Master's Thesis

Design and Implementation of a Configurable Generic Search Engine Indexing using Scalable Crawlers

Alhajras Algdairy

Examiner: Prof. Dr. Hannah Bast

Advisers: M. Sc. Natalie Prange

Albert-Ludwigs-University Freiburg

Faculty of Engineering

Department of Computer Science

Chair of Algorithms and Data Structures

April 25th, 2021

Writing Period

01. 12. 2020 – 25. 04. 2021

Examiner

Prof. Dr. Hannah Bast

Second Examiner

Prof. Dr. Thomas Brox

Advisers

M. Sc. Natalie Prange

Declaration

I hereby declare, that I am the sole author and composer of my thesis and that no other sources or learning aids, other than those listed, have been used. Furthermore, I declare that I have acknowledged the work of others by providing detailed references of said work.
I hereby also declare, that my Thesis has not been prepared for another examination or assignment, either wholly or excerpts thereof.

_____ _____

Place, Date Signature

Abstract

Web search indexing is an essential system that powers modern search engines. It automates the process of collection and organization of data from web pages to create an updated index of the web that can be optimally searched. Web search indexing consists of two essential components, a web crawler, in which search engine bots systematically traverse the web to find new or updated content based on rules declared beforehand, followed by the second component which is the indexing of the collected data. The process of web search indexing comes with its own challenges, including performance, managing dynamic content, and answering the question of what is the most relevant content. As the web continues to evolve and grow, the task of web search indexing will remain a key focus of search engine technology and research.The aim of this thesis is to design and implement a generic configurable web search indexing that can be used as a basic tool on different websites and can be further expanded and improved, and scaled. The approach included a simple UI design that allows users to configure and create crawlers and index the generated data.

Acknowledgments

First and foremost, I would like to thank:

- My parents for supporting me during the master's program.
- My wife for her love and support.
- Prof. Dr. Hannah Bast for accepting my topic and for her guidance and supervision.
- M. Sc. Natalie Prange for her thoughtful ideas and suggestions.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

Since the beginning of the Digital Revolution, known as the Third Industrial Revolution, in the latter half of the 20th century, the importance of data has increased as it became the new currency shaping the dynamics of our interconnected world. From social media platforms and e-commerce transactions to information sharing and entertainment consumption, online activities generate enormous amounts of data. The online data is sometimes referred to as the 'new oil' or the 'new currency', as it impacts almost the same economies and societies as oil. Businesses and organizations understand the power of data as they provide insight into consumer behaviour, refines business strategies, and enhance decision-making processes. Furthermore, the rise of artificial intelligence has further amplified the value of Internet data. Natural language processing (NLP) is becoming a new hot topic as all the giant firms race to create their model; however, data is the fuel to power those models. The more data is collected, the better the model can become. Consequently, collecting, analyzing, and leveraging internet data has become a cornerstone of competitiveness, innovation, and progress in the digital age.

The Internet data can be harvested by using automated software programs called Web crawlers, also known as web spiders or web bots. Their main goal is to discover, retrieve, and index information from websites.

Internet data can be harvested by using automated software programs called Web crawlers, also known as web spiders or web bots. Their main goal is to discover, retrieve, and index information from websites. The applications and use cases of internet crawlers are diverse and valuable, to name a few:

- Search Engines: Crawlers are essential components to build any search engine, such as Google, Bing, and Yahoo. Crawlers are run on supercomputers to crawl all the content on the internet index web pages and gather information about content, keywords, and links. This data is then used to rank and display search results, ensuring users can quickly find relevant information.

- Market Research: Businesses use web crawlers to collect data about their competitors, market trends, and consumer opinion. This information helps in making informed business decisions.

- Fraud Detection: Cybersecurity companies use crawlers to catch fraudulent activities by monitoring online transactions, identifying unusual patterns, and tracking potential threats.

- Content Monitoring: E-commerce platforms utilize crawlers to extract product prices from various websites. This enables them to offer consumers real-time price comparisons and assist in finding the best deals. Moreover, social media platforms use crawlers to monitor their content to prevent unwanted posts and images.

1.2 Problem Statement

The World Wide Web (WWW) contains enormous data that escalates with each passing day. The total data created and replicated is expected to grow to more than 180 zettabytes by 2025 according to Statista. This upward trajectory is expected to

continue due to the growing affordability of smartphones and the broadening reach of internet accessibility. Moreover, due to the COVID-19 pandemic, more companies started offering remote work, more local shops transformed into online stores, and more services switched to cloud-based. This social evolution over recent years has embedded the Internet as an integral cornerstone of our daily life.

The expansion of the Internet gives rise to an immense overflow of data, resulting in a noise that complicates the task of locating relevant information for both end users and organizational queues. To surmount this hurdle, the concept of Information Retrieval (IR) was coined by Calvin Mooers in 1951. IR involves the art of accessing and recovering data from an extensive pool of unstructured information. A particularly pragmatic manifestation of IR involves the extraction of data from the Internet, thus advocating the implementation of a universal algorithm for procuring and categorizing requisite information. In this pursuit, crawlers or spiders emerge as automated entities designed to adhere to predefined directives, allowing the automated fetching and extracting of data from the Internet.

