TRĐẠI HỌC BÁCH KHOA HÀ NỘI

TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BÁO CÁO BÀI TẬP LỚN

Đề tài: Lưu trữ và xử lý, phân tích dữ liệu thông tin tuyển dụng việc làm

Lớp : 144943

Học phần : Lưu trữ và xử lý dữ liệu lớn

Mã học phần : IT4931

Giảng viên hướng dẫn : TS. Trần Việt Trung

Danh sách thành viên nhóm 6:

Họ và tên - MSSV	Nhiệm vụ
Ngô Quốc Hiếu 20204746	Xây dựng hệ thống Hadoop
Phạm Thị Thu Hằng 20204742	Phân tích, thiết kế hệ thống, xử lý Spark
Hoàng Đình Kiên 20204837	Thu thập dữ liệu, làm báo cáo
Đoàn Sỹ Nguyên 20200445	Xử lý Elastic Search, Kibana
Trịnh Hữu Tùng 20204805	Xử lý Elastic Search, Kibana

Hà Nội, tháng 12 năm 2023

MỤC LỤC

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.

Để tiếp cận lĩnh vực này, nhóm chúng tôi đã quyết định tập trung vào một loại dữ liệu đủ lớn để thực hiện phân tích và lưu trữ. Thông tin tuyển dụng là một trong những lĩnh vực được nhiều người quan tâm, đặc biệt là với những người đang tìm kiếm việc làm. Việc khai thác thông tin về nhu cầu tuyển dụng có thể hỗ trợ người lao động tìm được công việc phù hợp, cũng như giúp các công ty điều chỉnh chiến lược tuyển dụng của mình. Đồng thời, những người đang làm việc có thể đánh giá mức độ phù hợp của họ với thị trường lao động. Phương pháp đánh giá và thống kê kỹ năng và kiến thức từ các tin tuyển dụng trên mạng có thể cung cấp cái nhìn toàn diện về thị trường lao động. Các công đoạn chính của giải pháp này bao gồm thu thập, lọc và biểu diễn, thống kê dữ liệu.Trong phạm vi của Bài tập lớn này, nhóm chúng

em thực hiện tạo một hệ thống thu thập dữ liệu từ một trang web tuyển dụng, sau đó vận dụng các kiến thức về lưu trữ và dữ liệu lớn để khai thác. Nguồn dữ liệu nhóm lựa chọn để nghiên cứu là dữ liệu liên quan đến việc làm trong lĩnh vực phần mềm, thu thập từ trang web CareerLink.

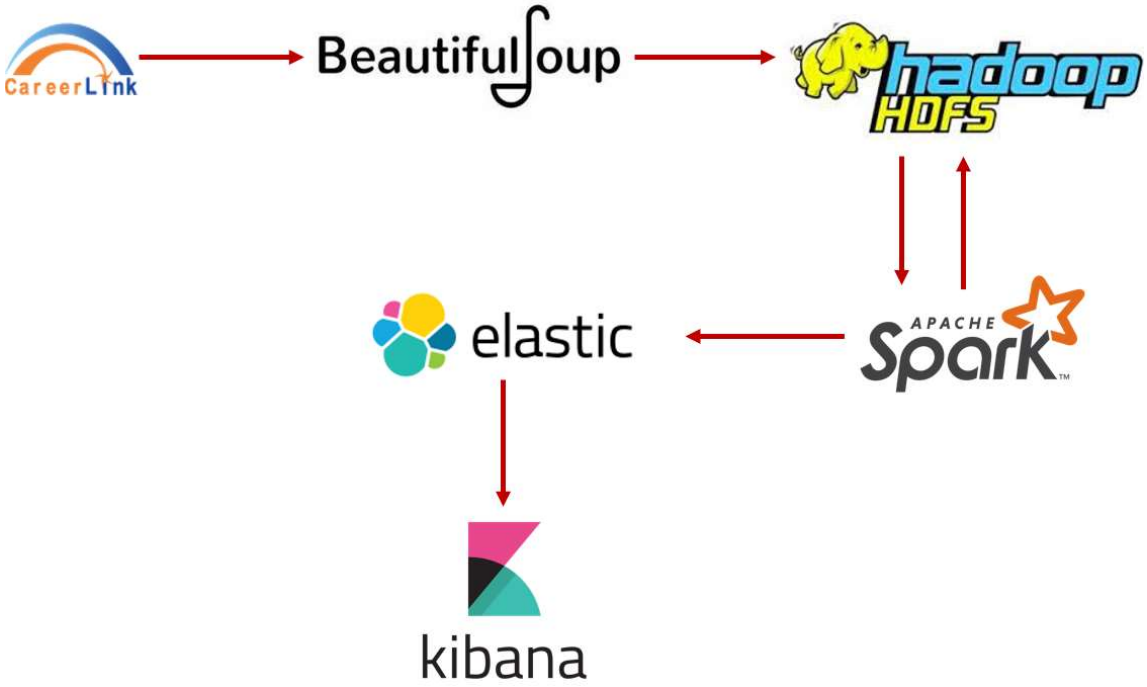
Bài tập lớn của nhóm chúng em bao gồm 3 nội dung chính:

- Tổng quan xây dựng hệ thống
- Xây dựng chương trình và hệ thống
- Nhận xét, đánh giá và hướng phát triển

Mặc dù đã cố gắng hoàn thiện sản phẩm nhưng không thể tránh khỏi những thiếu hụt về kiến thức và sai sót trong kiểm thử. Chúng em rất mong nhận được những nhận xét thẳng thắn, chi tiết đến từ thầy để tiếp tục hoàn thiện hơn nữa. Cuối cùng, nhóm chúng em xin được gửi lời cảm ơn đến thầy TS. Trần Việt Trung hướng dẫn chúng em trong suốt quá trình hoàn thiện Bài tập lớn. Nhóm chúng em xin chân thành cảm ơn thầy.

CHƯƠNG 1: TỔNG QUAN XÂY DỰNG HỆ THỐNG

1.1. Tổng quan hệ thống



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 phận 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.

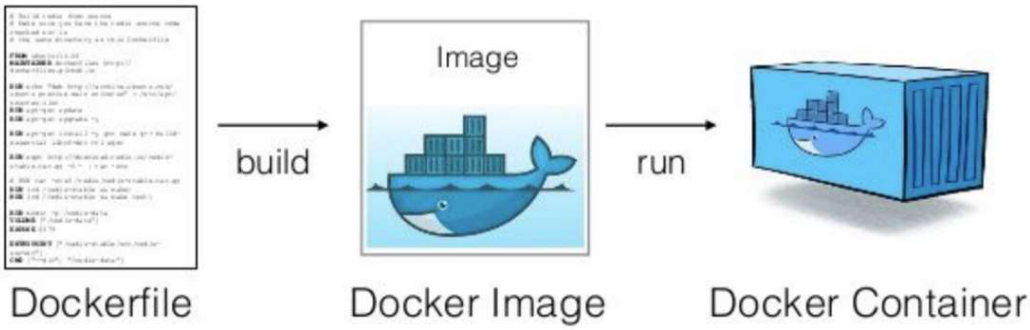
1. Bộ phận Thu thập Dữ liệu: Sử dụng thư viện BeautifulSoup4, chuyên dụng trong việc phân tích cú pháp HTML và XML để thu thập thông tin từ các trang web tuyển dụng.
2. Bộ phận Lưu trữ: Dữ liệu được lưu trữ vào Hadoop sử dụng HDFS File System (HDFS). Hadoop cung cấp khả năng lưu trữ phân tán, mở rộng, 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.

