Design and Implementation of Intelligent search engine based on Hadoop Distributed Environment

Priyal Reja[1], Swati Bhat[2], Ankita Marathe[3]

[1] *Student, Information Technology, MET Bhujbal Knowledge City, Maharashtra, India*
[2] *Student, Information Technology, MET Bhujbal Knowledge City, Maharashtra, India*
[3] *Student, Information Technology, MET Bhujbal Knowledge City, Maharashtra, India*

ABSTRACT

Today Search Engine are becoming necessity of most of the people in day-to-day life for navigation on Internet or finding anything. Search Engine answers millions of Query everyday. In addition, it returns some spontaneous result across highly available data. A Vertical Search Engine based on Hadoop called HVSE was designed and developed which was based on principle of traditional search and topic oriented web crawling. To overcome this tradition and to make user's query search easier , an Instant autocomplete suggestion method is added to develop Intelligent Search Engine based on Hadoop known as HBISE, which works in distributed cluster environment using Lucene and mapreduce programming model to carry out data processing. In order to deal with massive data and retrieving results with high efficiency and accuracy.

Keyword: *- Disteclat, Supervised Feature Selection, Hadoop, Mapreduc; etc….*

1. INTRODUCTION

Search Engines essential act as filters for wealth of information the Internet. The allow user to quickly and easily find information that is of genuine interest or valuable to them, without the need of wade through numerous irrelevant webpages. Traditional centralize search engine deploys all the module function of system on central server which results in high server load , also effects the working efficiency therefore it doesn't meets public demand at the edge of big data. Big data consist of main three challenges based on three V's Velocity, Variety and Volume. Drawback of the centralized search engine was solved by the vertical search engine based on Hadoop called as Hadoop Vertical Search Engine (HVSE)[1]. HVSE performs well and having higher efficiency while dealing with massive data but was topic oriented.

Hence, we will be proposing Hadoop Based Intelligent Search Engine (HBISE) works in distributed cluster environment with the use of Lucene, MapReduce and other technologies with an intelligent recommendation system. This recommendation system will be called instant autocomplete suggestion which will result in decreasing user's effort. Our proposed HBISE is more flexible than centralized search engine and HVSE. It performs better and higher efficiency which helps user in better retrieval of results. Our proposed search engine is designed to scale well and to keep with the growth of web[4]. It gives exactly what we want, even will recommend we will be wanting for. It ensures fast response time and efficient access.

2. RELATED TECHNOLOGIES

Search Engines are websites whose aim is to provide the result of the queries provided by the user. The major methods on which search engine works

2.1 The Principle of Search engine

Crawling is the process by which the search engines discovers the update content on the web, such as new sites and pages, changes to existing sites, and dead links. This is an Internet bot which browses the information for the purpose of Web Indexing systematically. Crawlers consume resources on the system they visit. They validate Hyperlinks and HTML code. Indexing connects , parses and stores data to facilitate fast and accurate information. The purpose of storing an index is to optimize speed and performance in finding relevant document for search query[4]. The Searcher completes the task of QueryParser parses the query, then the search program retrieves the relevant results and sorts them in the index database.

2.2 The Framework of Lucene

Lucene is a free end open source information retrieval software library. It is supported by Apache software foundation and is released under Apache software License. Lucene itself is an index and search library and does not contain crawling and HTML parsing functionality. The role of Lucene in Searching will be done as soon as the Query Statement is processed towards analyzer, the searcher retrieves the results from the database resulting in Top documents[2].

The next job of Lucene is of the Indexing function. The Document is analysed for the Token generation purpose, which will be useful in IndexWriter and Indexing the database. Suppose a document is a record submitted to Lucene Indexing before it gets stored into the index it should analyzed in the form of XML or pure document or consist of garbage etc. in order to perform preliminary operation on document. After the process of analysis, the documents needs to be divided into tokens mostly called as Term Vector or Token comes under the process of tokenization. Each Term Vector will consist of its position , offset , length. Whenever you search for the token for specific word, the first lookup occurs into the term vector to locate the Term or the document where it can be found in the index.

Fig -1: Indexing.

Once you have indexed data in index storage , so now you have multiple ways to retrieve data for example by using a Lucene API or QueryParser, you can use expression language to query the query parser. Once the QueryParser scan the expression with the help of analyzer , it submit Query Object to index searcher which retrieve the results.

