International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 02 | Feb 2021 www.irjet.net p-ISSN: 2395-0072

Review of Web Crawler

Duckki Lee

Department of Smart Software, Yonam Institute of Technology, Jinju, Korea

---

Abstract - *Development of smartphones and social networking services (SNS) has spurred the explosive increase in the volume of data, which continues to grow exponentially in volume with time. This recent trend has ushered in the era of big data. Proficient handling and analysis of big data will produce information of great use and value. However, being able to collect large volume of data is necessary before the analysis of big data. Since large data with reliable quality are mainly available on internet pages, it is important to search and collect relevant data from the internet pages. A web crawler refers to a technology that automatically collects internet pages of a specific site from this vast World Wide Web. It is important to select the appropriate web crawler taking into account the context when a large amount of data needs to be collected and the characteristics of the data to be collected. To facilitate selection of the appropriate web crawler, this study reviews the to this end, this paper examines the structure of web crawlers, their characteristics, and types of open source web crawlers.*

Key Words: Web Crawler, Web Crawling, Distributed Web Crawler, Scarpy, Selenium, Heitrix, Apach Nutch

1.INTRODUCTION

Development of smartphones and SNS has led to the explosive increase in the volume of data, which continues to grow exponentially with time. This recent trend has ushered in the era of big data[1][2]. Skilled analysis of big data will produce information of great utility and value. However, being able to collect large volume of data is necessary before the analysis of big data. Since large data with reliable quality are mainly available on internet pages, it is important to search and collect relevant data from the internet pages. A web crawler[3-12] refers to a technology that automatically collects internet pages of a specific site from the vast world wide web. There has been increasing emphasis on the importance of web crawlers as the utilization of big data has gradually become a norm across the globe and with the annual trend of exponential increase of web data. Therefore, it is important to use the appropriate web crawler and web crawling algorithm and collect the data fitting to the intended purposes taking into account the context when a large volume of data needs to be collected and the characteristics of the data to be collected. To this end, in this paper, we will examine the structure of Web crawlers, their characteristics, and types of open source web crawlers.

2. RELATED WORK

A web crawler indicates a technology that automatically collects web pages of a specific site from the vast World Wide Web. It is also called a web spider or a web robot. Web crawlers can be classified into general web crawler and distributed web crawler.

GENERAL WEB CRAWLER

The general web crawler[3-12] automatically collects data starting from a list of URLs to be visited, called seeds, in a single system. Since the general Web crawler manages the seeds in a single system, it is not affected by the problem of overlapped crawl, but it has the disadvantage that web crawling becomes very time consuming since web crawling is performed over a huge volume of web pages by a single system.

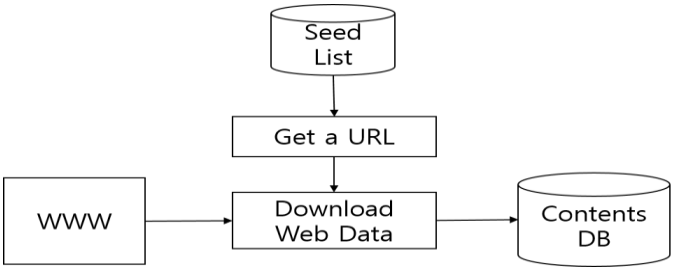

Fig -1: Architecture of Web Crawler

DISTRIBUTED WEB CRAWLER

The distributed web crawler[13-17] performs web crawling in a server-client environment, in which the server distributes the initial seed to the client and receives the web pages collected by the client. The client performs web crawling based on the seeds received from the server, and repeatedly performs seed extraction and crawling on the next web page. In the distributed web crawler, the overlapped crawl problem arises because the information on the web pages collected by each client is not shared between clients.

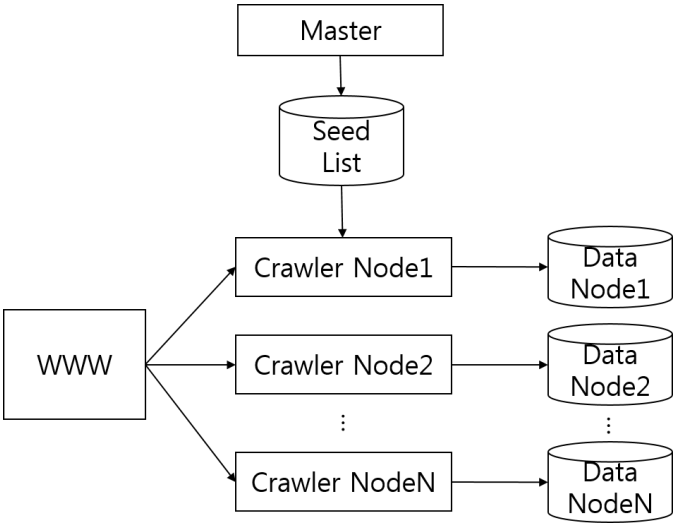

Fig -2: Architecture of Distributed Web Crawler

WEB CRAWLER CHARACTERISTICS

In this section, the characteristics of web crawlers[4][5][9][10] are outlined which would be helpful in determining a good web crawler.

