Distributed Document Clustering Using Parallel Computing Framework of Hadoop

Mr. Vitthal Kumbhar, Prof. Shyamrao Gumaste
vitthalkum@gmail.com, svgumaste@gmail.com

Abstract— Every day internet user's accesses data from various sources which in the form of text, images, audios and videos. This extraction of the data not limited to these terms, but it expands among vast area of searching things. But to give better services to user, data provider organization are searching technology which mainly focuses on challenging issues like accessing, storing, searching, sharing, transfer and visual presentation of data. Managing distributed unstructured data is impossible with traditional relational database system. Proposed system manages big data which is in the form of text, distributed among different text or pdf document. Paper focused on use of MapReduce framework as a parallel computing system of Hadoop. System proposes implementation of TF-IDF factor, k-means clustering on Hadoop. Also system proposes hierarchical clustering of documents. System reduces computing time to cluster data using Hadoop as compare to computing system implemented by using simple Java.

Index Terms— Hadoop, K-means clustering, MapReduce, Text mining.

INTRODUCTION

Data being mined in today's scenario is majorly from the internet through different devices such as mobiles, desktop computers, laptops etc. Search engine like Goggle, Yahoo, Twitter, facebook produces huge amount of data every day. To extract useful pattern for text mining is not easy with traditional database management system. There is need to form new application which reduces time to access data, avoid loss of data, provide high security to data and also perform inter machine communication. Also there is need to better understand how organizations view big data and to what limit they are currently using it to make beneficial to their businesses [1]-[6], [12]. Hadoop is new solution for over many of problems to handle big data. Google uses MapReduce parallelism of Hadoop and runs 1000 MapReduce jobs per day [13].

The large amount of information stored in unstructured form cannot simply be used for further processing by computers. For example information stored in unstructured text format is handled by computer as simple sequences of character strings. The presenting system mines big data which is stored in the form of text by applying k-means clustering with use of parallel computing framework of Hadoop.

Xindong Wu, Xingquan Zhu, Gong-Qing Wu, and Wei Ding [1] proposed Big Data Characteristics with HACE theorem. The characteristics defined by HACE theorem are in terms of data sources. These are as follow:

Heterogeneous and diverse sources: - Heterogeneous and diverse data sources generates huge amount of data. This is because different data sources have their different protocols for collection and storing of data.

Autonomous sources: - Autonomous data sources generate and collect information without any centralized control. For example, different web server provides a certain amount of information and each server works without depending on other servers

Complex and evolving relationship: - Multi-structure and multi-source data is complex data. Examples of complex data types are bills of materials, word processing documents, maps, time-series, images and video

To keep in mind challenging issues like heterogeneous, diverse and autonomous data sources, system uses MapReduce as a parallel computing framework of Hadoop to cluster data. Hadoop provides Map reduce parallel computing framework which clusters data parallely. In text and PDF document clustering, TF-IDF is important factor to calculate weight of each document. After calculating TF-IDF, data is clustered hierarchically by using K-means clustering algorithm [14], [15].

Paper composes with 6 sections. Introduction is given in First section. Second section focuses survey of related work. Third section elaborates system architecture along with TF-IDF and K-means algorithms. Section 4 gives mathematical modeling. Results shown in section 5 and paper conclude with conclusion in section 6.

RELATED WORK

Existing system mines data efficiently by using traditional relational database management system. Data used by existing application is in structured form. These applications are unable to handle data which is in unstructured form. The existing system uses methods for clustering are time consuming and also unable to perform inter-machine communication.

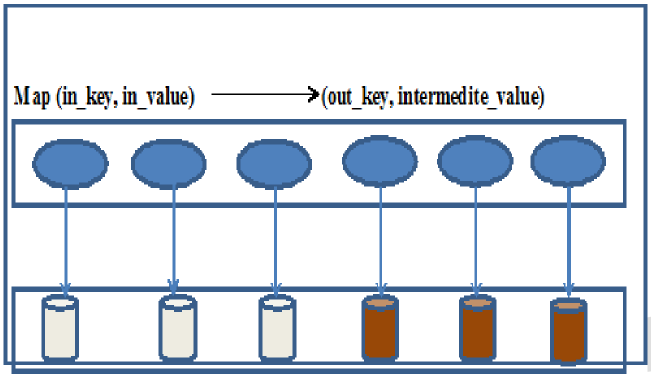

Fig.1 Map operation

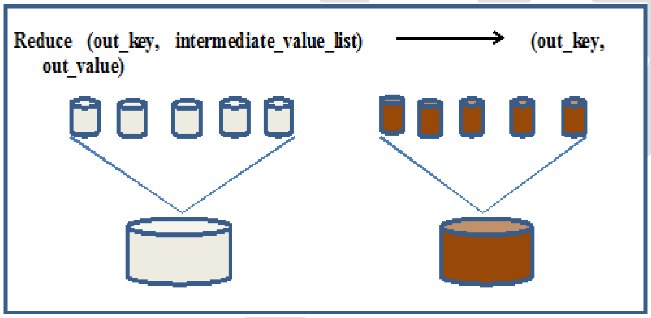

Fig.2 Reduce Operation

Emad A Mohammed, Behrouz H Far and Christopher Naugler [2], in their work they present MapReduce programming framework and its implementation platform Hadoop in clinical big data and related medical health informatics fields.

