ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 7, January 2016

CLUSTERING BIG DATA USING k-MEANS ALGORITHM

K.Gaja Lakshmi
PG Scholar, Department of Computer Science and Engineering
Sri Krishna College of Engineering and Technology
Coimbatore, India
Email:gajalakshmi193@gmail.com

Dr.D.Prabha
Professor, Department of Computer Science and Engineering
Sri Krishna College of Engineering and Technology
Coimbatore,India
Email:prabha@skcet.ac.in

***Abstract*- Big data has evolved due to significant raise in use of data intensive technologies. Clustering has become even more challenging as the size of the data increases. In order to enhance the speed of clustering in large data sets MapReduce framework is used. In this system, k-means algorithm is used in MapReduce to make the clustering method appropriate to large scale data. MapReduce framework is a programming paradigm for distributed parallel computing. By using this framework iteration dependency can be reduced which is one of the bottlenecks in k-means algorithm. In k-means, the initial cluster centers are chosen randomly as a result the final clusters that are obtained varies each time. In the proposed system, the problems of processing large scale data using k-means algorithm is handled and the iteration dependency is eliminated using map reduce framework. It provides an efficient way to assign data points to suitable clusters.**

Keywords: Clustering, k-means, MapReduce

1. INTRODUCTION

A new generation of technology that is designed to economically extract value from huge volumes of a variety data is referred as big data. It becomes very difficult to perform effective analysis using the existing traditional computing techniques because of its larger size. The key factors for the growth of big data are increase in storage capacities, increase of processing power and accessibility of data. To attain value from big data, an organized set of solutions for capturing, processing, analyzing the data and discovering new insights is needed. Accuracy in big data may further direct to confident decision making and accurate decisions can lead to greater operational efficiency, cost reduction and reduced risk. The availability of big data, low-cost commodity hardware and analytic software has produced a distinctive moment in the account of data analysis. It is one of the best solutions for our day to day problems such as large-scale extract, transform, and load issues by using commodity software and hardware. But still it makes certain operations like analytical, process and retrieval time intense and difficult. Clustering helps to overcome this problem by compacting it in a format that gives an informative version of the entire data.

Cluster analysis has seen a vigorous growth in recent years. There are numerous types of clustering which includes partition, hierarchical, model-based, density based and grid based clustering. A good clustering method will produce high quality clusters with high intra-class similarity and low inter-class similarity. Partitioning a huge set of objects into homogeneous clusters is a basic operation cluster analysis. The k-means algorithm is best suitable for implementing this operation because of its effectiveness in clustering huge data sets. In the proposed system, the focus is done on improving the efficiency of k-means, a partition clustering algorithm to obtain a high clustering performance.

The k-means algorithm starts with the random selection of initial centers. The data points are assigned to its nearest center. This is done by computing the distance between the cluster center and the data point. Then it recomputed the center by calculating the mean and continues until the obtained result does not change for two consecutive rounds. One of the main problems in k-means algorithm is iteration dependency which depends on choosing initial cluster centers. In order to eliminate the iteration dependency a programming model called MapReduce is used [4].

The MapReduce framework helps to parallelize large computations easily by using map and reduce operations. Mainly two functions have to be specified in the MapReduce framework the map and reduce function that implements the mapper and the reducer. The map function is used to assign each input data points to the closest center. The reduce function is used to update the new centers. A combiner function is used to combine the intermediate data of the same map task in order to reduce the cost of network communication [5].

2. LITERATURE SURVEY

Parallel k-means clustering based on MapReduce The key solutions to meet the scalability and performance requirements are by building efficient parallel clustering algorithms and techniques. To solve this problem Weizhong [1] proposed a parallel k-means clustering algorithm based on MapReduce is used. The computation is expressed in terms of a map and reduces function, and the underlying runtime system automatically parallelizes the computation. In this k-means algorithm is adapted with MapReduce framework which is implemented by hadoop to make the clustering method appropriate to large scale data. By applying appropriate <key, value> pairs, the proposed algorithm can be effectively executed in

parallel. Key refers to the index of the closest center point and value represents sample information. The experimental results reveal that the proposed system can scale well and capably process large datasets on commodity hardware.

