International Journal of Engineering Inventions
e-ISSN: 2278-7461, p-ISSN: 2319-6491
Volume 11, Issue 3 [May.-June. 2022] PP: 137-143

Information Retrival System for Document Clustering

S.Radhika[1],M.Sindhuja[2],M.Arjun[3], K.Govardhanareddy[4],K.Roshini[5]
Department Of Computer Science And Engineering, Raghu Institute Of Technology(Autonomous), Visakhapatnam, Andra Pradesh, India.

ABSTRACT: *Information retrieval (IR) is the field of computer science that deals with the processing of documents containing free text, so that they can be rapidly retrieved based on keywords specified in a user's query. The requirement for such Automated IR frameworks were in extraordinary interest to meet the business prerequisites according to the client's question. The cutting edge strategies suggested numerous IR frameworks utilizing different data mining calculations, novel map and reduce algorithms considering the default partitioner however none of them chipped away at the custom partitioners. The main objective of thisproposed work is to actualize a custom partitioner called Residue Modulo-K, where K speaks to the quantity ofreducers. The exploratory outcomes shows how the documents were clustered in the Hadoop environment, and tabulates the work carried out in the literature and compared with the proposed work using the necessary performance metrics.*

KEYWORDS: *Hadoop, Map Function, REduce Function, Clustering.*

Date of Submission: 27-05-2022 Date of Acceptance: 07-06-2022

I. INTRODUCTION

Big data is generally considered to be a problem and the problem arises essentially on the basis of two important factors, storage and processing [1]. Storage specifies the data volumes and the data variety where the data velocity is specified as processing.Traditional systems cannot manage unstructured data efficiently and one of the possible solutions is the Hadoop framework. The Hadoop framework is used to store and process large volumes of data (text data) as required and is a combination of two main HDFS and Map-reduce components[2]. HDFS includes a distributed file system and has an advantage over traditional file systems in terms of data loss, meta data and manual data splitting. Map-reduce is a programming model that performs distributed parallel processing between mappers and sends intermediate results to the reduction phase.

The Hadoop Distributed File System (HDFS) is used to store large volumes of data on the file system. The text file with a larger size is sent as an input to the Hadoop environment, the first data will be stored on HDFS and the automatic data split into a number of blocks (128mb per block size) will occur and the data will be stored in the respective data nodes. The meta-data for the entire text file will be stored on the name node, and in order to avoid data loss problems that occur frequently in traditional file systems, there is a secondary name node that is the name node backup that will improve the efficiency of HDFS in terms of data loss issues.

Map-reduces the distributed parallel processing model to include two main processing components: Job Tracker and Task Tracker [4]. Job tracker is used to assign job tasks to different task trackers where each task tracker is associated with each data node. The tasks assigned to the Job Tracker were read and write operations related to HDFS. The role of the task tracker is to perform the job assigned to the job tracker and send the acknowledgement back to the job tracker.

1.1 Problems of Big Data :

One of the biggest challenges in the real world was storing and processing the huge volumes of data engender from various sources such as Ecommerce, social networking, Surveillance cameras so on in day to day life. There is a need of processing and storing these huge volumes of data as per the requirements.

1. Dealing with data growth
2. Generating insights in a timely manner
3. Recruiting and retaining big data talent
4. Integrating disparate data sources
5. Validating data.

1.2 Solutions to Big Data:

1.2.1 Traditional Approach:

RDBMS was one of the approaches for storing and processing big data using Oracle, MS SQL server, DB2 and many more. But the problem is that only structured data is processed and stored in this approach A relational database management system (RDBMS) is a database management system (DBMS) that is based on the model invented by Edgar F.Codd, of IBM's San 45 Jose Research Laboratory. Most databases in widespread use are based on the database model. RDBMS have been a common choice for the storage of information in new databases used for financial records, manufacturing and logistical information, personnel data, and other applications since the 1980s. Relational databases have often replaced legacy hierarchical databases and network databases because they are easier to understand and use.

