ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 5, May 2019

Review on Optimised Parallel K-Means Clustering using YARN in Hadoop

Shesh Narayan Mishra[1], Shalini Vashisth[2]

M.TECH, Student, Department of CSE, SRCEM College, Palwal, Haryana, India[1]

Assistant Professor, Department of CSE, SRCEM College, Palwal, Haryana, India[2]

ABSTRACT: Document Clustering is a victor among the hugest unsupervised learning used for desire and annihilations variations from the norm by the occasion of grouping the mammoth and voluminous data. As the measure of the data is extending every day, it has transformed into a badly arranged action to process these data with compelled computational resources using Bog Data services like Apache Hadoop and Spark. The current time and period is to view Big Data services based on Parallel execution using YARN (Yet Another Resource Negotiator), which requires an improvement development to store and process the data in a reliably manner with asynchronous executions using Map and Reduce. Apache Hadoop offers a response for this issue by organizing techniques using parallel K-Means vide calculating the error rate using Euclidean and Manhattan functions as vocational model for more efficient and prompt delivery of information based on clusters. In this paper, we have inspected a computation to process the K-Means figuring in Hadoop by integrating the K-Means function using Euclidean and Manhattan with YRAN Cluster centres therein we will draw a connection on parallel and progressive execution, keeping various segments the comparable. The test outcome depicts that our figuring can capably process immense dataset on the conditions using parallel execution using K-Means and YARN.

KEYWORDS: Machine Learning, K-Means, Clustering, Big Data, Hadoop, YARN (Yet Another Resource Negotiator), Hadoop Distributed File System, Map and Reduce.

I. INTRODUCTION

Clustering is a noteworthy unsupervised learning method. A significance of bundling is "he route toward dealing with things into social events whose people are practically identical by one way or another or another" [4]. The gathering of various clusters is a great deal of articulates dissimilarity among them and dissimilar data to various clusters objects. K-Means is one of the fast gathering computation and a standout amongst the best 10 estimations in data mining. The K-Means have a couple of issues; the principal issue is overseeing high estimation and enormous bulk record. Map Reduce and Bulk asynchronous Parallelism is a programming model for execution for making and dealing with tremendous enlightening accumulation while dealing with voluminous data. To handle the issue see above, Map-Reduce used the Hadoop programming model and Bulk as synchronization Parallelism is used YARN programming model to deal with the inefficient issue in a gathering on tremendous instructive accumulations based on clusters. Map Reduce (MR) programming model is common for huge scale data examination and Hadoop is used for the use of this MR programming model on an immense scale. The architectural model of the MR model is to streamline trustworthiness and adjustment to non-basic disappointment, it doesn't store data in memory among guide and exercise assignments of MR punch [5]. The execution of the MR model, the data is go to the accompanying MR work, in the figuring absolutely in memory of the pack. Mass as synchronization Parallelism is redoing model contains a progression of super advances which will reduce the time of cluster creating using YARN. Each cluster development is executed in parallel by every companion in the time and distance estimation using parallel K-Means. The overly advance includes adjacent count, a methodology correspondence, and deterrent asynchronous [13]. For effective and swift results count we used Apache's YARN which will organize and independent model using Hadoop Distributed File System on the most astounding purpose of Hadoop that can be used technique any kind of data in the total memory of the gathering with at most computational power based on CPU and respective hardware. The essential duty of this

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

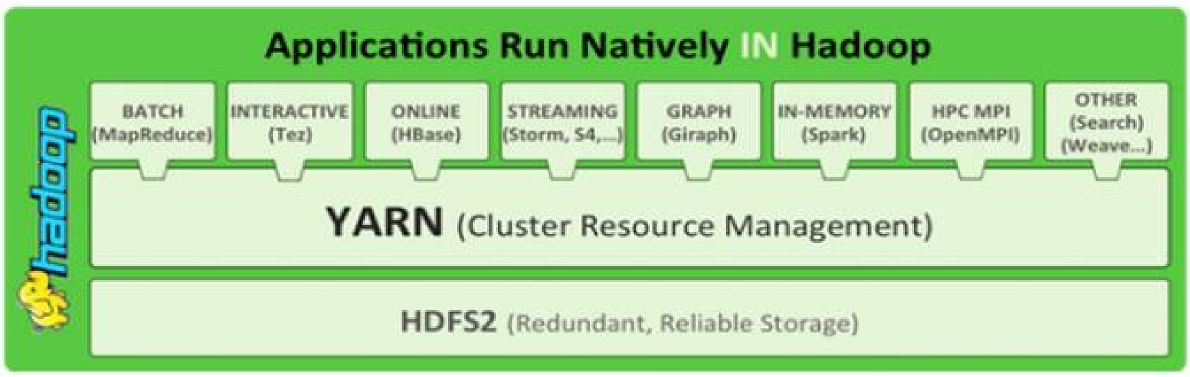
(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 5, May 2019

