2009 Second International Workshop on Computer Science and Engineering

The Implementation of a Web Crawler URL Filter Algorithm Based on Caching

Wang Hui-chang
School of Computer Science
Sichuan University
Chengdu Sichuan 610065, China
e-mail: huichangwang@qq.com

Ruan Shu-hua
School of Computer Science
Sichuan University
Chengdu Sichuan 610065, China
e-mail: ruanshuhua@scu.edu.cn

Tang Qi-jie
School of Computer Science
Sichuan University
Chengdu Sichuan 610065, China
e-mail: tangjiezhen1981@163.com

Abstract *— For large-scale Web information collection, the URL filter module plays important roles in a Web crawler which is a central component of a search engine. The performance of an URL filter module influents the efficiency of the entire collection system directly. This paper introduces one URL filter algorithm based on caching and its implementation. The performances of stability and paralleling of the algorithm are verified by the experiments for Websites which handle a large number of web pages. Experiment results show the algorithm proposed in this paper can achieve satisfactory performances through reasonable adjustments of its some parameters and it is suitable for the process of the URL filter of a Website which has a number of page navigator links and index pages especially.*

Keywords - Web Crawler; URL Filter; Caching

I. INTRODUCTION

Accompanied by the informationization of society and rapid development of the Internet, more and more information is on the Internet. These make search engine, one of the main applications of modern information technology, become one necessary tool by which people can get information through the Internet more easily and quickly. A web crawler, which is a central component of a search engine, impacts not only on the recall ratio and precision of a search engine, but also on the capacity of the data storage and efficiency of a search engine.

World Wide Web can be viewed as a directed graph without rules and borders, and a wide range of circles exists in it. Due to a large number of index pages and navigation contents existed in a Website, as well as a large number of relevant content links existed in a web page, the URLs get from web pages are much more than the actual number of URLs existed in the Internet.

The URL filter module, which is an important component of a crawler, is used to filter the URLs analyzed from the web pages downloaded by the crawler of a search engine so as to improve the efficiency of the crawler. We design an URL filter algorithm and implement it based on caching. Experiment results show the algorithm proposed in this paper can achieve satisfactory performances.

II. A WEB CRAWLER

A Web crawler (also known as a Web spider or a Web robot) is a program or an automated script which can get specified pages from the Internet, extract the links from these pages and follow the links to find new pages and links [1,2]. A Web crawler usually starts from a URL address alone or URL list to visit each pages specified by each URL, extracts the links in each page by the content analysis and eliminates the repetitive URLs after download each page, then adds the new links to the URL list.

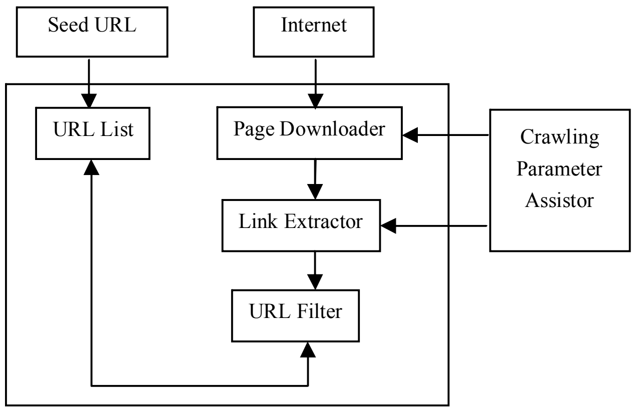

Figure 1. A simplified web crawler

It is a simplified Web crawler in Figure 1. According to Figure 1, a Web crawler starts an URL called the Seed URL to visit the Internet. The Page Downloader gets an URL from URL List to download the page from the Internet, transfers the page to the Link Extractor. The Page Downloader checks the parameters in accordance with the requirements of crawling to decide whether or not to download pages. As the crawler visits these URLs, the Link Extractor indentifies all the hyperlinks in the pages in accordance with the requirements of crawling and transfers them to the URL Filter, stores the results into URL list. The Crawling Parameter Assistor provides the parameter setting for the needs of all parts of the crawler.