1.2.2 Google's Solution:

Google came up with a solution to this problem using "Map Reduce" algorithm. This algorithm divides the task into small parts and assigns it to clusters connected over the network and collects the final result. Map Reduce is a programming model and an associated implementation for processing and generating large data sets. Users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key. Many real world tasks are expressible in this model.

1.3 Document Clustering:

DOCUMENT clustering is a special version of data clustering problem. Data clustering is defined as an organization of a set of object into disjoined subsets called clusters . The assignment should be made in such a way that objects within a cluster are similar to each other. In text clustering instead of using some general objects we use documents. Clustering algorithms are often used in web search engines automatically group web pages into categories which can be browsed easily by a user. The main purpose of using document clustering is to find documents relevant to a given query. To obtain such results a query must be formulated first.

1.3.1 Introduction to Hadoop:

Apache Hadoop is an open source software framework for storage and large scale processing of data-sets on clusters of commodity hardware. Hadoop is an Apache top level project being built and used by a global community of contributors and users. It is licensed under the Apache License 2.0. Hadoop was created by Doug Cutting and Mike Cafarella in 2005. It was originally developed to support distribution for the Nutch search engine project. Doug, who was working at Yahoo! at the time and is now Chief Architect of Cloudera, named the project after his son's toy elephant. Cutting's son was 2 years old at the time and just beginning to talk. He called his beloved stuffed yellow elephant "Hadoop" (with the stress on the first syllable Hadoop is an Apache open source framework written in java that allows distributed processing of large datasets across clusters of computers using simple programming models. A Hadoop frame-worked application works in an environment that provides distributed storage and computation across clusters of computers. Hadoop is designed to scale up from single server to thousands of machines, each offering local computation and storage. The Apache Hadoop framework is composed of the following modules:

i. **Hadoop Common**: contains libraries and utilities needed by other Hadoop modules.
ii. **Hadoop Distributed File System (HDFS):** a distributed file-system that stores data on the commodity machines, providing very high aggregate bandwidth across the cluster.
iii. **Hadoop YARN**: A resource-management platform responsible for managing compute resources in clusters and using them for scheduling of users' applications.
iv. **Hadoop MapReduce**: A programming model for large scale data processing using three phases of mapper, reducer and shuffle phase.

1.3.2 Apache Hadoop Ecosystem:

All the modules in Hadoop are designed with a fundamental assumption that hardware failures (of individual machines, or racks of machines) are common and thus should be automatically handled in software by the framework. Apache Hadoop's MapReduce and HDFS components originally derived respectively from Google's MapReduce and Google File System (GFS) papers. Beyond HDFS, YARN and MapReduce, the entire Apache Hadoop "platform" is now commonly considered to 7consist of a number of related projects as well: Apache Pig, Apache Hive, Apache HBase, and others.

II. LITERATURE SURVEY

1.1 **EXISTING SYSTEM :** In Hadoop, Map Reduce is a computation that decomposes large manipulation jobs into individual tasks that can be executed in parallel cross a cluster of servers. The results of tasks can be joined together to compute final results. A word count is a numerical count of how many words a document contains. Most word processors today can count how many words are in a document for the user. The word count is the number of words in a document or passage of text. Word counting may be needed when a text is required to stay within certain numbers of words. This may particularly be the case in academia, legal proceedings, journalism and advertising. Word count is commonly used by translators to determine the price for the translation job. Word counts may also be used to calculate measures of readability and to measure typing and reading speeds (usually in words per minute). When converting character counts to words, a measure of 5 or 6 characters to a word is generally used for English. Word Count example reads text files and counts how often words occur. In computer programming there are many ways to get a word count of a variable or other text. Below is one example of how you could get a word count. The input is text files and the output is text files, each line of which contains a word and the count of how often it occured, separated by a tab.