Jeffrey Dean, Sanjay Ghemawat [8], in their implementation, MapReduce runs on large cluster of commodity machines and is highly scalable. Work related to proposed system is to implement K-means and hierarchical clustering on Hadoop. As increase in nodes in Hadoop cluster reduces workload and time. Also Hadoop keep copies of data on each node. If any data on particular node goes fail, copy of that data would be available on another node.

MapReduce operation

As shown in fig1 and fig2, MapReduce operations perform with map and reduce function. In map function, master node takes input and divides into smaller sub problems. These sub problems are then distributed among worker node. Then worker node may again do same thing, leading to form multilevel tree structure. Worker node processes these sub problems and send the answer back to the master node [17].

In reduce function, master node collects the answers to all sub problems and combines them to desired output.

Hadoop Distributed File System

Apache Hadoop is an open-source software framework. Hadoop supports for data-intensive distributed applications and run applications on large clusters of commodity hardware. Hadoop was derived from Google's MapReduce and Google File System (GFS) papers [3].

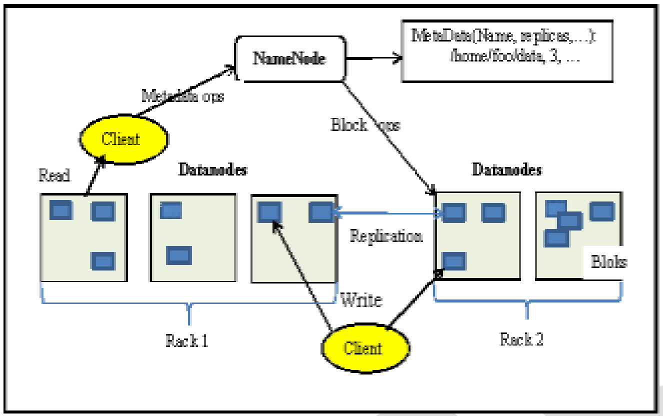

Fig.3 HDFS Architecture [10]

A small Hadoop cluster will include a single master and multiple worker nodes. The master node consists of a Job Tracker, TaskTracker, NameNode and DataNode. A slave or worker node acts as both a DataNode and Task Tracker, though it is possible to have data-only worker nodes and compute-only worker nodes. These are normally used only in nonstandard applications. To work with Hadoop, it requires Java Runtime Environment (JRE) 1.6 or higher. The standard start-up and shutdown scripts require Secure Shell (ssh) to be set up between nodes in the cluster [11].

The author Tian Xia [7], present ”An Improvement to TF-IDF: Term Distribution based Term Weight Algorithm”, Published by Journal of Software, Vol. 6, No. 3, March 2011, he present how to find Tf-Idf weight which is used in document clustering.

SYSTEM ARCHITECTURE

System proposes abstract level of Big Data to manage the Big Data stored in different location. This abstract model provides visual representation of data sources and creates fundamental data architecture so that more applications optimize data reuse and reduce computing costs. Figure 1 demonstrates idea of Big Data model. Big Data Model Architecture is represented through three layers. Physical layer represents sources of big data; these sources are in the form of structured or/and unstructured format. In next layer of model consist of management of physical data i.e. input data set, so that computing make easy in third layer. The last layer consists of computation in which data is retrieved for business value. By using these three layers it’s easy to retrieve information instead of accessing physical data.

A. Big Data model

Figure 4 shows three model layers; the physical layer, data modeling layer, computing layer. Physical layer indicates the data in a Big Data system. It contains different types of data such as audio, videos, business tables, emails, logs and so on. The data modeling layer also called as abstract data model manages physical data. The computing layer also called as application layer that retrieves information for business value. To separate physical data and data use, these three models

