Research Distributed Search Engine Based on Hadoop

Rui GU
Suzhou Industrial Park Institute of Services outsourcing
suzhou, china
gur@siso.edu.cn

***Abstract*—In the age internet, processing massive data appears bottlenecks based on the Lucene. In order to improve the timeliness of the massive data retrieval, this paper focuses on the research of distributed search engine based on Hadoop.First,this paper introduces the Hadoop Distributed File System (HDFS), MapReduce parallel programming model, HBase database. Then through the MapReduce calculation model build index file and store it to cluster HBase. At last, the final experiment shows the advantages of distributed search engine based on Hadoop.**

Keywords-component; Hadoop; HBase; MapReduce; index file; Lucene

I. INTRODUCTION

The rapid development of the Internet makes the amount of information rapidly increasing,which makes the increase of network traffic and the bottleneck of traditional centralized search engine. Existing centralized search engine from such vast amounts of information quickly retrieve the information really need to take a very long time, the query is also becoming more and more difficult, so today's search engine system should have distributed processing ability, can constantly expand system according to the increase of information. The construction distributed search engine based on Hadoop is to solve the massive data search problem .

In the paper, we start from the characteristics of massive data, establish the index file through MapReduce distributed computing model and Lucene,and store the index file to HDFS and HBase, which can realize retrieval of keywords in the file , the number of content and keywords appear in a file. Make full use of distributed cluster topology feature based on Hadoop, realized the distributed processing of search engine, which has high reliability and easy extensibility.

II. HADOOP OPEN DISTRIBUTED PROCESSING PLATFORM

A. Hadoop Distributed File System

Hadoop is open source distributed computing framework of Apache software foundation, which realizes parallel programming model and distributed file system, provides the underlying storage support for distributed computing .It has been applied in many large enterprises.

HDFS uses Master/Slave framework providing storages for distributed environment, which is managed by one node (NameNode) and N data section (DataNode). The NameNode manages file system namespace and access to the store files in the cluster, one NameNode and multiple DataNodes can be found in each cluster. The principle of the distributed file system is split a complete file into multiple Block, each Block is stored in the different DataNode daemon, and to perform the arduous work of distributed file system.

HDFS data block is read or written to the local file system of the actual file, when hope HDFS file for reading and writing, the file is segmented into multiple blocks ,the NameNode inform the client each block resides . The client directly communicate with DataNode daemon, process the data block corresponding to the local file. And then the DataNode can communicate with other DataNode, copy the data block to achieve redundancy. The architecture of distributed file system as shown in figure 1.

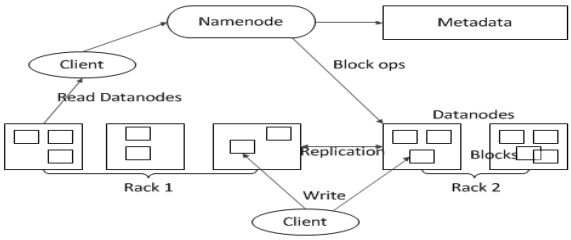

Figure 1 the architecture of distributed file system

The MapReduce engine also uses Master-Slave structure, which is composed of a JobTracker and a plurality of TaskTracker. JobTracker is the link between the application program and Hadoop, responsible for scheduling, monitoring MapReduce homework during the whole execution process. TaskTracker perform computational tasks, manage the implementation of each task in every child node. All of the TaskTracker is responsible for the implementation of a JobTracker assignment, can generate more than one JVM (Java virtual machine) to handle many tasks in parallel map or reduce.

B. Map/Reduce Models

MapReduce is a programming model proposed by google,which can achieve large-scale parallel computing and processing massive amounts of data on a large scale distributed system server. This model is mainly composed of the core operation Map and Reduce. the flow shown in Figure 2.