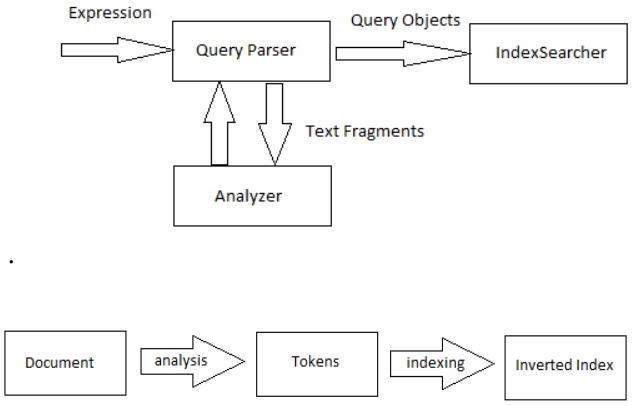

Fig -1: Searching

2.2 Hadoop

Hadoop is an open source framework of tools which supports running applications on Bigdata, Hadoop approach is to break data into small pieces so able to deal with BigData under distributed environment. The two main component of Hadoop which can be included in architecture the distributed storage system called Hadoop Distributed File System and high performance data processing called MapReduce.

Hadoop works on Linux based machine of low cost computers on the concept of master and slave architecture. The most important of characteristic of Hadoop is it is highly scalable, can consist of one computer or thousands of computers. Hadoop uses <key, value >pair which is fexible enough to work.

2.2.1 HDFS

HDFS is a file system designed for processing a BigData. The technique used by HDFS is the BigData to which the search engine deals with is divided into will be broken down into smaller equal pieces and then will be send to number of different computers. There exist a Masternode which will keep information of which data reside on which computer and the application can connect to master node to know where the data is residing. It consist of an unique Namenode and many Datanode. The Namenode is responsible to keep the track and details of which data is residing on which Data node, so when the application contacts to the Name node it tells the application go to this computer data to get the text. The application is not dependent on name node once it knows the data is at a particular Data node it directly to that node and collect the data.

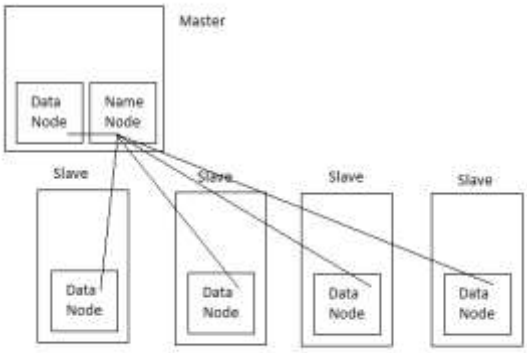

Fig -1: Hadoop HDFS

If in the case of hardware failure, Hadoop keeps hardware failure in mind. By default, it maintains three copies of each file which are scattered along each computers. This is the very important feature of Fault Tolerance.

2.2.2 MapReduce

MapReduce is a technique which divides a data and then rearrange the data instead of performing one big computation on one big data smaller computation perform on each smaller data then the result will gathered and aggregated and this will be the answer to the query. It also results in faster performance, MapReduce includes an unique Jobtracker and many Tasktracker. Jobtracker is responsible for scheduling the task to the task trackers whereas the Tasktracker manages the execution of particular task[3]. The Masternode then collects the answers to all the sub-problems and combines them in some way to form the output. Fault tolerance is not limited to the disk failure but it is also applicable to failure of Tasktracker services as well.

.

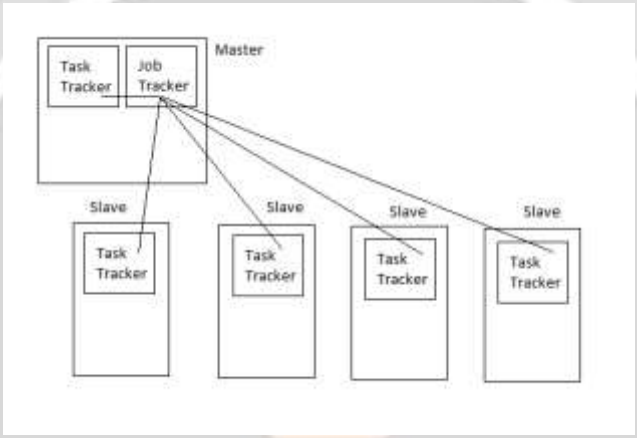

Fig -1: Hadoop MapReduce

3. PROPOSED SYSTEM

”Hadoop Based Intelligent Search Engine” will give the user immense power of searching and retrieving the exact results.The HBISE will be the approach of intelligently searching the query by relevantly matching the user’s query. There will be three PC nodes among which one will be the Masternode and the other two will be the Slavenode. The user will be able to enter the query through the Slavenodes.

