LITERATURE REVIEW: — Parallel K-means Using MapReduce —

Patrick Killeen
School of Computer Science
University of Ottawa
Ottawa, Canada
pkill013@uottawa.ca

October 4, 2018

1 Introduction

Computing and storage has become cheaper over the years. There is also more and more data being created each day [9]. There is so much data that data analytics and knowledge discovery is becoming more feasible [1]. However, standard data mining algorithms cannot cope with this increasing amount of data [26]. This is pushing the Big Data analytics research field [29].

Parallel computing can be used to speed up algorithms. To speed up an algorithm using parallelism, it must be converted from serial to parallel. There are frameworks designed to handle fault tolerance, distributed computing, synchronization, data storage and processing. Hadoop MapReduce is an example of such a framework [15]. However, it isn't always straight forward how to convert a serial algorithm to parallel. For example, MapReduce is a framework for processing Big Data in jobs, but it isn't designed for many iterations [3]. K-means clustering algorithm [3] is based on iterations [19].

This paper paper intends to list the state of the art of parallel k-means using MapReduce. For my project I intend to chose a paper and implement the algorithm/approach of the parallel k-means and run experiments, to see if I can replicate the results.

2 Literature Review

2.1 Big Data

The growth of data creation is increasing at an exponential rate [9] [14]. Big Data is becoming a problem for processing large amounts of data with limited processing power [7][12]. Data mining algorithms need to be re-designed to handle such massive amounts of data. Big Data analytics is the field of performing machine learning on Big Data. According to [2], in Big Data there are three Vs, namely: Volume, Velocity, and Variety. Velocity represents the speed at which the data is flowing(a fast data stream). Volume represents the sheer size of the data. Variety represents the heterogeneous nature of the data, as data is coming from many data sources. There are two more dimensions: Variability and Complexity. Variability represents the difference of flow between data source. Complexity must also be taken into account [2].

With all this data it allows us to acquire knowledge [1]. The trend of huge amounts of data creation has encouraged the research field of Big Data analytics [4][8][29]. Big Data leads to efficient storage and processing requirements [17] in a distributed manner [7]. Therefore, efficient processing algorithms are required for analysing data [17]. Big Data analytics can tackled by using parallelization [8]. For example by using the Hadoop framework [14][28]. Clustering is an important aspect of Big Data analytics [28].

2.2 Data Mining and Clustering

Data mining consists of extracting knowledge and finding novelties from datasets by analysing patterns among the data elements [6][28]. Clustering is a data mining technique [2] [1][26], which is an unsupervised approach for finding outliers and grouping data together [7]. Clustering a dataset partitions it into groups of similar data elements [6][9][10], such that each group/cluster have data elements that are different from data elements in other clusters [1][11] [14]. Clustering is a popular technique for trying to solve data analytics problems [8][12]. It has applications such as information retrieval [14], stock exchange analysis [18], opinion mining [25], and image pattern recognition [31]. Knowledge discovery using data analytics can be divided into stages: data preprocessing, clustering the preprocessed data, and analysing the results for interesting patterns [23]. It becomes more difficult to perform data analytics as datasets become larger [9][11] [18][19][26] [32].

2.3 K-means

K-means algorithm is an efficient clustering algorithm [3] [5][10][30] [32][33]. K-means is one of the most popular unsupervised clustering algorithms [4][3] [8][9][13] [16][20], because of its simplicity and efficiency [10][12][28]. It can be used for clustering large datasets [13], which can be made of structured or unstructured data [6]. It has many different applications [5][20].

K-means attempts to cluster datasets into k clusters of similar elements [1]. It first starts the initialization stage by choosing k centroids (the centers of clusters) at random [19][22], and assigns each data element of the dataset to the nearest centroid using the Euclidean distance. At the end of this stage a new set of k centroids are calculated by taking average Euclidean distance of each data element within a cluster. This is done until the centroids converge [16][2]. Most of the computations performed in this algorithm are distance computations, which are performed when comparing all data elements' distance to each of the centroids [3].

However, the k-means isn't perfect, it has its limitations. It has trouble dealing with outliers [9][22]. That is, the algorithm assigns each data element to a cluster, but it may not make sense in some contexts to assign an outlier to a cluster. Furthermore, the resulting clusters's stability and accuracy are sensitive to the initial centroids chosen [33]. That is, the resulting clusters will vary depending on the initial centroids chosen [19], and random initial centroids prove to yield unstable cluster results [1][22]. Therefore, to optimize k-means, care must be taken when choosing the initial centroids [10]. One of the bottle-necks in k-means is the number of iterations [19]. The more iterations there are, the more the clusters will converge at the cost of increased computations. As datasets becomes extremely large, k-means begins to lack in performance [4] [13][30][32], and its results become unstable [9]. The larger the dataset, the more iterations that will be required for high quality clusters, which will therefore take more computations [32]. Execution time could be improved using parallelism [10].

