TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

──────── \* ───────

**TÌM KIẾM THÔNG TIN VÀ TRÌNH DIỄN THÔNG TIN**



**BÁO CÁO BÀI TẬP LỚN**

**Nhóm sinh viên:**

Phạm Minh Hiếu 20151362

Đào Duy Hòa 20151589

Phạm Huy Hoàng 20151556

Đặng Văn Hà 20151137

Giảng viên hướng dẫn: **TS. NGUYỄN BÁ NGỌC**

***Hà Nội, tháng 5 năm 2019***

**LỜI NÓI ĐẦU**

*Tìm kiếm và trình diễn thông tin là một trong những lĩnh vực rất quan trọng trong Hệ thống thông tin. Tìm kiếm và trình diễn thông tin là thực hiện việc thu thập các tài nguyên đến từ các nguồn khác nhau, sau đó triển khai hệ thống tìm kiếm dựa trên dữ liệu thu thập về.*

*Tìm kiếm và trình diễn thông tin được rất nhiều các công ty, các tập đoàn lớn trên thế giới và cả ở Việt Nam đầu tư và phát triển, như GOOGLE, Cốc Cốc, ...*

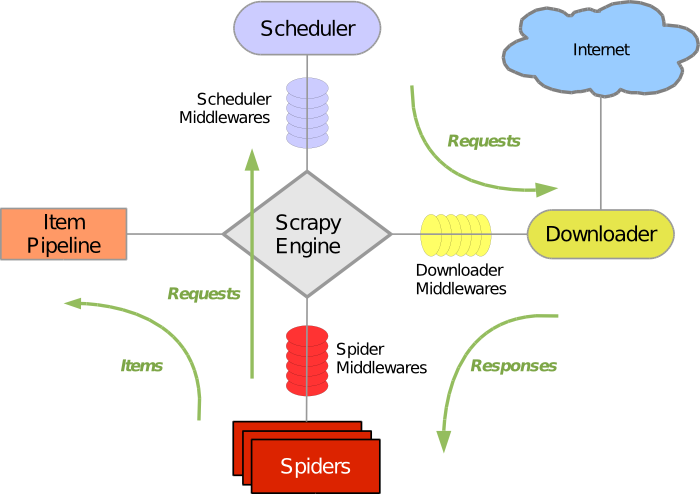
*Để tìm hiểu về Tìm kiếm và trình diễn thông tin, cũng như hoàn thành bài tập lớn của môn học này, nhóm em đã xây dựng một hệ tìm kiếm và trình diễn thông tin, dưới sự hướng dẫn của thầy Nguyễn Bá Ngọc. Sau đây là phần trình bày của nhóm em.*

*Chúng em xin chân thành cảm ơn thầy.*

**I. GIỚI THIỆU CÁC NỀN TẢNG**

**1.1. Giới thiệu Scrapy**

Mô hình cơ bản của 1 scrapy project bao gồm:



* **Scrapy engine:**

Chịu tránh nhiệm điều khiển các thành phần khác và kích hoạt sự kiện khi có một hành động cụ thể xảy ra

* **Scheduler:**

Nhận requests từ Scrapy engine và đẩy vào queue và gọi chúng lại khi mà engine cần request chúng.

* **Download:**

Download trang web về theo requests nhận được chứa trong Scheduler.

* **Spiders:**

Phần tích các thành phần trong trang web được tải về từ phần Download có thể tiếp tục gửi requests cho engine hoặc kết thúc bằng việc lấy được các items.

* **Item pipeline:**

Xửa lí dữ liệu sau khi đã được Spiders trích xuất ra (như lưu trữ trong cơ sở dữ liệu hay in ra file).

