International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 05 | May-2016 www.irjet.net p-ISSN: 2395-0072

An Efficient Cluster Optimization Technique for High Dimensional Data

Ms. PRIYANKA P[1], Mr. A V KRISHNA MOHAN[2]

[1]M.Tech Student, Dept. of CSE, SIT, Tumakuru, Karnataka, India
[2]Assistant Professor, Dept. of CSE, SIT, Tumakuru, Karnataka, India

---

Abstract –

A tool which provides a scalable data intensive along with MapReduce architecture is Hadoop. Hadoop tasks are executed in distributed manner i.e., to say Map task and Reduce task. Among them Map task's execution is done on a large cluster and hence execution of Map task will require resources that are too expensive. Minimizing the cost of these expensive resources is crucial. AT present days, the amount of information that is being generated by computational systems and all electronic devices to say telescopes, medical devices and so are too exploding. Analysis of these petabytes of data generated is too complex and so we will require new algorithms or the old ones must be reconfigured. To discover and extract information of a data, the most powerful technique used is named as clustering and the algorithm used to cluster data's according to their characteristics is named as K-mean algorithm. Though K-mean being a well-designed algorithm, a complete parallel version of K-mean algorithm is not being explored. In this paper, a parallel version of the algorithm is designed for high dimensional data. Advantages of this version are its reduced computational time when comparing it with the traditional design. Also, this method can process a large amount of data effectively.

Key Words: **Big data, Scheduling, Interactive MapReduce, Hadoop**

1.INTRODUCTION

In the present era, data is being generated every second through computational systems, social media, Internet of things and so on, leading to the increase in volume of data which will be fueling it even in future. The rate at which this volume of data grows is overwhelming and comes with variety, which need not necessarily be structured. Also, the data may contain some wealth information that may act as a key in competing business. This massive amount of data is termed as BigData.

Now, BigData as the name itself implies; it's a enormous quantity of data. Gartner, who went through researches based on BigData noted that BigData has three most important assets to say; It's huge-volume, variety and velocity. These three assets require some pioneering technologies for processing with which the cost effectiveness property should be added so as to give the capability for decesion making easier.

Volume in BigData refers to how huge or enormous or karge the datasets are and with this increase in volume how the size of data's in today's world reaching terabytes and now alsp zetabytes of information. So, due to this abandant infromation created, it turns out to be a gigantic challenge for storage as well as for analysis.

BigData's another asset is its Variety. Variety means that the dataset will contain both structured data (which means the data has a well designed structured and easy to analyse. Eg: Relational dataset..) and unstructured data (which doesnot have any specific structure. Eg: Images, Video..) These unstructured dataset will have many attributes say about hundreds or even thousands of attributes that will lead in to plentiful amount of information and such huge data's will cause failure in traditional visualization tools.

The most important asset in BigData becomes it's Velocity. Velocity is an attribute which notifies as how fast the data is being produced on a daily or hourly basis. Records show that there are 1.3 exabytes of mobile data traffic per month just by 5 billion mobile users, 144 billion emails are sent in a day by 2.2 billion email holder and nearly about 7 petabytes of photo data being uploaded on facebook each month.

BigData now, according to its characteristics, Volume, Velocity and Variety becomes more challenging in both analysis point of view as well as storage point of view. When it comes to storage, Bigdata uses HDFS in Hadoop to store data. HDFS is a Hadoop Distributed File System where the data is stored in Distributed manner which makes it easier to store data at a faster rate and also access or read the data file from the system at a equally faster rate.

After dealing with technologies based on collection of data, nowadays the challenging task is; how the analysis or the execution of these huge collection of data done? Scientists and researchers also state that the most important area to focus today is on computation of BigData. Social media websites i.e., Facebook, twitter and so have billion number of users and are being increased day by day where they produce hundreds or even gigabytes of data each minute. Also retailers and other businessman's continuously update with their costumer's data. Dealing with these huge volume of generated data is too complex and hence, some powerful new tool is required for the knowledge discovery of these data.

The colossal amount of data being collected will be divided in to two groups such that the object in each group would match with some similarities compared to the objects in other group and this technique is called Clustering. This technique Clustering is necessary for various applications that helps in analyzing patterns, decision-making,

classification of pattern, and image segmentation. Clustering is mainly used for knowledge discovery of the data.

