IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 1, Ver. IV (Jan – Feb. 2016), PP 23-29
www.iosrjournals.org

A Review: Hadoop Storage and Clustering Algorithms

Latika Kakkar[1], Gaurav Mehta[2]
[1](Department of Computer Science and Engineering, Chitkara University, India)
[2](Department of Computer Science and Engineering, Chitkara University, India)

Abstract *: In the last few years there has been voluminous increase in the storing and processing of data, which require convincing speed and also requirement of storage space. Big data is a defined as large, diverse and complex data sets which has issues of storage, analysis and visualization for processing results. Four characteristics of Big data which are–Volume, Value, Variety and Velocity makes it difficult for traditional systems to process the big data. Apache Hadoop is an auspicious software framework that develops applications that process huge amounts of data in parallel with large clusters of commodity hardware in a fault-tolerant and veracious manner. Various performance metrics such as reliability, fault tolerance, accuracy, confidentiality and security are improved as with the use of Hadoop. Hadoop MapReduce is an effective Computation Model for processing large data on distributed data clusters such as Clouds. We first introduce the general idea of big data and then review related technologies, such as could computing and Hadoop. Various clustering techniques are also analyzed based on parameters like numbers of clusters, size of clusters, type of dataset and noise.*
Keywords*: Big data, Cloud Computing, Clusters, Hadoop , HDFS, MapReduce*

I. Introduction

To analyze data from different aspects and to outline and sum up this data into valuable data i.e. important information is known as Data Mining. Data Mining is process of analyzing correlations and patterns among huge data in a database. Major data mining techniques involve classification, regression and clustering. Clustering is the process of assigning data into groups called clusters such that object in same cluster is more similar than the objects of other clusters. Clustering is a main task of Data Mining. It common technique for statistical data analysis used in many fields which includes pattern recognition, analysis of image, information retrieval. Now a day's big data is growing rapidly specially those related to internet companies. For example, Google, Facebook processes petabytes of data within a month.Figure1 shows the increase of the data volume in a rapid manner. There are various challenges as the datasets are increasing in a drastic manner. Due to advancement of information technology huge amount of data can be generated. On an average, YouTube uploads 72 hours of videos in every minute [1]. This creates the problem of collection and integration of tremendous data from widespread sources of data. The accelerated advancement of cloud computing and the Internet of Things further contributes the fine increase of data. Through this increase in volume and variety there is a great accentuation on the existing computing capacity. This huge data causes problem of storing and managing such large complex datasets with fulfilling the hardware and software infrastructure. The mining of such complex, heterogeneous and voluminous data at different levels of analyzing, perception, anticipating should be done carefully to unfold its natural properties so as to filter good decision making. [2]

II. Challenges Of Big Data

2.1 Confidentiality of Data

Various big data service providers use different tools to process and analyze data, due to which there are security risks. For example, the transactional dataset generally includes a set of complete operating data to execute the important business processes. Such data consists of lower granularity and delicate information like credit card numbers. Therefore, preventive measures are taken to protect such sensitive data, to ensure its safety.

2.2 Representation of Data

The datasets are heterogeneous in nature. Representation of Data aims to make meaningful data for analysis of computers and user comprehension. Improper representation of data affects the data analysis. Efficient data representation shows data structure, class and technologies which enables useful operations on different datasets.

2.3 Management of Data Life Cycle

The storage system used normally cannot support huge quantity of data. Data freshness is the key component which describes the values private in big data. Therefore, to decide which data shall be stored and which data shall be ignored, a principle associated with the analytical value should be developed.

2.4 Analytical Mechanism:

The analytical system processes large amount of heterogeneous data with limited time period. However, the traditional RDBMSs lack scalability and expandability, by which performance requirement is affected. Non-relational databases are advantageous in the analysis of unstructured data but there are still some problems in performance of non-relational databases. Few enterprises have employed hybrid architecture of database that has advantages of both types of databases e.g. Facebook and Taobao.

