Application of Distributed Web Crawlers in Information Management System

Bo Wen
School of Computer Science and Technology, Huaibei Normal University, Huaibei, 235000, China
E-mail: bowen1983@yeah.net

Technical paper

Keywords: web crawlers, Hadoop, information management system

Received: February 7, 2018

In the Internet era, cloud data and big data constantly develop, and Internet has become the main platform for enterprises and individuals to release information. As a result, a large amount of data generates, and people spend more energy on finding information that they want. The desire for accurately acquiring information needed becomes increasingly stronger. This study designed a distributed web crawlers system based on Hadoop and used it to do large-scale information management. The simulation experiment verified that the system could operate stably in information management system, which offers a reference for the application of distributed web crawlers in information management systems.

Povzetek: Razvit je distribuirani spletni preiskovalnik na osnovi Hadoopa za upravljanje informacij.

1 Introduction

Internet rapidly develops in the 21st century, accompanied by data volume increasing in exponential form on Internet. With the diversification of information, the management of information has become more and more difficult. How to timely and accurately search information through search engine and manage the information becomes crucial. Requirements on relevant technologies are also being improved constantly. With the development of computer, information management system has emerged. More efficient and simple information management systems are being developed. Qin [1] designed a SG-UAP development tool based on Eclipse development environment which was applicable to Windows operation system; a database platform was developed based on Oracle to provide Tomcat network information management service; the system managed network information through service-oriented architecture. Gupta et al. [2] analyzed management information service and proposed to manage network information with management information service and found that management information system could optimize network information and accurately collect and manage data. Zhao et al. [3] established a topic-focused crawler based scientific research information system to improve the information management level. Web crawlers can capture webpage information from the network and extracted and stored the key information to solve the urgent problem of information acquisition. But information collection based on web crawlers is facing with difficulties such as information repetition and existence of dynamic pages. Therefore distributed technologies are needed to solve the problems and enhance crawling efficiency. In the study of Su et al. [4], single-thread and multi-thread web crawlers were implanted into a distributed system to capture and store data with diversified and personalized operations, which enhanced the capturing speed. In the study of Zhang et al. [5], Hadoop based distributed web crawler system was optimized. The parameters were optimized through analysis on factors influencing crawling efficiency. Distributed web crawlers have great advantages in collecting and storing information; hence it can help establish a practical and high-efficient information management system. In this study, web crawlers were analyzed, and then a Hadoop based distributed web crawlers system was designed to manage network information. The simulation experiment suggested that the system could effectively collect and store network information and enhance the performance of single-node web crawlers, which provides a reference for the application of distributed network crawlers in information management system.

2 System related technologies

2.1 Web crawlers

Distributed web crawler is a program which crawls Web resources on the Internet according to some rules and provides the obtained network information to search engine. Therefore it is an indispensable part of search engine [6]. To achieve a high crawling ability, a web crawler should have the five characteristics [7].

(1) High performance

A large amount of information involves mass Uniform Resource Locator (URL). Distributed web

crawlers should timely and effectively capture useful information in webpage. The more the information in unit time is, the better the performance of web crawlers is.

(2) Expandability
Expandability should be improved to achieve a high performance of web crawlers. Expandability means that the whole crawler system will not be affect when the current web crawlers are being updated or doing other operations. Better expandability is needed in efficient crawling of information in different sites as the programming language and code editor are different in different websites.

(3) Robustness. Facing with a large number of servers, web crawlers may encounter emergencies such as crawler trap in the process of work. Reasonable processing of these conditions is a character of an excellent web crawler. Only when web crawlers have favorable robustness can they get back to work after interruption. Moreover the previously crawled content should be restored after setup.

(4) Friendliness: Web crawlers should protect relevant information of websites as per robots protocols. The crawling scope of web crawlers should be defined. Moreover additional burdens to websites should be avoided when web crawlers capture information.

(5) Updatability. Web crawlers should be able to perceive the alternation of websites and timely acquire new website content to replace the old one.