ROBUSTNESS: Since web crawling involves visiting many different web pages, web crawlers must be robust to withstand various problems such as incorrect HTML, incorrect web server actions or configuration, or other types of problems.

POLITENESS: Web crawlers must comply with the crawling speed policy for crawlers that each web server requires, and excessively frequent crawling beyond the scope of the policy should not be performed.

DISTRIBUTED: Web crawlers must be able to perform distributed crawling on multiple machines.

SCALABLE: Web crawlers must have a structure that can speed up crawling by providing additional machines or bandwidth.

PERFORMANCE AND EFFICIENCY: Web crawler must be able to utilize various system resources (e.g. processors, storage devices, network bandwidth) with efficiency.

FRESHNESS: Web crawlers must be able to operate in continuous mode. That is, it must be able to provide the latest data from previously collected web pages.

EXTENSIBLE: Web crawlers should be designed for smooth implementation of new data format or protocol.

3. Web crawler

3.1 Scrapy

Scrapy[18-21][31] is a Python framework for web scraping. Scrapy enables you to access the web pages you choose and process and store the data in the format of your choice, and it supports formats such as JSON, XML, and CSV. Scrapy is easy to use and well documented on its information. Skilled Python programmers would be able to set up and run Scrapy in minutes. However, it is difficult to export large amount of data without the support of distributed environment, dynamic extension, and continuous crawling.

3.2 Heritrix

Heritrix[22-23][32] is an open source Web crawler project created and managed in internet archive. Heritrix is characterized by well-organized documentation and easy setup and offers a mature and stable platform. It has excellent performance and good support for distributed crawl. However, since continuous crawling and dynamic extension are not

supported, Heritrix has a disadvantage of having to manually set the server structure.

3.3 Apach Nutch

Apach Nutch[24-26][33] is an open source software project for web crawling and was originally a subproject of Apache Lucene. It supports Hadoop-based distributed crawling, exploits the Hadoop ecosystem and implements MapReduce for processing. Since Apach Nutch uses the Hadoop ecosystem, it has advantages of fault-tolerance and scalability, which are also the strengths of Hadoop. However, it is slow in terms of disk access time between jobs, which is the disadvantage of Hadoop. In addition, it has other drawbacks such as not-established documentation and difficulty in setup.

3.4 Selenium

Selenium[27-30][34] is a framework used for automation testing of web applications. In the framework, there is a library called WebDriver for automating the control of web browsers, which allows you to run a web browser and crawl web pages. Since Selenium performs web crawling by a web browser, it has the advantages of being able to collect all the information on the web page that the user views, and its method of use is intuitive. However, since it performs web crawling by actually running a web browser, it has disadvantages of considerably slow crawling and taking up much memory.

4. CONCLUSIONS

In this study, we have compared the general Web crawler and the distributed web crawler and reviewed the characteristics of the Web crawler. The characteristics, advantages and disadvantages of open source web crawlers for web crawling, which are Scrapy, Heritrix, Apach Nutch, and Selenium were examined. There is no perfect, all-rounder web crawler to single out as yet, and thus it is important to select the appropriate Web crawler and collect data fitting to the purpose of use in consideration of the situations where web data needs to be collected and the characteristics of the data to be collected.

REFERENCES

[1] S. Sagiroglu and D. Sinanc, "Big data: A review," 2013 International Conference on Collaboration Technologies and Systems (CTS), San Diego, CA, USA, 2013, pp. 42-47, doi: 10.1109/CTS.2013.6567202.

[2] M. Volk, S. Bosse, and K. Turowski, “Providing Clarity on Big Data Technologies: A Structured Literature Review,” Proc. IEEE Conference on Business Informatics (CBI 17), IEEE Press, 2017, vol. 01, pp. 388– 397.

[3] C. Olston and M. Najork, "Web Crawling," Foundations and Trends in Information Retrieval. Vol. 4, No. 3, 2010.

[4] A. Heydon and M. Najork. Mercator: A scalable, extensible web crawler. World Wide Web, 2(4):219–229, 1999.

[5] M. Najork and A. Heydon. High-performance web crawling. SRC Research Report 173, Compaq Systems Research Center, Sep. 2001.

[6] C. Castillo, ‘‘Effective web crawling,’’ ACM SIGIR Forum, vol. 39, no. 1, pp. 55–56, 2005.

[7] C. Saini and V. Arora, "Information retrieval in web crawling: A survey," *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Jaipur, India, 2016, pp. 2635-2643, doi: 10.1109/ICACCI.2016.7732456.

