Parallel Optimal Grid-Clustering algorithm exploration on MapReduce Framework

B. Hanmanthu
Assistant Professor, Dept.of CSE,
Kakatiya Institute of Technology & Science,
Warangal-15, Telangana, India

R. Rajesh
Assistant Professor, Dept.of CSE,
Vagdevi College of Engineering,
Warangal-15, Telangana, India

P. Niranjan, PhD
Professor &Head, Dept.of CSE,
Kakatiya Institute of Technology & Science
Warangal-15, Telangana, India

ABSTRACT

The MapReduce frame work is one which is proven that is as the best suitable framework which can be used to carry out Big data analytics. The big data analytics playing a vital role in real time data analysis applications. Where as in the conventional data mining techniques the clustering technique is proven as that the most useful technique for effective data analysis. From our literature review we found that there are no sufficient clustering techniques suitable for processing big data. Taking this as a disadvantage we are exploring the optimal grid clustering techniques for big data analysis using MapReduce architecture. The initial level experiments conducted using this proposed model is shown magnificent upshot.

Keywords

Clustering algorithm, Parallel OptiGrid, Data analytics

1. INTRODUCTION

The enlarge of data exponential over the last recent years has introduced a new domain in the field of information technology called Big Data. As the storage space capability of the datasets increases that stretches the limits of conventional data storage and processing systems is often directs to as Big Data. These data require to process and analyze such massive datasets has introduced a new form of data analytics is called Big Data Analytics. This process takes account of analyzing massive assess of data of a combination of types to make known hidden blueprint, unidentified association and other useful information. This doesn't just magically appears there's usually planned, methodical development process used to create it many organizations like business, financial, insurance, health, etc.,. are increasingly using these Big Data analytics to obtain improved insights into their businesses, to increase their revenue and profitability and gain competitive advantages over competitor organizations.

Big data is already early in the game to get better insights and it is already altering the methods of business decisions are made and it's still. Though, because big data exceeds the capacity and capabilities of conventional storage systems, reporting and analytics systems, it pressure to new problem solving approaches. With the convergence of foremost computing, analyzable database technologies, the different kinds of data like wireless data, mobility and social networking, these all different sources generated data can achievable now to bring combined and development big data in many gainful ways.

Big data solutions are one which is attempting to inexpensively figure out the challenges of tremendous and fast-growing data volumes and understand its potential analytical value. For example, tendency analytics allows you to figure out what happened, while root cause and predictive analytics enable understanding of why it happened and what is likely to happen in the future. Meanwhile, opportunity and innovative analytics can be applied to identifying chances and improving in the future.

How the big data has transformed customer IT, Simply by witnessing it is clear that the assure of big data in healthcare is immense (think Google, Face book and Apple's sites, which all rely on processing and transmitting massive amounts of data).

Characteristically Big Data can be defined by three characteristics is called 3Vs (Volume, Velocity and Variety). In these circumstances it refers to data that may be too tremendous, dynamic and complex. This multidimensional data are difficult to trance, store, manage, and analyze using conventional data management tools. Thence, the modern situation imposed by Big Data is the present serious challenges at different level, including data clustering.

Volume: Big data means there is a lot of data terabytes or even petabytes of data (1,000 terabytes). Nowadays, lots of data to be processed are continually growing. This describes the fact that our improved use of new technologies (smart phones, social networks or media, connected machines etc.,) encourages to produce more and more data in our daily activities both personal and professional; the companies are facing problems a sudden increase of stored data. Surely, this volume continues to produce at high speed. It is predictable that the volume of data stored in the world doubles in every four years.

Variety: The term variety is made up of that the data is dirty mixed and has not always in organized forms. Surely, it can be use the data contained in websites, blogs, emails, exchanges on social networks (Face book, Twitter, LinkedIn,etc.,), images, video, audio, logs, data spatial (geo location), the biometrics, etc. Their origins are diverse: web, text mining, mining picture, etc. We need to combine heterogeneous sources to illustrate actionable conclusions. The variety of Big Data explains the complexity of using the information from conventional data warehousing infrastructure.

