Cluster Based Resource Allocation Using K-Medoid Clustering Algorithm

D. Arul Selve[#1], K. Kavitha[*2]

[#1] *M.Phil Research Scholar, Department of Computer Science, Mother Teresa Women's University*

[*2] *Assistant Professor, Department of Computer Science, Mother Teresa Women's University Kodaikanal, India*

Abstract — Map Reduce was proposed to make simpler the parallel meting out using a distributed computing platform that offers only two interfaces. Map Reduce has come out as a significant model for processing data in huge information centers. Map Reduce is a three phase algorithm consisting of Map, Shuffle and Reduce phases. Due to its extensive deployment, there have been numerous recent papers exactness practical schemes to get better the performance of Map Reduce systems. All these hard work focuses on one of the three phases to obtain performance enhancement.

To reduce network traffic within a Map Reduce employment, we deem to aggregate data to send them to distant reduce tasks with same keys. In existing system, a decomposition-based distributed algorithm and online algorithm is used to deal with the large-scale optimization problem and aggregation of data in a dynamic manner respectively. This paper mainly focus in detail on the system process of implementing Partitioning cluster based resource allocation using K- medoid clustering algorithm in decomposition based distribution algorithm. The common realization of k-medoid clustering is the Partitioning Around Medoids (PAM).

Keywords — *Map Reduce, Big Data, Data partition, Aggregation, K-Medoid Clustering*

I. INTRODUCTION

An epoch of Big Data has been evolved. Huge data is for the most part accretion of information sets so widespread and multifaceted that it is remarkably hard to handle them utilizing close by database admin devices. The principle challenges with big databases include inquiry, creation, examination, sharing and perception and stockpiling. As a matter of first significance, information is procured from assorted sources, for example, online networking, customary sensor information or undertaking information and so forth. Flume can be utilized to safe information from online networking. At that point, this information can be collected utilizing conveyed deed frameworks, for example, Google File System. These frameworks are very capable when number of peruses are high when contrasted with composes. Finally, information is dissected exploits map reducer with the goal that study can be keep running on this information proficiently and effectively.

Map Reduce has emerged as an important data processing model in data centers. Map Reduce is used in several applications including probing the web, URL frequency evaluation and indexing. In a typical application, the data on which Map Reduce operates is partitioned into chunks and assigned to dissimilar processors. Map Reduce is a three step procedure:

1) In Map phase, numerous parallel tasks are created to function on the pertinent data chunks to make transitional results. These results are stored in the type of key–value pairs.

2) In the Shuffle phase, the biased computation consequences of the map phase are transferred to the processors performing the reduce process.

3) In Reduce phase, each processor that carry out the reduce task aggregates the tuples make in the map phase.

A common obligation for routine data center needs is fast response time. In a huge data center where there are several Map Reduce jobs that run in tandem, a centralized master coordinates the obligation and scheduling of Map Reduce errands across the data center. The obligation problem is to choose which processor will execute a map or reduce task and the scheduling is to fix on in what order the tasks will be run on every processor.

The problem arise here is the network traffic happens during shuffle phase. To reduce this traffic an efficient traffic aware partition scheme is used along with data aggregation. To further reduce the network traffic and cost K –MEDOID algorithm is used in the above process. The remaining paper is ordered as follows. In section II, review recent related work. Section III presents a system model. IV presents the system process. Finally, section V presents the conclusion of the paper.

II. RELATED WORK

Huan Ke, Peng Ji, Song Guo, Miniyi Guo, have proposed a Decomposition based Distribution algorithm and an online algorithm in order to overcome the network traffic limitations and enhance network traffic by parallel approach and aggregation [1].

P.Costa, A.Donnelly, A.I. Rowstron & G.O'Shea have proposed a Map Reduce-like system to decrease the traffic by pushing aggregation from the edge into the network. However, it can be only applied

to the network topology with servers directly linked to other servers, which is of limited practical use [2].

Y. Dinesh Reddy & A. Pio Sajin, have proposed an efficient approach traffic aware partition and aggregation using K-means algorithm [3].

Condie et al. have introduced a combiner function that reduces the amount of data to be shuffled and merged to reduce tasks [4].

J. Lin and C.Dyer, have proposed an in-mapper combining scheme by develop the fact that mappers can preserve state across the processing of multiple input key/value pairs and defer release of intermediate data until all input records have been processed. Both proposals are inhibited to a single map task; disregard the data aggregation opportunities from multiple map tasks [5].

