How to Build a Good Web Crawler: Crawling Strategies and How to Crawl a Website without Being Blocked

Zhaokang Li (zl52)

1. Introduction

Nowadays, it has become an important part of human life to access the information from WWW (World Wide Web). Every user depends on the search engine to complete his desire to get information. Search engine uses the crawlers to get more relevant information. In reality, web crawler is a program, which automatically traverses the web by downloading documents and following links from page to page. They are mainly used to gather data from internet. This paper aims to show how to build a good web crawler. First, it talks about how web crawler works and compares different crawling strategies. Then it analyzes how to crawl a website without being blocked.

2. Working of Web Crawler

2.1 Crawler Architecture

Web crawler works beginning with an initial set of URLs which are called seed URLs. It downloads web pages for seed URLs and then extracts new hyperlinks in downloaded pages. The retrieved web pages are stored and indexed so that they can be retrieved when required later. The extracted URLs from downloaded page will be confirmed whether their documents have been downloaded. If they are not downloaded, the URLs are added to web crawlers for downloading. This process will be repeated until no more URLs need to be downloaded. Figure 1 illustrates this crawling process.

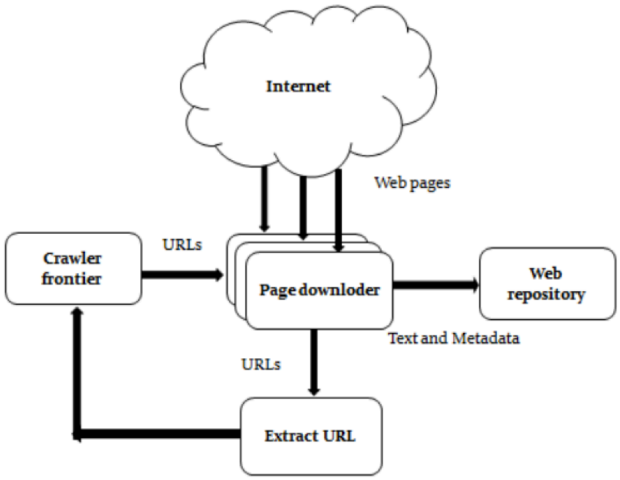

Figure 1 Web Crawler Architecture[1]

2.2 Crawling Strategies

So we can see that the major function of a web crawler is to insert new links into the frontier and to choose a fresh URL from the frontier for further processing. There are many strategies for how to choose a URL from the frontier. Different strategies can be used efficiently in different conditions. This section will introduce several general crawling strategies and give a comparison between them.

2.2.1 Breadth First Crawling

The Breadth First search algorithm performs the unique search around the neighbor nodes(hyperlinks). It starts by following the root node (Hyperlink) and scans the all the neighbor nodes at the initial level. If the targeted search is achieved, then the scanning is stopped otherwise it leads to the next level.

This type of algorithms is best suited where the branches are small and resultant objective is identical. When the branches or tree is very deep then this algorithm will not perform well.

2.2.2 Depth First Crawling

The Depth First search algorithm starts searching the objective from the root node and traverse next to its child node. If there are more than one child node, then left most node is given highest priority and traverse deep until no more child node is present. Then it starts from the next unvisited node and then continues in a similar manner.

By using this algorithm, the assurance of scanning of all node is achieved but when the number of child node is large then this algorithm takes more time and might go in to

infinite.

2.2.3 Targeted Crawling

Some search engines use random crawling process in order to target a certain type of page, e.g. pages on a specific topic or in a particular language. In addition to these heuristics, more generic approaches have been suggested. They are based on the analysis of the structures of hypertext links and techniques of learning: the objective here is being to retrieve the greatest number of pages relating to a particular subject by using the minimum bandwidth.

2.2.4 Page Rank Algorithm

This algorithm works on the importance of the web pages. It calculates inlinks or backlinks to that page. Then the page rank is given to each page as per bellow formula,

$$\mathrm{PR(A)} = (1-d) + d(PR(T1)/C(T1) + \cdots + PR(Tn)/C(Tn)),$$

where $\mathrm{PR(A)}$ is PageRank of site, d is damping factor, $\mathrm{T1}, \ldots \mathrm{Tn}$ is the number of links. After determining a page rank of website, the index has been generated to show the relevant on a web site contain to the search keywords.

2.2.5 Comparison

The comparison between these four strategies is given below:

Strategy	Search Pattern	Advantages	Disadvantages
Breadth-First Crawling	Scans neighbor node from root level, if result not achieved then go to next level.	When branches are small and resultant, the objective is identical.	When branches or tree is very deep, it goes into infinite.
Depth-First Crawling	Scan from the root node and traverse next to its child leftmost node.	Assurance of scanning of all node	Take more time when the child node is large.
Targeted Crawling	Use heuristics crawling process	Retrieve the greatest number of pages relating to a particular subject by using the minimum bandwidth.	Takes more time when specific topics are very large.
Page Rank Algorithm	Work on the importance of the web pages. It calculates in-links or backlinks to that page.	More accurate search result.	Difficult to manage and update page index repository.