2.1.1 Map Function – It takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (Key-Value pair).

2.1.2 Reduce Function – Takes the output from Map as an input and combines those data tuples into a smaller set of tuples.Disadvantages of Word Count problem

1. Word efficiency of Bigram is greater when compared to the word efficiency of the unigram.
2. Word efficiency is the state or quality of being efficient, or able to accomplish the retrieving of data with the least waste of time and effort;
3. It competency in performance.
4. It can be extremely time consuming.
5. It is subject to increased error, particularly when relational analysis is used to attain a higher level of interpretation
6. It is often devoid of theoretical base, or attempts too liberally to draw meaningful inferences about the relationships and impacts implied in a study
7. It is inherently reductive, particularly when dealing with complex texts
8. It tends too often to simply consist of word counts
9. It often disregards the context that produced the text, as well as the state of things after the text is produced
10. It can be difficult to automate.

2.2 PROPOSED SYSTEM:

Map-reduce is a functional programming model that performs two functions, namely *map and reduce functions. The key benefit we gain when using a map-reduce* architecture is that it is capable of processing greater amounts of text data with less time of execution as the number of mappers performs distributed parallel processing [11]. The ***Figure 2*** below shows the map-reduce architectural workflow.Residue Modulo-KMEMORY MANAGEMENT HADOOP DISTRIBUTED FILE SYSTEM (HDFS) .

III. ANALYSIS

3.1 Map Reduce Job Flow :

This post is to describe the map reduce job flow – behind the scenes, when a job is submit to hadoop through submit() or wait For Completion() method on Job object. This Map reduce job flow is explained with the help of Word Count map reduce program described in our previous post. submit() method submits the job to the hadoop and wait For Completion() method submits the job to hadoop only when it is not already submitted and it waits for the completion of submitted job. In YARN implementation, the run mode of map reduce job, can be set through map reduce.framework.name property in yarn-site.xml. The valid values are local, classic and yarn. local mode will submit the jobs to local job runner and classic model will submit the jobs through old Map reduce framework which is usually called as Mapreduce1. YARN mode will submit the jobs through new Map reduce framework. In this post, we will cover only YARN implementation specific flow.

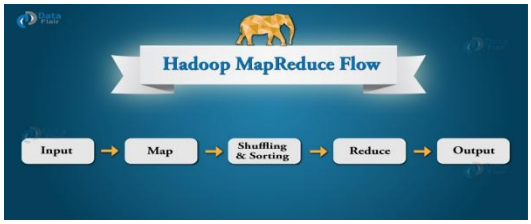


Figure 2 Map Reduce Job Flow

3.2 Hadoop API:

Hadoop ecosystem consists of four major components. They are

1. Hadoop common-contains set of predefined utilities and libraries that can be used by other modules within hadoop ecosystem
2. Hadoop Distributed File system-stores very large files running on cluster of commodity hardware. It creates several replicas of the data block to be distributed across different clusters for reliable and quick data access.
3. Hadoop YARN(Yet Another Resource Negotiator)-framework responsible for providing the computational resources (e.g., CPUs, memory, etc.) needed for application execution.

3.3 Architecture of HDFS:

It follows master-slave architecture where it consists of master node(Namenode) and multiple slave nodes(data nodes). The different components present in HDFS architecture are as follows:

a) Blocks
b) Namenode
c) Datanode
d) Secondary Namenode

3.4 Map Algorithm:

 Read the input text file
 Read the content character by character and line by line
Line value.toString ()
 Split every line into sequence of words
Words[ ] line.split (" ")
 Map <key, value> <-> <line offset, line content>
 Begin loop
String word Words []
End loop
 Assign count to every word of the line
 Output the respective word and count the every occurrence of word
Map <key, value> <-> <text, long writable>

