ISSN:0975-9646

Scaling Solr Performance Using Hadoop for Big Data

Tarun Patel[1], Dixa Patel[2], Ravina Patel[3], Siddharth Shah[4]

A D Patel Institute of Technology, Gujarat, India.

***Abstract*— E-Commerce websites generates huge churns of data due to large amount of transactions taking place every second and so their inventory should be updated as per transactions very quickly to remain stable in these competitive markets. Searching in web log files has become one of the important tasks for E-commerce companies to find such an important file in big data. The objective is to customize the search process to meet the precise needs of users in big data. The primary focus of the paper is to search for appropriate file in big data and scale the performance of Solr using Hadoop MapReduce. Solr is used in distributed environment through Hadoop**.

I. INTRODUCTION

As we know, E-Commerce industry is growing rapidly. Data is increasing exponentially and tremendously all over the world. The data which is collected, analysed and stored from multiple data sources is the biggest challenge for most E-commerce industries and also a lot of data is waiting to be analysed. For example, a bank-log file is one type of data that has to be analysed to search the issue or the application that has caused the failure, predict how many transactions are occurred and how many transactions are failed. The numbers of logs generated by any banking application are huge in size and are continuous. Log file is useful to collect information like IP address of the computer making the request (i.e. the client), its location, the HTTP status code, browser of that system etc.

Apache Hadoop can work on commodity hardware due to which the overall storing of these logs become cheap, as they can remain in Hadoop storage for longer time. Apache Hadoop is an open source project created by Doug Cutting and developed by the Apache Software Foundation. Now, to increase the profits of Ecommerce industries, mining the web log file will always be helpful because by mining the web log file E-commerce companies can easily search and calculate transaction. The problem arises when any business workflow or transaction fails. With such complex system, it become a big task for system administrator and manager to find and understand the issue in application, coordinate issue with other application and keep monitoring the workflow. When the multiple applications are involved it becomes more difficult to manage log across the application. Log management is one of the standard problems in big data. Proper and efficient searching can play an effective role in improving log files management. Apache Hadoop and Apache Solr can completely provide distributed environment to manage the different logs of multiple applications. Apache Solr is fast and has efficient searching capabilities to provide different searching features such as advanced Full Text search, highlighting the text and showing matching results. It also provides a faceted search to drill down and filter results to providing better browsing experience. It is said that 90% of the world's data is generated in last two years alone only. Every day, 2.8 quintillion bytes of data get generated. Twitter generates around 13 TB of data every day. Few years ago, companies were generating data and all others were consuming data, but now model is changed as now all of us are generating data and all of us are consuming data [1], [10]. The various sources of data generation are social media, scientific instruments, mobile devices, sensor technology etc. This all data will constitute into a Big Data. In information technology, Big Data is a collection of data sets, so large and complex that it becomes difficult to process and store them using on-hand existing tools and technologies. The Data which is generated is divided into 3 types [3].

1. Structured Data: - The data which is in tabular form.
2. Unstructured Data: - The data which is not in organized form. Metadata, Twitter tweets, and other social media posts are good examples of unstructured data.
3. Semi-Structured Data: - It is a form of structured data but do not forms a formal structure of data model. Example: - xml files [6].
4.

II. PROPOSED SYSTEM

Traditional search architectures fetch the data, process it, and send it directly to the search engine in a serial process. This is not ideal and takes large amount of time and even not cost efficient. This often causes false assumptions or misses deeper issues, leading to extra work for the search team, longer implementation processes, and degraded search capabilities. So, we use Hadoop and Solr platform in our system. The dataset is added to HDFS. The dataset is then provided to Solr where indexing is done on the data. Solr provide the desired document as output related to given input query [7].

A. *Apache Hadoop*

In Hadoop cluster, some nodes act as slave nodes and one as a master node. The architecture is divided in two layers: MapReduce layer and Hadoop Distributed File System (HDFS) layer. HDFS is a java-based file distribution system which provides reliable and scalable data storage that is designed to span large clusters of servers. In cluster all node have Solr platform for searching data or file in big data. Hadoop MapReduce is a software framework using which we can easily write applications which process big amount of data in-parallel on large clusters of commodity hardware in a reliable and fault-tolerant manner. When we compare SQL DBMS and Hadoop MapReduce, it is suggested that Hadoop MapReduce performs better than SQL DBMS. The traditional data base management system cannot handle a large dataset so we need to have Big Data technologies like Hadoop Framework. For Big Data analysis, Hadoop MapReduce is used in various areas [15]. To analyse web log file Hadoop is a suitable platform as the size of the web log is swelling day by day. Hadoop platform allows us to use thousands of nodes to store large scale data and analyse it. Hadoop cluster consists of thousands of nodes which store multiple blocks of log files. Log files are divided into blocks and these blocks are evenly distributed over a hundreds of cluster by Hadoop. After that, these blocks are also replicated over the multiple nodes to achieve reliability and fault tolerance [2], [10], [14]. Name Node will keep track of how web log data is fragmented into file blocks and which nodes store these blocks. Replication of web log file will be stored by Data node. On each slave node, Task Tracker is responsible for the execution of individual tasks [4], [10].

