Fine tuning of MapReduce jobs using parallel K Map clustering

[1] Divya Mishra, [2] Dr Suryakant yadav
[1]Noida International University, Greater Noida, India,
[2] Noida International University, Greater Noida, India.

Abstract: Advancement and development of information technology has led to huge growth in data, which pose huge challenge into data storage and analysis to conclude meaningful information from data. Size of data ranges from petabyte or exabyte range, to mine information out of large data set will increase the computing capacity. Thus to address and manage this high velocity data growth an advanced processing algorithms and methods are required for data analysis. To achieve same MapReduce word count program and P-KMeans a parallel clustering algorithm used in Hadoop .Through experiment, it has been concluded that execution time reduce when nodes count increases in a cluster, but also some of the important points has been observed while conducting experiment. Various performance change, and plotted results on different performance charts. This paper aims to study MapReduce applications and its performance verification and improvement recorded for Parallel KMeans algorithm on four nodes Hadoop Cluster.

Index Terms **- Hadoop, Big data, Data Clustering, K-Means algorithm Parallel Computing, Map-Reduce, word count.**

I. INTRODUCTION

Traditional data storage and processing capabilities are limited and was reliant on available infrastructure processing capability, storage and processing requirements, which deemed to be very different from today. Thus, those approaches and databases are facing Sevier technical challenges to accommodate fast growing Big Data storage and processing demands.

Current rapid advancement and ongoing progress in the social media, robotics, web, healthcare, mobile devices and research area are producing data, which is growing exponentially, and processing such data becomes a huge challenge. How to accommodate and handle these enormous collections of data, as well as further churn out meaningful information, has become a problematic issue. Business decision are result of detailed analysis of a vast and verity of big data related to the industry or organization. Therefore, to bring meaningful insight from data require efficient data analysis methods.

In distributed environment data mining unearth data relationship by using best practices of artificial intelligence, relationship databases etc. Besides analytics, also helps in data management and data modeling. Distributed Computing is a methodology, which resolves data processing and storage challenge by sharing the CPU processing of network systems. Every independent machine on network is known as node and cluster is a collection of nodes in a network. Apache Hadoop [1] is a distributed open source framework, which provides parallel and fault tolerant and scalable processing capability to work on big data. It was developed by Google's MapReduce and Google File Systems which was later adopted by Apache. It is suitable for applications, which produces huge and verity of data and desires quick results for data analysis and decision-making purpose by harnessing ability of Hadoop MapReduce parallel execution on distributed environment. It follows a Map and Reduce programming methodology by splitting of data and this data partition is then stored on various nodes of distributed network for parallel processing. Interim output chunks is then combined as result. Hadoop acts as a file system for the storage and organization of output data like in HDFS System. The Hadoop's fault tolerant ability automatically take care of node failure, and re-runs the failed task on idle or underutilized node. In an era of high demand to mine, fast growing big data, which requires fast and efficient information retrieval on a distributed environment of Hadoop. This minimizes execution time to complete assigned task but also mitigates the individual machine need to process hue amount of data. Google File Systems [4] and MapReduce concept gave power to process big data without worrying scale of data analytics to be done by providing ery convenient processing.

This has provided great data exploration and processing by exploring structured and unstructured data .There are dedicated research and specialized plug in software's are available e.g. Hive[5] , Zookeeper [6] ,Pig[7] which are suitable for certain type of processing requirements .The MapReduce offers tremendous processing flexibility to execute a task on a distributed manner with ease and quickly. Unstructured data analysis is one of the prominent challenges, which requires implementation of critical algorithms. The Hadoop is capable of storing and process many thousands of Exabyte of data. This is done by breaking data into small chunks and then combines the result. This facilitates fast workload sharing by all the nodes of network and hence increase the huge data execution and enhances overall performance of the network to provide faster data analytics.

This paper discusses and talks about the K-Means Algorithm design introduction and application of K- Means Clustering .Experiment results on a difference in MapReduce and K Means based map reduce.

II. MAPREDUCE- PROCESSING OF BIGDATA

In the MapReduce paradigm, MapReduce [2] has become the prominent batch-processing model. MapReduce is a very flexible programming method able to parallel process large volume efficiently on large number of computing nodes.
MapReduce methodology consisting of map and reduce functions, which allows application developers to customize them as per their requirement. Along with providing faster processing Map Reduce manages node failure and abstracting design complexity from the programmer.

