**THÔNG TIN NHÓM**

|  |  |  |
| --- | --- | --- |
| **Tên đề tài** | **KAFKA VỚI TWITTER VÀ SPARK** | **Đóng góp của thành viên** |
| **Họ và tên thành viên 1** | Trần Nguyễn Gia Long  - 1050080059 | Đóng góp: Cài đặt thực nghiệm trên IntelliJ IDEA , Spark , Kafka Điểm tự đánh giá quá trình: 8 Điểm tự đánh giá cuối kì: 8 |
| **Họ và tên thành viên 2** | Trần Kim Thiện  - 1050080076 | Đóng góp: Cài đặt thực nghiệm trên IntelliJ IDEA , Kafka , Spark  Điểm tự đánh giá quá trình: 7 Điểm tự đánh giá cuối kì: 7 |
| **Họ và Tên thành viên 3** | Nguyễn Thanh Danh  - 1050080044 | Đóng góp: Cài đặt thực nghiệm Spark, làm word Điểm tự đánh giá quá trình: 6 Điểm tự đánh giá cuối kì: 6 |
| **Họ và Tên thành viên 4** | Nguyễn Dương Đức Nguyên  - 1050080066 | Đóng góp: Cài đặt thực nghiệm Kafka, làm ppt Điểm tự đánh giá quá trình: 7 Điểm tự đánh giá cuối kì: 7 |

**BÁO CÁO TÓM TẮT**

• Lý do chọn đề tài:  
Trong thời đại ngày nay, dữ liệu được tạo ra với tốc độ chóng mặt từ nhiều nguồn khác nhau, bao gồm các trang web, ứng dụng di động, thiết bị IoT và các phương tiện truyền thông xã hội. Để xử lý lượng dữ liệu khổng lồ này, các doanh nghiệp cần sử dụng các công nghệ hiện đại như Kafka, Twitter và Spark.

Kafka là một hệ thống phân phối tin nhắn được sử dụng để lưu trữ và xử lý dữ liệu từ nhiều nguồn khác nhau. Twitter là một nền tảng truyền thông xã hội cho phép người dùng chia sẻ tin nhắn, hình ảnh và video. Spark là một nền tảng phân tích dữ liệu lớn được sử dụng để xử lý dữ liệu nhanh chóng và hiệu quả.

Kafka, Twitter và Spark là ba công nghệ quan trọng trong lĩnh vực xử lý dữ liệu lớn. Nghiên cứu về các công nghệ này có thể giúp chúng ta hiểu rõ hơn về cách thức hoạt động của chúng và cách sử dụng chúng hiệu quả hơn.

Các doanh nghiệp có thể sử dụng các công nghệ này để thu thập, lưu trữ và xử lý dữ liệu từ các nguồn khác nhau, chẳng hạn như dữ liệu bán hàng, dữ liệu khách hàng và dữ liệu truyền thông xã hội.

Đề tài này có tính thách thức cao. Để nghiên cứu về các công nghệ này, chúng ta cần có kiến thức nền tảng về khoa học máy tính, mạng máy tính và phân tích dữ liệu.  
• Tầm quan trọng:  
Giúp các doanh nghiệp thu thập, lưu trữ và xử lý dữ liệu từ các nguồn khác nhau.

Cung cấp các giải pháp phân tích dữ liệu lớn hiệu quả và tiết kiệm chi phí.

Thúc đẩy sự phát triển của các ứng dụng và dịch vụ mới dựa trên dữ liệu.

Các nội dung và kết quả  
• Mô tả bài toán  
• Giải pháp, mô hình  
• Các công nghệ sử dụng  
• Cài đặt Spark và Kafka   
• Demo  
• Đánh giá

Báo cáo chi tiết

1. Giới thiệu về Big Data

**1.1 ) Định nghĩa về Big Data**

Big Data là một tập dữ liệu lớn và phức tạp, đến mức các công cụ, ứng dụng xử lý truyền thống không thể tiếp nhận được.

Big Data là một kho báu ẩn giấu: Big Data chứa đựng rất nhiều thông tin giá trị, có thể được sử dụng để giải quyết nhiều vấn đề khác nhau. Tuy nhiên, để khai thác được giá trị của Big Data, cần có các kỹ thuật và công cụ phân tích phù hợp.

Big Data là một thách thức: Big Data có thể rất phức tạp và khó xử lý. Để xử lý Big Data hiệu quả, cần có các kiến thức và kỹ năng chuyên sâu.

Big Data là một cơ hội: Big Data có thể giúp các doanh nghiệp, tổ chức cải thiện hiệu quả hoạt động, đưa ra các quyết định kinh doanh tốt hơn và phục vụ khách hàng tốt hơn.

Big Data là một công nghệ không thể thiếu trong thời đại hiện nay: Big Data đang được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau, từ kinh doanh, y tế, giáo dục đến chính phủ.

Big Data là một công nghệ có thể gây ra những tác động tiêu cực nếu không được sử dụng một cách cẩn thận: Big Data có thể được sử dụng để xâm phạm quyền riêng tư, phân biệt đối xử hoặc tạo ra những thông tin sai lệch.

Để hiểu rõ hơn về định nghĩa này, chúng ta cần hiểu rõ các đặc điểm của Big Data, bao gồm:

Khối lượng lớn: Big Data thường có kích thước rất lớn, có thể lên đến hàng terabyte, petabyte hoặc thậm chí exabyte.

Tốc độ nhanh: Big Data được tạo ra với tốc độ rất nhanh, do đó cần phải được xử lý nhanh chóng để tránh bị quá tải.

Phức tạp: Big Data có thể bao gồm nhiều loại dữ liệu khác nhau, có thể là dữ liệu có cấu trúc, dữ liệu không có cấu trúc hoặc dữ liệu bán cấu trúc.

**Các đặc điểm của Big Data**

Khối lượng lớn: Big Data thường được đo bằng terabyte (TB), petabyte (PB), hoặc thậm chí exabyte (EB). Ví dụ, một ngày, thế giới tạo ra khoảng 2,5 quintillion byte dữ liệu.

Tốc độ nhanh: Big Data được tạo ra với tốc độ rất nhanh, do đó cần phải được xử lý nhanh chóng để tránh bị quá tải. Ví dụ, một phút trên mạng xã hội Twitter, có khoảng 600.000 tweet được tạo ra.

Phức tạp: Big Data có thể bao gồm nhiều loại dữ liệu khác nhau, có thể là dữ liệu có cấu trúc, dữ liệu không có cấu trúc hoặc dữ liệu bán cấu trúc.

Dữ liệu có cấu trúc: Dữ liệu có cấu trúc là dữ liệu được tổ chức theo một mô hình cụ thể, chẳng hạn như dữ liệu trong bảng tính hoặc cơ sở dữ liệu.

Dữ liệu không có cấu trúc: Dữ liệu không có cấu trúc là dữ liệu không được tổ chức theo một mô hình cụ thể, chẳng hạn như dữ liệu từ các trang web, mạng xã hội hoặc thiết bị di động.

Dữ liệu bán cấu trúc: Dữ liệu bán cấu trúc là dữ liệu có một số cấu trúc, chẳng hạn như dữ liệu từ các tệp XML hoặc JSON.

**Vai trò của Big Data**

Big Data có thể được sử dụng để giải quyết nhiều vấn đề khác nhau, bao gồm:

Tối ưu hóa hoạt động kinh doanh: Big Data có thể được sử dụng để phân tích dữ liệu khách hàng, dữ liệu bán hàng, dữ liệu sản xuất,... để giúp các doanh nghiệp đưa ra các quyết định kinh doanh tốt hơn.

Nâng cao trải nghiệm khách hàng: Big Data có thể được sử dụng để hiểu hành vi của khách hàng, từ đó giúp các doanh nghiệp cung cấp các sản phẩm và dịch vụ phù hợp hơn với nhu cầu của khách hàng.

Cải thiện hiệu quả hoạt động: Big Data có thể được sử dụng để tự động hóa các quy trình, từ đó giúp các doanh nghiệp tiết kiệm thời gian và chi phí.

**Ứng dụng của Big Data**

Big Data đang được ứng dụng trong nhiều lĩnh vực khác nhau, bao gồm:

Doanh nghiệp: Big Data được sử dụng trong nhiều lĩnh vực khác nhau của doanh nghiệp, bao gồm tiếp thị, bán hàng, sản xuất,...

Chính phủ: Big Data được sử dụng để phân tích dữ liệu dân số, dữ liệu kinh tế,... để giúp chính phủ đưa ra các quyết định chính sách tốt hơn.

Y tế: Big Data được sử dụng để phân tích dữ liệu y tế, từ đó giúp cải thiện chất lượng chăm sóc sức khỏe.

Giáo dục: Big Data được sử dụng để phân tích dữ liệu học tập, từ đó giúp cải thiện chất lượng giáo dục.

**Một số nhận định về Big Data của các chuyên gia**

Theo Gartner: "Big Data là một trong ba xu hướng công nghệ quan trọng nhất trong năm 2023 . Big Data là dữ liệu có khối lượng lớn, tốc độ nhanh và phức tạp đến mức các công cụ và kỹ thuật xử lý dữ liệu truyền thống không thể xử lý hiệu quả."

Theo The Economist: "Big Data là dữ liệu quá lớn để có thể lưu trữ và xử lý trong một cơ sở dữ liệu truyền thống."

Theo Jeff Deaver, nhà phân tích của Forrester Research: "Big Data là một cuộc cách mạng về công nghệ, tương đương với sự ra đời của máy tính điện tử."

Theo Don Tapscott, nhà tư tưởng và nhà lãnh đạo doanh nghiệp: "Big Data là một công cụ mạnh mẽ có thể được sử dụng để cải thiện mọi thứ, từ cách chúng ta chăm sóc sức khỏe đến cách chúng ta kinh doanh."

Theo Peter Drucker, nhà tư tưởng quản trị: "Thông tin là tài nguyên quan trọng nhất trong thế kỷ 21."

Theo World Economic Forum: "Big Data là một trong những công nghệ có khả năng tác động lớn nhất đến thế giới trong những năm tới."

Theo McKinsey Global Institute: "Big Data là dữ liệu có khối lượng lớn, tốc độ nhanh và đa dạng đến mức các công cụ và kỹ thuật xử lý dữ liệu truyền thống không thể xử lý hiệu quả.

Big Data có thể tạo ra giá trị kinh tế trị giá 3,3 nghìn tỷ USD cho nền kinh tế toàn cầu vào năm 2025."

Theo IBM: "Big Data là tập hợp dữ liệu quá lớn, quá phức tạp hoặc quá nhanh để có thể xử lý bằng các công cụ và kỹ thuật xử lý dữ liệu truyền thống."

**1.2) Đặc trưng của Big data**

**3V của Big Data**

3V của Big Data là một mô hình mô tả 3 đặc điểm chính của Big Data, bao gồm:

* Volume (Khối lượng): Big Data thường có kích thước rất lớn, có thể lên đến hàng terabyte, petabyte hoặc thậm chí exabyte.
* Velocity (Tốc độ): Big Data được tạo ra với tốc độ rất nhanh, do đó cần phải được xử lý nhanh chóng để tránh bị quá tải.
* Variety (Phức tạp): Big Data có thể bao gồm nhiều loại dữ liệu khác nhau, có thể là dữ liệu có cấu trúc, dữ liệu không có cấu trúc hoặc dữ liệu bán cấu trúc.

**4V của Big Data**

4V của Big Data là một mô hình mở rộng của 3V, bao gồm thêm 1 đặc điểm sau:

* Veracity (Độ chính xác): Big Data có thể có độ chính xác cao hoặc thấp, tùy thuộc vào nguồn dữ liệu.

**5V của Big Data**

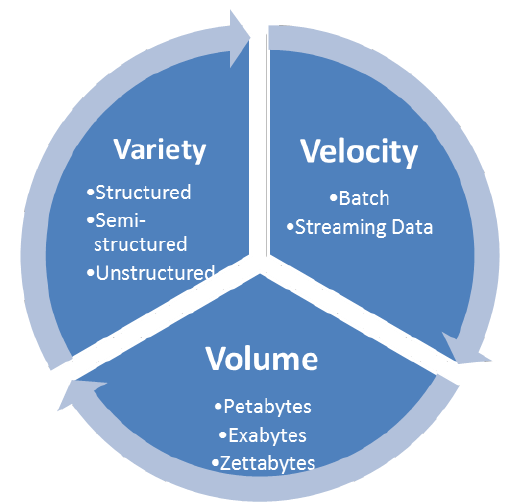
5V của Big Data là một mô hình mở rộng của 4V, bao gồm thêm 1 đặc điểm sau:

* Value (Giá trị): Big Data có thể được sử dụng để giải quyết nhiều vấn đề khác nhau, mang lại nhiều lợi ích cho các doanh nghiệp, tổ chức và xã hội.

**Sự khác biệt giữa 3V, 4V và 5V**

Sự khác biệt chính giữa 3V, 4V và 5V là số lượng đặc trưng mà chúng mô tả. 3V là mô hình cơ bản nhất, chỉ mô tả 3 đặc trưng của Big Data. 4V bổ sung thêm 1 đặc trưng về độ chính xác, và 5V bổ sung thêm 1 đặc trưng về giá trị.

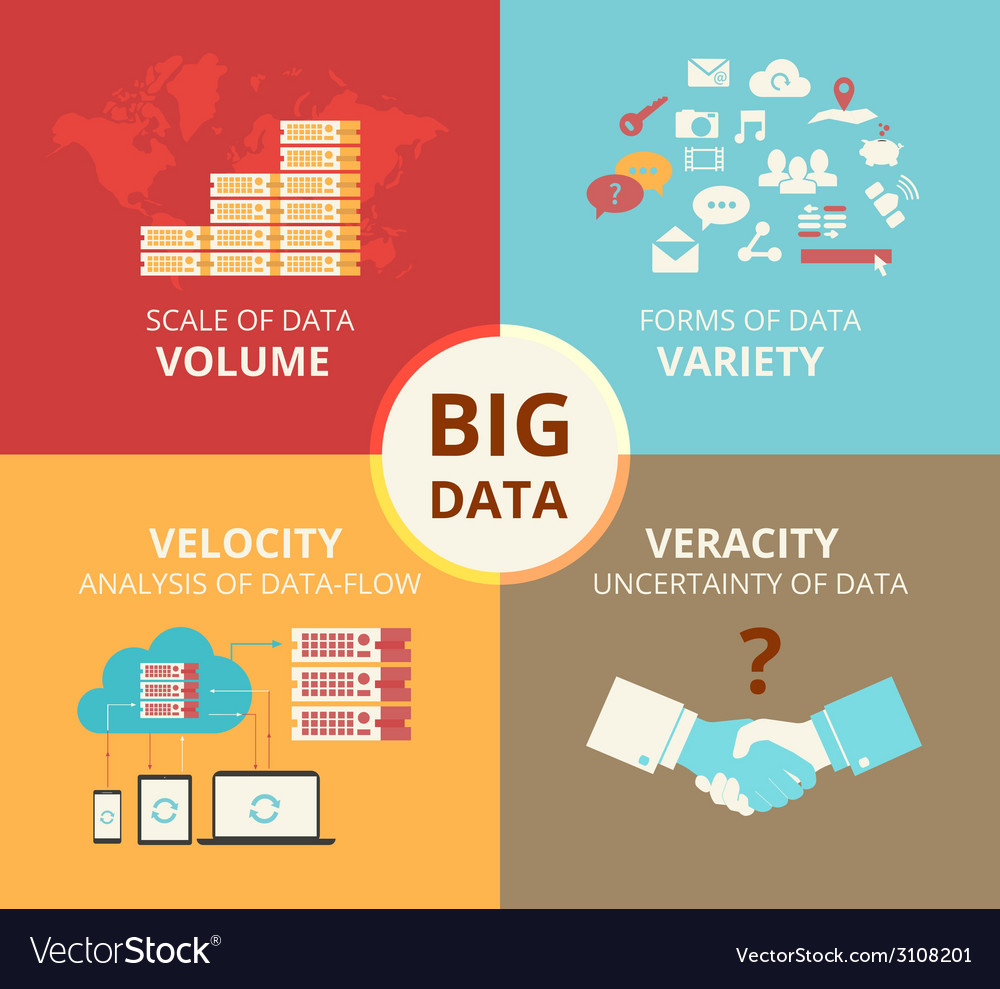
Mô hình 3V



**Mô hình 3V** là mô hình phổ biến nhất và được nhiều người sử dụng. Mô hình này tập trung vào các đặc trưng cơ bản nhất của Big Data, đó là khối lượng, tốc độ và phức tạp.

* Khối lượng: Khối lượng là đặc trưng cơ bản nhất của Big Data. Big Data thường có kích thước rất lớn, có thể lên đến hàng terabyte, petabyte hoặc thậm chí exabyte. Khối lượng lớn của Big Data khiến việc thu thập, lưu trữ và xử lý trở thành một thách thức.
* Tốc độ: Big Data được tạo ra với tốc độ rất nhanh, có thể là hàng terabyte dữ liệu được tạo ra mỗi giây. Tốc độ nhanh của Big Data khiến việc thu thập, lưu trữ và xử lý trở thành một thách thức.
* Phức tạp: Big Data có thể bao gồm nhiều loại dữ liệu khác nhau, có thể là dữ liệu có cấu trúc, dữ liệu không có cấu trúc hoặc dữ liệu bán cấu trúc. Sự phức tạp của Big Data khiến việc phân tích trở nên khó khăn hơn.

Mô hình 4V



**Mô hình 4V** là một mô hình mở rộng của 3V, bổ sung thêm 1 đặc trưng về độ chính xác. Đặc trưng này nhấn mạnh tầm quan trọng của việc đánh giá độ chính xác của Big Data.

**4V gồm :**

Volume (Khối lượng)

Khối lượng là đặc trưng cơ bản nhất của Big Data. Big Data thường có kích thước rất lớn, có thể lên đến hàng terabyte, petabyte hoặc thậm chí exabyte. Khối lượng lớn của Big Data khiến việc thu thập, lưu trữ và xử lý trở thành một thách thức.

* Thu thập: Thu thập Big Data có thể là một thách thức, vì cần phải có các công nghệ và tài nguyên phù hợp.
* Lưu trữ: Lưu trữ Big Data cũng có thể là một thách thức, vì cần phải có các cơ sở hạ tầng lưu trữ có khả năng mở rộng.
* Xử lý: Xử lý Big Data có thể là một thách thức, vì cần phải có các thuật toán và công cụ phân tích phù hợp.

Velocity (Tốc độ)

Big Data được tạo ra với tốc độ rất nhanh, có thể là hàng terabyte dữ liệu được tạo ra mỗi giây. Tốc độ nhanh của Big Data khiến việc thu thập, lưu trữ và xử lý trở thành một thách thức.

* Thu thập: Thu thập Big Data theo thời gian thực có thể là một thách thức, vì cần phải có các công nghệ và tài nguyên phù hợp.
* Lưu trữ: Lưu trữ Big Data theo thời gian thực cũng có thể là một thách thức, vì cần phải có các cơ sở hạ tầng lưu trữ có khả năng mở rộng.
* Xử lý: Xử lý Big Data theo thời gian thực có thể là một thách thức, vì cần phải có các thuật toán và công cụ phân tích phù hợp.

Variety (Phức tạp)

Big Data có thể bao gồm nhiều loại dữ liệu khác nhau, có thể là dữ liệu có cấu trúc, dữ liệu không có cấu trúc hoặc dữ liệu bán cấu trúc. Sự phức tạp của Big Data khiến việc phân tích trở nên khó khăn hơn.

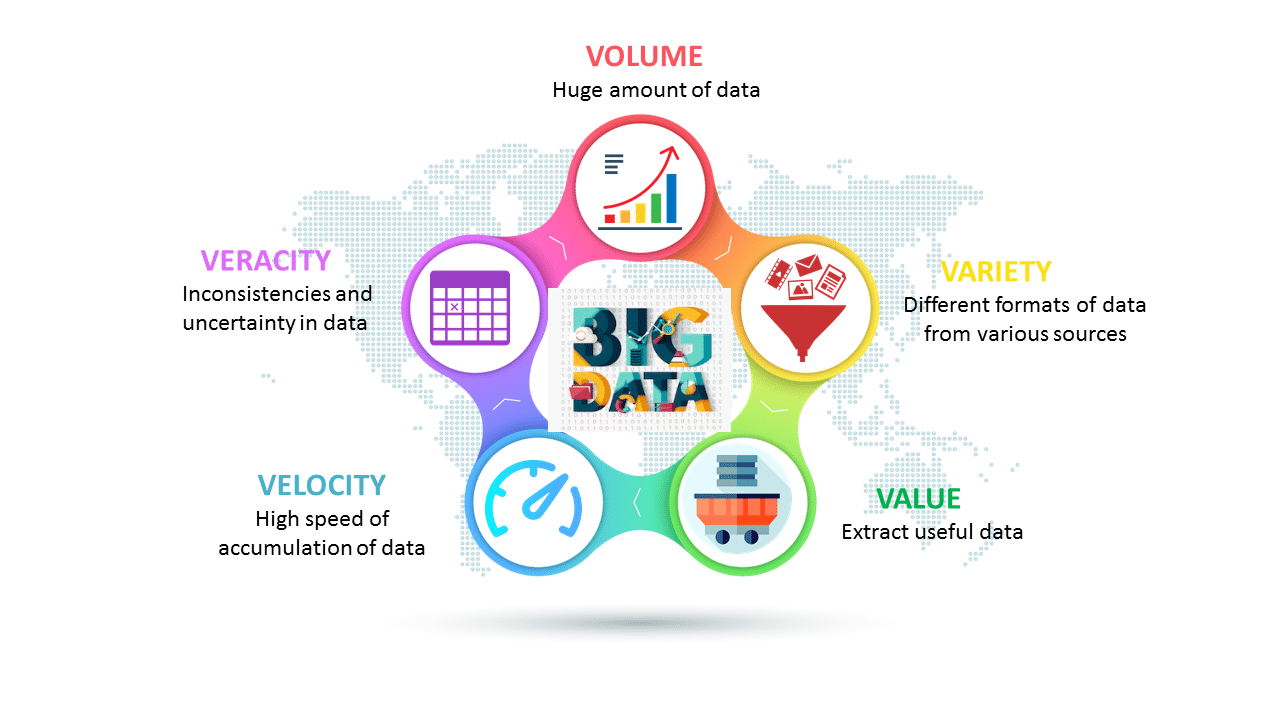
* Dữ liệu có cấu trúc: Dữ liệu có cấu trúc là dữ liệu được tổ chức theo một định dạng nhất định, chẳng hạn như bảng trong cơ sở dữ liệu.
* Dữ liệu không có cấu trúc: Dữ liệu không có cấu trúc là dữ liệu không được tổ chức theo một định dạng nhất định, chẳng hạn như văn bản, hình ảnh hoặc video.
* Dữ liệu bán cấu trúc: Dữ liệu bán cấu trúc là dữ liệu có một số phần được tổ chức theo một định dạng nhất định, chẳng hạn như dữ liệu từ mạng xã hội.

