

**TRƯỜNG ĐẠI 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 : 136842

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 31:

Họ và tên	Mã số sinh viên
Nguyễn Phương Trung	20194932
Trương Văn Hiên	20194276
Mai Minh Nhật	20194346
Trần Quốc Anh	20194225

*Hà Nội, tháng 2 năm 2023*

## MỤC LỤC

LỜI NÓI ĐẦU.....	3
CHƯƠNG 1: TỔNG QUAN XÂY DỰNG HỆ THỐNG .....	5
1.1. Tổng quan hệ thống.....	5
1.2. Chi tiết về thành phần hệ thống .....	6
1.2.1. SSH Server.....	6
1.2.2. Hadoop Cluster.....	7
1.2.3. Spark Cluster.....	8
1.2.4. ElasticSearch và Kibana.....	9
CHƯƠNG 2: XÂY DỰNG CHƯƠNG TRÌNH VÀ HỆ THỐNG .....	11
2.1. Luồng dữ liệu của hệ thống .....	11
2.2. Khởi động hệ thống HDFS.....	12
2.3. Quá trình thực hiện.....	14
2.3.1. Thu thập dữ liệu .....	14
2.3.2. Lưu dữ liệu vào Hadoop .....	16
2.3.3. Lọc dữ liệu bằng Spark.....	17
2.3.4. Biểu diễn dữ liệu bằng Kibana.....	21
CHƯƠNG 3: NHẬN XÉT, ĐÁNH GIÁ VÀ HƯỚNG PHÁT TRIỂN .....	23
3.1. Nhận xét, đánh giá.....	23
3.2. Hướng phát triển .....	23
DANH MỤC TÀI LIỆU THAM KHẢO .....	24

## LỜI NÓI ĐẦU

Trước đây, khi mạng Internet còn chưa phát triển, lượng dữ liệu con người sinh ra khá nhỏ giọt và thưa thớt, nhìn chung, lượng dữ liệu này vẫn nằm trong khả năng xử lý của con người dù bằng tay hay bằng máy tính. Tuy nhiên trong kỷ nguyên số, khi mà sự bùng nổ công nghệ truyền thông đã dẫn tới sự bùng nổ dữ liệu người dùng, lượng dữ liệu được tạo ra vô cùng lớn và đa dạng, đòi hỏi một hệ thống đủ mạnh để phân tích và xử lý những dữ liệu đó.

Khái niệm Big Data đề cập tới dữ liệu lớn theo 3 khía cạnh khác nhau, thứ nhất là tốc độ sinh dữ liệu (velocity), thứ hai là lượng dữ liệu (volume) và thứ ba là độ đa dạng (variety). Lượng dữ liệu này có thể đến từ nhiều nguồn khác nhau như các nền tảng truyền thông Google, Facebook, Twitter, ... hay thông số thu thập từ các cảm biến, thiết bị IoT trong đời sống, ... Và một sự thật rằng doanh nghiệp nào có thể kiểm soát và tạo ra tri thức từ những dữ liệu này sẽ tạo ra một tiềm lực rất lớn để cạnh tranh với những doanh nghiệp khác. Có thể nói rằng dữ liệu là sức mạnh của kỷ nguyên số cũng không hề ngoa một chút nào.

Để tiếp cận với lĩnh vực này, nhóm chúng em quyết định chọn một loại dữ liệu đủ lớn trong khả năng để tiến hành tiến hành phân tích và lưu trữ. Thông tin tuyển dụng việc làm là một trong những thông tin được nhiều người quan tâm, đặc biệt là những lao động đang cần tìm việc làm. Những thông tin này thường xuất hiện ở các nhóm tuyển dụng trên mạng xã hội và các trang web tuyển dụng, trang tuyển dụng riêng của công ty. Việc khai thác được thông tin nhu cầu tuyển dụng có thể giúp cho người lao động tìm được công việc phù hợp, các công ty có thể cân nhắc điều chỉnh, những người đang có việc làm có thể đánh giá được mức năng lực của mình có nhận được lợi ích phù hợp khi ở công ty không hay cũng như việc điều chỉnh các chương trình đào tạo để tạo ra nguồn nhân lực phù hợp sau này. Để biết được thị trường lao động đang cần gì, một giải pháp đơn giản mà hiệu quả là thực hiện đánh giá, thống kê những kỹ năng, kiến thức được miêu tả trong các đơn tuyển dụng của các công ty trên các trang mạng tìm việc làm. 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 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 TopCV.

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

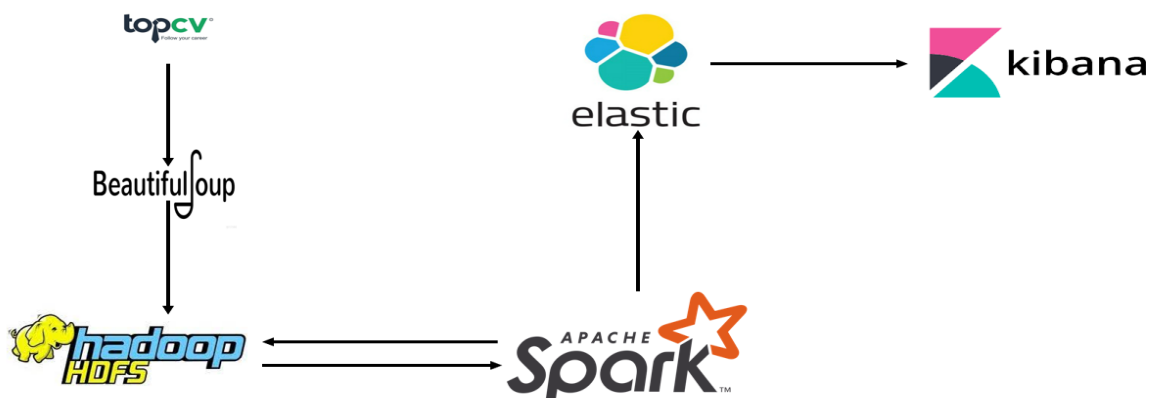
- Tổng quan xây dựng hệ thống
- Xây dựng chương trình và hệ thống
- Nhận xét, đánh giá và hướng phát triển