MapReduce has been developed for effective parallel processing of big data by breaking the load into independent sub units. Prominent benefit of MapReduce is that it conceals system level implementations and complexities from developer by allowing developer just to focus on its objective. MapReduce is a two-phase approach: Map and Reduce. Every Map phase takes input files present into various nodes of distributed file system. If Map function does, co- located with data partition system will attempt to move data portion to node where Map function exists with data portion to reduce data movement. Hence MapReduce backs concept of as "Moving data closer to compute" to optimize execution time. Post Map function finishes its processing, reduce function is then further run on all values having same interim key value and output key/value pairs as the result. The MapReduce framework based on master/slave architecture having a one node acting as master has Job Tracker and several slave nodes, which runs assigned task, and their status is maintained to Tasktrackers, one per node in the cluster. The Job Tracker acts as communicator between users who submits the job and the underline pool of hardware. Job Tracker accepts users assigned map/reduce jobs to the and further jobs are executed on sequence of submission first come/first-served basis. The Job Tracker does resource management required for map, reduce steps, allocated, and monitor slave nodes executing allocated task via tasktrackers. The Tasktrackers accepts and run tasks allocated by JobTracker, manage data movement among the map, and reduce step.

Below diagram Figure1, elaborates word count example of MapReduce processing.

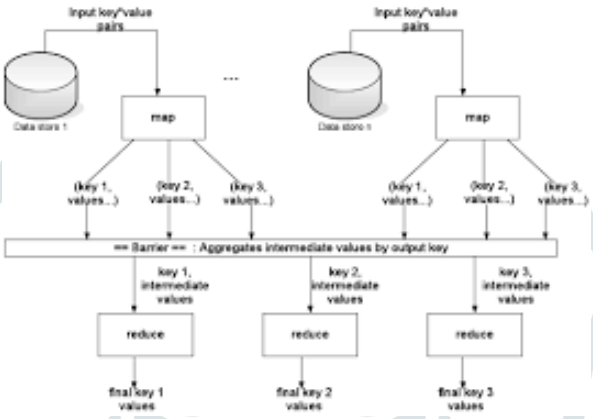


Fig.1 MapReduce process flow

In a homogeneous environment where all nodes are considered as possessing similar computing power and storage capacity. In case of node failure, MapReduce attempts to reruns failed tasks on a different idle/less-occupied node. There may be case when a node is processing a task slowly is termed as straggler; Map Reduce then kick off a speculative copy of straggler task ("backup task") on another node, which can finish task quickly. In absence of speculative execution, a job will be very slow and run for longer downgrading the performance. Stragglers can arise due to numerous reasons due to faulty hardware and in appropriate configuration. It is proved by various researchers that speculative execution can increase job response times by 44%.In Heterogeneous environment this problem is deteriorate the execution of speculative execution due to difference in nodes storage and processing capability.

In a homogeneous environment, all machines have equal computing speed and disk capacity. In case of node, failure happens MapReduce attempts to reruns failed tasks on a different node, which is idle/less-occupied at that moment. In some instances a node is executing a task very poorly or slowly known as straggler task, in that case Map Reduce kick-off a speculative copy of straggler task ("backup task") on another node, which can finish task quickly job will be as slow as the misbehaving task ,if no speculative task is there. Straggling tasks can also be triggered due to faulty hardware and misconfiguration Google [8] .It has been observed that speculative execution can improve job processing by 44%.In Heterogeneous environment lack of proper mechanism to address speculative task may increase processing time due to difference in nodes storage and processing capability.

III. HDFS

The HDFS or Hadoop distributed file system is capable of storing huge data sets and ensures data integrity and reliability. A cluster is formed by networking of large numbers of nodes consisting of both host and Client and they are associated with storage devices and executes user defined programs or application. Hadoop provides a distributed file system environment to a provide analysis on huge data sets using the MapReduce paradigm.

