www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 5 Issues 7 July 2016, Page No. 17144-17147

Improved Map Reduce K Mean Clustering Algorithm for Hadoop Architecture

Shweta Mishra, Vivek Badhe
shwetamishra197@gmail.com, vivekbadhe@ggct.co.in
Department of Computer Science & Engineering (CSE)
Gyan Ganga College of Technology
Jabalpur (M.P.) India

Abstract:
Cluster is a gathering of information individuals having comparable qualities. The procedure of setting up a connection or getting data from crude information by performing a few operations on the information set like grouping is known as information mining. Information gathered in reasonable situations is usually totally arbitrary and unstructured. Consequently, there is dependably a requirement for examination of unstructured information sets to determine important data. This is the place unsupervised calculations come into picture to prepare unstructured or even semi organized information sets by resultant. K-Means Clustering is one such method used to give a structure to unstructured information so that significant data can be separated. Discusses the implementation of the K-Means Clustering Algorithm over a distributed environment using Apache Hadoop. The key to the implementation of the K-Means Algorithm is the design of the Mapper and Reducer routines which has been discussed in the later part of the paper. The steps involved in the execution of the K-Means Algorithm has also been described and this based on a small scale implementation of the K-Means Clustering Algorithm on an experimental setup to serve as a guide for practical implementations.

Keywords: K-Means Clustering, MapReduce, Hadoop, Data Mining, Distributed Computing.

I.INTRODUCTION

With the advancement of the electronics and the communication technology information generation rate has gain tremendous growth. Huge –Huge Amount of the data has been generated per hour from various medium of the Internet of Thing (IoT). Its termed as Big Data. All the computational analyzer has focused on to mined information to get the required knowledge.
Data mining is an art and science of efficient extraction of the useful information from the large repository in scalable, reliable and cost effective manner. Association rule mining is the most popular method to get the required information using existing relations. Market analysis is mostly relies on the association rule mining. Apriori and FP (Frequent Pattern) are the two most common methods for the association mining.
Although technologies have been pushing our world and society to a smarter one, human behaviors in the society still keep some inherent features and complexities that are hard to explain. Understanding the intelligence of human behaviors in the real world has great significance in practical application, such as mobile network deployment, traffic engineering, urban planning and service recommendation. While studies on human behaviors were not new in social science, quantitative analyses were not common due to the lack of source of data. Thanks to the computers and networks as they can now give plenty of computational ways of collecting and analyzing data for social studies, which used to depend on surveys in traditional methodology. Thus in the new era of ‘‘Big Data’’, never before have researchers had the opportunity to mine such a wealth of information that promises to provide insights about the complex behaviors of human societies [1,2]. One goal of these researches is to quantitatively uncover the inherent feature of human behaviors and track how our behaviors evolve by mining petabytes of network data.
To distributively do the efficient extraction of the modern user Big Data Analysis is the most preferable choice i.e. Hadoop.

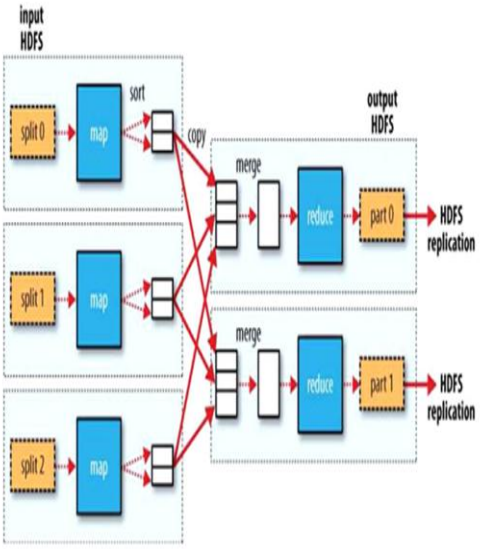
B. Big Data

Author of [2] has describes the working of MapReduce model proposed by Google for efficient parallel data extraction technique to operate on BIG DATA.
MapReduce is a programming model and an associated implementation proposed by Google for processing and generating large data sets in a distributed computing environment that is amenable to a broad variety of real-world tasks. Dean and
Ghemawat [7] described the MapReduce programming model as follows:

The computation takes a set of input key/value pairs, and produces a set of output key/value pairs. The user of the MapReduce library expresses the computation as two functions: Map and Reduce.

- Map-function takes an input pair and produces a set of intermediate key/valuepairs. The MapReduce library groups together all intermediate values associated with the same intermediate key I and transforms them to the Reduce function.
- Reduce-function accepts an intermediate key I and a set of values for that key. It merges together these values to form a possibly smaller set of values. Typically, just zero or one output value is produced per Reduce invocation.

