ISSN: 2321-7782 (Online)

Volume 3, Issue 11, November 2015

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

Parallel Two Phase K-Means Based On Mapreduce

Bharath Kumar Gowru[1]
Assistant Professor,
Dept of CSE, Amrita Sai Institute of Science & Technology
India

Pavani Potnuri[2]
Assistant Professor,
Dept of CSE, Amrita Sai Institute of Science & Technology
India

Abstract: Clustering is defined as a process of creating collection of abstract objects into classes of related objects such as objects in the same class are related to each other than those in other classes. Clustering is one of the efficient techniques in data mining for doing static data analysis in number of domains for example data retrieval...Etc. These days technology is ever-increasing so the user data volume also increasing. Sometimes the data volume will be in tera bytes or more. Hence performing clustering on the large amount of data is becoming complicated. As a solution to this problem, a new clustering algorithm is proposed in Hadoop framework to group the large volumes of data. Hadoop framework has different strategies for saving large data efficiently. We can store this large data across many systems which are located in same place or different using Hadoop framework. Hadoop framework programming model is MapReduce in which map and reduce two phases will perform distributed computations efficiently on large volumes of data. The proposed algorithm is implemented in Hadoop framework following MapReduce programming model.

Keywords: Data Clustering, K-means, Parallel Distributed Computing and MapReduce.

I. INTRODUCTION

Data analysis is defined as a process of data cleaning, inspecting, data transforming, and data modeling with the aim of dig out useful information [1]. Based on the analysis results we can easily make decisions and conclusions in data processing. Data analysis has several approaches in different domains. Data mining is one of the data analysis techniques which focus mainly on data modeling and information discovery.

Data mining is the process of finding out different patterns from the large data sets. Data sets may be gathered from different repositories like database systems...Etc. The main objective of the data mining process is to dig out useful information from the data sets and transform analysis results into an understandable structure to use further. Data mining provides six classes of data analysis techniques which are anomaly detection, association rule mining, clustering, data classification, regression [2], and data summarization tasks. Clustering is defined as a process of creating collection of abstract objects into classes of related objects such as objects in the same class are related to each other than those in other classes. Clustering is one of the efficient techniques in data mining for doing static data analysis in number of domains for example data retrieval...Etc.

In data mining techniques K-means is a popular clustering algorithm for cluster analysis. K-means clustering groups the n observations into k clusters where each observation assigned to the cluster which has the nearest mean form it. The data on which data clustering is to be performed may be in structured or unstructured format. With the development of information technology, a large volume of data is growing day by day towards a terabytes or more. Performing clustering on such type of large amounts of data [3] is complex task now. While analyzing structured data most of the applications use a relational database to store the data. So in such cases we require some superior machines to run bigger databases to process large amounts of structured data which is cost effective [4].

1.1 Background and Motivation

The current parallel versions of K-means algorithms have several drawbacks like performance problems, synchronization and memory utilization. In the existing parallel K-means algorithms data sets are needed to be partitioned into equal parts among all the slave nodes to be processed. In the processing time intermediate data [5] will be collected from the slave nodes to update the global information by a Master node and then again that global information is needed to be broadcast to all the slave nodes. Therefore, synchronization is required at the end of each iteration in K-means. Sometimes the memory size of the slave nodes is smaller than the size of the data subset then data will be swapped between the memory and hard disk of the slave nodes which may slow down the performance of the algorithm [6]. Hadoop framework which is open source used for writing and executing distributed applications to process on large data sets across clusters of systems using simple programming model. Hadoop framework has different and better strategies for data storage and data processing to solve the problems existing in the current algorithms.

1.2 Objective

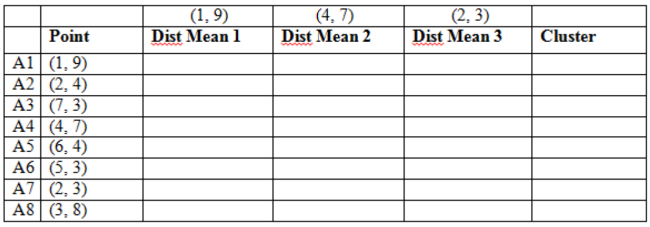
The main objective of this system is resolving the synchronization problems between master and slave nodes and data swapping problems in slave nodes in the distributed clustering process which could be implemented by parallel two phase k-means in Hadoop [7] framework based on mapreduce programming model.