A resource computing which is scalable and the storage of data through the Internet [1]-[2] is provided by a most popular technique called Cloud computing. Regardless to where the cloud services are provided, this technique permits the cloud user to access services and how they will be delivered which seems similar to other commodity [3]. For data-intensive applications we have a well-known cloud-computing platform called Hadoop [4]. Cloud computing system has large number of nodes, which leads to the hardware failure. Based on the statistical analysis of hardware failure in [5]-[6], the probability of hardware failure cloud-computing system will not be a big issue. Although some hardware failures have chances of damaging the disk data of nodes and so as a result the running applications may face some failure in fetching the data from the disk. To overcome this failure and to provide high availability of data [7]-[8], a technique named data replication is being used in cloud-computing environment.

In this paper, an efficient clustering approach for High dimensional data is proposed to classify the KDD dataset. The KDD dataset is a intrusion detection of dataset which constitutes of attack data's and normal data's. To classify these data we use clustering approach through the use of K-mean algorithm.

2. LITERATURE SURVEY

Though we have numerous researches based on clustering and many clustering methods being in practice, K-mean clustering algorithm has been given a major attention by most of the researchers as it exploids both sequential and as well parallel execution; parallel in other words say to be an distributed approach.

K-mean being a very simple and straight forward algorithm, it is differentiated from others for it's iterative nature which results with good scalability, even as the size of datasets increases. The right choice of selection of initial centriod's will help in improvising the performance [11] of the design.

On the other hand, In Canopy Clustering the authors of [12] and [13] have explained that by dividing or partitioning the subsets in to overlapping subsets, the number of iterations can be reduced. But this Clustering failed to improvize the execution time. So, parallel or distributed K-mean algorithm raised as a solution which results in faster execution.

Many reasearchers attempted to execute or parallelize the K-mean algorithm in parallel or distributed manner. Authors of [10]-[14] used Mapreduce framework along with the Hadoop distributed file System (HDFS) having an idea to distribute the computational job's to all the nodes being alive in the system. So, the authors applied K-mean technique locally on the nodes of the system and the results generated were taken by the master nodes and these master nodes generate a global centriod so that k-mean algorithm can be aplied again on them. The generated global centriod were assigned to all the worker nodes alive in the system where inturn these centriods will be used as initial centriods ny the worker nodes. The same procedure will be repeated as long as there is no change between the initial centriod and the master nodes' final centriod. Here, the time complexity of the system was completely dependent on the number of iterations it took to complete the exection.

On the other hand, authors in [15]-[16] presented a more similar technique which used MPI interface and a massively parallel code was created which used thousands of worker nodes to analyze a large volume of datasets. Basically, this approach came to into idea for reducing the number of iterations or improve the sampling method which improvizes the time complexity anf reach fast coverage.

3. OBJECTIVE

Our main objective is to identify and understand the dynamics concerned in BigData technoligies. Also understand Hadoop, architecture of the same, it's distributed storage system i.e., setup Hadoop and explore it. Also to analyse the KDD (Knowledge Discovery Dataset) and in Hadoop framework, we show data processing. Also in Hadoop we discriminate between normal and malicious URL datasets and show that it's processing time is reduced compared to that processed using traditional framework.

4. PROPOSED SYSTEM

In our system we make use of MapReduce framework. This framework's job is to split the input data into smaller chunks; which are then Map functions job to take care of the processing them and they are done completely synchronized in a parallel manner. The output of the Map function is then fed to the Reduce task as input. Fig-1 shows the complete overview of MapReduce. This whole scenario of the Mapreduce framework explained that the framework involves scheduling the jobs then monitoring them and also re-executing them if so the execution was failed.

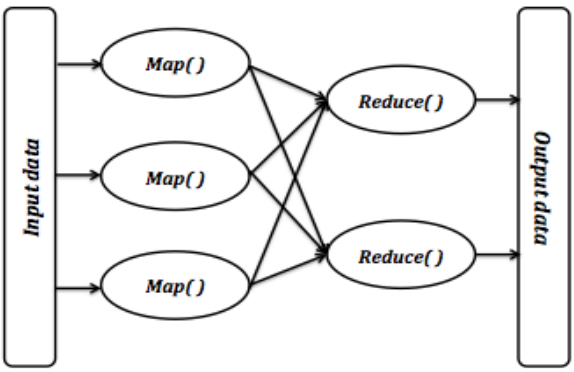

Fig -1: MapReduce framework

A cluster basically is comprised of multiple engines. The number of map tast and Reduce task in a MapReduce framework is constituted as a Mapreduce job and this mapreduce job is executed on the cluster. In the Map task, the working node reads the local data from the dataset and then applies the Map function to the input sets and then writes the output data on to the blob storage which is the intermediate storage in Mapreduce framework. The job of the worker node is: based on the output keys which are produced by Map function to distribute or schedule Map data, so that the data's which represent in the same key is been situated on the same azure worker node. So now, each and every group of output map data is processed by the worker node in a paralel manner.