One form of IR is a web search engine. A web search engine is a system engineered to index the Internet. Users can search for articles, documents and pages by entering keywords. The search will provide a list of the most related result that matches the search query. Using the crawlers explained earlier; the engine can index the collected information and optimize the search process using different algorithms and techniques.

Although search engines such as Google, Bing and DuckDuckGo display remarkable proficiency in their web crawling and indexing capabilities, specific businesses, like those in E-commerce, have a distinct interest in demonstrating the most competitive pricing for a given product, a key insight into their competitive landscape. However, more than a straightforward Google search is needed, as the search query index and rank the documents on the Internet based on Google's vendor parameters. These parameters include preeminent brand visibility, user geolocation, SEO optimization

proficiency, and hidden variables, excluding the lowest price criterion from the page ranking equation. A second issue arises from the format of search results, with each search engine providing a distinctive result format. Companies may want to exclude some portion of the Internet from the index and rank. Also, they should prioritize some pages more than others.

The previous requirements are tiny use case, among others, that limits companies from using a simple search on Google. Search engines need to be tweaked and configured to match the domain of interest as E-commerce in the previous example and to match a specific use case like the price comparison mentioned.

The main concern is that businesses are often interested in only a portion of the Internet that intersects with their domain and expertise. Furthermore, the criteria for indexing and page ranking depend heavily on their use case and is vital to their business to take control of it and configure it as fit. Hiring domain expertise is inevitable for any business. However, the data scientists often have to go throw some basics steps to get their crawler up and running; those steps cost money and time; it would be helpful to have an infrastructure that allows the data scientist to have starting script that can be extended easily and needs little to no programming knowledge.

Amount of data created, consumed, and stored 2010-2020, with forecasts to 2025 Published by Petroc Taylor , Sep 8, 2022 https://www.statista.com/statistics/871513/worldwide-data-created/: :text=The

1.3 Contribution

In this thesis, we aim to answer the following questions:

- What are the challenges and bottlenecks to creating a scalable, configurable generic search engine?
- Can we implement a basic tool that can be easily scaled to crawl different websites independently from their DOM structure?
- Can we create a proper User Interface UI that allows users to crawl and index targeted websites from the internet?
- Can we integrate the indexing and crawling processes in the same tool?
- Can we find meaningful evaluation metrics for the implemented search engine?

1.4 Chapter Overview

2 Related Work

The concept of web crawling dates back to the early 1990s when the World Wide Web was still in its infancy.

WebCrawler, created by Brian Pinkerton in 1994, is considered the first true web crawler-powered search engine. While some may claim that title for Wandex is due to its potential, it was never designed to be used in this way. Wandex lacked some critical features to make it a general-purpose search engine.

One of the major innovations of WebCrawler was its full-text searchability. This ability made it popular and highly functional. It continues to operate as a search engine, although not as popular as Google, Yahoo, Bing, Yandex, or Baidu.

Modern web crawlers face many challenges and complexities, such as dynamic content, user interaction, authentication, robots.txt files, and ethical issues. Some examples of modern web crawlers are Googlebot, Bingbot, and Internet Archive

2.1 High-Level-Google-Architecture

- Talk about google solution here

The Google search engine's structure comprises distinct elements, depicted in Figure 1. Google's services operate across multiple Server Farms, employing the Linux operating system and utilizing programming languages such as C, C++, and Python

for server-side coding [9, 10]. In 2006, it was approximated that Google possessed over 450,000 servers across the globe [6]. Google acquires web pages through a program named Web Crawler (Web Spider or Web Robot), which traverses the Internet, expanding the list of web pages. Alternatively, web pages can be directly added through Google's site. Google employs the "Googlebot" as its web crawler program. The URL Server transmits a roster of Uniform Resource Locators (URLs) to the Googlebot, each assigned a unique identifier known as a "docID" for reference. Subsequently, these web pages are directed to the "Store Server," where compression occurs before storage in a "Repository." The "Indexer" and "Sorter" collaborate during the indexing process. Documents within the "Repository" are accessed by the "Indexer," which decompresses, parses, and transforms them into a series of word occurrences termed "hits." These hits are categorized and dispatched to storage units referred to as "Barrels." The "Indexer" also identifies keywords within the web pages. Managing the storage of document IDs, locations within the "Repositories," statuses, content, and URL titles is the responsibility of the "Doc Index" component. The "Lexicon" element houses an extensive compilation of words and associated pointers. Google employs the "PageRank" algorithm to assess and rank pages based on relevance. This algorithm evaluates pages by considering hits within the pages and their positions, including hits within the title and body sections, among other factors [9].