II. LITERATURE SURVEY

Clustering is defined as a process of creating collection of abstract objects into classes of related objects such as objects in the same class are related to each other than those in other classes. In data mining techniques K-means is a popular clustering algorithm for cluster analysis. The process of K-Means algorithm is very simple to learn [8]. The process involves some steps. Firstly we define k cluster centroids one for each cluster. Usually it is preferred to choose initial centroids far away from each other. These k centroids will be considered as initial cluster centroids. Afterward we assign each observation belonging to a given datasets to the cluster whose centroid has the nearest mean from it. When all the data points are assigned we need to recalculate the k new centroids for the clusters resulted before. This process is iterated for some number of times. After completion of each iteration we will observe the new k centroid values [9]. Usually the process is repeated until the no more changes occur in k centroids. In general we follow two methods for selecting initial k centroids. One is forgy where we assign the positions of the k cluster centroids to k observations chosen randomly from the dataset. Another one is Random partition here we assign a cluster randomly to each observation or data point and then compute mean value. The computed mean value will be considered as initial cluster centroid.

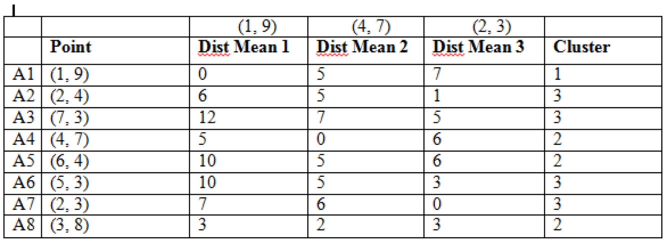
Problem: Cluster the given following eight points as (x, y) representing locations into three clusters A1(1, 9) A2(2, 4) A3(7, 3) A4(4, 7) A5(6, 4) A6(5, 3) A7(2, 3) A8(3, 8). Select initial cluster centers as: A1(1, 9), A4(4, 7) and A7(2, 3). Consider the following distance function between two points p1=(x1, y1) and p2=(x2, y2) is defined as: d(a, b) = |x2 – x1| + |y2 – y1| .

Here we follow k-means algorithm to find the new three cluster centers following the second iteration.



		(1, 9)	(4, 7)	(2, 3)	
	Point	Dist Mean 1	Dist Mean 2	Dist Mean 3	Cluster
A1	(1, 9)				
A2	(2, 4)				
A3	(7, 3)				
A4	(4, 7)				
A5	(6, 4)				
A6	(5, 3)				
A7	(2, 3)				
A8	(3, 8)				

Table 1: K-means Example Initial Centroids



		(1, 9)	(4, 7)	(2, 3)	
	Point	**Dist Mean 1**	**Dist Mean 2**	**Dist Mean 3**	**Cluster**
A1	(1, 9)	0	5	7	1
A2	(2, 4)	6	5	1	3
A3	(7, 3)	12	7	5	3
A4	(4, 7)	5	0	6	2
A5	(6, 4)	10	5	6	2
A6	(5, 3)	10	5	3	3
A7	(2, 3)	7	6	0	3
A8	(3, 8)	3	2	3	2

Table 4: K-Means Example After First Iteration

The K-means is one of the well known Data Mining clustering algorithm. There are a number of parallel versions of the this algorithm implemented on different programming frameworks, such as Parallel Virtual Machine (PVM), Message Passing Interface (MPI) [10].

2.1 K-Means On Message Passing Interface Framework:

The parallel K-means of Kantabutra and Couch is implemented in the master/slave model on the Message-Passing Interface (MPI) framework and that algorithm is executed on a cluster or network of workstations [11]. This algorithm contains one master to store all data sets. Master divides K data subsets and transforms each subset to a slave.