1.2. **Chi tiết về thành phần hệ thống**

1.2.1. Docker



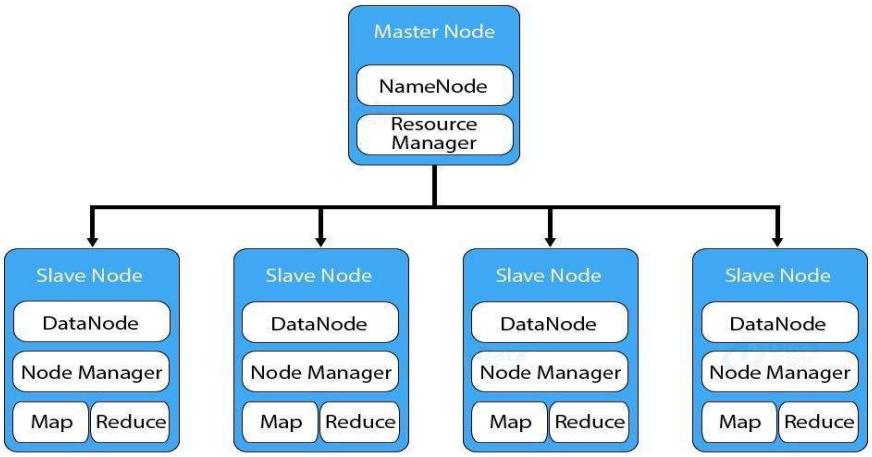
Docker, một nền tảng ảo hóa hóa mức độ hệ thống, cung cấp một giải pháp hiệu quả cho việc triển khai ứng dụng và môi trường mà không cần lo lắng về sự không tương thích giữa hệ điều hành và các phụ thuộc. Docker giúp đơn giản hóa quá trình xây dựng, đóng gói, và chia sẻ ứng dụng trong các môi trường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

1.2.2. Hadoop Cluster

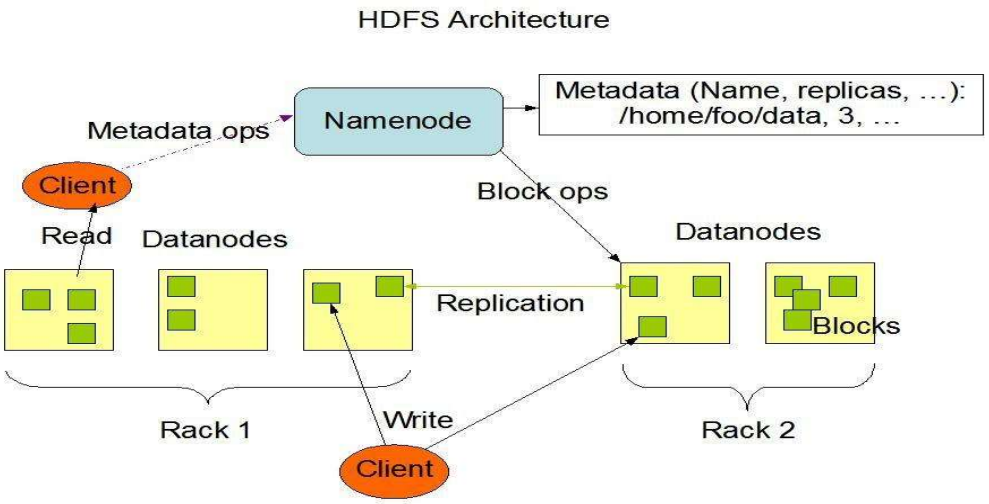
Hadoop Cluster là hệ thống file phân tán, cung cấp khả năng lưu trữ dữ liệu khổng lồ và tính năng tối ưu hoá việc sử dụng băng thông giữa các node.



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.

1.2.3. 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.

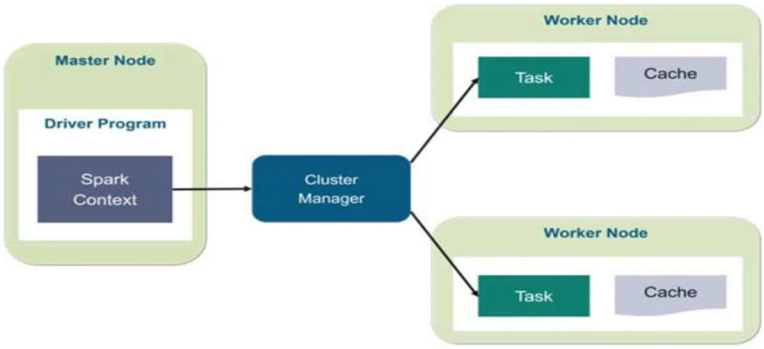
Spark cho phép xử lý dữ liệu theo thời gian thực, vừa nhận dữ liệu từ các nguồn khác nhau đồng thời thực hiện ngay việc xử lý trên dữ liệu vừa nhận được.

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.

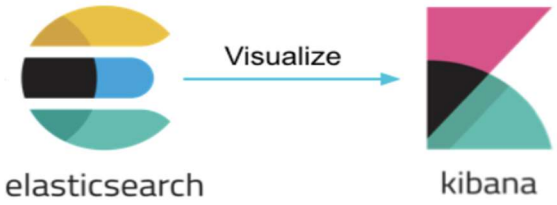
Apache Spark xây dựng các lệnh xử lý dữ liệu của người dùng thành Đồ thị vòng có hướng hoặc DAG. DAG là lớp lập lịch của Apache Spark; nó xác định những tác vụ nào được thực thi trên những nút nào và theo trình tự nào.



1.2.4. ElasticSearch và Kibana

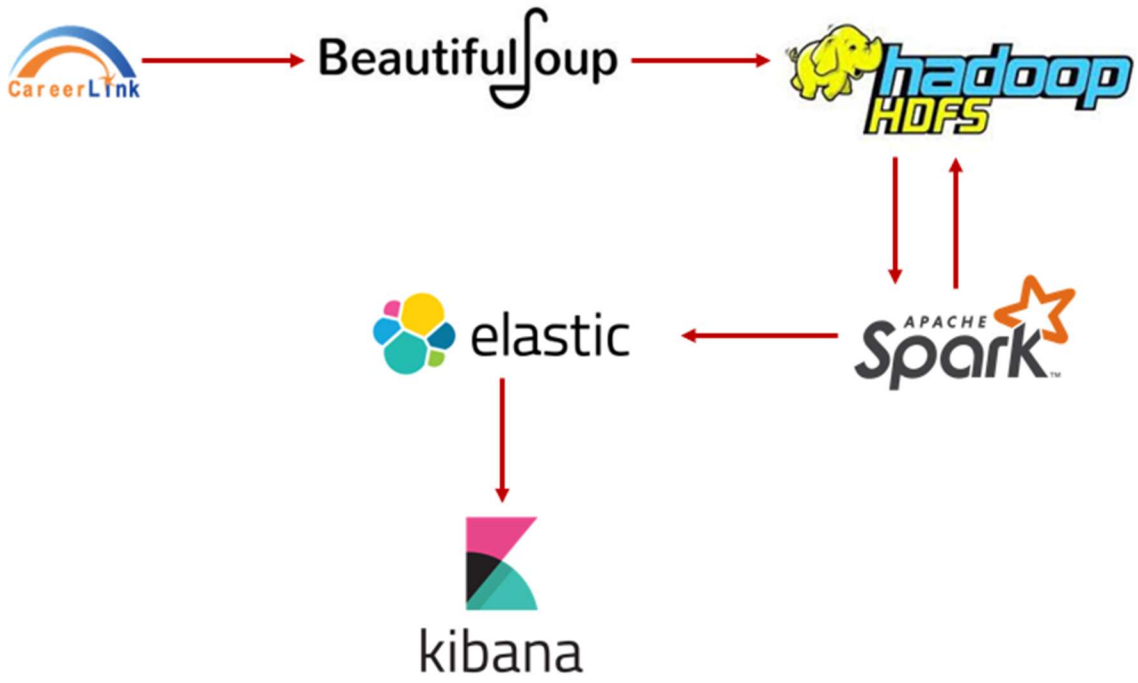
Dữ liệu sau khi được làm sạch bởi Spark cần được biểu diễn dưới dạng bảng biểu, đồ thị để mang đến cho người dùng góc nhìn trực quan nhất.

