Research and Implementation of Web Crawler Technology on Cloud Platform

Zijing Shi

Changjiang Institute of Technology, Wuhan, Hubei, 430212, China

Keywords: Web Crawler Technology, Cloud Platform, Implementation Way

Abstract: With the continuous development of the Internet, our access to information has gradually been replaced by the network, but at the same time the amount of network information is growing at an alarming rate. In the face of such massive data, how to quickly and accurately collect the required data is the current research hotspot. At present, many companies have adopted distributed web crawling technology to improve the efficiency of crawling, and use multiple machines to crawl network data from the Internet in parallel. This paper designs and implements a distributed web crawler system built on the cloud platform, which utilizes various features of the cloud platform to improve the performance and scalability of the web crawler.

1. Introduction

The various information on the network brings us convenience, but at the same time, because of its complicated data and large amount of data, it also brings us the problem of information overload, so it faces the massive network data on the Internet, How to get the data we need quickly and efficiently is an increasingly urgent problem to solve. At the end of the last century, the Yahoo search engine solved the problem of Internet information overload at that time, which can be regarded as creating a miracle on the Internet. However, the development speed of the Internet is very rapid, and the amount of network data continues to grow. This problem of network information overload has appeared again. Yahoo's previous catalogue information collection website has been unable to solve the problem of massive data. . Since then, Google's search engine has slowly evolved to replace Yahoo's status and become the next generation of access to Internet information. Google is using web crawler technology to solve the problem of information overload. It first collects each from the Internet. A variety of network information is collected, processed and saved, and then the collected information is indexed, so that users can quickly retrieve the information they need by using Google search engine. This mode makes it unnecessary for users to use the directory navigation page to free users from it, which greatly improves the efficiency and quality of information acquisition, and Google has become the second generation of the Internet.

2. Web crawler basic structure

A web crawler is a program that automatically extracts network information from the Internet, usually as a tool for network data collection. All web pages on the Internet can be viewed as a huge graph. Each node in the graph represents a web page. The directed edge represents a hyperlink between the web page and the web page. Using the graph traversal algorithm, starting from some web pages, you can Keep accessing other web pages on the internet. Therefore, a common web crawler generally starts with an initial URL, extracts a new URL from the downloaded web page while downloading the web page, adds it to the URL queue, and then retrieves the new URL from the URL queue. The URL performs this process, and the process is looped until the previously set conditions are not met.

The basic flow of a generic web crawler is as follows: 1) establish an initial URL set; 2) add the initial URL set to the URL queue to be crawled; 3) take a URL from the URL queue to be crawled, use the URL The corresponding webpage is downloaded from the Internet, and then the webpage is parsed, the parsed webpage target data is saved in the webpage library, and the parsed new URL is put into the newly crawled URL queue. 4) Extract a URL from the newly retrieved URL queue and then de-reprocess it. If the URL has not been crawled, save the URL to the URL queue to be crawled, otherwise discard the URL. URL. As can be seen from the above process, the web crawler is a ring structure, and its operation is a cyclic process. The following is a sub-module to introduce the web crawler: 1) The initial URL set. The initial URL is the starting address of the web crawler, which is usually a pre-selected URL to be crawled. 2) The URL queue to be crawled. The URL queue to be crawled stores a list of tasks for the web crawler, and the crawler will retrieve the URL from this list to crawl the web page. 3) DNS (Domain Name System) analysis. The DNS resolution module is designed to reduce the number of accesses to the DNS server, thereby improving crawl efficiency. 4) Web page download. The module downloads the specified resources from the Internet based on the specified URL. The HTTP resource (HyperText Transfer Protocol) protocol is commonly used to fetch web resources. According to the response code of the protocol, it is judged whether the fetching is successful, for example, sending an HTTP request to the host where the resource pointed to by the URL is located, if the request is successfully obtained. The resource returns a status code of 200, indicating that the request was successful. If the requested resource is not on the server, a 404 error code is usually returned. HTTP status response codes are usually divided into five types, each consisting of three digits, starting with 1~5 five digits.

3. Research on web crawlers on cloud platforms

The friendly web visual management interface allows ordinary users to perform crawling tasks using the web crawler system of this article by performing simple operations on the web page. Users of each crawler task can be highly customized to meet the task requirements, and can monitor the various nodes of the system to understand the current resource usage. It mainly includes three parts: crawler task creation, crawler task management, and crawler node management. The crawler task creation is to create a crawler task by the user inputting the crawler task condition through the webpage; the crawler task management is to display the crawler task situation on the webpage, and can perform the crawler task, stop the crawler task, delete the crawler task, and execute the crawler task for each crawler task. Edit the crawler task operation; crawler node monitoring is to display the status of the current crawler node on the web page.

Web crawler is a program that automatically extracts web pages. For large-scale data collection, it usually takes a long time to complete on a single machine, and the crawling efficiency is low. Therefore, the web crawler of this paper combines the characteristics of the cloud platform of the lab to realize a web crawler system by using the virtual machine computing resources, network resources and storage resources provided by the web crawler. Unlike the traditional web crawler system, the web crawler based on the cloud platform, the crawler program does not run on the actual computer, but the virtual machine applied to the cloud platform. This is beneficial to the expansion of resources and can be done on demand. Application.

