ATLANTIS PRESS

Advances in Social Science, Education and Humanities Research, volume 99

3rd International Conference on Social Science and Higher Education (ICSSHE-17)

Design and Implementation of Digital Library Retrieval System Based on Hadoop

DayingWang
Library, Shaanxi Normal University
Xi'an, China
wdy_ying@snnu.edu.cn

***Abstract*—In order to obtain the information, the system has been designed based on Hadoop as our experimental platform. It adopts HDFS distributed storage system to improve reliability, fault tolerance and scalability. To reduce retrieval latency, our system implements the distributed computing framework MapReduce, where the Map function maps the data processing task to multiple nodes, and the Reduce function aggregates the processing result of each node into one node, To achieve high-performance retrieval, full text information retrieval framework Lucene has also been adopted. Lucene are able to build unified information resource index, and sort the retrieved data resources by relevance to ensure accurate retrieval. Moreover, to improve the user experience, our system provides friendly interfaces to the user query and display through JSP based web designing. When handling the amount of data is small, distributed multi-node run will have longer execution time rather than that of a single node. The experimental result s show that, our system is able to handle massive data and provide real-time and accurate results to help users make quick decision.**

Keywords—Hadoop, library retrieval distributed computing, Lucene

I. INTRODUCTION

As the scale of Internet develops explosively, how to search resources effectively has become a big challenge. How to search and apply the large-scale data is an issue that needs to be dealt with urgently. When tacking this problem, we are often faced with the following two challenges:

First, analyzing and processing mass data has a prominent problem, which is the time delay caused by the magnitude of computation. Distributed computing is an effective way to address this issue. This is achieved by assuming tasks by multiple nodes, resulting in time saving. However, some problems of multi-node computation still exist, such as the optimization of input files and the operations management of distributed tasks.

Second, users often don't know the modes and structures of databases, and are not familiar with the query languages of traditional databases. As a result, traditional query methods don't enable users to search needed information. Besides, mass data is stored in various databases. The query difficulty increases due to the diversity of data. Therefore, traditional query methods seriously impede user queries. Users don't know how to exploit beneficial information from mass data based on their own intentions.

Considering the above problems, we design and implement the digital library retrieval system based on Hadoop by referencing current mainstream retrieval systems. The core of Hadoop platform is HDFS distributed storage systems and MapReduce distributed computing structures[1-3]. HDFS distributes mass data to different servers for storage and management. MapReduce enables distributed computing, thus increasing retrieval efficiency and reducing retrieval delay[4]. It eliminates parallel applications and file transfers for developers, who can only focus on service logics. Furthermore, Lucene full text information retrieval system allows us to accomplish the generation of indices and information queries of keyword retrieval systems. It well adapts to the distributed environment and retrieves the indexed data. The query results are automatically displayed according to relevance, which improves the system usability.

II. OVERALL SYSTEM DESIGN

A. Framework and Structure

By the base and concept of cloud computing, digital library retrieval system based on Hadoop constructs the infrastructure and service of digital libraries. The steps mainly include the construction of library resources, service applications, servers offering computing, management platform, network storage infrastructure and so on. The system of digital library retrieval system based on Hadoop consists of the application level, platform level and infrastructure level, as displayed in Fig. 1.

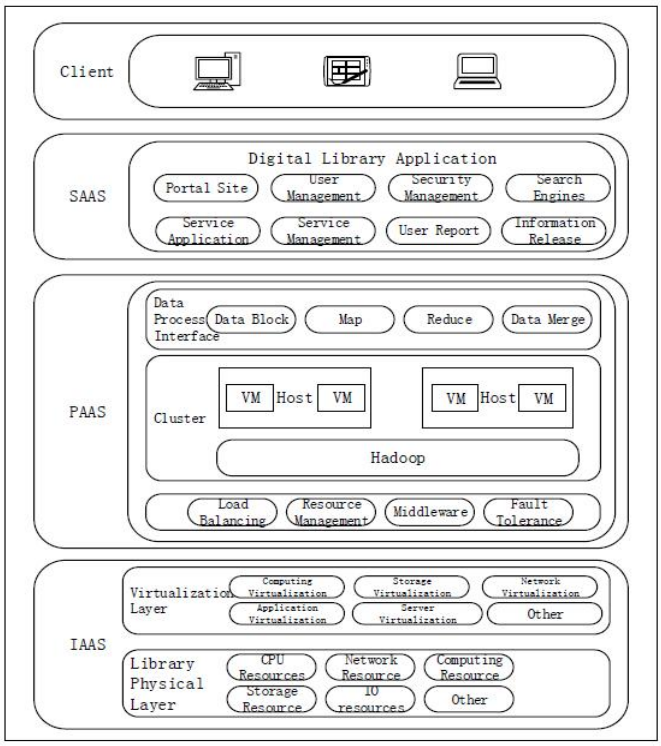

Fig. 1.Structure of System

1) Application level

The application level provides users with needed applications and user interactive interfaces, allowing developers to quickly formulate large-scale and user-oriented service applications. This level is also called Software as a Service, SaaS[5], which is designed to provide needed applications, including portals for digital library resources, search engines for convenient resource retrieval, and automatic integrated management systems. User management, security management, service application, service management, user reports and information release are offered.

