ISSN No. 0976-

Volume 8, No. 1, Jan-Feb 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Analyzing mobile phone usage using clustering in Spark MLLib and Pig

Shefali Arora
Delhi, India
arorashef@gmail.com

Abstract: K-means is a common method of clustering data points using a predefined number of clusters. Apache Spark is a computing technology used for fast computation of data. By making use of its machine learning library called MLLib, we analyze mobile data obtained from Opencellid.org by clustering according to latitude and longitude values ,using K-means algorithm. Once each data point is assigned its cluster number , the dataset is loaded into Apache Pig to calculate the number of users in each cluster. Thus, we can analyse the number of users using a mobile network in a particular range of latitude and longitude.

Keywords: Spark, Pig, clustering, mobile, data, analysis

INTRODUCTION

Apache Hadoop is an open source framework created in 2005 to handle big data problems It is built on the top of MapReduce and HDFS. HDFS is a master –slave architecture which actually stores data by replication among nodes while Mapreduce is a framework which processes data in a parallel manner.

Apache Spark as an interface is much better than traditional Mapreduce as it provides a better performance ad also options to perform machine learning tasks, using its library called MLlib.

RDDs or resilient distributed datasets are used in Spark to perform various complex operations. These can be stored in memory without using replication. RDDs can support a large number of algorithms and querying operations. K-means clustering is implemented using RDDs in MLlib.

Whereas Apache Pig is an abstraction of MapReduce. It is a scripting language and Hadoop ecosystem tool, used to perform data manipulation operations in Hadoop.

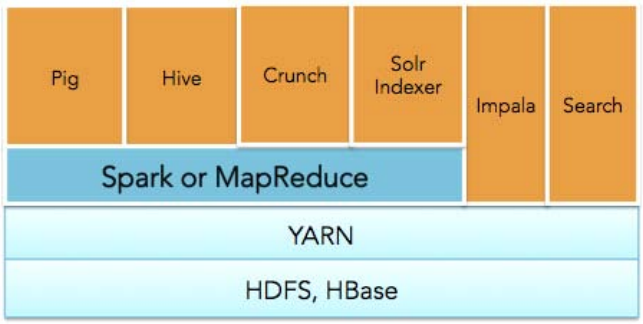

Figure 1.Placement of Spark and Pig in Hadoop architecture

LITERATURE SURVEY

Gopalani and Arora[1] compared the performance of K means algorithm using traditional MapReduce algorithms and Apache Spark The results in Apache Spark were proved tobe better.

Zaharaia et al[2]. worked on the implementation of Apache Spark, while maintaining the scalability and fault tolerance of Mapreduce. Spark makes use of RDDs to achieve these goals. It was found that Spark outperforms Mapreduce in performance by a factor of 10.

Shanahan and Dai[5] worked on the analysis of large distributed data using Apache Spark.

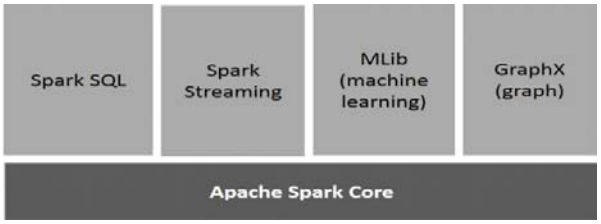

Figure 2. MLlib as a machine learning component of Spark

Meng et al.[6] worled on MLlib , Spark's open source machine learning library for various statistical operations. MLlib supports various languages and APIs for development of high end machine learning pipelines. It has scalable implementations of machine learning algorithms which include regression, classification and clustering.

Hartigan and Wong [8] designed a K means algorithm which takes matrix of M points in N dimensions as the input and divides it into K clusters so that within-cluster sum of squares is minimized.

Bradley and Fayyad[9] gave an efficient technique of K means clustering by refining the initial conditions. This helps the algorithm to converge to a better local minimum.

Kanungo et al.[10] worked on a filtering algorithm based on k-means algorithm. Kd-tree is the data structure used and it is observed that this algorithm runs faster as separation between clusters increases.

DATA ANALYSIS

Dataset used

The dataset used has been obtained from opencellid.org and consists of around 1GB data. This data consists of location information and network types of mobile users . We consider the latitude and longitude values as centroids or cluster centers for k means clustering.

K means clustering

Considering N data points which are to be partitioned into k disjoint subsets S_j. It is done using the following steps:

•k sets are defined to partition data points.

•A centroid μ is assigned randomly from among the data points.

•Each data point is assigned a cluster which is nearest to the centroid.

•This is repeated till there is no change in the assignment of clusters.

•The sum of squares is assigned using the following formula:

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 \quad (1)$$

Here x_n denotes the nth data point and μ_j denotes the centroid.

USING SPARK MLLIB FOR KMEANS CLUSTERING

The architecture of data analysis has been shown as below:

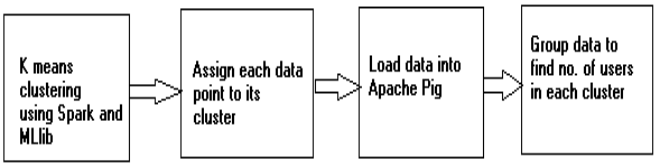

Figure 3. The process of data anlaysis.