paper is dense as seeks after: The inefficiency issue in a gathering on gigantic enlightening files is clarified by using a dispersed preparing framework Map Reduce and Bulk Asynchronous Parallelism using K-Means based in Euclidean Distances Calculation and Test and Train Error calculations.

Apache Hadoop YARN : Apache Hadoop YARN is the resource management and job scheduling technology in the open source Hadoop distributed processing framework. One of Apache Hadoop's core components, YARN is responsible for allocating system resources to the various applications running in a Hadoop cluster and scheduling tasks to be executed on different cluster nodes. YARN stands for Yet Another Resource Negotiator, but it's commonly referred to by the acronym alone; the full name was self-deprecating humor on the part of its developers. The technology became an Apache Hadoop subproject within the Apache Software Foundation (ASF) in 2012 and was one of the key features added in Hadoop 2.0, which was released for testing that year and became generally available in October 2013.The addition of YARN significantly expanded Hadoop's potential uses. The original incarnation of Hadoop closely paired the Hadoop Distributed File System (HDFS) with the batch-oriented MapReduce programming framework and processing engine, which also functioned as the big data platform's resource manager and job scheduler. As a result, Hadoop 1.0 systems could only run MapReduce applications -- a limitation that Hadoop YARN eliminated. Before getting its official name, YARN was informally called MapReduce 2 or NextGen MapReduce. But it introduced a new approach that decoupled cluster resource management and scheduling from MapReduce's data processing component, enabling Hadoop to support varied types of processing and a broader array of applications. For example, Hadoop clusters can now run interactive querying, streaming data and real-time analytics applications on Apache Spark and other processing engines simultaneously with MapReduce batch jobs.


Applications Run Natively IN Hadoop
BATCH (MapReduce)
INTERACTIVE (Tez)
ONLINE (HBase)
STREAMING (Storm, S4,...)
GRAPH (Giraph)
IN-MEMORY (Spark)
HPC MPI (OpenMPI)
OTHER (Search) (Weave...)
YARN (Cluster Resource Management)
HDFS2 (Redundant, Reliable Storage)
hadoop
Figure 1 : Hadoop Ecosystem using YARN Cluster Resource Management and Depicting the Applications Run Natively in Hadoop over the HDFS (Hadoop Distributed File System)

Hadoop YARN Features and Functions : In a cluster architecture, Apache Hadoop YARN sits between HDFS and the processing engines being used to run applications. It combines a central resource manager with containers, application coordinators and node-level agents that monitor processing operations in individual cluster nodes. YARN can dynamically allocate resources to applications as needed, a capability designed to improve resource utilization and application performance compared with MapReduce's more static allocation approach.