2.5 Energy Management

Large amount of electrical energy is consumed for the analysis, storage, and transmission of big data. Therefore, power-consumption control and management mechanism of system or computer were required for big data with the constraints that the expansion and accessibility are ensured.

2.6 Scalability

The analytical system designed for big data should be compatible with all types of datasets. It must be able to process drastically increasing complex datasets.

2.7 Reduction of Redundancy and Data Compression

Redundancy reduction and data compression is very useful to diminish the cost of the entire system on the basis that there is no affect on the data. For example, mostly the data developed by sensor networks have high degree of redundancy that may be compressed and filtered. [3-5]

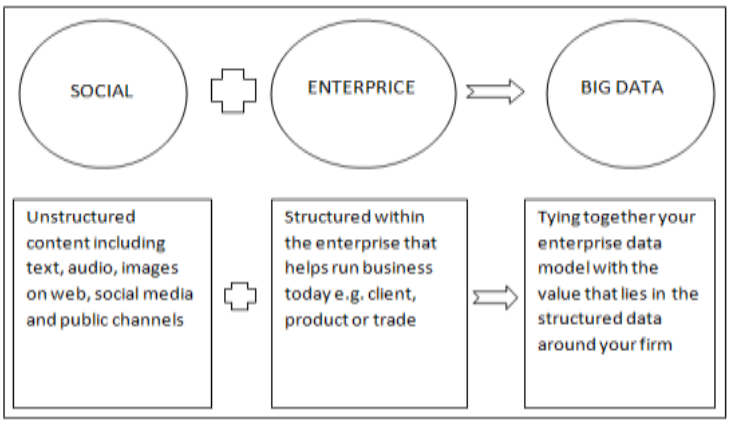

Fig. 1 Composition of Big Data

III. Related Technologies

Fundamental technologies that are closely related to big data:

3.1Cloud Computing

Cloud computing refers to parallel and distributed system that consists of computers that is visualized and inter-connected. This system is represented as a computing resource based on service agreements established through intervention between the customers and the service providers. Cloud computing enables the organizations to consume the resources as a utility rather than developing and maintaining the internal computing infrastructures.

Benefits of cloud computing:

- Pay per use: Users can pay only for the resources and workload they use which means that the computing resources are measured at a granular level.
- Self-service provisioning: The computing resources can be operated by end users for the workload on demand.
- Elasticity: With the increase and the decrease of the computing needs of the companies, they can scale up and scale down the resources.

3.2 Cloud Computing and Big Data Affiliation

Cloud computing and big data are closely related. The key components of cloud computing are shown in Fig.2. For the storage and processing of big data cloud computing is the solution. On the other hand, the big data also increases the development of cloud computing. The technology of cloud computing of distributed storage efficiently manages the big data and the parallel computing capability of cloud computing improves the big data analysis.

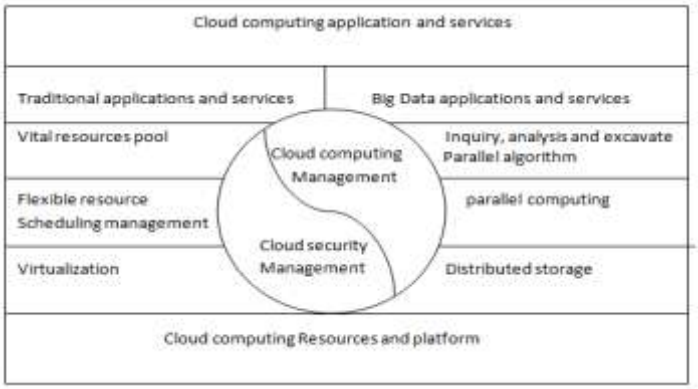

Fig. 2 Key components of cloud computing

3.3 Hadoop

The storage infrastructure has to provide two basic requirements. First is to provide service for information storage and secondly to provide interface for efficient query access and data analysis. Traditionally, structured RDBMSs were used for storing, managing and data analysis but with the increasing growth of big data, efficient and reliable storage mechanism is needed. Hadoop is a project of the Apache Software Foundation. Hadoop [6] is an open-source software framework implemented using Java and is designed to be used on large distributed systems of commodity hardware. The software framework of Hadoop is designed in such a way that it manages hardware failure automatically. Hadoop is very popular software tool due to it being open-source. Yahoo, Facebook, Twitter, LinkedIn, and others are currently using Hadoop. [8]

