Hadoop-MapReduce: A Platform for Mining Large Datasets

Maedeh Afzali
Manav Rachna International University, Faridabad
Email Id: Maedeh.af@gmail.com

Nishant Singh
Manav Rachna International University, Faridabad
Email Id: nishantsingh040@gmail.com

Suresh Kumar
Manav Rachna International University, Faridabad
Email Id: suresh.fet@mriu.edu.in

***Abstract* – In today's technical world, big data is playing an important role in our life. The various digital devices are generating a deluge of data, which are too large and complex. Unfortunately, the conventional data mining techniques don't have the capability to analysis and manage these enormous amounts of data. These upcoming challenges have encountered us to a need of a new platform that helps us to improve the process of analyzing and extracting the useful knowledge and insights from large data sets. The aim of paper is to describe the Hadoop-MapReduce framework. In addition, one of the famous data mining algorithms in the field of association rule mining, called Apriori algorithm is implemented in MapReduce programming language, and it is executed on Hadoop platform.**

Keywords – Association Rules, Apriori Algorithm, Apriori-MapReduce Algorithm*, *Big Data, Big Data Mining, Hadoop, MapReduce.

I. INTRODUCTION

Data mining is a step in the process of knowledge discovery for finding patterns of data by the application of data analysis and discovery algorithms [1]. Nowadays, technology revolution has facilitated people with a variety of connected digital components, which generate an enormous amount of data at an unprecedented scale. According to IDC Digital Universe study in 2011, 130 Exabytes of data were created and stored in 2005. In 2010, the amount grew to 1,277 Exabytes and it was anticipated that in 2015 these data will grew up to 7,910 Exabytes [2]. This ever increasing growth has witnessed us to challenges in exploring the large volumes of data and extracting useful knowledge from them. To overcome these challenges, the process of knowledge extraction should be efficient and close to real time. The big data concept provides us the capability to process, analyze, store, understand and extract useful knowledge from these oceans of data, which cannot be done through existing data mining tools. In addition, Hadoop framework is an effective platform for implementing the conventional data mining methods with the help of a simple programming model called MapReduce.

The remainder of this paper is organized as follows. In Section 2, some related work on big data mining and data mining techniques is discussed. Further, in section 3 the Hadoop framework and its main components are explained in detail. In section 4, some well-known data mining techniques are briefly reviewed. Finally, for an experiment the most important data mining algorithm in the field of association rules called Apriori algorithm is implemented in MapReduce language and it is executed within the Hadoop framework.

II. LITERATURE REVIEW

D. Che et. al [3] have described big data mining as a technique which not only fetch the requested information but also some hidden patterns and relationships between data. They believe that big data mining techniques can solve the issues and challenges which were faced by existing mining techniques and algorithms. Further, heterogeneity, volume, velocity, accuracy and trust are the most prominent issues and challenges in big data mining.

W. Fan and A. Bif [4], have described big data as a new term which is used to identify the large datasets. These datasets cannot be processed and managed with the existing data mining techniques and software tools. They have discussed that big data mining has the capability to extract useful information from the large datasets. They have discussed the existing controversy about big data. Finally, they have provided a brief introduction to big data tools such as Apache Hadoop, Apache S4, Mahout, R and Storm.

A. P. Deshmukh, et al [5], described Hadoop distributed file system. They have discussed that HDFS is suitable to hold very large amounts of data. Finally, the Hadoop architecture and its components such as, blocks, NameNode and DataNodes are described.

J. Woo [6], presented a MapReduce algorithm based on Apriori algorithm that is a popular algorithm to collect the itemsets that occurred frequently. They have implemented and executed the Apriori-MapReduce algorithm on Hadoop framework. By focusing on the time complexity, their results showed that the proposed algorithm provides high performance computing when the map and reduce nodes are added, as compared to the normal Apriori algorithm. Further, the produced itemsets by the algorithm can be adopted to compute and produce an association rule for market analysis.

Ning . Li, et al [7], implemented a parallel Apriori algorithm based on MapReduce in order to provide a fast and efficient algorithm that can handle large volumes of data, which is becoming a challenging issue nowadays. Their experimental

978-9-3805-4421-2/16/$31.00 ©2016 IEEE

results demonstrated that the algorithm can efficiently process large data sets on commodity hardware and it is scalable.

III. HADOOP-MAPREDUCE FRAMEWORK

Hadoop is a framework that provides a distributed file system and helps us to analysis and process large datasets, through MapReduce programming model [8]. The original Hadoop 1.0 consists of two main components called HDFS and MapReduce.

A. HDFS

HDFS is a distributed file system used to store files across a collection of servers in a Hadoop cluster. HDFS is implemented in Master/Slave architecture. Basically, there are two significant services running on an HDFS, named as NameNode and DataNodes. In every Hadoop cluster, there is a single NameNode, which runs on the master node. It maintains the file system namespace and keeps all the information about the files and directories. On the other hand, there are more than one DataNode available, usually one per node in a cluster. They are responsible to store and retrieve the application data when they are told by NameNode [9].