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

2.1. Luồng dữ liệu của hệ thống

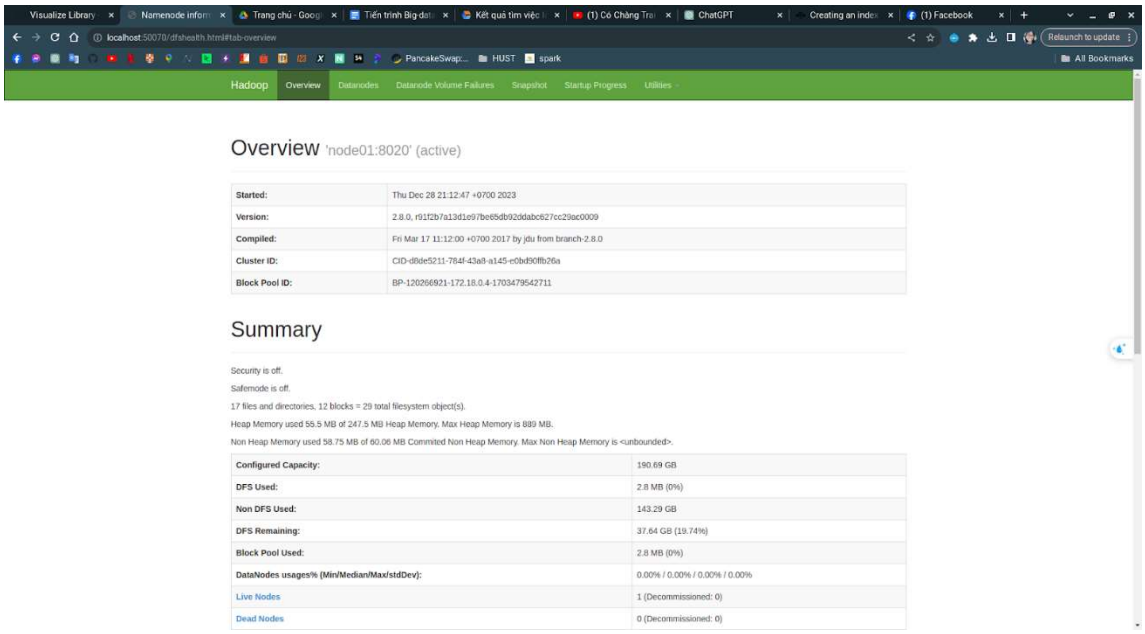


Luồng dữ liệu của hệ thống chúng em xây dựng gồm 4 quá trình:

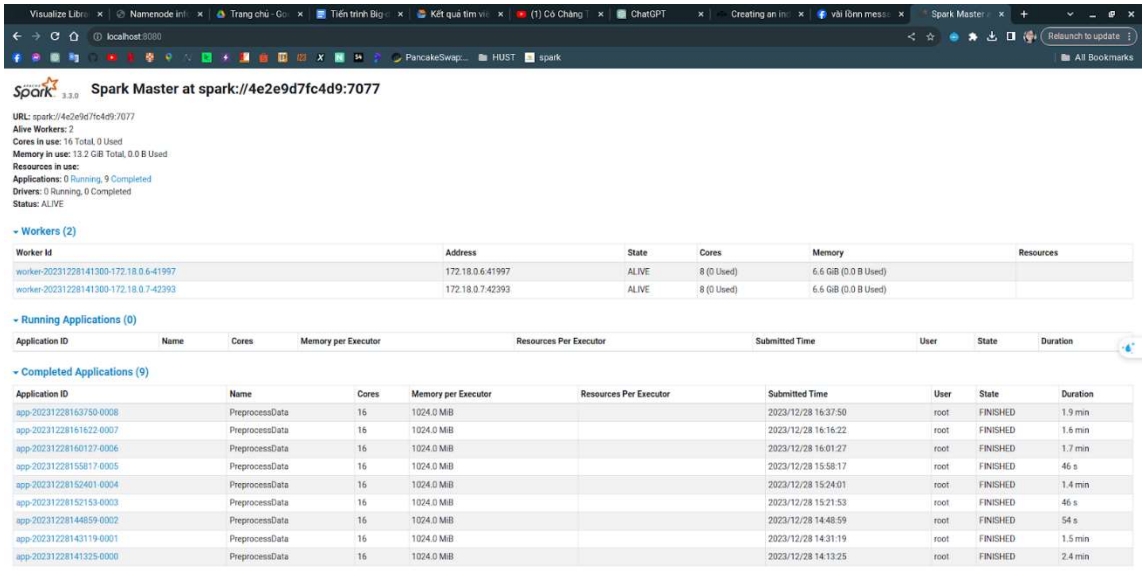
1. Thu thập dữ liệu trên website CareerLink.
2. Lưu dữ liệu vào Hadoop.
3. L ọc, làm sạch dữ liệu trên Hadoop bằng Spark. Sau đó lưu thành 2 bản: 1 bản lưu trả về Hadoop, 1 bản gửi lưu vào Elasticsearch.
4. Biểu diễn dữ liệu trên Elasticsearch dưới dạng biểu đồ, đồ thị, danh sách bảng sử dụng Kibana.

2.2. Khởi động hệ thống HDFS

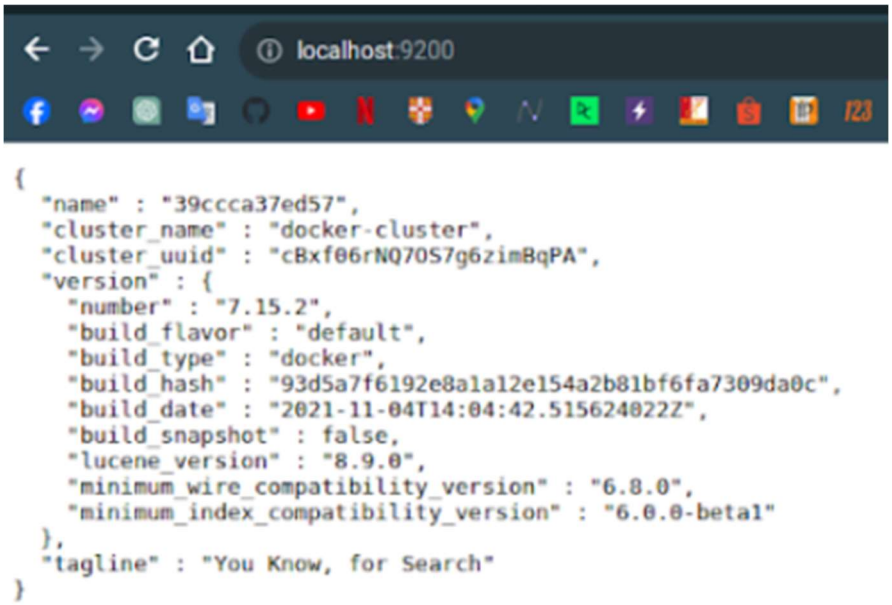
docker-compose up



Khởi động spark master: master.sh



Khởi động Elasticsearch:



2.3. Các trải nghiệm khi xây dựng chương trình và hệ thống

2.3.1. Thu thập dữ liệu

Dữ liệu của hệ thống là dữ liệu tuyển dụng liên quan đến lĩnh vực phần mềm, có thể được thu thập tại website CareerLink. Tại thời điểm dữ liệu được thu thập, trên CareerLink có tổng 170 trang, file html của mỗi trang có chứa link đến đơn tuyển dụng của từng công ty. Hệ thống sẽ truy cập vào từng link và thu thập thông tin theo các thẻ. Mỗi đơn tuyển dụng 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.

Website CareerLink: Kết quả tìm việc làm | Việc Làm, Tìm Việc, Tuyển Dụng | CareerLink.vn

Một bản ghi sẽ bao gồm các trường sau:

- Tên công việc

- Tên công ty
- Mô tả công việc
- Địa điểm công việc
- Mức lương
- Kinh nghiệm
- Yêu cầu ứng viên
- Hình thức công việc
- Cấp bậc
- Giới tính
- Ngành nghề

Chương trình thu thập dữ liệu của hệ thống được lưu ở file crawl_data.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). Để tăng tốc độ thực thi, hệ thống sử dụng một bash script để chạy song song 44 luồng cùng lúc, mỗi luồng thu thập dữ liệu trên 10 trang liên tiếp. Dữ liệu trả về được lưu ở 17 file json, tương ứng với kết quả chạy đồng thời của 44 luồng, mỗi file json sẽ bao gồm 25x10 = 250 bản ghi từ 10 trang đã thu thập.