III.SYSTEM MODEL

Map Reduce is the programming pattern for scalability of massive data across many Hadoop Clusters. Map Reduce actually have two separate and distinct tasks. The first is the map job, which takes a set of data and individual elements in it are broken down into small chunks (key/value pairs). The reduce job combines those data chunks into a smaller set of chunks. The sequence of the name Map Reduce itself implies, the reduce job is always performed after the map job.

Map Reduce algorithm generally includes three phases: Map Phase, Shuffle Phase, and Reduce Phase.

The system model also includes the distributed algorithm to solve the problem on multiple machines in a parallel manner. The basic idea of this algorithm is to decompose the original large-scale problem into several distributive solvable sub problems that are coordinated by a high-level master problem.

It also includes an online algorithm which dynamically adjusts data partition and aggregation during the execution of map and reduces tasks**.**

K- MEDOID algorithm is also used which is commonly known as PAM algorithm. PAM stands for "Partition Around Medoids". The algorithm is intended to find a sequence of objects called *Medoids* that are located in centre of the clusters. Objects that are tentatively defined as Medoids are placed into a set S of *selected objects*.

IV.SYSTEM PROCESS

The system process is first started with the architecture of the system which includes the stages in the system followed by the detailed description of the stages.

System architecture includes Data Uploading, Significance of Shuffle Process, Segmentation by K-Medoid Clustering, Intermediate Results, Task Assignment, Scheduling Map-Reduce Task, Data loading and Processing of Task.

Fig 1: System Architecture

A. Data uploading:

Based on data locality, the map tasks in the workers are scheduled by master. The output of the map task is divided into as many partitions based on the number of reducers for the job. The same transitional key should be assigned to the same detachment which assures the correctness of the carry out. The intermediate key/value pairs are sorted. The worker receives the sorted pairs with the corresponding reduce task to be executed. Data locality constraint are not taken into consideration for scheduling of reduce task. The amount of data is significant that has to be transferred through the network in the shuffle process.

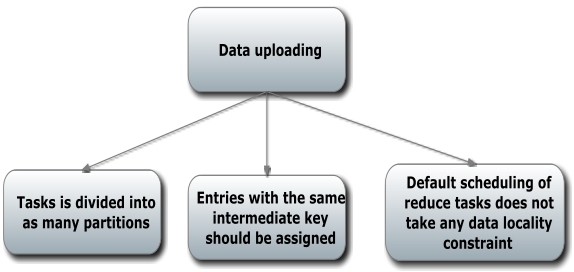

Fig 2: Data Uploading

B. Clustering:

To cluster the uploaded data partition based clustering is used. K-MEDOID algorithm which is commonly known as PAM algorithm is implemented

which is a development of K-Means algorithm. K-Medoid or PAM clustering is tougher to noise and outliers as compared to K-Means as it reduces a sum of pair wise difference instead of a sum of squared Euclidean distances. By clustering based on attribute before task assignment make the jobber easier to assign the task and reduce the time and cost for assignment.

. Euclidean distance or Euclidean metric is the "regular" space between two points in Euclidean space. Euclidean space becomes a metric space, with this distance. The norm associated with it is called the Euclidean norm. In our dataset, the latitude and longitude of place to find the distance between locations are taken.

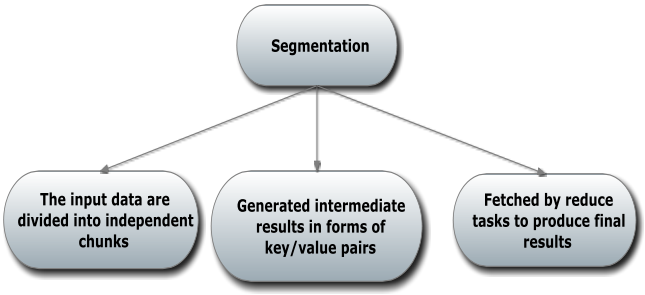

Fig 3: Segmentation- K-Medoid Clustering

C. Task Assignment:

The task assignment is made in access tier. The access tier is composed of cost-effective Ethernet switches connecting rack VMs. The access switches are associated through Ethernet to a set of aggregation switches. In turn they are connected to a layer of core switches. An inter-rack link is the most disputable resource as all the VMs hosted on a rack transfer data across the link to the VMs on other racks. VMs in our work, are distributed in three different racks, and the map-reduce tasks are scheduled.