III. THE COMMON METHODS OF THE URL FILTER

In order to avoid downloading the same page repeatedly, we need to eliminate the repetitive URLs. The URL filter can easily become a bottleneck in the performance of the crawler system because the URL list to be inquired is growing bigger and bigger.

The main idea of the URL filter is to determine whether an URL exists in the known URL list. The URL query algorithm is the key. The basic process is for each string of the URL analyzed by a web page to be compared with the each string in the known URL list. If it is a repetitive URL, then be abandon directly, or a new URL, then be added to the URL list.

The common methods of the URL filter are the followings.

A. *The URL filter based on a hash table*

Hash table is a linear list stored by hashing method. According to the method of dealing with conflict, the given hash function h(key) converts a set of keywords to a limited set of consecutive addresses (or range) in which the “map” of keywords is the storage location. The process is called hashing and the storage addresses is called hashing addresses or hash table. Hash table is suitable to deal with finding string because it converts the string to an integer and associates the string with an address to control the complexity of search algorithm within O(1), which is entirely superior to the other search algorithms [3].

The URL was stored as the key of a hash table if we use hash table to store the URL list. For each URL requesting to add into list, we can search hash table fast to be aware of the URL whether or not a repetitive one. If not a repetitive one, we can deposit it into the hash table. Hash table also has the shortcomings that cannot be sorted and traversed and is not suitable for mass data storage for being subject to the limitations of memory capacity.

B. *The URL filter based on a Bloom Filter*

A Bloom Filter is a collection querying algorithm with high efficiency in space and time by using a hash function and representing data with a bit vector, which is suitable for mass data querying operation [4]. For a Web crawler, there is an inconsistency between a large number of URLs and loading URLs into memory for fast querying. But for each URL, a Bloom Filter produces random k-values by using k-hash functions and set 1 on the corresponding location of a bit vector. So if we use a Bloom Filter to store URL list, the size of the URL list is only the size of the bit vector and we can optimize the space greatly and make querying fast under the restrictions of time and the rate of miscarriage of justice. Towards every URL which requests to add into the list, k kinds of hashing calculations shall be done on it. And k numbers generated as per calculation method shall be used to set the correspondence position of the bit vector to be 1. Before each positioning, determination shall be firstly given whether or not the k correspondence positions of the bit vector through hashing calculations have been set to be 1.If so, determination can be given the URL has already consisted in the list and is a repetitive one. Otherwise position 1 opposite to the bit vector shall be added to the URL list as a new URL.

A Bloom Filter has the advantages of high speed and saving storage space, so mass data can be represented easily and accessed fast, but in the use of a Bloom Filter to determine whether an element belongs to a set, there will be a certain false positive rate. That is, an element that is inexistent in a set may be mistakenly believed that it already consisted in the set. But an element that is already existent in a set can’t be mistakenly believed that it has not consisted in this set, there is no false negative. A Bloom Filter can save a lot of storage space by allowing a small amount false positive rate.

IV. THE URL FILTER ALGORITHM BASED ON CACHING

The integrated ideas to deal with the URL filter are that the crawler program downloads a page to the local and parses out URLs from the page, then passes the list of URL to the URL Filter for eliminating the duplications. URL Filter as a key component of the crawler provides non duplicating URL address for crawler to crawl on the Internet, and as a storage place for parsed URLs communicates with database to complete the operations of access URL data at the same time.

Due to the limited capacity of computer memory, the capacity of URL caching queue is limited accordingly. There is a large number of duplicating URL addresses in some site, and their presenting frequencies are relatively high. The URL filter algorithm based on caching will greatly improve the rate of hits by putting these high presenting frequency URLs into the URL caching queue. According to a cache replacement algorithm, we can improve the efficiency of the crawler system by maintaining the URL caching queue with a high hit rate and reducing the frequency of database access. The specific working principle is: when the crawler parses out an URL, we query in the caching queue first, if find the URL to increase its repetition rate to the page but not to deal with, if not find it to query the database on a relatively slow speed, then if find it to put it into the different caching queues according to if it had been visited. Otherwise, load the new URL list which includes all the URLs under the same directory into the caching queue with waiting for deal and reduce the number of database access.