B. *Apache Solr*

Solr is an open source Java-based search platform developed by the Apache Software Foundation. It is part of Apache Lucene project and uses the Lucene Index. It runs as a stand-alone server or as part of other application servers. Provides search features like Full-Text Searching, Hit Highlighting, Fact Search and Browse, Geospatial Searching. Solr achieve fast search responses because, instead of searching the text directly, it searches the index instead. This type of index is called an inverted index [7], [8], [9].

C. *System Architecture*

The figure [1] shows Solr and Hadoop configuration. Solr provide UI to the end user and to get any type of query and it will execute that query. It consists of component such as Hadoop Distributed File System (HDFS) which is a file system that provides high throughput access to application data.

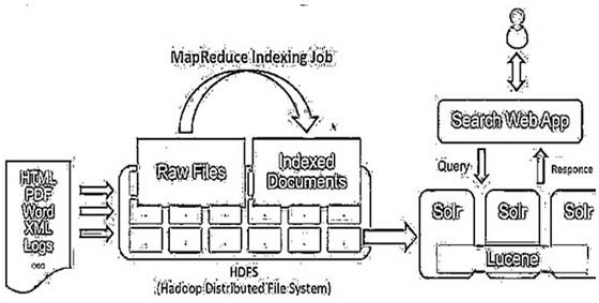


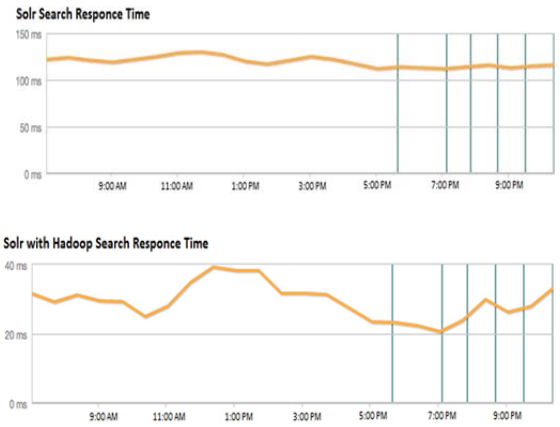
Figure [1]

III. IMPLEMENTATION

First of all, we created a Hadoop Cluster in Ubuntu operating system. We create a name node that is a master node and other data nodes. After creating cluster, add different files into HDFS. Then install Apache Solr into all the nodes of cluster. In Solr we will configure schema.xml first before adding any document. That schema file declares what kind of fields there are, which field should be used as unique key or primary key, how to index and search each field and which fields are required. Solrconfig.xml is the second file to configure. The elements of this file that are data directory location, cache parameter, request handlers and search components are used. Then we added the documents related to the schema file. The indexing of the documents is done by using Lucene indexing in the Solr. Then create a new user interface that is more user friendly for searching than provided by the Solr. The user now can enter the query for the desired document they want. The result is obtained in the form of the fields consisting key value pair.

IV. RESULT

Solr searches the index rather than searching full document content line wise making search process efficient. The Solr running in Hadoop (i.e. distributed environment) provide faster result than running sequentially in a single node. The following graph shows the same.



V. CONCLUSION

This paper thus provides insights to analyse and process various datasets. As the data is increasing day by day, to search a desired document from such a big data can be enhanced and made faster by implementing Solr in distributed environment through Hadoop.

REFERENCES

[1] What is big data: - IBM?

[2] "Why Big Data is a must in E-Commerce", Guest post by Jerry Jao, CEO of Retention Science. http://www.bigdatalandscape.com/news/whybig-data-is-a-must-in-ecommerce

[3] Tom White, (2009) "Hadoop: The Definitive Guide. O'Reilly", Scbastopol, California.

[4] Apache-Hadoop, http://Hadoop.apache.org

[5] L.K. Joshila Grace, V.Maheswari, Dhinaharan Nagamalai, "Analysis of Web Logs and Web Users in Web Mining", International Journal of Network Security & Its Applications (IJNSA), Vol.3, No.1, January 2011

[6] https://en.wikipedia.org/wiki/Semistructured_data

[7] "Searching and Indexing on Big Data" By Mayuri Bomewar, Snehal Bailmare, Mohinee Jadhao, Karishma Gaikwad. http://ijrise.org/asset/archive/16May4.pdf

[8] http://hortonworks.com/apache/solr/

[9] https://www.youtube.com/watch?v=7WIbU3ZlTe4&list=PL9ooVrP1hQOFFnF_1Cmi0t8aJLqMg0Wtx

[10] http://www.tutorialspoint.com/hadoop/hadoop_big_data_overview.htm

[11] https://cwiki.apache.org/confluence/display/solr/Running+Solr+on+HDFS

[12] http://www.slideshare.net/cloudera/solrhadoopbigdatasearch

[13] https://dzone.com/articles/solr-hadoop-big-data-love

[14] http://hortonworks.com/hadoop-tutorial/searching-data-solr/

[15] "Scaling Big Data with Hadoop and Solr", Hrishkesh Karambelkar, 2013

[16] Liang Yan *, Shuang Guang Liu, Ding Xiong Lao, "Solr Index Optimization Based on MapReduce", Applied Mechanics and Materials, Vols. 556-562, pp. 3506-3509, 2014