[8] G. S. Manku, A. Jain, and A. Das Sarma, "Detecting near-duplicates for web crawling," in Proc. 16th Int. Conf. World Wide Web, 2007, pp. 141–150.

[9] M. Najork, “Web Crawler Architecture,” Book Chapter in Encyclopedia of Database Systems, Springer-verlage, pp. 3462- 3465, Sept 2009

[10] C. D. Manning, P. Raghavan, and H. Schütze, “Web crawling and indexes,” in *Introduction to Information Retrieval*, Cambridge: Cambridge University Press, 2008, pp. 405–420.

[11] S. Pandey and C. Olston, “User-Centric Web Crawling,” Proc. Int’l World Wide Web Conf. (WWW ’05), 2005

[12] Pant, G., Srinivasan, P., Menczer, F., "Crawling the Web". Web Dynamics: Adapting to Change in Content, Size, Toplogy and Use edited by M. Levene and A. Poulovassilis, Springer-verlog, pp. 153-175. Nov, 2004.

[13] P. Boldi, B. Codenotti, M. Santini, and S. Vigna. Ubicrawler: a scalable fully distributed web crawler. Software, Practice and Experience, 34(8):711–726, 2004.

[14] Bal S K, Geetha G. Smart distributed web crawler// International Conference on Information Communication and Embedded Systems. IEEE, 2016:1-5.

[15] M. S. Kumar and P. Neelima, “Design And Implementation of Scalable, Fully Distributed Web Crawler For a Web Search Engine,” International Journal of Computer Applications (0975-8887), 2011

[16] S. Zhong and Z. Deng, "A web crawler system design based on distributed technology," Journal of Networks, vol. 6, no. 12, pp. 1682– 1689, 2011.

[17] Shkapenyuk and T. Suel, "Design and implementation of a high-performance distributed web crawler," in Data Engineering, 2002. Proceedings. 18th International Conference on, 2002, pp. 357–368.

[18] Daniel Myers and James W McGuffee, "Choosing scrapy," Journal of Computing Sciences in Colleges, vol. 31, no. 1, pp. 83–89, 2015.

[19] Wang J, Guo Y. Scrapy-Based Crawling and User-Behavior Characteristics Analysis on Taobao// International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery. IEEE Computer Society, 2012:44-52.

[20] Daiyi L I, Xie L, Qian S, et al. Design and Implementation of Distributed Crawler System Based on Scrapy. Journal of Hubei University for Nationalities, 2017.

[21] Fan Y. Design and Implementation of Distributed Crawler System Based on Scrapy// 2018:042086.

[22] K. Sigursson Incremental crawling with Heritrix. Proceedings of the 5th International Web Archiving Workshop, 2005.

[23] G. Mohr, M. Stack, I. Ranitovic, D. Avery, and M. Kimpton, "An introduction to heritrix an open source archival quality web crawler," in In IWAW04, 4th International Web Archiving Workshop. Citeseer, 2004.

[24] Khare et al., "Nutch: A flexible and scalable open-source web search engine," Tech. Rep., 2004.

[25] Z. Laliwala, and A. Shaikh. "Web crawling and data mining with Apache Nutch." Packt Publishing, 2013.

[26] Sebastian Nagel. "Web crawling with Apache Nutch." ApacheCon EU (2014).

[27] J. Peng, Y. Ma, F. Zhou, S. Wang, Z. Zheng and J. Li, "Web Crawler of Power Grid Based on Selenium," 2019 16th International Computer Conference on Wavelet Active Media Technology and Information Processing, Chengdu, China, 2019, pp. 114-118, doi: 10.1109/ICCWAMTIP47768.2019.9067730.

[28] A. Holmes and M. Kellogg, "Automating functional tests using Selenium," AGILE 2006 (AGILE'06), Minneapolis, MN, USA, 2006, pp. 6 pp.-275, doi: 10.1109/AGILE.2006.19.

[29] A. Bruns, A. Kornstadt and D. Wichmann, "Web Application Tests with Selenium," in IEEE Software, vol. 26, no. 5, pp. 88-91, Sept.-Oct. 2009, doi: 10.1109/MS.2009.144.

[30] P. Ramya, V. Sindhura and P. V. Sagar, "Testing using selenium web driver," 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT), Coimbatore, 2017, pp. 1-7, doi: 10.1109/ICECCT.2017.8117878.

[31] http://scrapy.org

[32] https://webarchive.jira.com/wiki/spaces/Heritrix

[33] http://nutch.apache.org/

[34] https://www.selenium.dev/

BIOGRAPHIES

Duckki Lee is currently an Assistant Professor in the Department of Smart Software, Yonam Institute of Technology. His research interests include big data system, web application, and web crawler.