Báo Cáo viết chương trình Crawler data

1. Crawl data là gì

Crawl data là quá trình **thu thập dữ liệu** của công cụ tìm kiếm nhằm tìm **nội dung mới** hoặc cập nhật **những thay đổi** trên trang cũ. Những định dạng được thu thập dữ liệu gồm: html, hình ảnh, video…

Đầu tiên, **Crawl dữ liệu**(quá trình crawling) diễn ra khi công cụ tìm kiếm ghé qua website của bạn. Sau đó, Search Engine mới thực hiện quá trình **Indexing**(Lập chỉ mục).

Nếu bạn chưa đọc về toàn bộ các quá trình mà công cụ tìm kiếm thực hiện đối với website, mời bạn đọc qua **cách hoạt động của Search Engine**.

2. Web Crawler là gì

Gắn liền với quá trình thu thập dữ liệu thì bạn cũng nên biết đến "công nhân" thực hiện hoạt động này: **Web Crawler**.

Web Crawler (trình thu thập web) là một **bot internet** thực hiện **thu thập dữ liệu qua World Wide Web**. Crawler được công cụ tìm kiếm lập trình sẵn nhằm mục đích lập chỉ mục cho các nội dung thu thập được.

Trình thu thập thông tin còn có các tên gọi khác là spider, spiderbot… Nhưng cách mọi người thường gọi nhất vẫn là **[Tên công cụ tìm kiếm + bot]**. Chẳng hạn như: Googlebot, Bingbot, Yandexbot…

3. Quá trình crawl data của trình thu thập

Khi bạn nắm các khái niệm cơ bản thì không quá khó để hiểu **quá trình thu thập dữ liệu**. Quá trình này được diễn ra như sau:

1. Crawling được bắt đầu khi công cụ tìm kiếm (Search Engine – SE) phát hiện một liên kết.
2. Dựa vào liên kết, SE sẽ khởi động trình thu thập web để thu thập thông tin của trang đích.
3. Trong trang đích này, chúng sẽ phát hiện những liên kết mới. Crawler sẽ nhân đôi để quá trình thu thập trang hiện tại vẫn được diễn ra với 1 lượt crawl data. Trình thu thập web còn lại sẽ sang trang đích của các liên kết khác.
4. Quá trình này được lặp đi lặp lại liên tục.

Tuy nhiên, điều này sẽ tiêu tốn rất nhiều tài nguyên của SE (quá tải về lưu lượng và dung lượng). Do đó, Search Engine cập nhật những **nguyên tắc hoạt động cho web crawler** (thuật toán).

Nguyên tắc mà bạn cần quan tâm nhất trong bài viết này là:

Nếu trang có hơn 1 liên kết đến cùng 1 trang đích, trình thu thập web **chỉ thu thập một lần từ link đầu tiên nó phát hiện**.

Ở đây bạn có thể hiểu:Bạn có thể đặt bao nhiêu internal link (**liên kết nội bộ**) tuỳ thích. Nhưng duy nhất chỉ 1 link đầu tiên có giá trị.
Quá trình này được giới hạn và mỗi website có một ngân sách thu thập dữ liệu (crawl budget) khác nhau. Trong bài viết này TIEN ZIVEN sẽ hướng dẫn cách nâng cao hiệu suất của mỗi lần crawl data. Còn về cách tối ưu ngân sách Cào sẽ được nói chi tiết trong bài viết **Crawl Budget là gì**?

4. Tại sao cần tối ưu và Cách tối ưu quá trình crawl dữ liệu

Tối ưu crawl data là quá trình giúp trình thu thập web lấy được nhiều thông tin nhất trong một lần cào.
Quá trình này vô cùng quan trọng vì:

- Giúp nâng cao hiệu suất trong một lần thu thập dữ liệu của Web crawler.
- Tạo điều kiện để công cụ tìm kiếm hiểu nội dung tốt hơn.
- Công cụ tìm kiếm sẽ đánh giá chất lượng nội dung và thực hiện quá trình lập chỉ mục.

Tài liệu tham khảo:

- Sách
  1. **"Web Scraping with Python: Collecting Data from the Modern Web"** của Ryan Mitchell
     - Một hướng dẫn toàn diện về web scraping với Python, bao gồm cả BeautifulSoup và Scrapy.
  2. **"Data Mining: Concepts and Techniques"** của Jiawei Han, Micheline Kamber và Jian Pei
     - Cung cấp cái nhìn tổng quan về khai thác dữ liệu và các phương pháp phân tích dữ liệu.
  3. **"Python for Data Analysis"** của Wes McKinney
     - Tập trung vào sử dụng Python và thư viện Pandas để phân tích dữ liệu, cũng như cách thu thập dữ liệu từ web.
- Tài liệu trực tuyến
  1. **BeautifulSoup Documentation**
     - Tài liệu chính thức cho thư viện BeautifulSoup: BeautifulSoup Docs
  2. **Scrapy Documentation**
     - Tài liệu chính thức cho Scrapy, một framework mạnh mẽ cho web scraping: Scrapy Docs
  3. **Selenium Documentation**
     - Tài liệu cho Selenium, thường được sử dụng để tự động hóa trình duyệt và lấy dữ liệu từ các trang web động: Selenium Docs
  4. **Requests Documentation**
     - Tài liệu cho thư viện Requests, giúp gửi các yêu cầu HTTP một cách dễ dàng: Requests Docs
- Khóa học

1. **Web Scraping with Python and BeautifulSoup on Udemy**
   - Khóa học chi tiết giúp bạn hiểu về cách scraping dữ liệu với Python và BeautifulSoup.
2. **Scrapy for Beginners on Udemy**
   - Khóa học hướng dẫn bạn cách sử dụng Scrapy để thu thập dữ liệu từ web.
3. **Data Science: Web Scraping with Python on Coursera**
   - Khóa học này cung cấp kiến thức cơ bản về web scraping và phân tích dữ liệu.

- ## Hướng dẫn và bài viết
  1. **Web Scraping Using Python: A Step-by-Step Guide**
     - Một hướng dẫn từng bước để bắt đầu với web scraping: Real Python
  2. **A Complete Guide to Web Scraping with Python**
     - Bài viết toàn diện về web scraping với Python, bao gồm các thư viện khác nhau: DataCamp
  3. **Web Scraping with Selenium in Python**
     - Hướng dẫn sử dụng Selenium để scraping dữ liệu: GeeksforGeeks
  4. **Handling AJAX Requests in Web Scraping**
     - Hướng dẫn cách xử lý các yêu cầu AJAX khi scraping dữ liệu từ các trang web: Medium Article
- ## Diễn đàn và cộng đồng
  1. **Stack Overflow**
     - Một nơi tuyệt vời để đặt câu hỏi và tìm kiếm giải pháp liên quan đến web scraping: Stack Overflow
  2. **Reddit - Web Scraping Subreddit**
     - Một cộng đồng trên Reddit nơi bạn có thể thảo luận về web scraping và nhận được trợ giúp từ những người khác: Reddit Web Scraping
  3. **GitHub**
     - Nơi bạn có thể tìm thấy nhiều dự án mã nguồn mở liên quan đến web scraping: GitHub Search
- ## Công cụ và thư viện
  1. **Pandas**
     - Thư viện mạnh mẽ để xử lý và phân tích dữ liệu sau khi đã scraping: Pandas Documentation
  2. **lxml**
     - Thư viện để xử lý XML và HTML nhanh chóng và hiệu quả: lxml Documentation
  3. **Regex (Regular Expressions)**
     - Công cụ mạnh mẽ để tìm kiếm và xử lý văn bản trong dữ liệu scraping: Regex Tutorial