**Luồng dữ liệu trong Scrapy (The data flow)**

1. Engine mở một tên miền, chỉ định Spider xử lí tên miền này và hỏi Spider URL sẽ dùng để lấy dữ liệu
2. Engine lấy URL đầu tiên từ Spider và đặt lịch trình cho chúng tại Scheduler
3. Engine hỏi Scheduler URL tiếp theo để lấy dữ liệu
4. Scheduler trả về URL tiếp theo và Engine sends URL này tới Download
5. Tải trang được trang về qua Download và gửi về Engine
6. Engine nhận dữ liệu tải về và đẩy vào Spider để xử lí
7. Spider xử lí và gửi tài nguyên có được vào Item Pipeline hoặc gửi yêu cầu tiếp tới Engine
8. Tiếp tục lại bước 2 nếu Spider gửi tiếp dữ liệu

**1.2 ElasticSearch**

***1.2.1 ElasticSearch là gì***

ElasticSearch(ES) là một document oriented database. Nhiệm vụ của nó chính là store và retrieve document. Trong ES, tất cả các document được hiển thị trong JSON format. Nó được xây dựng trên Lucene – phần mềm tìm kiếm và trả về thông tin  (information retrieval software) với hơn 15 năm kinh nghiệm về full text indexing and searching.

ElasticSearch nên được sử dụng để:

* Tìm kiếm text thông thường
* Tìm kiếm text và dữ liệu có cấu trúc
* Tích hợp dữ liệu
* Tìm kiếm theo tọa độ (Geo Search)
* Lưu trữ dữ liệu dạng JSON

***1.2.2 Các khái niệm cơ bản trong ElasticSearch***

**Cluster**: Một tập hợp Nodes (servers) chứa tất cả các dữ liệu.

**Node**: Một server duy nhất chứa một số dữ liệu và tham gia vào cluster’s indexing and querying.

**Index**: Hãy quên SQL Indexes đi. Mỗi ES Index là 1 tập hợp các documents.

**Shards**: Tập con các documents của 1 Index. Một Index có thể được chia thành nhiều shard.

**Type**: Một định nghĩa về schema of a Document bên trong một Index (Index có thể có nhiều type).

**Document**: Một JSON object với một số dữ liệu. Đây là basic information unit trong ES.

***1.2.3 Indexing và Searching***

ElasticSearch sử dụng một REST API cho việc tìm kiếm và lưu trữ document. Dưới đây là một ví dụ về indexing (storing) a document:

$ curl -XPUT 'http://localhost:9200/blog/post/1' -d '{

"author": "lucas",  
   "tags": ["java", "web"],  
   "title": "A Fancy Title",  
   "context": "A nice post content..."

}'

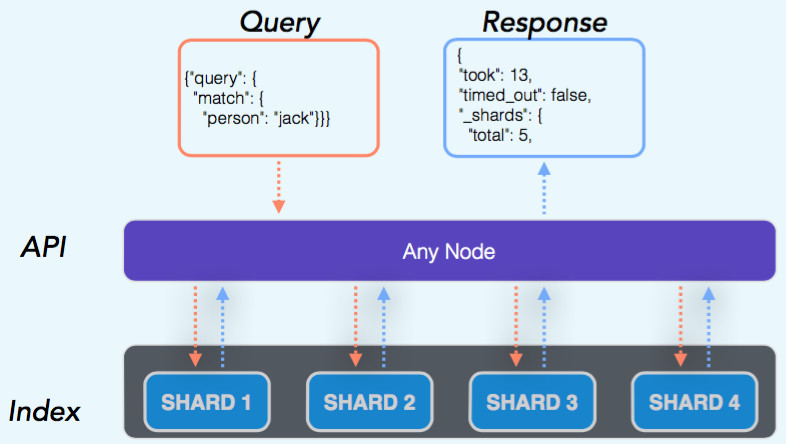
Với câu lệnh này, chúng ta đang tạo ra một blog post trong index "blog" với một type "post"  và một "id=1"

Dưới đây là request để lấy về một document:

$ curl -XGET 'http://localhost:9200/blog/post/\_search' -d '{  
    "query" : {  
        "term" : { "author" : "lucas" }  
    }  
}'

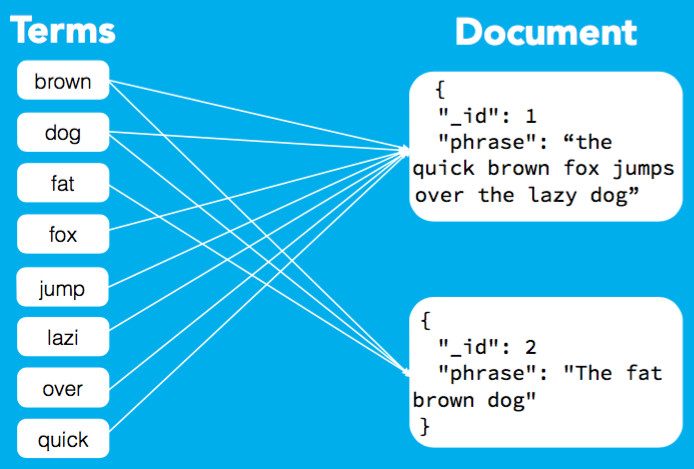
Chúng ta đang tìm kiếm bất kỳ document mà giá trị của trường "author" có từ khóa "lucas".

