Survey on Clustering on the Cloud by Using Map Reduce in Large Data Applications

M Chaitanya Kumari[1], P Nagendra Babu[2]

[1]M.Tech Student, Computational Engineering in CSE, RGUKT APIIIT – RK VALLEY, Andhra Pradesh, India

[2]M.Tech Student, Computational Engineering in CSE, RGUKT APIIIT – RK VALLEY, Andhra Pradesh, India

***Abstract*— The term Clustering implies grouping of objects depends upon their similarity. In another way clustering is the process of grouping a set of objects so that objects within a group or cluster have high similarity, but comparing objects with other clusters must have high dissimilarity. In Cloud computing multiple users can access a single server to retrieve and update their data without purchasing licenses for different applications. The need of clustering on cloud is to retrieve the appropriate data because now a days we are dealing with peta bytes of data. For this reason we are using map reduce frame work which handles huge amounts of data by using two phases such as "Map" and "Reduce". Several algorithms such as K-Means, K-Medoids, CLARA, and CLARANS are used in clustering. If we use CLARA with Hadoop Map Reduce frame work cloud will be very effective and we can achieve better efficiency.**

***Keywords*— CLARA, Map Reduce, Scientific Computing, Cloud Computing, Scheduling algorithm, K-Means**

I. INTRODUCTION

Scientific computing [3] deals with large-scale scientific modelling and simulation in different domains like climate research, astrophysics, bio informatics and mechanical engineering. Execution of large and accurate simulations in these domains requires significant computing resources. This computing has been closely connected to high performance computing by utilising the computing resources of super computers, computer clusters and grids to perform the large scale calculations needed. Cloud computing provides easier access to public resources and multiple users can access a single server to retrieve and update their data without purchasing licenses for different applications. Cloud computing relies on sharing of resources and if focuses on maximizing the effectiveness of shared resources. Hadoop map reduce framework deals with huge amounts of data like Google to mine the data effectively. For this we need clustering techniques with different algorithms and we can achieve efficiency by using map reduce frame work with clustering algorithms. In this paper, we can study reducing complex clustering algorithms to map reduce on the cloud with specific methods named as Map and Reduce.

II. CLOUD COMPUTING

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to shared pool of computing resources such as networks, servers, centralized data storage, applications and online services that can be rapidly provisioned and released with minimal management effort or service provider interaction. It reduces the environmental damage as well since less power, air conditioning. In cloud computing number of users can have facility to access a single server to get and update their data without purchasing licenses for different applications. The networking method supports the transmission of data from one end-point such as Local Network to the Cloud i.e. Wide Area Network and then decomposed to another intended end-point. In this computing we no need to buy every configurable resources but we have to pay for using that resources so we can reduce the budget burden. Various types of services provided by the Cloud are Software as a Service, Platform as a Service and Infrastructure as a Service. Many products developed by Google, it has come up with a wide variety of apps like Google Docs, Google Maps, Gmail which are based on Software as a Service provided in the cloud.

III. HADOOP MAP REDUCE

Hadoop dedicated to scalable, distributed, data-intensive computing. It deals with peta bytes of data because every day Google processed more than 20 peta bytes of data. To handle such large amount of data, the data processing frame work must be highly scalable and provide fault tolerance. Map Reduce has been proved to meet requirements such as constant thread of hard ware or network failures with Google's own patented frame work and other large business using alternative map reduce frame works like Hadoop for their huge data processing needs. It is less suitable for complex algorithms because it mainly handles huge amounts of data and it is designed for embarrassingly parallel data processing algorithms [7].

Hadoop is a Java-based software frame work that enables data-intensive application in a distributed environment, this code usually written in Java-though it can be written in other languages with the Hadoop streaming API. The Hadoop Big data i.e. huge amounts of data structure management structure comprises of two parts such as Distributed Data Processing implemented using the Map-Reduce concept and Distributed Data Storage implemented by the Hadoop Distributed File System. The three major categories of machine roles in a Hadoop deployment are Client Machines, Namenode (Master Node) and Datanodes (Slave nodes). HDFS is the primary storage system used by Hadoop applications. It splits the data into several pieces called data blocks. It creates multiple replicas of data blocks and distributes them on computer nodes throughout a cluster to enable reliable, extremely rapid computations [8].

