TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol. 12, No. 9, September 2014, pp. 6954 ~ 6957
DOI: 10.11591/telkomnika.v12i9.4682

Investigation of Distributed Search Engine Based on Hadoop

Ning Chen*, Chai Xiangyang
College of Computer Science, Xi'an Polytechnic University, Xi'an, China
*Corresponding author, e-mail: chennvictor@gmail.com

Abstract

This paper begins with a review on the research status of search engine, followed by discussion on goals of search engine, and then the principle of distributed computing is explained. Consequently the MapReduce distributed computing model and the Hadoop distributed file system (HDFS) are analyzed in detail. Finally the distributed search engine architecture is presented. On the basis of the architecture, future challenges and opportunities of the distributed search engine are highlighted.

Keywords*: search engine, hadoop, mapreduce, distributed file system architecture*

1. Introduction

The emergence and development of search engine are inseparable from the vigorous development of the Internet. Under the information boom, how to meet users' requirements of finding contented pages quickly is increasingly becoming a more and more important hotspot, so the goals which the search engine needs to meet, can be summarized as: more comprehensive, quicker, more accurate. Google adopted the PageRank algorithm to evaluate the weights of the sites according to the webpage of mutual links, which greatly improve the precision of the search engine. Google needs to deal with the huge amounts of data and complicated calculation, which can be run on a cheap cluster of cloud computing platform and keep the high efficiency and the good scalability. Hadoop is an open source software that is a distributed computing programming tool and distributed file system platform, which mainly includes two parts: the MapReduce distributed computing model and Hadoop distributed file system (HDFS). They are open source implementation based on Google MapReduce computing model and Google file system [1] .

2. Distributed Computing

The basic principle of distributed computing is that a complex problem is divided into several subproblems and these subproblems are calculated by independent parallel computing devices. MapReduce is an important technology of Google, it is a kind of simplified parallel computing programming model, which makes these developers who have not much parallel computing experience can develop parallel applications [2]. MapReduce can implement the massive data retrieval, which can divide massive data into a plurality of small blocks calculated in a distributed method, and then map them to a Reduce center, so as to achieve rapid processing [3]. MapReduce distributed programming model has two core operation: Map (mapping) and Reduce (reduction). The principle of MapReduce is the divide and conquer method. In the computing model, the main node firstly splits the input data sets into smaller subdata sets; second, the subdata sets are processed by work nodes. If the running work nodes in the model are too many (hundreds of thousands), work nodes may do the above operation again, then the problem sets will become into a multi-layer tree structure. When the subdata sets are calculated, the results will be returned to the master node. The master node collects all the data sets results and classifies them, then master node gets final results. The execution process is shown in Figure 1.

Built on the basis of distributed computing, the procedures can be automatically distributed to a large cluster that is composed of general machines and can be executed

Received October 9, 2013; Revised June 7, 2014; Accepted June 27, 2014

concurrently. The system will deal with the details about distribution of input data, such as crossing clusters of machines, processing machine's failure, and managing communication between different machines. Such a model allows programmers with not much experience of concurrent processing or knowledge of distributed systems to make full use of the resources of distributed system.

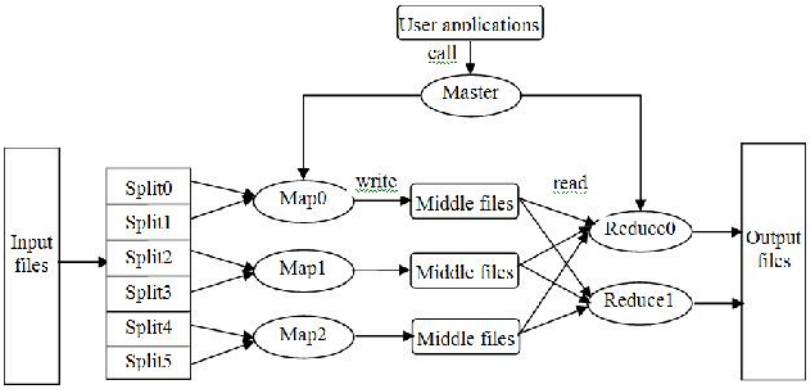

Figure 1. Execution Process in MapReduce Model

3. Distributed Storage

HDFS adopts Master-Slave architecture. An HDFS cluster is composed of a NameNode and a certain number of DataNodes. NameNode is a central server that responsible for the management of the file system's namespace and the access to files by clients. Datanodes are the work horses of the file system. They store and retrieve blocks when they are informed (by clients or the NameNode), and they send heartbeat report back to the NameNode periodically with lists of blocks. Without the NameNode, the file system can not be used. It is important to make the NameNode resilient to failure, so Hadoop provides two mechanisms for this. The first way is to back up the files that make up the persistent state of the file system metadata. Hadoop can be configured so that the NameNode writes its persistent state of multiple file systems. It is also possible to run a secondary NameNode, despite its name is not a NameNode. Its main role is to periodically merge the namespace image with the audit log to prevent the audit log from becoming too large [4].