Veracity (Độ chính xác)

Big Data có thể có độ chính xác cao hoặc thấp, tùy thuộc vào nguồn dữ liệu. Độ chính xác của Big Data ảnh hưởng đến chất lượng của phân tích.

* Độ chính xác của dữ liệu: Độ chính xác của dữ liệu phụ thuộc vào nhiều yếu tố, chẳng hạn như phương pháp thu thập dữ liệu, chất lượng của dữ liệu đầu vào và các lỗi tiềm ẩn trong quá trình xử lý dữ liệu.
* Độ chính xác của phân tích: Độ chính xác của phân tích phụ thuộc vào độ chính xác của dữ liệu.

Mô hình 5V



**Mô hình 5V** là một mô hình mở rộng của 4V, bổ sung thêm 1 đặc trưng về giá trị. Đặc trưng này nhấn mạnh tiềm năng to lớn của Big Data trong việc giải quyết nhiều vấn đề khác nhau.

**5V gồm :**

Volume (Khối lượng)

Khối lượng là đặc trưng cơ bản nhất của Big Data. Big Data thường có kích thước rất lớn, có thể lên đến hàng terabyte, petabyte hoặc thậm chí exabyte. Khối lượng lớn của Big Data khiến việc thu thập, lưu trữ và xử lý trở thành một thách thức.

* Thu thập: Thu thập Big Data có thể là một thách thức, vì cần phải có các công nghệ và tài nguyên phù hợp.
* Lưu trữ: Lưu trữ Big Data cũng có thể là một thách thức, vì cần phải có các cơ sở hạ tầng lưu trữ có khả năng mở rộng.
* Xử lý: Xử lý Big Data có thể là một thách thức, vì cần phải có các thuật toán và công cụ phân tích phù hợp.

Velocity (Tốc độ)

Big Data được tạo ra với tốc độ rất nhanh, có thể là hàng terabyte dữ liệu được tạo ra mỗi giây. Tốc độ nhanh của Big Data khiến việc thu thập, lưu trữ và xử lý trở thành một thách thức.

* Thu thập: Thu thập Big Data theo thời gian thực có thể là một thách thức, vì cần phải có các công nghệ và tài nguyên phù hợp.
* Lưu trữ: Lưu trữ Big Data theo thời gian thực cũng có thể là một thách thức, vì cần phải có các cơ sở hạ tầng lưu trữ có khả năng mở rộng.
* Xử lý: Xử lý Big Data theo thời gian thực có thể là một thách thức, vì cần phải có các thuật toán và công cụ phân tích phù hợp.

Variety (Phức tạp)

Big Data có thể bao gồm nhiều loại dữ liệu khác nhau, có thể là dữ liệu có cấu trúc, dữ liệu không có cấu trúc hoặc dữ liệu bán cấu trúc. Sự phức tạp của Big Data khiến việc phân tích trở nên khó khăn hơn.

* Dữ liệu có cấu trúc: Dữ liệu có cấu trúc là dữ liệu được tổ chức theo một định dạng nhất định, chẳng hạn như bảng trong cơ sở dữ liệu.
* Dữ liệu không có cấu trúc: Dữ liệu không có cấu trúc là dữ liệu không được tổ chức theo một định dạng nhất định, chẳng hạn như văn bản, hình ảnh hoặc video.
* Dữ liệu bán cấu trúc: Dữ liệu bán cấu trúc là dữ liệu có một số phần được tổ chức theo một định dạng nhất định, chẳng hạn như dữ liệu từ mạng xã hội.

Veracity (Độ chính xác)

Big Data có thể có độ chính xác cao hoặc thấp, tùy thuộc vào nguồn dữ liệu. Độ chính xác của Big Data ảnh hưởng đến chất lượng của phân tích.

* Độ chính xác của dữ liệu: Độ chính xác của dữ liệu phụ thuộc vào nhiều yếu tố, chẳng hạn như phương pháp thu thập dữ liệu, chất lượng của dữ liệu đầu vào và các lỗi tiềm ẩn trong quá trình xử lý dữ liệu.
* Độ chính xác của phân tích: Độ chính xác của phân tích phụ thuộc vào độ chính xác của dữ liệu.

Value (Giá trị)

Big Data có thể được sử dụng để giải quyết nhiều vấn đề khác nhau, mang lại nhiều lợi ích cho các doanh nghiệp, tổ chức và xã hội.

* Các lĩnh vực ứng dụng Big Data: Big Data có thể được ứng dụng trong nhiều lĩnh vực khác nhau, bao gồm kinh doanh, y tế, giáo dục, chính phủ và khoa học.
* Các lợi ích của Big Data: Big Data có thể mang lại nhiều lợi ích, chẳng hạn như cải thiện hiệu quả, tăng doanh thu, phát triển sản phẩm mới và đưa ra quyết định tốt hơn.

**1.3)Những vấn đề liên quan**

**Phân tích dữ liệu Big Data**

Phân tích dữ liệu Big Data là quá trình sử dụng các kỹ thuật và công cụ để trích xuất thông tin và kiến thức từ Big Data. Phân tích dữ liệu Big Data có thể được sử dụng để giải quyết nhiều vấn đề khác nhau, chẳng hạn như:

* Cải thiện hiệu quả: Phân tích dữ liệu Big Data có thể được sử dụng để xác định các lĩnh vực có thể được cải thiện, chẳng hạn như quy trình, sản phẩm hoặc dịch vụ.
* Tăng doanh thu: Phân tích dữ liệu Big Data có thể được sử dụng để xác định các cơ hội mới để tăng doanh thu, chẳng hạn như mở rộng thị trường hoặc phát triển sản phẩm mới.
* Phát triển sản phẩm mới: Phân tích dữ liệu Big Data có thể được sử dụng để hiểu nhu cầu và mong muốn của khách hàng, từ đó phát triển các sản phẩm và dịch vụ mới đáp ứng nhu cầu đó.
* Đưa ra quyết định tốt hơn: Phân tích dữ liệu Big Data có thể được sử dụng để cung cấp thông tin chi tiết và hỗ trợ cho việc ra quyết định.

**Kỹ thuật và công cụ để phân tích dữ liệu Big Data**

Có nhiều khác nhau có thể được sử dụng. Một số kỹ thuật phổ biến bao gồm:

* Machine learning: Machine learning là một lĩnh vực của trí tuệ nhân tạo cho phép máy móc học hỏi từ dữ liệu mà không cần được lập trình rõ ràng.
* Data mining: Data mining là quá trình khám phá dữ liệu để tìm ra các mẫu và xu hướng ẩn.
* Visualization: Visualization là việc sử dụng hình ảnh và đồ họa để thể hiện dữ liệu một cách trực quan.

**Một số thách thức**

Phân tích dữ liệu Big Data có thể gặp phải một số thách thức, chẳng hạn như:

* Kích thước: Dữ liệu lớn có thể rất lớn, khiến việc thu thập, lưu trữ và xử lý trở thành một thách thức.
* Tốc độ: Dữ liệu lớn có thể được tạo ra với tốc độ rất nhanh, khiến việc thu thập và xử lý theo thời gian thực trở thành một thách thức.
* Phức tạp: Dữ liệu lớn có thể bao gồm nhiều loại dữ liệu khác nhau, khiến việc phân tích trở nên khó khăn hơn.

**Ứng dụng**

Phân tích dữ liệu Big Data có thể được áp dụng trong nhiều lĩnh vực khác nhau, bao gồm:

* Kinh doanh: Phân tích dữ liệu Big Data có thể được sử dụng để cải thiện hiệu quả, tăng doanh thu, phát triển sản phẩm mới và đưa ra quyết định tốt hơn.
* Y tế: Phân tích dữ liệu Big Data có thể được sử dụng để cải thiện kết quả điều trị, phát triển các phương pháp điều trị mới và ngăn ngừa bệnh tật.
* Chính phủ: Phân tích dữ liệu Big Data có thể được sử dụng để cải thiện an ninh, cung cấp các dịch vụ công tốt hơn và đưa ra quyết định chính sách.
* Khoa học: Phân tích dữ liệu Big Data có thể được sử dụng để khám phá các hiện tượng mới, phát triển các mô hình mới và đưa ra các giả thuyết mới.

**Đánh giá của chuyển gia**

Phân tích dữ liệu Big Data là một lĩnh vực đang phát triển nhanh chóng. Các kỹ thuật và công cụ mới được phát triển liên tục, mở ra nhiều khả năng mới cho việc phân tích dữ liệu.

* Theo Gartner: Phân tích dữ liệu Big Data là quá trình thu thập, xử lý và phân tích dữ liệu lớn để tìm ra thông tin và kiến thức có thể được sử dụng để đưa ra quyết định.
* Theo IBM: Phân tích dữ liệu Big Data là quá trình sử dụng các kỹ thuật và công cụ để tìm hiểu về dữ liệu lớn.
* Theo SAS: Phân tích dữ liệu Big Data là quá trình sử dụng các kỹ thuật và công cụ để hiểu dữ liệu lớn và sử dụng dữ liệu đó để đưa ra quyết định.

**Xây dựng cơ sở hạ tầng Big Data**

Xây dựng cơ sở hạ tầng Big Data là quá trình thiết kế, triển khai và quản lý các hệ thống và công nghệ cần thiết để thu thập, lưu trữ, xử lý và phân tích dữ liệu lớn.

**Cơ sở hạ tầng Big Data bao gồm các thành phần sau:**

* Thiết bị phần cứng: Các thiết bị phần cứng cần thiết để lưu trữ và xử lý dữ liệu lớn, chẳng hạn như máy chủ, ổ đĩa lưu trữ, mạng và hệ thống phân tán.
* Phần mềm: Các phần mềm cần thiết để thu thập, lưu trữ, xử lý và phân tích dữ liệu lớn, chẳng hạn như hệ điều hành, hệ quản trị cơ sở dữ liệu, công cụ phân tích và công cụ machine learning.
* Dịch vụ: Các dịch vụ cần thiết để hỗ trợ cơ sở hạ tầng Big Data, chẳng hạn như dịch vụ đám mây, dịch vụ quản lý và dịch vụ bảo mật.

Xây dựng cơ sở hạ tầng Big Data là một quá trình phức tạp và đòi hỏi nhiều nguồn lực.

**Có nhiều yếu tố cần được xem xét khi xây dựng cơ sở hạ tầng Big Data, bao gồm:**

* Kích thước dữ liệu: Cơ sở hạ tầng Big Data cần phải có khả năng lưu trữ và xử lý lượng dữ liệu lớn.
* Tốc độ dữ liệu: Cơ sở hạ tầng Big Data cần phải có khả năng xử lý dữ liệu với tốc độ cao.
* Phức tạp dữ liệu: Cơ sở hạ tầng Big Data cần phải có khả năng xử lý dữ liệu có cấu trúc, không có cấu trúc và bán cấu trúc.
* Chi phí: Chi phí xây dựng và vận hành cơ sở hạ tầng Big Data có thể rất cao.

**Các bước xây dựng cơ sở hạ tầng Big Data thường bao gồm:**

1. Xác định nhu cầu: Bước đầu tiên là xác định nhu cầu dữ liệu của tổ chức. Điều này bao gồm xác định loại dữ liệu cần được thu thập, kích thước dữ liệu, tốc độ dữ liệu và độ phức tạp của dữ liệu.
2. Thiết kế cơ sở hạ tầng: Sau khi xác định nhu cầu, cần thiết kế cơ sở hạ tầng Big Data phù hợp. Điều này bao gồm lựa chọn các thiết bị phần cứng, phần mềm và dịch vụ phù hợp.
3. Triển khai cơ sở hạ tầng: Bước tiếp theo là triển khai cơ sở hạ tầng Big Data. Điều này bao gồm cài đặt thiết bị phần cứng, cài đặt phần mềm và cấu hình hệ thống.
4. Quản lý cơ sở hạ tầng: Sau khi triển khai, cần quản lý cơ sở hạ tầng Big Data. Điều này bao gồm giám sát hiệu suất, bảo mật và bảo trì hệ thống.

Cơ sở hạ tầng Big Data là một yếu tố quan trọng để tổ chức có thể thu thập, lưu trữ, xử lý và phân tích dữ liệu lớn. Việc xây dựng cơ sở hạ tầng Big Data cần được thực hiện một cách cẩn trọng và cân nhắc để đảm bảo đáp ứng nhu cầu của tổ chức.

**Một số lợi ích của việc xây dựng cơ sở hạ tầng Big Data:**

* Cải thiện hiệu quả: Cơ sở hạ tầng Big Data có thể giúp tổ chức cải thiện hiệu quả bằng cách tự động hóa các tác vụ và cung cấp thông tin chi tiết để đưa ra quyết định.
* Tăng doanh thu: Cơ sở hạ tầng Big Data có thể giúp tổ chức tăng doanh thu bằng cách hiểu nhu cầu của khách hàng và phát triển các sản phẩm và dịch vụ mới.
* Phát triển sản phẩm mới: Cơ sở hạ tầng Big Data có thể giúp tổ chức phát triển sản phẩm và dịch vụ mới bằng cách cung cấp thông tin chi tiết về thị trường và nhu cầu của khách hàng.
* Đưa ra quyết định tốt hơn: Cơ sở hạ tầng Big Data có thể giúp tổ chức đưa ra quyết định tốt hơn bằng cách cung cấp thông tin chi tiết và hỗ trợ cho việc ra quyết định.

**Một số khía cạnh khác khi xây dựng cơ sở hạ tầng Big Data :**

* Chiến lược dữ liệu: Chiến lược dữ liệu là một kế hoạch tổng thể cho việc thu thập, lưu trữ, xử lý và phân tích dữ liệu. Chiến lược dữ liệu là cần thiết để đảm bảo rằng cơ sở hạ tầng Big Data được xây dựng phù hợp với nhu cầu của tổ chức.
* Chiến lược an ninh: Dữ liệu lớn có thể chứa thông tin nhạy cảm, vì vậy cần có một chiến lược an ninh mạnh mẽ để bảo vệ dữ liệu khỏi bị truy cập trái phép hoặc bị sử dụng sai mục đích.
* Chiến lược quản lý: Cơ sở hạ tầng Big Data đòi hỏi nhiều nguồn lực để quản lý, vì vậy cần có một chiến lược quản lý hiệu quả để đảm bảo rằng hệ thống hoạt động trơn tru và hiệu quả.

**Một số thách thức thường gặp :**

* Kích thước dữ liệu: Dữ liệu lớn có thể rất lớn, khiến việc thu thập, lưu trữ và xử lý trở thành một thách thức.
* Tốc độ dữ liệu: Dữ liệu lớn có thể được tạo ra với tốc độ rất nhanh, khiến việc thu thập và xử lý theo thời gian thực trở thành một thách thức.
* Phức tạp dữ liệu: Dữ liệu lớn có thể bao gồm nhiều loại dữ liệu khác nhau, khiến việc phân tích trở nên khó khăn hơn.
* Chi phí: Chi phí xây dựng và vận hành cơ sở hạ tầng Big Data có thể rất cao.

**Một số phần mềm và công cụ hỗ trợ phổ biến**

Để xây dựng và triển khai một dự án Big Data, cần có sự hỗ trợ của các phần mềm và công cụ chuyên dụng. Các phần mềm và công cụ này có thể giúp thực hiện các nhiệm vụ như thu thập, lưu trữ, xử lý và phân tích dữ liệu lớn.

**Một số phần mềm và công cụ hỗ trợ dự án Big Data phổ biến:**

* Phần mềm và công cụ thu thậ p dữ liệu: Các phần mềm và công cụ này giúp thu thập dữ liệu từ nhiều nguồn khác nhau, chẳng hạn như máy tính, thiết bị di động, cảm biến và mạng xã hội. Một số phần mềm và công cụ thu thập dữ liệu phổ biến bao gồm:
  + Apache Hadoop: Một khung công tác mã nguồn mở cho phép phân tích dữ liệu phân tán.
  + Apache Spark: Một khung công tác mã nguồn mở cho phép xử lý dữ liệu nhanh chóng.
  + MongoDB: Một cơ sở dữ liệu NoSQL cho phép lưu trữ dữ liệu phi cấu trúc.
  + Elasticsearch: Một công cụ tìm kiếm và phân tích dữ liệu thời gian thực.
  + Kafka: Một hệ thống phân phối thông điệp cho phép truyền dữ liệu theo thời gian thực.
* Phần mềm và công cụ lưu trữ dữ liệu: Các phần mềm và công cụ này giúp lưu trữ dữ liệu lớn một cách an toàn và hiệu quả. Một số phần mềm và công cụ lưu trữ dữ liệu phổ biến bao gồm:
  + Amazon S3: Dịch vụ lưu trữ đám mây của Amazon.
  + Google Cloud Storage: Dịch vụ lưu trữ đám mây của Google.
  + Microsoft Azure Storage: Dịch vụ lưu trữ đám mây của Microsoft.
  + OpenStack Swift: Một hệ thống lưu trữ đám mây mã nguồn mở.
* Một số IDE phổ biến cho phát triển Big Data bao gồm:
* IntelliJ IDEA là một IDE mạnh mẽ và toàn diện được sử dụng để phát triển các ứng dụng phần mềm. Nó cung cấp một loạt các tính năng và công cụ hỗ trợ cho việc phát triển các ứng dụng Big Data, bao gồm:

Hỗ trợ cho các ngôn ngữ lập trình Big Data phổ biến, chẳng hạn như Java, Scala, Python và R.

Tích hợp với các công cụ và thư viện Big Data phổ biến, chẳng hạn như Apache Hadoop, Apache Spark và Apache Hive.

Các tính năng và công cụ hỗ trợ cho việc gỡ lỗi, kiểm tra và tối ưu hóa các ứng dụng Big Data.

* Eclipse là một IDE mã nguồn mở được sử dụng rộng rãi cho phát triển phần mềm. Nó cung cấp một số plugin và tiện ích bổ sung hỗ trợ cho việc phát triển Big Data, bao gồm:

Plugin Hadoop Tools cung cấp hỗ trợ cho Apache Hadoop và các công cụ liên quan.

Plugin Spark Tools cung cấp hỗ trợ cho Apache Spark và các công cụ liên quan.

Plugin Hive Tools cung cấp hỗ trợ cho Apache Hive và các công cụ liên quan.

* Visual Studio Code là một IDE mã nguồn mở nhẹ được sử dụng rộng rãi cho phát triển phần mềm. Nó cung cấp một số plugin và tiện ích bổ sung hỗ trợ cho việc phát triển Big Data, bao gồm:

Plugin PySpark cung cấp hỗ trợ cho Python và Apache Spark.

Plugin Java Spark cung cấp hỗ trợ cho Java và Apache Spark.

Plugin Scala Spark cung cấp hỗ trợ cho Scala và Apache Spark.

* Phần mềm và công cụ xử lý dữ liệu: Các phần mềm và công cụ này giúp xử lý dữ liệu lớn một cách hiệu quả. Một số phần mềm và công cụ xử lý dữ liệu phổ biến bao gồm:
  + Apache Hadoop: Một khung công tác mã nguồn mở cho phép phân tích dữ liệu phân tán.
  + Apache Spark: Một khung công tác mã nguồn mở cho phép xử lý dữ liệu nhanh chóng.
  + Hadoop Hive: Một công cụ phân tích dữ liệu dựa trên Hadoop.
  + Hadoop Pig: Một ngôn ngữ lập trình được thiết kế để xử lý dữ liệu phân tán.
  + Hadoop HBase: Một cơ sở dữ liệu bảng phân tán.
* Phần mềm và công cụ phân tích dữ liệu: Các phần mềm và công cụ này giúp phân tích dữ liệu lớn để tìm ra các mẫu và xu hướng ẩn. Một số phần mềm và công cụ phân tích dữ liệu phổ biến bao gồm:
  + IBM SPSS: Một phần mềm phân tích dữ liệu thống kê.
  + SAS Enterprise Miner: Một phần mềm phân tích dữ liệu thống kê.
  + Microsoft Excel: Một phần mềm bảng tính có thể được sử dụng để phân tích dữ liệu.
  + R: Một ngôn ngữ lập trình thống kê.
  + Python: Một ngôn ngữ lập trình phổ biến có thể được sử dụng để phân tích dữ liệu.

