Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 7.056

IJCSMC, Vol. 9, Issue. 10, October 2020, pg.78 – 82

K-means++ Clustering Using MapReduce Framework for Large Datasets

Kodati Divya[1]; V. Purushothama Raju[2]

[1]Department of Computer Science and Engineering, Shri Vishnu Engineering College for Women (A), Bhimavaram, Andhra Pradesh 534202, India

[2]Department of Computer Science and Engineering, Shri Vishnu Engineering College for Women (A), Bhimavaram, Andhra Pradesh 534202, India

DOI: 10.47760/IJCSMC.2020.v09i10.010

Abstract— Clustering techniques are important to analyze the data due to the heavy increase of data in modern investigation. In real world these techniques are mostly used in many domains, some of them are financial analysis, social networks, digital marketing, etc. To support clustering over large scale datasets, public cloud infrastructure will play the major role for presentation of the data and financial trades. In our work, MapReduce based K-means++ clustering technique is proposed to allow the better grouping of data into appropriate clusters. The results of the paper denote that the present algorithm can efficiently process large amount of data.

Keywords— Clustering technique, MapReduce frame work, Cloud computing.

I. INTRODUCTION

Clustering is a technique which divides the large amount of data in to small number of groups. Clustering techniques are widely used in every sector because every industry contains large amount of data. By using clustering techniques we can easily group the required data from the datasets. Clustering techniques are based on the similarity and dissimilarity of the data objects.

MapReduce [1] is a technique used to process the data and a program model for distributed computing based on java. The MapReduce algorithm is divided into two tasks. They are Mapper phase and Reducer phase. Mapper task takes a set of data and divide the set of data into another set of data, where the data is divided into multiple tuples. The tuples are nothing but keys and values. Reducer task takes the output of the Mapper phase as an input and combines those data tuples into a smaller set of tuples. The Reducer task takes place after the completion of Mapper task only.

The main disadvantage in K-means algorithm [2] is the initializations of the centroids are very weak. It is very difficult to find out the centroids for every cluster. Sometimes one cluster may contain two centroids, two clusters may share only one centroid and some clusters may end up with no centroids.

To overcome the disadvantages in K-means algorithm, we proposed K-means++ algorithm. The initialization of the centroids for the clusters is little bit faster for K-means ++. So, the quality of the clustering is increased in K-means ++ algorithm while compared with K-means algorithm.

II. LITERATURE SURVEY

Doreswamy, et al. [3], the K-means is the most common clustering algorithm along with MapReduce to analyze huge quantity of data. By using K-means clustering algorithm, it is very difficult to find out the cluster centers for clusters. Sometimes the clusters may end up with no centroids. To overcome this problem, we proposed K-means++ clustering algorithm can find the cluster centers more easily while compared with K-means algorithm.

Palak Sachar, et al. [4], the genetic and K-means clustering algorithms are used for big data analytics. They are combined to make clustering process faster and lesser iterations. Our proposed algorithm can work more faster and take less number of iterations. The memory utilization is also less. So, the memory cost is much lesser than genetic and K-means algorithms.

Dongxi Liu, et al. [5], K-means clustering is outsourced with some privacy to give protection to the data, the output data is compared with the trapdoor information provided by the data owner. By doing this process it does not provide enough privacy. In our work, we used this technique as well as we are not decrypting owner's data at anywhere. The data is in the form of encryption only. So, the cloud server also cannot know about the information in it.

III. PROPOSED SYSTEM

In this paper, we proposed efficient MapReduce based K-means++ clustering algorithm over large datasets in cloud. In this we have four modules, they are data owner, cloud server, K-means++ clustering with MapReduce and clustering center. The data owner module will send the encrypted data objects to the cloud server to find out the clusters. The cloud server module will map and reduce the data objects to optimize the data. By using K-means++ clustering algorithm, the cloud server will find out the clusters without any decryption. The data is not decrypted at any place. So, we are giving protection to the data. To find out the cluster centers one clustering center module is taken by the cloud server. The clustering center module will find out the centers for clusters and update the result to the cloud.

To find out the K numbers of cluster centers from the taken data, first the algorithm selects one centroid from the data points randomly. Then the distance between each data point and the previously chosen centroid is calculated. After calculating the distances, other centroid is selected from the data points. The probability of selecting a point as a centroid is directly proportional to the nearest, previously chosen centroids. The process is repeated until to get K number of centroids. The final clustering results are out sourced by the cloud server.