Information management system needs to collect and store diversified data on the Internet. With the explosive growth of data, the traditional stand-alone web crawlers have gradually been not as good as before. Hence stronger and more comprehensive information management systems are needed.

2.2 Hadoop

Hadoop, a basic framework of distributed system developed by Apache Software Foundation, is composed of many ordinary, low-cost single computers. It can rapidly and flexibly process mass data. It has the following advantages.

(1) High reliability. Its ability in processing data is highly reliable.

(2) Strong fault tolerance. Hadoop can automatically replicate many copes and allocate failed tasks.

(3) High scalability. Hadoop can process and allocate data between hundreds of servers and easily expand to thousands of nodes.

(4) High efficiency. Hadoop can efficiently transfer data between different nodes.

(5) Low cost. Compared to other commercial data warehouse, Hadoop is open-source.

Hadoop has two core parts. One is distributed file system, i.e. Hadoop Distributed File System (HDFS). HDFS is capable of storing large files, for example, files in a size of more than 100 TB. HDFS is also featured by strong fault tolerance. It can operate on low-cost hardware. The other core is MapReduce computational model which can concurrently calculate mass data and have favorable extensibility and fault tolerance. It has a huge advantage in data processing.

2.3 Application values of distributed web crawlers in information management system

In view of the advantages of distributed system and the properties of web crawlers, distributed web crawler is feasible. Distributed web crawler is composed of web crawler and distributed system, which is capable of fulfill different tasks by making the best use of information on the Internet. It effectively makes up the defects of the stand-alone web crawler. It can capture more websites and collect and store more data. Therefore Hadoop based distributed web crawler has high application values in information management system.

3 Design of information management system

3.1 Design of distributed web crawler system architecture

3.1.1 Design of physical architecture

To satisfy the aforementioned characteristics, cost of PC server should be saved, and moreover Hadoop based distributed architecture should be extensible [8]. The system should allocate the crawled page data on different nodes using its ability of distributed storage capacity. Moreover a strong fault tolerance was needed to set the number of data copies and reallocate the failed tasks on other nodes. The distributed architecture could enhance the overall performance of crawlers to the large extent.

The physical architecture of web crawlers in this study included Hadoop cluster and Storm cluster [9]. To reduce the pressure on Hadoop cluster during operation, separate deployment was adopted. Crawler tasks were divided into multiple tasks and operated on multiple Slave nodes based on the distributed storage and calculation abilities of distributed architecture. The collected data were stored in clusters. Then the data generated when crawlers crawled and analyzed webpage were written into Kafka, and Storm was used to calculate index results in real time. The physical architecture is shown in Figure 1.

3.1.2 Design of logic structure

The logic structure of distributed web crawlers is shown in Figure 2. It included batch processing part and real-time calculation part. Batch processing was mainly realized based on Hadoop platform, and it was responsible for achieving crawling tasks and storing data in Hbase. Real-time calculation was realized based on Storm platform, and it was responsible for calculating relevant data generated in system operation and storing the results in iRedis.

3.2 Modules of distributed web crawlers

The system module of the distributed web crawler was

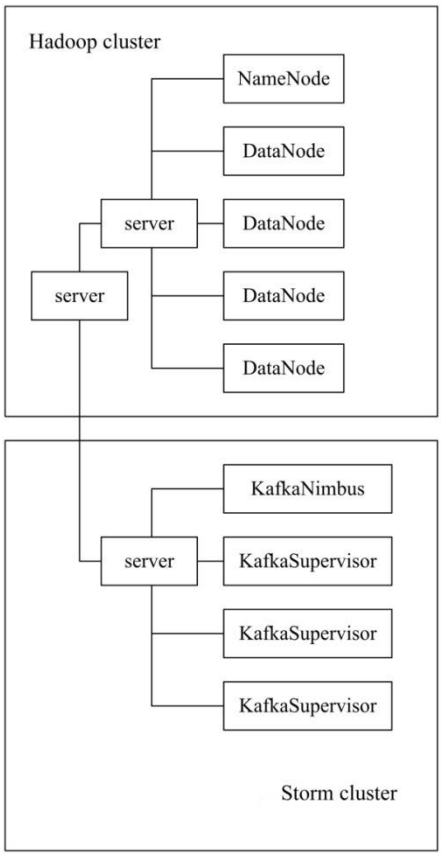

Figure 1: Design of the physical architecture of distributed web crawlers.

composed of the following parts.