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 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

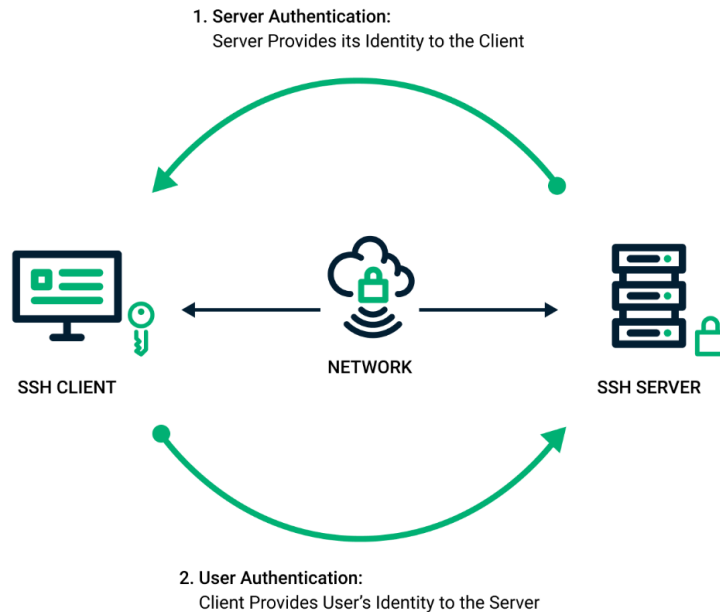


Hệ thống được xây dựng gồm 4 phần với các chức năng nhằm thu thập, xử lý, lưu trữ và trực quan hoá dữ liệu tuyển dụng từ thông tin tuyển dụng trong trang web. Các thành phần của hệ thống bao gồm:

1. Bộ phần thu thập dữ liệu: sử dụng BeautifulSoup4, là một thư viện để phân tích cú pháp các văn bản dạng HTML và XML, chuyên dụng trong việc thu thập dữ liệu từ các trang web.
2. Bộ phần 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.
3. Bộ phần 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 Elasticsearch.
4. Bộ phần biểu diễn dữ liệu: dữ liệu sau khi được xử lý bởi Spark được đưa vào Elasticsearch thông qua một thư viện mã nguồn mở là Elasticsearch for Apache Hadoop.

## 1.2. Chi tiết về thành phần hệ thống

### 1.2.1. SSH Server



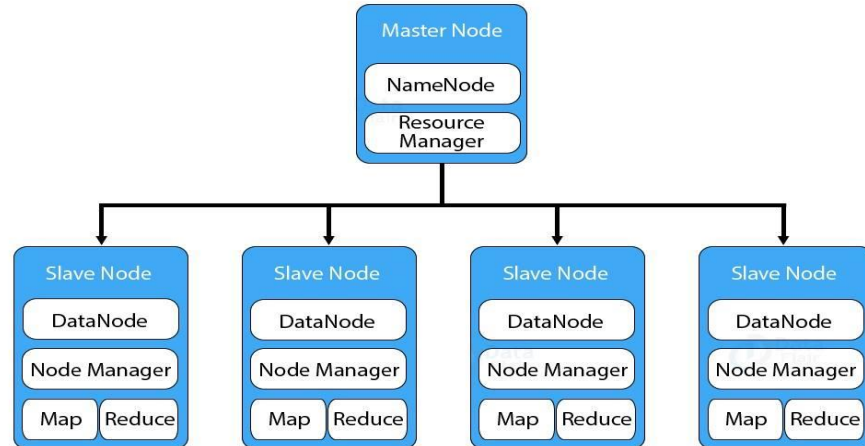
SSH, hay Secure (Socket) Shell, bao gồm cả giao thức mạng lẫn một bộ tiện ích để triển khai giao thức đó. SSH sử dụng mô hình client-server, kết nối một ứng dụng Secure Shell client (nơi session được hiển thị) với một SSH server (nơi session chạy). Triển khai SSH thường hỗ trợ cả các giao thức ứng dụng, dùng cho giả lập terminal hay truyền file.

Hadoop core sử dụng Shell (SSH) để giao tiếp với các slave node và để khởi chạy các quy trình máy chủ trên các slave node. Việc sử dụng cơ chế key-pair giúp việc giao tiếp giữa các máy không cần nhập nhiều lần mật khẩu mà vẫn đảm bảo độ bảo mật.

Khi Cluster đang hoạt động trong môi trường phân tán và việc giao tiếp cần thực hiện nhanh, SSH giúp cho NodeManager và các DataNode có thể giao tiếp với Namenode nhanh chóng.

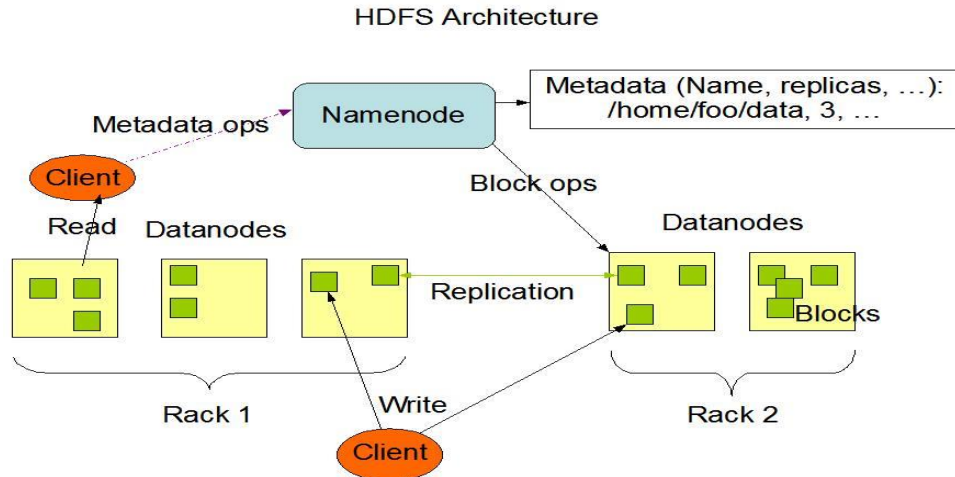
### 1.2.2. Hadoop Cluster

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 liệu được chia thành các chunk và được lưu thành nhiều bản sao, giúp cho cụm Hadoop có khả năng chịu lỗi.