3.1 Crawling Module

The query will be entered in the String format by the user on the Masternode and the Tokenization will be performed and distributed computing will come into occurance. For the Query entered, the Crawling or spidering based concept which means the retrieval of up-to-date resultant data is been provided to the Indexer[2]. The String query will be searched by the Crawler in the file contents on the Slavenodes. As it will follow the <key, value> pair, the mapper will set as <String filename, String file_content> which will be reduced to <url, text>. The Crawler’s result, will be carry forwarded for the indexing preference[1].

3.2 Indexing Module

Indexing gives the structuring technique to retrieve the results from the file based on some attributes on which indexing will be done. Now the Lucene job will be to convert the <url ,text> into <keyword, url> format[2]. The keywords will be containing the token and the url will include the links containing the keywords[1].

3.3 Instant Autocomplete Suggestion Module

The further process will be to mine the frequent urlList from the transactional database. A subset of a frequent urlList must also be a frequent urlList.

Disteclat Algorithm

The Disteclat algorithm will determine the frequent urlList that can be used to determine association rules which highlight general trends in the database. Therefore the Disteclat algorithm will reduce the number of candidates being considered by only exploring the urlList whose support count is greater than the minimum support count. This algorithm will help the user to get the help of keywords to be entered in the search statement which will act as a suggestion by using recently searched keywords and its respective url links.

So, Firstly, we will build a Candidate list of *k*-urlList. Secondly, we will extract a Frequent list of *k*- urlList using the support count, followed by third , we will use the Frequent List of *k*- urlList in determining the Candidate and Frequent List of *k+1*-itemsets. Lastly, We will repeat until we will have an empty Candidate or Frequent of *k*-urlList and then , we will return the list of *k-1*-itemsets.

4. ACKNOWLEDGEMENT

We take this opportunity to thank our Head of the Department Prof. Namita Kale and project guide Mr.Ratan Deokar for their valuable guidance and for providing all the necessary facilities, which were indispensable in the completion of this project report. We are so thankful to all the staff members of the Department of Information Technology of MET Bhujbal Knowledge City, Adgaon for their valuable time, support, comments, suggestions and persuasion. We would also like to thank the institute for providing the required facilities, Internet access and important books.

5. REFERENCES

[1]. Cheng Lin,Ma Yajie, "Design and Implementation of Vertical Search Engine Based on Hadoop", College of Information Science and Engineering, Wuhan University of Science and Technology, Wuhan, Hubei ,430081, China, 2015.

[2]. Zhou jingcai, Hu huaping, Yue hong, "Design and implementation of Lucene-based full-text retrival system", Computer Engineering and Science , 2015.

[3]. Zhou jingcai, Hu huaping, Yue hong, "Map Reduce Text Clustering Using Vector Space Model", Department of Computer Applications, Sri Padmavathi Mahila Visvavidyalayam, Tirupati ,September,2014.

[4]. Krishan Kant Lavania, Sapna Jain, Madhur Kumar Gupta, and Nicy Sharma "Google: A Case Study (Web Searching and Crawling)", International Journal of Computer Theory and Engineering, Vol. 5, No. 2, April 2013.

[5]. s Fan chenxi., " The Research and Application of Search Engine based on Hadoop" Zhejiang Sci-Tech University, 2013.

[6]. Xu jianying, " Study on Development Trend of Search Engine", Modern Information , 2011.

[7]. Wang jinsheng,Shi yunmei,Zhang yangshen, " Key technologies of distributed search engine based on Hadoop", Journal of Beijing Insititute of Machinery, 2011.

[8]. Wu wenzhong,Yi ping., "Applications of Distributed Search Engine Based On MapReduce", Computer System Applications,2012.