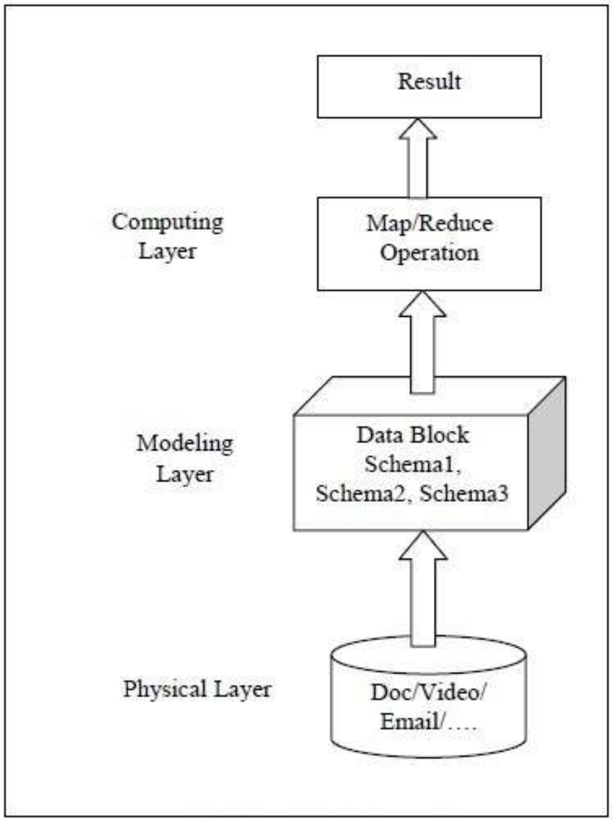

Fig.4 Big data model layers

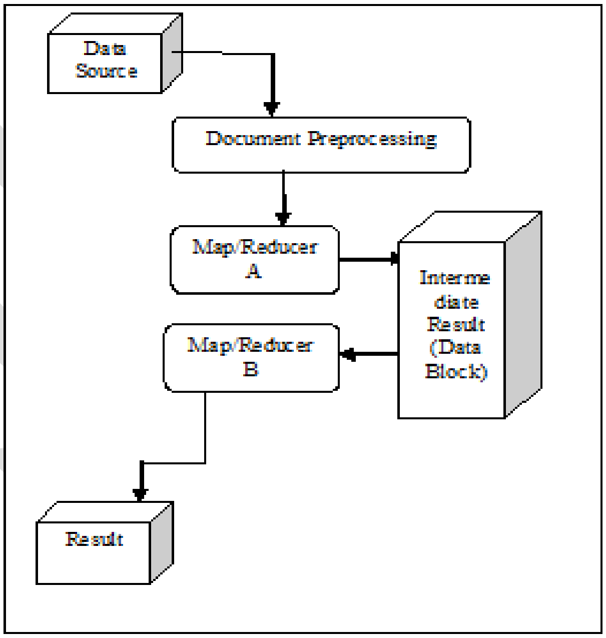

Fig.5 Proposed System Architecture

can be used to build data models. By using these three models application can access data through this data model without accessing the physical data. This data model makes flexibility in applications and data management.

As shown in fig 5, input data source consist of text documents or pdf documents undergoes document preprocessing before MapReduce operation. Document preprocessing in which documents are processed as follow:

1. Store stream of pdf document file in random access file create using COS Document constructor.
2. Parse the Pdf documents.
3. Extract the text from pdf documents and write it into newly created text file.
4. Tokenize each word from sentence and remove common words such as is, the, but, of etc.
5. Remove punctuations like comma, dot etc. from series of tokenized word.

TF-IDF Calculation

Term frequency-Inverse document frequency (TF-IDF) [7], [16] is calculated to find effect of terms that occurs frequently in corpus.

$$\text{TF-IDF}=\log\left(N(d) / F_t\right)$$

Where, TF-IDF =Inverse document frequency of term N(d)=Number of documents in corpus F_t=Frequency of a term t in corpus.

Cosine Similarity function

Cosine similarity function is used to find similarity between documents to form the cluster. The formula for finding cosine similarity is given by.

$$\text{Sim}(x,y)=\frac{\sum_{i=0}^{n} x_i y_i}{\sqrt{\sum_{t=0}^{n} x_t^2}\sqrt{\sum_{t=0}^{n} y_t^2}}$$

Where, x_i=is the TF-IDF weight of i^{th} term in first document

y_i=is the TF-IDF weight of i^{th} term in second document

K-means Algorithm for document clustering

Input: Set of pdf or text documents. K- no. of cluster,

I. Identify unique words present in the input dataset; here dataset is pdf or text document.
II. Generate input vector by calculating weight using TF-IDF
III. Generate similarity matrix by using cosine similarity.
IV. Specify no. of input cluster i.e. Value of k.
V. Select randomly k no. of documents
VI. Place one of k selected documents in each cluster based on similarity between documents and present document in the cluster.
VII. Compute centroids for each cluster.
VIII. Again apply similarity measures to find the similarity between the centroids and the input documents.
IX. Now place the documents in the clusters based on similarity between documents and the centroids of clusters.
X. After placing all the documents in the clusters, compare the precious iteration clusters with current iteration clusters.
XI. In all clusters if documents in current iteration similar to documents in previous iteration then stop the process, else repeat the steps through step 7.

.

Hierarchical document clustering Algorithm

Input: ‘n’ number of text or pdf documents.