Velocity: In the view of velocity of Big Data can be referred that to the speed at which data is generated, captured and exchanged. Certainly, these data are generated and grow improbably apace. So the collection, analysis and use of data should more often be done in real time, it is even possible to stop storing information and analyzing flow (i.e., streaming), to visualize the true conclusions.

While conventional data warehouse analytics tend to be based on periodic - daily, weekly or monthly-loads and updates of data, big data is processed and analyzed in real- or near-real-time. This is important in healthcare for areas such as clinical decision support, where access to up-to-date information is vital for correct and timely decision-making and elimination of errors. Current data is needed to support automated decision-making; after all, you can't use five-minute-old data to cross a busy street. Machine-driven decisions cannot be trusted without up to date data, forcing expensive and time-consuming manual reviews of each decision.

In 2012, Gartners defined big data as follows: Big data are monolithic volume, high velocity, and large variety information assets that require new forms of processing to enable strengthened decision making, penetration discovery and process optimization. Additionally, another V is added called veracity, shows the uncertainty of the data, as it usually comes from a variety of sources like as given below

- Web and social media data such as interaction data like Face book, Twitter and other blogs etc.
- Electronic Health Records (EHR) such as Health claims and other records that can be structured or unstructured.
- Reading from sensor, meters and other devices.
- Biometrics - Fingerprints, genetics, handwriting and retinal scan data, it consists of X-rays and other medical images and blood pressures and pulse and similar type of data.
- Person generated data such as e-mail records, doctors/nurse notes, paper documents etc.

In real time applications of Big data MapReduce framework is proven that efficient for quick processing. A Map Reduce program is one which can be composed of a Map() procedure is called method that can carry out filtering and sorting such as sorting students by first name into queues, one queue for each name and a Reduce() method is one that carry out a join operation such as counting the number of students in each queue, yielding name frequencies. The "MapReduce System" is also called "infrastructure" or "framework" arranging the processing by marshalling the distributed servers, running the a variety of tasks in parallel, managing all communications and data transfers between the various parts of the system, and providing for redundancy and fault tolerance.

MapReduce is a framework for processing parallelizable problems across huge datasets using a large number of computers (nodes), together referred to as a cluster (if all nodes are on the same local network and use similar hardware) or a grid (if the nodes are shared across geographically and administratively distributed systems, and use more heterogeneous hardware). Processing can occur on data stored either in a file system (unstructured) or in a database (structured). MapReduce can take benefit of the locality of data, processing it near the place it is stored in order to reduce the distance over which it must be transmitted.

"Map" step: Each member node applies the "map()" role to the local data, and writes the output to a temporary storage. A master node arranges that for redundant copies of input data, out of this only one is processed.

"Shuffle" step: Member nodes are redistribute data based on the output keys (produced by the "map()"function), such that all data belonging to one key is located on the alike member node.

"Reduce" step: Member nodes now process each group of output data, per key, in parallel.

MapReduce allows for distributed processing of the map and reduction operations. Provided that each mapping operation is independent of the others, all maps can be performed in parallel – although in practice this is limited by the number of independent data sources and/or the number of CPUs near each source. Similarly, a set of 'reducers' can perform the reduction phase, provided that all outputs of the map operation that share the same key are presented to the same reducer at the same time, or that the reduction task is associative. While this process can often appear inefficient compared to algorithms that are more sequential, MapReduce can be applied to significantly larger datasets than "trade good" servers can handle – a large server farm can use MapReduce to sort a petabyte of data in only a few hours. The parallelism also offers some possibility of recovering from partial failure of servers or storage during the operation: if one mapper or reducer fails, the work can be rescheduled – presumptuous the input data is still available.

MapReduce are both the functions Map and Reduce functions of outlined with respect to data structured in (k, v) pairs. In this dyad k for key and v for value Map takes one dyad of data with a type in one data domain, and returns a list of pairs in a different domain: Map(k1,v1) → list(k2,v2)

The Map function is applied in parallel to every pair in the input dataset. A list of pairs for each call can be produces. Subsequent to that, the MapReduce framework collects all pairs with the same key from all lists and groups them together, creating one group for each key.