A. Functionality of a Map Reduce

Map Reduce is a programming model as well as a framework that supports the model. The main idea of the Map reduce model is to hide details of parallel execution and allow users to focus only on data processing strategies. The Map reduce model consists of two primitive functions: Map and Reduce. Fig. 1 explains Hadoop map reduce architecture [11].

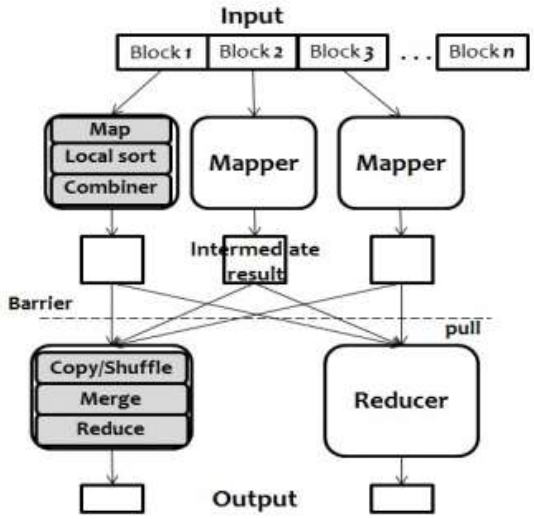

Fig. 1 Hadoop Architecture

The functional pieces of Map and Reduce is seems as Master/ Slave nodes. Master node takes the large problem as input and slices it into smaller sub problems then distribute these smaller problems into different Worker or Slave nodes. Worker nodes again split these problems into smaller pieces and it will go like multi-level tree structure. Worker processes smaller problem and hands back to master.

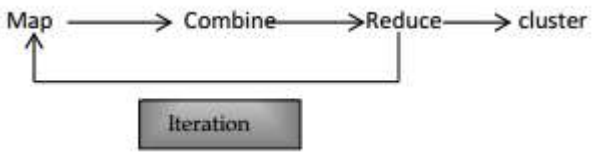

Fig. 2 Iterative method to form final Cluster

The above figure Fig.2 explains iterations of map reduce function [9]. Master node takes the answers to the sub problems and combines them in a pre-defined way to get the output or answer to original problem. Due to this scalability, simplicity and the low cost to build large clouds of computers. Map reduce is a very promising tool for large scale data analysis. Fig. 3 explains Map reduce function execution [10].

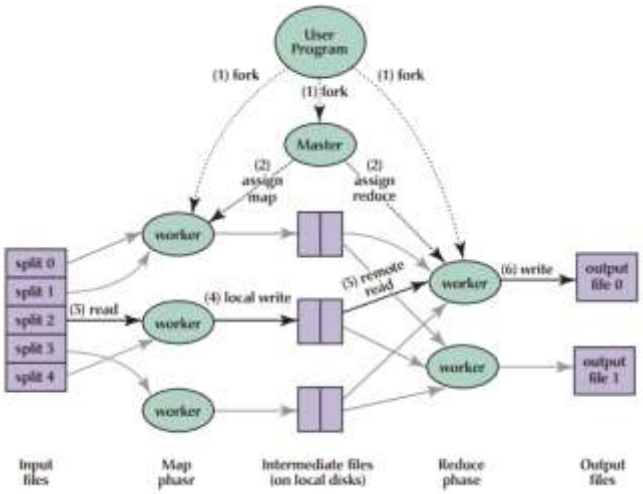

Fig. 3 Execution Overview

IV. CLUSTERING ALGORITHMS

Cluster analysis is the process of partitioning a set of data objects or observations into subsets. In that subsets each subset is a cluster, such that objects in a cluster are similar to one another, yet dissimilar to objects in another clusters. The set of clusters resulting from a cluster analysis can be referred to as a "Clustering". Generally clustering is known as "Unsupervised Learning" because the class label information is not present in given data. For this reason clustering is a form of "Learning by Observation", rather than learning by examples. Clustering is also called as segmentation in some applications because clustering partitions large data sets into groups according to their similarity. Clustering can also be used for "Outlier Detection", where outliers (values that are "far away" from any cluster) may be more interesting than common cases.