2.4 Map Reduce

As we enter the Big Data era, a lot of research is put into MapReduce [12]. MapReduce is a framework for processing Big Data [20] in parallel [5] [8][9] in a distributed manner. Map Reduce is a framework proposed by Google for processing Big Data, which involves storing, appending, and also running jobs seamlessly in a parallel, distributed, and fault tolerant manner [15]. Apache's Hadoop Map Reduce is written in Java and is open source [1], which is a version of Google's Map Reduce[13]. MapReduce has 2 phases, the Map phase and the Reduce phase [1][14][34]. User defined map and reduce functions are used to process key-value pairs as inputs [13]. Map Reduce partitions data into subsets during the mapping phase, and assigns each partition to a worker machine to be processed [32]. Once each worker has processed each partition, the results are combined in the reduction phase. The framework was created to meet the requirements of Big Data, trying to make sense of all these data available. It is a popular and is used by many companies. It can be deployed to many 100s of machines for processing Big Data [1]. There is no data cached between two consecutive MapReduce jobs [3]. There aren't very many data mining algorithms that are implemented using MapReduce [8]. The MapReduce jobs have a lot of I/O cost from reading and upon job completion writing to the file system. Therefore, many iterations in algorithms using MapReduce should be avoided when minimizing performance costs [30].

2.5 Hadoop Distributed File System

Hadoop framework also provides a distributed file system [8][13]. HDFS is in charge of storing and processing large amounts of data on distributed nodes [14], creating replications when necessary [1].

2.6 Parallel k-means using Map Reduce

For applying k-means to Big Data, k-means can be parallelized using MapReduce. There is a difficulty when combining k-means with MapRecude. k-means uses iterations by definition and Map Reduce doesn't support iterations [3][27], since each MapReduce job has a lot of I/O costs associated [27]. This suggests directly mapping k-means iterations to the MapReduce jobs would prove to have low performance. There are many solutions in the literature that attempt to optimize k-means in a parallel environment running it in Hadoop's MapReduce in a variety of ways. Many works' goals are to minimize the execution time while maximizing cluster quality [11].

In the literature there are many solutions for implementing the k-means algorithm in a parallel environment using Map Reduce such as [34], and [35]. [8] is a survey on k-means clustering using MapReduce for Big Data. [29] uses genetic algorithm steps. [24] improves serial Two-Phase K-means using Incremental k-means algorithm. [21] studies this problem with real-time time-series data. There are many approaches proposed in the literature, such as optimizing the initial centroids chosen, minimizing the number of k-means iterations, minimizing the number of distance calculations, optimizing hardware configuration, outlier removal, etc. [22] and [9] remove outliers in attempt to optimize k-means over MapReduce.

There are quite a few applications to parallel k-means over MapReduce. [25] proposes an aspect based summary generation solution by mining opinions. [26] designs a k-means over MapReduce algorithm and compares it to serial k-means for document datasets. [31] implements k-means over MapReduce for image pattern recognition.

2.6.1 Initial Centroid Optimization

The following literatures attempt to optimize the initial centroids chosen: [1], [20], [2], [30], [7], [9], [19], [28], [33], [23], and [10]. [2] optimizes the initial centroids using data dimentionality density. [7] varies the centroids and data to optimize k-means. [10] optimizes initial centroid choice using the PSO meta-heuristics, which improves cluster quality and execution time. [9] uses two approaches, a), removing outliers from the datasets, and b), automating the initial centroid selection. [19] uses Min-Max normalization technique to choose better initial centroids, which requires assigning weights/priority to attributes of the dataset. The work done by [28] achieves better accuracy than the traditional k-means by taking averages of the dataset for better selecting the initial centroids. [33] compares their algorithm, Adaptively Disperse Centroids K-means Algorithm to Mahout. [23] introduces a preprocessing phase to compute the initial centroids, and then focuses on evaluation of cluster quality using data preprocessing, clustering, and pattern recognition.

2.6.2 Minimizing k-means Iterations

[16] introduces a preprocessing stage to k-means over MapRecude using k-d tree, to allow completion of the k-means algorithm in one MapReduce job. This work shows that with the same centroid configuration, their approach is faster and produces similar cluster quality compared to other literatures. [18] aims to optimize the execution time while keeping 80% accuracy of clusters by reducing iterations. [20] demonstrates (via simulation) that their work reduces the number of k-means iterations and increases the speed of the iterations. This works does so by analysing the dataset's distribution for initial centroid selection, and dynamically chooses between the Euclidean distance and Manhattan distance algorithms for comparing data element's distances. [27] proposes a single-pass MapReduce job for parallelizing k-means, called mrk-means. This work uses re-clustering and increases cluster quality. [30] automatically determines the number of clusters that will be generated, and only requires one MapReduce job, minimizing I/O cost to the file system.

