# A Distributed Web Crawler Model based on Cloud Computing

JIANKUN YU[1, a *], MENGRONG LI[2, b] and DENGYIN ZHANG[2, c]

[1]College of Internet of things, Nanjing University of Posts and Communications, No.66, Xin Mofan Road, Nanjing, Jiangsu Province, China

[2]College of Telecommunications & Information Engineering, Nanjing University of Posts and Communications, No.66, Xin Mofan Road, Nanjing, Jiangsu Province, China

[a]ycyujiankun@126.com, [b]583069505@qq.com, [c]zhangdy@njupt.edu.cn

**Keywords:** web crawler; cloud computing; distributed; cloud-based web crawler.

**Abstract.** With the rapid development of the network, distributed Web Crawler was introduced for fetching the massive web pages. However, the traditional distributed Web Crawler has disadvantages in load balancing between different nodes. In addition, the number of fetching web pages had not grown up linearly in the case of extended crawling nodes. This paper proposes a distributed web crawler model which runs on the Hadoop platform. The characteristics of Hadoop guarantees the scalability of the crawler model proposed by this paper. At the same time, the crawler model makes good use of HBase to guarantee the storage service of massive web context data. This paper also proposed a method of load balancing which is based on the feedback of crawling nodes. The crawler model has been proved to have good performance in load balancing and node extension.

## Introduction

By September 2014, the number of websites have been more than 1 billion all around the world and is still in rapid growth. Due to the explosive growth of the Internet information content, web search engines are becoming increasingly important as the main means of locating relevant information. A complete search engine is comprised of different parts like web crawler system, page index system, page sort system and user interface [1]. Till now, a lot of scholars are carrying out the relevant research works about web search engine. These works intend to solve the problems like how to increase the speed of web crawler, how to store and index the massive web pages and how to sort the pages which are returned to users as a result.

Traditional web crawlers are deployed on a single server. But with the rapid development of the network, information growing quickly, the crawling ability of traditional simple stand-alone web crawler has been unable to keep up with the growth of information on the Internet. Distributed web crawler architecture was introduced for crawling web pages and providing index service. Distributed web crawlers can be classified as independent architecture with no scheduling center, dynamic allocation architecture with a scheduling center [5] and static allocation architecture by the architecture. An independent architecture with no scheduling center may crawl a lot of duplicated pages. A dynamic allocation architecture with a scheduling center may risk of a single point failure. Static architecture always assign pages to servers according the IP address or geographical location. It is difficult to ensure the load balancing on different servers.

Cloud computing is the integration of traditional computing and network technology, such as distributed computing, parallel computing, network storage and load balancing technique. This paper proposes an improved distributed web crawler model based on the traditional distributed crawler architecture and cloud computing which will overcome the load balancing problem and be easily to extend crawling nodes.

## A distributed web crawler model based on cloud computing

**System architecture.** The distributed web crawler proposed by this paper is based on master-slaver architecture as shown in Fig. 1. The model consists of several specialized components - cloud storage system, scheduling system, distributed processing system and multiple crawling systems. All the components can run on different machines. We store all the data which can be used in the cloud storage system. Such as the initial seed URLs, to be fetched URLs, fetched URLs and the content of fetched web pages. The scheduling system is responsible for assigning URLs to the crawling nodes. And it should make load balancing between all the crawling nodes. Crawling systems run on different crawling nodes which are in charge of fetching the web pages assigned to them. Distributed processing system receives the pages fetched by crawling system and then execute some processing work. The crawling system and processing system can be easily extended under the framework of Hadoop.