The Anatomy of a Large-Scale Hypertextual Web Search Engine (google paper)

"Effective web crawling"

"Introduction to Information retrievel"

3 Background

3.1 History

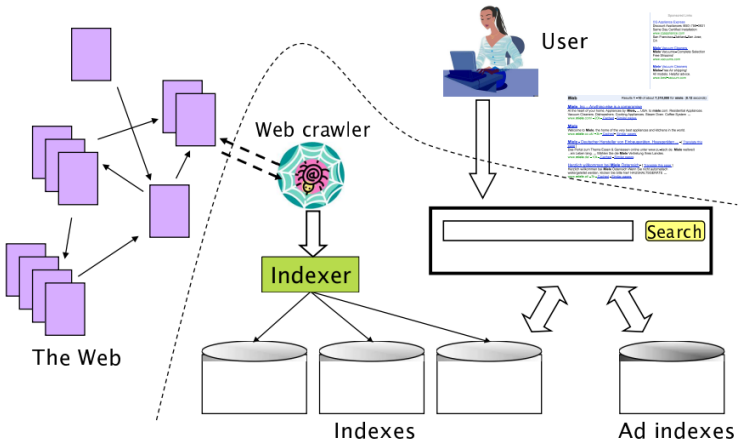
The World Wide Web is an unlimited space to share provide and share information. Those information can have different format and cover different doamins. The use case of the web is only limtied by the developers imagination. This is benifital as the Web keept evolving rapidaly form Web 1.0 to Web 2.0 to Web 3.0. Web 1.0 used static pages to serve information, those information were moslty news, blogs and personal langing pages. Some refre to the Web 1.0 as "the read-only web". Although Web 1.0 was massive however most content were created was by deverlopers or at least users who knew basics of the HTML and CSS, moreover by that time content were only static they did not depen on fancy JavaScript libraries and frameworks like Angular and React, this made it limited to some use cases only. Fast forward, pages become more dynamic after using sessions, databases and clint rendering schemas. Those changes made the Web focused not only reading and gathering information by gave the power to more audiounce who did not know any programming or coding to participate and interact with the Web via browsers. Social media, e-commerce and trading stocks platforms was one of the reasons made the internet buble inflate, Use cases where unlimited as useres could create and deploy their own websites by using

simple tools as Content Manament System CMS. This made Web 2.0 known as "the participative social web".

3.2 Web Search Engine

Web Search Engine is a software that collect information from the web and index them effecianly to obtimipze the searching process by the user. Users enter their queries to ask for information, the engine carries out the queries and lookup the pre built orginized index and return a relevant results. The returned result is presented by Search Enngine Results Pages as known as SERPs. The result then ranked based on predefined cretieria.

Web search engines use web crawler or spider to collect and harvest the intenret jumping from one page to another. Each page can contain several links, the crawler task is to find the links and to visit them and harvest them also. Followed by crawlers, indexing is the next process where information are orginized and optimized for search.



3.3 Cralwer

Web crawler or spidedr is a software which gather pages information from the web, to prived the neccasary datat to the indexer to build a searhc engine. The essintial role of crawlers is to effecitnalt and reliably collect as much infromation from the web.

3.4 Cralwer Specifications

Crawlers can have a wide vireity of features and specifications, however some are necessary to include and others are vital to have a reliable useable one.

- Robustness: Web crawler can be fragile and easy to break, this is due to the nature of the dynamic contnets on the web and the internet connection. Web crawlers must identify those edge cases and obsticals and tackle them.

- Politeness: The implmentation of the crawler can be unintentially mellisous and dangerous if not designed correclty. A Deniel of service DoS and a Distributed Denial of service DDoS attacks can occure due to a bad crawler implmentation. Hence crawlers must respect websites policies and avoid breaking up web services and load the servers.

- Distributed: To make the crawler efficint and

  The crawler should have the ability to execute in a distributed fashion across multiple machines.

Scalable: The crawler architecture should permit scaling up the crawl rate by adding extra machines and bandwidth. Performance and efficiency: The crawl system should make efficient use of various system resources including processor, storage and network bandwidth. Quality: Given that a significant fraction of all web pages are of

poor utility for serving user query needs, the crawler should be biased towards fetching "useful" pages first. Freshness: In many applications, the crawler should operate in continuous mode: it should obtain fresh copies of previously fetched pages. A search engine crawler, for instance, can thus ensure that the search engine's index contains a fairly current representation of each indexed web page. For such continuous crawling, a crawler should be able to crawl a page with a frequency that approximates the rate of change of that page. Extensible: Crawlers should be designed to be extensible in many ways – to cope with new data formats, new fetch protocols, and so on. This demands that the crawler architecture be modular.

3.5 Indexing

2.4 Web crawling issues page 27 book Effective Web Crawling by Carlos Castillo

4 Approach

The approach usually starts with the problem definition and continues with what you have done. Try to give an intuition first and describe everything with words and then be more formal like 'Let g be ...'.

4.1 Problem Definition

Start with a very short motivation why this is important. Then, as stated above, describe the problem with words before getting formal.

4.2 First Part of the Approach

4.3 N-th Part of the Approach

5 Datasets

6 Experimental Evaluation

7 Summary of Results

8 Conclusions and Future Work

Bibliography

[KimathiKimathi2020] Kimathi2020Kimathi, G. 2020June. What Is Ray Tracing Technology and How It Works in GPUs. What is ray tracing technology and how it works in gpus. https://www.dignited.com/62084/how-ray-tracing-works