2.6.3 Minimizing Number of Distance Computations

[4] reduces the number of distance computations performed, and achieve the same results. [3] reduces the distance computations using triangle inequality by using Extended Vector and Bounds Files. They compare both the Extended Vector and Bound Files approaches together. [6] carefully designs the Mapper and Reducer. Similarly, [5] attempts to reduce the number of reads and write to disk by the Mapper and Reducer. [32] reduces the number of iterations up to 30% and keeps up to 98% accuracy.

2.6.4 Optimiznig Hardware Configuration in Hadoop Environment

[13] explores the use of CPUs and GPUs using OpenCL to optimize k-means using MapReduce. [15] analyses the performance when adjusting processor micro-architecture parameters. [17] validates the importance of k-means over MapReduce by conducting experiments and varying the number of nodes in Hadoop environment.

References

[1] Nadeem Akthar, Mohd Vasim Ahamad, and Shahbaaz Ahmad. Mapreduce model of improved k-means clustering algorithm using hadoop mapreduce. In *Computational Intelligence & Communication Technology (CICT), 2016 Second International Conference on*, pages 192–198. IEEE, 2016.

[2] Nadeem Akthar, Mohd Vasim Ahamad, and Shahbaz Khan. Clustering on big data using hadoop mapreduce. In *Computational Intelligence and Communication Networks (CICN), 2015 International Conference on*, pages 789–795. IEEE, 2015.

[3] Sami Al Ghamdi and Giuseppe Di Fatta. Efficient parallel k-means on mapreduce using triangle inequality. In *Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence & Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), 2017 IEEE 15th Intl*, pages 985–992. IEEE, 2017.

[4] Sami Al Ghamdi, Giuseppe Di Fatta, and Frederic Stahl. Optimisation techniques for parallel k-means on mapreduce. In *International Conference on Internet and Distributed Computing Systems*, pages 193–200. Springer, 2015.

[5] Prajesh P Anchalia. Improved mapreduce k-means clustering algorithm with combiner. In *Computer Modelling and Simulation (UKSim), 2014 UKSim-AMSS 16th International Conference on*, pages 386–391. IEEE, 2014.

[6] Prajesh P Anchalia, Anjan K Koundinya, and NK Srinath. Mapreduce design of k-means clustering algorithm. In *Information Science and Applications (ICISA), 2013 International Conference on*, pages 1–5. IEEE, 2013.

[7] Soumyendu Sekhar Bandyopadhyay, Anup Kumar Halder, Piyali Chatterjee, Mita Nasipuri, and Subhadip Basu. Hdk-means: Hadoop based parallel k-means clustering for big data. In *Calcutta Conference (CALCON), 2017 IEEE*, pages 452–456. IEEE, 2017.

[8] Rujal D Bhandari and Dipak P Dabhi. Extensive survey on k-means clustering using mapreduce in datamining.

[9] Amira Boukhdhir, Oussama Lachiheb, and Mohamed Salah Gouider. An improved mapreduce design of kmeans for clustering very large datasets. In *Computer Systems and Applications (AICCSA), 2015 IEEE/ACS 12th International Conference of*, pages 1–6. IEEE, 2015.

[10] Abdelhak Bousbaci and Nadjet Kamel. A parallel sampling-pso-multi-core-k-means algorithm using mapreduce. In *Hybrid Intelligent Systems (HIS), 2014 14th International Conference on*, pages 129–134. IEEE, 2014.

[11] Abdelhak Bousbaci and Nadjet Kamel. Efficient data distribution and results merging for parallel data clustering in mapreduce environment. *Applied Intelligence*, pages 1–21, 2017.

[12] Osama A Doreswamy and BR Manjunatha. Scalable k-means algorithm using mapreduce technique for clustering big data.

[13] Sandip A Ganage and Dr RC Thool Heshsham Abdul Basit. Heterogeneous computing based k-means clustering using hadoop-mapreduce framework. *International Journal of Advanced Research In Computer Science and Software Engineering*, 3(6), 2013.

[14] Bharath Kumar Gowru and Pavani Potnuri. Parallel two phase k-means based on mapreduce. *International Journal of Advance Research in Computer Science and Management Studies*, 22:3–11, 2015.