Iv. CLUSTERING OF DATA

Clustering is an activity of grouping of similar kind of objects in groups. The main objective of clustering is to maximize the intra-cluster similarities. The classification is dependent on appropriate attribute selection and must have a relationship between fields of user concerns. An appropriate Classification discover a relation between various objects of semi structured or unstructured data .There are various clustering algorithms which are used to group data sets based on attributes of user interest .Key parameters to perform clustering depends upon computing distance, density and distribution of data. Based on the data extraction requirement and data sets type an appropriate clustering algorithm is selected to extract data from them. Groups are divided based on similarity and then assigned certain labels useful for identity.

There are many methods to partition data in groups .There are different algorithms are applied on each group based on the properties and data distribution. Key clustering models described as below.

A. CENTROID BASED CLUSTERING

In this type of partition method, attempt to identify each cluster range of values. When there is minimal difference between object and other cluster object become part of that cluster Number of cluster are pre-determined, and is one the biggest challenge with this clustering algorithm.

B. DISTRIBUTION BASED CLUSTERING

In this, type of partition method also an attempt to identify each cluster range of values. When there is minimal difference between object and other cluster object become part of that cluster Number of cluster are pre-determined, and is one the biggest challenge with this clustering algorithm.

C. CONNECCTIVITY BASED CLUSTERING

In this approach, every entity is related to its neighbors and the power of relationship among them depending upon the magnitude of that relationship on the distance between them. Hence, objects create with clusters near.

D. DENSITY BASED CLUSTERING

Clusters in these algorithms are created based on how dense members of a data set are, in a dedicated location, .Performance degradation has been observed using this clustering method.

V. K-MEANS CLUSTERING

This is a very important data analysis technique. It has been proved that huge data set of can be processed efficiently by this kind of analysis, giving optimized results for many distinct types of data in different scenarios.

K-Means Clustering is an approach designed to reorganize data into unstructured data sets or semi structured data sets. Ability to manage huge data and in simplistic manner makes this k-Means very popular and successful method to arrange data. It takes clusters number and the initial set of centroid as input parameters. Further distance of each object from centroid is calculated in data set of the respective cluster. The object with closer distance assigned to the cluster. Using K Means, clustering deriving the distance from chosen mean to a point is very popular and key method used to group the items of a data set .Distance calculated by Euclidean distance a very popular parameter for comparison of points.

In K means clustering, cluster centroid is an essential metric, which is a point whose coordinates coincides to the mean of all points of cluster. There is possibility that certain items that may or may not be associated to any cluster and thus cannot be specified within dataset are known as outliers. Algorithms focus to calculate a minimum squared difference between datasets objects and cluster centroid.

Key points of algorithm are described below in length:

- ‘K’ represents number of cluster, which must be decided well sin advance.
- Derive distance of each centroid from for each of the selected set of K clusters.
- Consider each object of the given set and compute its distance from centroids of the all K clusters. This distance will determine which closest cluster object will be added.
- Recalculate cluster centroids after each allocation or a set of assignments.
- This is a recursive process and updated regularly.

VI. K-MEANS CLUSTERING USING MAPREDUCE

K-MEANS is an unsupervised algorithm used for clustering. The K-Means algorithm uses cluster mean to break the data into different k clusters in such manner that each clusters is having high characteristic similarities K-Means clustering is an iterative algorithm.

The K-means clustering algorithm is very effective and produces great results in clustering large data sets. The K-Means algorithm partition given objects based on their nature into k clusters, where k is a pre-defined cluster number Main exercise is to define k centroids, each cluster having one centroid. Centroid of a cluster is computed in such a manner that it is closely related to rest of the cluster object.

Key approach to design MapReduce process for K-means is two-phase method. First step computes distance of all points in cluster from its mean and second phase recalculates the fresh means from centroids. These two phases acts like Map and reduce functions of MapReduce.

In order to define Map and Reduce programs two files are essential one holds information of clusters and their centroids while other file contains the data points to be clustered. Further Euclidian distance is calculated from cluster centroids. This will enable K means clustering of data by programming using Map and Reduce routines as per requirement.

$$d(P,Q) = \sqrt{((x1(P)-x1(Q))^2 + (x2(P)-x2(Q))^2 + ...)}$$
$$= \sqrt{\left(\sum_{j=1}^{p} (xj(P)-xj(Q))^2\right)}$$

After identifying input, output and functionality of the Map and Reduce routines. Using K-Means algorithms Map and Reduce programs are designed as below.