3.3.1 Hadoop Distributed File System (HDFS)

The Hadoop Distributed File System (HDFS) is the file system component of the Hadoop framework. Hadoop Distributed file System is used to store huge data on commodity hardware [6]. HDFS is designed for a large amount of big data files. The size of data to be stored using Hadoop Distributed file System could be sized from gigabytes to terabytes [7]. The property and feature of HDFS is that it can support millions of files and has scalability to hundreds of nodes. HDFS has separate storage places for metadata and application data [8]. The metadata is stored in the NameNode server and application data are stored on DataNodes which contain blocks of data. The communication between the different nodes of the HDFS is done using a TCP-based protocol [8]. Fault tolerance is the ability of a system to function adequately and correctly even after some system components gets failed. Hadoop and HDFS are highly fault tolerant as HDFS can be spread over multiple nodes that is cost effective. It duplicates the data so that if one copy of data is lost then still there are backup copies available. [9]

3.3.2 Mapreduce Processing Model

Hadoop MapReduce processes big data in parallel and provides output with efficient performance. Map-reduce consist of Map function and Reduce function. Map function executes filtering and sorting of large data sets. Reduce function performs the summary operation which combines the result and provides the enhanced output. Hadoop HDFS and Map-Reduce are delineated with the help of Google file system. Google File System (GFS) is developed by Google is a distributed file system that provide organized and adequate access to data using large clusters of commodity servers. Map phase: The Master node accepts the input and then divides a large problem is into smaller sub-problems. It then distributes these sub-problems among worker nodes in a multi-level tree structure. These sub-problems are then processed by the worker nodes which execute and sent the result back to the master node. Reduce phase: Reduce function combines the output of all sub-problems and collect it in master node and produces final output. Each map function is associated with a reduce function.

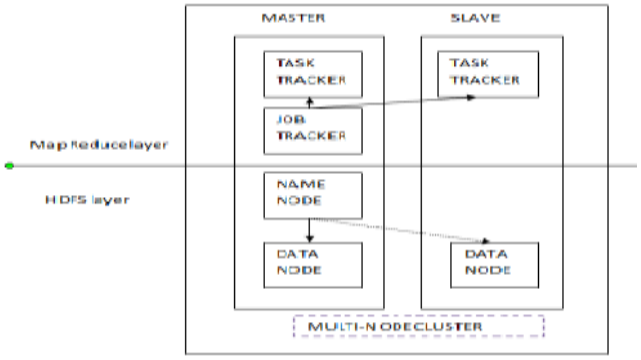

Fig 3 A Multinode Hadoop Cluster

Figure 3 shows a multi-node Hadoop cluster. It consists of a Master node and slave nodes. Task tracker and datanode operate on slave nodes. Master node consists of Task tracker, Datanode, Job tracker and Namenode. The Namenode is used to operate the file system. The job scheduling is executed by job tracker. The datanode stores the information which is in knowledge of the Namenode. On user interaction with Hadoop system, the Namenode becomes active and all the information about datanode data is processed by the Namenode.