(1) URL splitting and injection module: firstly read the URL path of user, then obtain URL list, and split it into several parts and allocate to TaskTracker.
(2) Webpage access module: acquire webpage according to URL links and download and save it locally.
(3) Webpage analysis module: analyze the captured webpage in aspects of structure and content.
(4) Link filtering module: filter the acquired URL and eliminate ineffective and repeated links.
(5) Data storage module: Save data in the database of HDFS.

3.3 Design of key technology

3.3.1 URL standardization

URL is a kind of character which can show information resources on www, and information resource has one and only has one URL [10]. URL standardization meant standardizing URL and transforming a URL to a qualified equivalent URL. Its transformation was realized by replacing /xx/../ with /, /../ with /, /./ with / and xx//yy with /.

3.3.2 Allocation of crawler tasks

Before crawling based on the distributed architecture, tasks were allocated to the distributed clusters [11]. When some node failed, tasks should be reallocated. For Hadoop cluster with n nodes, a URL was selected from URL set, Topn URLs were divided into N sets, and the sets were allocated to different nodes of Hadoop set to do crawling tasks. If some node failed, Master would allocate the failed task to other nodes without affecting the crawling speed of the current nodes. The network pressure of websites should be considered in the process of crawling.

3.3.3 Balance politeness

Crawling of the same website should follow the principle of balance politeness [12]. URLs were ranked according to score rules; then URL was taken out one by one from the URL set and allocated to N subsets; the number of URLs in one set and the number of URLs from the same Host in one set should be limited. In this way, the pressure of webpage could be reduced when web crawlers were crawling information.

3.3.4 Webpage revisit

Network usually has favorable dynamic property. When web crawlers fulfilled a crawling task, then the webpage might change. Therefore web crawlers should update website content at a certain time interval and the content which needed to be crawled.

3.3.5 Data deduplication

There are many same data on the network. Therefore network data should be processed by deduplication.

(1) Webpage content was separated into words, i.e.

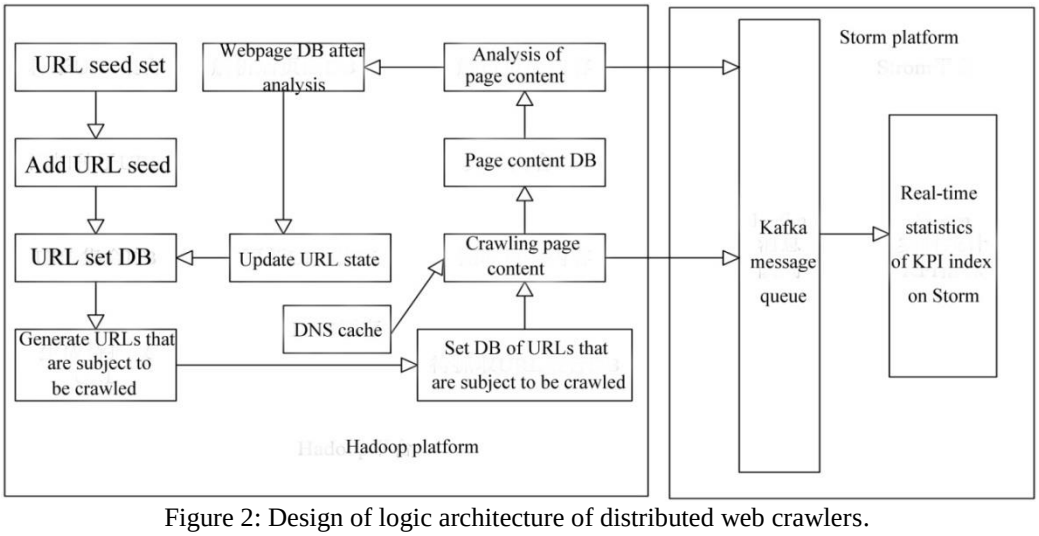

Figure 2: Design of logic architecture of distributed web crawlers.

characteristic vectors. The occurrence frequency of every word in documents was taken as weight.