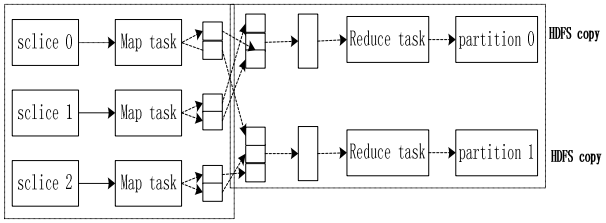

Figure 2. Map/Reduce model implementation process

978-1-4799-1843-0/15 $31.00 © 2015 IEEE
DOI 10.1109/ICNISC.2015.149

First the file data will be uploaded to the HDFS,then read the document file block (the default size of 64 MB) line by line and map the line number and content for the initial key value input, the Map function to separate the value part, to extract the keywords and the absolute file path , forming an intermediate key-value pair <key, value>.

Reduce function is scheduled by the Master host, which as a input for the corresponding processing, generated by the Map function key-value pairs, the same key will be merge to the result of the output result<key, value >. In the practical application, Map and Reduce can be free specified, that is brings great flexibility and the efficiency is very high. so it is very suitable for distributed search the large amount of data of simple data types.

C. HBase Data Store

HBase is the open source implemented by Google Bigtable, is a high reliability, high performance, columns and scalable distributed storage database system, using HBase technology set up large-scale structured cluster on cheap server.

HBase using a classic master/slave model. HBase cluster nodes is divided into HMaster and HRegionserver. As a primary cluster node, HMaster is responsible for maintaining various metadata information in HBase cluster. HRegionserver is responsible for reading and writing HBase data.

HBase can support millions rows of data tables, when HBase data capacity reaches a certain capacity threshold, its underlying data file will be split into multiple Region. A data sheet may consist of hundreds of different data.

HBase provides similar to relational database concepts of HTable, but unlike relational database based on the line for storage, HTable is based on the column for storage. The composition of HTable is divided into:

- Row Key: The Table's primary key, records in the table according to the Row Key.
- Timestamp: each data operations corresponding time stamp, can be seen as a version of data.
- Column Family: the table in a horizontal direction with one or more Column Family, Column Family can be any number of the Column. The Column Family support dynamic extensions without predefined the amount and type of the Column, all the Column are stored in a binary format, users need to type conversion.

III. DISTRIBUTED SEARCH ENGINE RESEARCH

A. Establish of the Index File

The paper use open source full-text retrieval development kit Lucene to realize the establishment of the index and query. With the increase of index file to a certain extent, search efficiency appears bottleneck. Through local large file and analysis word segmentation by Lucene, and eventually set up inverted index file by MapReduce. The specific procedures to realize part of the code is as follows:

```
String[] files = folder.list();
for (int i = 0; i < files.length; i++) {
File file = new File(folder, files[i]);
Document doc = getDocument(file);
writer.addDocument(doc); // Adding an index file
      }
```

The paper make use of distributed computing framework based on Hadoop, modified the Mapper, Reduce and TableReducer three classes. In the process of the whole Map Reduce, the map is the key method of input, need to Map the output of the keywords contained in the key value, so need to be isolated from the file content keywords, implements to < keyword, the absolute path to the file > as the output of the Map.

Finally, we need to overload reduce method, from the result output map separating the file name, then the name of the file containing the same key combination, to achieve the <keyword, file list> as reduce the output stored in HDFS and HBase, reduce the keyword as rowkey, file list as the value loaded into HBase. Part of the code is as follows:

```
job.setJarByClass(Jobmain.class);
job.setMapperClass(IndexMapper.class);
job.setCombinerClass(IndexCombiner.class);
job.setReducerClass(IndexReducer.class);
job.setOutputFormatClass(TableOutputFormat.class);//
Output to HBase
```

B. MapReduce Data Analysis

Through the Reduce function processing data set contains a lot of duplicate records, such as"hello f1:1","hello,f2:2","hello f3:4".