Ngoài các phần mềm và công cụ chuyên dụng, còn có một số phần mềm và công cụ phổ biến có thể được sử dụng để hỗ trợ dự án Big Data. Một số phần mềm và công cụ phổ biến bao gồm:

* Hệ điều hành: Hệ điều hành là cần thiết để chạy các phần mềm và công cụ Big Data. Một số hệ điều hành phổ biến bao gồm Linux, Unix và Windows.
* Dụng cụ dòng lệnh: Dụng cụ dòng lệnh có thể được sử dụng để thực hiện các tác vụ cơ bản như quản lý tập tin và thư mục. Một số dụng cụ dòng lệnh phổ biến bao gồm Bash, Python và R.
* Công cụ quản lý dự án: Công cụ quản lý dự án có thể giúp theo dõi tiến độ và quản lý các nguồn lực của dự án. Một số công cụ quản lý dự án phổ biến bao gồm Jira, Trello và Asana.
* Công cụ cộng tác: Công cụ cộng tác có thể giúp các thành viên trong nhóm làm việc cùng nhau hiệu quả. Một số công cụ cộng tác phổ biến bao gồm Slack, Zoom và Microsoft Teams.

**Hệ sinh thái Hadoop**

**Apache Hadoop** hay Hadoop là một software framework hỗ trợ các ứng dụng phân tán dữ liệu chuyên sâu theo một giấy phép miễn phí. Nó cho phép các ứng dụng làm việc với hàng ngàn máy tính tính toán độc lập và petabyte dữ liệu. Hadoop được bắt nguồn từ các bài báo MapReduce của Google và Google File System (GFS).

**Hadoop** là một trong những dự án hàng đầu của Apache, được xây dựng và được sử dụng bởi một cộng đồng những người đóng góp toàn cầu, viết bằng ngôn ngữ lập trình Java. Yahoo! đã đóng góp lớn nhất cho dự án, và Hadoop được sử dụng rộng rãi trên khắp các doanh nghiệp.

**Có thể tóm tắt ngắn gọn về Hadoop như sau:**

* Là một framework cho phép phát triển các ứng dụng phân tán.
* Viết bằng Java tuy nhiên hỗ trợ được các ngôn ngữ khác như C++, Python, Perl bằng cơ chế streaming.
* Là một ứng dụng Linux-based (chỉ chạy trên môi trường Linux).

**Nguồn gốc**

Được tạo nên nhờ vào Doug Cutting và Mike Cafarella - hai nhà khoa học máy tính người Mỹ, đồng sáng lập dự án phần mềm mã nguồn mở Hadoop.

Doug Cutting sinh năm 1969, tốt nghiệp Đại học Berkeley với bằng cử nhân khoa học máy tính. Ông là người tạo ra Nutch, một công cụ tìm kiếm mã nguồn mở. Ông cũng là người đồng sáng lập Apache Software Foundation, tổ chức phát triển và duy trì các dự án phần mềm mã nguồn mở.



Mike Cafarella sinh năm 1972, tốt nghiệp Đại học California, Santa Cruz với bằng cử nhân khoa học máy tính. Ông là người đồng sáng lập Nutch và cũng là người đồng sáng lập Apache Software Foundation.



Năm 2005, Hadoop được tạo ra bởi Doug Cutting và Mike Cafarella khi làm việc tại Yahoo. Doug Cutting đặt tên Hadoop theo con voi đồ chơi của con trai mình.

**Vấn đề mà Hadoop phải giải quyết:**

* Làm việc với khối lượng dữ liệu khổng lồ (tính bằng Petabyte).
* Xử lý trong môi trường phân tán, dữ liệu lưu trữ ở nhiều phần cứng khác nhau, yêu cầu xử lý đồng bộ
* Các lỗi xuất hiện thường xuyên.
* Băng thông giữa các phần cứng vật lý chứa dữ liệu phân tán có giới hạn.
* Kích thước dữ liệu: Dữ liệu lớn có thể rất lớn, khiến việc thu thập, lưu trữ và xử lý trở thành một thách thức. Hadoop sử dụng một hệ thống tệp phân tán (HDFS) để lưu trữ dữ liệu lớn trên nhiều máy tính.
* Tốc độ dữ liệu: Dữ liệu lớn có thể được tạo ra với tốc độ rất nhanh, khiến việc thu thập và xử lý dữ liệu theo thời gian thực trở thành một thách thức. Hadoop sử dụng mô hình lập trình MapReduce để xử lý dữ liệu phân tán song song.
* Phức tạp dữ liệu: Dữ liệu lớn có thể bao gồm nhiều loại dữ liệu khác nhau, khiến việc phân tích trở nên khó khăn hơn. Hadoop hỗ trợ nhiều loại dữ liệu, bao gồm dữ liệu có cấu trúc, không có cấu trúc và bán cấu trúc.

**Các thành phần lõi của Hadoop là gì?**

Có hai thành phần chính của nền tảng Hadoop.

* **Hadoop MapReduce** là cách chia 1 vấn đề dữ liệu lớn hơn thành các đoạn nhỏ hơn và phân tán nó trên nhiều máy chủ. Mỗi máy chủ có 1 tập tài nguyên riêng và máy chủ xử lý dữ liệu trên cục bộ. Khi máy chủ xử lý xong dữ liệu, chúng sẽ gởi trở về máy chủ chính.
* **Hadoop Distributed File System (HDFS)** là hệ thống tập tin ảo. Có 1 sự khác biệt lớn giữa các hệ thống tập tin khác và Hadoop. Khi chúng ta di chuyển 1 tập tin trên HDFS, nó tự động chia thành nhiều mảnh nhỏ. Các đoạn nhỏ của tập tin sẽ được nhân rộng và lưu trữ trên nhiều máy chủ khác (thường là 3) để tăng sức chịu lỗi và tính sẵn sàng cao.

Bên cạnh 2 thành phần lõi, Hadoop cũng chứa 1 số modules:

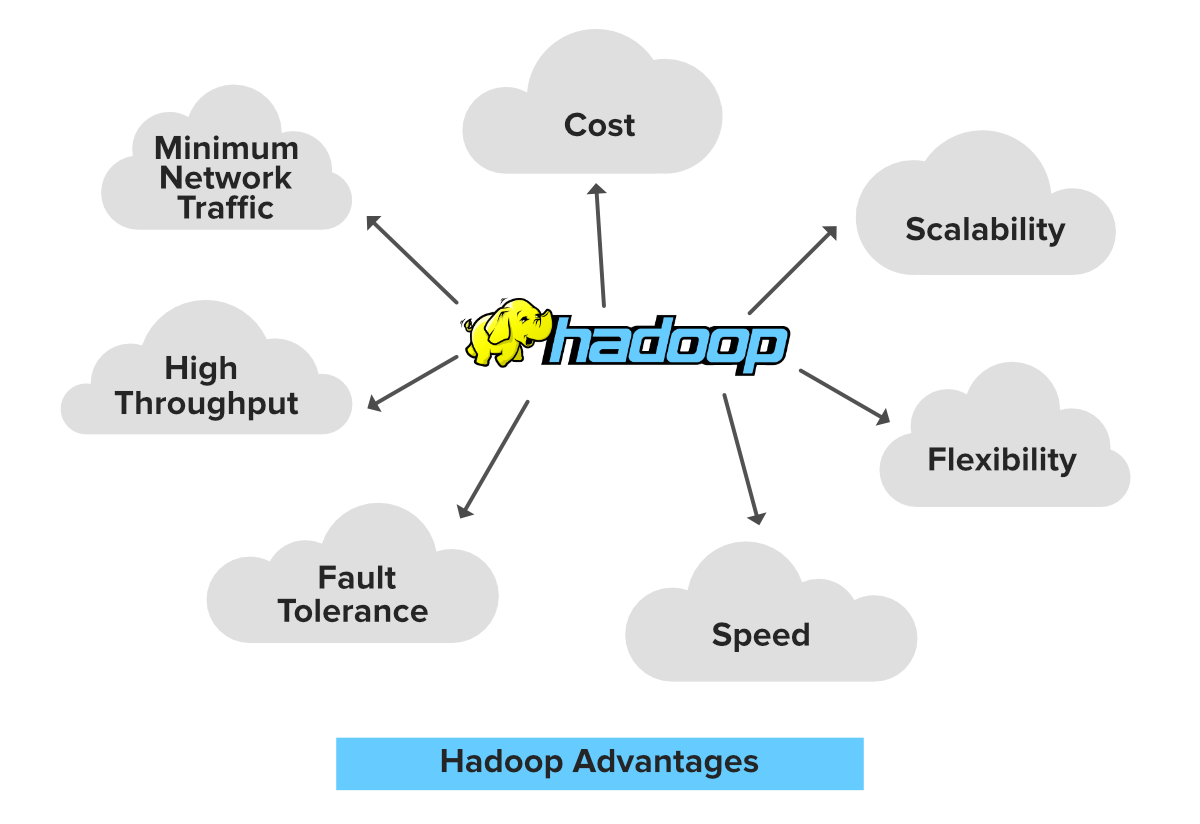
* **Hadoop Common**: các tiện tích thông dụng cho các module khác của Hadoop.
* **Hadoop Yarn**: một nền tảng cho lập lịch và quản lý tài nguyên cụm (cluster).

Ngoài các thành phần chính ,Hadoop ecosystem còn bao gồm nhiều công cụ và dịch vụ bổ sung, như:

* Hive: Công cụ truy vấn dữ liệu dựa trên SQL.
* Pig: Ngôn ngữ lập trình xử lý dữ liệu phân tán.
* HBase: Hệ thống cơ sở dữ liệu NoSQL phân tán.
* Spark: Khung công tác tính toán phân tán.
* Kafka: Hệ thống phân phối tin nhắn.
* Sqoop: Công cụ di chuyển dữ liệu giữa Hadoop và các hệ thống khác.

**Hướng giải quyết của Hadoop**

* Quản lý file phân tán: Hadoop Distributed File System (HDFS) sẽ chia nhỏ dữ liệu ra thành nhiều phần. Dữ liệu được quản lý một cách có hệ thống. Ý tưởng được sử dụng ở đây là “di chuyển tính toán vào trong dữ liệu” thay vì di chuyển dữ liệu để tính toán” như cách thông thường.



* Các tiến trình hoạt động độc lập: MapReduce là mô hình tổ chức của Hadoop, theo đó các node hoạt động một cách độc lập cao. Mặc dù giữa các node vẫn có giao tiếp nhưng bị giới
* hạn một cách tối đa, do đó giảm thiểu rủi ro khi một node bị lỗi.
* Khả năng mở rộng tự nhiên: nhiều chương trình được viết triên các hệ thống phân tán khác ngoài Hadoop, mỗi lần mở rộng hệ thống từ 10 máy lên 100 hay 1000 máy thì đòi hỏi phải sửa đổi cấu trúc chương trình rất là nhiều. Cách thiết kế của Hadoop giúp mở rộng chương trình một cách tự nhiên mà không cần phải thay đổi nhiều hoặc thay đổi rất ít trong một vài trường hợp đặc biệt.

**Tại sao dùng Hadoop?**

Các điểm thuận lợi khi dùng Hadoop:



* Robus and Scalable – Có thể thêm node mới và thay đổi chúng khi cần.
* Affordable and Cost Effective – Không cần phần cứng đặc biệt để chạy Hadoop.
* Adaptive and Flexible – Hadoop được xây dựng với tiêu chí xử lý dữ liệu có cấu trúc và không cấu trúc.
* Highly Available and Fault Tolerant – Khi 1 node lỗi, nền tảng Hadoop tự động chuyển sang node khác.

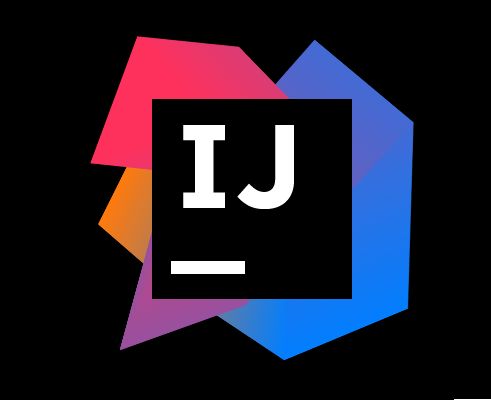
**Intellij IDEA**

Intellij IDEA là một trình IDE dùng để lập trình Java (nó cũng được sử dụng để lập trình một số ngôn ngữ khác như Node.js, python…)

Nhìn chung Intellij IDEA khá giống với Eclipse vì nó chủ yếu dùng cho Java nhưng vẫn có thể hỗ trợ các ngôn ngữ khác và có rất nhiều các plugin hỗ trợ.

Intellij IDEA có 2 bản là bản miễn phí (community) và bản trả phí (ultimate). Bản trả phí thì hỗ trợ thêm JavaScript, TypeScript, các plugin GWT, Vaadin… check các đoạn code trùng lặp…

Phần mềm IntelliJ IDEA, sản phẩm nổi tiếng của JetBrains đã nhận được rất nhiều giải thưởng. Phần mềm được thiết kế để cải tiến năng suất cho các nhà phát triển. IntelliJ IDEA cung cấp trình soạn thảo thông minh, trình phân tích mã và tập hợp mạnh mẽ của refactorings hỗ trợ một loạt các ngôn ngữ lập trình, các khuôn khổ và công nghệ, và đã sẵn sàng để sử dụng.



**Lợi ích chính IntelliJ IDEA**

* Cho phép các nhà phát triển tập trung phát triển và quản lý tất cả các tác vụ thông thường.
* Cho phép viết, gỡ lỗi, tái cấu trúc, kiểm tra và tìm hiểu mã của bạn mà không bị ảnh hưởng.
* Xử lý liền mạch cơ sở mã hỗn hợp của Java, Ruby, Groovy, Python và Scala.
* Tự động duy trì chất lượng mã.
* Theo dõi và sửa lỗi trên tất cả các cấp độ – từ các câu đến kiến trúc tổng thể.
* Tạo mã “sạch”, nhanh chóng thực hiện mã trong thời gian ngắn nhất.
* Được thiết kế để làm việc trên tất cả các quy mô – từ cá nhân đến doanh nghiệp.
* Hỗ trợ tất cả các ngôn ngữ, công nghệ và framework chính.
* Làm việc với các hệ thống điều khiển phiên bản phổ biến và TeamCity, server tích hợp liên tục.

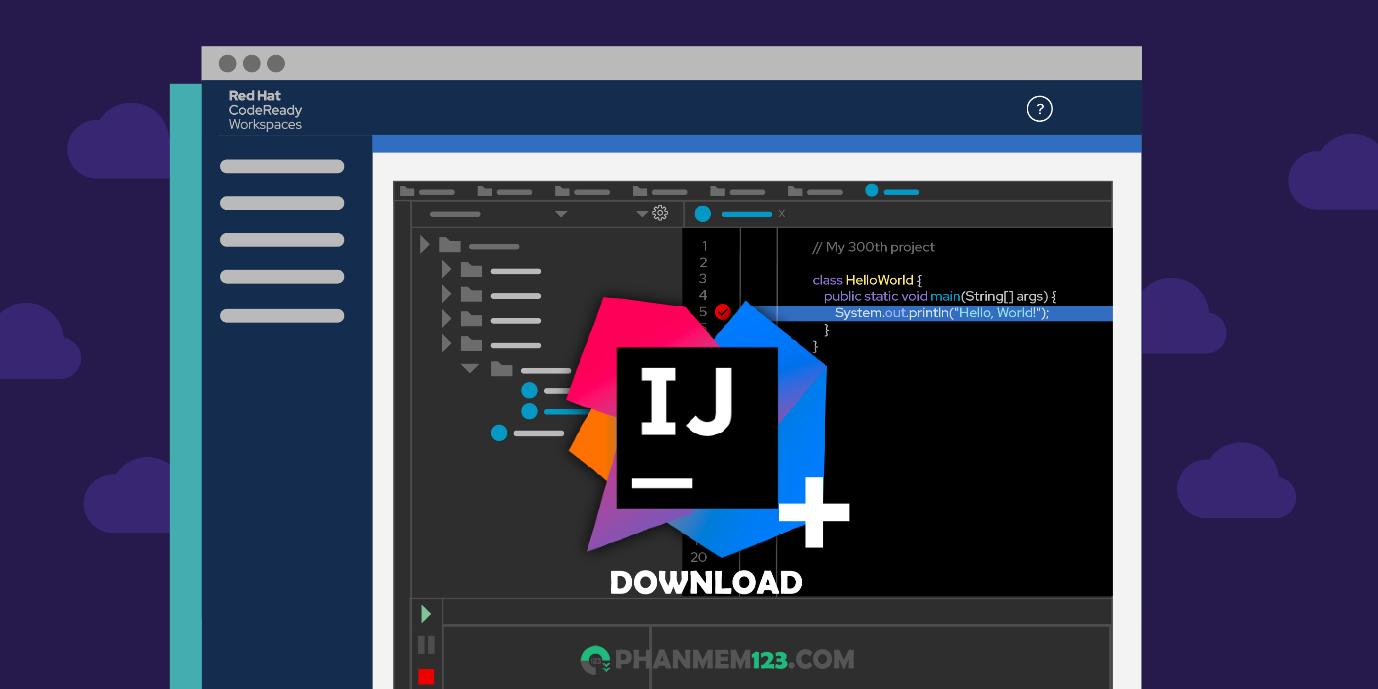
**CÁC TÍNH NĂNG CHÍNH CỦA INTELLIJ IDEA**

**Hỗ trợ mã hóa thông minh**

* Có trên 60 mã kiểm tra với khả năng phân tích mã siêu nhanh
* Tính năng sửa lỗi nhanh thông minh
* Tạo mã tự động và hỗ trợ Styling Consistent cho ngôn ngữ, công nghệ và ứng dụng server.
* Cung cấp các phương tiện tuyệt vời cho sự phát triển Java, JavaScript/ ActionScript /Flex, HTML/XHTML/CSS, XML/XSL, PHP, Ruby/JRuby, Groovy, SQL, FreeMarker/Velocity, JSP, JSF và JSF 2.0, EJB, AJAX, GWT, GWT 2.0, Google App Engine, Struts, Struts 2, JBoss Seam, Spring, Spring 2.5 and 3.0, Spring Security 2.0, Hibernate/JPA and JPA 2.0, Tapestry, Web Beans, Bean Validation, OSGi, Android, và nhiều hơn thế nữa.
* Hỗ trợ triển khai và gỡ lỗi từ xa cho WebLogic, WebSphere, Glassfish, Geronimo, Tomcat, JBoss và bất kỳ máy chủ ứng dụng tương thích JSR-45 nào.

**Bảo trì chất lượng mã**

* Tái cấu trúc trên 60 ngôn ngữ chéo
* Phân tích phụ thuộc
* Phát hiện mã trùng lặp
* Được tích hợp mã bao phủ với JUnit/ TestNG



**Môi trường tích hợp nhóm**

* Tích hợp kiểm soát phiên bản: Perforce, ClearCase, Subversion, CVS, Visual SourceSafe, TFS, Git và Mercurial
* Khả năng tương tác với Eclipse và Maven
* Tích hợp với JetBrains TeamCity

**Cấu hình máy tính cài đặt IntelliJ IDEA**

* Mac OS X 10.5 hoặc cao hơn, up to 10.7 (Lion)
* Microsoft Windows 7 (incl.64-bit)/Vista/2003/XP
* GNOME hoặc KDE desktop
* Tối thiểu 1 GB RAM, đề xuất 2 GB RAM
* 300 MB dung lượng ổ đĩa cứng và tối thiểu 1 GB cho bộ nhớ caches
* Độ phân giải màn hình tối thiêu 1024×768
* JDK 1.6 hoặc cao hơn.

**Apache Spark**



Apache Spark là một framework mã nguồn mở cho phép xử lý dữ liệu phân tán. Spark được phát triển bởi AMPLab của Đại học California, Berkeley và được Apache Software Foundation quản lý.