MapReduce framework also permits for the processing to be in distributed operation i.e., Map operation and reduce operation are distributed processing. These processing is provided onlt when each or every mapping operation is self-reliant or in other words independent of others and parallel performace is availble or can be done for each maps; though in the real-time scenarios, the number of independent data sources are limited and not only the number of independent nodes but also the number of VM's near each source. In the same manner, the Reducer task is responsible for the reduction phase, if and only if the output of the map operation (from the same key) is offered on to the same Reducer also on the same time. In other words; reduction function is associative. But, this method can be in efficient when compared to the methods that have sequential approach. Although mapReduce frame work has an advantage that this framework can be applied to a larger volume of data which a normal server fails to handle (A large server can execute peta byte of data in just few hours). Also the parallelism technique allows in recovery of any partial failure i.e., if any Maper or a reducer fails then the job will be rescheduled to any other working node getting the input from the distributed storage.

4.1 Parallel K-Mean:

A K-mean algorithm is a clustering method which partitions the data's into clusters (say n points are partitioned into k clusters). Firstly, different kinds of data are identified (say k different data's) and they are named as centers for k different clusters. Then, all the data's (say n data's) will be partitioned into clusters based on the centres. So, each cluster will have similar kinds of data.

Parallel K-mean algorithm requires MapReduce job. The Map task is to identify the data's closest center and to assign that data to that particular cluster. The Reducer job is to update with the new centers. Also we make use of a combiner function to reduce the cost of network communication, where as the combiner function is designed to deal with the intermediate values which has same key and same map task in a partial combination manner.

4.1.1 Map Phase:

The HDFS[9] storage is done in a sequential fine where the data is represented in a <key,value> pairs, where each one identifies a record in the dataset. Here, the key represents the start point of the data file and are presented in types of bytes and the value represents the string of the contents present in that data file. This dataset is fragmented and are duplicated in to all mappers. Each map task of the Parallel K-mean algorithm will build a global variant center (which holds information about the cluster center). Using this information a mapper finds the closest center for a particular sample or data. The intermediate values now will be composed pf two different parts: the closest center's inders; the sample information.

4.1.2 Combine Phase:

Once the map task execution is done, we make use of a combiner so that the intermediate data which comes from the map task is combined. As the storage of these intermediate data's is done in the local disk, the consumption of the communication cost is decreased. Here, in combiner we only take the partial sum values of the points that are assigned in the same cluster. A record of the number of samples in the cluster which are from the same map task are used to calculate the mean of objects in each cluster.

4.1.3 Reduce Phase:

Now, the output of the combiner function is fed as input to the reducer function. As said in combiner phase, the output of the combiner phase will hold the partial sum of values of same number of samples that reside in the same cluster which now turns as input to the reducer function. Here, in reducer function, the total number of samples which are assigned are summed up as this helps to find the new centers and that can be used for next iteration.

5. SYSTEM DESIGN

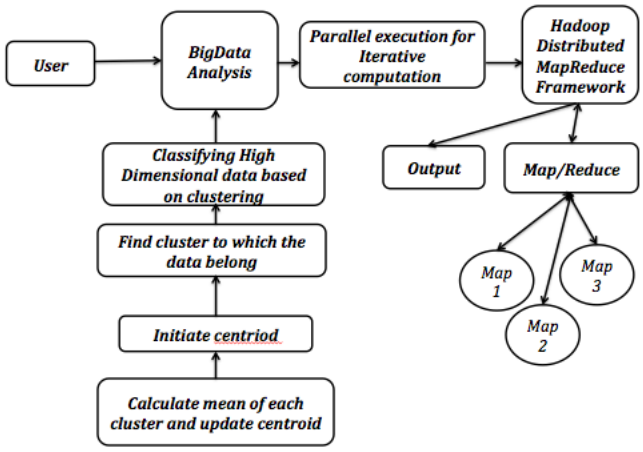

Fig -2: Design flow of our proposed system

For our implementation purpose, we take in KDD datadets as input datasets to the proposed system. KDD dataset consists of huge amount of data. We use these standard dataset to examine, where these datasets contain a variety of

intrusions which replicate a military network environment that is these intrusion dataset contain two types of data's in it: one being the attack data and the other being the normal data which means the data passed without any types of attacks.