Input:-

K number of clusters

Data Points $X_n = \{x_0, x_1, \ldots\ldots x_{n-1}\}$ $x \in X_n$

Output:-

$C = \{c_0, c_1, \ldots\ldots, c_{k-1}\}$ $k \in C_k$

Begin

Step 1: C ← fetch random data points from X_n

Step 2: ←X_n (C)

Step 3: foreach O log times do C

Step 4: Select each point $x \in X_n$ with independent probability

Step 5: Compute { X_n (C)}

Step 6: Re-clustering and update points in C into k clusters

Fig 1: Algorithm

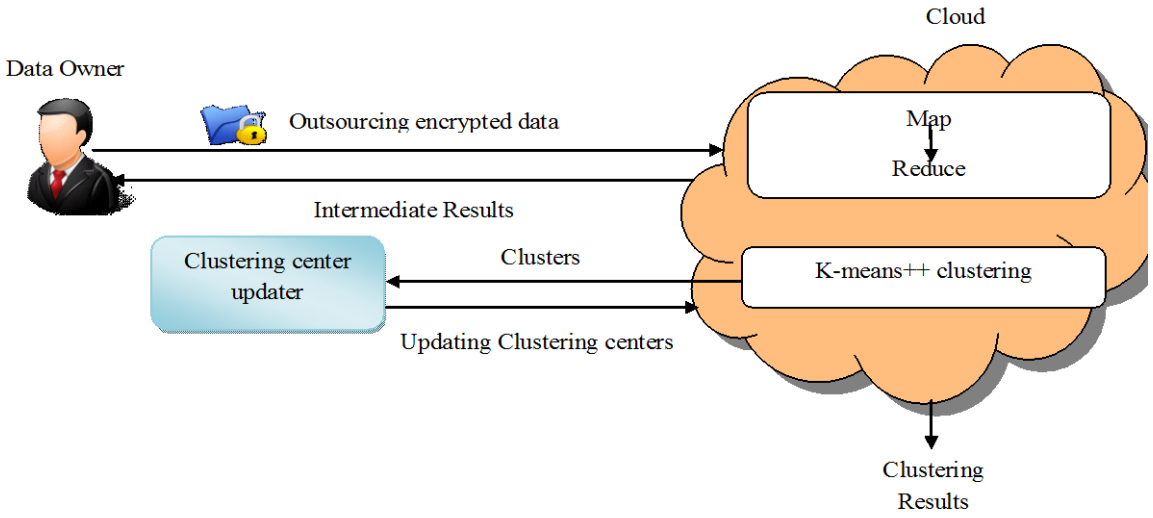

Fig 2: System architecture

As shown in the figure 2, the system architecture consists of two types of entities: the dataset owner, and the cloud server. Encrypted data objects are outsourced from the dataset owner to find out the clusters by the cloud server. At first the cloud server will apply MapReduce to the data objects and then by using K-means++ clustering algorithm the cloud server will find out the clusters without any decryption of the data. To find out the clusters, the cloud server will interact with data owner for encrypted intermediate inputs and outputs. The cloud server will take one clustering center and send the clusters to the clustering center to find out the centers for the clusters. After finding the cluster centers the final clustering results are outsourced by the cloud server.

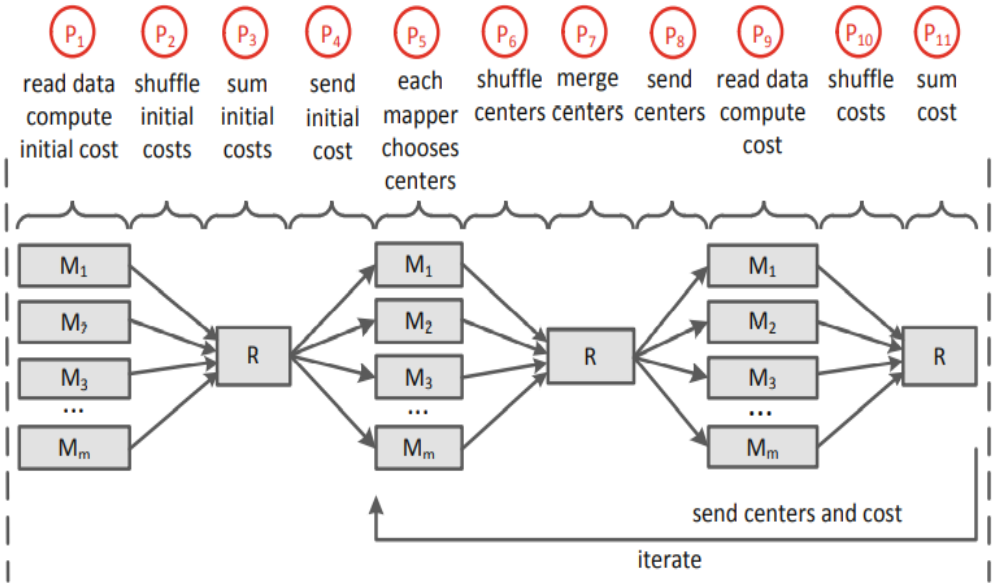

Fig 3: k-means++ with MapReduce Flow Analysis

As shown in the figure 3, M1, M2, M3,……., Mm are mappers, R is the reducer and P1, P2, P3,……,P11 are phases. Here, the total analysis is divided into three jobs. From P1 to P3 it is job1. From P4 to P7 it is job2 and from P8 to P11 it is job3. Job1 is used to calculate the initial clustering cost. Job2 will choose new centers and Job3

is responsible for calculating the clustering the clustering cost of this iteration. In phase P1, each mapper reads the input data and the first random center C1, and then evaluates the squared distance between each point. In phase P2, all the elements are shuffled to the same reducer. In phase P3, the reducer will calculate the sum of the elements and outputs the result to phase P4. In phase P4, the reducer will send the initial clustering cost to the P5. In phase P5, each mapper chooses its own centers. All these chosen centers are passed to the same reducer in phase P6. P6 will shuffle the centers and P7 will merge the centers at reducer. The merged centers are sent to the mapper in phase P8. Job3 is same as job1. From P5 to P11 it is single iteration. The centers and cost obtained at P11 is sent back to the phase P5.