**Spark được thiết kế để giải quyết một số vấn đề chính liên quan đến Hadoop, bao gồm:**

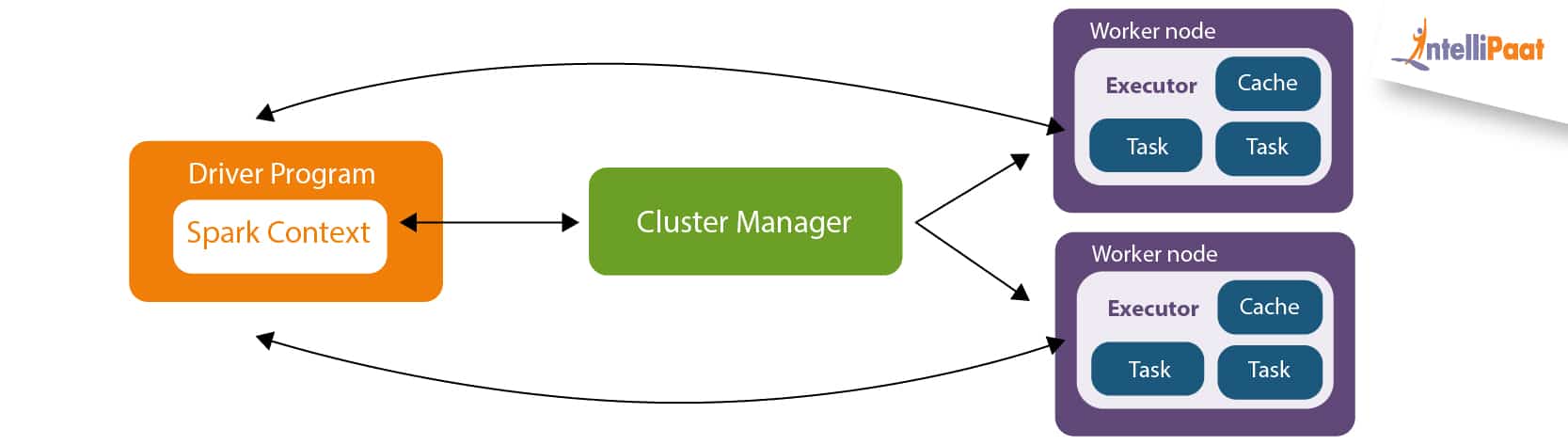
* Tốc độ: Spark có thể xử lý dữ liệu nhanh hơn Hadoop bằng cách sử dụng bộ nhớ trong thay vì lưu trữ dữ liệu trên đĩa.
* Linh hoạt: Spark có thể được sử dụng để xử lý nhiều loại dữ liệu và ứng dụng khác nhau.
* Khả năng mở rộng: Spark có thể được mở rộng để đáp ứng nhu cầu của các dự án Big Data lớn.

**Spark cung cấp một số tính năng và chức năng chính, bao gồm:**

* Xử lý trong bộ nhớ: Spark có thể xử lý dữ liệu trong bộ nhớ, điều này giúp tăng tốc độ xử lý.
* Xử lý song song: Spark có thể xử lý dữ liệu song song trên nhiều máy tính, điều này giúp tăng hiệu suất.
* Linh hoạt: Spark có thể được sử dụng để xử lý nhiều loại dữ liệu và ứng dụng khác nhau.
* Khả năng mở rộng: Spark có thể được mở rộng để đáp ứng nhu cầu của các dự án Big Data lớn.

**Spark được sử dụng trong nhiều lĩnh vực khác nhau, bao gồm:**

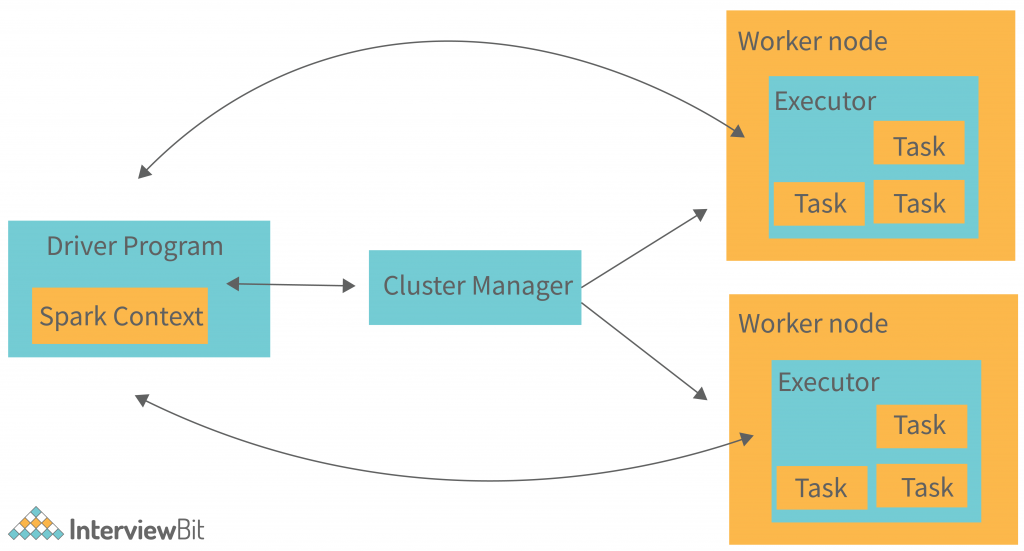
* Phân tích dữ liệu: Spark được sử dụng để phân tích dữ liệu lớn từ nhiều nguồn khác nhau.
* Trí tuệ nhân tạo: Spark được sử dụng để xây dựng các mô hình học máy và trí tuệ nhân tạo.
* Khoa học dữ liệu: Spark được sử dụng để thực hiện các phép tính khoa học và thống kê trên dữ liệu lớn.



**Dưới đây là một số ví dụ về cách Spark được sử dụng trong thế giới thực:**

* Các công ty sử dụng Spark để phân tích dữ liệu khách hàng để hiểu nhu cầu và hành vi của khách hàng.
* Các tổ chức tài chính sử dụng Spark để phát hiện gian lận và rủi ro.
* Các cơ quan chính phủ sử dụng Spark để phân tích dữ liệu từ các nguồn khác nhau, chẳng hạn như dữ liệu giám sát và dữ liệu dân số.

**Spark là một công nghệ đang phát triển nhanh chóng và có tiềm năng cách mạng hóa cách chúng ta xử lý dữ liệu lớn.**

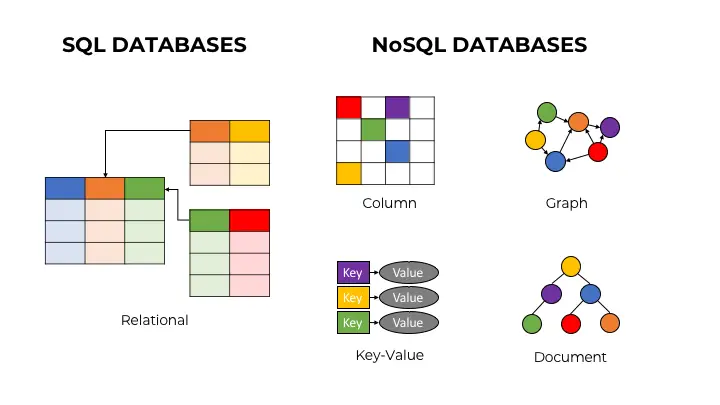


* Spark có thể được sử dụng để xử lý dữ liệu thời gian thực. Spark có thể xử lý dữ liệu trong bộ nhớ, điều này cho phép nó xử lý dữ liệu thời gian thực mà không cần lưu trữ dữ liệu trên đĩa.
* Spark có thể được sử dụng để xử lý dữ liệu lặp đi lặp lại. Spark có thể được sử dụng để xử lý dữ liệu lặp đi lặp lại một cách hiệu quả, điều này có thể hữu ích cho các ứng dụng học máy và trí tuệ nhân tạo.
* Spark có thể được sử dụng để xử lý dữ liệu phi cấu trúc. Spark có thể được sử dụng để xử lý dữ liệu phi cấu trúc, chẳng hạn như dữ liệu từ các nguồn như mạng xã hội và cảm biến.

Dữ liệu phi cấu trúc có thể được lưu trữ trong các cơ sở dữ liệu NoSQL

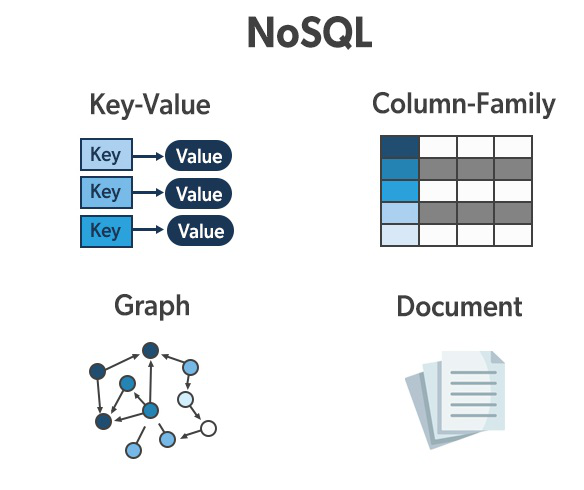
Các cơ sở dữ liệu NoSQL là các cơ sở dữ liệu không sử dụng mô hình dữ liệu quan hệ. Các cơ sở dữ liệu NoSQL thường được sử dụng để lưu trữ dữ liệu phi cấu trúc.

**NoSQL**

Cơ sở dữ liệu phi quan hệ NoSQL là một loại cơ sở dữ liệu không sử dụng mô hình dữ liệu quan hệ. Trong mô hình dữ liệu quan hệ, dữ liệu được lưu trữ trong các bảng, mỗi bảng có một tập hợp các hàng và cột. Các hàng đại diện cho các thực thể, chẳng hạn như khách hàng hoặc sản phẩm, và các cột đại diện cho các thuộc tính của các thực thể đó.

**Cơ sở dữ liệu NoSQL sử dụng các mô hình dữ liệu khác nhau, chẳng hạn như:**

* Khóa-giá trị: Mỗi dữ liệu được lưu trữ dưới dạng một cặp khóa-giá trị, trong đó khóa là một khóa duy nhất và giá trị là dữ liệu thực tế.
* Dữ liệu tài liệu: Dữ liệu được lưu trữ dưới dạng các tài liệu JSON hoặc XML, mỗi tài liệu đại diện cho một thực thể.
* Kho lưu trữ cột: Dữ liệu được lưu trữ theo cột, thay vì theo hàng. Điều này làm cho nó lý tưởng cho các ứng dụng yêu cầu truy vấn theo cột, chẳng hạn như phân tích dữ liệu.
* Cơ sở dữ liệu đồ thị: Dữ liệu được lưu trữ dưới dạng một mạng lưới các nút và cạnh. Các nút đại diện cho các thực thể và các cạnh đại diện cho mối quan hệ giữa các thực thể đó.



**Cơ sở dữ liệu NoSQL có một số ưu điểm so với cơ sở dữ liệu quan hệ, bao gồm:**

* Khả năng mở rộng: Cơ sở dữ liệu NoSQL có thể được mở rộng dễ dàng hơn cơ sở dữ liệu quan hệ.
* Khả năng chịu lỗi: Cơ sở dữ liệu NoSQL có thể chịu được sự cố của một số máy chủ mà không làm gián đoạn dịch vụ.
* Tính linh hoạt: Cơ sở dữ liệu NoSQL có thể được sử dụng cho nhiều loại ứng dụng hơn cơ sở dữ liệu quan hệ.

**Tuy nhiên, cơ sở dữ liệu NoSQL cũng có một số nhược điểm, bao gồm:**

* Hiệu suất truy vấn: Cơ sở dữ liệu NoSQL có thể không hiệu quả như cơ sở dữ liệu quan hệ cho các truy vấn phức tạp.
* Tính tương thích: Cơ sở dữ liệu NoSQL có thể không tương thích với các ứng dụng hiện có được thiết kế cho cơ sở dữ liệu quan hệ.

**Cơ sở dữ liệu NoSQL được sử dụng rộng rãi trong các ứng dụng sau:**

* Xử lý dữ liệu lớn: Cơ sở dữ liệu NoSQL được sử dụng để lưu trữ và truy cập dữ liệu lớn từ các nguồn khác nhau, chẳng hạn như mạng xã hội, dữ liệu cảm biến, và dữ liệu thương mại.
* Ứng dụng web thời gian thực: Cơ sở dữ liệu NoSQL được sử dụng để lưu trữ và truy cập dữ liệu cho các ứng dụng web thời gian thực, chẳng hạn như trò chơi trực tuyến và ứng dụng nhắn tin.
* Ứng dụng di động: Cơ sở dữ liệu NoSQL được sử dụng để lưu trữ và truy cập dữ liệu cho các ứng dụng di động.

**Một số các cơ sở dữ liệu NoSQL phổ biến:**

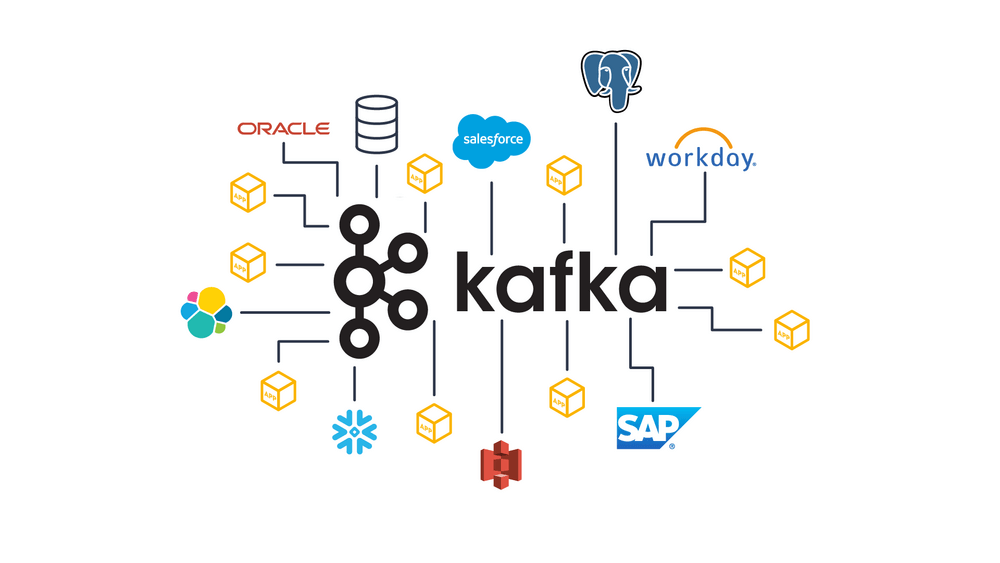
* MongoDB: Một cơ sở dữ liệu tài liệu dựa trên JSON.
* Cassandra: Một cơ sở dữ liệu khóa-giá trị phân tán.
* HBase: Một cơ sở dữ liệu cột mở rộng dựa trên Hadoop.
* Neo4j: Một cơ sở dữ liệu đồ thị.
* Redis: Một cơ sở dữ liệu khóa-giá trị tốc độ cao.

**Apache Kafka**

Apache Kafka là một nền tảng phát trực tuyến sự kiện phân tán mã nguồn mở. Apache Kafka được xây dựng nhằm mục đích xử lý dữ liệu streaming real-time (theo thời gian thực). Nói một cách đơn giản - Apache Kafka được phát triển để lưu trữ các streams of records (luồng ghi dữ liệu).

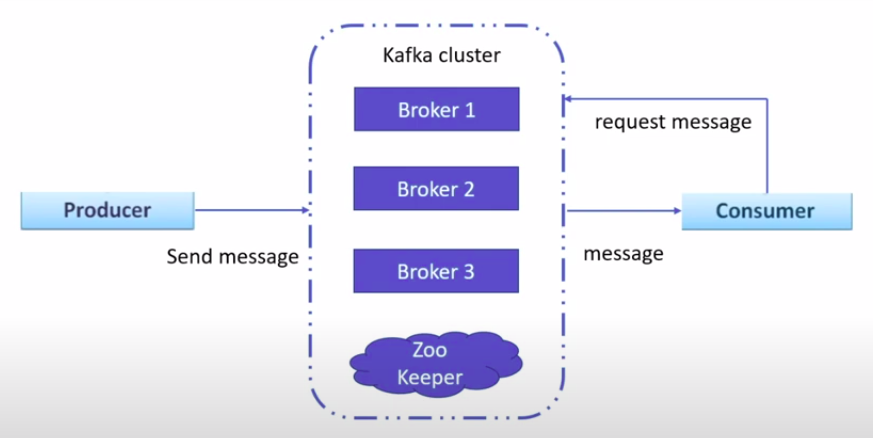
Ngày nay, hàng tỷ nguồn dữ liệu liên tục tạo ra các luồng data record, bao gồm các luồng sự kiện. Một sự kiện là một bản ghi kỹ thuật số về một hành động đã xảy ra và thời gian nó xảy ra. Thông thường, một sự kiện là một hành động thúc đẩy một hành động khác như một phần của quy trình. Một khách hàng đặt hàng, chọn chỗ ngồi trên chuyến bay, hoặc gửi đơn đăng ký đều là những ví dụ về các sự kiện. Một sự kiện không nhất thiết phải liên quan đến một người — ví dụ: báo cáo của máy điều nhiệt được kết nối về nhiệt độ tại một thời điểm nhất định cũng là một sự kiện.

Các luồng này tạo cơ hội cho các ứng dụng phản hồi dữ liệu hoặc sự kiện trong thời gian thực. Nền tảng stream dữ liệu cho phép các nhà phát triển xây dựng các ứng dụng liên tục sử dụng và xử lý các luồng này ở tốc độ cực cao, với mức độ trung thực và chính xác cao dựa trên thứ tự xuất hiện chính xác của chúng.

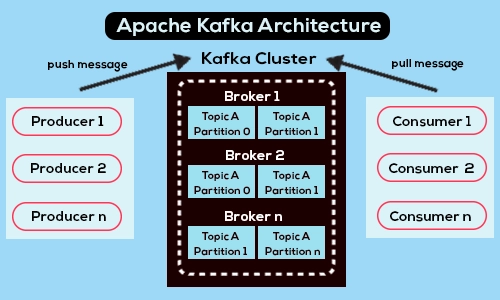
Ngày nay, Kafka đã phát triển thành nền tảng stream dữ liệu phân tán được sử dụng rộng rãi nhất, có khả năng nhập và xử lý hàng nghìn tỷ bản ghi mỗi ngày mà không có bất kỳ độ trễ hiệu suất có thể nhận thấy nào theo quy mô khối lượng

**Cách hoạt động của Kafka**

Kafka được xây dựng dựa vào mô hình subcribe/publish nên tương tự với hệ thống message khác.  Những ứng dụng gửi message tới node kafka và thông báo chúng sẽ được xử lý bởi các ứng dụng được gọi là consumers.  Khi những messages này gửi tới kafka node thì chúng đều được lưu trữ tại nơi gọi là topic. Sau đó, consumer hoàn toàn có thể subcribe đến topic và lắng nghe các messages. Khi đó, messages có thể là thông tin bất kỳ như: giá trị cảm biến, hành động của người dùng,...

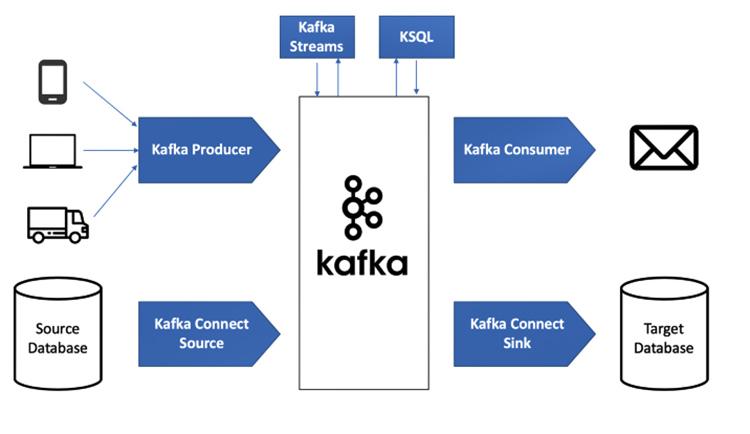


Còn topic sẽ được xem là tên danh mục mà những message được lưu trữ rồi đẩy vào.  Đa phần topics trong Kafka có kích cỡ rất lớn vì vậy, bạn không nên lưu trữ toàn bộ dữ liệu của topic vào ở trên một node. Nguồn dữ liệu này nên phân chia rõ ràng thành nhiều Partitions sẽ cho phép bạn thực hiện subcribe song song với topic cụ thể bằng biện pháp phân chia dữ liệu có trong một topic cụ thể Mỗi một Partition đều sẽ được đặt trên máy riêng biệt và cho phép nhiều consumer có thể đọc dữ liệu từ một topic diễn ra song song.  Ngoài ra, để gia tăng sự khả dụng của partition thì mỗi partition đều sở hữu giá trị replicas của riêng nó.