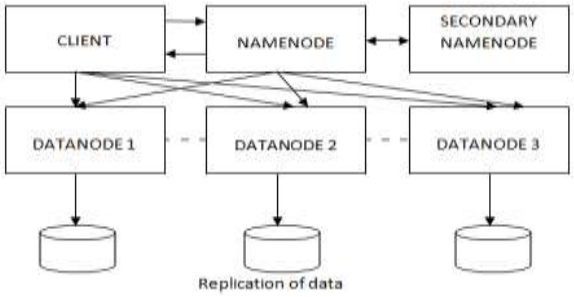

Fig.4 HDFS file system

Figure 4 shows the HDFS file system. Namenode is used to access and store user data. Thus Namenode becomes a single point of failure. For that a secondary Namenode is always active which takes snapshots from Namenode and stores all the information itself. On failure of the Namenode, all the data can be recovers from the secondary Namenode. Hadoops's HDFS can handle Fault Tolerance, but there are some other points of Hadoop and HDFS that could require some improvement. There is the problem of scalability of the NameNode [8]. The Namenode could subject to physical hardware constraints because Namenode is the single node.For example, because the namespace is kept in memory, there could be probability of HDFS to become unresponsive if the namespace grows to aa limit that it has to use node's memory. The solution to this issue is to use more than one Namenode but this can create problems in communication [8]. Another solution to this problem is to store only large data files in the HDFS. By using large data files the size of namespace will be reduced. Moreover the file system like HDFS is designed for large data files. Hadoop also lacks in providing good high level support. This issue can occur with open source software. Enhancing Hadoop's functionality on a system can be difficult without proper support [11]. HDFS is also sensitive to scheduling delays which restricts it to provide its full potential. Thus the node could have to wait for its new task. This issue is particularly found in HDFS client code [12].

4 Comparisons of Various Clustering Algorithms

4.1K-Means Algorithm

K-Means is a clustering algorithm that partitions objects into k clusters, where each cluster has a centre point called centroid clustering algorithm finds the desired number of distinct clusters and their centroids.
Algorithmic steps for K-Means clustering [21]
1) To find the value of K. K is the number of desired clusters.
2) Initialization – Now choose starting points that can be used to estimate centroids of clusters.
3) Classification – To check each object in the dataset and assigning it to the centroid nearest to it.
4) Calculation of centroid – After assignment of the objects in the cluster, new k centroids is calculated.
5) Meeting criteria – The steps 3 and 4 are repeated until the objects stop changing their cluster s and centroid of all clusters get fixed.

4.2 Fuzzy C-Mean

Algorithmic steps for Fuzzy C-Means clustering [22]

We let X = {x_1, x_2, x_3 ..., x_n} be the set of data points and V = {v_1, v_2, v_3 ..., v_c} be the set of centers.

1) Randomly *'c'* cluster centers are selected.

2) The fuzzy membership *'μ_{ij}'* is calculated using:

$$\mu_{ij} = 1 / \sum_{k=1}^{c} (d_{ij} / d_{ik})^{(2/m-1)}$$

3) Then fuzzy centers *'v_i'* are computed using:

$$v_j = (\sum_{i=1}^{n} (\mu_{ij})^m x_i) / (\sum_{i=1}^{n} (\mu_{ij})^m), \forall j = 1, 2, ...c$$

4) Steps 2and 3 are repeated until the minimum 'J' value is achieved or

$|| U^{(k+1)} - U^{(k)} || < \beta$, where, *'k'* is the iteration step, *'β'* = termination criterion between [0, 1], *'U'*= $(\mu_{ij})_{n*c}$' is the fuzzy membership matrix, *'J'* = objective function.

FCM iteratively changes the centers of clusters to exact location in a dataset. When fuzzy logic is introduced in K-Means clustering algorithm, it becomes Fuzzy C-Means algorithm. FCM clustering is based on fuzzy behavior. The structure of FCM is basically similar to K-Means. In [20] experimental results showed that K-Means takes less elapsed time than FCM. It can be concluded that in K-Means algorithm, the number of final clusters are to be defined beforehand. Such algorithms may be susceptible to local optima, can be outlier-sensitive and memory space required. The time-complexity of the K-Means algorithm is O(ncdi) and that of FCM algorithm is O(ndc2i). [20] Concluded that K-Means algorithm is better than FCM algorithm. Fuzzy c-means requires high computation time than K-Means because in FCM, fuzzy measures are also required to be calculated. FCM can handles issues related to pattern understanding, noise in data and thus provides faster solutions. FCM can be used for discovering association rules, retrieval of image and functional dependencies.

