Dis-Dyn Crawler: A Distributed Crawler for Dynamic Web Page

Jianfu Cai^{1, a}, Hua Zhang^{2, b}

¹Beijing University of Posts and Telecommunication, Beijing 100876, China;

²Beijing University of Posts and Telecommunication, Beijing 100876, China;

^abuildhappy512@163.com, ^bzhanghua_288@bupt.edu.cn

Keywords: distributed crawler, dynamic web page, HtmlUnit.

Abstract. Nowadays, it has become a widespread approach for achieving rich information in modern web applications using AJAX, which cause two serious problems for web crawler. One is the incomplete information getting from web page due to the inability to parse dynamic web page. Another is the efficiency of the crawler. In order to solve the above problems, this paper proposes a distributed dynamic web crawler naming Dis-Dyn Crawler. This system uses HtmlUnit to page dynamic pages and choose Redis and ZMQ (Message Queue Zero) to realize the distribution feature, which improve the efficiency of the crawler. The experimental results show that Dis-Dyn Crawler has better performance than Nutch-a distributed crawler system, and the dynamic page parsing efficiency is also improved.

Introduction

Web crawler is important comprising modules in the search engine[1], which can search various information from the internet and can help the users position the related information. The crawler process is the process of search engine carrying out webpage analysis[2], so it is very important to improve the efficiency of web crawler.

In recent years, with the development of Web2.0, Ajax technology obtains the wide application and this technology blends JavaScript, DOM (Document Object Model) and asynchronous server-side communication technology to reach better the interaction of users [3]. Frequently one URL page can change the page status seen by user through executing JavaScript code [4], but it will cause that the traditional web crawler can't obtain the information wanted by the users when withdrawing the web page information, because these information may be produced after the JavaScript document is executed in the client end and JavaScript can't be analyzed in the common web crawler tool, so the crawler task fails. For example, widely applied Google, Bing and the other search engine can't process dynamic web page at most time. Even the official Google suggests that if the dynamic web page in our website wants to be searched by Google, the server of this website shall provide HTML snapshot for each AJAX website (i.e. the contents seen by users through browser) [5].

In order to extract more and valuable information from the dynamic web page and guarantee the efficiency of crawler, this thesis adopts HtmlUnit[6] as the analyzing tool of dynamic page and introduces Redis and ZeroMq[7] to improve the efficiency of the whole system. Aiming at different websites needing to develop different crawler tool[8], using SOA structure thinking for reference, this thesis extracts some time-consuming operation to form independent service to mount to the system, like the dynamic page analyzing and so on, which not only guarantees the efficiency of distributed system but also makes the system function more perfect.

Dis-Dyn Crawler structure

For making the crawler system having distributed and dynamic characters in the meantime, we adopt HtmlUnit, ZeroMq, Redis and the other technologies to design one new web crawler system. Meanwhile, we introduce SOA structure so that the time-consuming operation in the crawler can be

separated to form the independent service, and then the throughput capacity of the whole crawler system can be improved. Figure 1 shows the whole system structure, composed by four functional modules such as crawler module, route module, resource downloading module and dynamic page analyzing module. Each module represents one kind of service and finishes the corresponding service

Fig.1. The Architecture of Dis-Dyn Crawler

Crawler module. One crawler module cluster includes several crawler modules that will begin to obtain from one or one group of seeds URL and analyzing the content of each page within the website. The purpose of designing this thesis into the cluster model is to make several nodes (multi-process or multi-machine) can crawl one website at the same time and the same URL can't be processed repeatedly among each node to realize the distributed working effect better. After analyzing the specific page, extract corresponding URL and the metadata information needed by users from the page. During realizing the project, we keep these URL into Redis buffer memory for the message interaction of each node among crawler module clusters, preventing different nodes from treating the same URL repeatedly. Extraction of information adopts Jsoup tool [9]. The internal realization of crawler module is shown as the following figure:

Fig.2. The Architecture of Crawler Model

Route module. This module is the communication center of the whole distributed system, used for retransmission of information and service management. The nodes in the other modules can be its client end to finish the message communication within the system. At the time of starting, the service node only sends one served registration message to the route module to inform the provided service by this node of this module, and then the route module will retransmit the information to the corresponding service modules once the information arrives. When the crawler module has the request for processing dynamic page or downloading resource, the route module will retransmit the corresponding request into the corresponding service modules among clusters. This module adopts ZeroMq message-oriented middleware [10], because this middleware is fast enough to meet the demand of distributed clusters [7].