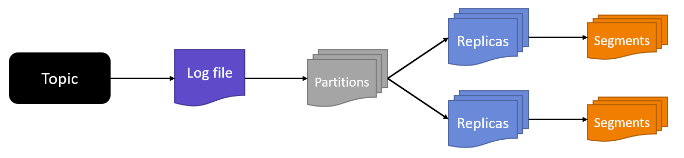


**Một số khái niệm cần hiểu rõ để sử dụng được kafka**

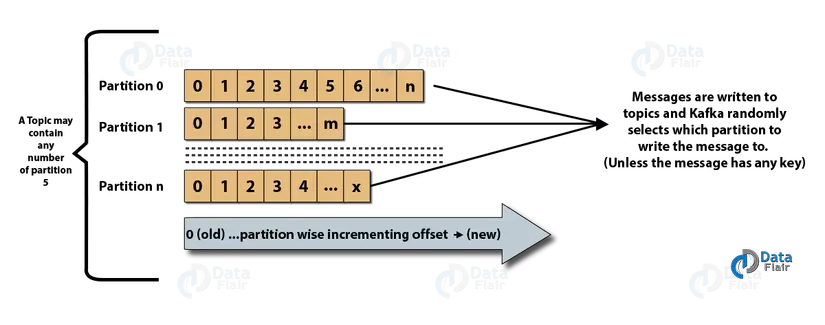
* ***Producer***: là thành phần tạo ra và gửi các message. Mặc định, producer sử dụng cơ chế round-robin (cơ chế luân phiên) để chọn partition lưu message khi gửi message cho topic.
* ***Consumer***: là thành phần nhận và đọc các message



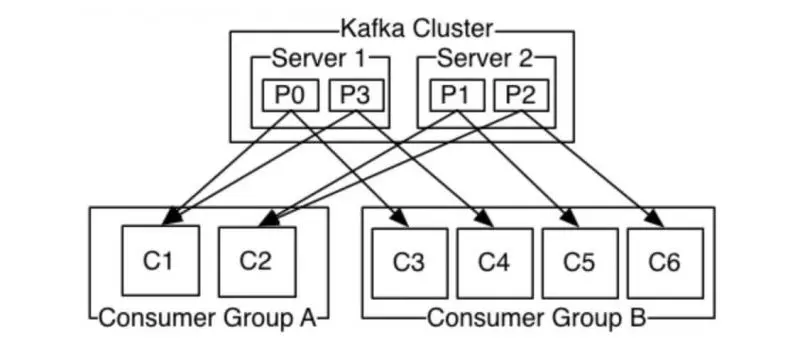
* ***Broker***: Một máy chủ Kafka được gọi là broker. Broker nhận đc message từ producers , gán địa chỉ cho chúng và lưu trữ chúng trên ổ đĩa. Comsumers kết nối broker, truy cập các message thông qua địa chỉ và đọc chúng. Message được lưu trong broker trong 1 thời gian(thiết đặt được) hoặc trong giới hạn dung dượng (thiết đặt được); nếu vượt qua giới hạn, các message sẽ bị xóa để nhường chỗ cho các message mới.
* ***Cluster***: là 1 tập các broker. Số broker trong cluster nên là số lẻ. Message trong broker được nhân bản và sao lưu trên các broker khác nhằm đảm bảo nếu 1 broker bị lỗi, dữ liệu luôn sẳn sàng phục vụ từ 1 broker khác. ZooKeeper được kafka cluster sử dụng để giám sát, quản lý điều phối các broker nhằm cung cấp một cơ chế đồng bộ linh hoạt và mạnh mẽ giữa các broker.
* ***Topic***: là 1 một chuổi(stream) message được producer gửi tới broker, từ đó consumer nhận được message. Tên topic là duy nhất trong toàn hệ thống. Mỗi topic được lưu dưới dạng log file, được chia thành nhiều partition, mỗi partition được cấu trúc trong 1 thư mục và tin nhắn được lưu trong các file .log(còn gọi là **log segment** với size default là 1G) bên trong các thư mục này. Các partition được trải rộng ở nhiều máy chủ, tùy theo cấu hình **replication-factor** mà tổng số partitions là khác nhau. Nếu cấu hình partitions là 5 và tham số replication-factor là 3(giới hạn max do cluster có 3 broker) thì chúng ta có 15 partitions để lưu tin nhắn, mỗi tin nhắn sẽ được sao lưu ở 3 partition.



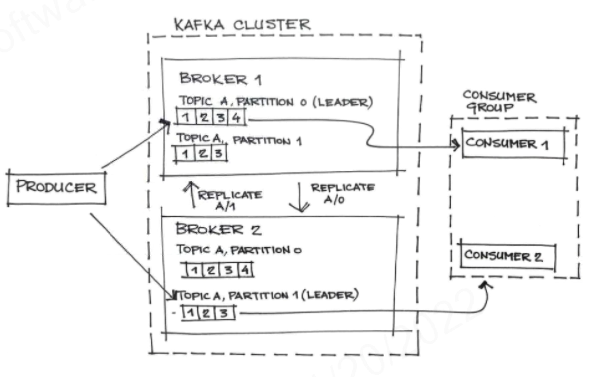
* ***Partition***: là 1 phân vùng của topic lưu dưới dạng log, chứa 1 chuổi tin nhắn có thứ tự và bất biến. Topic có thể bao gồm nhiều partition, mặc định là 1. Số lượng partition trong 1 topic có thể thiết lập tùy chỉnh được. Khi hệ thống có nhiều broker, các partition được phân tán trên cluser. ZooKeeper sẽ điều phối, quyết định partition nào sẽ được ghi và đọc dữ liệu.



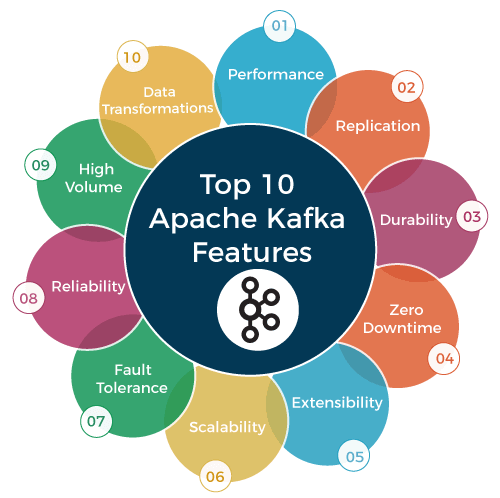
* ***Consumer group***: Một nhóm các consumer được nhóm lại với nhau để thực hiện 1 nghiệp vụ hoặc 1 nhiệm vụ. Mỗi consumer-group có 1 id-group đơn nhất. Mỗi consumer-group có thể truy cập 1 hoặc nhiều topic.



* ***Offset***: Là id của message ứng với số message đã được đọc ở 1 partition(cá nhân mình thấy nó giống con trỏ), nó định vị tin nhắn trong partition. Tất cả các offset được lưu và quản lý trong 1 topic của hệ thống gọi là “*\_\_consumer\_offset*“. Offset chỉ lưu giá trị commit gần nhất(đánh dấu dữ liệu đã được thay đổi trạng thái, đồng bộ trạng thái đó cho các broker, set giá trị mới cho offset). Khi consumer đọc(hoặc producer ghi) dữ liệu thành công, nó phải được commit. Nếu consumer chưa xử lý dữ liệu thành công, chưa commit giao dịch đọc thì consumer đó có thể thử đọc lại hoặc 1 consumer khác có thể đọc tiếp từ vị trí dữ liệu đó.



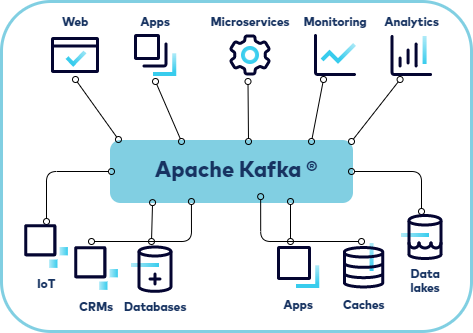
**10 tính năng hàng đầu của Apache Kafka**



* Data Transformations (Biến đổi dữ liệu) : Kafka cung cấp quy trình để suy ra các luồng dữ liệu mới bằng cách sử dụng các luồng dữ liệu từ producers
* Fault Tolerance (Khả năng chịu lỗi ) : Hệ thống Kafka có thể xử lý các lỗi với các máy chủ chính và cơ sở dữ liệu
* Scalability: Apache Kafka có thể xử lý khả năng mở rộng ở cả bốn chiều, tức là event producers, event processors, event consumers và event connectors. Nói cách khác, có khả năng mở rộng dễ dàng.
* High-Volume: Kafka có thể dễ dàng xử lí khối lượng dữ liệu khổng lồ (huge volume of data streams)
* Reliability: Vì Kafka được phân tán(distributed), phân vùng (partitioned), nhân bản (replicated) và chịu lỗi (fault tolerant), nên nó rất đáng tin cậy (Reliable).
* Durability: Nó bền (durable) bởi vì Kafka sử dụng Distributed commit log, điều đó có nghĩa là các messages vẫn tồn tại trên disk (messages persists on disk as fast as possible).
* Performance: Đối với cả publishing và subscribing messages, Kafka đều có thông lượng cao. Ngay cả khi nhiều TB messages được lưu trữ, nó vẫn duy trì hiệu suất ổn định.
* Zero Downtime: Kafka rất nhanh và đảm bảo zero downtime (không bị gián đoạn hoạt động) và zero data loss (không mất dữ liệu).
* Extensibility: Có nhiều cách để các ứng dụng có thể tích hợp và sử dụng
* Replication: Bằng cách sử dụng các pipelines, nó có thể sao chép các sự kiện (By using ingest pipelines, it can replicate the events.).

**Nhược điểm của Kafka**

* **Chưa có bộ công cụ giám sát hoàn chỉnh:** Có nhiều tool khác nhau nhưng mỗi tool chỉ đáp ứng một tính năng quản lý nhất định, chẳng hạn như: Kafka tool (offset manager) GUI tool - quản lý topic và consumer, Lense - hỗ trợ query message, Akhq - toolbox quản lý Kafka và view data bên trong Kafka
* **Giảm hiệu suất:** Kích thước message tăng khiến cho consumer và producer phải compress(nén) và decompress(giải nén) message, từ đó làm bộ nhớ bị chậm đi, ảnh hưởng đến throughput(lượng dữ liệu có thể được truyền, lưu trữ và xử lý trong một giây) và làm giảm hiệu suất.
* **Xử lý chậm:** Đôi khi số lượng queues trong Kafka cluster tăng đột biến khiến Kafka xử lý chậm hơn.
* **Chi phí:** Kafka có thể tốn kém để triển khai và duy trì, đặc biệt là đối với các ứng dụng có lưu lượng truy cập cao. Kafka yêu cầu máy chủ mạnh mẽ để lưu trữ và xử lý dữ liệu. Ngoài ra, bạn cần phải mua bản quyền của Apache Kafka để sử dụng nó trong sản xuất.

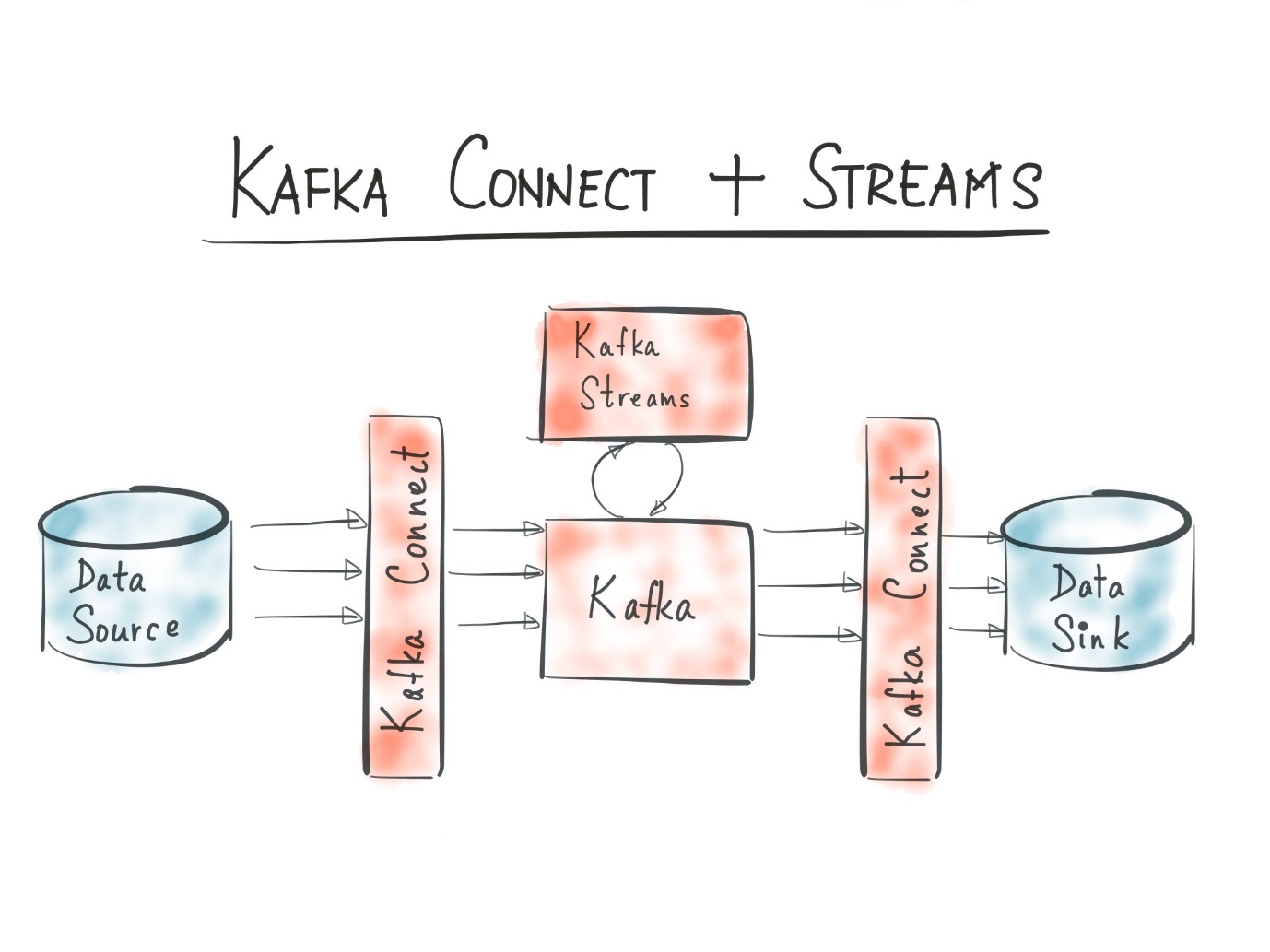


**Ứng dụng của Kafka**

Dưới đây là một số áp dụng hay ứng dụng phổ biến của Kafka

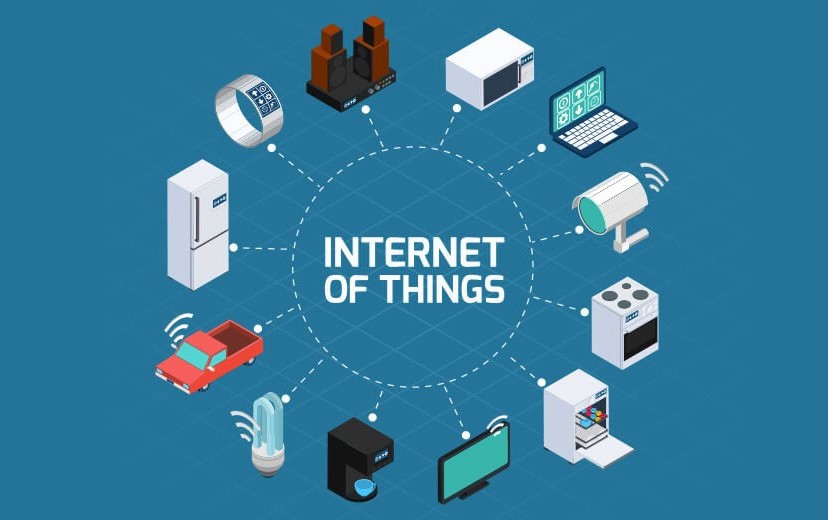
***Stream processing***

Bài toán này yêu cầu người dùng phải phân tích luồng dữ liệu bởi tốc độ cao. Nếu bạn muốn phân tích hành vi và xem sản phẩm của người dùng trên web thì cần phải ghi lại dưới dạng dữ liệu thô đơn giản như: click chuột, đơn hàng.



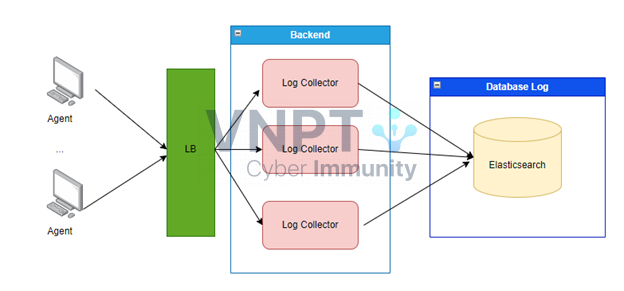
**Hệ thống IoT**

Hiện nay, Internet of Things đã cho phép thiết bị có thể gửi những dữ liệu thu thập được vào hệ thống máy chủ. Vì vậy, bạn có thể sử dụng Kafka để nhận các dữ liệu này thay cho việc tự phát triển API. Thông thường, dữ liệu thu thập bởi thiết bị có thể là Big data với tần suất gửi đi lớn nên bạn cần yêu cầu xử lý cũng như đáp ứng trong real-time.



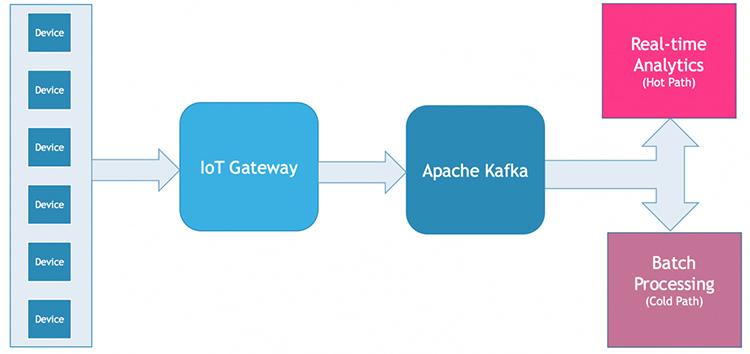
**Phân tích sự kiện và log**

In Log trong quá trình hệ thống đang chạy đặc biệt quan trọng cho quá trình bảo trì. Nếu người dùng có nhu cầu phân tích các log  thì nên đẩy dữ liệu log và Kafka.



**Thực hiện chuyển đổi ngôn ngữ lập trình**

Ứng dụng này áp dụng cho các nhu cầu chuyển đổi ngôn ngữ lập trình cấp thấp sang cấp cao. Giải pháp về share memory không thể giúp bạn tích hợp 2 ngôn ngữ lập trình lại với nhau. Vì vậy, giải pháp cho trường hợp này là đẩy kết quả dữ liệu của chương trình viết vào Kafka. Sau đó, sử dụng chương trình muốn chuyển đổi để lắng nghe topic rồi lấy ra dữ liệu tương ứng để phân tích và xử lý.



**Các trường hợp cần sử dụng Big Data (ứng dụng của Big Data)**

Big Data là một tập hợp dữ liệu rất lớn và phức tạp mà các ứng dụng xử lý dữ liệu truyền thống không xử lý được. Big Data có thể được sử dụng trong nhiều trường hợp khác nhau, bao gồm:

**Tìm hiểu xu hướng:** Big Data có thể được sử dụng để tìm hiểu các xu hướng trong dữ liệu, chẳng hạn như xu hướng tiêu dùng, xu hướng thời tiết và xu hướng xã hội.

**Phân tích khách hàng:** Big Data có thể được sử dụng để phân tích dữ liệu khách hàng, chẳng hạn như dữ liệu mua hàng, dữ liệu tương tác trên mạng xã hội và dữ liệu lịch sử. Điều này có thể giúp các doanh nghiệp hiểu khách hàng của mình tốt hơn và cung cấp các sản phẩm và dịch vụ phù hợp hơn.

**Phát hiện gian lận:** Big Data có thể được sử dụng để phát hiện gian lận, chẳng hạn như gian lận tài chính và gian lận bảo hiểm.

**Tự động hóa:** Big Data có thể được sử dụng để tự động hóa các quy trình, chẳng hạn như quy trình xử lý đơn hàng và quy trình chăm sóc khách hàng.

**Lập kế hoạch:** Big Data có thể được sử dụng để lập kế hoạch, chẳng hạn như lập kế hoạch sản xuất và lập kế hoạch marketing.

**Ứng dụng cụ thể của Big Data trong đời sống:**

Trong lĩnh vực y tế: Big Data có thể được sử dụng để nghiên cứu các bệnh tật, phát triển các phương pháp điều trị mới và cải thiện chất lượng chăm sóc sức khỏe.

Trong lĩnh vực kinh doanh: Big Data có thể được sử dụng để cải thiện hiệu quả hoạt động, tăng doanh số bán hàng và giảm chi phí.

Trong lĩnh vực chính phủ: Big Data có thể được sử dụng để cải thiện dịch vụ công, giải quyết các vấn đề xã hội và bảo vệ an ninh quốc gia.

Trong lĩnh vực giáo dục: Big Data có thể được sử dụng để cá nhân hóa việc học, cải thiện hiệu quả giảng dạy và đánh giá học sinh.

Trong lĩnh vực giải trí: Big Data có thể được sử dụng để cá nhân hóa trải nghiệm giải trí, tạo ra nội dung mới và tối ưu hóa các chiến dịch tiếp thị.