B. MapReduce

MapReduce is a parallel programming framework that provides a parallel and distributed platform in order to simplify the difficulties encountered while processing and analyzing large data sets. MapReduce processes the large amount of structured and unstructured data by using map and reduce functions.
As shown in figure 1, the client writes a Map function, which takes an input pair and produces a set of intermediate key/value pairs, and then the reduce function combines all intermediate values associated with the same intermediate key to produce the output [10,11]. Based on this definition the map and reduce functions are formalized as follows:

Map function: $map: (key1, value1) \Rightarrow list(key2, value2)$

Reduce function: $reduce: (key2, list\ (value2)) \Rightarrow (key3, value3)$

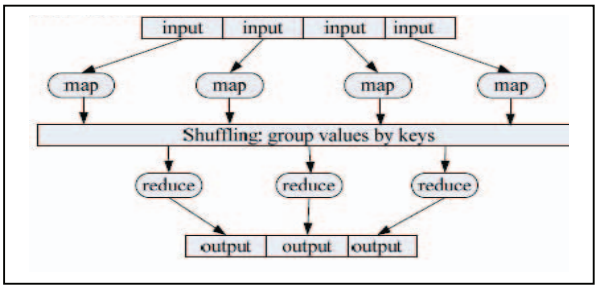

Fig. 1. Mapreduce Model

IV. DATA MINING TECHNIQUES

There are different types of data mining techniques, which can be used with big data. The main techniques used with data mining are as follows.

A. Association Analysis

Mining association rules in order to find the frequent patterns is the most significant field in data mining. These methods analyze the datasets and discover the patterns that occur frequently. Basically, to measure the strength of an association rule two criteria's, called Support and Confidence are used. Consider 'a' $\rightarrow$*'b', a*s an implication expression for an association rule, where 'a' and 'b' are disjoint item sets i.e.$(a \cap b = \emptyset)$. Support determines how frequently an item occurs in a given dataset, while confidence determines how frequently items in the subset 'b' appear in the transactions that contain item 'a' [12, 13]. A strong association rule should provide rules that satisfy a minimal support and minimal confidence threshold. Further, the result of an association analysis should contains the rules where $support \geq minsup$ and $confidence \geq minconf$. Here *minsup* and *minconf* are the corresponding threshold values [12]. For instance, the customers who have placed milk in their shopping basket, how likely are they going to place bread at the same time. This fact can be depicted by association rule $milk \Rightarrow bread\ [support = 20\%, confidence = 70\%]$, support of 20% means that 20% of all transactions contain milk and bread, while a confidence of 70% means that 70% of the customers who placed milk on their shopping basket, also placed bread.

B. Classification and Prediction

Classification is a data mining technique used to find a model that describes and distinguishes data classes or concepts. Moreover, classification can be used for the purpose of predicting certain outcome based on a given output [14]. For an instance, classification is used to classify countries based on their climate or classify books based on their topic. There are different types of classification models such as, classification by decision tree induction, Bayesian classification, neural networks, Support Vector Machines (SVM) and classification based on associations.

C. Cluster Analysis

Clustering is the process which is used to identify and group the items into classes of objects called clusters. A cluster is a group of data elements having similar characteristics within the same cluster and is dissimilar to the objects in other clusters. For example, it can be used to group books for book management in a library, grouping the blood group, grouping the genes and proteins with similar functionality [14].

D. Outlier Analysis

Outlier is used to analysis the objects in a dataset that are different from the remaining data. These data objects do not follow the general behavior or model of the data [14]. For

example, in a class of student with engineering field, the student from the medical field is considered as an outlier.

V. APRIORI-MAPREDUCE ALGORITHM

In this section, the most important data mining algorithm in the field of association rules called Apriori algorithm is discussed. For convenient, a software is developed in Java language, which runs the MapReduce-Aproiri algorithm based on the given inputs on Hadoop and provides the desired output.

A. Aproiri Algorithm

The Classic Apriori algorithm is based on the association rules as discussed in section 4. The primary idea of this algorithm is to find the frequent patterns from a transactional dataset [7]. The process of the algorithm is as follows.

Step1: User provides the minimum support and confidence.
Step2: Initially it generates the itemsets having a single candidate and then eliminates the itemsets which there support value is lower than the selected minimum support.
Step3: It joins the itemsets that are selected in step 2 with each other to generate the itemsets having with two candidates, in order to create the frequent itemsets having two candidates.
Step4: Repeat the steps likewise step3 until no more Itemsets.