It is because of this kind of access mechanism that the hit rate of the caching queue is very high, which means the most of URLs parsed by crawler program are in the cache queue, only about 10%-20% of them needed to check in the database. This greatly saves time to query a new URL and need not query database in the most situations. In general, the sequence of crawler program to determine whether the URL is a repetition is the caching queue first, then database.

The URL filter algorithm based on caching includes some parameters, such as caching capacity, exchange rate of cache and external storage, depth of crawling etc. For a variety of network environment, the corresponding parameters can be adjusted to achieve the corresponding performance requirements.

In this paper, the crawler URL filter system based on the cache strategy was implemented by which combined with Java 2 platform and MySql5.0 DBMS.

The URL filter algorithm which is based on caching shows in Figure 2.

The description of the algorithm is as the followings:

1) To initialize the lists of URL in the URL Filter.
   There are two URL caching lists in the URL Filter, one is visitingList in which URLs will be visited and the other is visitedList in which URLs had been visited respectively.

If there is an URL list of Websites to be crawled in database, select the number of initNum of URLs which are ordered descending by the number of repetition to initialize visitingList, otherwise initialize the visitingList with the index page of the Website.

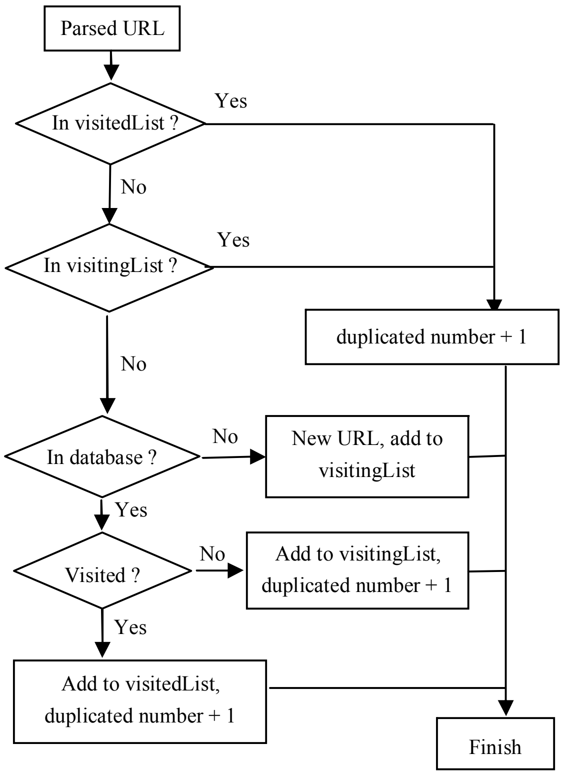

Figure 2. URL filter algorithm based on caching

2) To add the URLs parsed from one page and eliminated duplication into the related URL lists in the URL Filter.
The specific filter steps are: for a parsed URL, searching in the visitedList of URL Filter first, if found, it is a repetition of some URL and increase its duplicated number, otherwise searching in visitingList. If found, it is a repetition of some URL and increase its duplicated number, otherwise searching in database. If not found, it is a new URL and add it to visitingList, otherwise determining it a repetition of some URL, increasing its duplicated number and adding it to the related URL lists according to it had been visited or not.
3) If the size of the URL caching lists in the URL Filter is larger than the setting value of the crawling parameters, transfer and store the URL collection with the setting capacity into database, otherwise load the some block of URLs from database into the URL lists according to the replacement strategy.

V. ALGORITHM IMPLEMENTATION AND PERFORMANCE EVALUATION

The performance of the URL filter algorithm based on caching, which is used in large-scale webpage to eliminate duplication, was analyzed and validated through experiments by the prototype system.