1. Create ‘n’ no. of folders.
2. Put each doc in individual folder

```
   3. For every document in each folder do the following
       I. map the document using map function
      II. Calculate TF-IDF weight value
   4. Repeat the following till criterion function converges
   5. For every calculated document do the following
       I. Reduce the document using reduce function
      II. If Tf-Idf matches with other documents then merge folders
   6. Merge all disjoint folders in a root folder.
 Output: Hierarchical clustered documents.
```

The above algorithm is Hierarchical document clustering algorithm where the input is in the form of text or pdf documents. MapReduce function processes documents parallely to determine the TF-IDF values. The initial clusters are chosen as a value of K from the corpus and then each cluster is kept in a separate folder.

Mathematical Modeling

Term frequency-Inverse document frequency (TF-IDF) [7] is calculated to find effect of terms that occurs frequently in corpus.

$$\text{TF-IDF}=\log\left(N(d) / F_t\right)$$

Where, TF-IDF =Inverse document frequency of term N(d)=Number of documents in corpus Ft=Frequency of a term t in corpus.

Cosine Similarity function

Cosine similarity function is used to find similarity between documents to form the cluster. The formula for finding cosine similarity is given by.

$$\mathrm{Sim}(x,y)=\frac{\sum_{i=0}^{n} x_i y_i}{\sqrt{\sum_{i=0}^{n} x_i^2}\sqrt{\sum_{i=0}^{n} y_i^2}}$$

Where, x_i=is the TF-IDF weight of ith term in first document

y_i is the TF-IDF weight of ith term in second document.

Expected Result

Result of the proposed system is based on number of input documents and initial value of cluster.

Expected result will be in the form of TF-IDF values of each word of each file. Partition clusters as per given input as an initial value of K to the system and Hierarchical Clusters.

Conclusion And Future Scope

This paper present small use of MapReduce feature of Hadoop to work on big data. This algorithm requires initial values. It requires presumed set of cluster as an input. Future scope of this system is to take advantage of MapReduce infrastructure of Hadoop to parallel computing. Also manage failure of any node by creating copy on another node. Future scope also involves implementing new algorithms on Hadoop and measure the performance. The design and implementation of algorithm is main contribution to this work.

ACKNOWLEDGMENT

I express true sense of gratitude towards my project guide Prof. S. V. Gumaste, HOD and Associate Professor in Computer Department. I would also thanks to Prof. Sandip Kahate, PG-Coordinator, Computer Engineering department and Dr. Kharat Sir, Principal SPCOE, for their guidance and for allowing me to use the college resources and facilities.

REFERENCES:

[1] Xindong Wu, Fellow, IEEE, Xingquan Zhu, "Data Mining with Big Data", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 26, NO. 1, JANUARY 2014

[2] Emad A Mohammed, Behrouz H Far and Christopher Naugler, "Applications of the MapReduce programming framework to clinical big data analysis: current landscape and future trends", Mohammed et al. BioData Mining 2014.

[3] Ankit Darji, Dinesh Waghela," Parallel Power Iteration Clustering for Big Data using MapReduce in Hadoop", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 6, June 2014 ISSN: 2277 128X.

[4] Bharti Thakur, Manish Mann, "Data Mining for Big Data: A Review", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 5, May 2014

[5] Dave Chappelle, "Big Data & Analytics Reference Architecture", An Oracle White Paper, September 2013.

[6] "IBM What Is Big Data: Bring Big Data to the Enterprise," http://www-01.ibm.com/software/data/bigdata/, IBM, 2012.

[7] Tian Xia, "An Improvement to TF-IDF:Term Distribution based Term Weight Algorithm", Published by Journal of Software, Vol. 6, No. 3, March 2011.

[8] Jeffrey Dean, Sanjay Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters", Google, Inc 2004.

[9] Anand Rajaraman, Milliway Labs, Jeffrey D. Ullman, "Mining of Massive Datasets" Jure Leskovec Stanford Univ, Copyright © 2010, 2011, 2012, 2013, 2014

[10] Jinbao Zhu, Allen Wang, "Data Modeling for Big Data", CA Technologies Copyright © 2012 CA.

[11] http://hortonworks.com/hadoop/hdfs/

[12] http://en.wikipedia.org/wiki/Big_ data.

[13] http://erwin.com/expert_blogs/detail/opetarational_data_hadoop_and_new_modeling.

[14] http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html

[15] http://codingwiththomas.blogspot.in/2011/05/k-means-clustering-with-mapreduce.html?m=1.

[16] https://code.google.com/p/hadoop-clusternet/wiki/RunningMapReduceExampleTFIDF

[17] Neep Shaha, Dr. Sunita Mahajan, "Distributed Document Clustering Using K-means", IJARCSSE, Volume 4, Issue 11, November 2014