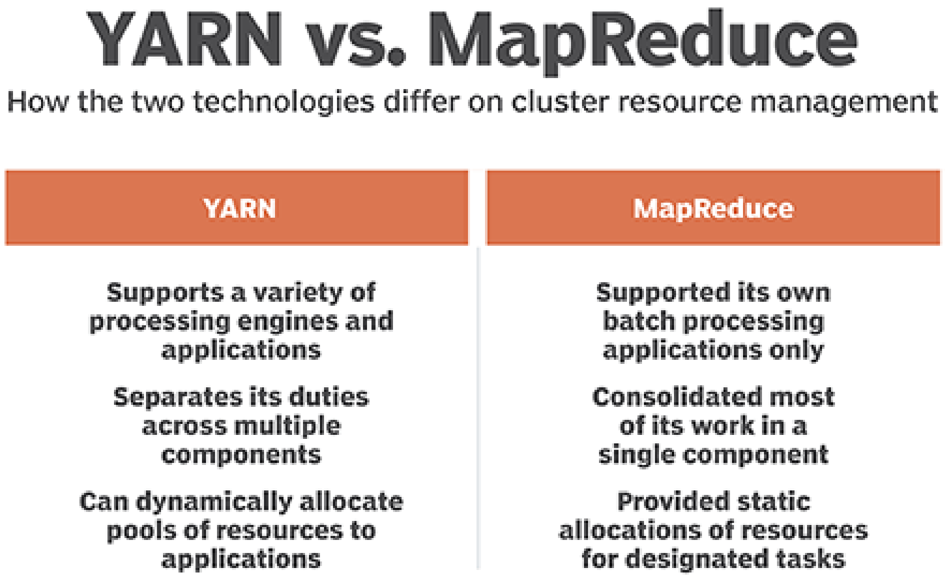
ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 5, May 2019



YARN vs. MapReduce

How the two technologies differ on cluster resource management

YARN	MapReduce
Supports a variety of processing engines and applications	Supported its own batch processing applications only
Separates its duties across multiple components	Consolidated most of its work in a single component
Can dynamically allocate pools of resources to applications	Provided static allocations of resources for designated tasks

Figure 2 : Comparison Between YARN and Map Reduce Based on Cluster Resource Management.

K-Means Distance Calculations: In order to measure the similarity or regularity among the data-items, distance metrics plays a very important role. It is necessary to identify, in what manner the data are interrelated, how various data dissimilar or similar with each other and what measures are considered for their comparison. The main purpose of metric calculation in specific problem is to obtain an appropriate distance /similarity function. Metric learning has emerged as a popular issue in many learning tasks and also it can be applied in a wide variety of settings, since many learning problems involve a definite notion of distance or similarity [1,4]. A metric function or distance function is a function which defines a distance between elements/objects of a set [4,5]. A set with a metric is known as metric space. This distance metric plays a very important role in clustering techniques. The numerous methods are available for clustering. In the current paper, the solution of k-means clustering algorithm using Manhattan distance metric is proposed. Normally, the task is to define a function Similarity(X,Y), where X and Y are two objects or sets of a certain class, and the value of the function represents the degree of "similarity" between the two. Formally, a distance function is a function Dist with positive real values, defined on the Cartesian product X x X of a set X. It is called a metric of X if for each x, y, z ε X: - Dist(x,y)=0 if x=y (the identity axiom); - Dist(x,y) + Dist(y,z) ≥ Dist(x,z) (the triangle axiom); - Dist(x,y)=Dist(y,x) (the symmetry axiom). Metric space metric provides a set X. Bulk synchronization Parallelism is programming model consists of a sequence of super steps. Each BSP superstep is implementation in parallel by every peer in the BSP computation. The super step consists of a local computation, a process communication, and barrier synchronization [13]. For Map Reduce and Bulk Asynchronous Parallelism computation we used Apache's YARN platform built on the top of Hadoop that can be used process any kind of data in the collective memory of the cluster. The main contribution of this paper is summarized as follows: The inefficiency problem in clustering on large data sets is solved by using a distributed computing framework Map Reduce and Bulk Asynchronous Parallelism.

Euclidean Distance: Euclidean distance computes the root of square difference between co-ordinates of pair of objects.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Suppose we have Cluster A = (−1, 2) and Cluster B = (1, 6) and Clusters AB = $2\sqrt{5}$ when we calculate the distance using the Euclidean Distance Formula:
The slope of the line containing Cluster A and Cluste B is found using the slope formula

$m = \frac{y_2 - y_1}{x_2 - y_1}$ so for this example m = 2. We need the factor $\sqrt{1+m^2}$ which is $\sqrt{5}$.

The geometric coordinate for A is $-1(\sqrt{5}) = -\sqrt{5}$ and the geometric coordinate for B is $\sqrt{5}$ (the x coordinate for B is 1).