Cụ thể :

**Big Data có thể được sử dụng trong lĩnh vực y tế để:**

* Nghiên cứu các bệnh tật: Big Data có thể được sử dụng để phân tích dữ liệu từ các nguồn khác nhau, chẳng hạn như dữ liệu hồ sơ bệnh án, dữ liệu nghiên cứu và dữ liệu từ các thiết bị đeo được. Điều này có thể giúp các nhà khoa học hiểu rõ hơn về các bệnh tật và phát triển các phương pháp điều trị mới.
* Phát triển các phương pháp điều trị mới: Big Data có thể được sử dụng để thử nghiệm các phương pháp điều trị mới một cách hiệu quả và an toàn hơn. Điều này có thể giúp rút ngắn thời gian cần thiết để phát triển các phương pháp điều trị mới.
* Cải thiện chất lượng chăm sóc sức khỏe: Big Data có thể được sử dụng để cá nhân hóa chăm sóc sức khỏe cho từng bệnh nhân. Điều này có thể giúp cải thiện kết quả điều trị và giảm chi phí chăm sóc sức khỏe.
* Dự đoán bệnh tật: Big Data có thể được sử dụng để phân tích dữ liệu từ các nguồn khác nhau để dự đoán nguy cơ mắc bệnh của một người. Điều này có thể giúp các bác sĩ phát hiện và điều trị bệnh sớm hơn.
* Chẩn đoán bệnh: Big Data có thể được sử dụng để phân tích hình ảnh y tế, chẳng hạn như ảnh chụp X-quang, MRI và CT. Điều này có thể giúp các bác sĩ chẩn đoán bệnh chính xác hơn.
* Lập kế hoạch điều trị: Big Data có thể được sử dụng để phân tích dữ liệu lịch sử bệnh án của một bệnh nhân để lập kế hoạch điều trị tốt nhất. Điều này có thể giúp cải thiện kết quả điều trị.
* Theo dõi bệnh tật: Big Data có thể được sử dụng để theo dõi hiệu quả của các phương pháp điều trị và phát hiện các tác dụng phụ sớm hơn.
* Tối ưu hóa các dịch vụ y tế: Big Data có thể được sử dụng để tối ưu hóa các dịch vụ y tế, chẳng hạn như lập kế hoạch nhân sự và quản lý nguồn lực. Điều này có thể giúp cải thiện hiệu quả và hiệu quả của hệ thống chăm sóc sức khỏe.
* Tối ưu hóa việc sử dụng thuốc: Big Data có thể được sử dụng để phân tích dữ liệu sử dụng thuốc để xác định các bệnh nhân có nguy cơ cao gặp phải các tác dụng phụ hoặc tương tác thuốc. Điều này có thể giúp các bác sĩ kê đơn thuốc an toàn và hiệu quả hơn.
* Tăng cường sự tham gia của bệnh nhân: Big Data có thể được sử dụng để tạo ra các ứng dụng và dịch vụ giúp bệnh nhân hiểu rõ hơn về tình trạng sức khỏe của họ và tham gia tích cực hơn vào quá trình điều trị. Điều này có thể giúp cải thiện kết quả điều trị và giảm chi phí chăm sóc sức khỏe.
* Cải thiện chăm sóc sức khỏe từ xa: Big Data có thể được sử dụng để phát triển các công nghệ cho phép các bác sĩ chăm sóc bệnh nhân từ xa. Điều này có thể giúp cải thiện khả năng tiếp cận chăm sóc sức khỏe ở những khu vực có ít nguồn lực y tế.

**Big Data có thể được sử dụng trong lĩnh vực kinh doanh để:**

* Tìm hiểu khách hàng: Big Data có thể được sử dụng để phân tích dữ liệu khách hàng, chẳng hạn như dữ liệu mua hàng, dữ liệu tương tác trên mạng xã hội và dữ liệu lịch sử. Điều này có thể giúp các doanh nghiệp hiểu khách hàng của mình tốt hơn và cung cấp các sản phẩm và dịch vụ phù hợp hơn.
* Tối ưu hóa hoạt động: Big Data có thể được sử dụng để tối ưu hóa các hoạt động kinh doanh, chẳng hạn như chuỗi cung ứng, sản xuất và bán hàng. Điều này có thể giúp doanh nghiệp giảm chi phí và cải thiện hiệu quả.
* Phát hiện gian lận: Big Data có thể được sử dụng để phát hiện gian lận, chẳng hạn như gian lận tài chính và gian lận bảo hiểm. Điều này có thể giúp doanh nghiệp bảo vệ tài sản và lợi nhuận của mình.
* Tạo ra các sản phẩm và dịch vụ mới: Big Data có thể được sử dụng để tạo ra các sản phẩm và dịch vụ mới đáp ứng nhu cầu của khách hàng. Điều này có thể giúp doanh nghiệp tăng doanh số bán hàng và lợi nhuận.
* Tối ưu hóa chiến lược tiếp thị: Big Data có thể được sử dụng để phân tích dữ liệu về hành vi của khách hàng để xác định các cơ hội tiếp thị mới. Điều này có thể giúp doanh nghiệp tiếp cận khách hàng mục tiêu một cách hiệu quả hơn.
* Tăng cường trải nghiệm khách hàng: Big Data có thể được sử dụng để cá nhân hóa trải nghiệm khách hàng, chẳng hạn như các đề xuất sản phẩm và dịch vụ. Điều này có thể giúp doanh nghiệp giữ chân khách hàng và tăng doanh số bán hàng.
* Tối ưu hóa chuỗi cung ứng: Big Data có thể được sử dụng để dự đoán nhu cầu của khách hàng và điều chỉnh chuỗi cung ứng cho phù hợp. Điều này có thể giúp doanh nghiệp giảm chi phí và cải thiện hiệu quả.
* Phát hiện biến động thị trường: Big Data có thể được sử dụng để phân tích dữ liệu về thị trường để xác định các cơ hội và rủi ro đầu tư. Điều này có thể giúp doanh nghiệp đưa ra các quyết định đầu tư hiệu quả hơn.

Big Data là một công nghệ có tiềm năng to lớn để cải thiện hiệu quả kinh doanh. Các ứng dụng của Big Data trong lĩnh vực kinh doanh đang tiếp tục phát triển và mở rộng.

**Ứng dụng của Big Data trong lĩnh vực kinh doanh đối với một số ông lớn công nghệ:**

* Amazon sử dụng Big Data để cá nhân hóa trải nghiệm mua sắm của khách hàng. Công ty sử dụng dữ liệu về lịch sử mua sắm của khách hàng để đề xuất các sản phẩm và dịch vụ liên quan.
* Netflix sử dụng Big Data để đề xuất phim và chương trình truyền hình cho khách hàng. Công ty sử dụng dữ liệu về lịch sử xem của khách hàng để xác định các nội dung mà khách hàng có thể quan tâm.
* Walmart sử dụng Big Data để dự đoán nhu cầu của khách hàng. Công ty sử dụng dữ liệu về lịch sử mua sắm của khách hàng để điều chỉnh lượng hàng tồn kho cho phù hợp.
* Uber sử dụng Big Data để tối ưu hóa các chuyến đi. Công ty sử dụng dữ liệu về vị trí của khách hàng và tài xế để xác định các chuyến đi hiệu quả nhất.

**Big Data có thể được sử dụng trong lĩnh vực chính phủ để:**

* Cải thiện dịch vụ công: Big Data có thể được sử dụng để cải thiện hiệu quả và hiệu quả của các dịch vụ công, chẳng hạn như chăm sóc sức khỏe, giáo dục và giao thông.
* Giải quyết các vấn đề xã hội: Big Data có thể được sử dụng để giải quyết các vấn đề xã hội, chẳng hạn như nghèo đói, thất nghiệp và tội phạm.
* Bảo vệ an ninh quốc gia: Big Data có thể được sử dụng để bảo vệ an ninh quốc gia, chẳng hạn như phát hiện và ngăn chặn các mối đe dọa khủng bố.

**Ứng dụng cụ thể của Big Data trong lĩnh vực chính phủ:**

* Dự đoán thiên tai: Big Data có thể được sử dụng để phân tích dữ liệu từ các nguồn khác nhau, chẳng hạn như dữ liệu vệ tinh và dữ liệu thời tiết, để dự đoán thiên tai. Điều này có thể giúp các cơ quan chính phủ chuẩn bị và ứng phó với các thảm họa thiên nhiên.
* Phát hiện gian lận: Big Data có thể được sử dụng để phát hiện gian lận trong các chương trình phúc lợi và các dự án chính phủ khác. Điều này có thể giúp chính phủ bảo vệ tài sản của mình.
* Tối ưu hóa quy trình: Big Data có thể được sử dụng để tối ưu hóa các quy trình chính phủ, chẳng hạn như quy trình cấp phép và quy trình tuyển dụng. Điều này có thể giúp chính phủ hoạt động hiệu quả hơn.
* Cải thiện giao thông: Big Data có thể được sử dụng để cải thiện giao thông, chẳng hạn như giảm tắc nghẽn và cải thiện an toàn.
* Tăng cường an ninh: Big Data có thể được sử dụng để tăng cường an ninh, chẳng hạn như phát hiện và ngăn chặn tội phạm.

**Ứng dụng của Big Data trong lĩnh vực chính phủ đối với 1 số quốc gia :**

* Chính phủ Hoa Kỳ sử dụng Big Data để phát hiện và ngăn chặn các mối đe dọa khủng bố. Chính phủ sử dụng dữ liệu từ các nguồn khác nhau, chẳng hạn như dữ liệu truyền thông xã hội và dữ liệu giao thông, để xác định các cá nhân hoặc nhóm có nguy cơ khủng bố.
* Chính phủ Singapore sử dụng Big Data để cải thiện giao thông. Chính phủ sử dụng dữ liệu từ các camera giám sát và cảm biến giao thông để dự đoán và điều chỉnh giao thông.
* Chính phủ Brazil sử dụng Big Data để cải thiện dịch vụ chăm sóc sức khỏe. Chính phủ sử dụng dữ liệu từ các hồ sơ bệnh án để cải thiện chất lượng chăm sóc sức khỏe và giảm chi phí.

**Big Data có thể được sử dụng trong lĩnh vực giáo dục để:**

* Cá nhân hóa việc học: Big Data có thể được sử dụng để phân tích dữ liệu về học sinh, chẳng hạn như điểm số, lịch sử học tập và sở thích, để tạo ra các chương trình học được cá nhân hóa phù hợp với nhu cầu của từng học sinh. Điều này có thể giúp học sinh học tập hiệu quả hơn và đạt được kết quả tốt hơn.
* Tăng cường hiệu quả giảng dạy: Big Data có thể được sử dụng để phân tích dữ liệu về giảng viên, chẳng hạn như phương pháp giảng dạy và kết quả học tập của học sinh, để cải thiện hiệu quả giảng dạy. Điều này có thể giúp giảng viên giảng dạy hiệu quả hơn và giúp học sinh học tập hiệu quả hơn.
* Tối ưu hóa hoạt động giáo dục: Big Data có thể được sử dụng để phân tích dữ liệu về trường học, chẳng hạn như ngân sách, cơ sở vật chất và nhân sự, để tối ưu hóa hoạt động giáo dục. Điều này có thể giúp trường học hoạt động hiệu quả hơn và cung cấp các dịch vụ giáo dục tốt hơn cho học sinh.

**Ứng dụng cụ thể của Big Data trong lĩnh vực giáo dục:**

* Dự đoán kết quả học tập: Big Data có thể được sử dụng để phân tích dữ liệu về học sinh để dự đoán kết quả học tập của họ. Điều này có thể giúp giáo viên và phụ huynh phát hiện sớm các học sinh gặp khó khăn và cung cấp hỗ trợ kịp thời.
* Chẩn đoán tình trạng học tập: Big Data có thể được sử dụng để phân tích dữ liệu về học sinh để chẩn đoán các vấn đề học tập của họ. Điều này có thể giúp giáo viên và phụ huynh xác định các nguyên nhân của các vấn đề học tập và đưa ra các giải pháp phù hợp.
* Tùy chỉnh nội dung học tập: Big Data có thể được sử dụng để tạo ra các nội dung học tập được tùy chỉnh phù hợp với nhu cầu của từng học sinh. Điều này có thể giúp học sinh học tập hiệu quả hơn và đạt được kết quả tốt hơn.
* Tối ưu hóa quy trình giảng dạy: Big Data có thể được sử dụng để phân tích dữ liệu về giảng viên để cải thiện hiệu quả giảng dạy. Điều này có thể giúp giảng viên giảng dạy hiệu quả hơn và giúp học sinh học tập hiệu quả hơn.
* Tối ưu hóa hoạt động giáo dục: Big Data có thể được sử dụng để phân tích dữ liệu về trường học để tối ưu hóa hoạt động giáo dục. Điều này có thể giúp trường học hoạt động hiệu quả hơn và cung cấp các dịch vụ giáo dục tốt hơn cho học sinh.

**Ứng dụng của Big Data trong lĩnh vực giáo dục ở một số quốc gia :**

* Hệ thống giáo dục của Singapore sử dụng Big Data để cá nhân hóa việc học cho học sinh. Hệ thống sử dụng dữ liệu về học sinh, chẳng hạn như điểm số, lịch sử học tập và sở thích, để tạo ra các chương trình học được cá nhân hóa phù hợp với nhu cầu của từng học sinh.
* Đại học Stanford sử dụng Big Data để phân tích dữ liệu về giảng viên để cải thiện hiệu quả giảng dạy. Trường sử dụng dữ liệu về giảng viên, chẳng hạn như phương pháp giảng dạy và kết quả học tập của học sinh, để xác định các lĩnh vực cần cải thiện và đưa ra các giải pháp phù hợp.
* Quỹ Bill & Melinda Gates sử dụng Big Data để tối ưu hóa hoạt động giáo dục. Quỹ sử dụng dữ liệu về trường học, chẳng hạn như ngân sách, cơ sở vật chất và nhân sự, để xác định các lĩnh vực cần cải thiện và hỗ trợ các trường học cải thiện hiệu quả hoạt động.

**Big Data có thể được sử dụng trong lĩnh vực giải trí để:**

* Cá nhân hóa trải nghiệm giải trí: Big Data có thể được sử dụng để phân tích dữ liệu về người dùng, chẳng hạn như lịch sử xem, lịch sử mua hàng và sở thích, để tạo ra các trải nghiệm giải trí được cá nhân hóa phù hợp với nhu cầu của từng người dùng. Điều này có thể giúp người dùng tìm thấy nội dung giải trí mà họ yêu thích và có trải nghiệm giải trí thú vị hơn.
* Tối ưu hóa chiến lược tiếp thị: Big Data có thể được sử dụng để phân tích dữ liệu về người dùng để xác định các cơ hội tiếp thị mới. Điều này có thể giúp các nhà tiếp thị tiếp cận khách hàng mục tiêu một cách hiệu quả hơn.
* Tạo ra các sản phẩm và dịch vụ giải trí mới: Big Data có thể được sử dụng để phân tích dữ liệu về người dùng để xác định các nhu cầu và thị hiếu mới. Điều này có thể giúp các nhà sản xuất và nhà phát triển tạo ra các sản phẩm và dịch vụ giải trí mới đáp ứng nhu cầu của người dùng.

**Ứng dụng của Big Data trong lĩnh vực giải trí đối với 1 số nền tảng lớn :**

* Netflix sử dụng Big Data để cá nhân hóa trải nghiệm xem của người dùng. Công ty sử dụng dữ liệu về lịch sử xem của người dùng để đề xuất các nội dung giải trí liên quan.
* Spotify sử dụng Big Data để cá nhân hóa trải nghiệm nghe nhạc của người dùng. Công ty sử dụng dữ liệu về sở thích âm nhạc của người dùng để tạo ra các danh sách phát được cá nhân hóa.
* YouTube sử dụng Big Data để tối ưu hóa chiến lược tiếp thị. Công ty sử dụng dữ liệu về người dùng để xác định các cơ hội tiếp thị mới.
* Disney sử dụng Big Data để tạo ra các sản phẩm và dịch vụ giải trí mới. Công ty sử dụng dữ liệu về người dùng để xác định các nhu cầu và thị hiếu mới.

**2. Bài toán**

**Bài toán của nhóm :** Sử dụng Kafka lấy dữ liệu trên Twitter , sử dụng Structured Streaming trong Spark lấy dữ liệu từ Kafka ,sử dụng dữ liệu trong Spark cho các ứng dụng

**3. Giải pháp**

Đầu tiên chúng em sẽ cào dữ liệu từ một trang trung gian , cào gián tiếp chứ không phải cào trực tiếp do Twitter đã thay đổi chính sách API của mình vào tháng 1 năm 2023, khiến việc cào dữ liệu trực tiếp từ Twitter trở nên khó khăn hơn. Trước đây, các nhà phát triển có thể sử dụng API của Twitter để truy vấn dữ liệu từ Twitter, bao gồm tweet, người dùng, thẻ và hơn thế nữa. Tuy nhiên, chính sách mới giới hạn số lượng truy vấn mà một ứng dụng có thể thực hiện trong một khoảng thời gian nhất định.  
Chi phí cào dữ liệu Twitter có thể lên tới hàng triệu đô la, tùy thuộc vào khối lượng dữ liệu và thời gian cào. Ví dụ, nếu bạn muốn cào toàn bộ lịch sử tweet của một tài khoản, bạn sẽ cần phải cào hàng nghìn tỷ tweet. Điều này sẽ tốn rất nhiều thời gian và tiền bạc.

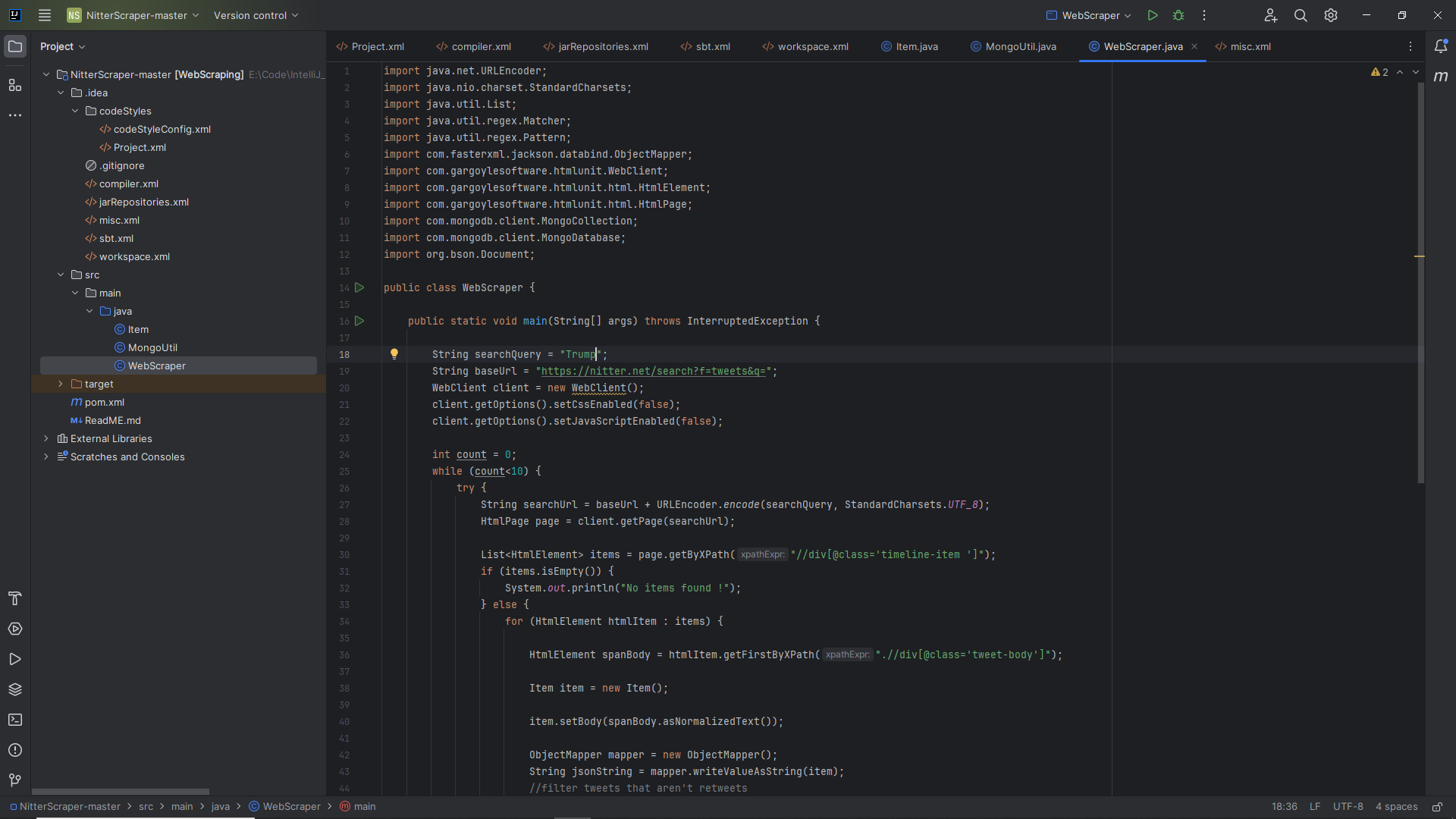
Dưới đây là một số ví dụ về chi phí cào dữ liệu Twitter:

* Cào 100 triệu tweet: Sử dụng API của bên thứ ba như Tweepy, chi phí sẽ vào khoảng 20.000 đô la.
* Cào 1 tỷ tweet: Sử dụng API của bên thứ ba như Tweepy, chi phí sẽ vào khoảng 200.000 đô la.
* Cào toàn bộ lịch sử tweet của một tài khoản: Sử dụng API của bên thứ ba như Tweepy, chi phí có thể lên tới hàng triệu đô la.