3.5 Reduce Algorithm:

 Read the input from the shuffle phase in form of <key, list <values>>
47 Sum 0
 Iterate over the loop
Begin
Long writable value: values
Sum Sum + value. get ()
End loop
 Assign the value sum to every word that is obtained from the shuffle phase. Hadoop MapReduce processes a huge amount of data in parallel by dividing the job into a set of independent tasks (sub-job). In Hadoop, Map Reduce works by breaking the processing into phases: Map and Reduce Map Reduce is the data processing layer of Hadoop (other layers are HDFS – data processing layer, Yarn – resource management layer). Map Reduce is a programming paradigm designed for processing huge volumes of data in parallel by dividing the job (submitted work) into a set of independent tasks (sub-job). You just need to put the custom code (business logic) in the way map reduce works and rest things will be taken care by the engine. The data processed by Map Reduce should be stored in HDFS, which divides the data into blocks and store distributedly, for more details about HDFS follow this HDFS comprehensive tutorial.

IV. REQUIREMENTS

4.1 SOFTWARE REQUIREMENTS SPECIFICATION :

4.1.1 Hardware Requirements :

- Minimum **of 2**GB RAM.
- IDE Eclipse.
- HDFS(Hadoop Distributed File System).

4.1.2 Software Requirements :

- Ubuntu operating system.
- JAVA 6.
- Map Reduce framework.

- Core java and dependency jar files.

V. IMPLEMENTATION & RESULTS

Implementation of Bigram is done on IDE Eclipse environment by downloading core and dependency JAR files from Apache Hadoop Repository on Ubuntu Operating System using JAVA6 with a minimum of 2GB RAM..

5.1 IDE Eclipse:

Eclipse is an integrated development environment (IDE) used in computer programming, and is the most widely used Java IDE. It contains a base workspace and an extensible plug-in system for customizing the environment. Thus, every plug-in developed integrates with Eclipse in the same way as other plug-ins; in this respect, all features are "created equal". Eclipse provides plug-ins for a wide variety of features, some of which are from third parties using both free and commercial models. one can use the New Java Class wizard to create a Java class.

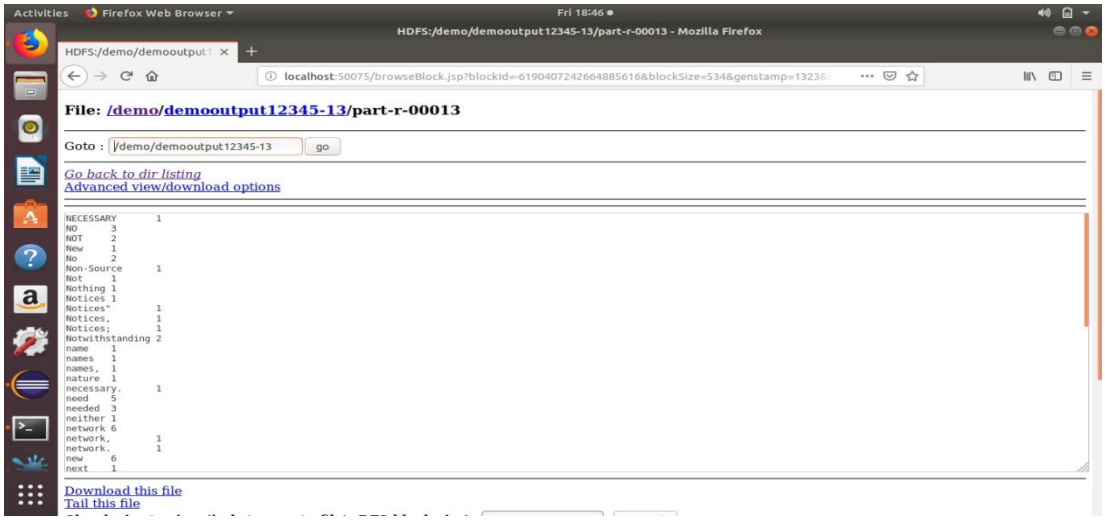
5.2 MapReduce programs work in two phases:

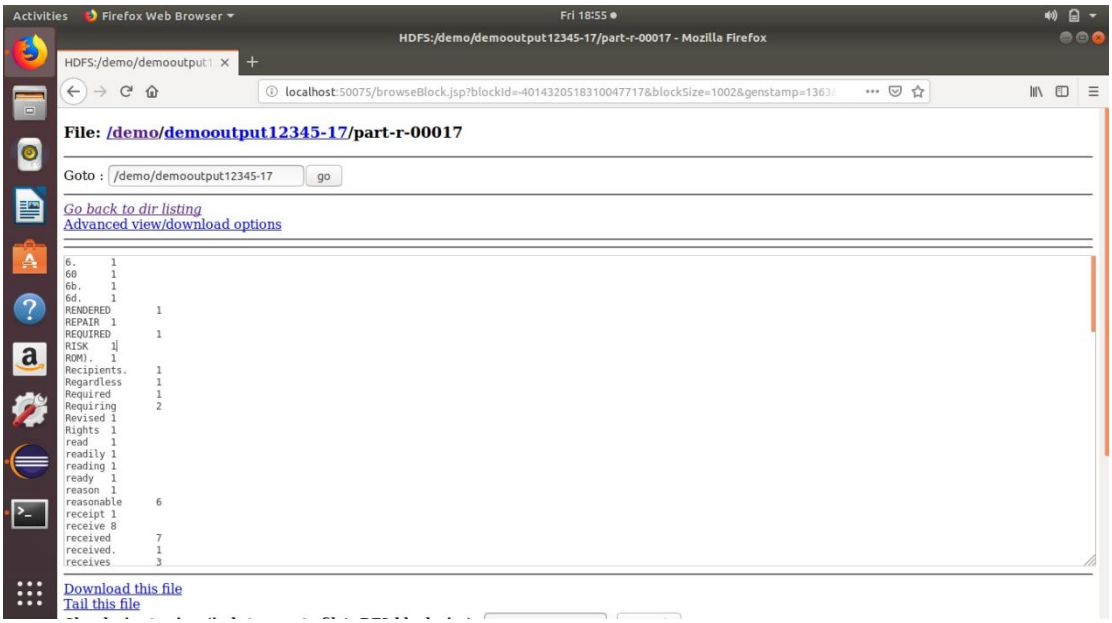
1. Map phase
2. Reduce phase.

Input to each phase are **key-value** pairs. In addition, every programmer needs to specify two functions: **map function** and **reduce function**. The whole process goes through three phase of execution namely, Input to a MapReduce job is divided into fixed-size pieces called **input splits** Input split is a chunk of the input that is consumed by a single map .

- **Mapping**- This is very first phase in the execution of map-reduce program. In this phase data in each split is passed to a mapping function to produce output values. In our example, job of mapping phase is to count number of occurrences of each word from input splits (more details about input-split is given below) and prepare a list in the form of <word, frequency>
- **Shuffling**- This phase consumes output of Mapping phase. Its task is to consolidate the relevant records from Mapping phase output. In our example, same words are clubed together along with their respective frequency.
- **Reducing**- In this phase, output values from Shuffling phase are aggregated. This phase combines values from Shuffling phase and returns a single output value. In short, this phase summarizes the complete dataset. All the logic between user's map function and user's reduce function is called shuffle. Shuffle then spans across both map and reduce. After user's map() function, the output is in-memory circular buffer. When the buffer is 80% full, the background thread starts to run. The background thread will output the buffer's content into a spill file. This spill file is partitioned by key. And within each partition, the key-value pairs are sorted by key.After sorting, if combiner function is enabled, then combiner function is called. All spill files will be merged into one MapOutputFile. And all Map tasks's MapOutputFile will be collected over network to Reduce task. Reduce task will do another sort. And then user's Reduce function will be called.