The absolute value of the difference is $\left|-\sqrt{5}-\sqrt{5}\right|$ which is $2\sqrt{5}$ as achieved using Euclidean Distance.

Bulk Asynchronous Parallelism : Bulk Asynchronous Parallelism computer consists of processors connected by communication network. A computation proceeds in a series of global super steps. The super step consists of a local computation, a process communication, and barrier a-synchronisation. YARN is a distributed computing framework based on Bulk Asynchronous Parallel based on K-Means computing technique for massive scientific computations.

II. Related Study

In the recent years, many clustering algorithms for big data have been proposed which are based on distributed and parallel computation. Mac Queen in 1967 firstly proposed this technique. The standard algorithm was first proposed by Stuart Lloyd in 1957 as a technique for pulsecode modulation. Sometimes it is referred as Lloyd-Forgy because in 1965, E.W.Forgy published essentially the same method [5].

According to K. A. Abdul Nazeer et al [6] the major drawback of the k-means algorithm is about selecting of initial centroids which produces different clusters. But final cluster quality in algorithm depends on the selection of initial centroids. Two phases includes in original k means algorithm: first for determining initial centroids and second for assigning data points to the nearest clusters and then recalculating the clustering mean. But this enhanced clustering method uses both the phases of the original k-means algorithm. This algorithm combines a systematic method for finding initial centroids and an efficient way for assigning data points to clusters. But still there is a limitation in this enhanced algorithm that is the value of k, the number of desired clusters, is still required to be given as an input, regardless of the distribution of the data points.

According to Y. S. Thakare et al. [7], the performance of k-means algorithm which is evaluated with various databases such as Iris, Wine, Vowel, Ionosphere and Crude oil data Set and various distance metrics. It is concluded that performance of k-means clustering is depend on the data base used as well as distance metrics.

Soumi Ghosh et al. [8] proposed a comparative discussion of two clustering algorithms namely centroid based K-Means and representative object based Fuzzy C-Means clustering algorithms. This discussion is on the basis of performance evaluation of the efficiency of clustering output by applying these algorithms. The result of this comparative study is that FCM produces closer result to the K-means but still computation time is more than k-means due to involvement of the fuzzy measure calculations.

Sakthi et al. [9] proposed that due to the increment in the amount of data across the world, analysis of the data turns out to be very difficult task. To understand and learn the data, classify those data into remarkable collection. So, there is a need of data mining techniques.

R. Amutha et al. [10] proposed that when two or more algorithms of same category of clustering technique is used then best results will be acquired. Two k-means algorithms: Parallel k/h-Means Clustering for Large Data Sets and A Novel K-Means Based Clustering Algorithm for High Dimensional Data Sets. Parallel k/h-Means algorithm is designed to deal with very large data sets. Novel K-Means Based Clustering provides the advantages of using both HC and K-Means. Using these two algorithms, space and similarity between the data sets present each nodes is extended.

Nidhi Singh et al. [11] proposed the comparative analysis of one partition clustering algorithm (k means) and one hierarchical clustering algorithm (agglomerative).On the basis of accuracy and running time the performance of k-means and hierarchical clustering algorithm is calculated using WEKA tools. This work results that accuracy of k-means is higher than the hierarchical clustering for iris dataset which have real attributes and accuracy of hierarchal

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 5, May 2019

clustering is higher than the k-means for diabetes dataset which have integer, real attribute. So for large datasets k means algorithm is good.

Shi Na et al. [12] Proposed the analysis of shortcomings of the standard k-means algorithm. As k-means algorithm has to calculate the distance between each data object and all cluster canters in each iteration. This repetitive process affects the efficiency of clustering algorithm.

III.PROPOSED METHODOLOGY

The proposed scheme is responsible to generate the voluminous corpus data from various sources and then migrating to HDFS thereinafter the pre-processing will remove the noise from the voluminous dataset and create the segregated data partitions to be passed to YARN for clusters creating and Parallel processing. Consequently, K-Means along-with Euclidean Distance function will calculate the data similarity distance and error rate among the cluster using a parallel processing model by YARN and at last the resulted clustered will be promoted as a result. Below diagram elaborate the proposed scheme.