Fig. 2. Pseudo-code of Classic Apriori Algorithm

B. Apriori-MapReduce Algorithm

In the classic Apriori algorithm, to find the frequent itemsets, algorithm has to check the dataset several times. When it comes to a tremendous amount of data, these several time checking becomes problematic, and it makes the computational and mining process expensive and time consuming. However, to solve such upcoming issues, there is needed for using a faster platform such as Hadoop. Therefore, the MapReduce implementation of Aproiri algorithm is provided, and it is executed within the Hadoop framework. Figure 3 illustrates the Apriori-Mapreduce algorithm that runs on MapReduce framework such as Apache Hadoop. First, the user provides the value for minimum support (min_Sup) and the maximum set size (max_Set) of the output. The algorithm starts with step 1, which runs on the map nodes, that generate the itemset (C_1) having a single candidate. In step 2, the outputs from the map nodes are collected through reducer in order to, generate the itemsets which satisfy the minimum support threshold (L_1). In step 3, Hash Base technique is used to improve the performance. Ideally, Hash Base techniques provide a k-itemset whose corresponding hashing bucket count is below the threshold and cannot be frequent. Further, buildHashTree() function is used to create the itemsets with k candidates. Finally, the reducer uses the itemset generated in step 3 and applies the minimum support threshold, to create the final itemset called L. The two steps 3 and 4 are continued until it reaches to max_Set value provided by the user.

VI. IMPLEMENTATION

In order to make the execution process easier software is developed in Java language, which executes the MapReduce-Apriori algorithm on Hadoop. The screenshot of software is depicted in figure 4.

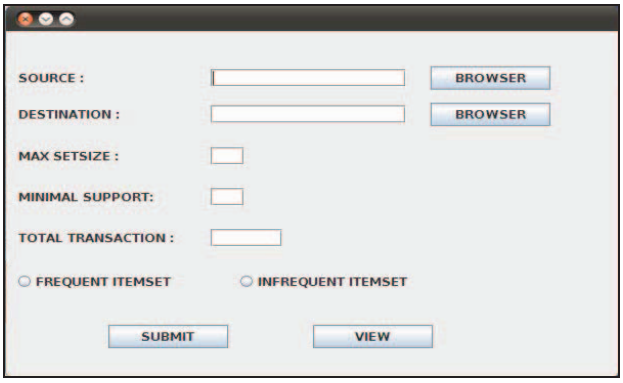

Fig. 4. Software Framework

As shown in figure 4, the location of the input and output files should be selected in the source and destination panel. Then, the Max Set size takes the size of required output itemsets. Besides, as discussed earlier, to provide a strong association rule, there should be a minimal threshold for support and confidence. Therefore, desired minimal support should be mentioned in software. Finally, the total number of transaction available in the input database should be entered in Total Transaction. For an experiment, the T10I4D100K dataset that contains transactional data sets is used. The table 1 illustrates a small part of the dataset that is given as input.
The algorithm is executed for frequent itemset with minimal support = 0.5 and infrequent itemset with minimal support = 0.75. The results of the execution are illustrated in table II, III and IV.

VII. CONCLUSION AND FUTURE SCOPE

The new generation of data, that is produced today, has motivated us to a need for a faster and more flexible platform. The efficient solution is Hadoop framework, which helps us to speed up the computational and mining process. In this paper, the Hadoop framework and its components are discussed in detail. In addition, the Apriori algorithm is redesigned into a MapReduce platform; therefore, its sequential computation

can be converted to a parallel format. Finally, a software is created which takes the transaction dataset and run the MapReduce-Apriori algorithm within the Hadoop platform and provide the frequent and infrequent itemset as output.

Input: D - {transactional database}
min_Sup - {minimal support}
max _Set - {Maximum set size}
Output: L - {large itemsets in D}

Step 1. In map nodes : Generate the itemsets having single item (C_1)
based on inputs
C_1= itemset {D}

Step 2. In reduce: use the C_1 itemsets and to generate L_1 which, select the itemsets which satisfy the minimum support threshold.
L_1= {c ∈C_1 ≥ min_Sup} //For Frequent Itemset
L_1= {c ∈C_1 <min_Sup} //For Infrequent Itemset

for (k=2; k ≤ max_Set;k++) do

Step 3: In map nodes :

C_{k-1}= apriori_gen(L_{k-1}, L_{k-1})
for i=0 TO previousLargeItemset-1
if (prune(largeItemsetMap,(C_{k+i-1}))

candidate_itemset=createCandidateItemset(C_{k+i-1})
end if
end for

C_k= buildHashtree(candidate_itemset, k-1)