4.3 Hierarchical Clustering

Hierarchical clustering is a clustering algorithm which is used to build hierarchy of clusters. There are two strategies for hierarchical clustering:

Agglomerative: In Agglomerative each object initiates in its own cluster and cluster pairs are merged as we move up in the hierarchy. Agglomerative is bottom up approach.

Divisive: In divisive all the observations start in one cluster, and when we move downwards, splitting of cluster is performed iteratively. Divisive is top down approach. Dendogram is used to show results of hierarchical clustering. The complexity of agglomerative clustering is $O(n^3)$,thus in case of large datasets performance of agglomerative clustering is too slow. The complexity of divisive clustering is $O(2^n)$, which is more worse. Hierarchical clustering has limitations linear time complexity. These algorithms can also be used as a combined approach to produce better results. Agglomerative hierarchical clustering is used because of its quality and K-means is used because of its run-time efficiency. Comparative analysis of various clustering algorithms, K-means, Hierarchical, Self Organization map algorithm, Expectation Maximization algorithm and Fuzzy-c algorithm is shown in fig 5. [21, 22, 23, 24, 25]

4.4 Self-Organization Map (SOM)

The SOM [26] is self organization map is used to find better mapping from high dimensional input space to a two dimensional node representation. In SOM, object represented by same node are grouped in same cluster.

SOM algorithm pseudo code [26]:

1. Size and dimension of map is chosen.
2. With every new sample vector:

2.1 The distance between every codebook vector and new vector is computed

2.2 Using learning rate that decreases in time and distance radius on the map, all codebook vectors with new vector is computed.

Table 1. Comparative analysis of various clustering algorithms

Algorithms	SIZE OF DATASET	NO. OF CLUSTERS	TYPE OF DATASET	OUTLIERS
K-Means clustering algorithm	With large datasets, quality of algorithm is very good.	Performance is better as no of clusters increases but shows less accuracy	With random or ideal dataset results are intermediate	Very sensitive to noise
Hierarchical clustering algorithm	With small datasets, quality of algorithm is very good.	Performance is intermediate as no of clusters increases and has better accuracy	With random or ideal dataset results are better	Sensitive to noise
Self Organization map algorithm	With small datasets, quality of algorithm is good.	Performance is less as no of clusters increases but shows more accuracy to classify objects to clusters.	With random or ideal dataset results are better	Sensitive to noise
Expectation Maximization algorithm	With large datasets, quality of algorithm is very good.	Performance is better as no of clusters increases but shows less accuracy	With random or ideal dataset results are intermediate	Very sensitive to noise
Fuzzy-c algorithm	With huge datasets, quality of algorithm is average.	Intermediate Performance and intermediate accuracy	Average results	Very sensitive to noise

IV. Conclusion

For efficient processing of Big data, Hadoop proves to be a good software. HDFS is a very large distributed file system that uses commodity clusters and provides high throughput as well as fault tolerance. This is very useful property of Hadoop because in case of data loss or system crashes, it can be adverse to business with irreversible consequences. For efficient retrieval of data from Hadoop , comparative analysis of various clustering algorithm is made based on various input parameters (Table1).

References

[1] Mayer-Schönberger V, Cukier K, Big data: A revolution that will transform how we live, work, and think. Houghton Mifflin Harcourt, 2013.

[2] Chen M, Mao S, Liu Y, Big data: A survey: Mobile Networks and Applications, 19(2), 2014 Apr 1, 171-209.

[3] Labrinidis A, Jagadish HV, Challenges and opportunities with big data, Proceedings of the VLDB Endowment, 5(12), 2012 Aug 1,2032-3.

[4] Chaudhuri S, Dayal U, Narasayya V, An overview of business intelligence technology, Communications of the ACM,54(8), 2011 Aug 1,88-98.

[5] Agrawal D, Bernstein P, Bertino E, Davidson S, Dayal U, Franklin M, Gehrke J, Haas L, Halevy A, Han J, Jagadish HV, Challenges and Opportunities with big data, 2011-1.

[6] Apache Hadoop, http://hadoop.apache.org

[7] Borthakur D, The hadoop distributed file system: Architecture and design, Hadoop Project Website, 2007 Aug 11, 1.