Ví dụ về 1 bản ghi thu thập được từ 1 đơn tuyển dụng:

NHÂN VIÊN HỖ TRỢ KỸ THUẬT ĐTDĐ/LAPTOP [CÁI BÈ - TIỀN GIANG]

Công Ty Cổ Phần Bán Lẻ Kỹ Thuật Số FPT

Huyện Cái Bè, **Tiền Giang**

7 triệu - 12 triệu

0 - 1 năm kinh nghiệm

Ngày đăng tuyển 28-12-2023 | Hết hạn trong: **30 Ngày tới**

Nộp đơn ngay | Lưu

Mô tả | Kỹ năng yêu cầu | Chi tiết công việc | Liên hệ | Về công ty

Mô tả công việc

- Cài đặt ứng dụng và phần mềm cho các dòng ĐTDĐ/Laptop
- Tư vấn, hỗ trợ cho khách hàng về kỹ thuật và chức năng của sản phẩm
- Tiếp nhận và xử lý các trường hợp bảo hành liên quan đến phần mềm sản phẩm...
- Hỗ trợ nhân viên bán hàng về kỹ thuật
- Thời gian làm việc: Ca xoay (Ca 1: 8h00 – 15h00/ Ca 2: 15h00 – 22h00).

*** Phụ trách công việc tại các của hàng FPT Shop khu vực Tiền Giang (Cái Bè), địa chỉ cụ thể trao đổi khi phỏng vấn.**

Kinh nghiệm / Kỹ năng chi tiết

- Nam cao 1m65 trở lên, tuổi từ 18 - 27
- Tốt nghiệp THPT trở lên.
- Rành về kiến thức sản phẩm, am hiểu các hệ điều hành điện thoại, laptop
- Yêu thích công nghệ, ham học hỏi, chịu khó tìm tòi cập nhật kiến thức công nghệ mới
- Ngoại hình ưa nhìn, giọng nói rõ ràng, dễ nghe
- Kỹ năng giao tiếp tốt, năng động, vui vẻ và thân thiện

Mô tả

Loại công việc
Nhân viên toàn thời gian

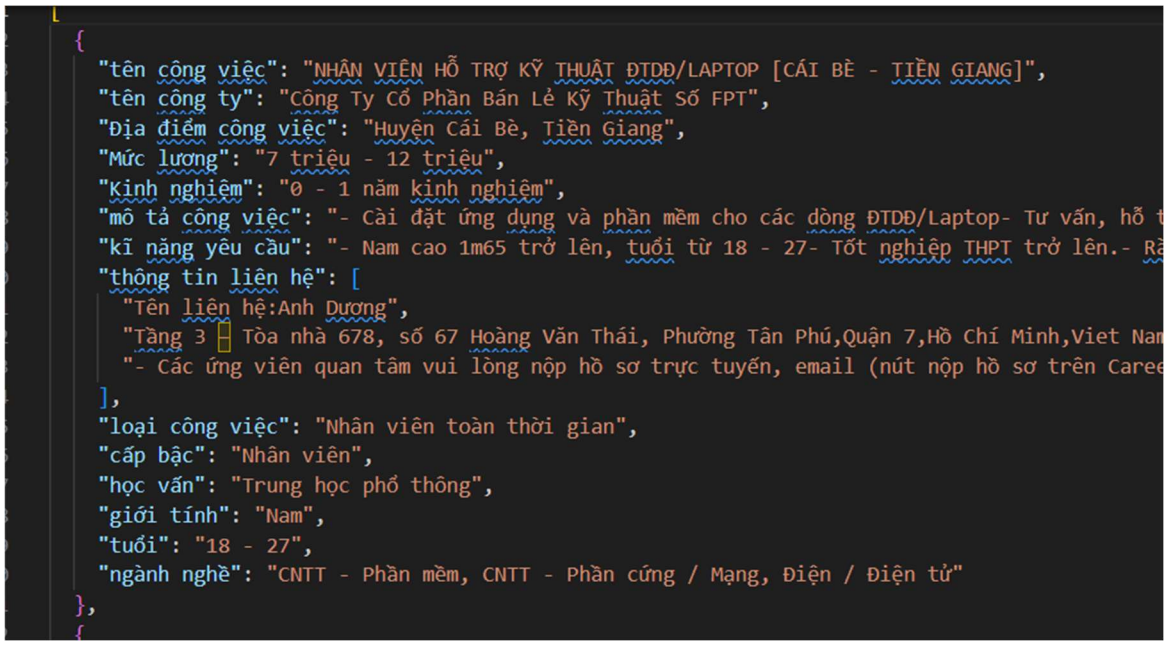
Cấp bậc
Nhân viên

Học vấn
Trung học phổ thông

Giới tính
Nam

Tuổi
18 - 27

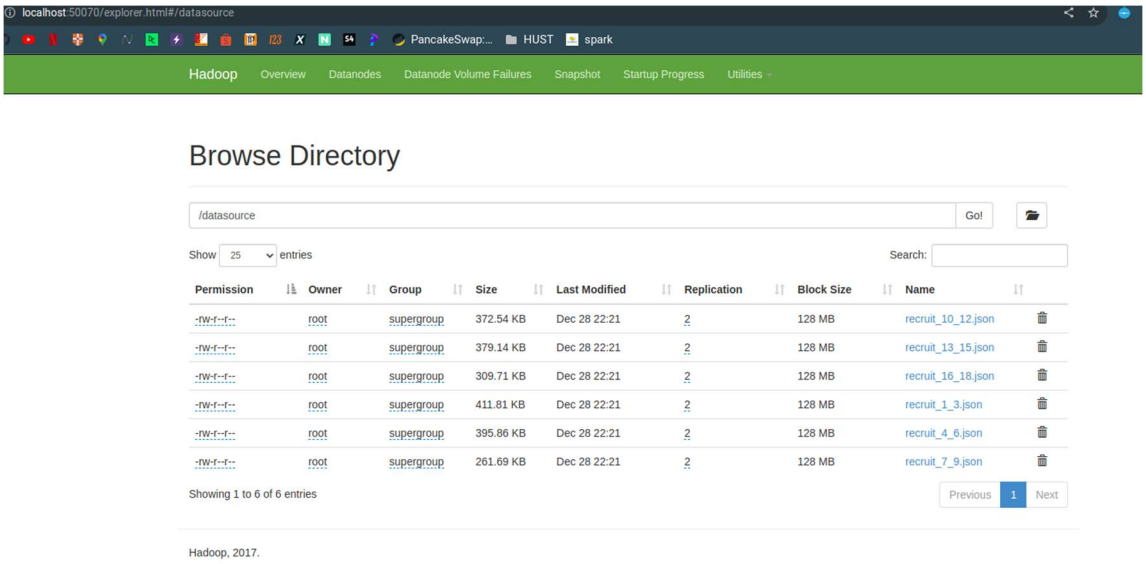
Ngành nghề
CNTT - Phần mềm , CNTT - Phần cứng / Mạng , Điện / Điện tử