2) Platform level

The platform level provides developers with development platforms, API interfaces and middleware. In this level, services regarding cloud computing are sealed, providing applications with resources needed for operation, management and maintenance. This level is also called Platform as a Service, PaaS. These services consist of distributed computing clusters, parallelapplication design and development environment, distributed file systems, mass data storage management system and other management tools. In the digital library resource retrieval system, five functions are provided. One is the information management interface, providing resource management, theme management, project management, etc. Another is the platform core service interface, providing full text retrieval, virtual resource integration, statistical analysis of customer behavior, etc. The third is the information release interface, mainly providing resource directions, information retrieval, theme release and portal tailoring. The final is the information processing interface, mainly providing content collection, editing, release and statistical analysis of information production.

3) Infrastructure level

The infrastructure level is the basis for the overall structure, lying at the bottom of the structure. It is the virtualized combination of hardware resources and related management functions, namely, Infrastructure as a Service, IaaS[6], which decides the service scope and capabilities of digital library resource retrieval systems. It consists of a virtualized level and a library-entity level. The virtualized level provides cluster computing of virtual servers, bottom digital library resource storage and dynamic resource pool integrated with networks. This is designed to improve the management of digital library resource storage and to share the stored resources, thus providing transparent usability and expandability. This level consists of computing virtualization, storage virtualization, server virtualization network virtualization, etc.

B. System Framework Design

The digital library retrieval system consists of three parts, that is, the visible interface in the front, the middle Lucene full text information retrieval framework and the bottom Hadoop distributed processing modules. The visible interface is the website of keyword retrieval system. The system allows users to input the retrieval information by the website interface and display the data in the website. In the Lucene full text information retrieval framework, information is indexed. Information queries are accomplished by the internal retrieval modules, and the results are output in sequence. Hadoop distributed processing modules have two links. The first is uploading the index files to HDFS, which distributes the files to various storage nodes. The second link is completed by MapReduce distributed computing framework, which seals Lucene query tasks and maps them into various nodes. Then, results are summarized by the reduce function.

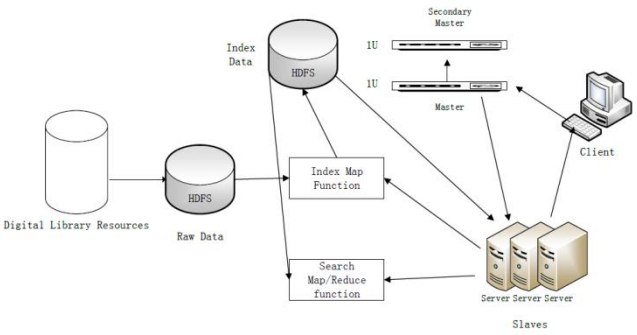

Fig. 2.System Function Design

As shown in Fig. 2, the design of the retrieval function in the system mainly includes building index files and achieving thesearch function. According to the ideas of distributed storage systems and distributed computing, digital library resources to be indexed need to be stored in the distributed file system HDFS, and the IndexMap function is designed. As the main node of clusters, Master's JobTracker is in charge of distributing IndexMap function to child nodes' Slave. So, every Slave's TaskTracker is in charge of operating IndexMap functions, which enables its own digital library resources to produce index files, and store these files in the distributed file system HDFS.

Index files are stored in the HDFS, and design Map/Reduce functions for retrieval. SearchMap functions are designed to conduct parallel search for index files, producing <key/value> pairs. SearchReduce functions are designed to order all the key-value pairs based on the values and obtain the results.

When clients apply for retrieval, the main node receives the request and Master's JobTracker distributes SearchMap functions to all the Slaves. Every Slave's TaskTracker in the system executes the SearchMap function, and sends the produced <key/value> pairs to the SearchReduce function operated by Slave. Then, the retrieval results are put in a parallel order. Finally, the ultimate results are obtained.

C. *System Realization*

1) Realization Environment

Hadoop platform should be established and use two Lenovo servers with VMware by a completely distributed mode. The virtual machine operates in win7 system, with one machine operating two systems. The names of nodes are respectively named masters, slave1 and slave2. As the main node, Masters operates on a Lenovo server, and is in charge of task assignment and state monitoring. As child nodes, slave1 and slvae2 operate on another Lenovo server, and are in charge of task execution and state reporting. Masters control the start and shutdown of the two slave nodes. In the meantime, the masters node is set as NameNode, and slave1 and slave2 are set as DataNode, constructing HDFS distributed file systems.

