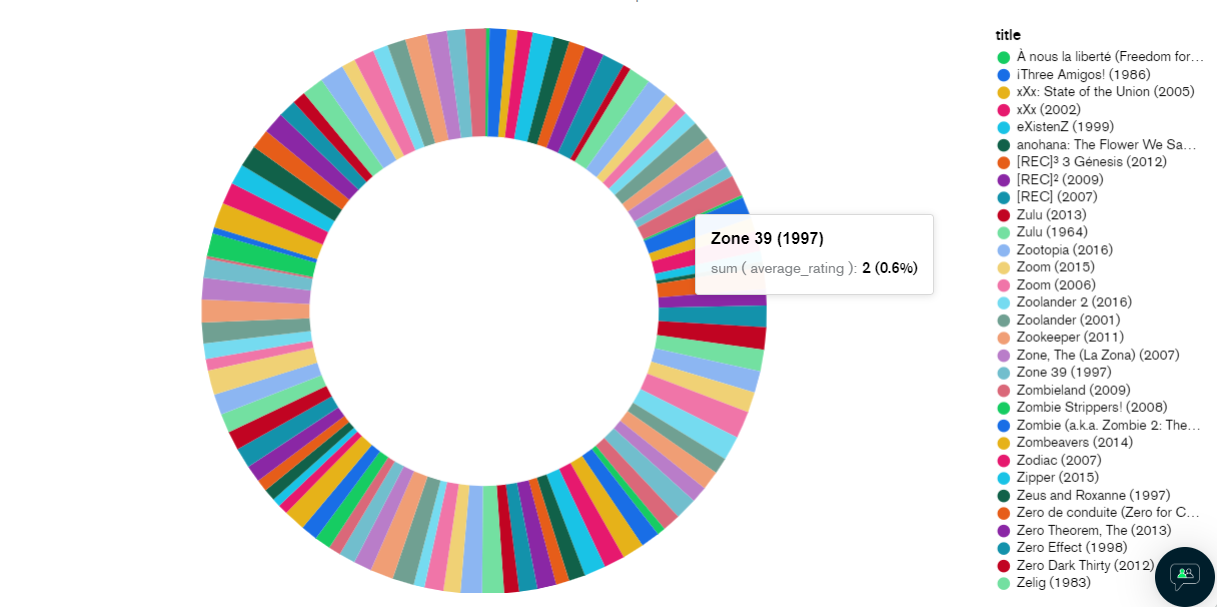
Ảnh có chứa văn bản

Mô tả được tạo tự động

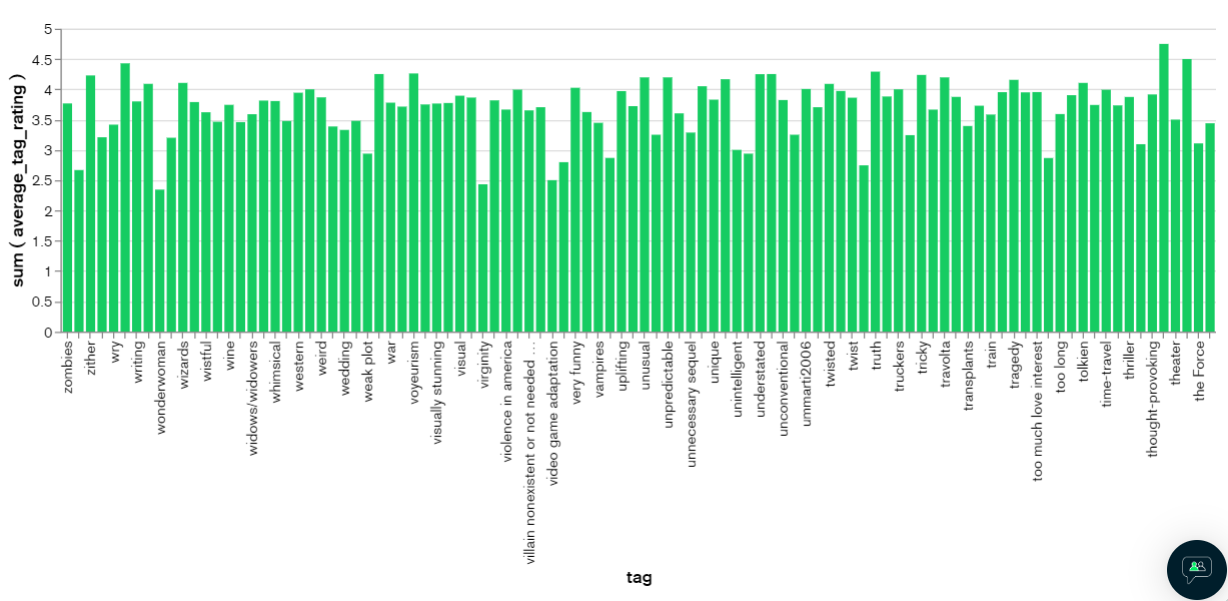
* + 1. **Biểu diễn dữ liệu trên MongoDB Charts**

Một số các thống kê về dữ liệu được cho trong các hình dưới đây.