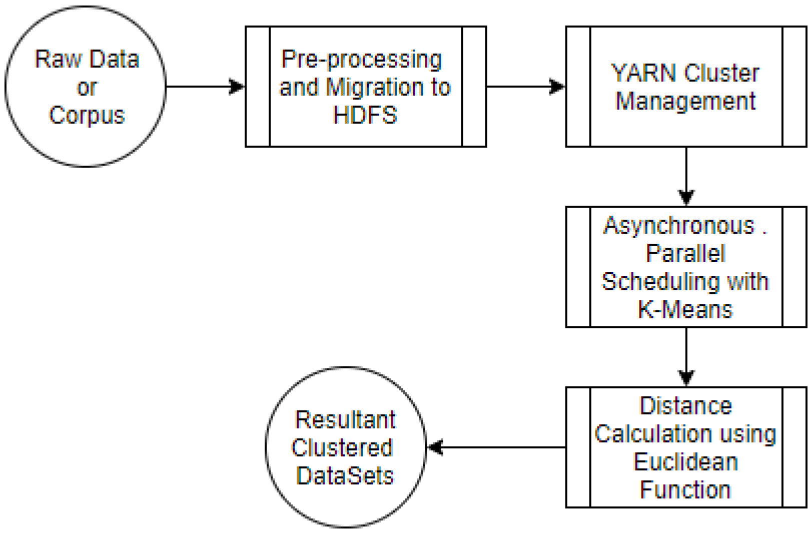

Raw Data or Corpus
Pre-processing and Migration to HDFS
YARN Cluster Management
Asynchronous . Parallel Scheduling with K-Means
Distance Calculation using Euclidean Function
Resultant Clustered DataSets

Figure 3: Proposed Scheme Comprising of Workflow Model Yielding Raw Data or Corpus Collection, Pre-Processing usimg Hadoop Distributed File System, YARN Cluster Management, Asynchronous Parallel Scheduling with K-Means and Cluster Distance Calculations using Euclidean Function.

IV.CONCLUSION

In the above proposed scheme we will inculcate the new scenario into K-Means clustering using YARN resource management vide asynchronous parallel scheduling of cluster along-with K-Means and which will segregate the clusters into sub-clusters and parallel execution of datasets will be done using K-Means whereas the distance evaluation between the clusters comprising the datasets will be formed using Euclidean Function under the governance of K-Means and the results will be produced more efficiently and promptly from various previous models.

REFERENCES

1. Agrawal R., Faloutsos C., Swami A. Efficient similarity search in sequence databases. Proc. 4 Th Int. Conf. On Foundations of Data Organizations and Algorithms, 1993. – Chicago. pp. 69-84.
2. Archana Singh, Jyoti Agarwal, Ajay Rana January 2013. Performance Measure of Similis and FPGrowth Algorithm. International Journal of Computer Applications (0975 – 8887) Volume 62– No.6.[3] Archana Singh, Megha Chaudhary, Dr (Prof.) Ajay Rana