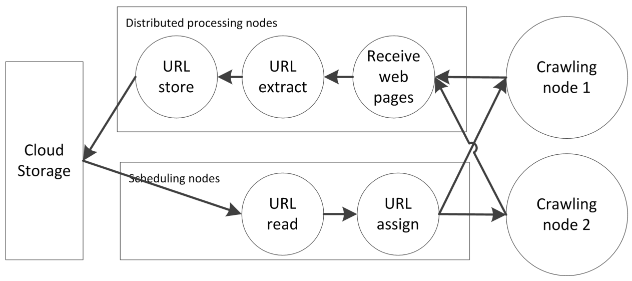


Fig. 1. Architecture of distributed web crawler

**Crawling process.** The distributed crawler system uses HBase to provide storage service. According to the characteristics of HBase, the row-key cannot be duplicated. We can define the URL as the row-key so that every URL will be crawled only once. The following steps are a complete crawling process which is shown in Fig. 2:

a) When the system is initialized, the scheduling system reads the URLs from 'to be fetched' table, and then assigns the URLs to different crawling nodes according to the load balancing algorithm which will be proposed in the next section.

b) The crawling node will fetch the web pages asynchronousily as soon as it receives the URLs from the scheduling system. When all the URLs are fetched, the crawling node will send all the web pages to HDFS and request new URLs from the scheduling system. At the same time, the scheduling system will notify the distributed processing system to start a new page processing job.

c) The distributed processing system will check the format of the web pages and convert the page encode to UTF-8. Then extract the URLs from the web content which will be marked as to be fetched. Then the URLs and the web content will be saved in HBase.

d) The crawling system will exit when all the URLs in HBase have been crawled.

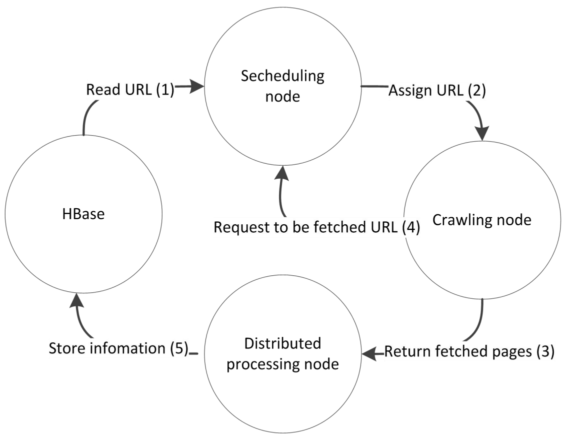


Fig. 2. The steps of web crawling

**Load balancing.** Load balancing strategy proposed by this paper is a kind of dynamic algorithm based on the feedback of crawling nodes. The main factor of the load balancing strategy is the time of every crawling node costs. When all the crawling nodes finish the crawling job at nearly the same time, we can regard the crawling nodes as balanced.

Initially, the scheduling system will maintain a base value which represents the URLs count assigned to a single crawling node. Also, for a single crawling node, an average time value pavg will be calculated to indicate every single job of a crawling node costs. An average time value tavg will be calculated to indicate every single job of all crawling nodes costs. The scheduling system will assign the URLs according to Eq. 1. Thus the crawling nodes which crawl slower will be assigned less URLs.

$$c = \max(base * tavg / pavg, base / 4). \quad (1)$$

## Experiment

**Computer configuration.** We do experiments on 10 virtualized computers running Hadoop1.0.4 and HBase0.94. One of them is master node and the rest are slaver nodes. The CPU of every virtualized computer is 2.0GHz and the RAM is 1G. The distributed crawling system proposed by this paper will be compared with the well-known open-source crawler Nutch2.3 in crawling speed and load balancing.