Algorithm 1 Mapper design for K-Means Clustering

```
0: procedure KMEANMAPDESIGN
0:    LOAD Cluster file
0:    fp = Mapclusterfile
0:    Create two list
0:    listnew = listold
0:    CALL read (Mapclusterfile)
0:    newfp = MapCluster()
0:    dv =0
0:    Assign correct centeroid
0:    read(dv)
0:    calculate centerorid
0:    dv = minCenter()
0:    CALL KmeansReduce()
0: end procedure=0
```

Algorithm 2 Reducer design for K-Means Clustering

```
0: procedure KMEANREDUCEDESIGN
0:    NEW ListofClusters
0:    COMBINE resultant clusters from MAP CLASS.
0:    if cluster size too high or too low then
      RESIZE the cluster
0:       C_Max = findMaxSize(ListofClusters)
0:       C_min = findMinSize(ListofClusters)
0:       if C_max > (1/20) totalSize then Resize(cluster)
0:          WRITE cluster FILE to output DIRECTORY.
0:
```

Algorithm 3 Implementing KMeans Function

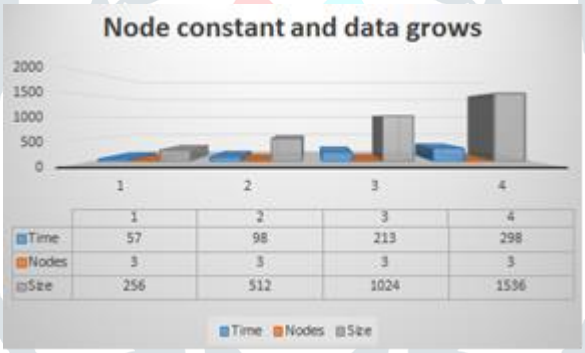
```
0: procedure KMEANS FUNCTION
0:   if Initial Iteration then LOAD cluster file from DIREC
     TORY
0:   elseREAD cluster file from previous iteration
0:     Create new JOB
0:     SET MAPPER to map class defined
0:     SET REDUCER to reduce class define
0:     paths for output directory
0:     SUBMIT JOB
0:
```

VII. THE EXPERIMENTAL SETUP

Experimental set-up has used three nodes connected via a LAN and switch. One of the Nodes acts as a Master node, which monitors data and tasks execution along with node failure handling. Rest of the cluster nodes are slaves' nodes. All nodes have are of same configuration and have Intel – Core 3 processor. Maximum five nodes cluster can be prepared over a private LAN network. Hadoop software installed by downloading Apache from Apache Hadoop official website.

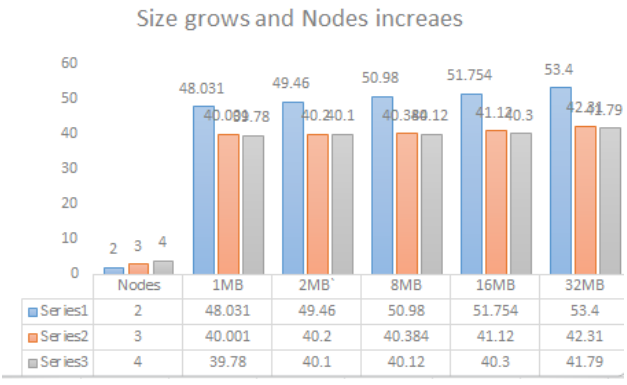
Grep operation was executed on different scenarios using parallel K-Means.

Scenario 1: When nodes are constant and data set size grows.