The JavaRDD class in MLlib is used to import all the methods available in MLlib for clustering. The number of clusters are taken to be 500 and number of iterations are fixed to be 20.

The following steps are followed:

•Data is split into three files of different sizes i.e with 7500,12000 and 50000 records respectively.

•Generate cluster centers using trained Kmeans model in MLlib and predict cluster number for each record of dataset.

•Merge original records with their respective cluster number.

•The dataset is loaded into Apache Pig[13]. Apache Pig scripts are converted into MapReduce jobs to perform operations. The GROUP BY operation in Pig is used to group data by clusters .

•Store back the file on HDFS and analyse the count of users in each clusters.

•We have visualized the results using a software called Carto, which is used to generate heat density maps.

RESULTS

The final dataset obtained consists of the cluster number and the number of respective users in each of them. A snapshot of it is shown as below table.

This has been obtained for all the 500 clusters.

The final results are visulaized using a heat density map in Carto. This shows the number of users in each region of the world using a particular mobile network.

8.243939	47.53788	0	884
8.373987	51.36643	1	1
7.845726	51.95994	2	1
6.488998	46.73647	3	5
7.699842	49.22478	4	104
7.446776	50.09589	5	5
7.159288	46.79905	6	13
9.551518	47.05494	7	28
5.950253	46.13079	8	426
8.837661	40.72771	9	66
8.28196	48.85386	10	1
8.928125	50.31571	11	1

Figure 4. A sample of data with number of users in each cluster (based on clustering of latitude and longitude)

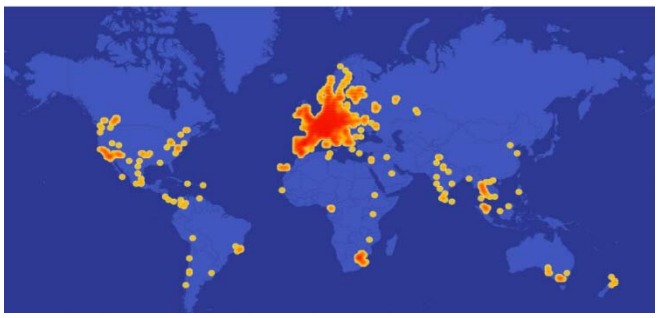

Figure 5. Visulaization of number of users in each cluster.

CONCLUSION AND FUTURE WORK

A large amount of data obtained from different sources can be channelized and visualized using various Big data tools and can help to identify a number of patterns. In our case, analysis of a large amount of mobile data has helped us to gauge the diversity of people who use mobile network in a particular range of latitude and longitude. Using machine learning methods like clustering , we have efficiently grouped a large set of people and their affinity towards different kinds of mobile networks.

As future work, we can use this data and machine learning methods in MLlib to find more network problems or congestion in mobile networks using Hadoop ecosystem tools.

REFERENCES

1. S. Gopalani,R. Arora ," Comparing apache Spark and MapReduce with performance analysis using K-means , in International Journal of Computer Applications , vol. 113 – No.1,New York,2015.
2. M. Zaharia,M. Chowdhury,M.J. Franklin, S. Shenker,I. Stoica,University of California, Berkeleympp. 1-7.
3. Hadoop Mapreduce tutorial : http://hadoop.apache.org/common/docs/r0.20.0/mapred_tutorial.html.
4. B. Nitzberg and V.Lo., "Distributed shared memor : a survey of issues and algorithms", in Computer , 24(8) : pp. 52-60, 1991.
5. G. Shanahan and L. Dai, "Large scale distributed data Science using Apache Spark",in Proceedings of the 21th ACM SIGKDD International Conference on Knowledge discovery and Data Mining,pp. 2323-2324,Australia, August 2015.
6. X. Meng,J. Bradley, B. Yavuz,E. Sparks,S. Venkataraman,D. Liu, J. Freeman, D. Tsai, M. Made, S. Owen, D. Xin, R. Xin, M. Franklin, R. Zadeh,M. Zaharia and A. Talwalkar, "Journal of Machine Learning Research 17 , pp. 1-7,2016.
7. S.B. Kotsiantis , "Supervised machne learning : A review of classification techniques, in Informatica,31,pp. 249-268.
8. J. A Hartigan, and M.A Wong, "Algorithm AS 136 : A K-means clustering algorithm, " in Journal of the Royal Statistical Society. Series C (Applied Statistics), Vol 28, No.1, pp. 100-108, 1978.
9. P. Bradley and U.M. Fayyad, "Refining initial points for K means clustering," in Technical report, Microsoft research, May 1998.
10. T. Kanungo,D.M Mount,N.S Netanyahu,C.D Piatko, R. Silverman and A.Y. Wu, "An efficient k-means clustering algorithm: analysis and implementation," in IEEE Transactions on Pattern analysis and Machine Intelligence, IEEE, pp. 881-892, 7 August 2002.
11. J. Yu, J. Wu, M. Sarwat , "GeoSpark : A cluster computing framework for processing large scale spatial data", in Proceedings of 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM Digital Library Article 70, Washington,2015.
12. https://opencellid.org/
13. C. Olston,B. Reed,U. Srivastava, R. Kumar, A. Tomkins, "Pig-lation : A not so foreign language for data processing", in Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, ACM ,pp.1099-1110, Vamcouver Canada, 2008.