VI. SAMPLE OUTPUT





VII. CONCLUSION & FUTURE SCOPE

7.1 CONCLUSION:

The main goal of the IR system is to extract the required information from the broad text corpus according to the user's query. Many of the conventional IR systems proposed by different researchers used standard data mining algorithms and the necessary conclusions were drawn. Some of them have also been listed in the literature, but very few members have carried out their research on Big Data Systems, which provides the ability to process data on a large scale. Most of the work that is performed by the researchers on Big Data systems suggested a novel map and reduce algorithms, but only the default partitioner would be managed by the architecture itself. For our proposed work, along with a novel map and reducing algorithms, we used a custom partitioner named Residue Modulo K, where K represents the number of reducers that the design invokes. Performance indicators have been estimated for each and every clustered document and the maximum likelihood values have been achieved in terms of significance. Our acquired outcomes likewise depicts the cluster size and number of words available in every cluster after document clustering which makes us easy to analyse the performance metrics adequately which is no place referenced in any past exploration discoveries.

7.2 FUTURE SCOPE:

In addition to the custom partitioner, the use of combiners between the map and the shuffle step in the map decreases the time of execution significantly, and even the data size increases exponentially as the mappers operate in parallel as the combiners. By design, the architecture determines whether to use the combinations does not depend on the load of the input document. There is still room for implementing a custom combiner class that reduces the data transfer time and the computation time of the processed input file there by increasing the efficiency of the IR system.

REFERENCES

[1]. N. Elgendy and A. Elragal, "Big data analytics: A literature review paper," 2014, doi: 10.1007/978-3-319-08976-8_16.
[2]. D. P. and K. Ahmed, "A Survey on Big Data Analytics: Challenges, Open Research Issues and Tools," *Int. J. Adv. Comput. Sci. Appl.*, 2016, doi: 10.14569/ijacsa.2016.070267.
[3]. S. N. Khezr and N. J. Navimipour, "MapReduce and Its Applications, Challenges, and Architecture: a Comprehensive Review and Directions for Future Research," *Journal of Grid Computing*. 2017, doi: 10.1007/s10723-017-9408-0.
[4]. J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. ACM*, 2008, doi: 10.1145/1327452.1327492.
[5]. H. Tran and M. Shcherbakov, "Detection and prediction of users attitude based on real-time and batch sentiment analysis of facebook comments," 2016, doi: 10.1007/978-3-319-42345-6_24.
[6]. S. W. Kim and J. M. Gil, "Research paper classification systems based on TF-IDF and LDA schemes," *Human-centric Comput. Inf. Sci.*, 2019, doi: 10.1186/s13673- 019-0192-7.
[7]. M. Z. Hossain, M. N. Akhtar, R. B. Ahmad, and M. Rahman, "A dynamic K- means clustering for data mining," *Indones. J. Electr. Eng. Comput. Sci.*, 2019, doi: 10.11591/ijeecs.v13.i2.pp521-526.
[8]. S. Heidari, M. Alborzi, R. Radfar, M. A. Afsharkazemi, and A. Rajabzadeh Ghatari, "Big data clustering with varied density based on MapReduce," *J. Big Data*, 2019, doi: 10.1186/s40537-019-0236-x.
[9]. B. Ordin, D. S. Ballı, and N. U. Satı, "An incremental approach for solution of text clustering problem," *Int. J. Knowl. Eng. Data Min.*, vol. 6, no. 3, p. 273, 2019, doi: 10.1504/ijkedm.2019.102488.

[10]. Q. Li, S. Li, S. Zhang, J. Hu, and J. Hu, "A review of text corpus-based tourism big data mining," *Applied Sciences (Switzerland)*. 2019, doi: 10.3390/app9163300.
[11]. W. Du, D. Qian, M. Xie, and W. Chen, "Research and implementation of mapreduce programming oriented graphical mo.