**Cách mà ElasticSearch thực hiện truy vấn:**



Khi gửi một query đến ES cluster, query sẽ nhận một trong các node và sau đó nó xác định shard nào nên được truy vấn. Sau đó, **“query coordinator” node**gửi query cho mỗi shard để thực hiện truy vấn song song. Sau khi lần lượt từng node trả lời các truy vấn với các kết quả từng phần, node sẽ merge kết quả và gửi trở lại cho user.

**Index:**

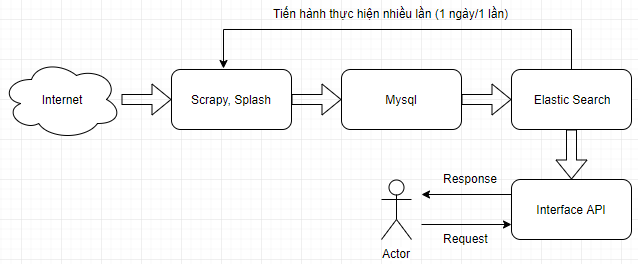
****

Mỗi document gửi tới ES được lưu trữ qua một thuật toán và sau đó được gửi đến shard. ES cố gắng để phân tán document thông qua các shard.

Khi lưu trữ document, ES tạo ra inverted index, map các thuật ngữ/từ khóa xuất hiện trong document này tới chính document đó.

Khi sử dụng inverted index, nó có thể tìm kiếm thông qua terms như một binary tree (sử dụng thứ tự chữ cái) làm giảm thời gian tìm kiếm.

**II. MÔ HÌNH TRIỂN KHAI**

**2.1. Mô hình của search engine được thực hiện**

*Sơ đồ hoạt động của search engine*

Mô hình của search engine gồm các phần sau:

1. **Scrapy, splash:** Crawl dữ liệu của các bài báo trên internet (2 báo sẽ được lấy là Dân trí và kenh14). Sau đó lưu tiến hành tiền xử lí chuẩn hóa dữ liệu crawl về.
   * Kiểm tra trùng lặp với dữ liệu cũ bằng Jaccard với các short\_content của các bài báo cũ trong MySQL. (với n\_gram được chọn là 2 và 3. Ngưỡng của jaccard lớn nhất sẽ là không vượt quá 0.5). Nếu bài báo bị Jaccard phát hiện trùng lặp sẽ bị loại bỏ không được chuyển sang bước đẩy vào MySQL
   * Bước này sẽ được thực hiện nhiều lần bằng cách dùng crontab trong linux (giúp hẹn giờ gọi trong ngày)
2. **MySQL:** đẩy dữ liệu đã crawl và chuẩn hóa lên MySQL
3. **Elastic Search**: Đẩy dữ liệu lưu trong MySQL vào trong elastic search

* Thực hiện tách từ và tách truy vấn để tăng độ phủ cho kết quả được trả về

1. **Interface API:** Chờ nhận tín hiệu của người dùng. Gọi đến elastic search với query được ghép với yêu cầu người dùng. Lấy dữ liệu và trả lại lên giao diện ở máy người dùng

* Có giao diện nâng cao giúp người dùng lọc các trường về thời gian
* Phân trang khi dữ liệu trả về quá lớn

**III. CHƯƠNG TRÌNH DEMO**

**3.1. Các công nghệ sử dụng**

* **PHP:** Xây dựng interfaceAPI
* **Scrapy, Splash:** crawl dữ liệu
* **MySQL:** Lữu dữ liệu crawl được
* **Elastic search:** Máy tìm kiếm

**3.2. Hình ảnh chương trình demo**