HDFS là nơi lưu trữ dữ liệu của Hadoop, HDFS 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 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.

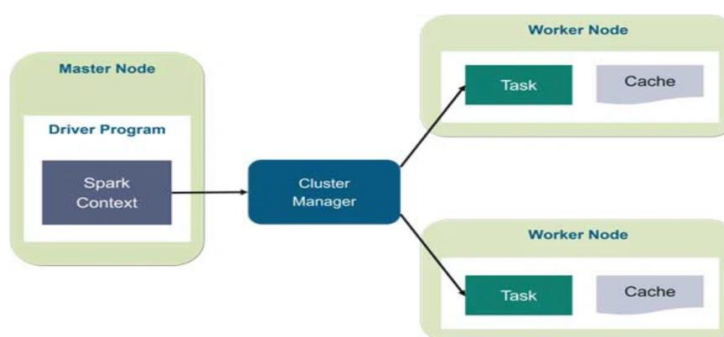
Những điểm nổi bật của Spark:

- 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

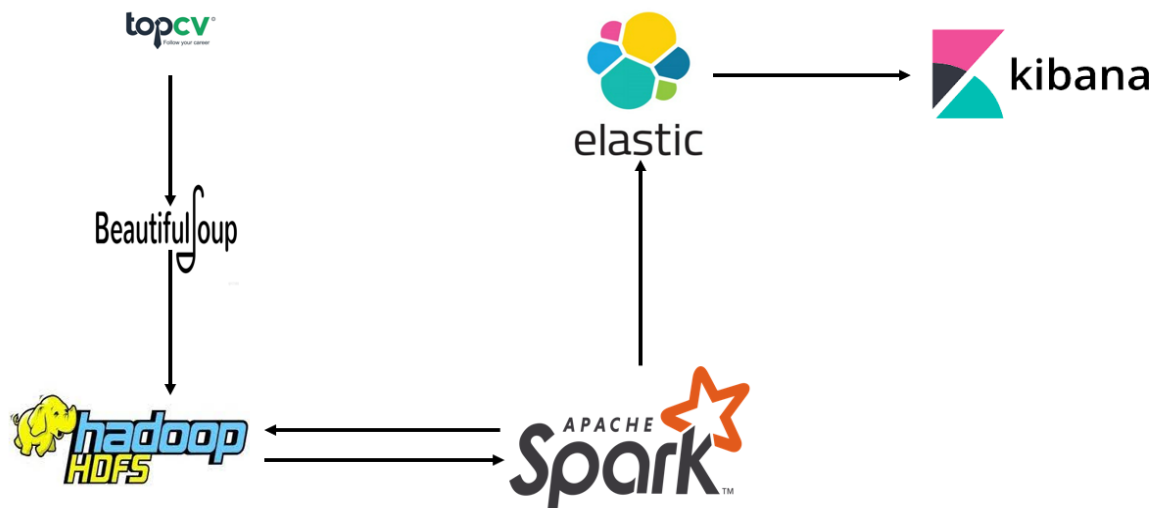
Dữ liệu sau khi được làm sạch bởi Spark cần được biểu diễn dưới dạng bảng biểu, đồ thị để mang đến cho người dùng góc nhìn trực quan nhất. Elasticsearch và Kibana là những ứng dụng phù hợp để đảm nhận vai trò này. Là một công cụ tìm kiếm (với tốc độ gần thời gian thực) và phân tích dữ liệu phân tán, 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 họa 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 TopCV.
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.

## 2.2. Khởi động hệ thống HDFS

*hdfs namenode -format*

*start-dfs.sh*

*start-yarn.sh*

Sử dụng lệnh `jps` xem các tiến trình đang chạy

Localhost:

The screenshot shows the Hadoop NameNode web interface in a browser. The address bar shows `localhost:50070/dfshealth.html`. The interface has a green header with tabs: Hadoop, Overview (selected), Datanodes, Snapshot, Startup Progress, and Utilities. The main content area is titled "Overview 'localhost:9000' (active)". It contains a table with the following information:

Started:	Sun Apr 06 15:52:11 IST 2014
Version:	2.3.0, r1567123
Compiled:	2014-02-11T13:40Z by jenkins from branch-2.3.0
Cluster ID:	CID-5edbd0da-c69f-425b-bbc7-a662ac5d45dc
Block Pool ID:	BP-1127675761-127.0.1.1-1396692597591

Below the table is a "Summary" section with the following text:

Security is off.  
Safemode is off.  
35 files and directories, 17 blocks = 52 total filesystem object(s).  
Heap Memory used 34.01 MB of 88.5 MB Heap Memory. Max Heap Memory is 889 MB.  
Non Heap Memory used 40.17 MB of 40.69 MB Committed Non Heap Memory. Max Non Heap Memory is -1 B.

At the bottom, a "Configured Capacity:" row shows "91.54 GB".

The screenshot shows the "Datanode Information" page in the Hadoop NameNode web interface. The header has tabs: Hadoop, Overview, Datanodes (selected), Datanode Volume Failures, Snapshot, Startup Progress, and Utilities. The main content area is titled "Datanode Information" and has a sub-section "In operation". Below this is a table with the following columns: Node, Last contact, Admin State, Capacity, Used, Non DFS Used, Remaining, Blocks, Block pool used, Failed Volumes, and Version. The table contains one row of data for the node `localhost:50010 (10.20.0.142:50010)`.