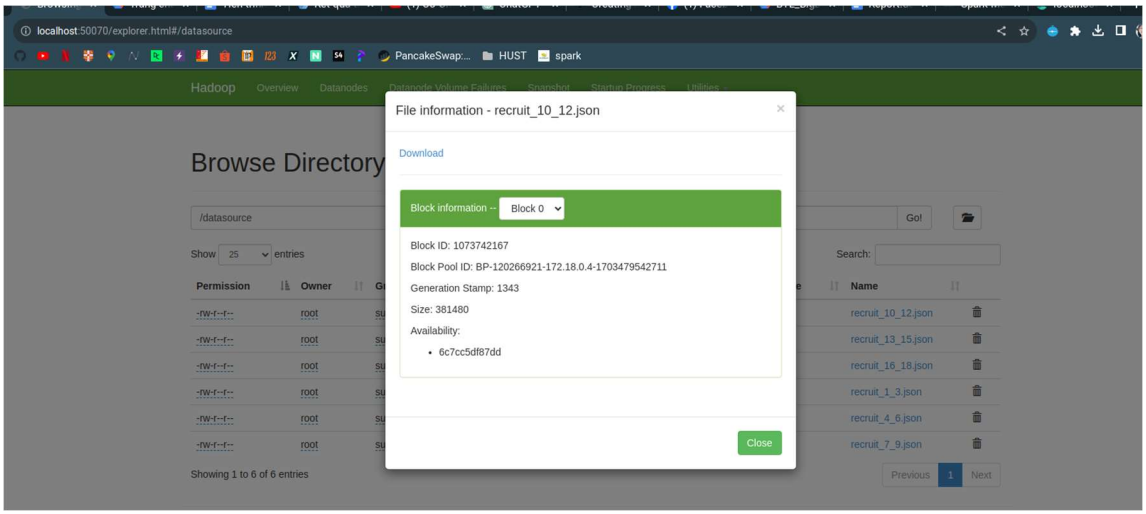

```
[
  {
    "tên công việc": "NHÂN VIÊN HỖ TRỢ KỸ THUẬT ĐTDĐ/LAPTOP [CÁI BÈ - TIỀN GIANG]",
    "tên công ty": "Công Ty Cổ Phần Bán Lẻ Kỹ Thuật Số FPT",
    "Địa điểm công việc": "Huyện Cái Bè, Tiền Giang",
    "Mức lương": "7 triệu - 12 triệu",
    "Kinh nghiệm": "0 - 1 năm kinh nghiệm",
    "mô tả công việc": "- Cài đặt ứng dụng và phần mềm cho các dòng ĐTDĐ/Laptop- Tư vấn, hỗ t
    "kĩ năng yêu cầu": "- Nam cao 1m65 trở lên, tuổi từ 18 - 27- Tốt nghiệp THPT trở lên.- Rà
    "thông tin liên hệ": [
      "Tên liên hệ:Anh Dương",
      "Tầng 3 - Tòa nhà 678, số 67 Hoàng Văn Thái, Phường Tân Phú,Quận 7,Hồ Chí Minh,Viet Nam
      "- Các ứng viên quan tâm vui lòng nộp hồ sơ trực tuyến, email (nút nộp hồ sơ trên Caree
    ],
    "loại công việc": "Nhân viên toàn thời gian",
    "cấp bậc": "Nhân viên",
    "học vấn": "Trung học phổ thông",
    "giới tính": "Nam",
    "tuổi": "18 - 27",
    "ngành nghề": "CNTT - Phần mềm, CNTT - Phần cứng / Mạng, Điện / Điện tử"
  },
  {
```

2.3.2. Lưu dữ liệu vào Hadoop

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



Dữ liệu được lưu trên datanode là node02



2.3.3. 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:

```
schema = StructType([
    StructField("tên công việc", StringType(), True),
    StructField("tên công ty", StringType(), True),
    StructField("Địa điểm công việc", StringType(), True),
    StructField("Mức lương", StringType(), True),
    StructField("Kinh nghiệm", StringType(), True),
    StructField("mô tả công việc", StringType(), True),
    StructField("kĩ năng yêu cầu", StringType(), True),
    StructField("thông tin liên hệ", StringType(), True),
    StructField("loại công việc", StringType(), True),
    StructField("cấp bậc", StringType(), True),
    StructField("học vấn", StringType(), True),
    StructField("giới tính", StringType(), True),
    StructField("tuổi", StringType(), True),
    StructField("ngành nghề", StringType(), True),
])
```

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 :

- LabeledKnowledges: một từ điển có dạng ‘tên kiến thức’: ‘lĩnh vực’
- KnowledgeGroups: một tập hợp các mảng, mỗi mảng là một lĩnh vực và các kiến thức thuộc lĩnh vực đó
- Educations: một mảng gồm các trình độ học vấn của nhân sự
- FrameworksPlattforms : một mảng gồm tên các frameworks, plattforms mà công ty tuyển dụng yêu cầu.
- ITLanguages: một mảng gồm tên các ngôn ngữ lập trình mà công ty tuyển dụng yêu cầu.

- Languages: một mảng gồm tên các các ngoại ngữ mà công ty tuyển dụng yêu cầu
- DesignPatterns : một mảng gồm tên các design patterns mà công ty tuyển dụng yêu cầu.
- Knowledges: một mảng gồm tên các kiến thức, các kỹ năng mà công ty tuyển dụng yêu cầu.
- SalaryPatterns : một mảng gồm các mức lương mà công ty tuyển dụng chi trả.
- Provinces: Một mảng gồm tên các tỉnh thành của nơi làm việc

Các trường thông tin FrameworksPlattforms, Languages, DesignPatterns, Knowledges, Educations được trích xuất theo cùng một cách là tìm các xâu trong dữ liệu gốc mà khớp với các xâu được định nghĩa sẵn (gọi là các pattern) tương ứng với mỗi trường.

Ví dụ, với trường FrameworksPlattforms:

```
framework_plattforms = ['Docker', 'OSP', 'Premiere', 'directAdmin', 'typography', 'Prometheus', 'visual weight', 'Kubernetes', 'JDBC', 'UnitTest',
            'Servlets', 'cPanel', 'MySQL', '.NET', 'Ruby on Rails', 'JSP', 'IdentityServer', 'VoIP', 'AdobeXD', 'CMake', 'Autocad',
            'Spring', 'Django', 'CRM', 'K8S', 'Nginx', 'firmware', 'Google Trend', 'psd', 'CSRF', 'Reactjs', 'Struts', 'WebSocket',
            'Webpack', 'Spine', 'Vue', 'METEOR', 'Rancher', 'VFX', 'node js', 'Angular', 'Flask', 'ASP.NET', 'Google Analystics', 'Zend',
            'Symfony', 'Express', 'Google Protobuf', 'J2EE', 'Ansible', 'WebForm', 'Videoscribe', 'CakePHP', 'Hibernate', 'Git', 'Oracle',
            'Plesk', 'Log4j', 'JSON', 'Visio', 'Grafana', 'SDLC', ' EELinux', ' Redis', 'Redux', 'WinForm', 'Figma', 'CodeIgniter',
            'Power BI', 'Bootstrap', 'WPF', 'Aerospike', 'bash shell', 'Laravel', 'SQL Server']
```

Trường Knowledges:

```
knowledges = ['game', 'Jira', 'lắp đặt', 'interaction design', 'đồ họa', 'DevOps', ' AI ', ' async', 'Quality Assurance',
              ' Security ', 'Google Drive', 'NFT', 'mạng máy tính', 'Wordpress', 'Machine Learning', 'Consult', 'White Box', 'sale',
              'kiểm thử', 'đánh giá chất lượng', 'networking', 'distributed system', 'UI/UX', 'Windows', 'Unit Test',
              'Jenkins', 'Chatbot', 'quản trị mạng', 'Solidity', 'tester', 'Corel Draw', 'Illustrator', 'Git', 'Android','phân tích dữ liệu',
              'Black Box', 'Office', 'chạy quảng cáo', 'Unix', 'IT Support', 'Data mining', 'data analys','Cài đặt','hệ điều hành', 'cấu trúc dữ
              ' TCP', 'qa', 'Animate', 'crypto', 'CI/CD', 'Defi', 'frontend', 'sửa chữa','Kinh tế', 'SVN', 'phần cứng', ' sync','BrSE', 'bảo mật'
              'Powerpoint', 'smart contract', 'Linux', 'SCM', 'backend', 'Marketing', 'XSS', 'Photoshop', 'HTTP', 'Word', 'router', 'IOS',
              'WebSocket', 'thuật toán', 'TestRail', 'CSDL', 'Sketch', 'blockchains', 'multithreading', 'hướng đối tượng', 'Front-end',
              'latex', 'Restful', 'Subversion', 'java web', 'Mobile', 'Excel', 'design pattern']
```

Đối với trường SalaryPatterns thì việc làm sạch dữ liệu sẽ phức tạp hơn. Bởi vì mức lương được biểu diễn dưới nhiều hình thức khác nhau như là 2000$, 20000000 VNĐ… Vì vậy hệ thống sẽ đồng nhất lương theo đơn vị triệu VNĐ và thống kê lương theo các khoảng 5 triệu VNĐ. Mức lương trong các đơn tuyển

dụng sẽ được chia vào các khoảng tương ứng, biểu diễn bằng một mảng các số nguyên là chặn dưới của mỗi khoảng.