IV. RESULT ANALYSIS

We used Intel i5 processor with 2GB RAM and Windows operating system. We used the sample dataset and implemented the algorithm by using java language.

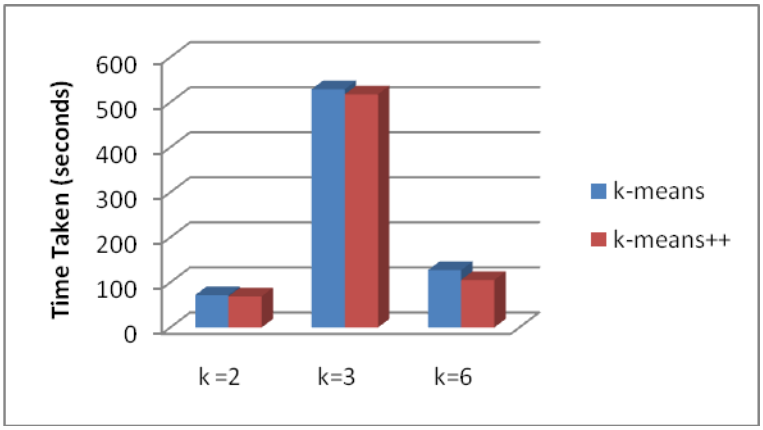

Fig 4: Performance Comparison

As shown in the above figure 4, we are taking the number of clusters (k) on x axis and time taken in seconds on y axis. By calculating the performance and time taken by both K-means [6] and K-means++ [7] algorithms, the results are occurred as shown in the above graph. The blue bar indicates the time taken by the K-means algorithm and the red bar indicates the time taken by the K-means++ algorithm. If K=2 the time taken for K-means is 71.794s and the time taken for K-means++ is 69.04s. If K=3 the time taken for K-means is 528.79s and the time taken for K-means++ is 517.8s. If K=6 the time taken for K-means is 127.18s and the time taken for K-means++ is 105.51s. The time taken for K-means++ is less then K-means. So, we can say K-means++ algorithm will work faster than K-means algorithm.

V. CONCLUSIONS

In this work, we proposed a K-means++ clustering using MapReduce framework for large datasets. Our plan of action realizes that the efficiency of the clustering is increased while compared to K-means clustering. By using MapReduce with K-means++ clustering algorithm, it makes suitable for cloud environment. When the encrypted data objects are received, K-means++ clustering is directly applied on it without any decryption by the cloud server. The cloud server will interrelate with the data owner for some amount of encrypted intermediates. The clustering will come to an end when the clustering result does not change or the preplanned iterations are completed. After getting the cluster, the cluster is allocated with a closet clustering centers. Due to its linear nature, K-mean requires more number of iterations. Our proposed algorithm K-means++ based on MapReduce can reduce the number of iterations.

REFERENCES

[1] Shweta Pandey and Vrinda Tokekar, "*Prominence of MapReduce in Big Data Processing*", 2014 Fourth ICCSNT. IEEE, Bhopal, India, pp 555-560, April 2014.

[2] Jiawei Yuan, Yifan Tian, "*Practical Privacy-Preserving MapReduce Based K-means Clustering over Large-scale Dataset*", pp 568-579, 2019 IEEE.

[3] Doreswamy, Osama A.Ghoneim and B.R. Manjunatha "*Scalable K-Means Algorithm Using MapReduce Technique for Clustering Big Data*" IJCTET. New Delhi, India. Special Issue SACAIM, pp 408-414 e-ISSN: 2278-621X, March 2016.

[4] Palak Sachar and Vikas Khullar. "*Hadoop Based Big Data Clustering using Genetic & K-Means Algorithm*" IOSR-JCE, Volume 19, Issue 1, Ver. IV, PP 114-121, Jan.-Feb. 2017.

[5] Dongxi Liu, Elisa Bertino and Xun Yi, "*Privacy of Outsourced k-Means Clustering*," ASIA CCS '14: 9th ACM symposium on Information, computer and communications security, Pages 123–134. Publication: ASIA CCS '14: Proceedings of the 9th ACM symposium on Information, computer and communications security June 2014 Pages 123–134.

[6] Ka-Chun Wong, "*A Short Survey on Data Clustering Algorithms*", 2015 Second ICSMI. IEEE, Hong Kong, China, pp 64-68. Nov. 2015.

[7] D. Arthur and S. Vassilvitskii, "*k-means++: the advantages of careful seeding*," in Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, ser. SODA '07. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035. [Online]. Available: http://portal.acm.org/citation.cfm?id= 1283383.1283494.