The basic unit of storage in HDFS is a data block that generally is 64M, which is the same size with the partition in the MapReduce programming model. These blocks are preserved in memory. The HDFS file system uses a replication strategy to achieve high reliability. The number of replications is 3, which means that the same time each block will have 3 copies. The replications are stored in 3 DataNodes, each in different rack. The metadata of these blocks is registered in the NameNode. When a DataNode has something wrong, the data of the block can still be read from other DateNodes.

Data reading and storage mechanism in the HDFS distributed file system are different from the general file system. When users need to read a file in the file system, users should submit "read requests" to the NameNode, then users get the metadata after NameNode query metadata tables, finally connection is broken by NameNode. Next, users directly access DataNodes to obtain the required blocks and get the entire files. When users need to save the files, users also submit "write requests" to NameNode. A Namenode writes the file name in the namespace, then the Namenode splits the files into many fragments depending on the size of the file and query the metadata table for the distribution of free block files. After returning to the user data, the connection is broken. Next, users get access to the DataNodes and write data to the blocks. The architecture of distributed file system as shown in Figure 2.

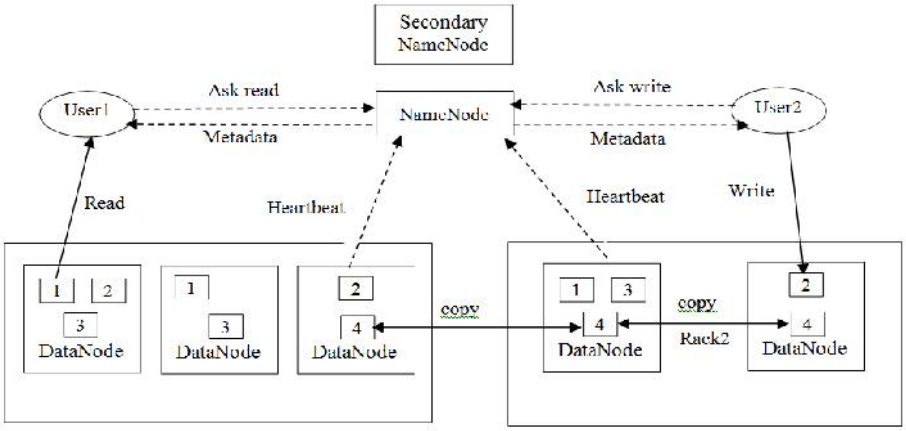

Figure 2. Distributed File System

4. Key Technologies of Search Engine

4.1. Search Engine Architecture

As one of the most technically application in Internet, in order to get access to mass data, respond users quickly and accurately, search engines need complex architecture and algorithms. Search engines get entire Internet information through the web crawler. The module of removing duplicated web pages (RDWP), which filters the web pages downloaded by the crawler module and gets rid of the duplicated pages. After this, the search engine can parse the web pages, extract the main content of web pages and links to other pages. In order to get a quick response, web pages content and links are stored by "inverted index", an efficient query data structure. Saving the links to other pages is important, because this link is available in the web pages relevance ranking stage. Through the "link analysis", we can determine the relative importance of pages, which is helpful for users with accurate search results [5-8].