The visible interface in the integration window between users and systems, using JSP+Tomcat+MySQL mode. The website adopts JSP+servlet technology, producing the query and display interface. Taking information confidentiality into account, systems are designed to conduct retrieval after user login. User registering, login and admin interfaces are also added. MySQL databases are used to save personal data. The installation is based on Java's Tomcat servers, operating the JSP page and servlet program.

The process of Lucene building indexes mainly includes the pretreatment of data, distribution of words, producing indexes and index storage. In order to accomplish the establishment of indexes, Lucene needs to use plug-ins to analyze the content of texts from different formats of information resources, and construct the corresponding Document cases. The content of texts from the cases is divided and used to construct Field cases. Field cases describe the attributes of the document, such as the topic, author, abstract, body, key word, etc. As the retrieval results, these attributes are shown to the user. After the Field is established, distribution of words needs to be conducted. Finally, Index Writer object's addDocument method is used to add data to the index document.

When users conduct the retrieval, keywords they enter are received from the website, and the query task is formed by the keywords. Then, the task is sealed and transferred to each node by MapReduce distributed computing framework. Meanwhile, the main nodes divide the inverted index database stored in HDFS into pieces. Each task executes one piece, and gets the retrieval results of the piece. Finally, the reduce function summarizes the results and sends them to the website, which then shows them to the user.

2) Performance Testing

Keyword retrieval systems, based on Hadoop distributed processing platform, implement the high-quality search for information resources, fulfilling use needs. The system is designed to compute mass data by Hadoop platform's strong ability to conduct distributed processing, so that users can effectively retrieve what they want. In order to test the advantage that distributed computing has over single-node computing, we compare the performance of the three-node system and single-node system, as is depicted in Fig. 3.

We conduct an experiment to test the search speeds of the two. The size of retrieval data is treated as the independent variable, and the time function is added to the program. The time spent on retrieval is computed by the precise time before and after the retrieval. Then,the website displays it.

After several experiments, we get the comparison in Fig. 3. It shows that when we deal with less data, the execution time of multi-node is longer than single-node. The reason is that Hadoop takes some time to start up each node, and also spends some time in producing and transferring middle and final files. So, the execution time of multi-node is longer. However, as the scale of data grows, the advantage of Hadoop distributed processing becomes prominent. Therefore, this indicates that the processing of mass data by the Hadoop keyword retrieval system is comparatively effective.

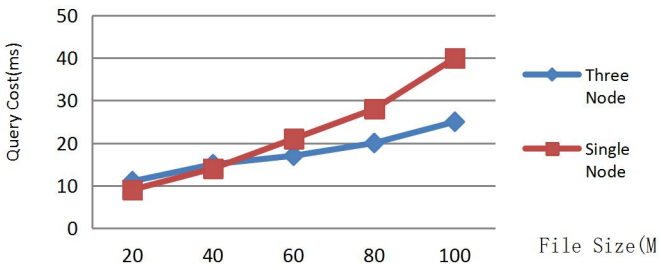

Fig. 3.Comparison between Retrieval Time

III. CONCLUSION

This study designs and implements the digital library retrieval system based on Hadoop. Mass retrieval of library resources can be conducted, overcoming the problem of the low efficiency of the standalone mode, which considerably increases the retrieval efficiency and is highly usable.

REFERENCES

[1] B.Hayes, "Cloud computing", Communications of the ACM, vol.51,2008, pp.9-11

[2] ChenKang, Zheng Weimin."Cloud Computing,"System Implementation and Current Research. Journal of Software, vol.20, 2009,pp.1338-1348.(In Chinese)

[3] F. PetrilloUmberto, R. Gianluca,C. Giuseppe,G. Raffaele, "FASTdoop: A Versatile and Efficient Library for the Input of FASTA and FASTQ Files for MapReduce Hadoop Bioinformatics Applications,"Bioinformatics, Oxford: England 16 January 2017

[4] Hao Shukui,"Brief Analysis of the Architecture of Hadoop HDFS and Map Reduce,"Designing Techniques of Posts and Telecommunications, vol.7, 2012, pp.33-42.

[5] Yao Xiaoxia, Zhao Yongchao, Chen Ling, Wang wenqing, "The Practice on SaaS-Based CALIS Service Sharing," Journal of Academic Libraries,vol.30,2012, pp.24-29.(In Chinese)

[6] Ren Siqi, Huang Guobin, Wang Fengxuan, Li Xiaojuan, "Analysis on the Patterns and Experience of Foreign Libraries Applying IaaS Cloud Computing Services," Library and Information Service, vol.59, 2015, pp.38-44.(In Chinese)