Điều này làm cho việc cào dữ liệu Twitter trở nên tốn kém hơn và khó thực hiện hơn. Các nhà phát triển hiện cần phải sử dụng các phương pháp khác để truy cập dữ liệu Twitter, chẳng hạn như sử dụng API của bên thứ ba hoặc cào dữ liệu từ các nguồn khác, chẳng hạn như **Nitter**.

Và chúng em đã chọn con đường cào dữ liệu từ Twitter thông qua **Nitter ,**nhìn chung, Nitter là một lựa chọn thay thế tốt cho Twitter , nó còn toàn bộ dữ liệu từ Twitter và không phải tốn chi phí để cào dữ liệu trực tiếp .

Nitter là một trang web cho phép người dùng xem các tweet mà không cần đăng nhập vào twitter. Nitter cũng cung cấp một API công khai để lấy dữ liệu từ twitter mà không cần sử dụng các token xác thực. Để lấy dữ liệu từ nitter mà không thông qua twitter, chúng ta có thể sử dụng kafka và spark streaming để tạo một hệ thống xử lý dữ liệu theo thời gian thực. Các bước chính như sau: - Tạo một producer kafka để gửi các yêu cầu API đến nitter và nhận về các phản hồi JSON chứa các tweet. - Tạo một consumer kafka để nhận các phản hồi JSON từ producer và lưu trữ chúng vào một topic kafka. - Tạo một spark streaming job để đọc dữ liệu từ topic kafka và xử lý chúng theo nhu cầu. Ví dụ, chúng ta có thể lọc, phân tích, biến đổi, hoặc lưu trữ dữ liệu vào một hệ thống lưu trữ khác. - Theo dõi và kiểm soát hệ thống xử lý dữ liệu bằng các công cụ như kafka manager, spark UI, hoặc grafana. Bằng cách sử dụng giải pháp này, chúng ta có thể lấy dữ liệu từ nitter mà không cần thông qua twitter, giảm thiểu rủi ro bị chặn hoặc hạn chế bởi twitter, tăng hiệu suất và khả năng mở rộng của hệ thống xử lý dữ liệu.



Đoạn code này là một chương trình Java để lấy dữ liệu từ trang web Nitter, một phiên bản giao diện người dùng của Twitter không chứa quảng cáo và theo dõi. Đây là ý nghĩa của từng phần trong đoạn code:

- `public class WebScraper`: Định nghĩa một lớp Java có tên là `WebScraper`.

- `public static void main(String[] args) throws InterruptedException`: Phương thức chính để chạy chương trình.

- `String searchQuery = "Trump";`: Tạo một chuỗi tìm kiếm với giá trị " Trump ".

- `String baseUrl = "https://nitter.salastil.com/";`: Định nghĩa URL cơ bản của Nitter.

- `WebClient client = new WebClient();`: Tạo một đối tượng `WebClient` để tương tác với trang web.

- `client.getOptions().setCssEnabled(false);`: Vô hiệu hóa CSS để tăng tốc độ tải trang.

- `client.getOptions().setJavaScriptEnabled(false);`: Vô hiệu hóa JavaScript để tăng tốc độ tải trang.

- `int count = 0;`: Khởi tạo biến đếm.

- `while (count<10) { ... }`: Vòng lặp để thực hiện quá trình lấy dữ liệu 10 lần.

- Trong vòng lặp:

- `String searchUrl = baseUrl + URLEncoder.encode(searchQuery, StandardCharsets.UTF\_8);`: Tạo URL tìm kiếm bằng cách kết hợp URL cơ bản và chuỗi tìm kiếm đã mã hóa.

- `HtmlPage page = client.getPage(searchUrl);`: Lấy trang HTML từ URL tìm kiếm.

- `List<HtmlElement> items = page.getByXPath("//div[@class='timeline-item ']");`: Lấy danh sách các phần tử HTML chứa tweet từ trang Nitter.

- Kiểm tra và xử lý các tweet:

- Tạo đối tượng `Item` để lưu trữ thông tin tweet.

- Chuyển đổi thông tin tweet thành chuỗi JSON.

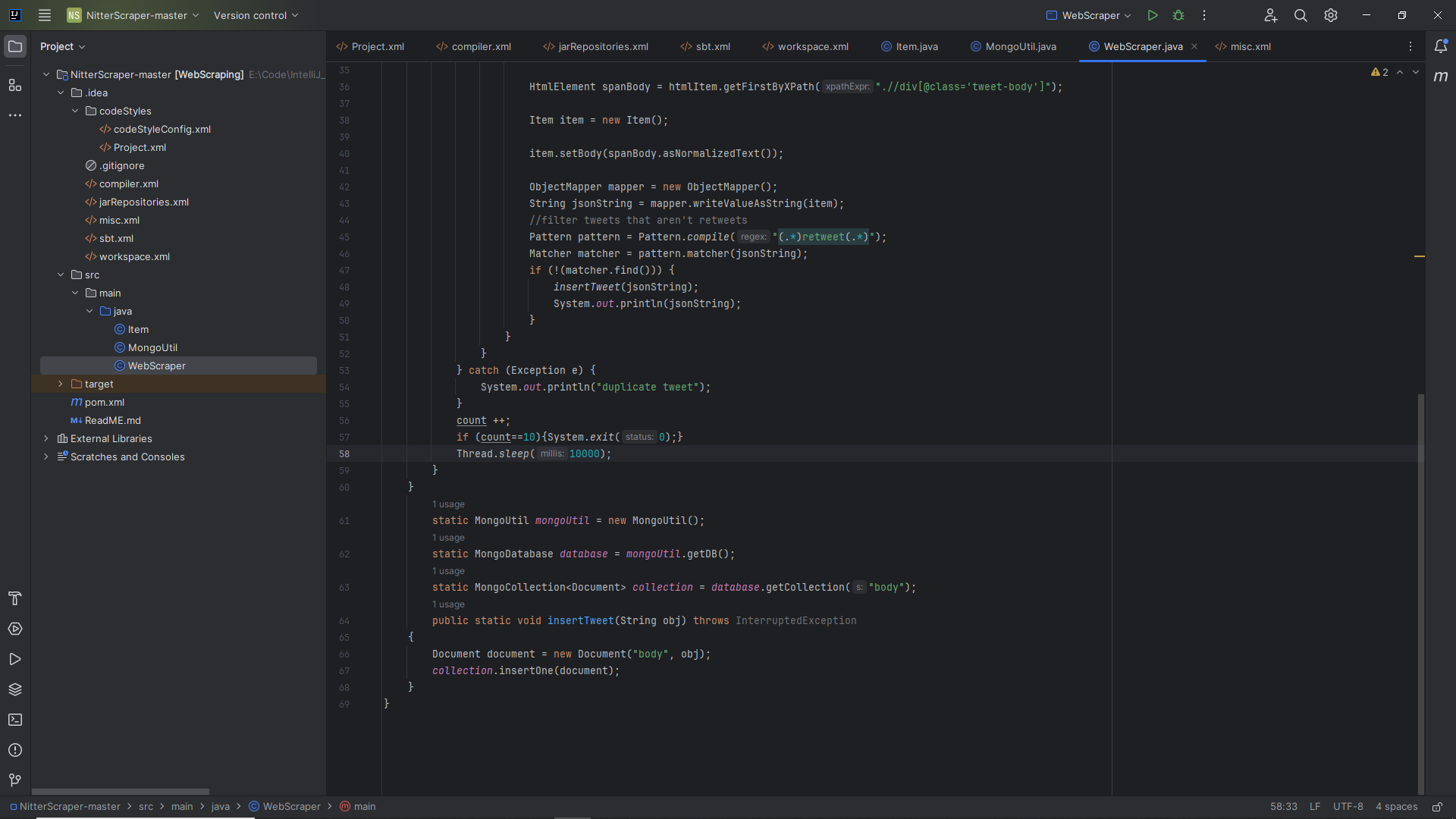
- Lọc các tweet là retweet bằng cách sử dụng biểu thức chính quy (regex).

- Chèn tweet vào cơ sở dữ liệu hoặc hiển thị ra màn hình.

- Xử lý ngoại lệ và tăng biến đếm.

- Dừng chương trình sau 10 lần lặp hoặc sau mỗi lần lặp chờ 60 giây.

Đoạn code này sử dụng thư viện HtmlUnit để tương tác với trang web và thư viện Jackson để xử lý dữ liệu JSON. Mục tiêu của chương trình là thu thập dữ liệu tweet từ Nitter mà không thông qua Twitter, và có thể được sử dụng cho việc phân tích dữ liệu hoặc theo dõi thông tin từ người dùng cụ thể trên Twitter.



static MongoUtil mongoUtil = new MongoUtil();

static MongoDatabase database = mongoUtil.getDb();

static MongoCollection<Document> collection = database.getCollection("body");

public static void insertTweet(String obj) throws InterruptedException {

Document document = new Document("body", obj);

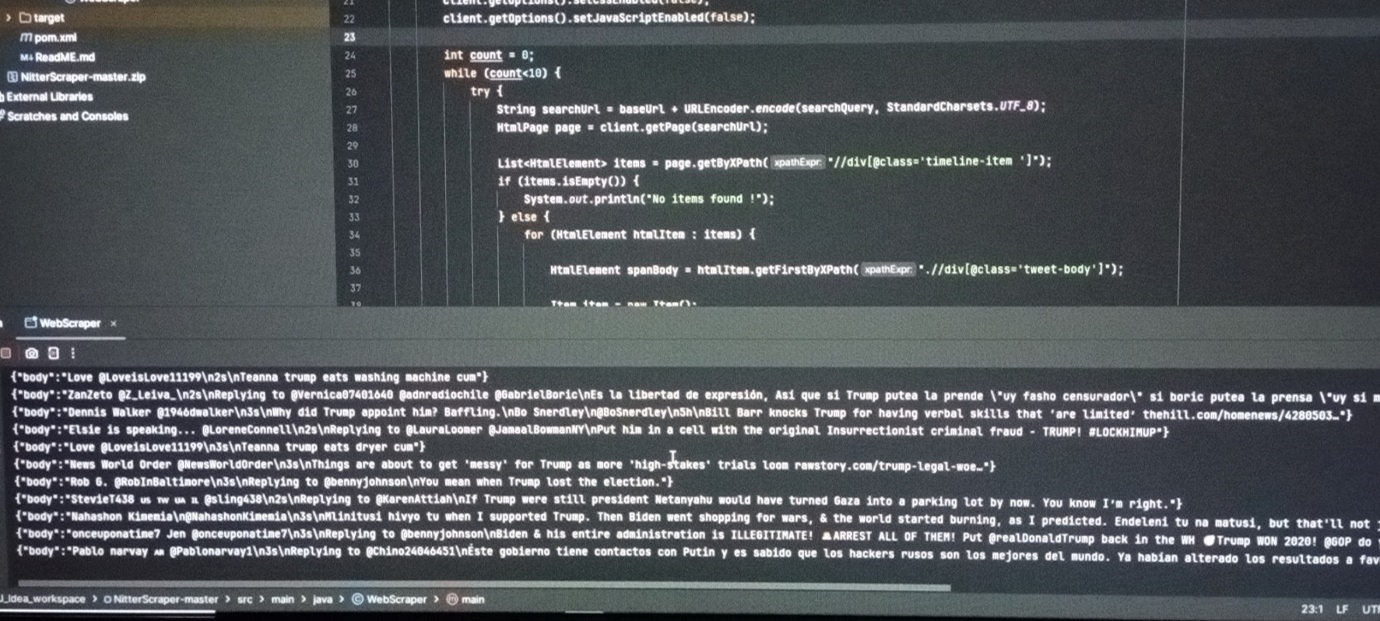
collection.insertOne(document);

}

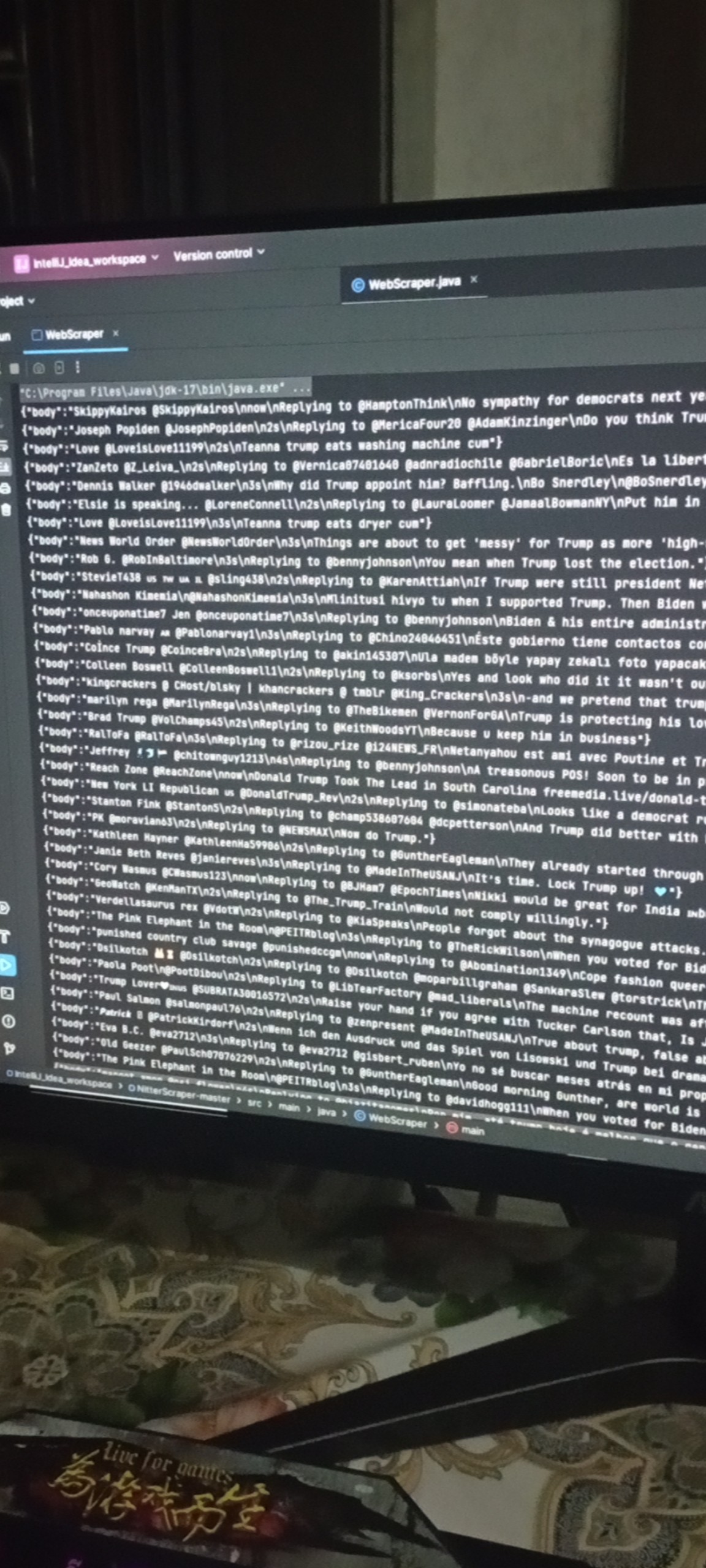
* Dòng 1: Tạo một đối tượng MongoUtil.
* Dòng 2: Lấy ra cơ sở dữ liệu MongoDB.
* Dòng 3: Lấy ra bộ sưu tập MongoDB.
* Dòng 5-8: Tạo một phương thức insertTweet() để chèn một tweet vào bộ sưu tập MongoDB. Phương thức này nhận vào một chuỗi obj là nội dung của tweet. Sau đó, nó tạo ra một đối tượng Document với trường "body" chứa nội dung tweet và chèn nó vào bộ sưu tập MongoDB bằng phương thức insertOne().

Bắt đầu cào dữ liệu : thay đổi tên để truy vấn trong phần : String searchQuery = “…

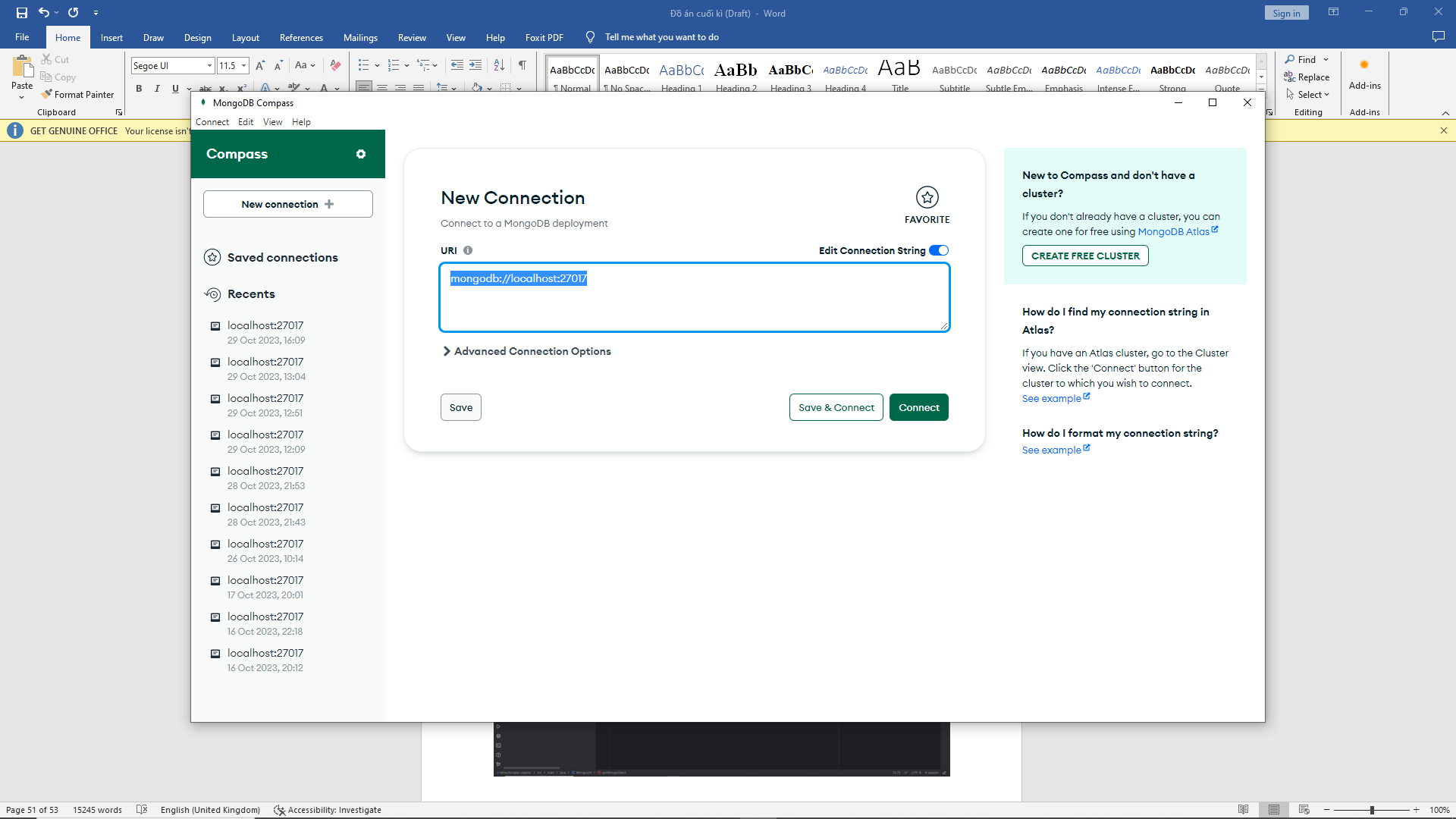
Ở đây chúng em cào dữ liệu có Tweet trên là "Trump";