[15] Joseph Issa. Performance characterization and analysis for hadoop k-means iteration. *Journal of Cloud Computing*, 5(1):3, 2016.

[16] Shikai Jin, Yuxuan Cui, and Chunli Yu. A new parallelization method for k-means. *arXiv preprint arXiv:1608.06347*, 2016.

[17] Amresh Kumar, M Kiran, and BR Prathap. Verification and validation of mapreduce program model for parallel k-means algorithm on hadoop cluster. In *Computing, Communications and Networking Technologies (ICCCNT), 2013 Fourth International Conference on*, pages 1–8. IEEE, 2013.

[18] Oussama Lachiheb, Mohamed Salah Gouider, and Lamjed Ben Said. An improved mapreduce design of kmeans with iteration reducing for clustering stock exchange very large datasets. In *Semantics, Knowledge and Grids (SKG), 2015 11th International Conference on*, pages 252–255. IEEE, 2015.

[19] K Gaja Lakshmi and D Prabha. Clustering big data using normalization based k-means algorithm. 2016.

[20] Qing Liao, Fan Yang, and Jingming Zhao. An improved parallel k-means clustering algorithm with mapreduce. In *Communication Technology (ICCT), 2013 15th IEEE International Conference on*, pages 764–768. IEEE, 2013.

[21] Yongzheng Lin, Kun Ma, Runyuan Sun, and Ajith Abraham. Toward a mapreduce-based k-means method for multi-dimensional time serial data clustering. In *International Conference on Intelligent Systems Design and Applications*, pages 816–825. Springer, 2017.

[22] Li Ma, Lei Gu, Bo Li, Yue Ma, and Jin Wang. An improved k-means algorithm based on mapreduce and grid. *International Journal of Grid & Distributed Computing*, 8(1), 2015.

[23] Veronica S Moertini and Liptia Venica. Enhancing parallel k-means using map reduce for discovering knowledge from big data. In *Cloud Computing and Big Data Analysis (ICCCBDA), 2016 IEEE International Conference on*, pages 81–87. IEEE, 2016.

[24] Cuong Duc Nguyen, Dung Tien Nguyen, and Van-Hau Pham. Parallel two-phase k-means. In *International Conference on Computational Science and Its Applications*, pages 224–231. Springer, 2013.

[25] V Priya and K Umamaheswari. Ensemble based parallel k means using map reduce for aspect based summarization. In *Proceedings of the International Conference on Informatics and Analytics*, page 26. ACM, 2016.

[26] Tanvir Habib Sardar and Zahid Ansari. An analysis of mapreduce efficiency in document clustering using parallel k-means algorithm. *Future Computing and Informatics Journal*, 2018.

[27] Saeed Shahrivari and Saeed Jalili. Single-pass and linear-time k-means clustering based on mapreduce. *Information Systems*, 60:1–12, 2016.

[28] Rajashree Shettar and Bhimasen V Purohit. A mapreduce framework to implement enhanced k-means algorithm. In *Applied and Theoretical Computing and Communication Technology (iCATccT), 2015 International Conference on*, pages 361–363. IEEE, 2015.

[29] Puja Shrivastava, Laxman Sahoo, Manjusha Pandey, and Sandeep Agrawal. Akmaugmentation of k-means clustering algorithm for big data. In *Intelligent Engineering Informatics*, pages 103–109. Springer, 2018.

[30] Ankita Sinha and Prasanta K Jana. A novel mapreduce based k-means clustering. In *Proceedings of the First International Conference on Intelligent Computing and Communication*, pages 247–255. Springer, 2017.

[31] Anil R Surve and Nilesh S Paddune. A survey on hadoop assisted k-means clustering of hefty volume images. *International Journal on Computer Science & Engineering*, 6(3):113–117, 2014.

[32] Duong Van Hieu and Phayung Meesad. Fast k-means clustering for very large datasets based on mapreduce combined with a new cutting method. In *Knowledge and Systems Engineering*, pages 287–298. Springer, 2015.

[33] Bin Wang, Zheng Lv, Jinwei Zhao, Xiaofan Wang, and Tong Zhang. An adaptively disperse centroids k-means algorithm based on mapreduce model. In *Computational Intelligence and Security (CIS), 2016 12th International Conference on*, pages 142–146. IEEE, 2016.

[34] Hongbo Xu, Nianmin Yao, Qilong Han, and Haiwei Pan. Parallel implementation of k-means clustering algorithm based on mapreduce computing model of hadoop. *Metallurgical & Mining Industry*, (4), 2015.

[35] Weizhong Zhao, Huifang Ma, and Qing He. Parallel k-means clustering based on mapreduce. In *IEEE International Conference on Cloud Computing*, pages 674–679. Springer, 2009.