[8] Shvachko K, Kuang H, Radia S, Chansler R, The hadoop distributed file system. In Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium ,2010 May 3, 1-10.

[9] Evans J. Fault Tolerance in Hadoop for Work Migration. CSCI B34 Survey Paper, 3(28),2011,11.

[10] Bessani AN, Cogo VV, Correia M, Costa P, Pasin M, Silva F, Arantes L, Marin O, Sens P, Sopena J. Making Hadoop MapReduce Byzantine Fault-Tolerant, DSN, Fast abstract, 2010.

[11] Wang F, Qiu J, Yang J, Dong B, Li X, Li Y. Hadoop high availability through metadata replication. In Proceedings of the first international workshop on Cloud data management, 2009 Nov 2, 37-44.

[12] Shafer J, Rixner S, Cox AL, The hadoop distributed filesystem: Balancing portability and performance, In Performance Analysis of Systems & Software (ISPASS), 2010 IEEE International Symposium, 2010 Mar 28,122-133.

[13] Dean J, Ghemawat S, MapReduce: simplified data processing on large clusters, Communications of the ACM. 51(1),2008 Jan 1,107-13.

[14] Xie J, Yin S, Ruan X, Ding Z, Tian Y, Majors J, Manzanares A, Qin X, Improving mapreduce performance through data placement in heterogeneous hadoop clusters, In Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE International Symposium, 2010 Apr 19, 1-9.

[15] Cattell R. Scalable SQL and NoSQL data stores , ACM SIGMOD Record, 39(4), 2011 May 6, 12-27.

[16] Chaiken R, Jenkins B, Larson PÅ, Ramsey B, Shakib D, Weaver S, Zhou J, SCOPE: easy and efficient parallel processing of massive data sets, Proceedings of the VLDB Endowment, 1(2), 2008 Aug 1, 1265-76.

[17] Grossman RL, Gu Y, Sabala M, Zhang W, Compute and storage clouds using wide area high performance networks, Future Generation Computer Systems, 25(2), 2009 Feb 28,179-83.

[18] Liu X, Han J, Zhong Y, Han C, He X, Implementing WebGIS on Hadoop: A case study of improving small file I/O performance on HDFS, In Cluster Computing and Workshops, CLUSTER '09. IEEE International Conference, 2009 Aug 31, 1-8.

[19] Grace RK, Manimegalai R, Kumar SS, Medical image retrieval system in grid using Hadoop framework, In Computational Science and Computational Intelligence (CSCI), International Conference, Vol. 1, 2014 Mar 10, 144-148.
[20] Ghosh S, Dubey SK, Comparative analysis of k-means and fuzzy c-means algorithms, International Journal of Advanced Computer Science and Applications,4(4),2013,34-9.
[21] Kanungo T, Mount DM, Netanyahu NS, Piatko CD, Silverman R, Wu AY, An efficient k-means clustering algorithm: Analysis and implementation, Pattern Analysis and Machine Intelligence, IEEE , 24(7), 2002Jul, 881-92.
[22] Rao VS, Vidyavathi DS. Comparative investigations and performance analysis of FCM and MFPCM algorithms on iris data, Indian Journal of Computer Science and Engineering,1(2),2010,145-51.
[23] Joshi A, Kaur R, A review: Comparative study of various clustering techniques in data mining, International Journal of Advanced Research in Computer Science and Software Engineering, 3(3), 2013 Mar, 55-7.
[24] Mingoti SA, Lima JO, Comparing SOM neural network with Fuzzy c-means, K-means and traditional hierarchical clustering algorithms, European Journal of Operational Research, 174(3), 2006 Nov 1, 1742-59.
[25] Steinbach M, Karypis G, Kumar V, A comparison of document clustering techniques, In KDD workshop on text mining, Vol. 400, No. 1, 2000 Aug 20,525-526.
[26] Abbas OA, Comparisons between Data Clustering Algorithms, Int. Arab J. Inf. Technol... 5(3), 2008 Jul 1,320-5.