Node	Last contact	Admin State	Capacity	Used	Non DFS Used	Remaining	Blocks	Block pool used	Failed Volumes	Version
localhost:50010 (10.20.0.142:50010)	0	In Service	193.63 GB	28 KB	9.92 GB	183.71 GB	0	28 KB (0%)	0	2.7.1

Below the table is a "Decommissioning" section with a table that has columns: Node, Last contact, Under replicated blocks, Blocks with no live replicas, and Under Replicated Blocks in files under construction. This table is currently empty.

At the bottom, it says "Hadoop, 2015."

## IT4931 – Lưu trữ và xử lý dữ liệu lớn

Khởi động spark master: master.sh



Spark Master at spark://master:7077

URL: spark://master:7077  
Active Workers: 1  
Cores in use: 8 Total, 0 Used  
Memory in use: 6.6 GiB Total, 0.0 GiB Used  
Resources in use:  
Applications: 0 Running, 0 Completed  
Drivers: 0 Running, 0 Completed  
Status: ALIVE

Workers (1)

Worker ID	Address	State	Cores	Memory	Resources
worker-20220726143747-192.168.1.24-33189	192.168.1.24:33189	ALIVE	8 (0 Used)	6.6 GiB (0.0 GiB Used)	

Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
----------------	------	-------	---------------------	------------------------	----------------	------	-------	----------

Completed Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
----------------	------	-------	---------------------	------------------------	----------------	------	-------	----------

Khởi động spark worker: worker.sh



Spark Worker at 192.168.1.24:33189

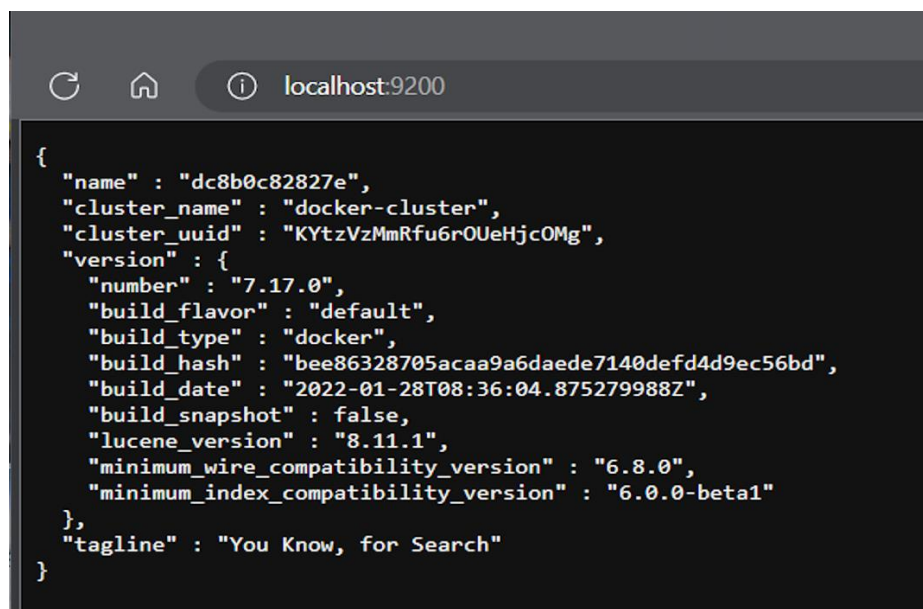
ID: worker-20220726143747-192.168.1.24-33189  
Master URL: spark://master:7077  
Cores: 8 (0 Used)  
Memory: 6.6 GiB (0.0 GiB Used)  
Resources:

Back to Master

Running Executors (0)

ExecutorID	State	Cores	Memory	Resources	Job Details	Logs
------------	-------	-------	--------	-----------	-------------	------

Khởi động Elasticsearch:



```
{
  "name" : "dc8b0c82827e",
  "cluster_name" : "docker-cluster",
  "cluster_uuid" : "KYtzVzMmRfu6rOUeHjcOMg",
  "version" : {
    "number" : "7.17.0",
    "build_flavor" : "default",
    "build_type" : "docker",
    "build_hash" : "bee86328705acaa9a6daede7140defd4d9ec56bd",
    "build_date" : "2022-01-28T08:36:04.875279988Z",
    "build_snapshot" : false,
    "lucene_version" : "8.11.1",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

## 2.3. Quá trình thực hiện

### 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 TopCV. Tại thời điểm dữ liệu được thu thập, trên TopCV 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 TopCV: [https://www.topcv.vn/tim-viec-lam-it-phan-mem-c10026?salary=0&exp=0&company\\_field=0&sort=up\\_top&page=](https://www.topcv.vn/tim-viec-lam-it-phan-mem-c10026?salary=0&exp=0&company_field=0&sort=up_top&page=)

Một bản ghi sẽ bao gồm các trường sau:

- Tên công ty tuyển dụng
- Mô tả công việc
- Yêu cầu ứng viên
- Quyền lợi
- Cách thức ứng tuyển

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  $25 \times 10 = 250$  bản ghi từ 10 trang đã thu thập.

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

**Mô tả công việc**

- Plan, organize and develop user-facing features for the components in our dynamic platform;
- Write and optimize client-side code of the web applications, create a fast application with good UI/UX;
- Work with other members to develop and integrate new features including other third-party systems and plugins into our platform;
- Evaluate and identify new technologies for implementation and incorporation;
- Communicate with our business and product heads to understand clients' requirements;
- Respond and follow up to incorporate feedback and draw new insights;
- Prioritize tasks to meet multiple deadlines;
- Identify and correct bottlenecks and fix bugs.

**Yêu cầu ứng viên**

- At least 2 years of development experience with .NET/.NET Core (C#, Web API), NoSQL
- Knowledge of OOP, Dependency Injection, Design Patterns, Programming Principles, Unit test
- Experience in Single Page Application (ReactJs, VueJs...)
- Experience in Html/CSS/JavaScript
- Experience in Cloud service - Microsoft Azure is an advantage
- Good English communication skills.
- Good organization skills and attention to detail.
- Over 2 years of Web/Backend development experience.
- Good communication and teamwork.
- Enthusiastic to provide interesting and useful applications for users
- Passion for programming - web development
- Willing to learn and do applied jobs.

**Quyền lợi**

- 05 working days/week (From Monday to Friday), applying flexible working hours
- 2 days of remote WFH per week (based on the team's decision)
- Lunch + Gasoline + Coffee Allowance
- Health, Social and Unemployment Insurance (based on gross-based salary, according to Labor Code) and PVI Health Insurance
- 13th-month salary and Performance bonus
- Annual salary review
- 12 days annual leave plus extra 02 days company leave
- Company trips, sponsored team building, monthly Happy Hour, Sport Clubs (Soccer, Badminton, Pingpong, Yoga) and other joyful events;
- A culture of relentless learning with free courses in specialized skills, soft skills, and English;
- Yearly health-checkup;
- Seniority benefits: allowance & PVI Health Insurances for family members
- Technical-certificate bonus
- Japanese-certificate bonus
- Employee Referral Incentive

**Cách thức ứng tuyển**

Ứng viên nộp hồ sơ trực tuyến bằng cách bấm **Ứng tuyển ngay** dưới đây.

**ỨNG TUYỂN NGAY** **LƯU TIN**