(2) The Hash value of every characteristic vector was calculated [13], and moreover those vectors were processed by weighed accumulation.

(3) The result larger than 0 was denoted as 1 and otherwise as 0, and the final results were Simhash signature values [14].

(4) The similarity of data was determined according to different Simhash signature values.

4 Concrete implementation of distributed crawler

URL initial module was combined with parallel circulation model to analyze the procedures of URL insertion, URL list generation, web crawling and data update in the data crawling experiment of distributed crawler. A module circulation formed from link update in link library, crawl list generation, URL crawling execution, key information analysis to link update in link library. The module composition and flow circulation can benefit the concrete implementation of distributed crawler. The concrete implementation flow is shown in Figure 3.

5 System test and results analysis

Before testing of the network management, the test environment should be adjusted. VMware Workstation was installed in window host. Then Hadoop cluster was established in the virtual machine. In the development of the system, Java codes written by Eclipse IDE were installed in the host. Hadoop Eclipse plug-in units was installed and connected to Hadoop clusters. Data were processed using Hadoop Distributed File System (HDFS) and MapReduce calculation model.

5.1 Functional test

5.1.1 Test content and scheme

Functional test included the following content.

(1) Webpage crawling test
In the initial URL set, 0, 1 and 4 URL link seeds were added. Then three conditions, i.e. effective crawling, partially effective crawling and ineffective crawling, were considered. After crawling, whether the downloaded target data satisfied standards or not were checked.

(2) Filter test on URL link
The URL link log sheet which was subject to be crawled was checked to determine whether link standardization and deduplication operations should be performed or not.

(3) Webpage data extraction test
Whether the analysis module was corrected and could effectively extract data on webpage and store the data in relevant documents or not was checked.

(4) Test on webpage category classification
The system classified webpage into different categories and checked whether the classification was corrected or not.

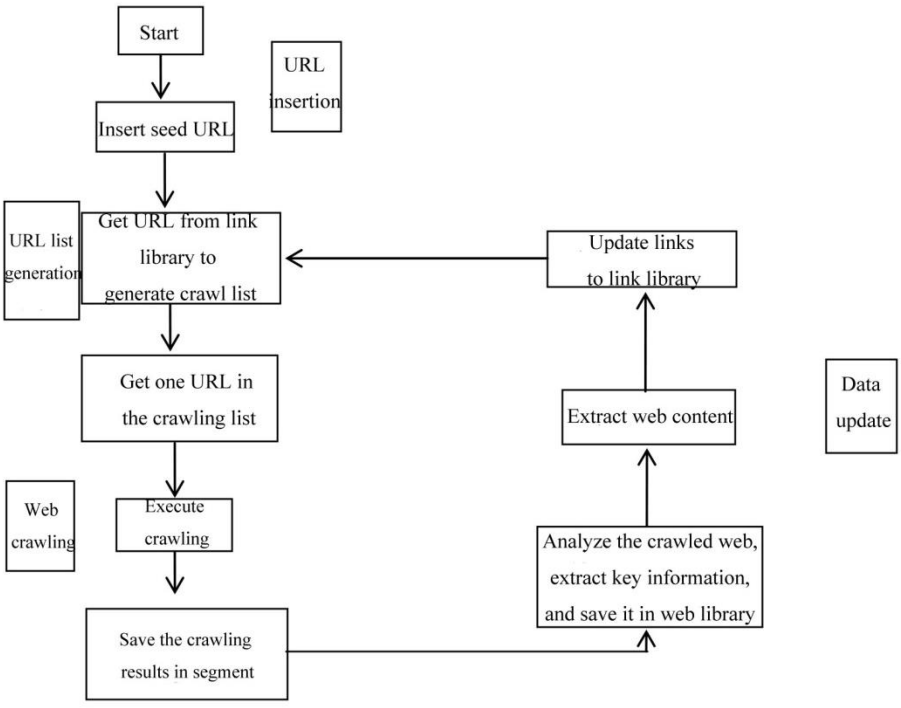

Figure 3: The implementation flow of the distributed crawler.