Dưới đây cho một số ví dụ về việc chuyển đổi mức lương:

```
salary_patterns = ["lương(?:từ| )+ ((?:\d+|\.)+)", "((?:\d+|\.|-| )+(?:triệu| )+)đồng",
                   "(?:\d|\.|,)+.000.000", "(?:\d+| |-)+\d+ *(?:triệu|m)", "\$(?:\d+|,)", "(?:\d+|,)+ *(?:USD|\$)+",
                   "(?:\d|\.|,)+,000,000"]
```

Với mỗi trường, hệ thống dùng thư viện regex của python để tìm kiếm các pattern và trích xuất ra dữ liệu tương ứng. Lọc các thông tin về frameworks và plattfornms:

```
@udf(returnType=StringType())
def extract_location(dia_diem_cong_viec):
    for province in patterns.provinces:
        if re.search(province, dia_diem_cong_viec, re.IGNORECASE):
            return province
    return None
    # return len(dia_diem_cong_viec)

@udf(returnType=ArrayType(StringType()))
def extract_job_type(nganh_nghe):
    job_type =[]
    if re.search('CNTT - Phần mềm', nganh_nghe, re.IGNORECASE):
        job_type.append('software')
    if re.search('CNTT - Phần cứng / Mạng', nganh_nghe, re.IGNORECASE):
        job_type.append('hardware')
    return job_type


@udf(returnType=StringType())
def get_grouped_knowledge(knowledge):
    for x in knowledge:
        res = patterns.labeled_knowledges.get(x)
        if res is not None:
            return res

@udf(returnType=ArrayType(StringType()))
def extract_education(hoc_van, ki_nang_yeu_cau):
    res=[]
    for edu in patterns.educations:
        if re.search(edu, hoc_van+ " "+ki_nang_yeu_cau, re.IGNORECASE):
            res.append(edu)
    return res
```

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:

```
raw_recruit_df = spark.read.schema(schema).option("multiline", "true").json(
    "hdfs://node01:8020/datasource/*.json")
extracted_recruit_df = raw_recruit_df.select(raw_recruit_df["tên công việc"].alias("JobName"),
                                             raw_recruit_df['tên công ty'].alias("CompanyName"),
                                             udfs.extract_location("Địa điểm công việc").alias("Location"),
                                             udfs.extract_exp_pattern('Kinh nghiệm').alias("Experience"),
                                             raw_recruit_df['loại công việc'].alias("JobType"),
                                             raw_recruit_df["cấp bậc"].alias('Level'),
                                             udfs.extract_education("học vấn", "kĩ năng yêu cầu").alias('Education'),
                                             raw_recruit_df["giới tính"].alias('Sex'),
                                             udfs.extract_old_pattern("tuổi").alias("Old"),
                                             udfs.extract_framework_plattform("mô tả công việc",
                                                                              "kĩ năng yêu cầu").alias(
                                                 "FrameworkPlattforms"),
                                             udfs.extract_IT_language("mô tả công việc", "kĩ năng yêu cầu").alias(
                                                 "JobLanguages"),
                                             udfs.extract_language("kĩ năng yêu cầu").alias("Languages"),
                                             udfs.extract_design_pattern("mô tả công việc",
                                                                         "kĩ năng yêu cầu").alias("DesignPatterns"),
                                             udfs.extract_knowledge("mô tả công việc", "kĩ năng yêu cầu").alias(
                                                 "Knowledges"),
                                             udfs.normalize_salary("Mức lương").alias("Salaries"),
                                             raw_recruit_df['thông tin liên hệ'].alias("Contact"),
                                             udfs.extract_job_type("ngành nghề").alias("JobSummary"),
                                             ).withColumn('Knowledge', udfs.get_grouped_knowledge("Knowledges"))
print('extract successuly!!!!')
extracted_recruit_df.cache()
extracted_recruit_df.show(5)

salaries_not_null= queries.get_not_null_salary(extracted_recruit_df)
salaries_not_null.show(5)
```

Các dòng đầu của dataframe lọc từ dataframe ban đầu:

CompanyName	FrameworkPlattforms	Languages	DesignPatterns	Knowledges	Salaries
CÔNG TY TNHH ZINZ...	[Vue, Laravel]	[Python, PHP, Ruby]	[]	[]	[0, 5, 10]
CÔNG TY TNHH QUỐC...	[Premiere]	[]	[]	[Marketing]	[10]
CÔNG TY TNHH SOFT...	[]	[PHP, Java]	[]	[]	[]
CÔNG TY TNHH CÔNG...	[]	[]	[]	[]	[15]
CÔNG TY TNHH CÔNG...	[MySQL, Zend, Cak...	[PHP, css]	[]	[]	[25]

Tiền xử lý và lưu dữ liệu: Dataframe extracted_recruit_df về cơ bản là đã có thể tiến hành biểu diễn trên Kibana, tuy nhiên ta vẫn cần tiến hành tiền xử lý thêm một só bước để việc biểu diễn dễ dàng hơn. Khi người dùng quan tâm đến một nhóm các kiến thức mà thị trường tuyển dụng đang yêu cầu, thay vì các tri thức riêng rẽ, ví dụ như quan tâm đến một nhóm các kiến thức về blockchain và bảo mật, thay vì chỉ quan tâm đến các kiến thức cụ thể như smart contract hay Defi. Lúc này, chương trình cần gán nhãn trước các cho các kiến thức về một nhóm kiến

thức. Với các nhãn này, từ dataframe extracted_recruit_df có thể đếm ra được các bản ghi chứa một nhóm tri thức cụ thể.

Nhãn của một số kiến thức yêu cầu:

```
labeled_knowledges={'AI': 'AI', 'Machine Learning': 'AI', 'Data mining': 'AI', 'Chatbot': 'AI', 'data analys': 'AI',

                    'blockchains': 'blockchain_crypto', 'crypto': 'blockchain_crypto', 'NFT': 'blockchain_crypto',
                    'smart contract': 'blockchain_crypto', 'Solidity': 'blockchain_crypto', 'Defi': 'blockchain_crypto',
                    'XSS': 'blockchain_crypto',
                    'data analys': 'Data', 'Data mining':'Data', 'phân tích dữ liệu':'Data',
                    'lắp đặt': 'hardware', 'sửa chữa': 'hardware', 'phần cứng': 'hardware', 'router': 'hardware', 'cài đặt': 'hardware',
                    'Corel Draw': 'hardware', 'Switch': 'hardware',

                    'Word': 'office', 'Excel': 'office', 'Powerpoint': 'office', 'Office': 'office',

                    'Illustrator': 'photoshop', 'Photoshop': 'photoshop', 'Animate': 'photoshop',

                    'cấu trúc dữ liệu': 'programming_basic', 'thuật toán': 'programming_basic', 'OOP': 'programming_basic',
                    'hướng đối tượng': 'programming_basic', 'hệ điều hành': 'programming_basic', 'multithreading': 'programming_basic',

                    'Black Box': 'tester', 'tester': 'tester', 'White Box': 'tester', 'Unit Test': 'tester',
                    'TestRail': 'tester', 'kiểm thử': 'tester',

                    'bảo mật':'security', 'Security': 'security',

                    'SVN': 'version_control', 'SCM': 'version_control', 'Git': 'version_control',
                    'Front-end': 'Web','frontend': 'Web', 'backend':'Web', 'java web':'Web',
                    'mạng máy tính': 'Mạng', 'quản trị mạng':'Mạng'}
```

