Cluster Computing (2019) 22:S12909–S12915
https://doi.org/10.1007/s10586-018-1799-6

Map Reduce for big data processing based on traffic aware partition and aggregation

G. Venkatesh[1] · K. Arunesh[1]

Received: 14 November 2017 / Revised: 28 December 2017 / Accepted: 9 January 2018 / Published online: 5 February 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract
Big data refers to data sets whose volume is 500+ terabytes of data per day. The velocity makes it difficult to capture, manage, process and analyze 2 million records per day. Another characteristics of big data is variability which makes it difficult to identify the reason for losses in i.e., images, audio, video, sensor data and log files etc., Hadoop can be used to analyze this huge amount of data using Hadoop an approximate early result for executing the job partially becomes available for the user even before completion of job which reduce the response time. In Layers 3 Traffic aware clustering programming model is used for processing big data which includes the data processing function map by sort and reducing techniques. The implementation of the layers three traffic aware clustering method will be on the top of Hadoop which is partitioned into HDFS fixed sized blocks and generates intermediate output as a collection of <num, data> pairs. The conventional hash function method is used for partitioning intermediate data among reduced task but it is not traffic efficient. In this paper to reduce network traffic cost, a Map Reduce task is done by designing data partition and aggregator that can reduce task merged traffic from multiple map tasks. The proposed algorithm is more efficient to reduce response time and the simulation results have showed proposal can reduce network traffic.

Keywords Traffic aware cluster · Data aggregation · Map Reduce · k-means cluster · Hadoop · Layers 3 traffic aware clustering

1 Introduction

Big data is an evolving term that describes combinations of larger amount of structured, unstructured and semi structured data that cannot be processed using conventional computing techniques [1]. Facebook, YouTube or social network have enormous amount of data that falls under the category of bigdata and it requires it to be collect and manage on a daily basis. The volume, velocity, variability of data sets makes them difficult to be captured, managed, and processed by traditional method and tools. Hadoop Map Reduce programming model is now being used for processing big data while Map Task performs the input data by partitioning into fixed blocks and generate the output as a collection of <num, data> pairs [2]. After completion of Map task these pairs are shuffled across different reduced tasks based on <num, data> pairs and each reduced task accepts only one num key at a time. The process data for that num key produces the result as <num, data> pairs. The Hadoop Map Reduce architecture [3] consists of one Job tracker performed at a time but many Task Trackers work simultaneously. Hadoop Map Reduce is the modified version of online Map Reduce which supports online aggregation [4] and reduces response time. Conventional Map Reduce method materializes the intermediated results of mapper and it does not allow pipelining between Map and Reduce tasks. This method has the advantage of simple recovery in case of failures. Once the mappers have finished the tasks, the reducers start executing the tasks.

Partitioning clustering allocate documents into a fixed number of clusters and the well known partitioning methods are k-means clustering [5] and its variants .The basic K-means clustering method [6] initially allocates a set of objects to a number of cluster randomly from which the mean

✉ G. Venkatesh
venkijmc@gmail.com

K. Arunesh
arunesh_naga@yahoo.com

[1] Department of Computer Science, Sri S Ramasamy Naidu Memorial College (Affiliated to Madurai Kamarajar University), Sattur, Tamil Nadu 626203, India

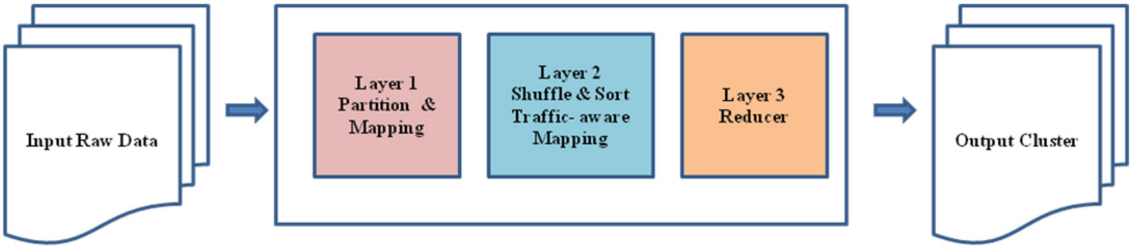

Input Raw Data
Layer 1 Partition & Mapping
Layer 2 Shuffle & Sort Traffic- aware Mapping
Layer 3 Reducer
Output Cluster