5.1.2 Test results

The system could do webpage crawling according to the prescribed initial URL set and added the crawled URLs into URLs which were subject to be crawled. Standardization and deduplication were performed before addition. The extracted data were stored in relevant documents. Moreover it could rapidly classify webpage.

5.2 Performance test

5.2.1 Test content and scheme

(1) Test on collection scale
After a period of webpage crawling, the size of the collected webpage data was calculated to measure the collection scale.

(2) Test on operation speed
During crawling, the size of the collected web data, i.e. x, was calculated after n hours of movement. The computational formula for crawling speed v was v = x/n.

5.2.2 Test results

Table 1 shows the data collection speed of the clusters based on four nodes. The operation of the system included webpage downloading, web analysis, extraction of record information on the network and classification of web text. This study could basically satisfy the requirements according to the data in Table 1.

5.3 Test on expandability

5.3.1 Test content and scheme

Test on expandability: the number of nodes on Hadoop platform was changed. Then test was performed when the number of coordinated nodes was 1, 2 and 3 to determine whether the operation was normal and what were the effects on the performance of the system.

5.3.2 Test results

Figure 4 demonstrated the data collection and analysis of the system when the time and number of nodes were different. It could be noted that the operation speed was the highest when there was only one node; the operation speed had remarkable improvement with the increase of nodes, but the speed of each node had no significant changes. Through test, it was concluded that the expandability could satisfy the predetermined requirements.

Internet plays an increasingly important role in the production and life of people and has been the main source of information. Distributed web crawlers can grab key data among mass data, which is greatly helpful to information acquisition. Bal et al. [15] put forward intelligent distributed crawler crawling network based on client-server architecture. In the architecture, load is managed by server. Every time when crawlers were loaded, URLs were dynamically allocated to allocate load to others, which enhanced the ability of information crawling. Kumar et al. [16] developed distributed semantic web crawlers and successfully crawled and

Table 1: The operation results of the information management system.

Number	1	2	3	4	5
Segment name	Segment20171002093417	Segment2017100213672	Segment2017100360349	Segment2017100413725	Segment2017100547436
Size (MB)	39.21	82.61	180.44	305.14	400.62
Operation time (h)	0.6	1.1	2.4	4.5	5.7

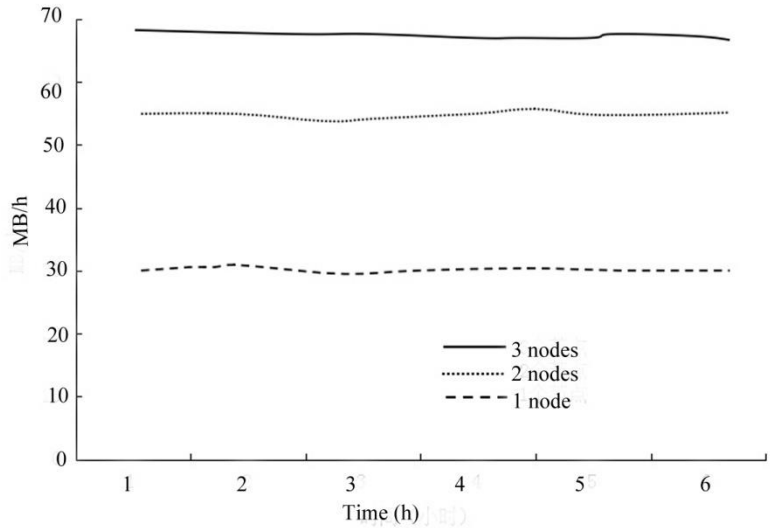

Figure 4: The test results of system expandability.

utilized HTML compiled by owl/RDf and semantic web. In information management system, distributed web crawlers can give full play to its advantages because it can effectively crawl information needed among mass data and efficiently collect and manage them. The application of distributed web crawlers can achieve efficient and safe management of information and has high practicability.