The Reduce function is then applied in parallel to each group in turn produces a collection of values in the same domain, which: Reduce (k2, list (v2)) → list (v3)

Each Reduce call typically produces either one value v3 or an empty return; however one call is allowed to return more than one value. The returns of all calls are collected as the desired result list. Therefore the MapReduce framework transforms a list of (k, v) pairs into a list of values. This behavior is different from the typical functional programming map and reduces combination, which accepts a list of arbitrary values and returns one single value that combines all the values returned by map.

It is required but not sufficient to have implementations of the map and reduce abstractions in order to implement MapReduce. Distributed implementation of MapReduce needs a means of connecting the processes performing the Map and Reduce phases. This may be a distributed file system. Other options are possible, such as direct streaming from mappers to reducers, or for the mapping processors to serve up their results to reducers that query them.

The map reduce model is by default the parallel processing model. Out of different data mining techniques as the clustering is most important technique we are exploring the parallel optimal grid-clustering techniques on the map reduce architecture.

Parallel optimal grid-clustering: The processing of huge quantity of data forces a parallel computing to attain results in reasonable time. In this section, we examine some parallel algorithms and distributed clustering used to treat Big Data; the parallel classification divides the data partitions that will be distributed on different machines. This makes an individual classification to speed up the calculation and increases scalability.

A parallel k-means algorithm was proposed by Dhillon and Modha [1], which was then implemented on an IBM SP2 POWER parallel with 16 nodes. In the other hand, Stoffel and Belkoniene [2] have implemented a further parallel version of the k-means algorithm using 32 machines on an Ethernet network and which showed an almost linear acceleration for large data sets. The scalability of the parallel k-means algorithm has also been demonstrated by others [3],

MapReduce is a job partitioning mechanism (with large volumes of data) for a distributed execution on a large number of servers. Principle is to decompose a job (the map part) into smaller jobs. The jobs are then dispatched to different servers, and the results are collected and consolidated (the reduce part).

The paper is organized as follows: the second section represents the related work which shows different state of the art papers. The third section describes the proposed model for the parallel Optimal-Grid clustering Algorithm for big data analytics and its experimental study. The fourth section provides conclusion and future scope of the paper there after the list of references.

2. PREVIOUS WORK

The data that is 90% of the data in the world today has been formed in the last two years alone at present the speed of data creation has improved [4] this massive amount of data is being viewed by business organizations and researchers as a great potential resource of knowledge that needs to be discovered. A simple definition by Jason Bloomberg [5]: “Big Data: a massive amount of both structured and unstructured data that is so bulky that it's difficult to process using conventional database and software techniques.” This is also in accordance with the definition given by Jim Gray in his seminal book [6]. To deal with these challenges, innovative software programming frameworks to multithread computing tasks have been developed [7,8].These programming frameworks are intentional to acquire their parallelism not from a supercomputer, but from computing clusters: large collections of commodity hardware, including conventional processors (computing nodes) connected by Ethernet cables or inexpensive switches. These software programming frameworks commence with a innovative form of file system, known as a distributed file system (DFS) [9], which features much larger units than the disk blocks in a conventional operating system. DFS also provides duplication of data or redundancy to defend against the frequent media failures that occur when data is distributed over potentially thousands of low cost computing nodes [10].

As the primary objectives of this paper is to adopt good clustering technique for map reduce architecture we are exploring different clustering techniques for their efficiency and so that we could adopt a clustering technique for clustering on map reduce architecture. One of the frameworks developed for analyzing and transformation of very huge datasets is Hadoop that employs MapReduce [11-12].MapReduce is a programming paradigm that provides scalability across many servers in a Hadoop cluster with a broad variety of real-world applications [13-14]. Zomaya et al. [15] present a survey of existing clustering algorithms of different categories (Partitioning-based, Hierarchical-based, Density-based, grid-based and model based). Their objective was to find the finest performing for Big Data. To get that in their work they established a comparison between five categories with their most representative algorithm;