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 5, May 2019

3. Gaurav Dubey 2011 .Online Mining of data to Generate Association Rule Mining in Large Databases. IEEEInternational Conference on Recent Trends in Information Systems.
4. Fast Distance Metric Based Data Mining Techniques Using P-trees: k-Nearest-Neighbor classification and kClustering : A Thesis Submitted to the Graduate Faculty Of the North Dakota State University.
5. Malwindersingh, Meenakshibansal, A Survey on Various K-Means algorithms for Clustering, IJCSNS International Journal of Computer Science and Network Security, VOL.15 No.6, June 2015.
6. K. A. Abdul Nazeer, M. P. Sebastian, Improving the Accuracyand Efficiency of thek-means Clustering Algorithm, Proceedings of the World Congress on Engineering 2009 VOL.1 2009, July 1 - 3, 2009, London, U.K.
7. Y. S. Thakare, S. B. Bagal, Performance Evaluation of K-meansClustering Algorithm with Various Distance Metric, International Journal of Computer Applications (0975 n 8887) Volume 110 ñ No. 11, January 2015.
8. Soumi Ghosh, Sanjay Kumar Dubey, Comparative Analysis ofK-Means and Fuzzy C-Means Algorithms, International Journal of Advanced Computer Science and pplications, Vol. 4, No.4, 2013.
9. Sakthi,M., Thanamani ,A., An Enhanced K Means Clustering using Improved Hopfield Artificial Neural Network and Genetic Algorithm, International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Vol-2, 2013.
10. R.Amutha, Renuka. K, Different Data Mining Techniques And Clustering Algorithms, International Journal Of Technology Enhancements And Emerging Engineering Research, VOL 3, ISSUE 11, ISSN 2347-4289.
11. Nidhi Singh,. Divakar Singh, Performance Evaluation of K-Means and Heirarichal Clustering in Terms of Accuracy and Running Time, International Journal of Computer Science and Information Technologies, Vol. 3 (3) , 2012,4119-4121.
12. Shi Na, Liu Xumin, Guan Yong, Research on K-means Clustering Algorithm: An Improved K-means Clustering Algorithm, Intelligent Information Technology and Security Informatics,2010 IEEE Third International Symposium on 2-4 April, 2010(pp. 63-67).
13. Cui, Xiaoli and Zhu, Pingfei and Yang, Xin and Li, Keqiu and Ji,Changqing, Optimized big data K-means clustering using MapReduce, The Journal of Supercomputing, 70, 1249-1259 (2014)
14. Yan, W., Brahmakshatriya, U., Xue, Y., Gilder, M.and Wise, B. p-PIC:Parallel power iteration clustering for big data. Journal of Parallel and Distributed computing, 73(3), 352-359.(2013)
15. Joaquin Perez Ortega, Ma. Del Rocio Boone Rojas and Maria J. Somodevilla Garcia. Research issues on Kmeans Algorithm: An Experimental Trial Using Matlab.
16. Rasmussen, E.M., Willett, P.: Efficiency of Hierarchical Agglomerative Clustering
17. Using the ICL Distributed Array Processor. Journal of Documentation 45(1), 1–24 (1989)
18. Li, X., Fang, Z.: Parallel Clustering Algorithms. Parallel Computing 11, 275–290 (1989)
19. Rasmussen, E.M., Willett, P.: Efficiency of Hierarchical Agglomerative Clustering Using the ICL Distributed Array Processor. Journal of Documentation 45(1), 1–24 (1989)
20. Li, X., Fang, Z.: Parallel Clustering Algorithms. Parallel Computing 11, 275–290 (1989)
21. Olson, C.F.: Parallel Algorithms for Hierarchical Clustering. Parallel Computing 21(8), 1313–1325 (1995)
22. Dean, J., Ghemawat, S.: MapReduce: Simplified Data Processing on Large Clusters. In: Proc. of Operating Systems Design and Implementation, San Francisco, CA, pp. 137–150 (2004)
23. Dean, J., Ghemawat, S.: MapReduce: Simplified Data Processing on Large Clusters. Communications of The ACM 51(1), 107–113 (2008)
24. Ranger, C., Raghuraman, R., Penmetsa, A., Bradski, G., Kozyrakis, C.: Evaluating MapReduce for Multi-core and Multiprocessor Systems. In: Proc. of 13th Int.
25. Symposium on High-Performance Computer Architecture (HPCA), Phoenix, A (2007)
26. Lammel, R.: Google's MapReduce Programming Model - Revisited. Science of Computer Programming 70, 1–30 (2008)
27. Hadoop: Open source implementation of MapReduce, http://lucene.apache.org/hadoop/
28. Ghemawat, S., Gobioff, H., Leung, S.: The Google File System. In: Symposium on Operating Systems Principles, pp. 29–43 (2003)
29. MacQueen, J.: Some Methods for Classification and Analysis of Multivariate Observations. In: Proc. 5th Berkeley Symp. Math. Statist, Prob., vol. 1, pp. 281–297 (1967)
30. Borthakur, D.: The Hadoop Distributed File System: Architecture and Design (2007)
31. Xu, X., Jager, J., Kriegel, H.P.: A Fast Parallel Clustering Algorithm for Large Spatial Databases. Data Mining and Knowledge Discovery 3, 263–290 (1999)