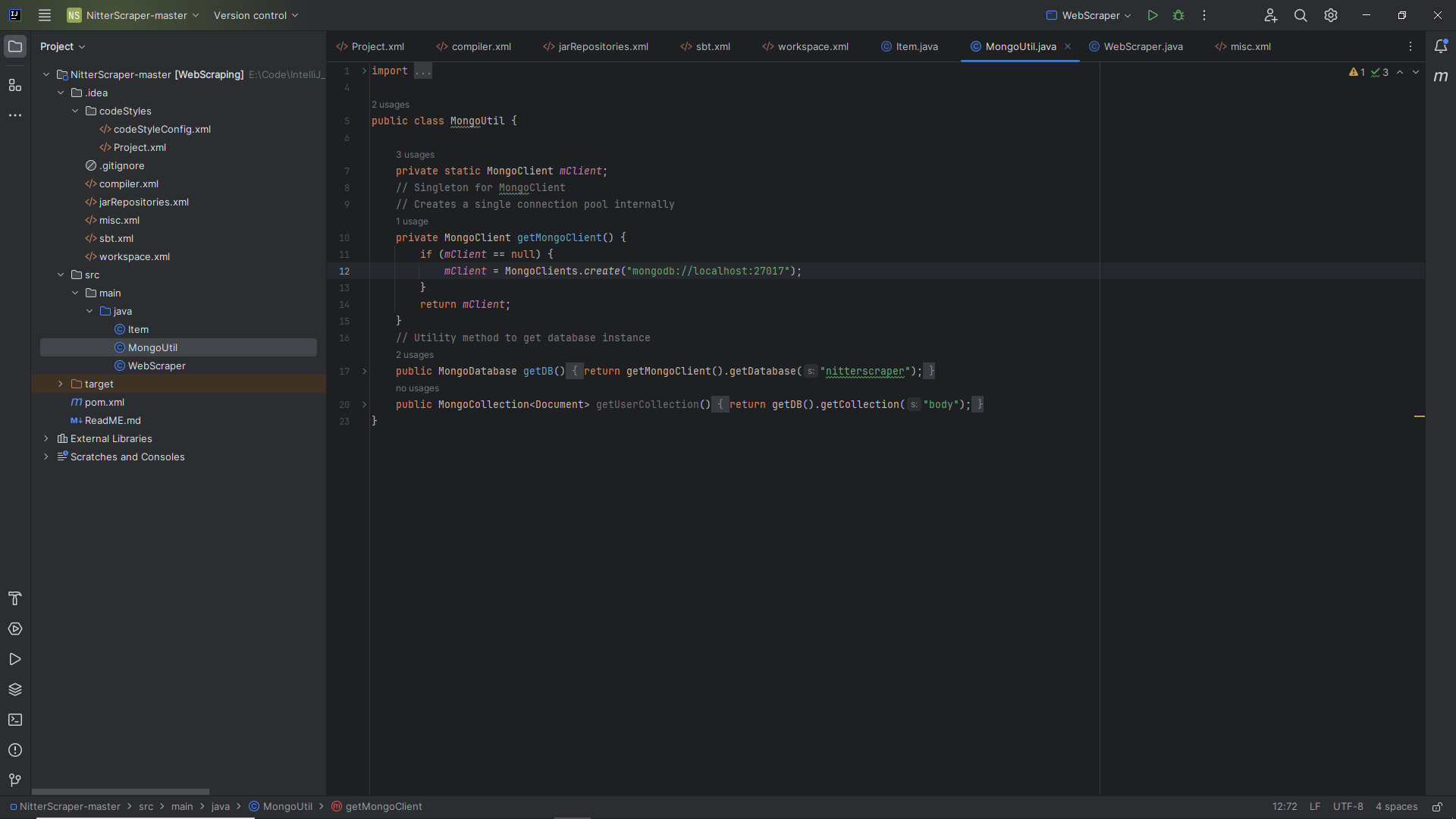
Từng dòng cào sẽ chạy lênh từ từ , trong vòng 1 -2 phút sẽ cào được hàng trăm tweet khác nhau đều có chuỗi “Trump”

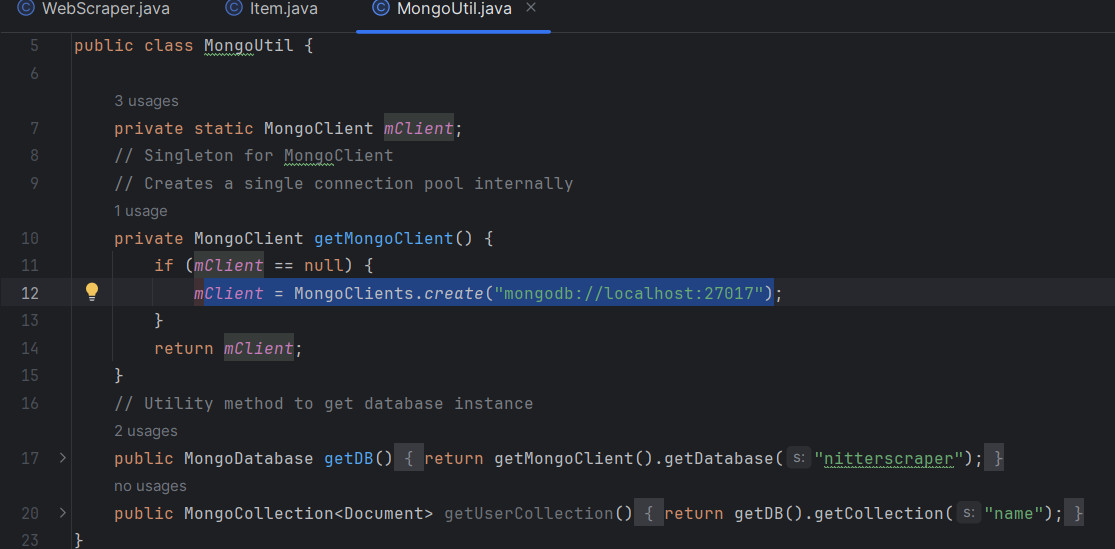


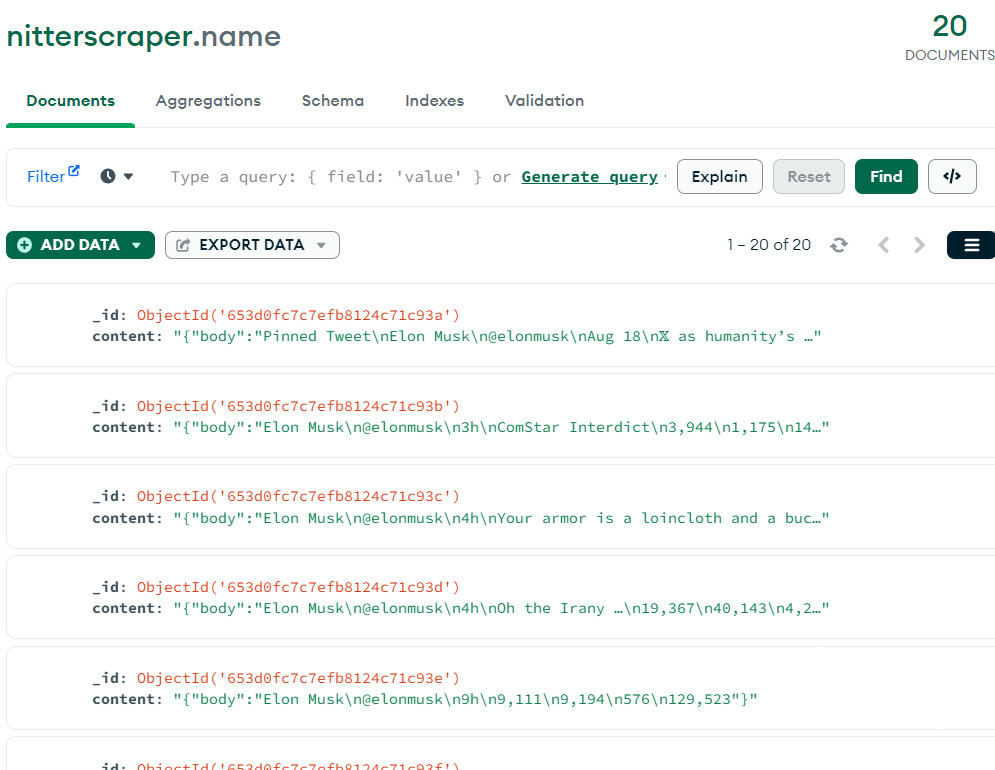
Tiếp theo đó là Lưu dữ liệu

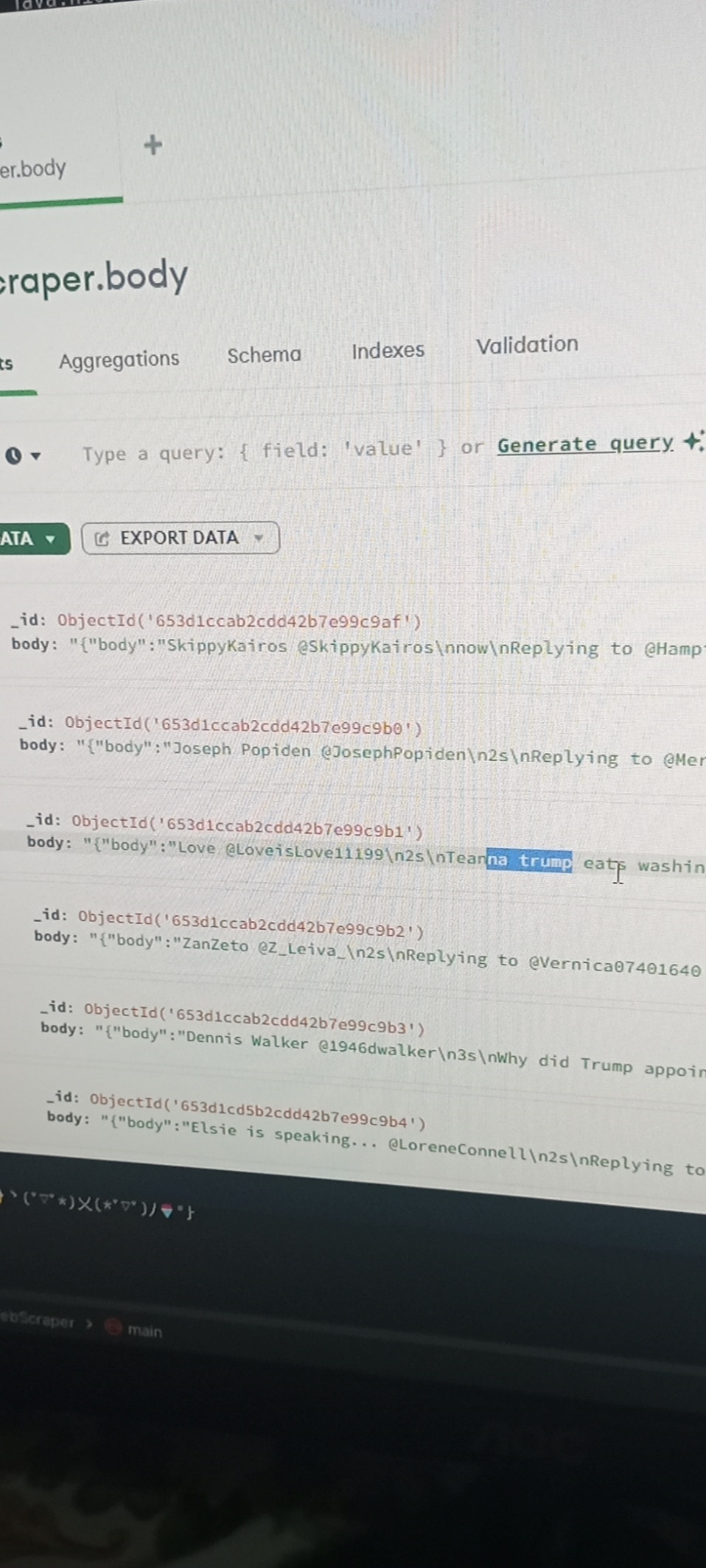
Chúng em đã chọn lưu vào MongoDB . Bằng cách copy đường từ Localhost MongoDB 

Nitter scaper sẽ cào dữ liệu và lưu vào collection mới tạo









Như ở trên tất cả Tweet có “Trump” sẽ được lấy về và đem vào MongodB lưu trữ sau



static MongoUtil mongoUtil = new MongoUtil();

static MongoDatabase database = mongoUtil.getDb();

static MongoCollection<Document> collection = database.getCollection("body");

public static void insertTweet(String obj) throws InterruptedException {

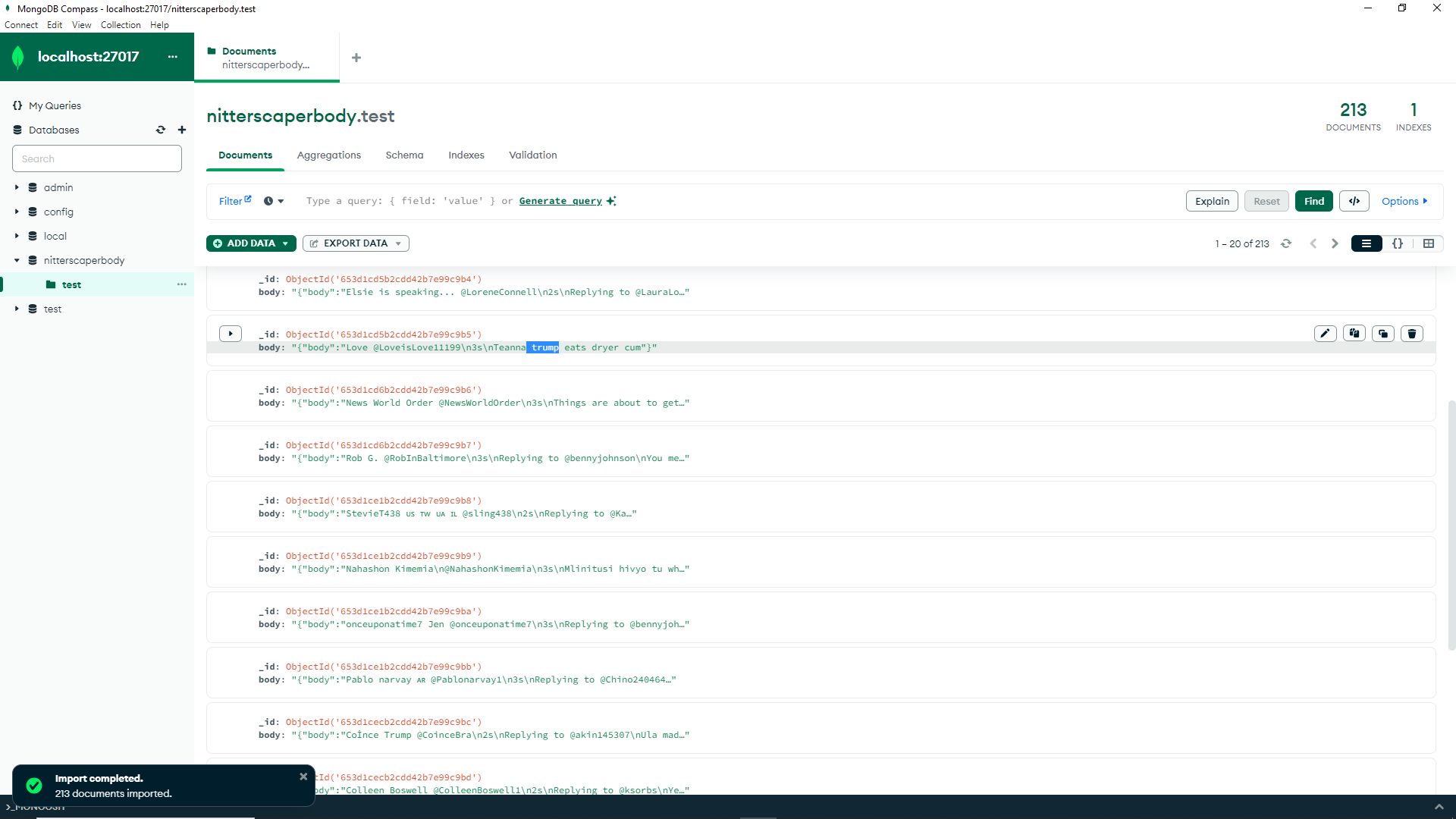
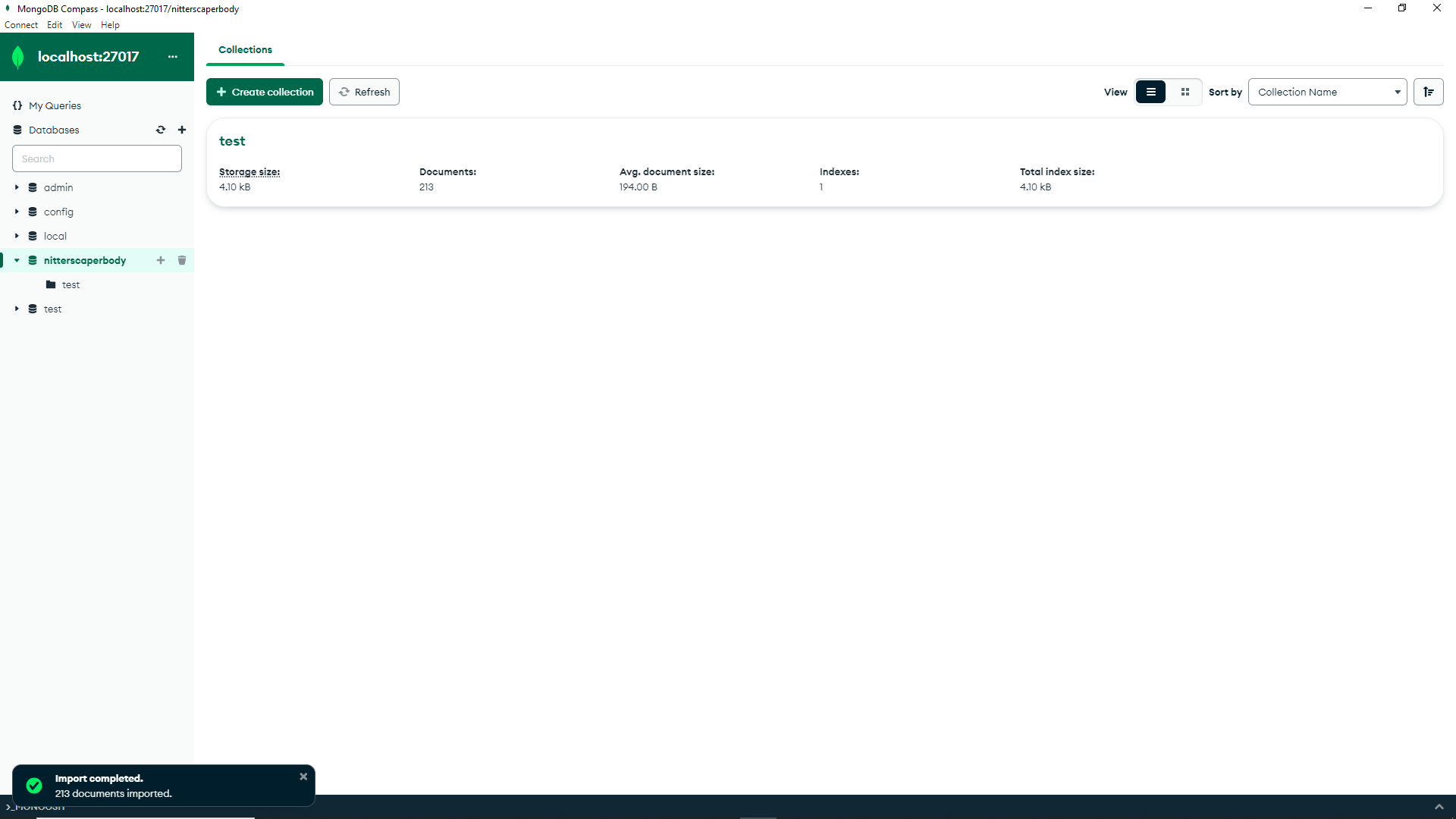
Document document = new Document("body", obj);

collection.insertOne(document);

}

Đoạn mã trên thực hiện các công việc sau:

* Dòng 1: Khai báo lớp MongoUtil.
* Dòng 4-10: Tạo một đối tượng MongoClient để kết nối tới MongoDB server. Nếu đối tượng MongoClient chưa được khởi tạo, phương thức getMongoClient() sẽ tạo ra một đối tượng mới và lưu trữ nó trong biến mClient. Nếu đã có đối tượng MongoClient, phương thức này sẽ trả về đối tượng đó.
* Dòng 12: Tạo một phương thức getDB() để lấy ra cơ sở dữ liệu MongoDB.
* Dòng 16: Tạo một phương thức getUserCollection() để lấy ra bộ sưu tập MongoDB chứa tweet.

**Nguồn tham khảo :**

* <https://www.tutorialspoint.com/apache_kafka/apache_kafka_cluster_architecture.htm>
* <https://stackjava.com/intellij-idea/intellij-idea-la-gi-cai-dat-intellij-idea-tren-windows.html>
* <https://vihoth.com/phan-mem-van-phong/jetbrains/phan-mem-intellij-idea-java-idea-thong-minh-nhat>
* <https://viblo.asia/p/hang-doi-thong-diep-apache-kafka-jvEla6145kw>
* <https://phoenixnap.com/kb/install-spark-on-windows-10>
* <https://viblo.asia/p/tim-hieu-ve-apache-spark-ByEZkQQW5Q0>
* <https://200lab.io/blog/apache-spark-la-gi/>
* <https://blog.vu-review.com/kafka-la-gi.html>
* <https://www.cloudkarafka.com/blog/2016-11-30-part1-kafka-for-beginners-what-is-apache-kafka.html>
* [https://www.facebook.com/notes/cộng-đồng-big-data-việt-nam/tiếp-cận-kafka-thông-qua-confluent-platform/884414712076055/](https://www.facebook.com/notes/c%E1%BB%99ng-%C4%91%E1%BB%93ng-big-data-vi%E1%BB%87t-nam/ti%E1%BA%BFp-c%E1%BA%ADn-kafka-th%C3%B4ng-qua-confluent-platform/884414712076055/)
* <https://data-flair.training/blogs/kafka-architecture/>
* <https://bigdataviet.wordpress.com/2015/08/08/hadoop-la-gi/>
* <https://duythanhcse.wordpress.com/2021/01/01/cai-dat-hadoop-tren-windows/>
* <https://cloudgeeks.net/tim-hieu-ve-kien-truc-hadoop-ecosystem/>
* <https://www.youtube.com/watch?v=do8w5Txpq34>
* <https://www.youtube.com/watch?v=rv9A7Y8qLOI>
* <https://niithanoi.edu.vn/bo-cong-cu-big-data-huu-ich-ban-nen-biet.html>
* <https://viblo.asia/p/tim-hieu-ve-hadoop-bJzKmOBXl9N>
* <https://viblo.asia/p/kafka-apache-WAyK8pa6KxX>