Scalable k Means++

In k-means a proper initialization is essential for obtaining a good final solution. A major drawback of the k-means++ is its sequential process, which restricts its applicability to massive data: one must make k passes over the data to locate a good initial set of centers. The proposed k-means++ initialization algorithm by Bahman [2] does this by obtaining an initial set of centers that is more close to the optimum solution. It starts with a random set of k centers. In each iteration, a clustering is derived from the current set of centers. The centroids of these derived clusters then become the centers for the next iteration. The iteration is then repeated until a stable set of centers is obtained. The iterative portion of the above method is called Lloyd's iteration. In this work it shows how to drastically decrease the number of passes needed to attain, in parallel, a good initialization. The proposed initialization algorithm k-means obtains almost a finest solution after a logarithmic number of passes, and then shows a constant number of passes success.

Speeding up k-means clustering by bootstrap averaging

The k-means clustering is time consuming as it converges to a local optimum of its loss function and the solution converged to be is mainly sensitive to the initial starting positions. Ian [3] proposed an approach called bootstrap averaging. It builds many models by creating small bootstrap samples of the training set and building a single form from each, identical cluster centers are then averaged to produce averaging similar cluster centroids by the position of attributes and their values to create signature. Grouping of clusters is done based on the signature. Bootstrap averaging on a portion of the dataset yields accurate results as clustering the entire data set .The results indicates that the size of data is directly proportional to the number of iterations of algorithm until convergence.

Adapting k-means for clustering in big data

The information that is available around is not available directly the needs have to be extracted from big data. The information is not directly available and needs to be extracted from big data. Existing technologies are inadequate to be deployed for big data analysis. An approximate algorithm proposed by Mugdha [4] is a novel method for big data analysis which is very fast, scalable and has high accuracy. It overcomes the disadvantage of k-means of uncertain number of iterations by fixing the number of iterations, without losing the precision. It is observed that the number of iterations essential for improved clustering is in fact a feature of the dataset itself. The number of iterations required in the proposed algorithm depends on the number of attributes to be considered for clustering. It also overcomes the problem of centroid initialization such that centroids don't have to be updated every iteration. Deciding the primary and secondary attributes is considered in the proposal to be provided as an input by the user. Machine learning concepts can be used to decide the priority of attributes instead of asking from the user.

3. SYSTEM ARCHITECTURE

The fig: 1 describes the architecture of the proposed system. The input data set is divided into required number of partitions to be processed parallel in MapReduce framework. The partitions are send to the mapper phase which is used to compute k clusters for each partition of the dataset and the local clusters are obtained. Next in the reducer phase the clusters are merged to obtain the final resultant clusters.