```
{
  "name": "CÔNG TY CỔ PHẦN DỊCH VỤ VÀ PHÁT TRIỂN CÔNG NGHỆ BEAE VIỆT NAM",
  "Mô tả công việc": "- Sử dụng live chat để trả lời các câu hỏi của khách hàng liên quan tới sản phẩm",
  "Yêu cầu ứng viên": "- Trình độ tiếng anh tối thiểu IELTS 6.5- Hiểu về eCommerce (ưu tiên Shopify)",
  "Quyền lợi": "- Thu nhập up to 30 triệu, được chia cổ phần % dự án nếu có đóng góp lớn cho sản phẩm",
  "Cách thức ứng tuyển": "Ứng viên nộp hồ sơ trực tuyến bằng cách bấm Ứng tuyển ngay dưới đây. ỨNG TUYỂN NGAY",
},
{
  "name": "Công ty cổ phần đầu tư GEMS Việt Nam",
  "Mô tả công việc": "Hiện tại công ty Cổ Phần Đầu Tư và Công Nghệ Gems Tech đang cần tuyển dụng PHP Developer",
  "Yêu cầu ứng viên": "- Có kinh nghiệm từ 6 tháng - 1 năm sử dụng một trong các Framework PHP: CakePHP, Laravel, CodeIgniter",
  "Quyền lợi": "- Lương cứng + Thưởng + Phụ cấp: Thu nhập từ 15 - 20 triệu (Deal theo năng lực và kinh nghiệm)",
  "Cách thức ứng tuyển": "Ứng viên nộp hồ sơ trực tuyến bằng cách bấm Ứng tuyển ngay dưới đây. ỨNG TUYỂN NGAY",
},
{
  "name": "Công ty TNHH beework vietnam",
  "Mô tả công việc": "- Công việc đơn giản ai cũng có thể làm được, được đào tạo nhanh từ 3-5 ngày-",
  "Yêu cầu ứng viên": "- Không yêu cầu bằng cấp, kinh nghiệm, chuyên môn, được đào tạo hướng dẫn từ chuyên gia",
  "Quyền lợi": "- Offline: 6.000.000đ/tháng + phụ cấp vé xe + thưởng nhân viên xuất sắc- Online: 5.000.000đ/tháng",
  "Cách thức ứng tuyển": "Ứng viên nộp hồ sơ trực tuyến bằng cách bấm Ứng tuyển ngay dưới đây. ỨNG TUYỂN NGAY",
},
{
  "name": "Công ty Dịch vụ MobiFone Khu vực 6",
  "Mô tả công việc": "Nghiên cứu, đề xuất, tư vấn bộ giải pháp CNTT và đề xuất ý tưởng chuyển đổi số",
  "Yêu cầu ứng viên": "- Tốt nghiệp đại học các chuyên ngành CNTT/Điện tử viễn thông hoặc tương đương",
  "Quyền lợi": "- Tổng thu nhập (Gross): trên 250 triệu/năm, lương cứng từ 10 triệu/tháng.- Tham gia các chương trình phúc lợi khác",
  "Cách thức ứng tuyển": "Ứng viên nộp hồ sơ trực tuyến bằng cách bấm Ứng tuyển ngay dưới đây. ỨNG TUYỂN NGAY",
},
```

### 2.3.2. Lưu dữ liệu vào Hadoop

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

<input type="checkbox"/>	Permission	Owner	Group	Size
<input type="checkbox"/>	-rw-r--r--	hadoopuser	supergroup	221.98 KB
<input type="checkbox"/>	-rw-r--r--	hadoopuser	supergroup	220.82 KB
<input type="checkbox"/>	-rw-r--r--	hadoopuser	supergroup	267.98 KB
<input type="checkbox"/>	-rw-r--r--	hadoopuser	supergroup	280.62 KB
<input type="checkbox"/>	-rw-r--r--	hadoopuser	supergroup	279.8 KB
<input type="checkbox"/>	-rw-r--r--	hadoopuser	supergroup	286.93 KB
<input type="checkbox"/>	-rw-r--r--	hadoopuser	supergroup	273.25 KB
<input type="checkbox"/>	-rw-r--r--	hadoopuser	supergroup	251.85 KB
<input type="checkbox"/>	-rw-r--r--	hadoopuser	supergroup	256.68 KB
<input type="checkbox"/>	-rw-r--r--	hadoopuser	supergroup	294.59 KB
<input type="checkbox"/>	-rw-r--r--	hadoopuser	supergroup	271.88 KB
<input type="checkbox"/>	-rw-r--r--	hadoopuser	supergroup	249.83 KB
<input type="checkbox"/>	-rw-r--r--	hadoopuser	supergroup	275.85 KB
<input type="checkbox"/>	-rw-r--r--	hadoopuser	supergroup	262.82 KB
<input type="checkbox"/>	-rw-r--r--	hadoopuser	supergroup	252.24 KB
<input type="checkbox"/>	-rw-r--r--	hadoopuser	supergroup	243.01 KB
<input type="checkbox"/>	-rw-r--r--	hadoopuser	supergroup	261.39 KB
<input type="checkbox"/>	-rw-r--r--	hadoopuser	supergroup	289.44 KB
<input type="checkbox"/>	-rw-r--r--	hadoopuser	supergroup	247.98 KB
<input type="checkbox"/>	-rw-r--r--	hadoopuser	supergroup	298.32 KB
<input type="checkbox"/>	-rw-r--r--	hadoopuser	supergroup	271.13 KB
<input type="checkbox"/>	-rw-r--r--	hadoopuser	supergroup	270.68 KB

Dữ liệu được lưu trên 2 datanode slave1 và slave2

[Download](#)
[Head the file \(first 32K\)](#)
[Tail the file \(last 32K\)](#)

Block information -- Block 0

Block ID: 1073742354

Block Pool ID: BP-193947846-172.18.0.6-1656340883026

Generation Stamp: 1531

Size: 189494

Availability:

- slave2
- slave1

Close



### 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("name", StringType(), True),
    StructField("Mô tả công việc", StringType(), True),
    StructField("Yêu cầu ứng viên", StringType(), True),
    StructField("Quyền lợi", StringType(), True),
    StructField("Cách thức ứng tuyển", 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 :

- Company Name : tên công ty tuyển dụng.
- FrameworksPlatforms : một mảng gồm tên các frameworks, platforms mà công ty tuyển dụng yêu cầu.
- Languages: một mảng gồm tên các ngôn ngữ lập trình 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.
- Salaries : một mảng gồm các mức lương mà công ty tuyển dụng chi trả.

Các trường thông tin FrameworksPlatforms, Languages, DesignPatterns, Knowledges 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 Knowledges:

```
knowledges = ['game', 'Jira', 'lập đặt', 'interaction design', 'đồ họa', 'DevOps', 'AI', 'async', 'Quality', 'Security', 'Google Drive', 'NFT', 'mạng máy tính', 'Wordpress', 'Machine Learning', 'Consult', 'kiểm thử', 'đánh giá chất lượng', 'networking', 'distributed system', 'UI/UX', 'Windows', 'Uni', 'Jenkins', 'Chatbot', 'quản trị mạng', 'Solidity', 'tester', 'Corel Draw', 'Illustrator', 'Git', 'Black Box', 'Office', 'chạy quảng cáo', 'Unix', 'IT Support', 'Data mining', 'data analys', 'c', 'TCP', 'qa', 'Animate', 'crypto', 'CI/CD', 'Defi', 'frontend', 'sửa chữa', 'SVN', 'phần cứng', 'Powerpoint', 'smart contract', 'Linux', 'SCM', 'backend', 'Marketing', 'XSS', 'Photoshop', 'HT', 'WebSocket', 'thuật toán', 'TestRail', 'CSDL', 'Sketch', 'blockchains', 'multithreading', 'hướn', 'latex', 'Restful', 'Subversion', 'java web', 'Mobile', 'Excel']
```

Đối với trường Salaries 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:

Mức lương	Mảng quy đổi
8 triệu VNĐ	[5]
26-38 triệu VNĐ	[25,30,35]
2000\$	[45]

Mảng các chuỗi được định nghĩa trước dùng để trích xuất thông tin liên quan:

```
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 platforms:

```
@udf(returnType=ArrayType(StringType()))
def extract_framework_platform(mo_ta_cong_viec,yeu_cau_ung_vien):
    return [framework for framework in patterns.framework_platforms if re.search(framework, mo_ta_cong_viec + " " + yeu_cau_ung_vien, re.IGNORECASE)]

@udf(returnType=ArrayType(StringType()))
def extract_language(mo_ta_cong_viec,yeu_cau_ung_vien):
    return [language for language in patterns.languages if re.search(language.replace("+", "\\+").replace("(", "\\(").replace(")", "\\)"), mo_ta_cong_viec + " " + yeu_cau_ung_vien, re.IGNORECASE)]

@udf(returnType=ArrayType(StringType()))
def extract_knowledge(mo_ta_cong_viec,yeu_cau_ung_vien):
    return [knowledge for knowledge in patterns.knowledges if re.search(knowledge, mo_ta_cong_viec + " " + yeu_cau_ung_vien, re.IGNORECASE)]
```

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:

```
extracted_recruit_df=raw_recruit_df.select(raw_recruit_df["name"].alias("CompanyName"),
    udfs.extract_framework_platform("Mô tả công việc","Yêu cầu ứng viên").alias("FrameworkPlatforms"),
    udfs.extract_language("Mô tả công việc","Yêu cầu ứng viên").alias("Languages"),
    udfs.extract_design_pattern("Mô tả công việc","Yêu cầu ứng viên").alias("DesignPatterns"),
    udfs.extract_knowledge("Mô tả công việc","Yêu cầu ứng viên").alias("Knowledges"),
    udfs.normalize_salary("Quyền lợi").alias("Salaries")
)
extracted_recruit_df.cache()
extracted_recruit_df.show(5)
```

Các dòng đầu của dataframe lọc từ dataframe ban đầu:

CompanyName	FrameworkPlatforms	Languages	DesignPatterns	Knowledges	Salaries
CÔNG TY TNHH ZINZ...	[Vue, Laravel]	[Python, PHP, Ruby]	[[]]	[[]]	[0, 5, 10]
CÔNG TY TNHH QU...	[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 số 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', 'Security': 'blockchain_crypto',
    'lắp đặt': 'hardware', 'sửa chữa': 'hardware', 'phần cứng': 'hardware', 'router': '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',
    'Black Box': 'tester', 'tester': 'tester', 'White Box': 'tester', 'Unit Test': 'tester',
    'TestRail': 'tester', 'kiểm thử': 'tester',
    'SVN': 'version_control', 'SCM': 'version_control', 'Git': 'version_control'}
```

Chương trình sử dụng 1 hàm udf để đánh nhãn các string trong cột Knowledge của dataframe extracted\_recruit\_df. Tuy nhiên, để hàm udf tìm được dictionary trong lúc đánh nhãn thì cần phải broadcast dictionary trước.

Ở đây các từ trong dictionary được broadcast và biến thành broadcast variable, là biến mà chỉ được phép đọc giá trị của biến trên mỗi máy, không cho phép sửa đổi giá trị nhằm mục đích đảm bảo cùng giá trị của biến broadcast trên tất cả các node. Khi Spark nhận thấy code cần đến broadcast variable, nó sẽ gửi dữ liệu này đến các executor cần sử dụng và lưu tại bộ đệm ở phía các executor đó. Điều này sẽ giúp giảm chi phí truyền tải dữ liệu.

Hàm broadcast nhận và udf để map các string trong cột Knowledge của dataframe extracted\_recruit\_df:

```
def broadcast_labeled_knowledges(sc, labeled_knowledges):  
    """  
    broadcast the mapped of labeled_knowledges to group data in knowledge field  
    """  
    global mapped_knowledge  
    mapped_knowledge = sc.broadcast(labeled_knowledges)  
  
    @udf(returnType=StringType())  
    def labeling_knowledge(knowledge):  
        try :  
            return mapped_knowledge.value[knowledge]  
        except :  
            return None
```

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. Thư viện có thể tải về từ Maven Repository dưới dạng file jar (ví dụ elasticsearch-hadoop-7.17.5.jar ).

Sau khi upload folder src và file elasticsearch-hadoop-7.17.5.jar lên spark-master, chương trình có thể thực thi bằng spark-submit như sau:

```
./bin/spark-submit --master spark://master:7077 --jars elasticsearch-hadoop-7.17.5.jar --driver-class-path elasticsearch-hadoop-7.17.5.jar src/main.py
```

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



The screenshot shows the Spark Master web interface at spark://master:7077. It displays cluster statistics and a list of running and completed applications.

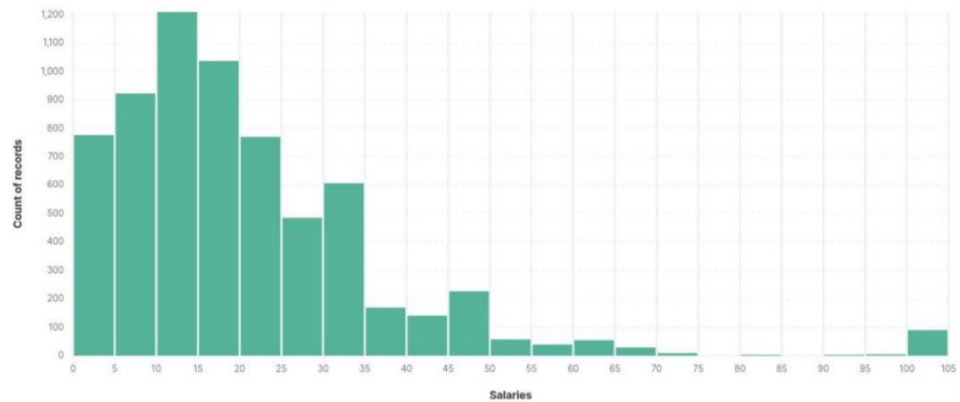
Spark Master at spark://master:7077						
URL: spark://master:7077						
Active Workers: 1						
Cores in use: 0 Total, 0 Used						
Memory in use: 0.0 GB Total, 0.0 GB Used						
Resources in use:						
Applications: 1 Running, 1 Completed						
Drivers: 0 Running, 0 Completed						
Status: ALIVE						
<b>Workers (1)</b>						
Worker ID	Address	State	Cores	Memory	Resources	
worker-20220726141747-292.168.1.24-33189	192.168.1.24:33189	ALIVE	8 (8 Used)	5.6 GB (1024.0 MB Used)		
<b>Running Applications (1)</b>						
Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User
app-20220726143134-0001	PreprocessData	8	1024.0 MB		2022/07/26 14:31:34	hadoopuser
<b>Completed Applications (1)</b>						
Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User
app-20220726143033-0000	PreprocessData	8	1024.0 MB		2022/07/26 14:30:33	hadoopuser

### 2.3.4. Biểu diễn dữ liệu bằng Kibana

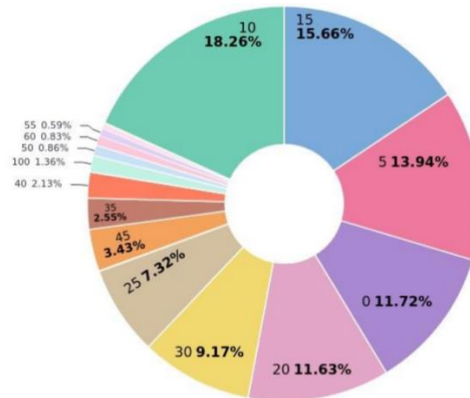
Dữ liệu lưu tại Elasticsearch sẽ được dùng Kibana để biểu diễn

Ví dụ:

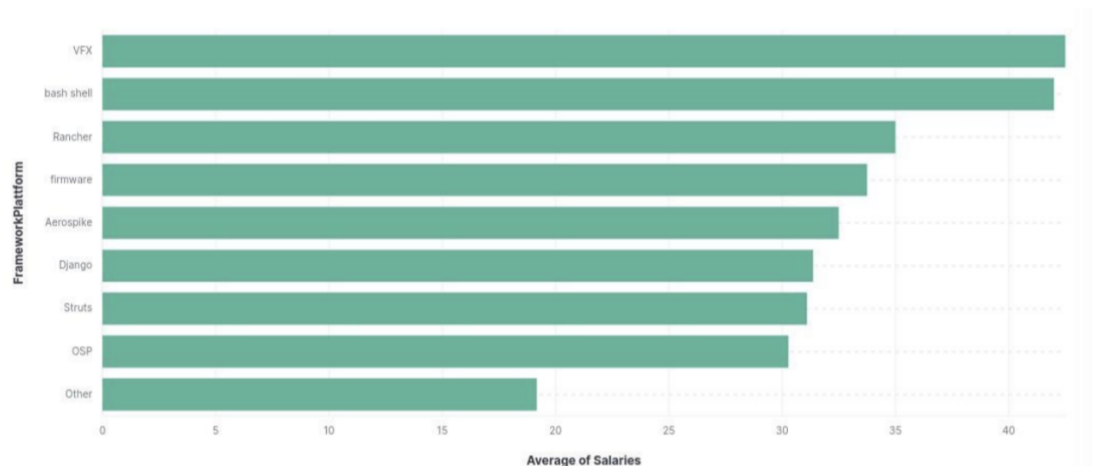
1. Thống kê mức lương:



2. Phân bố khoảng lương:

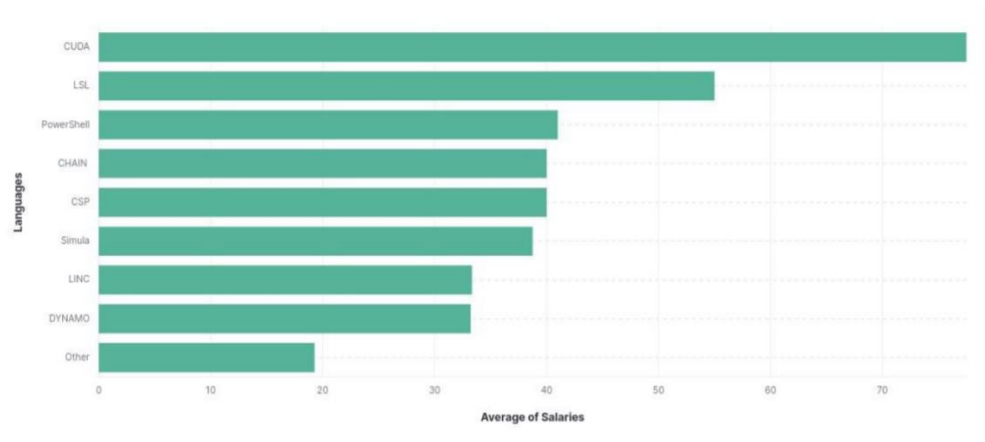


3. Trung bình mức lương đối với các framework:

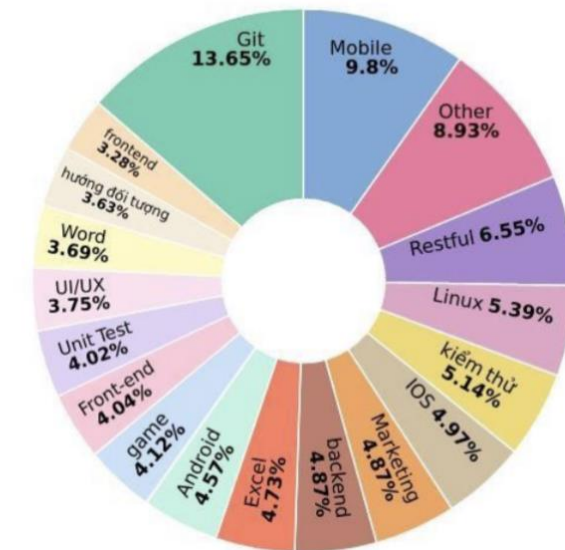




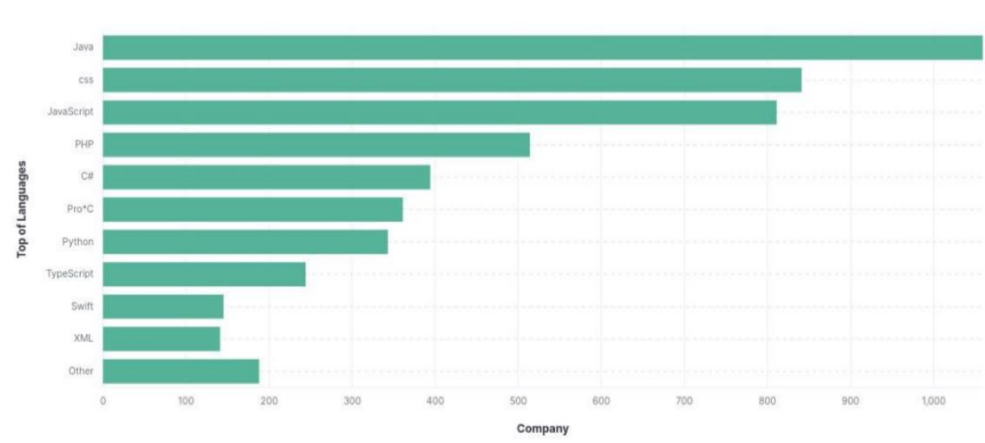
4. Trung bình mức lương đối với các ngôn ngữ lập trình:



5. Tỷ lệ % các lĩnh vực tuyển dụng:



6. Ngôn ngữ lập trình được tuyển dụng nhiều nhất:



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

### 3.1. Nhận xét, đánh giá

Hệ thống cho thấy những lợi ích mà một hệ thống BigData 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 của nhóm không khai thác được tối đa hệ thống. Lượng dữ liệu được thu thập khá ít, hoàn toàn 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 thủ công mà chưa được tự động hóa.

### 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.

Sử dụng Spark Streaming để phân tích và cải thiện tốc độ ghi dữ liệu.

## 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](https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html)
5. <https://viblo.asia/p/tim-hieu-ve-hadoop-bJzKmOBXI9N>
6. <https://viblo.asia/p/tim-hieu-ve-apache-spark-ByEZkQQW5Q0>
7. <https://www.youtube.com/watch?v=maf2-CVYnA>
8. <https://www.youtube.com/watch?v=hRtInGQhBxs&list=PLJIKGwy-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)