**Result analysis.**

A. Result of crawling speed

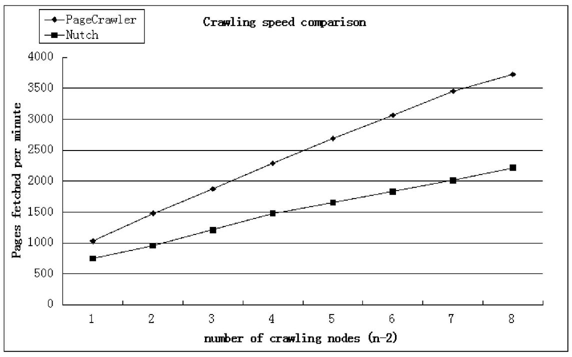
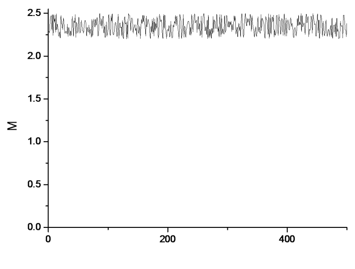


Fig. 3. Crawling speed comparison between PageCrawler and Nutch

As we can see from the Fig. 3, the crawling system proposed by this paper shows better crawling speed performance than Nutch and the performance grows nearly linearly when adding crawling nodes to the system.

B. Load balancing performance



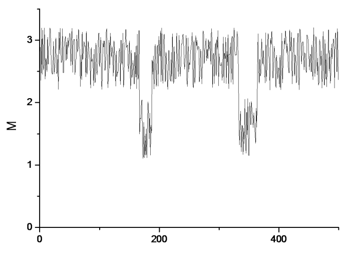


Fig. 4. PageCrawler network load

Fig. 5. Nutch network load

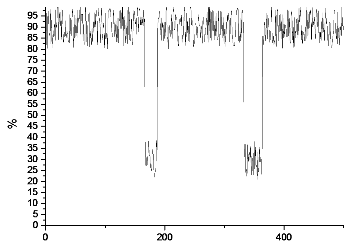


Fig. 6. PageCrawler CPU load

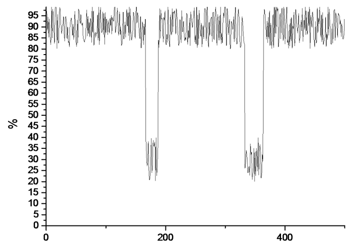


Fig. 7. Nutch CPU load

We gathered the network loads and CPU loads from one of the crawling nodes. The data shows that PageCrawler can make full use of the network resource. Because of PageCrawler and Nutch both use the MR framework of Hadoop platform. The CPU may not be fully used when submitting and distributing jobs.

## Summary

We have proposed a distributed web crawler model based on cloud computing which aims to solving the extension and load balancing problem exists in traditional distributed crawling system. By parallelling the crawling nodes and processing nodes, we can better use the network resources and CPU resources of a single computer. Web crawler is an important component of web search engine. In order to making a deep study of the search engine, we will research on text duplication detection and page indexing which are also vital to a complete search engine.

## Acknowledgement

The work was partially supported by Swedish Research Links [No.348-2008-6212], the National Natural Science Foundation of China [61571241], Industry-university-research Prospective joint project of Jiangsu Province [BY2014014]， and Major projects of Jiangsu Province university natural science research [15KJA510002].

## References

[1] Ge D, Ding Z. A Task Scheduling Strategy Based on Weighted Round-Robin for Distributed Crawler[C]//Utility and Cloud Computing (UCC), 2014 IEEE/ACM 7th International Conference on. IEEE, 2014: 848-852.

[2] Fei L, Fan-Yuan M, Yun-Ming Y, et al. Distributed high-performance web crawler based on peer-to-peer network[M]//Parallel and Distributed Computing: Applications and Technologies. Springer Berlin Heidelberg, 2005: 50-53.

[3] Zhu K, Xu Z, Wang X, et al. A full distributed web crawler based on structured network[M]//Information Retrieval Technology. Springer Berlin Heidelberg, 2008: 478-483.

[4] Shkapenyuk V, Suel T. Design and implementation of a high-performance distributed web crawler[C]//Data Engineering, 2002. Proceedings. 18th International Conference on. IEEE, 2002: 357-368.

[5] Wu M, Lai J. The Research and Implementation of parallel web crawler in cluster[C]//Computational and Information Sciences (ICCIS), 2010 International Conference on. IEEE, 2010: 704-708.