In [16] the authors focus on the most popular and most used algorithms in the literature like k-means, they presents some comparative work of these algorithms. Another recent research [17] presents a general view of data mining algorithms and platforms that can be used in the field of Big Data by discussing different challenges and characteristics. There are thousands of clustering algorithms; hence we pick a representative algorithm from each category of partitioning based, hierarchical, density based, and grid partitioning algorithms, as in [18]. CLARA (Clustering LARge Applications) relies on the sampling approach to handle huge data sets [18]. FCM [19] is a representative algorithm of fuzzy clustering which is based on K-means concepts to partition dataset into clusters. The FCM algorithm is a “soft” clustering method in which the objects are assigned to the clusters with a degree of belief. as a result, an object may belong to more than one cluster with different degrees of belief. BIRCH algorithm [20] builds a dendogram known as a clustering feature tree (CF tree). The CF tree can be built by scanning the dataset in an incremental and dynamic way. Thus, it does not need the whole dataset in move forward. The DENCLUE algorithm [21] methodically models the cluster distribution according to the sum of influence functions of all of the data points. Paper [21] discusses some of Big Data mining algorithms to find the most appropriate among them using a comprehensive comparison. Nagpal and Mann’s paper [22] does not address all the clustering technique it is interested only to study density based clustering algorithms such as DBSCAN DENCLUE and to discuss their advantages and disadvantages. Others in [23] are paying attention in studying classification algorithms that can be used in statistics and apply them to specific databases. Researchers in [24] present a review of some old algorithms that can handle large data set as Nearest Neighbor Search, Decision Tree and Neural Network. In [25], Herawan et al. talk about unlike clustering techniques including MapReduce, parallel classification using MapReduce. They present an outline of different categories of data mining clustering algorithms. OptiGrid algorithm [26] is a dynamic programming approach to obtain an optimal grid partitioning. This is achieved by constructing the best cutting hyper planes through a set of selected projections. These projections are then used to find the optimal cutting planes; each plane separating a dense space into two half spaces. It reduces one dimension at each recursive step therefore is good for handling large number of dimensions.

From the above literature survey we found that out of different clustering techniques Optigrid has more number of advantageous like handling multidimensional data. This could be best suitable to manage huge voluminous big data. Considering these issues we decided to explore Optigrid for map reduce architecture in this paper.

3. PROPOSED WORK

3.1 The Parallel OptiGrid Algorithm for big data analytics

In the below architecture we have given the input data this input data is portioned and given to different mappers and results are collected at reducer parts

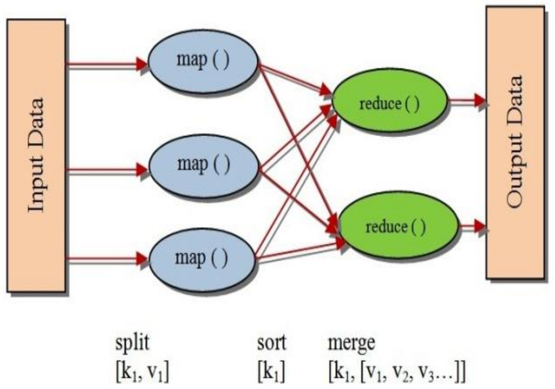

Fig.1: Map Reduce Architecture for Optigrid

The big data which is collection of real time data comprises various numbers of dimensions. To handle such difficult dimensions there is need Parallel Optigrid clustering model to carry out high dimensional clustering. The MapReduce model used in this paper to implement clustering is shown in figure.1. According to model the given data will be first split among the multiple data computing machine. This storage will be handled by distributed file system that is adopted for the implementation. In our implementation we adopted Hadoop distributed files system to manage the data split. Once the data split is been finished next level the concern data mining will be perform at map systems where the local level data will be processed and individual level of results will be obtained. Then using which the shuffler will shuffle the result into map systems where the eventually set of consolidated data mining results will obtained. The final results will be viewed for the user. The same framework is used for parallel optigrid clustering model.

The MapReduce model parallel optigrid includes three steps. In first step the data will be separated among all map systems, then in next step local Optigid technique will be applied on all map machines and map clustering points will be obtain where in final step the at reducer systems the mixing of data from reducer will be perform and global Optigrid will be applied to get final result.