Step 4: In reduce: use the $C_k$itemsets and to generate L which, select the itemsets which satisfy the minimum support threshold.
L_k = {c ∈C_k≥min_Sup} // frequent Itemset
L_k = {c ∈C_k<min_Sup} // infrequent Itemset

return L_k

Fig. 3. Pseudo-code of Apriori-MapReduce Algorithm

TABLE I. INPUT TRANSACTIONS

Transaction Database	
Transaction Id	***Transactions***
1	102, 103
2	102, 103, 107
3	102, 103, 105
4	102, 103, 105, 107
5	103, 105, 107
6	102, 103, 105, 107
7	102, 105, 107
8	102, 103, 105, 107

TABLE II. FREQUENT ITEMSET OUTPUT

Itemsets with One Candidate		**Itemsets with Two Candidate**		**Itemsets with Three Candidate**	
Itemset	***Counts***	***Itemset***	***Counts***	***Itemset***	***Counts***
102	7	102, 103	6	102, 103, 105	4
103	7	102, 105	5	102,103, 107	4
105	6	102, 107	5	102, 105, 107	4
107	6	103, 105	5	103, 105, 107	4
-	-	103, 107	5	-	-
-	-	105, 107	5	-	-

TABLE III. INFREQUENT ITEMSET OUTPUT

Itemsets with One Candidate		**Itemsets with Two Candidate**	
Itemset	***Counts***	***Itemset***	***Counts***
102	7	102, 105	5
103	7	102, 107	5
105	6	103, 105	5
107	6	103, 107	5
-	-	105, 107	5

TABLE IV. INFREQUENT ITEMSET OUTPUT

Itemsets with Three Candidate		**Itemsets with Four Candidate**	
Itemset	***Counts***	***Itemset***	***Counts***
102, 103, 105	4	102, 103, 105, 107	3
102,103, 107	4	-	-
102, 105, 107	4	-	-
103, 105, 107	4	-	-

REFERENCES

[1] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in", *AI magazine*, Vol. 17, pp. 37-54, 1996.

[2] IDC. The 2011 Digital Universe Study: Extracting Value from Chaos. [Online] Available from: *http://www.emc.com/collateral/analyst-reports/idc extracting-value-from-chaos-ar.pdf.*

[3] D. Che, M. Safran, and Z. Peng, "From big data to big data mining: challenges, issues, and opportunities", *in Database Systems for Advanced Applications*, pp. 1-15, 2013.

[4] W. Fan and A. Bifet, "Mining big data: current status, and forecast to the future", *ACM SIGKDD Explorations Newsletter*, Vol. 14, pp. 1-5, 2013.

[5] A. P. Deshmukh and K. S. Pamu, "Introduction to hadoop distributed file system", *IJEIR,* Vol. 1, pp. 230-236, 2012.

[6] J. Woo, "Apriori-Map/Reduce Algorithm", *in The 2012 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2012),* Las Vegas, 2012.

[7] N. Li, L. Zeng, Q. He, and Z. Shi, "Parallel implementation of apriori algorithm based on MapReduce", *in Software Engineering, Artificial Intelligence, Networking and Parallel & Distributed Computing (SNPD)*, pp. 236-241, 2012.

[8] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The hadoop distributed file system", *in Mass Storage Systems and Technologies (MSST)*, 2010 IEEE 26th Symposium on, pp. 1-10, 2010

[9] D. Borthakur, "The Hadoop Distributed File System: Architecture and Design", The Apache Software Foundation, 2007.

[10] N. Mirajkar, S. Bhujbal, and A. Deshmukh, "Perform wordcount Map-Reduce Job in Single Node Apache Hadoop cluster and compress data using Lempel-Ziv-Oberhumer (LZO) algorithm", arXiv preprint arXiv:1307.1517, 2013.

[11] S. G. Jeffrey Dean, "MapReduce: Simplified Data Processing on Large Clusters", *Appeared in Proceedings of the Sixth Symposium on Operating System Design and Implementation*, San Francisco, CA, , 2004.

[12] Association Analysis: Basic concepts and Algorithms, 2015, [Online]. Available: *http://www-users.cs.umn.edu/~kumar/dmbook/ch6.pdf*

[13] P. Stanišić and S. Tomović, "Apriori multiple algorithm for mining association rules", *Information Technology And Control*, Vol. 37, 2015.

[14] J. Han, M. Kamber, and J. Pei, "Data mining: concepts and techniques: concepts and techniques", Elsevier, 2011.

[15] J. Woo and Y. Xu, "Market basket analysis algorithm with Map/Reduce of cloud computing", *in The 2011 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2011),* Las Vegas, 2011.

[16] P. Agarwal, M. L. Yadav, and N. Anand, "Study on Apriori Algorithm and its Application in Grocery Store*", International Journal of Computer Applications,* Vol. 74, pp. 1-8, 2013.

[17] S. Kamepalli, R. R. Kurra, Y. Sundara Krishna, and A. Based, "Mining Infrequent and Non-Present Itemsets from Transactional Data Bases", *International Journal of Electrical & Computer Science.*