Dynamic page processing module. This module mainly provides dynamic page analyzing function. There are several dynamic page processing modules in one dynamic page processing module cluster. After the crawler module meets the dynamic page (if the extracted metadata information is empty, we will regard this page as on dynamic page), the sound code information of this page will be resent to the dynamic page processing module by the route module to finish the analyzing of corresponding dynamic page. We choose HtmlUnit as the dynamic page analyzing tool

at last. HtmlUnit is one built-in browser without surface, which can realize most functions of browser, including JavaScript document execution & page rendering even simulating some clicking events. For improving the analyzing efficiency, we will cache the JavaScript document, CSS document and so on needed by HtmlUnit into Redis database. When carrying out page rendering, you don't need download from the internet each time, decreasing the network requirement and improving the analyzing efficiency. The specific working flow of dynamic page processing module is shown as the following figure:

Fig.3. The Workflow of Dynamic Page Processing Model

Each module within the system can communicate by ZeroMq message-oriented middleware, coordinating and working better, which not only finishes the analyzing of dynamic page with high efficiency but also makes full use of each node resource to improve the performance of the whole system.

Experimental results

All our tests are completed on Ubuntu12.0.4, gainestown, 4G internal storage and 3.10 GHz system.

To verify the distribution grasping efficiency of the system better, we choose Douban movie website for test. Take the grasped website quantity of stand-alone crawler system and Dis-Dyn Crawler system within 30 minutes respectively for contrast. Set 2, 4 and 8 machines within clusters respectively to carry out the performance test. In addition, we do the contrastive analysis with the performance of open source distributed crawler tool Nutch. We take BBS of BUPT as the test website to compare the performance of Dis-Dyn Crawler and Nutch.

Fig. 4. Performance comparison between Crawler Dis-Dyn and Nutch

Next we will analyze the efficiency of Dis-Dyn Crawler analyzing dynamic page. Take the target website--zhihu.com to carry out grasping to compare the performance of Dis-Dyn Crawler and dynamic page analyzing tool Crawljax in dynamic page analyzing.

Fig. 5. Performance of analysis for dynamic pages of Dis-Dyn Crawler and CrawlJax

The test result shows that Dis-Dyn Crawler has better performance result in aspects of grasping and dynamic page analyzing.

Conclusion

This text puts forward and designs one distributed dynamic page crawler frame. This frame adopts Redis and ZeroMQ tool to finish the function of distributed grasping, introduces HtmlUnit to carry out dynamic page analyzing. The whole structure uses SOA structure thinking, make the efficiency of the entire distributed dynamic page crawler system become higher and support dynamic page analyzing better at last.

References

- [1] BING Z, BO X, LIN Z Q, CHUANG Z. A Distributed Vertical Crawler Using Crawling-Period Based Strategy, Proceedings of 2010 2nd International Conference on Future Computer and Communication, V.1, 306-311, 2010.
- [2] A. Guerriero, F. Ragni, C. Martines. A dynamic URL assignment method for parallel webcrawler, Computational Intelligence for Measurement Systems and Applications (CIMSA), 119-123, 2010.
- [3] MESBAH A, DEURSEN A V, LENSELINK S. Crawling AJAX-Based Web Applications through Dynamic Analysis of User Interface State Changes, ACM Transactions on the Web (TWEB), V6:Article No.3, 2012.
- [4] Seyed M. Mirtaheri, Di Zou, Gregor V. Bochmann, et al. Dist-RIA Crawler: A Distributed Crawler for Rich Internet Applications, Proceedings of 2013 Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 105-112, 2013.
- [5] Information on <https://support.google.com/webmasters/answer/174992?hl=en>.
- [6] Information on <http://htmlunit.sourceforge.net/>.
- [7] Pieter Hintjens. ZeroMQ- Messaging for Many Applications, first ed. O ' Reilly Media, California, 2013.
- [8] YAN G, KUI L, KAI Z. Board Forum Crawling: A Web Crawling Method for Web Forum, Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, 745-748, 2006.
- [9] Information on <http://jsoup.org/>.
- [10] Information on <http://zeromq.org/>.