We can generate different types of clusters on the same data. Clustering type depends on the data size, complexity of data, efficiency and performance in both space and time. Commonly we are using K-Means algorithm [6] to form the clusters from the given data sets because this is easy algorithm to understand clustering and very easy to implement by using Java Code. K-means algorithm is the most well-known and commonly used clustering method. It takes the input parameter, k, and partitions a set of n objects into k clusters so that the resulting intra-cluster similarity is high whereas the inter cluster similarity is low [1]. This algorithm randomly choose 'k' objects from D (data sets) as the initial cluster centres. Depends upon their similarity reassign each object to the cluster to which the object is the most similar, based on the mean value of the objects in cluster. Now, update the cluster means, i.e. calculate the mean value of the objects for each cluster until no change of allocating objects to similar clusters. This is the basic algorithm and it will not handle the large amounts of data. Instead of taking means in this algorithm we can take medoids which algorithm called as PAM (Partitioning Around Medoids) for increasing efficiency. Compare to K-Means, K-Medoid is more robust in the

presence of noise and outliers. In this both methods we have to specify the no.of clusters by user.

PAM will work well for small data sets. Another clustering algorithm called CLARA (Clustering Large Applications) select set of data and apply PAM on that data and returns best clustering as the output. The complexity for each iteration in PAM is O (k (n-k) ^{2}) and in CLARA is O (ks^2 + k(n-k)) where 's' is the sample size, 'k' is the no.of clusters and 'n' is the no.of objects. A good clustering based on sampling will not necessarily represent a good clustering of the whole data set if the sample is biased. So we need to go for other algorithms such as CLARANS and Hierarchical methods.

V. MAP REDUCE WITH LARGE DATA APPLICATIONS IN CLOUD

On the cloud network, Hadoop map reduce framework is there to deal with large data applications. Hadoop map reduce works well with simple and embarrassingly parallel algorithms. It is focussed on data processing and is less suitable for complex algorithms of scientific algorithm. We already known that CLARA is a complex algorithm and map reduce is deals with large data applications so if we use CLARA with Hadoop map reduce framework cloud will be very effective and we can achieve efficiency because of non-embarrassingly parallel algorithm.

Everything can be reduced into two Map reduce jobs, both will execute on different tasks. First job chooses a number of random subsets from the input data sets, clusters each of them concurrently using PAM, and outputs the results. The second map reduce job calculates the quality measure for each of the results of the first job, by checking them on the whole data set concurrently inside one map reduce job. Input to a map reduce application is a list of key-value pairs and the core of the application consists of two methods, Map and Reduce. Map method processes each key-value pair in the input list separately, and outputs the result as one or more key-value pairs. As a result of having only two map reduce jobs, the job latency stays minimal and the input data set is only read twice.

Map (key, value) → [(key, value)]

The output of the map method is grouped by the key and divided between different tasks. Reduce method gets a key and a list of all values assigned to this key as an input, performs user defined aggregation on it and outputs one or more key-value pairs.

Reduce (key,[value]) → [(key, value)]

To define a map reduce application, user only has to write these two methods and frame work handles everything else: data distribution, communication, synchronization and fault tolerance. This makes writing distributes applications with map reduce much easier, as the frame work allows the user to concentrate on the algorithm and is able to handle almost everything else. Apart from Hadoop we can also use Message Passing Interface (MPI) [2] to parallelize and implement the CLARA algorithm in Python. It provides a number of commands to send messages between processes running on separate machines, which are typically used for distributing data and coordinating concurrent tasks. We can compare the efficiency and scalability of our CLARA map reduce implementation to the most widely used parallel computing solution [3].