Fig. 1 Block diagram of the proposed layers 3 traffic aware clustering algorithm

of each cluster is calculated for each iteration and each object is reallocated to the nearest mean. In the Map phase, the mapper takes a single (num, data) pair as input and generate any number of (num,data) pairs as output. The map operates as stateless in which the logic operates on a single pair at a time even if in practice several input pairs are delivered to the same mapper. To recapitulate for the map stage, the user simply designs a map function that maps an input <num, data> pair to any number of output pairs. Frequently the map phase is used to specify the desired location of the input by changing its key. The shuffle stage is automatically handled by the Map Reduce technique and the engineer has nothing to do for at this stage. The system implementing Map Reduce routes all values that are in aggregate with an individual key to the same reducer. In the reduce stage, the reducer takes all of the values associated with a single num n and outputs any number of <num, data> pairs. One of the sequential aspects of Map Reduce computation is that it reduces all of the maps before the reduce stage can begin. The reducer has access all the values with the same key and it can perform sequential computations. The parallelism is exploited by observing that reducers operating on different num keys can be executed concurrently. Document clustering [7] performs collection of similar documents into classes where similarity is some function on a document and it does not need separate training process, where the documents in the same clusters are similar and the documents in the different clusters are not relevant.

In this paper, a new proposal algorithm is introduced by layers 3 traffic aware clustering programming model. Figure 1 illustrates the block diagram of the proposed system, the design, and implementation of the massive datasets, the processing model based on Map Reduce and Hadoop.

2 Problem identification

Big data is a complex enormous amount of datasets which has become a buzz in the market, due to various challenges and major issues to be handled in the big data. Map Reduce [8,9] is the prominent technique for processing the enormous amount of data by designing and develop the new method for clustering. In the map phase, it performs the task of converting the input into the form of <num, data> after which the output is sorted and shuffle in the shuffle phase of second layer in which is again sorted at the aggregator stage [10]. The third layer is the reducing phase, which performs the task fetching from its own share of data partitioned from all map tasks to produce the final solution. Here the clustering technique for the map phase is used where big data is analyzed and cluster of similar objects are formed. In this three layer method, the data produced by the map phase are ordered, partitioned and aggregated to the appropriate machines which executes the reduce phase. The network traffic [11] from all the three layers can cause a great volume of network traffic, which causes a serious constraint on the efficiency of data analytic applications. One of the problems in the network traffic is difficult in the process of data with given time.

3 Methodology

3.1 Layers 3 traffic aware clustering method

This proposed method is a layers 3 traffic aware clustering algorithm based on parallel K-means [3,12] and distance metric [13]. Traffic aware clustering combines the benefits of both the K-means and distance metric algorithms. The benefit of parallel K-means is that it is not complex and forms clusters using the distance metrics and traffic aware and so it has less execution time. This proposed system works in 3 layers.

Layer 1: In this layer, the data is partitioned into different clusters and different parts to minimize into the boundary points [14].

Layer 2: This layer performs shuffling and sorting the clusters based on traffic aware.

Layer 3: This performs sorting that is the input is fetched to reduce the clusters. The clustering is a function of grouping the set of objects that are combined into a single cluster and this cluster contains only similar objects. The dissimilar objects are formed into another cluster. The goal of the clustering is to partition unstructured data objects into a structured data objects. K-means clustering algorithm is used to partition the different sets of data into clusters and it is n-objects based on attributes into k-partitions where k>n to form a vector space. The work begin with a decision on a value of k = number of clusters. First the data is partitioned into k clusters and then each data point is taken in sequence and the distance from the centroid of the clusters is calculated. If the data points which are not closer to the centroid are present in the cluster then those data points are switch that clusters. This process is repeated until the centre does not change. The system processes the given input data, where the data is shuffled and reduced into small required data. To reduce the network traffic within a Map Reduce task [11,15], the aggregate data is considered the aggregate data with similar keys before sending the mapper output to the reducer stage. The data aggregation creates opportunities for multiple tasks on different machines thus goal of minimize the network traffic [16,17] by data partition and aggregation for Map Reduce task. Layers 3 traffic aware clustering algorithm is proposed for big data applications and the final result demonstrates the reduction in the network traffic cost.