At the map systems the local partitioning is done using a multidimensional grid defined by at most q cutting planes. The map systems will consider each cutting plane is orthogonal to at least one projection. The individual map point density at cutting planes is bound by the density of the orthogonal projection of the cutting plane in the projected space. The q cutting planes are chosen to have a minimal point density. There will be same threshold will be given all the map systems.

```
Local OptiGrid input(Map data set M; p; low_Div_score)
    1.  Discover a set of contracting projections
        $T = \{T_0, T_1, \ldots\ldots T_n\}$
    2.  compute all projections of the data set
        $M \rightarrow \{T_0 (M), T_1 (M), \ldots\ldots\ldots, T_n(M)\}$
    3.  Initialize a list of cutting planes
        TOP_DIV ← Null, DIV ← Null;
    4.  for j=0 to n do
        (a) DIV ← Discover Top_local_Divs($T_j$(M))
        (b)DIV_ SCORE ← Score Top_ local _Divs(Ti(M))
        (c) Insert all cutting planes with a score ≥ low_Div _score into TOP_DIV
    5.  if TOP_ DIV =Null : Then return M as a cluster
    6.  Discover the p cutting planes with highest_score
        from TOP_DIV and delete the rest
    7.  build a Multidimensional Grid $G_d$ defined by the
        cutting planes in TOP_DIV and insert all data points $x \epsilon M$ into Grid $G_d$
    8.  Discover clusters, i.e. determine the highly populated grid cells in $G_d$ and add them to the set of cluster K
    9.  refine(K)
    10. for eachCluster $K_i \epsilon K$ do OptiGrid($K_i$, p,low_Div_ score)
```

At the reducer systems all the local optigrid clusters brought to the specific system. Where the final global optigid will be applied to get final set of optimized high dimensional clusters. In case of running reducer optigrid all individual clusters will be considered as individual datasets and final set of clustering will be obtain.

4. EVALUATION

The MapReduce framework established with 8 number of system in system where one system act as server node and other as computing nodes. Each of this system with Intel core i3 processors, 2GB 2DDR2 RAM, 1Gbps Ethernet connection and hard disk with 2TB capacity. The MapReduce implemented with software version is Hadoop2.0.0-cdh4.4.0 and MapReduce1 runtime (Classic) running on Ubuntu 14.4 operating system. The maximum numbers of map task are 64 and maximum number of reducers is 2.

In this experimental study we use UCI dataset with 5759126 numbers of records with 12 attributes and 2 clusters. The main task of the data set is to extract comparative clusters. The primary set of result shown good hopeful results.

5. CONCLUSION

Considering the absence of effectual clustering techniques in this paper we intend the extension of optimal grid clustering techniques for Bigdata analysis using MapReduce architecture. The initial level experiments conducted using this proposed model is shown magnificent results. The further optimization of technique for real time data sets is can be worthful imminent extension.

6. ACKNOWLEDGMENT

Our thanks to the management members and principal of Kakatiya Institute of Technology and Science-Warangal who have facilitated resources to read and compute in order to develop this model and narrate this article and our sincere thanks to Mr.B.RghuRam for helping to make this paper in materialize and also we extend our gratitude to our beloved Head, CSE Prof.P.Niranjan who encouraged us research and publish this paper.

7. REFERENCES

[1] I. S. Dhillon, and D.S. Modha, “A data-clustering algorithm on distributed memory multiprocessors,” In Large-Scale Parallel Data Mining. Springer Berlin Heidelberg, p. 245-260, 2000.

[2] K. Stoffel and A. Belkoniene, "Parallel k/h-means clustering for large data sets," In Euro-Par'99 Parallel Processing. Springer Berlin Heidelberg, p. 1451-1454, 1999.

[3] H. S. Nagesh, S. Goil, and A. Choudhary, "A scalable parallel subspace clustering algorithm for massive data sets," In Parallel Processing, 2000. Proceedings. International Conference on. IEEE, p. 477-484, 2000.

[4] A. Fahad, N. Alshatri, Z. Tari, A. ALAmri, A. Y. Zomaya, I. Khalil, F. Sebti, and A. Bouras, "A Survey of Clustering Algorithms for Big Data: Taxonomy & Empirical Analysis," IEEE transactions on emerging topics in computing, 2014.

[5] The Big Data Long Tail. Blog post by Bloomberg, Jason. On January 17, 2013. [online] http://www.devx.com/blog/the-big-data-long-tail.html.