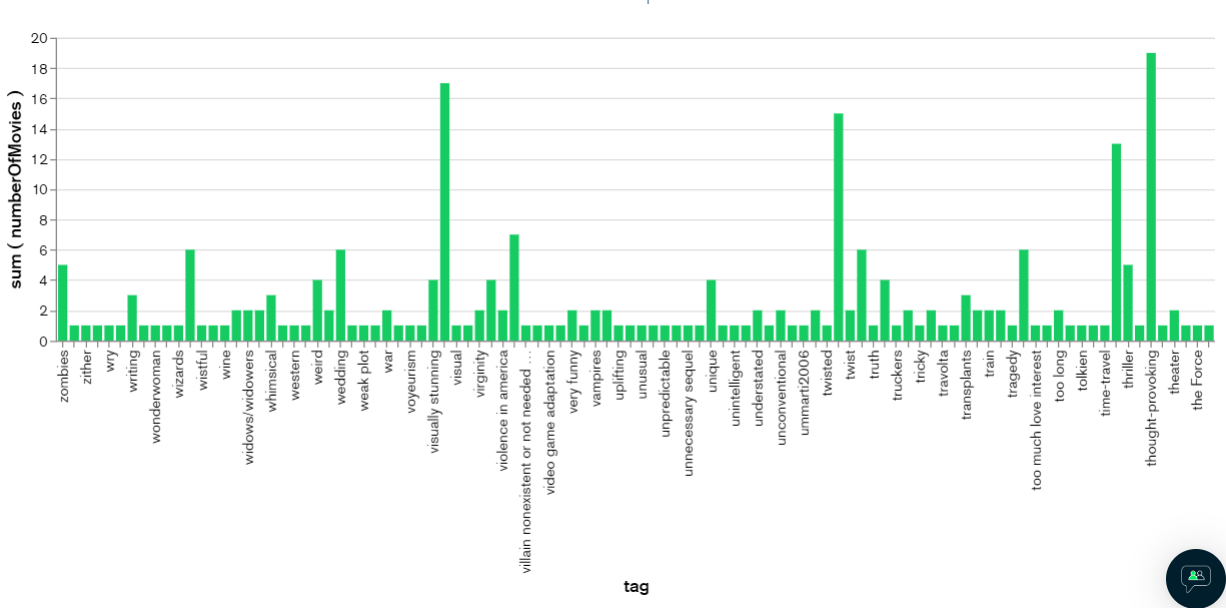




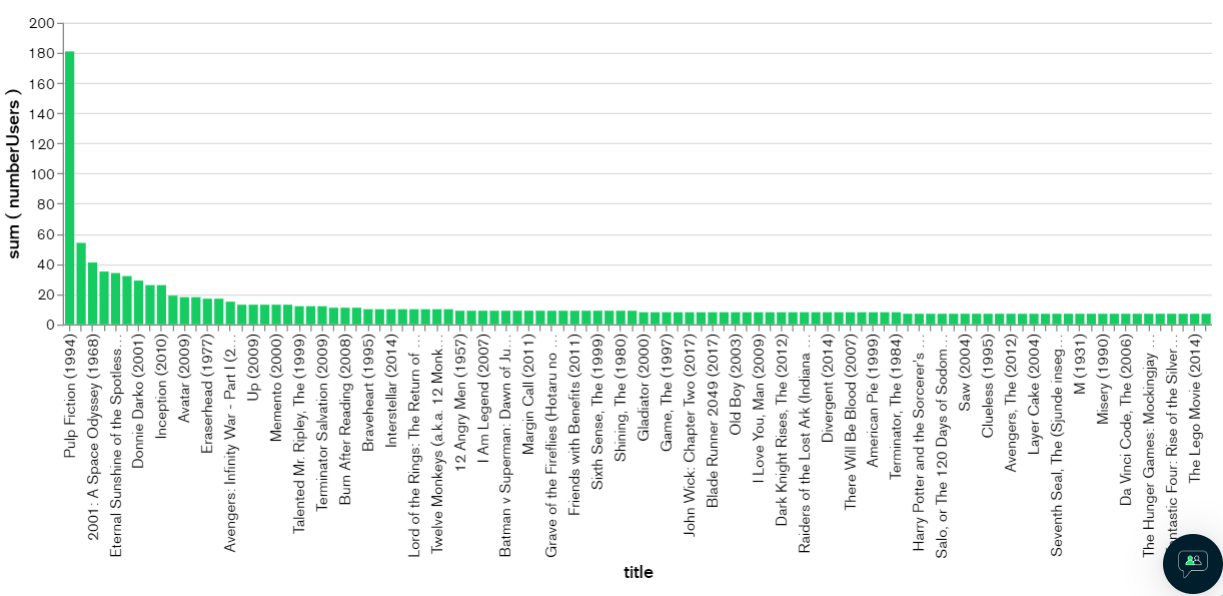
Thống phân bố rating trung bình của từng phim