After each iteration of this K-means, a new centroid is computed in slave and later then broadcasted to other slaves. After this centroid broadcasting, data sets are also transmitted to slaves. Here slave nodes keep only data points which are nearer to the center in that slave [12]. This task involves a big data communication between slaves. Hence this algorithm is not appropriate for big data sets.

2.2 K-Means On Parallel Virtual Machine:

The parallel K-means algorithm by Zhang et al is developed in the master/slave model based on the PVM framework. And it is executed on a network of systems. In this algorithm, the master reads the datasets and also randomly initializes cluster set [13].

In each iteration master node transforms the cluster set to all slave nodes. The master divides the datasets into some x subsets. x may be larger than K and as a result master sends each subset to a slave node [14]. A slave clusters the received data subset based on the sent cluster set and then transforms the intermediate result again back to the master.

The master node re-calculates the cluster centroids based on the intermediate results. This is repeated until the cluster set does not change. This parallel version of K-means requires synchronization between nodes at the end of each an iteration.

2.3 Issues in the Existing Algorithms:

In conclusion, a number of parallel versions of K-means use the same data parallel approach. Here in those algorithms each slave node uses data which is initialized by the master node. Slave nodes Processes sub dataset [15] it received and requires synchronization in sending local data to the master node or distributing to other slave nodes before doing the next iteration.

This approach has several drawbacks. The master node frequently has to load the complete data set to distribute to the computing slave nodes, hence synchronization is required after the completion of each iteration, and a number of scans over the data set is also compulsory. Communication cost among the slave nodes is also much higher. If data subset size is bigger than the slave node memory size slave nodes swaps the data between memory and hard disk.

III. Proposed Approach

Any application development follows the some software engineering implementation models such as waterfall model, incremental model…etc. the appropriate [16] model is selected based on the requirements of the project. These process models give guidelines to the software developer in developing applications. Here the two phase K-Means is implemented in two modules one is K-MeansClusteringJob and another one implements mapper and reducer phases of K-Means.The SRS phase involves two tasks:

1. **Problem Analysis:** Here in this phase the main problem is identified and the main objective or goal of the application identified.
2. **Specific Requirements:** Here, the specific requirements such as tools to be used and other requirements are gathered.

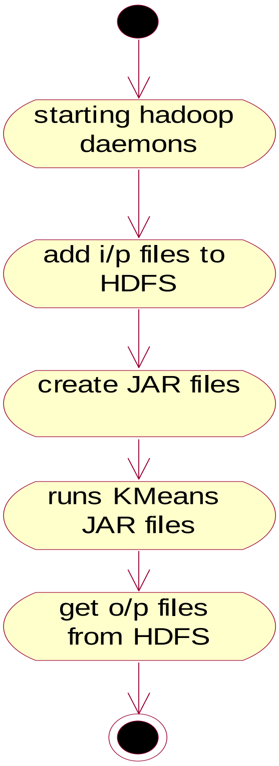


Figure1: Implementation of two phase k-means

3.1 Implementation:

The project has been implemented in Hadoop Framework. It has been developed using Java Development Tool Kit in Ubuntu as Eclipse as the IDE. This application uses the packages provides in JAVA API for coding mapper and reducer classes. For giving input to the application input files must be added to HDFS. To run a MapReduce job, users should implement a map function, a reduce function, and should know the input data, and an output data location. When executed [17], Hadoop carries out the following steps:

- Hadoop framework breaks the input data into multiple data splits and executes the map function once for each data split, giving the data split as the input for the map function. The map function results one or more key/value pairs after completion of execution.
- Hadoop framework collects all the key-value pairs resulted from the map function, and sorts them by the key, and aggregates the values with the similar key.
- For each different key, Hadoop framework runs the reduce function once while passing the key and list of values for that key as input.
- The reduce function may output one or more key-value pairs, and Hadoop writes them to a file as the final result.

3.2 Map and Reduce phases Of Parallel Two Phase K-Means

The parallel two phase k-means algorithm map and reduce functions implemented as below briefly:

- In the map step finds out the cluster center values randomly into memory from a sequence input file.