Fig-2 represents our design flow diagram, the user submit jobs to Hadoop MapReduce framework and parallel iterative computation is started Map-Reduce framework is start computing and output of these jobs is stored in HDFS and log of task execution like cpu execution time, Heap size etc.... is shown. Mean while for big data analysis data is classified based on cluster and find that to which data belong to which cluster for that we have calculate mean of each cluster and update centroid.

6. CONCLUSION

Though more significant amount of research attention for high dimentional data clustering is attracted and algorithms being proposed, still the enormous or continuously increasing data in applications results in clustering of these data a very challenging task. So, we have proposed a K-mean algorithm which is a efficient parallel clustering algorithm based on MapReduce.

In our design we have partitioned between the normal and attack data of KDD dataset implementing parallel K-mean algorithm using Hadoop framework.

We use Hadoop so as to improvise the performance compared to that of traditional technique and we evaluate the performance based on the computation time. Also the proposed algorithm can process the datasets on comodity hardware efficiently.

ACKNOWLEDGEMENT

My Sincere thanks are due to my guide Assistant Professor Mr. A V Krishna Mohan, for the valuable guidance offered during every stage of my research work. I thank Principal, SIT and the management of SIT for the support, encouragement and facilities provided at the institute.

REFERENCES

[1] zM. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing," Dept. EECS, California Univ., Berkeley, Tech. Rep. UCB/EECS-2009-28, Feb. 2009.

[2] M. D. Dikaiakos, D. Katsaros, P. Mehra, G. Pallis, and A. Vakali, "Cloud Computing: Distributed Internet Computing for IT and Scientific Research," IEEE Internet Comput. vol. 13, no. 5, pp. 10–13, Sep. 2009.

[3] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility," Future Gener. Comput. Syst., vol. 25, no. 6, pp. 599–616, Jun. 2009.

[4] (2013) Apache Hadoop Project. [Online]. Available: http://hadoop.apache.org

[5] K. V. Vishwanath and N. Nagappan, "Characterizing Cloud Computing Hardware Reliability," in Proc. ACM Symp. Cloud Computing, Jun. 2010, pp. 193–204.

[6] B. Schroeder and G. A. Gibson, "Disk Failures in the Real World: What Does an MTTF of 1,000,000 Hours Mean to You?" in Proc. 5th USENIX Conf. File and Storage Technologies, Feb. 2007, pp. 1–16.

[7] F. Wang, J. Qiu, J. Yang, B. Dong, X. Li, and Y. Li, "Hadoop High Availability through Metadata Replication," in Proc. 1st Int. Workshop Cloud Data Manage., 2009, pp. 37–44.

[8] W. Li, Y. Yang, J. Chen, and D. Yuan, "A Cost-Effective Mechanism for Cloud Data Reliability Management Based on Proactive Replica Checking," in Proc. 2012 12th IEEE/ACM Int. Symp. Cluster, Cloud and Grid Computing (CCGrid), May 2012, pp. 564–571.

[9] Borthakur, D.: The Hadoop Distributed File System: Architecture and Design (2007).

[10] I. K. Savvas and M. T. Kechadi, "Mining on the cloud: K-means with mapreduce," in 2nd Int. Conference on Cloud Computing and Service Sciences, CLOSER, 2012, pp. 413–418.

[11] R. Jin, A. Goswami, and G. Agrawal, "Fast and exact out-of-core and distributed k-means clustering," Knowledge and Information Systems, vol. 10, no. 1, pp. 17–40, 2006.

[12] S. Guha, R. Rastogi, and K. Shim, "Cure: an efficient clustering algorithm for large databases," ACM SIGMOD International Conference on Management of Data, pp. 73–84, 1998.

[13] A. McCallum, k. Nigam, and L. H. Ungar, "Efficient clustering of high-dimensional data sets with application to reference matching," ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 169–178, 2000.

[14] W. Zhao, H. Ma, and Q. He, "Parallel k-means clustering based on mapreduce," 1st International Conference on Cloud Computing, vol. Springer: Cloud Computing 5931, pp. 674–679, 2009.

[15] J. Kumar, R. T. Mills, F. M. Hoffman, and W. W. Hargrove, "Parallel k-means clustering for quantitative ecoregion delineation using large data sets," Procedia Computer Science, vol. 4 (2011), pp. 1602–1611.

[16] J. Zhang, G.Wu, X. Hu, S. Li, and S. Hao, "A parallel k-means clustering algorithm with mpi," in Parallel Architectures, Algorithms and Programming (PAAP), 2011, pp. 60–64.