Not only Hadoop framework but also others have tried to improve the Map reduce frameworks to provide better support for iterative map reduce applications. Frame works like HaLoop which extends Hadoop map reduce frame work by supporting iterative map reduce applications and Twister [4] is a Map reduce frame work, which is advertised as an iterative map reduce frame work. Twister map reduce frame work distinguishes between static data that does not change in the course of the iterations and normal data which may change at every iteration. Spark [5], a frame work that supports iterative applications, yet retains the scalability and fault tolerance of map reduce. Spark focuses on caching the data between different map reduce like task executions by introducing resilient distributed datasets that can be explicitly kept in memory across the machine in the cluster.

VI. CONCLUSION

Clustering on cloud by using the some of the clustering algorithms will get appropriate results and we can achieve efficiency. In clustering algorithms CLARA is a complex algorithm and it is non-embarrassingly parallel algorithm. We know Hadoop map reduce deals with huge amounts data so we are working with Hadoop frame work to reduce CLARA to Hadoop by using two functions map and reduce. We can do this same thing with other algorithms like CLARANS, K-Means etc. In future we can reduce CLARANS to map reduce frame work because Hadoop frame work is less suitable for complex algorithms and very efficient for large data sets.

Acknowledgment

This material is based upon clustering on cloud and other related papers. We would like to thank P Ravi Kumar, Co-ordinator of M.Tech RGUKT RK Valley, for his support to complete this work

REFERENCES

[1] Weizhong Zhao,Huifang Ma and Qing He, *parallel K-Means Clustering Based on MapReduce*, Institute of Computing Technology: Cinese Academy of Sciences, pp. 674-679, 2009.

[2] M.Snir, S.W.Otto, D.W.Walker, J.Dongarra, and S.Huss-Lederman. *MPI: The Complete Reference*.MIT Press,1995.

[3] Jakovits, Pelle and Satish Narayana, *Clustering on the Cloud:Reducing CLARA to Map Reduce*,Mobile cloud Lab,Institute of Computer Science: University of Tartu,2013.

[4] J.Ekanayake, H. Li, B. Zhang, T. Gunarathne, S.H. Bae, J. Qui, and G. Fox. Twister: a runtime for iterative map reduce. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, HDPC' 10, pages 810-818, New York, NY, USA, 2010. ACM.

[5] M. Zaharia, M. Chowdhury, M.J. Franklin, S. Shenkar, and I. Stoica. Spark: cluster computing with networking sets. In 2nd *USENIX conf. on Hot topics in cloud computing*, HotCloud'10, page 10, 2010.

[6] Chen. Li, Yanfeng. Zhang, Minghai. Jiao and Ge. Yu, *Mux-Kmeans: Multiple Kmeans for Clustering Large-Scale Data Set*, Northeastern University: China, 2014.

[7] Leykin, Anton; Verschelde, Jan; Zhuang, Yan (2006). "Parallel Homotopy Algorithms to Solve Polynomial Systems". *Proceedings of ICMS 2006*.

[8] Gruman, Galen (2008-04-07). "What cloud computing really means". *InfoWorld*. Retrieved 2009-06-02.

[9] Hiremath. Shruthi and Chandra.Pallavi, *Efficient Clustering Algorithm for Storage Optimization in the Cloud*, VIT University: Vellore, Tamilnadu, 2013.

[10] Dean. Jeffrey and Ghemawat Sanjay, *MapReduce: Simplified Data Processing on Large Clusters*, USENIX Association: Google, Inc, 2004.

[11] Ha Lee. Kyong, Joon Lee.Yoon, Choi.Hyunsik, Chung. Yon Dohn and Moon.Bongki, *Parallel Data Processing with MapReduce: A Survey*, KAIST: SIGMOD Record, December 2011, Vol. 40, No. 4.