- Iteration occurs over every cluster center for all key/value pairs in the input file.
- Compute the distances and keep the nearest center which has the lowest distance into a defined vector.
- Write down the cluster center with its computed vector to the file system.
- In the reduce phase already it receives all the associated vectors of each cluster center. After Iterations are completed over every value vector and determine the average of the vector.
- Sum each vector and calculate mean value. This is the fresh cluster center, keep it into a File.
- Check the convergence between the old cluster center and the new center. If it they are not equal, increase counter value and repeat the process until nothing was changed anymore.

The parallel two phase KMeans implemented following modules. One is KMeans ClusteringJob, which is implemented to configure the MapReduce job settings. One is KMeans mapper class, KMeans reducer class, DistanceMeasure to calculate distance between cluster centroid and dataset points. Cluster Center to initialize cluster centers. Vector Writable to write the data on console.

3.3 Two phase k-means results

Figure2 shows the comparison of k-means and two phase k-means. We executed two phase k-means and conventional k-means on different datasets which has different instances. Shown in the given figure3, by taking number of instances of the input file on X-Axis and Execution time in seconds on Y-Axis.

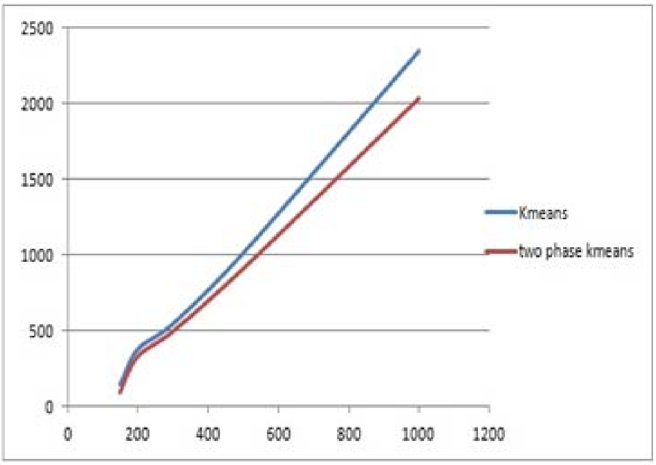


Figure2: Comparison results

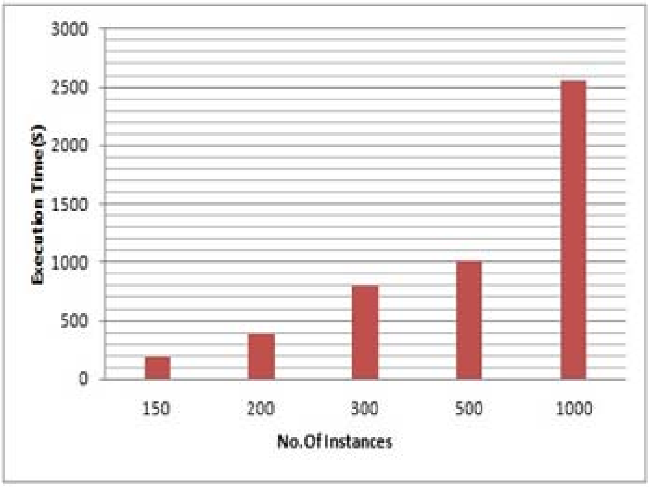


Figure3: K-means Results

IV. Conclusion and Future Scope

4.1 Conclusion

The Parallel Two Phase K-means based On MapReduce has achieved a good speed up Ratio on tested data sets. In Hadoop framework master and slave nodes works independently so the problems like communication overhead and synchronization problems are resolved. Hadoop framework has better strategies for data storage. HDFS file system is designedfor distributed data processing under frameworks such as Hadoop.