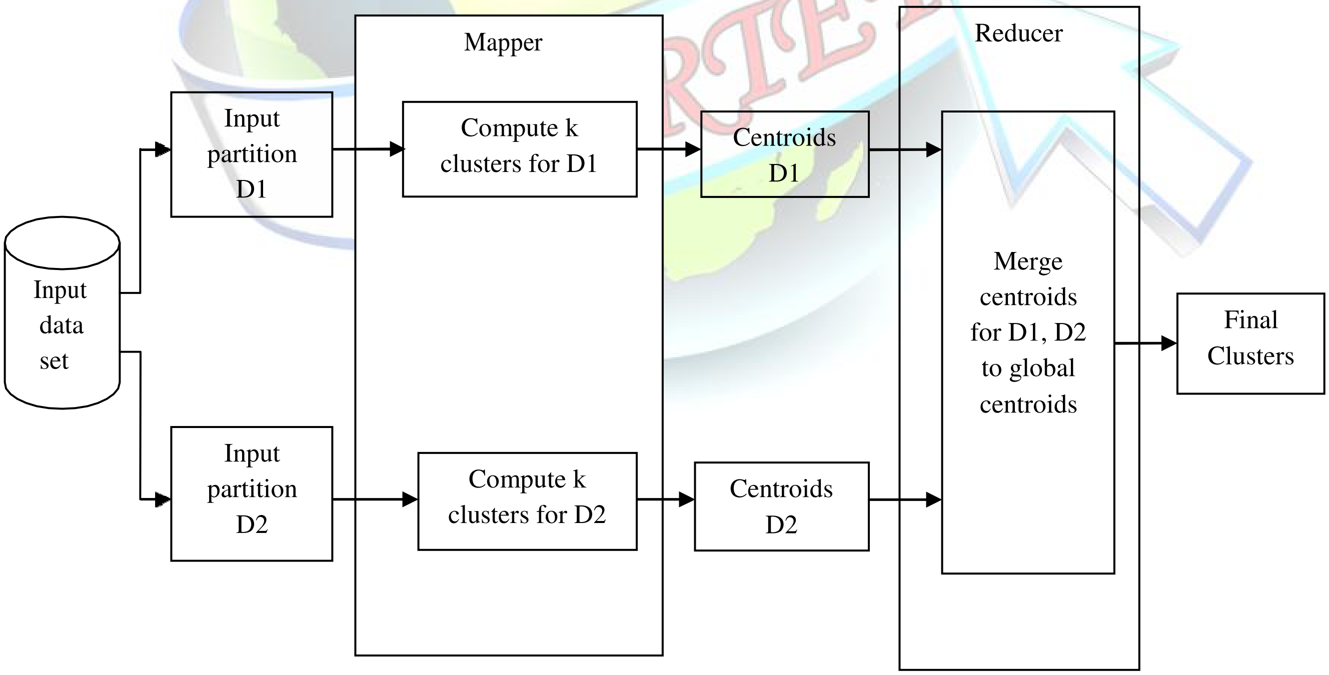

Fig1: Implementation of k-means algorithm using MapReduce

4. DESCRIPTION OF MODULES

In the proposed system the process involves three phases. The first stage involves collecting the input dataset and partitioning. The second stage involves implementing map function in k-means and then implementing the reduce function.

Input dataset collection and partitioning

The input dataset that has to be processed is done in MapReduce framework. The MapReduce framework is a programming model for data-intensive computing. The framework based on hadoop requires a pair of map and reduces functions. The map and reduce function are used for processing and generating large data sets. The input dataset is represented as a sequence file of <key, value>pairs, each represents a record in the dataset. The key denotes the offset in bytes of the record to the start position of the data file, and the value denotes the string content of the record. The collected dataset is divided into equal partitions and the framework assigns one partition to each map function.

Implementation of map function in k-means

In map function the user is required to handle the input of a pair of key value and produces a set of intermediate key and value pairs. It then combines the intermediate values with same key and passes them to reduce function. For every Input Split the MapReduce framework generates a map task, and it is generated by the InputFormat of job. Each <key, value> correspond to a map task. For every map task, it has centers which is an array carrying the information about centers of the clusters. A mapper can compute the closest center point for each sample with the given information. Iterate over each cluster centroid for each input key/value pair. Compute the euclidean distance and save the nearest cluster with the lowest distance to the input key/value pair. Perform the map task the input <key, value> will be processed to form a new <key, value>. This method is called “divide into groups”. The intermediate values consist of two parts: the index of the closest center point and the sample information.

Reduce function

After every map task, a combiner is applied to combine the intermediate data of the same map task. In order to compute the mean value of the objects for every cluster, the record is done on the number of samples in the same cluster and in same map task. The MapReduce framework guarantees the input to every reducer to be arranged by key. The procedure by which the system sorts and passes map output to reducers is known as shuffle. In reduce function, sum every sample and calculate the total number of samples assigned to the same cluster. The new centers are obtained which are used for next iteration till the convergence criteria are met.

5. CONCLUSION AND FUTURE WORK

The proposed algorithm aims at the problems of conventional k-means algorithm for big data. The main factor which affected the performance of clustering is the iteration in k-means. The proposed system overcomes those problems by using MapReduce framework. Experiment results on different dataset shows that the proposed algorithm is efficient. In future the focus is needed on improving the initial center points as they are chosen randomly.

REFERENCES

[1] ZhaoW, Ma H, He Q, "Parallel k-means clustering based on MapReduce", Cloud computing- Springer, pp: 674-679, 2009

[2] Bahmani B, Moseley B, Vattani A et al, "Scalable k-means++", Journal Proceedings of Very Large Data Bases Endowment, pp: 622– 633, 2012

[3] Davidson I, Satyanarayana A, "Speeding up k-means clustering by bootstrap averaging", IEEE data mining workshop on clustering large data sets, pp: 49-60, 2003

[4] Mugdha Jain, Chakradhar Verma, "Adapting k-means for clustering in big data", International Journal of Computer Applications, pp:19-24,2014

[5] Xiaoli Cui, Pingfei Zhu, Xin Yang, Keqiu Li Changqing Ji, "Optimized big data k-means clustering using MapReduce", Journal of Supercomputing, Springer, Volume 70, pp:1241-1259