Fig. 1shows the data flow and different phases of the MapReduce framework:



MapReduce Framework presented in [1]

Where HDFS stands for Hadoop distributed file system. While author has explain MapReduce with reference to Hadoop tool according to author [2], In Hadoop, each MapReduce job can be divided into 7 phases of execution, seeing Fig. 1. These are (1) mappers reading records, (2) map functions running to produce intermediate output, (3) combining the outputs to local storage, (4) shuffling the mapper output to reducers based on the partition function, (5) merging all mapper outputs, (6) reduce function running, (7) writing the reduce output to HDFS.

In Hadoop, the reduce task includes (4) (5) (6) (7) and its load comes from these phases. (4) (5) are overlapped with the map task execution. This is the synchronous mechanism between map and reduce task. Using it, a MapReduce job avoids network congestion, reduces the execution time and improves the Hadoop performance. The reason is that it fully utilizes the network resource and makes the shuffling and merging in a reduce task not waiting for all map tasks competition.

II.RELATED WORK

There have been extensive studies on various clustering methodologies. In this chapter, we will look at some of the studies carried out in the specific area of k-means clustering and MapReduce. Since the development of k-means clustering algorithm in mid 1970's, there have been many attempts to tune the algorithm and improve its performance. There are two important areas that concern most researchers. First, the accuracy of k-means clustering which is dependent on the choice of the number of clusters and the position of initial centroids. Secondly, the iterative nature of k-means algorithm that impacts scalability of the algorithm as the size of the dataset increases. Researchers have come up with algorithms that:

- Improve the accuracy of final clusters
- Help in choosing appropriate initial centroids
- Reduce the number of iterations
- Enhance scalability with increased dataset size or dimensionality
- Handle outliers well

The effectiveness of clustering is dependent on the choice of initial centroids. The right set of initial centroids create clean clusters and reduce the number of iterations. Jin, Goswami and Aggarwal, [12] in their paper, have presented a new algorithm called the Fast and Exact K- means clustering. This algorithm uses sampling technique to create the initial cluster centres and thereafter requires one or a small number of passes on the entire dataset to adjust the cluster centres. The study has shown that this algorithm produces the same cluster centres as the original k-means algorithm. The algorithm has also been tested on distributed system of loosely coupled machines.

In the paper, on Efficient Clustering of High-Dimensional datasets, Andrew McCallum, Ka- mal Nigam and Lyle Ungar have explained how one can reduce the number of iterations by first partitioning the dataset into overlapping subsets and iterating over only the data points within the common subset. This technique is called Canopy Clustering and it uses two differ- ent similarity measures. First, a cheap and approximate similarity measure is used to create the canopy subsets and then a more and accurate measure is used to cluster the points within the subset. This reduces the total number of distance calculations and, thereby, the number of iterations. [9]

While the above studies largely concentrated on tuning the k-means algorithm and reducing the number of

iterations, there have been many other studies on the scalability of k-means algorithm. Researchers have identified algorithms and frameworks to improve scalability. In the recent days, more research has been done on MapReduce framework.

III.PROPOSED ALGORITHM AND RESULT

Hadoop is an open source distributed computing platform, which mainly consists of distributed computing framework Map Reduce and distributed file systems HDFS. MapReduce is one of the core components of Hadoop, and it is easy to realize distributed computer programming by MapReduce on Hadoop platform.

MapReduce is a software framework for parallel computing programming model of large-scale data sets, having obvious advantages in dealing withthe huge amount of data.

Operation mechanism of MapReduce is as follows:

(1)Input: MapReduce framework based on Hadoop requires a pair ofMap and Reduce functionsimplementing the appropriate interface or abstract class,and should also be specified the input and output location and other operating parameters.In this stage, the large datain theinput directory will be divided into several independent data blocks for the Map function of parallel processing.

(2)MapReduce framework puts the application of the input as a set of key-value pairs <key, value>. In the Map stage, the framework will call the user-defined Map function to process each key-value pairs <key, value>, while generating a new batch ofmiddle key-value pairs<key, value>.

(3)Shuffle : In order to ensure that the input of Reduceoutputted by Map have been sorted, in the Shuffle stage, the framework uses HTTP to get associated key-value pairs <key,value> Map outputs for each Reduce; MapReduce frameworkgroups the input of the Reduce phase according to the key value.

(4)Reduce : This phase will traverse the intermediate data for each unique key, and execute user-defined Reduce function. The input parameter is < key, {a list of values} >, the output is the new key-value pairs< key, value >.

(5)Output : This stage will write the results of the Reduce to thespecified output directory location. Operation mechanism ofMapReduce is shown in Figure 1.