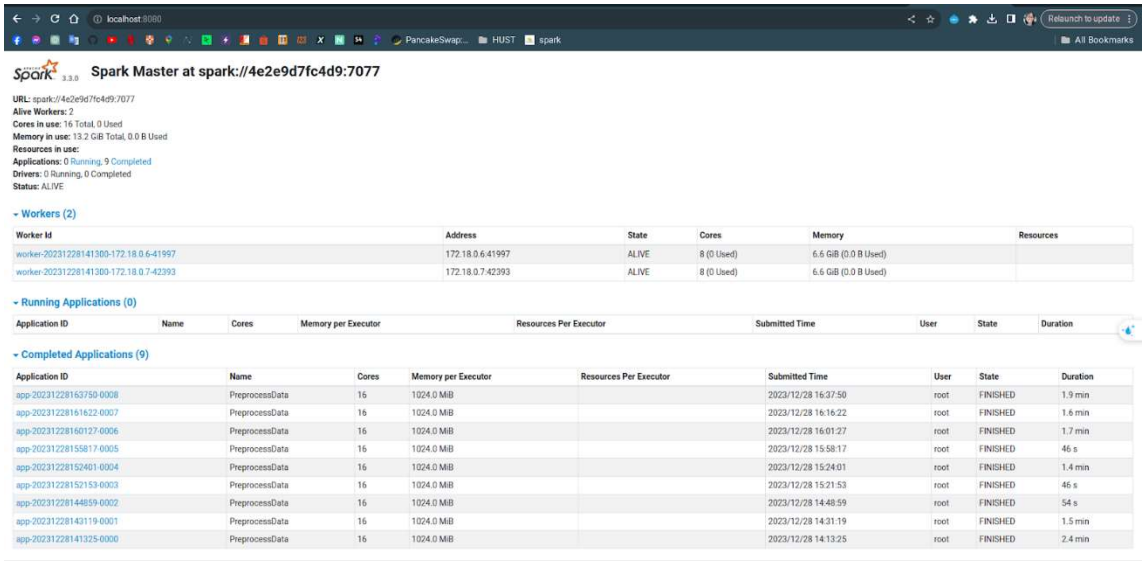
Trình sử dụng 1 hàm udf để đánh nhãn các string trong cột Knowledge của dataframe extracted_recruit_df.

```
@udf(returnType=StringType())
def get_grouped_knowledge(knowledge):
    for x in knowledge:
        res = patterns.labeled_knowledges.get(x)
        if res is not None:
            return res
```

Dữ liệu lúc này đã sẵn sàng để lưu về Hadoop và Elasticsearch, chương trình sử dụng 2 hàm save_dataframes_to_hdfs() và save_dataframes_to_elasticsearch() để tiến hành lưu trữ.

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:

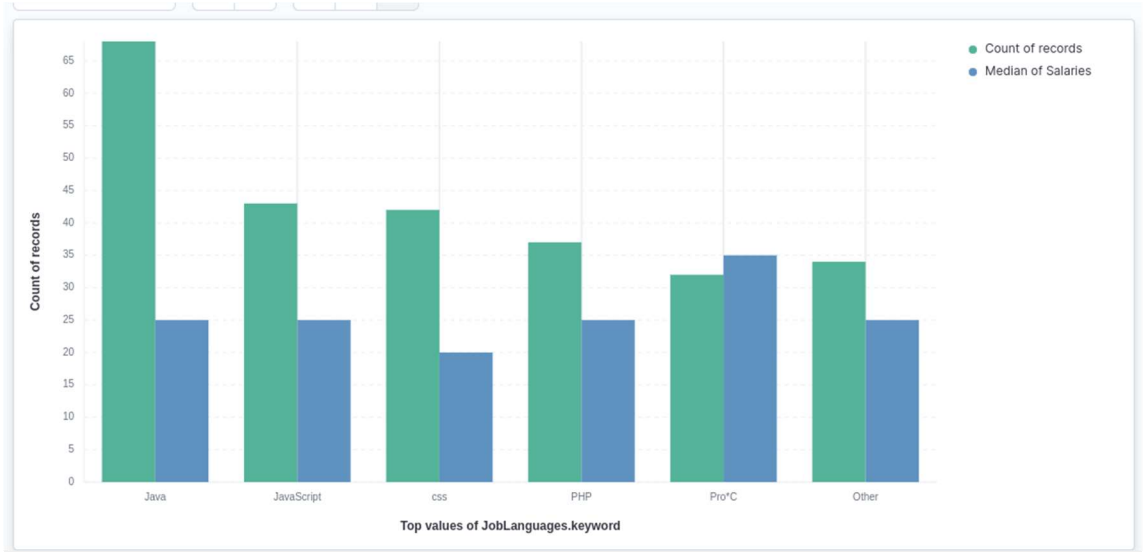


2.3.4. Biểu diễn dữ liệu bằng Kibana

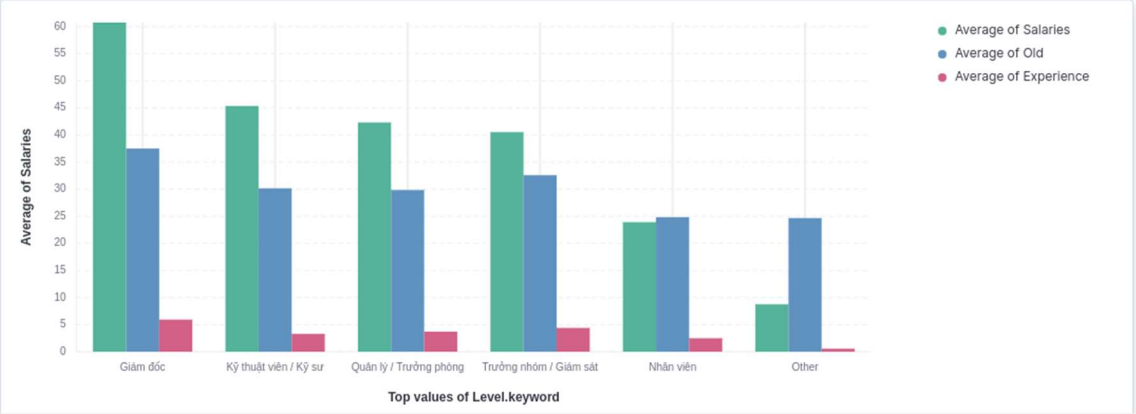
Dữ liệu lưu tại Elasticsearch sẽ được dùng Kibana để biểu diễn

Ví dụ:

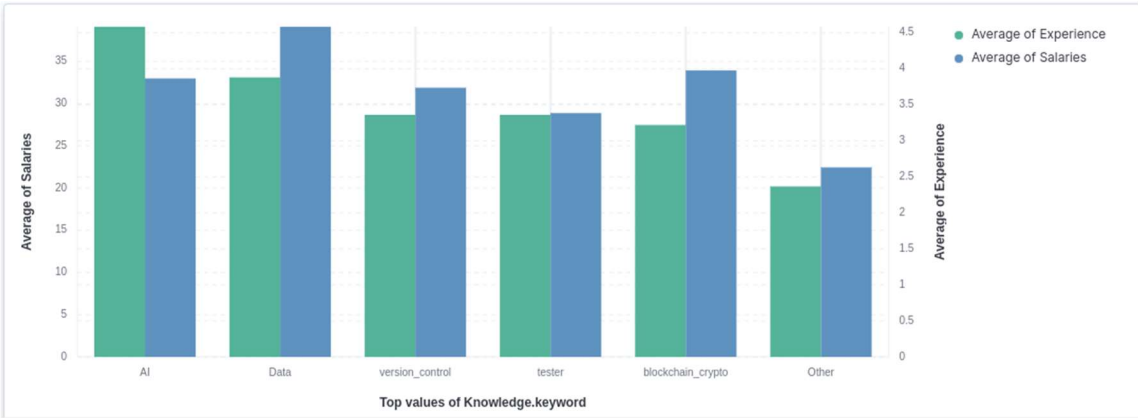
1. Mối quan hệ giữa lương và các ngôn ngữ lập trình