These three records represent user query three times, records include keywords corresponding file name and the number of keywords appearing in the file. However, when the statistical analysis can only be seen as a query, it need to remove duplicate data. MapReduce parallel computing model using data reduplications process shown in Figure 3:

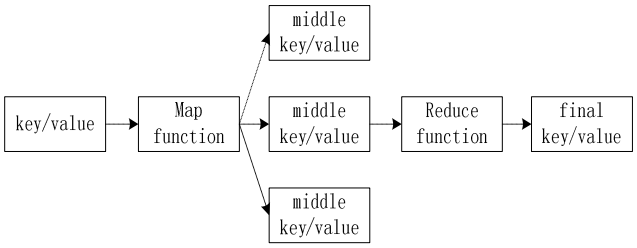

Figure 3 MapReduce data analysis model

(1) The large amount of file are directly uploaded to the HDFS after word segmentation technology by Lucene, and then read the document file block (the default size of 64 MB) line by line and map the line number and content for the initial input keys. Map function will be separated values partially processed to extract the keywords, absolute file path, forming an intermediate key-value pair.

(2) Reduce function reads the absolute path to isolate the file name as the value, and then automatically merged with the same key, with the keywords matching filename is encapsulated in the iterator, the iterator key use belong to the same group of all the key of the first key, after merger the same keywords, the result of Reduce are output to keys of<key, value>,and stored in HDFS.

(3) In the Reduce function, repeat the query words filter in order to complete the data duplicate removal treatment. Finally indexed file form from MapReduce will be stored in HBase.

functions as the initial input data for the next operation.

(4) The index files are stored in HBase established by MapReduce. The Row Key storage is the query keywords, value storage is the file name, keyword in the document frequency statistics and the contents of the file attribute.

(5) By keywords query in the HBase, which can be get very efficient query results.

IV. DISTRIBUTED SIMULATION RESULTS

This paper implemented in a distributed search engine is run on Hadoop cluster environment to complete operation and test. we ues 4 machine set up Hadoop cluster environment(CPU:AMD 3.10G;Memory:2G;Hardware:500G), one as a Master node is responsible for start the process of NameNode and jobTracker.The rest three are DataNode and TaskTracker process.

This experiment simulated multi-user concurrent query requests. For local 100M to the 10G files distributed search simulation experiments and compared with based on Lucene centralized search engine. The results are shown in the table below:

TABLE 1: THE SEARCH RESULTS

search	***index file size MB***	***centeralized query: MS***	***distribute query: MS***
first time	100	300	20
second time	521	400	110
third time	1024	500	180
fourh time	2048	640	340
fifth time	4096	760	400
sixth time	9510	3500	500

In a centralized search mode, when a single server processing search requests, index files size between 1G and 10G Lucene query time exponentially.

In the distributed search mode, although the advantage is not very obvious when a small amount of data stored in HBase, As with the continuous increase of the amount of data, the query is to focus the search very fast real-time, query time control in less than one second within 10G files.

results. the advantage of MapReduce is very efficient to query massive data, which can overcome search bottleneck of Lucene.

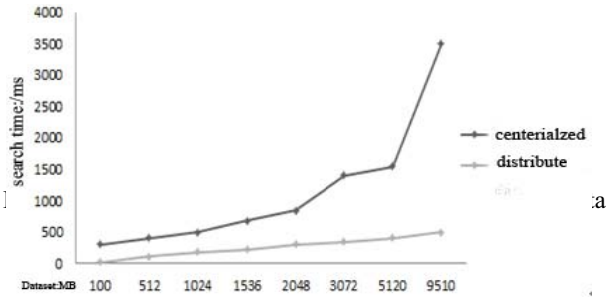

Figure 4. Contrast results data

V. CONCLUSION

This paper discusses the structure of the Hadoop, realized the basic function of distributed search engine based on Hadoop system, which can be realization parallel computing of large data sets. Establishment of index files and store it to HBase cluster by using MapReduce programming model, and then combined with the Lucene query returns the final result. By comparing the Lucene search engine and distrubute search engine based on Hadoop, the results show that the distributed search engine based on Hadoop in data processing aspects of the strong superiority.

REFERENCES

[1] Bigtable: A Distributed Storage System for Structured Data. FAY CHANG, DEAN, JEFFREY, ACM Transactions on Computer Systems; Jun2008, Vol. 26 Issue 2

[2] Google MapReduce, Google, Inc., 2008

[3] The Google File System, Ghemawat,Sanjay; Leung, Shun-Tak, Operating Systems Review, 2003

[4] Google's MapReduce programming model — Revisited, Ralf Lämmel, Science of Computer Programming, Volume 70, 2008