6 Conclusion

In conclusion, distributed network crawlers based information management system could precisely satisfy the requirements of web crawling, with a high performance and expandability. Moreover it can effectively reduce repeated visit and download of resources to improve efficiency of information searching. It can also reduce the time and money spent on resource acquisition because of the low cost. Therefore it can be applied for extracting network information. This work provides a reference for the application of distributed network crawlers based information management system in data extraction.

7 References

[1] Qin Y., Xuan H., Zhang B. (2016). Intelligent Management System of Power Network Information Collection Under Big Data Storage. *13th Global Congress on Manufacturing and Management (GCMM 2016), MATEC Web of Conferences*, Zhengzhou.

[2] Gupta C. L. P., Sharma S., Tripathi S. (2015). Importance of Management Information System in Electronic-Information Era. *East Carolina University*, 1(2).

[3] Zhao Q. A. (2016). Research and Implementation of Scientific Research Information Management System Based on the Topic Web Crawler. *Anhui: Anhui University*, pp. 1-46.

[4] Su L., Wang F. (2017). Web crawler model of fetching data speedily based on Hadoop distributed system. *IEEE International Conference on Software Engineering and Service Science*, Beijing, pp. 927-931.

[5] Zhang X., Xian M. (2015). Optimization of Distributed Crawler under Hadoop. *International Conference on Engineering Technology and Application*, 22:02029.

[6] Qu X., Hu R., Zhou L., Wang L., Zhu Q. (2015). Expert Achievements Model for Scientific and Technological Based on Association Mining. *International Symposium on Distributed Computing and Applications for Business Engineering and Science*, Guiyang, pp. 272-275.

[7] Bahrami M., Singhal M., Zhuang Z. (2015). A cloud-based web crawler architecture. *International Conference on Intelligence in Next Generation Networks*, Paris, pp. 2016-223.

[8] Pu Q. (2016). The Design and Implementation of a High-Efficiency Distributed Web Crawler. Dependable, Autonomic and Secure Computing, International Conference on Pervasive Intelligence and Computing, International Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress, Auckland, pp. 100-104.

[9] Kim M., Han S., Cui Y., Lee, H. Cho H., S. Hwang. (2014). CloudDMSS: robust Hadoop-based multimedia streaming service architecture for a cloud computing environment. *Cluster Computing*, 17(3): 605-628.

[10] Bhagyashree E., Tanuja K. (2015). Phishing URL Detection: A Machine Learning and Web Mining-based Approach. *International Journal of Computer Applications*, 123.

[11] Santhosh K. D. K., Kamath M. (2014). Design and implementation of competent web crawler and indexer using web services. *International Conference on Advanced Communication Control and Computing Technologies*, Ramanathapuram, pp. 1672-1677.

[12] Dąbek Osb T. M. (2012). Strengthen the faith as the task of the Pastors of the Church. The Apostles Peter and Paul as examples for the Pastors of the Church for proclaim and, *Scriptura Sacra*, (16): 19.

[13] Dong C. (2015). Asymmetric color image encryption scheme using discrete-time map and hash value. *Optik - International Journal for Light and Electron Optics*, 126(20): 2571-2575.

[14] Qiao Y., Yun X., Zhang Y. (2016). Fast Reused Function Retrieval Method Based on Simhash and Inverted Index. *Trustcom/BigDatase/ISPA*, Tianjin, PP. 937-944.

[15] Bal S. K., Geetha G. (2016). Smart distributed web crawler. *International Conference on Information Communication and Embedded Systems*, Chennai, pp. 1-5.

[16] Kumar N. and Singh M. (2016). Framework for Distributed Semantic Web Crawler. *International Conference on Computational Intelligence and Communication Networks*, Jabalpur, pp. 1403-1407.