4.2 Future Scope

As data clustering has attracted a significant amount of research attention, many clustering algorithms have been proposed in the past decades. However, the enlarging data in applications makes clustering of very large scale of data a challenging task. Here the parallel Two Phase *k*-means clustering algorithm based on MapReduce, takes advantage of MapReduce's parallel computation capability to make the algorithm accelerated. And because of the number of nodes involved in the computation can be dynamically changed, it makes the method with high scalability. K-Means based on mapreduce in Hadoop fully distributed mode will give better efficient results than in distributed mode in large data sets.

References

1. Zhang, Y., Xiong, Z., Mao, J., Ou, L.: The Study of Parallel K-Means Algorithm. In:Proceedings of the Sixth World Congress on Intelligent Control and Automation (WCICA 2006), vol. 2, pp. 5868–5871 (2006)
2. Tian, J., Zhu, L., Zhang, S., Liu, L.: Improvement and Parallelism of k-Means Clustering Algorithm. Tsinghua Science & Technology 10(3), 277–281 (2005)
3. Kraj, P., Sharma, A., Garge, N., Podolsky, R., McIndoe, R.A.: ParaKMeans:Implementation of a parallelized K-means algorithm suitable for general laboratory use. BMC Bioinformatics 9, 200 (2008)
4. Pakhira, M.K.: Clustering Large Databases in Distributed Environment. In: IEEE International Advance Computing Conference (IACC 2009), pp. 351–358 (2009)
5. Kantabutra, S., Couch, A.L.: Parallel K-means clustering algorithm on NOWs. NECTEC Technical Journal 1(6), 243–247 (2000)
6. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations, Berkeley, California, vol. (1), pp. 281–297. University of California Press, Los Angeles (1967)
7. Pham, D.T., Dimov, S.S., Nguyen, C.D.: An Incremental K-means Algorithm. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 218, 783–795 (2004)
8. Pham, D.T., Dimov, S.S., Nguyen, C.D.: A two-phase k-means algorithm for large datasets. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 218(10), 1269–1273 (2004).
9. Dean, J., Ghemawat, S.: MapReduce: Simplified Data Processing on Large Clusters. In: OSDI 2004: Sixth Symposium on Operating System Design and Implementation, San Francisco, CA, pp. 137–150 (2004)
10. Chu, C.-T., Kim, S.K., Lin, Y.-A., Yu, Y., Bradski, G.R., Ng, A.Y., Olukotun, K.: Mapreduce for machine learning on multicore. In: NIPS, pp. 281–288 (2006)
11. Zhao, W., Ma, H., He, Q.: Parallel K-Means Clustering Based on MapReduce. In: Jaatun, M.G., Zhao, G., Rong, C. (eds.) Cloud Computing. LNCS, vol. 5931, pp. 674–679. Springer, Heidelberg (2009)
12. Zhou, P., Lei, J., Ye, W.: Large-Scale Data Sets Clustering Based on MapReduce and Hadoop. Journal of Computational Information Systems 7(16), 5956–5963 (2011)
13. Frank, A., Asuncion, A.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine (2010), http://archive.ics.uci.edu/ml
14. VMware virtualization technology, http://www.vmware.com (accessed in May 2013)
15. Kernel based virtual machine, http://www.linux-kvm.org (accessed in May 2013)
16. Linux Foundation Collaborative Projects, http://www.xen.org/products/xenhyp.html (Last accessed in May 2013)
17. Openstack: Open source software for building private and public cloud, http://www.openstack.org/ (Last accessed in May 2013)

AUTHOR(S) PROFILE

Gowru Bharath Kumar received the Bachelor’s Degree in Information Technology from Bapatla Engineering College in 2008-2012. He is received him master’s degree in computer science and technology from V R Siddhartha Engineering College. Now he is working as Assistant Professor in Amrita Sai Institute of Science& Techonology. His Research areas are Data Mining, Text Mining and Web Mining.

Pavani Potnuri received the Bachelor’s Degree in Information Technology from NOVA Engineering college affiliated to JNTUH. She is received her master’s degree in Computer Science and Engineering from SAARADA Engineering college affiliated to JNTUH. Now she is working as Assistant Professor in Amrita Sai Institute of Science & Technology. Her research areas are Data Mining and Software Engineering.