Thống phân bố rating trung bình theo từng tag

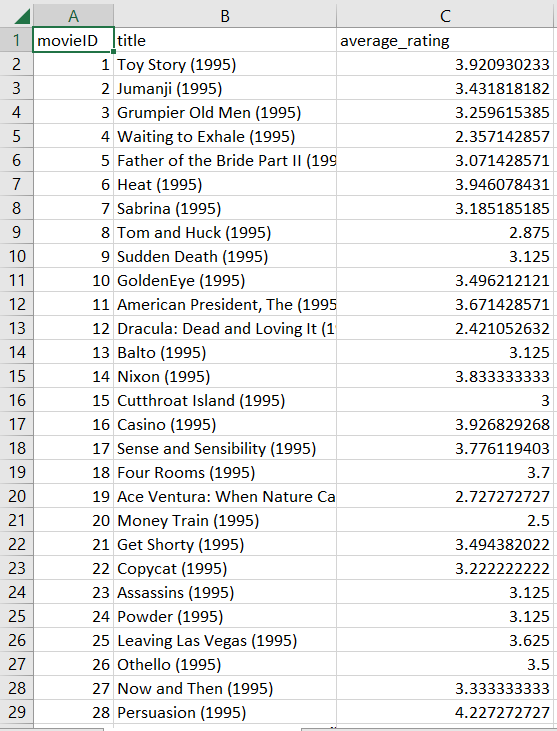


Thống kê phân bố số lượng phim theo từng tag



Thống kê phân bố số lượng người xem theo từng phim

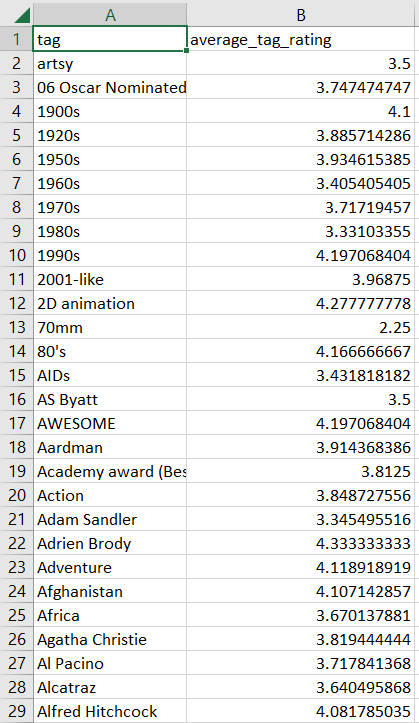
* + 1. **Kết quả thu được**

****

Rating trung bình mỗi phim



Số lượng phim theo mỗi tag



Rating trung bình từng tag



Số lượng người xem của từng phim

* 1. **Đánh giá khả năng chịu lỗi và mở rộng của Hadoop**

Khi nguồn dữ liệu cung cấp lớn và hệ thống không thể đủ không gian lưu trữ, ta có thể mở rộng hệ thống bằng cách thêm các datanode.

Khi triển khai hệ thống trong thực tế, việc một vài máy bị mất kết nối hoặc bị tắt nên không thể tham gia vào hệ thống là điều không thể tránh khỏi. Tại Hadoop Cluster, dữ liệu được lưu trữ phân tán và lưu thành các bản sao giúp cho hệ thống có khả năng chịu lỗi.

Có thể kiểm tra xem một datanode đã bị tắt chưa bằng cách xem thời gian liên lạc gần nhất giữa namenode và datanode. Theo cài đặt, cứ 3s datanode sẽ phải liên lạc vớinamenode lần, nếu thời gian liên lạc lớn hơn 3s, chứng tỏ đã có lỗi xảy ra tại datanode.

1. **Đánh giá và kết luận**

Hệ thống lưu trữ, xử lý và phân tích dữ liệu phim ảnh cho thấy những lợi ích mà một hệ thống Big Data đem lại như khả năng lưu trữ, tìm kiếm, biểu diễn lượng lớn dữ liệu, khả năng mở rộng khi lượng tài nguyên hiện tại không đủ, khả năng chịu lỗi trong một mạng phân tán khi có những thành phần trong mạng gặp trục trặc. Đây là những khả năng mà các hệ thống truyền thống không có hoặc khả năng đáp ứng còn hạn chế.

Bên cạnh đó, hệ thống của nhóm có một số nhược điểm. Việc sử dụng spark standalone của nhóm không khai thác được tối đa hệ thống. Lượng dữ liệu được thu thập khá ít, hoàn tàn có thể chạy trong 1 máy. Ngoài ra luồng thực hiện của hệ thống vẫn khá rời rạc, một số bước tải dữ liệu vẫn thực hiện bằng cách gõ code trên terminal mà chưa được tự động hóa.