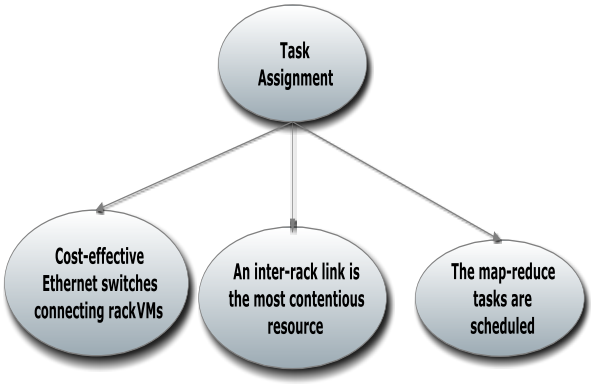

Fig 4: Task Assignment

D. Data loading:

The highest network traffic is achieved when there is only one reduce task under all algorithms is observed, since all key/value pairs may be delivered to the only reducer that locates far away, leading to a large amount of network traffic due to the many-to-one communication outline. As the number of reduce tasks increases, the network traffic decreases because more reduce tasks share the load of intermediate data. i.e., number of reduce task is inversely proportional to the network traffic.

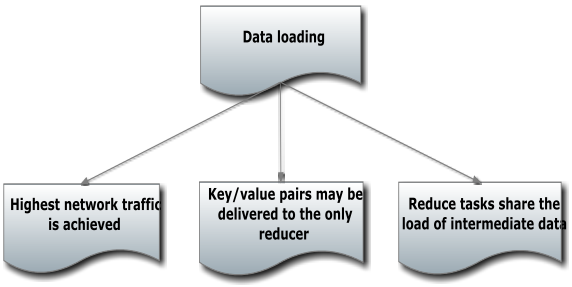

Fig 5: Data Loading

E. Processing of Task:

The accomplishment of a Map Reduce job into several time slots with a length of several minutes or an hour is divided by the system. The strictures are collected at time slot t with no supposition about their distributions. To reduce traffic cost, it needs to transfer an aggregator from one machine to another with some relocation cost. Meanwhile, the key assignment among reducers is attuned. The process let reducer to process the data with a key instead of reducer which is currently in charge of this key by using a function to denote the cost migration of all intermediate data received by reducers so far.

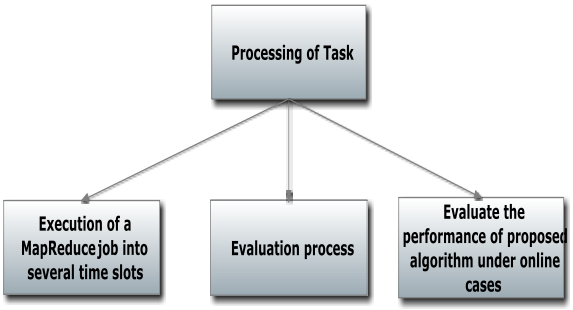

Fig 6: Processing of Task

V. Conclusions

This paper mainly focuses in detail about system process of implementing K-MEDOID clustering algorithm in Map Reduce aggregation and optimization of intermediate data partition to minimize network traffic cost for big data applications. This paper, use distributed algorithm to solve the problem on multiple machines in order to deal with the large-scale formulation due to big data. It also

extends the process to handle the Map Reduce job in an online manner even though some system parameters are not given.

REFERENCES

[1] Huan Ke, Peng Li, Song Guo, Minyi Guo, "On Traffic Aware Partition and Aggregation in Map Reduce for Big Data Applications". IEEE Transactions on Parallel and Distributed Systems. Citation Information: DOI 10.1109/ TPDS.2015.2419671

[2] P. Costa, A. Donnelly, A. I. Rowstron, and G. O'Shea, "Camdoop: Exploiting in-network aggregation for big data applications." In NSDI, vol. 12, 2012, pp. 3–3.

[3] Y. Dinesh Reddy and A. Pio Sajin , " An Efficient Traffic-Aware Partition and Aggregation for Big Data Applications using Map-Reduce". Indian Journal of Science and Technology, Vol 9(10), DOI: 10.17485/ijst/2016/v9i10/88981, March 2016

[4] T. Condie, N. Conway, P. Alvaro, J. M. Hellerstein, K. Elmeleegy, and R. Sears, "Map reduce online." in NSDI, vol. 10, no. 4, 2010, p. 20.

[5] J. Lin and C. Dyer, "Data-intensive text processing with map reduce," Synthesis Lectures on Human Language Technologies, vol. 3, no. 1, pp. 1–177, 2010.