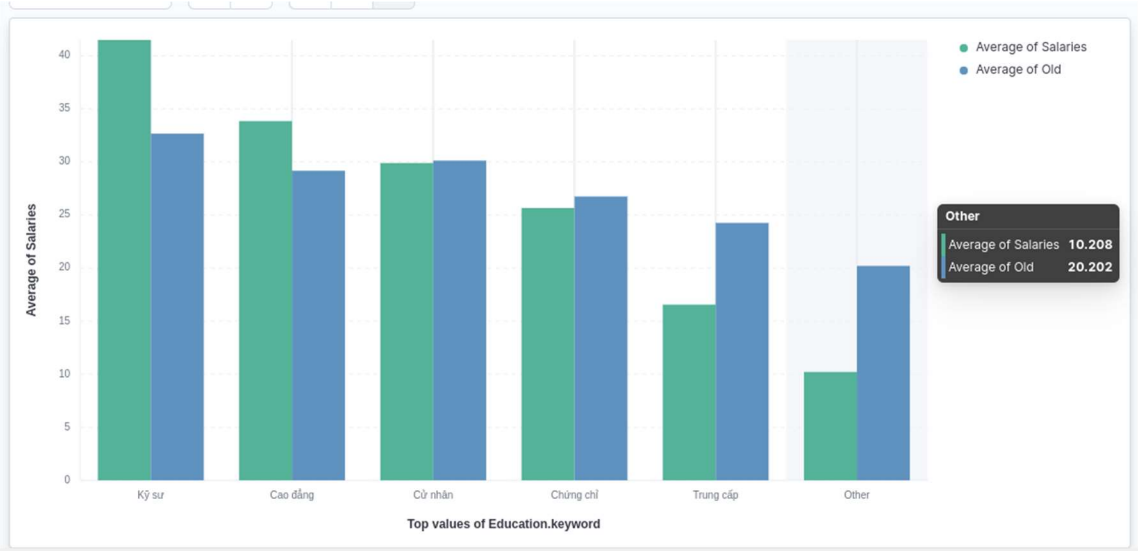
2. Mối quan hệ giữa lương, kinh nghiệm, số tuổi của các vị trí trong công ty



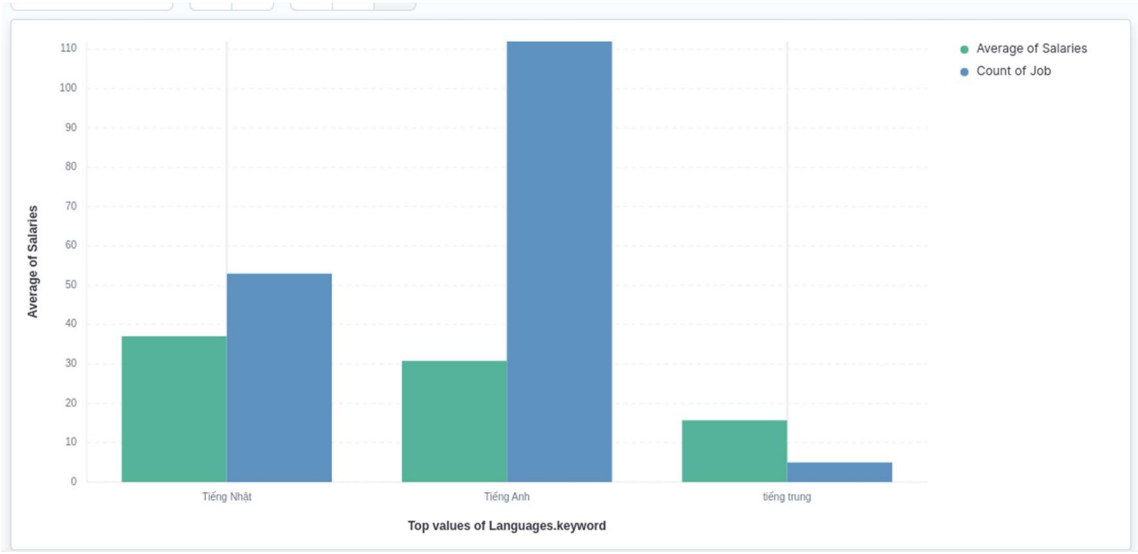
3. Mối quan hệ giữa lương và những phân ngành trong IT



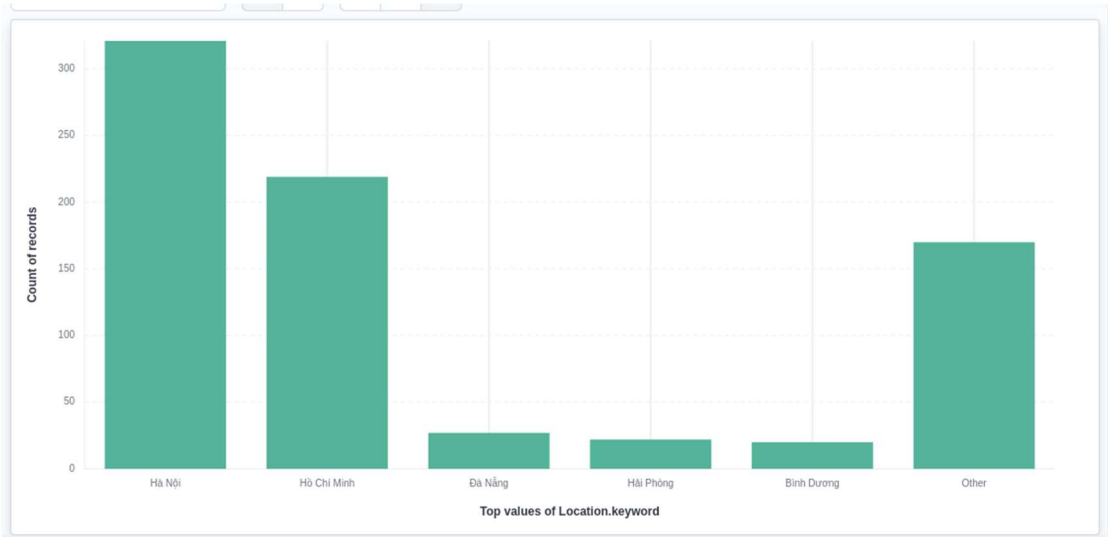
4. Mối quan hệ giữa bằng cấp, lương, tuổi



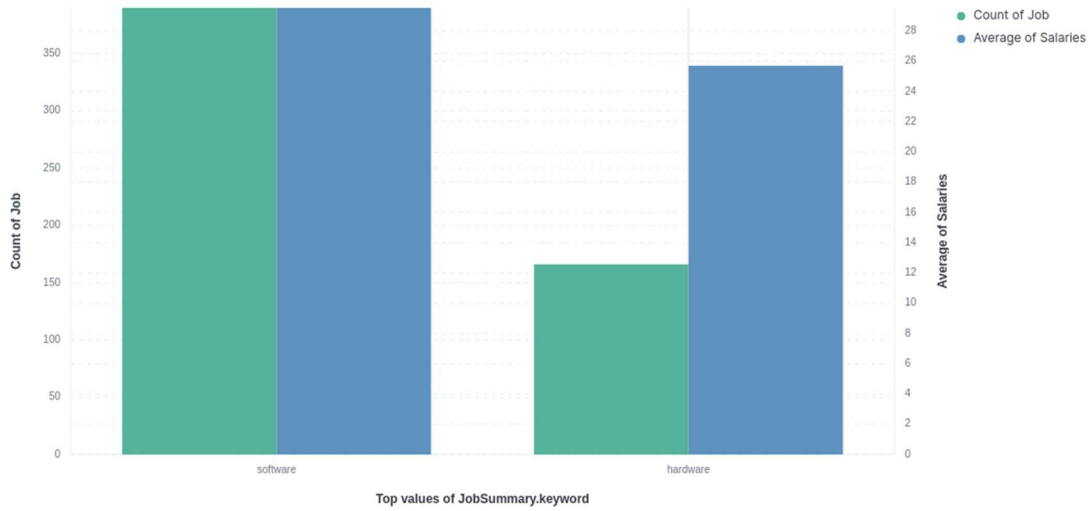
5. Mối quan hệ giữa ngoại ngữ và lương



6. Phân bố số lượng công việc trên các tỉnh thành



7. Tương quan giữa lương và số lượng công việc liên quan đến phần cứng và phần mềm



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

3.1. 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.

3.2. 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
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
8. https://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)