In the experiments, having set the parameters (caching capacity, exchange rate of cache and external storage, and depth of crawling) according to different crawling situations, we obtained basic data from a wide variety of Websites, and then got the interrelated data for URL filter based on the synthetic analysis of basic data. The effect comparison of URL filter algorithm, which is implemented in this paper, shows in Figure 3, in which the average links of a page (the average number of effective links per page in a Website) is as X-coordinate, the rate of hitting the cache (the ratio of a pending URL can be found in cache) is as Y- coordinate.

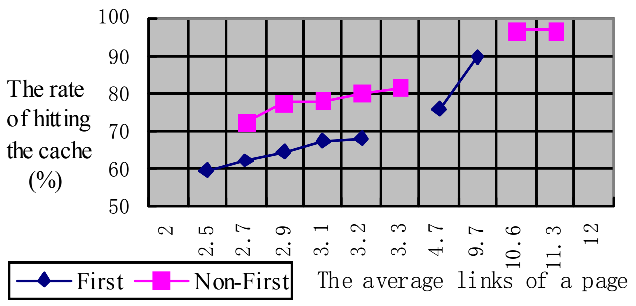

Figure 3. The effect comparison of URL filter

We know, from the experimental data, the higher the average links of a page, the higher the rate of hitting the cache, the more efficient the URL filter, especially when crawling the Website again, the rate of hitting the cache has been increased significantly and the performance of the URL filter has been improved accordingly. So that it is very efficient to crawl the Website which has a large number of page navigator links and index pages by using the web crawler URL filter algorithm based on caching. In addition, a large number of experimental data shows that the algorithm has consistent performances of stability and parallel processing while crawling all types of Websites. The experimental data shows in table 1.

The experimental data in table 1 indicated that with the increase in the number of links, the rate of hitting the cache also increase, in particular, the higher the average links of a page, the higher the efficiency of eliminating duplication. The prototype system, which centers on the database storage, has more than one processing node and each node can crawl parallel. The performance of eliminating duplication of each node will not reduce with the increasing of node, and the time processing performance of the whole system only will be influenced by database concurrent processing and network environment, so the stability and parallel performances of the algorithm are good.

Table I. URL filter data for different types of Websites

Website URL	NL	NLAF	RHCF	RHCNF
http://cs.scu.edu.cn	1408	399	59.5	100
http://www.jtstar.com	13986	1321	89.7	96.6
http://www.scu.edu.cn	23550	7837	64.5	77.9
http://news.sina.com.cn	38763	11861	67.1	78.7
www.xinhuanet.com	61918	10368	68.0	88.6

where:
NL — Number of links
NLAF — Number of links after filtering
RHCF — the Rate of Hitting the cache (First %)
RHCNF — the Rate of Hitting the cache (Non-First %)

This paper introduces one URL filter algorithm based on caching and its implementation. The stability and parallel performances of the algorithm are verified by the experiments for Websites which handle a large number of web pages. Experiment results show the algorithm proposed in this paper can achieve satisfactory performances through reasonable adjustments of its some parameters and is suitable for the process of the URL filter of a Website which has a number of page navigator links and index pages especially.

REFERENCES

[1] Yuan Wan, and Hengqing Tong, "URL Assignment Algorithm of Crawler in Distributed System Based on Hash," IEEE International Conference on Networking, Sensing and Control, 2008.

[2] WU Lihui, WANG Bin, and YU Zhihua, "Design and Realization of a General Web Crawler," Computer Engineering, February 2005, pp.123-124.

[3] Christopher Martinez, Wei-Ming Lin, and Parimal Patel, "Optimal XOR Hashing for A Linearly Distributed Address Lookup in Computer Networks," Symposium On Architecture For Networking And Communications Systems, 2005, pp. 203-210.

[4] Xiao-Guang Liu, and Jun Lee, "K-Divided Bloom Filter Algorithm and Its Analysis," Future generation communication and networking (fgcn 2007), 2007, 1(6-8), pp. 220-224.