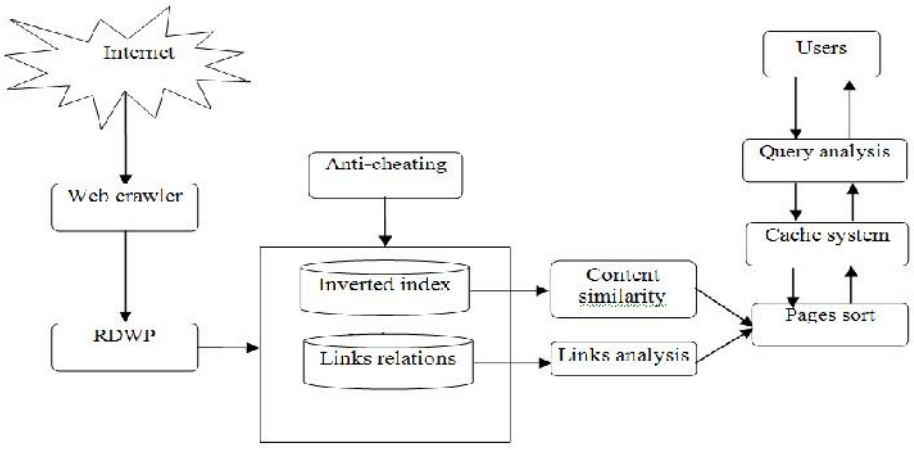

Figure 3. Architecture of Search Engine

Because the number of web pages is too large, the search engine not only needs to save web pages original information, but also the middle results. Using a single or a few machines is obviously unrealistic. Distributed search engines emerge as the times required. Google and other commercial search engines developed a set of cloud storage and cloud computing platform, that is Hadoop platform. Hadoop platform is composed with tens of thousands of ordinary PC, building a reliable storage and computing architecture of a massive information system to support the search engine. Technical architecture of a search engine as shown in Figure 3.

4.2. Distributed Search Engine

Distributed search engine can be divided into three subsystems: distributed crawler subsystem, distributed indexing subsystem and distributed retrieval subsystem. All of three modules use the MapReduce programming model of Hadoop, running in the distributed system environment. The design of each module adopts the object oriented model and uses the same distributed file system, to ensure data consistency. When the crawler module to download web pages, it has defined a series of MapReduce task to download information sources, to analysis pages, to extract URL links, to compute reverse links and PageRank. The index module uses the analytical package to make downloaded contents into a text document, and uses the word segmentation function segments to analyze grabbed contents, to extract the word sequence, to generate the inverted index. In constructing of the index, index module calls core class named Lucene to generate the index file, and saves it in a distributed file system. When the retrieval module provides data for users, the module needs to extract the word submitted by users from service pages to define the MapReduce tasks. The MapReduce tasks drive index module to retrieve data in the index library then index module will get the results and sorts, finally MapReduce tasks will present results to users.

5. Summary

In recent years, the research on distributed search engine has become more and more popular. It includes distributed computing, full text retrieval, Chinese word segmentation, query optimization and a series of technologies. But the research for search engine has some shortcomings. As Internet entrance, the search engine is very important for guiding and shunting network traffic flow, even up to a decisive role. Therefore, a variety of methods of "cheating" gradually popular. Using various means improves the web page search rankings, which will seriously affect the users' search experience. Therefore, how to automatically discover the web pages of cheating and punish them, become a very important part of search engine. It is found that the JobTracker of Hadoop platform also lack good task partition and scheduling algorithm, sometimes there is some nodes overloading, while other nodes are idle. In the future by introducing a more intelligent dynamic load balancing mechanism, adding the JobTracker dynamic task partition and scheduling algorithm, to make full use of the nodes. At the same time, improving Chinese word segmentation and pages scores strategies, we will get a better performance and higher accuracy of a distributed search engine. Anyway, distributed search engines greatly changed the way people access to information, the study of them or their applications will have profound significance.

References

[1] Wang Junsheng, Shi Yunmei, Zhang Yangsen. Key technologies of distributed search engine based on Hadoop. Journal of Beijing Information Science and Technology University. 2011; 26(4): 53-57.
[2] Wu Baogui, Ding Zhenguo. Research of Distributed Search Engine Based on Map /Reduce. *New Technology of Library and Information Service*. 2007; (8): 52-55.
[3] Peng Fang, Huang Qingyun, Qian Zhaopeng. The Research on Hadoop and Cloud Computing-Based Mass Data Storage Model of Computation. *Applied Mechanics and Materials*. 2012: 2899-2902.
[4] Dong Shoubin, Zhao Tiezhu. Performance Analysis of Distributed File System for Search Engine. *Journal of South China University of Technology (Natural Science Edition)*. 2011; 39(4): 7-13.
[5] Zhang Junlin. This is Search Engine. Beijing: Publishing House of electronics industry, 2012.
[6] Tom White. Hadoop Definitive Guide. America O'Reilly Media. 2009: 1-73.
[7] Liu Gang, Hou Bin, Zhai Zhouwei. The platform of Hadoop open source cloud computing. Beijing: University of Posts and Telecommunication Press. 2011.
[8] Owen O'Malley. Programming with Hadoop's Map/Reduce. ApacheCon EU, 2008