	1	2	3	4
Time	57	98	213	298
Nodes	3	3	3	3
Size	256	512	1024	1536

Fig.2 Performance when nodes are same, file data size grows



	Nodes	1MB	2MB	8MB	16MB	32MB
Series1	2	48.031	49.46	50.98	51.754	53.4
Series2	3	40.001	40.2	40.384	41.12	42.31
Series3	4	39.78	40.1	40.12	40.3	41.79

Fig.3 Performance when nodes increases along with file size

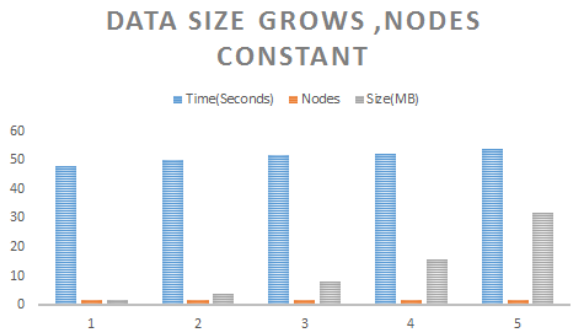


Fig.4 Results of PK-Means MapReduce Application

VIII. CONCLUSION AND FUTURE WORK

The amount of data generated in todays' technical era is enormous and require great data processing. This paper discuss about K-Means Clustering Algorithm implementation on a distributed network. K-Means algorithm is famous for efficiently arranging data of similar attributes and minimizes resource procurement and operational costs of executing huge volumes of data. The amount of information consumed and generated in today is growing rapidly and lead to discover necessity of processing big data sets. In order to meet rapidly growing data processing demand it was essential to deploy algorithms for data processing over a distributed environment to reduce operational and development cost along with ensuring resource utilization by improving execution speed. There is always scope of improvement in any to increase response and accuracy. Research work is going on since 1970; the first implementation of K-Means was implemented. Identification of initial set of centroids is an area of improvement in the K-Means Algorithm. Along with above discussed challenges few small issues are also worthy to be talked about. For instance, selecting appropriate set of initial centroids minimizes iterations number required to complete a task .Other issue is to scale algorithm to harness processing of big data, parallel implementation of K-Means Algorithm is a future research area to address above highlighted issue.

REFERENCES

[1] Dean, J., Ghemawat,:MapReduce: Simplified data processing on large clusters. 2004

[2] HadoopMapReduce, http://hadoop.apache.org/mapreduce/

[3] H Hu, Y Wen, T Chua, X Li, "Towards Scalable Systems for Big Data Analytics: IEEE 2014.

[4] Kyong-Ha Lee, Yoon-Joon Lee, Hyunsik Choi, Yon Dohn Chung, Bongki Moon, "Parallel Data Processing with MapReduce: A Surver", University of Arizona Deparment of Computer Science, 2011.

[5] Anjan K Koundinya, Srinath N K, A K Sharma, Kiran Kumar:,:Map/Reduce Design and Implementation of Apriori Algorith for handling Voluminous Data-Sets, ACIJ 2012

[6] J. Ekanayake, H. Li, B. Zhang, T. Gunarathne, S.H. Bae, J. Qiu, and G. Fox. Twister: A runtime for iterative mapreduce. In Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing, pages 810-818. ACM, 2010.

[7] D. Kondo, B. Javadi, P. Malecot, F. Cappello, and D. P. Anderson. Cost-benefit analysis of cloud computing versus desktopgrids. InIPDPS'09-Proceedingsofthe2009IEEE International Symposium on Parallel&Distributed Processing, 2009. IEEE Computer Society.

[8] Dhruba Borthakur, "The Hadoop Distributed File System: Architecture and Design", A*pache Software Foundation*, 2007.

[9] https://stanford.edu/~rezab/classes/cme323/S16/projects_reports/bodoia.pdf

[10] http://ai2-s2-pdfs.s3.amazonaws.com/23db/2d9c5a97f36f1b63ea249402b4be0919ebc9.pdf

[11] C. Ranger, R. Raghuraman, A. Penmetsa, G. Bradski, and C.Kozyrakis. Evaluating mapreduce for multi-core and multiprocessor systems. IEEE 2007

[12] P. Riteau, K. Keahey, and C. Morin. Bringing elastic mapreduce to scientific clouds. Proceedings of Cloud Computing and Its Applications 2011.

[13] M. C. Schatz. Cloudburst: highly sensitive read mapping with mapreduce. Bioinformatics, 2009.

[14] J.Xie,S.Yin,X.Ruan,Z.Ding,Y.Tian,J.Majors,A.Manzanares, and X. Qin. Improving mapreduce performance through data placement in heterogeneous hadoop clusters. IEEE 2010