The web management layer is designed to simplify the operation of the web crawler in this article, so that ordinary users can easily implement web crawling tasks with simple operations. The web management layer mainly includes three parts: creating a crawler task, managing a crawler task, and monitoring a crawler node. On the Web page, users can create new crawl tasks as needed. When creating a task, you need to enter some task configuration parameters, such as the initial URL set, web page content extraction rules, and other additional settings. For the successful crawl task, you can view the basic information through the web page. The displayed information mainly includes the name of the crawler task, the start time, the end time, the number of pages that have been crawled, the number of failed pages, and the running status of the task. These crawler tasks can be started, stopped, edited, and deleted. Through the crawler node monitoring page, you can view the running status of the current crawler node.

The control node not only is responsible for the message communication of the entire system, but also maintains the URL generated during the execution of the crawling task, uniformly manages the URLs resolved by all crawling nodes, and de-duplicates the URLs and stores them in the URL library. The control node needs to ensure smooth communication with each crawling node, so that the entire web crawler system continues to work stably, but does not perform any web crawling and web page parsing work. The main job of the crawler node is to download web pages, parse web pages, and store target data. The webpage download is the most important work of the crawler node. The crawler node obtains the URL from the control node and downloads the webpage. The crawling result is recorded regardless of success or failure, and the webpage that fails the download is processed according to the error code returned by the webpage. The web page parsing valid URL and target data from the downloaded original webpage, and sending the parsed URL to the control node, and the control node processes the URL and stores it into the URL library as the URL seed for the next webpage crawling. The crawler nodes do not communicate with each other. It is only responsible for downloading the webpage, and then sends the parsed URL to the control node, and the parsed target data is stored in the specified storage space, so that the coupling degree between the crawler nodes can be made. Smaller, which is conducive to the expansion of crawler nodes.

Web crawlers generate a large number of URLs during the run, and these URLs may have many identical or seemingly different but actually point to the same web page. If these URLs are not processed, the crawler will repeatedly fetch the same URL. Resources, repeated crawling not only wastes computing and storage resources, but may even form a "loop" to cause an infinite loop, seriously affecting the efficiency of the entire network crawler system. Therefore, in order to ensure that the system crawls web pages efficiently, before the URL is saved to the URL queue to be crawled, the newly extracted URL needs to be pre-processed and then deduplicated. URLs that look different but actually point to the same page need to be standardized, turning them into a compliant equivalent URL according to a set of specifications. For example, the two URL addresses "http://www.test.com/" and "HTTP://WWW.TEST.COM/", because the URL is not sensitive to the case of the protocol name and host name, these two URLs It actually points to the same location, converting its protocol name and domain name to lowercase when normalizing a URL, so that you can determine if two URLs are equivalent. A standardized URL needs to determine if the URL has been crawled, ie the URL is deduplicated. This article chooses Bloom filter because the distributed web crawler needs to be deployed on multiple machines, it is very likely that the crawler node will crash due to some failures during the crawling process. In addition, when the amount of data fetched is large, new crawling nodes may be added. Therefore, in order to be able to handle such a problem of dynamically adding or removing crawler nodes, it is necessary to monitor and manage the running status of all crawler nodes. This module displays the running status of all current crawler nodes through the web page. The displayed information includes the crawler node host name, the crawler node IP, the number of crawler tasks, whether the crawler node is in the task crawl state, and whether the crawler node is normal. The information is mainly from Read in the worker_node Table of the MySQL database. The module involves heartbeat and heartbeat. The receiving heartbeat module on the control node is responsible for receiving the heartbeat information of all crawler nodes and saving it to the MySQL library. The heartbeat module is sent to the crawler node, which is responsible for sending to the control node. Heartbeat information.

4. Conclusion

This paper gives an overview of the research status and related knowledge background of distributed web crawlers at home and abroad. Then, based on this, combined with the characteristics of the laboratory cloud platform, a distributed network crawler system on the cloud platform was designed, and the various modules of the web crawler were studied and designed in detail. Then, it is specifically implemented using the Java development language. Finally, for the actual crawling task, the web crawler system of this paper has been tested in detail in terms of function, performance and scalability, and the test results are analyzed.

References

[1] Shi Enming, Xiao Xiaojun, Lu Yu. Research and Design of Distributed High Performance Web Crawler Based on Cloud Platform[J]. Telecommunications Science, 2017(8).

[2] Shen Cong, Dai Xiaopeng, Fan Zhenyu. Design and Implementation of Mobile Agriculture Information Service System Based on Web Crawler[J]. Hunan Agricultural Sciences, 2017(6).

[3] Huang Yong, Zhu Weihua, Xi Jun. Design and Implementation of Patent Query Platform for Screw Expander Based on Web Crawler Technology[J]. Science and Technology, 2016(1): 36-36.

[4] Zhu Lina, Li Zeping. Research and Implementation of Web Reptile Technology [J]. Science and Technology Innovation, 2017(10):166-166.

[5] Lin Haixia, Si Haifeng, Zhang Weiwei. Research and Implementation of Topic Web Crawler Based on Java Technology [J]. Microcomputer Applications, 2009, 25(2): 56-58.