Web Crawler in Python

-By AbssZy

1. ABSTRACT

A parallel crawler is a crawler that runs multiple processes in parallel. The goal is to maximize the download rate while minimizing the overhead from parallelization and to avoid repeated downloads of the same page. By default, Python programs are single threaded. This can make scraping an entire site using a Python crawler extremely slow. Through this project, we will demonstrate web crawling of multiple child node in one go from the parent node using the concept of multithreading in python program.

2. INTRODUCTION

Python is a great language for writing web scrapers and web crawlers. Libraries such as BeauitfulSoup, requests and lxml make grabbing and parsing a web page very simple. By default, Python programs are single threaded. This can make scraping an entire site using a Python crawler extremely slow. We must wait for each page to load before moving onto the next one. Python supports threads which though not appropriate for all tasks, can help us increase the performance of our web crawler. A parallel crawler is a crawler that runs multiple processes in parallel. The goal is to maximize the download rate while minimizing the overhead from parallelization and to avoid repeated downloads of the same page. To avoid downloading the same page more than once, the crawling system requires a policy for assigning the new URLs discovered during the crawling process, as the same URL can be found by two different crawling processes. The Web comprises of voluminous rich learning content. The volume of ever growing learning resources however leads to the problem of information overload. A large number of irrelevant search results generated from search engines based on keyword matching techniques further augment the problem. Keeping in view the volume of content a significant crawler of semantic knowledge can be built on demonstrated multi-threaded unfocused crawler implemented in this project.

3. REQUIREMENTS

Anaconda Navigator, Spyder, BeautifulSoup, Requests, URLlib

4. KEYWORDS

Multithreading, Parallelization, Crawler, Unfocused, Concurrent Crawling

5. LITERATURE SURVEY

Web crawlers follow links in webpages and automatically download the page[1]. Crawling is the most basic as well as the most powerful web search procedure. A design and framework of a multithreaded web crawler has been proposed. The end result of the process is a collection of web pages. The experiment demonstrates how this process has better performance. It is possible to increase the performance of a search by understanding the user's need and the relevance
of the document.

The computer network is vast and act as a single huge network for data and message transportation across vast distances. The World wide web has millions of pages linked and is ever growing. Hence, they continuously keep track of the web an find new webpages.
This system called the semantic web works on shared meaningful knowledge representation. It proposes to make an AI application that will make the content of the web meaningful to the system. This will make the crawler must more efficient by crawling only those pages which has relevance to the users requirements[2].

Initially a seed URL is given input by the user. Crawler then fetches the webpage from the interne. Contents of the page are parsed and the URL are scraped and stored in Database. If the links are to be traversed, the links are clicked and passed onto the web crawlers. Crawler then searches the keyword if present in the page fetched using the URL. If present, it calculates the number of times it is present in the page. A brief evaluation study of the crawler in an uncontrolled and practical environment is the main objective of this research paper.[3]

The World Wide Web (or simply the web) is a vast, prosperous, superior, readily available and suitable source of information and its users are growing very fast nowadays. To extract information from the web, Search engines are used that access web pages as per the requirement of the users. Most of the data in the web is unmanaged so it is not possible to use the entire web at once in a single attempt, so search engines use the web crawler. Web crawler is an important part of search engine. This is a program that navigates the web and downloads reference to web pages.[4]

Search engines run multiple instances of crawlers on wide spread serversGet various information from them. Web crawler crawls from page topage in world wide. Get the webpage, load the page content and index itto find the engines database.This paper presents a systematic study of the web Crawler. The study ofweb crawler is very important because properly designed web crawler always gives good yield.Web crawler are an efficient component of the search engine. It is the core member responsible for searching the web and indexing the traversed webpages. It starts with a few seed pages, travels the web using the links in the webpages and inserts new link in the system database[5]

In the paper ―Finding Near-Duplicate Web Pages: A Large-Scale Evaluation of Algorithms‖ the author stated that Broder et al.'s [7] shingling algorithm and Charikar's [6] random projection based approach are considered \state-of-the-art" algorithms for finding near-duplicate web pages. Both algorithms were either developed at or used by popular web search engines. They compare the two algorithms on a very large scale, namely on a set of 1.6B distinct web pages.

6. METHEDOLOGY

An adopted methodology for web crawling is when a query is passed to search engine by user, web crawler starts then from a set of seed pages. Page downloader fetches the webpage for a particular URL. The downloaded page is passed onto the extractor which extracts contents and parses them to get the links. Frontier starts indexing all the received links. The goal is to maximize the download rate while minimizing the overhead from parallelization and to avoid repeated downloads of the same page. To avoid downloading the same page more than once, the crawling system requires a policy for assigning the new URLs discovered during the crawling process, as the same URL can be found by two different crawling processes. It works like an ackerman function where the key methods of a crawler are packed within a function that is called for requests.

7. CONCLUSION

When testing this script on several sites with performant servers, we were able to crawl several thousand URLs a minute with only 20 threads. Ideally, we would use a lower number of threads to avoid potentially overloading the site you are scraping. Although when crawling a heavy website like that of government, there are high risks of crawler being trapped on the website due to lack of semantic crawling.

However, it overcomes the wait for each page to load before moving onto the next one and help us increase the performance of our web crawler. A parallel crawler is a crawler that runs multiple processes in parallel. The goal is to maximize the download rate while minimizing the overhead from parallelization and to avoid repeated downloads of the same page to crawl multiple links using the concept of multithreading.

8. REFERENCES

[1] Chau, D.H., Pandit, S., Wang, S., Faloutsos, C. and Faloutsos, C., 2007, May. Parallel crawling for online social networks. In *Proceedings of the 16th international conference on World Wide Web* (pp. 1283-1284). ACM.

[2] Castillo, C., 2005, June. Effective web crawling. In *Acm sigir forum* (Vol. 39, No. 1, pp. 55-56). Acm. [3]Thelwall, M., 2001. A web crawler design for data mining. *Journal of Information Science*, *27*(5), pp.319-325.

[4] Shkapenyuk, V. and Suel, T., 2002, February. Design and implementation of a high-performance distributed web crawler. In *Proceedings 18th International Conference on Data Engineering* (pp. 357-368). IEEE.

[5] Boldi, P., Codenotti, B., Santini, M. and Vigna, S., 2004. Ubicrawler: A scalable fully distributed web crawler. *Software: Practice and Experience*, *34*(8), pp.711-726.

[6] M. S. Charikar. Similarity Estimation Techniques from Rounding Algorithms. In 34th Annual ACM Symposium on Theory of Computing (May 2002).

[7] A. Broder, S. Glassman, M. Manasse, and G. Zweig. Syntactic Clustering of the Web. In 6th International World Wide Web Conference (Apr. 1997), 393-404.