3. How to Avoid Getting Blacklisted While Scraping

In practice, web crawling is supposed to be performed responsibly, otherwise it will have detrimental effects on the scraped sites. If a crawler is performing multiple requests per second and downloading many large files, the server would be hard to keep up with requests from multiple crawlers. This section will introduce how websites detect spiders and analyze how to crawl a website without being blocked.

3.1 Identity

Every request made from a web browser contains a user-agent header. Therefore, using the same user-agent consistently can lead to the detection of a bot. User Agent spoofing is a good solution for this. Spoof the User Agent by creating a list of user agents and pick a random one for each request. Websites do not want to block normal users so we should try to look like one.

3.2 Robot Exclusion

Robot exclusion is a general way ton control web spider's access. The robots.txt file specifies rules for good behavior, such as how frequently bots are allowed to request pages. Web spiders should ideally follow rules in a robots.txt file. For example, like the robots.txt file for YouTube in Figure 2, we can specify "The User-agent" to determine which User-agent the rule applies to and specify "Disallow" to determine which files or folders shouldn't be crawled. In addition, Robots META Tag is individual document tag which can be used to exclude indexing or following links.

```
# robots.txt file for YouTube
# Created in the distant future (the year 2000) after
# the robotic uprising of the mid 90's which wiped out all humans.

User-agent: Mediapartners-Google*
Disallow:

User-agent: *
Disallow: /channel/*/community
Disallow: /comment
Disallow: /get_video
Disallow: /get_video_info
Disallow: /login
Disallow: /results
Disallow: /signup
Disallow: /t/terms
Disallow: /timedtext_video
Disallow: /user/*/community
Disallow: /verify_age
Disallow: /watch_ajax
Disallow: /watch_fragments_ajax
Disallow: /watch_popup
Disallow: /watch_queue_ajax
```

Figure 2 robots.txt File for YouTube[2]

3.3 HONEYPOTS

Honeypots usually are links that normal user can't see but a spider can access. Some websites install honeypots to detect web spiders. For example, some honeypot links to detect spiders will be have the CSS style display:none or will be color disguised to

blend in with the page's background color. To prevent capturing legitimate bots, access to the honeypot will be prevented in robots.txt file. Thus, legitimate crawlers such as the ones from Google would not open the honeypot but some scraping software which disregards such prohibitions will just follow the instructions given to it. Thus it is vital for crawlers to follow rules.

3.4 Cookies and JavaScript

Since most crawlers there are not real browsers, many of them have difficulty in executing JavaScript or storing cookies. This means that a way of protection against some of the bots could be to set a cookie or execute a JavaScript and do not render the contents unless the cookie is successfully stored or the script is executed.

However, this approach poses some usability issues since users could disable cookies and JavaScript themselves which will hinder them from accessing your page. We also need to think about ways to allow legitimate some crawlers (such as the ones from Google, Yahoo or Bing) if the content on the pages protected by this technique is important to be crawled by them.

3.5 CAPTCHA

A CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) is a type of challenge-response test used in computing to determine whether or not the user is human.[3] Frequently, a CAPTCHA features an image file of slightly distorted alphanumeric characters. A human can usually read the characters in the image easily while the bot program can hardly recognize it.

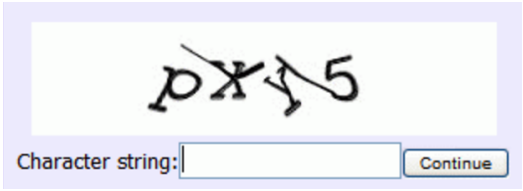

Figure 3 CAPTCHA for Yahoo Mail

We can add a CAPTCHA that users would have to solve before accessing any content. However, it will hinder the user experience quite a lot if the CAPTCHA is shown with each request. Therefore, it might be better to show the CAPTCHA only when you detect a large number of requests over a small period of time.

There is no ethical way to evade a CAPTCHA, because the entire purpose of a CAPTCHA is to verify that you are not a bot doing things automatically. However, there are still some tools to solve CAPTCHA with Optical Character Recognition (OCR), such as DeCaptcher and Captcha Sniper.

4. Conclusion

Building a web crawler for different purposes is not a difficult task, but choosing the right strategy will lead to the implementation of highly efficient web crawler application. The comparison of different crawling strategies given in this paper can help us choose a proper strategy for a specific task. In practice, there are many techniques to detect web spiders, so web crawler is supposed to perform politely and respect rules, otherwise it will have negative influence on the website and get blacklisted.

Reference

[1] Udapure T V. Study of web crawler and its different types[J]. IOSR Journals (IOSR Journal of Computer Engineering), 2014, 1(16): 1-5.

[2] https://www.youtube.com/robots.txt

[3] "The reCAPTCHA Project - Carnegie Mellon University CyLab". *www.cylab.cmu.edu. Retrieved 2017-01-13.*

[4] How to stop Search Engines from crawling your Website http://www.inmotionhosting.com/support/website/restricting-bots/how-to-stop-search-engines-from-crawling-your-website