[6] The Fourth Paradigm: Data-Intensive Scientific Discovery. Edited by Hey, T. , Tansley, S. and Tolle, K.. Microsoft Corporation, October 2009. ISBN 978-0-9825442-0-4.

[7] Rajaraman A, Ullman JD: Mining of Massive Datasets. Cambridge – United Kingdom: Cambridge University Press; 2012.

[8] Coulouris GF, Dollimore J, Kindberg T: Distributed Systems: Concepts and Design: Pearson Education; 2005

[9] de Oliveira Branco M: Distributed Data Management for Large Scale Applications. Southampton – United Kingdom: University of Southampton; 2009.

[10] A. Sherin, S. Uma, K.Saranya and M. Saranya Vani "Survey On Big Data Mining Platforms, Algorithms And Challenges". International Journal of Computer Science & Engineering Technology,Vol. 5 No, 2014.

[11] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The Hadoop distributed file system," in Proceedings of the IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST '10), pp. 1–6, IEEE, May 2010.

[12] D. Sobhy, Y. El-Sonbaty, and M. Abou Elnasr, "MedCloud: healthcare cloud computing system," in Proceedings of the International Conference for Internet Technology and Secured Transactions, pp. 161–166, IEEE, London, UK, December 2012.

[13] J.Dean and S.Ghemawat, "MapReduce: simplified data processing on large clusters," Communications of the ACM, vol. 51, no. 1, pp. 107–113, 2008.

[14] F. Wang, V. Ercegovac, T. Syeda-Mahmood et al., "Largescale multimodal mining for healthcare with mapreduce," in Proceedings of the 1st ACM International Health Informatics Symposium, pp. 479–483,ACM,November 2010.

[15] W.S. Li, J. Yan, Y. Yan, and J. Zhang, "Xbase: cloud-enabled information appliance for healthcare," in Proceedings of the 13th International Conference on ExtendingDatabase Technology (EDBT '10), pp. 675–680, March 2010.

[16] A.BEN AYED, M.BEN HALIMA and M. ALIMI, "Survey on clustering methods: Towards fuzzy clustering for Big Data," In Soft Computing and Pattern Recognition (SoCPaR), 6th International Conference of. IEEE, p. 331-336, 2014.

[17] Keim, D.et all A. Optimal Grid-clustering:Towards breaking the curse of dimensionality in high-dimensional clustering. In Proceedings of the 25th Conference on VLDB, 506-517, 1999.

[18] Kaufman, L., and Rousseeuw, P. J. Finding Groups in Data: An Introduction to Cluster Analysis, John Wiley & Sons, Inc., New York, NY, 1990.

[19] Bezdek, J. C., Ehrlich, R., and Full, W. Fcm: The fuzzy c-means clustering algorithm. Computers & Geosciences, 10(2):191–203, 1984.

[20] Zhang, T., Ramakrishnan, R., and Livny, M. Birch: an efficient data clustering method for very large databases. ACM SIGMOD Record, volume 25, pp. 103–114, 1996

[21] S.ARORA, I.CHANA, "A survey of clustering techniques for Big Data analysis," in Confluence The Next Generation Information Technology Summit (Confluence), 5th International Conference-. IEEE, p. 59-65, 2014.

[22] P. Batra NAGPAL, and P. Ahlawat MANN, "Survey of Density Based Clustering Algorithms," International journal of Computer Science and its Applications, vol. 1, no 1, p. 313-317,2011

[23] R. XU and D. WUNSCH, "Survey of clustering algorithms," Neural Networks, IEEE Transactions, vol. 16, no 3, p. 645-678, 2005.

[24] C. YADAV, S. WANG, et M. KUMAR, "Algorithm and approaches to handle large Data-A Survey," International Journal of computer science and network, vol 2, issue 3, 2013.

[25] A. S. Shirkhorshidi, S. Aghabozorgi, T. Y. Wah, and T. Herawan, "Big Data Clustering: A Review," In Computational Science and Its Applications–ICCSA 2014. Springer International Publishing, p. 707- 720. 2014.

[26] Hinneburg, A., and Keim, D. A. Optimal Grid-clustering:Towards breaking the curse of dimensionality in high-dimensional clustering. In Proceedings of the 25th Conference on VLDB, 506-517, 1999.

[27]