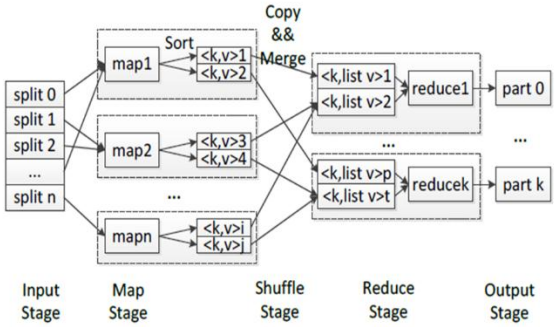


Fig.1.Operation mechanism of Map Reduce.

We generated the required number of d -dimensional data-points in the Start-up program. After trying different ranges and types of data, we chose to generate data-points in the range of 1 to 900. We generated 15 million 2d and 3d points. We then used the required number of data points randomly from the generated set.

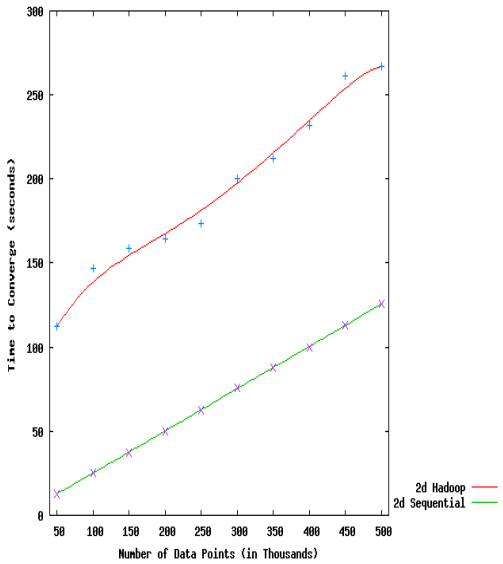


Figure 3: Comparative Plot : Hadoop MapReduce vs Sequential Clustering.

IV.CONCLUSION

In this paper, we have applied MapReduce technique to k-means clustering algorithm and clustered over data points. We compared the results with sequential k-means clustering. We have shown that k-means algorithm can be successfully parallelised and clustered on commodity hardware. MapReduce can be used for k-means clustering. The results also show that the clusters formed using MapReduce is identical to the clusters formed using sequential algorithm.

Our experience also shows that the overheads required for MapReduce algorithm and the intermediate read and write between the mapper and reducerjobs makes it unsuitable for smaller datasets.However, adding a combiner between Map and Reduce jobs improves performance by decreasing the amount of intermediate read/write. We also noted that the number of nodes available for map tasks affect performance and more the number of nodes, the better is the performance. Therefore, we believe, that MapReduce will be a valuable tool for clustering larger datasets that are distributed and cannot be stored on a single node.

REFERENCES

[1]Hadoop Page on Mahout. http://mahout.apache.org/, 2011.

[2]Hadoop Page on Disco. http://discoproject.org/, 2011.

[3]Hadoop Page on Pig. http://pig.apache.org/, 2011.
[4] Hadoop Page on Hive. http://hive.apache.org/, 2011.
[5] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters.
Communications of the ACM, 51(1):107–113, 2008.
[6] J. Ekanayake, T. Gunarathne, G. Fox, A.S. Balkir, C. Poulain, N. Araujo, and R. Barga. Dryadlinq for scientific analyses. In 2009 Fifth IEEE International Conference on e- Science, pages 329–336. IEEE, 2009.
[7] J. Ekanayake, H. Li, B. Zhang, T. Gunarathne, S.H. Bae, J. Qiu, and G. Fox. Twister: A runtime for iterative mapreduce. In Proceedings of the 19th ACM International Sym- posium on High Performance Distributed Computing, pages 810–818. ACM, 2010.
[8] D. Gillick, A. Faria, and J. DeNero. Mapreduce: Distributed computing for machine learning, 2006.
[9] S. Guha, R. Rastogi, and K. Shim. Cure: an efficient clustering algorithm for large databases. In ACM SIGMOD Record, volume 27, pages 73–84. ACM, 1998.
[10] S. Ibrahim, H. Jin, L. Lu, L. Qi, S. Wu, and X. Shi. Evaluating mapreduce on virtual machines: The hadoop case. Cloud Computing, pages 519–528, 2009.
[11] W. Jiang, V.T. Ravi, and G. Agrawal. Comparing map-reduce and freeride for data- intensive applications. In Cluster Computing and Workshops, 2009. CLUSTER'09. IEEE International Conference on, pages 1–10. IEEE, 2009.

[12] R. Jin, A. Goswami, and G. Agrawal. Fast and exact out-of-core and distributed k-means clustering. Knowledge and Information Systems, 10(1):17–40, 2006.

[13] W. Zhao, H. Ma, and Q. He. Parallel k-means clustering based on mapreduce. Cloud Computing, pages 674–679, 2009.