3.2 Layers 3 traffic aware clustering algorithm

Step 1: Input D: data set having n data points. A= {A_1, A_2, A_3 . . . A_n} be the set of data points and V= {V_1, V_2, V_3, V_4…V_n} be the set of centers.

Step 2: Map Reduce function executes in three stages ie. Map stage, shuffle stage- traffic aware and reduce stage.

Step 3: Iteratively improves partitioning of data into k document clusters.

Step 4: Mapper Phase

(i) Input in map function paradigm is based on sending to where the data resides and it is in form of file or directory and is stored in the Hadoop distributed file system (HDFS).

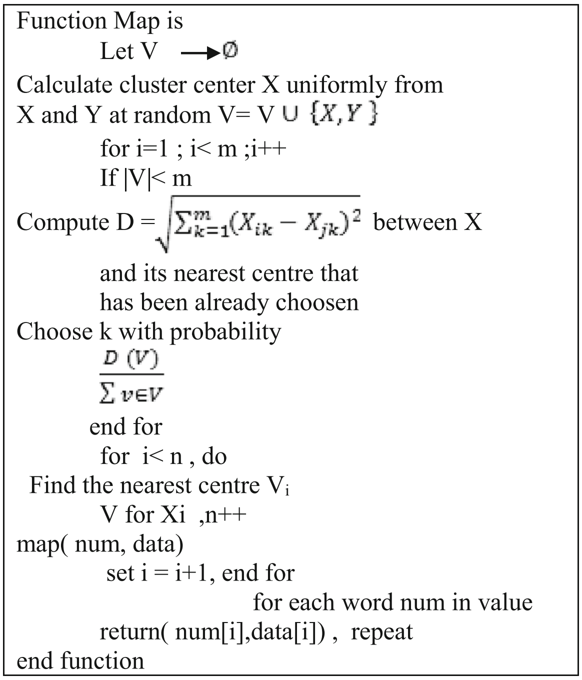
(ii) Calculate the distance between each data point and cluster centre using the distance metrics as follows.

$$D_{XY} = \sqrt{\sum_{k=1}^{m} \left(X_{ik} - X_{jk}\right)^2} \tag{1}$$

(iii) Data point is assigned to the cluster centre whose distance from the cluster centre is minimum for all the cluster centers.

(iv) The input passes to mapper function line by line and in the form is <num, data> pair as cluster centre and data points.

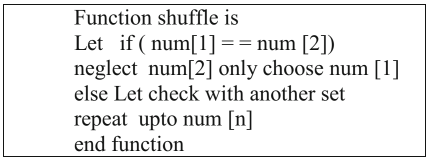
(v) The mapper function finds the nearest centre among k centers for the input point and process the data and creates small chucks of data.



```
Function Map is
          Let V $\longrightarrow \emptyset$
Calculate cluster center X uniformly from
X and Y at random V= V ∪ $\{X,Y\}$
          for i=1 ; i< m ;i++
          If |V|< m
Compute D = $\sqrt{\sum_{k=1}^{m}(X_{ik}-X_{jk})^2}$ between X
          and its nearest centre that
          has been already choosen
Choose k with probability
          $\frac{D\ (V)}{\sum v\in V}$
         end for
          for  i< n , do
 Find the nearest centre $V_i$
          V for Xi  ,n++
map( num, data)
           set i = i+1, end for
                              for each word num in value
          return( num[i],data[i]) ,  repeat
end function
```

Algorithm 1.Algorithm for Mapper Phase (Layer 1)

Step 5: Shuffle stage- traffic aware



```
Function shuffle is
Let   if ( num[1] = = num [2])
neglect  num[2] only choose num [1]
else Let check with another set
repeat  upto num [n]
end function
```

Algorithm 2. Algorithm for Shuffle stage - traffic aware (Layer 2)

Step 6: Reducer Phase

(i) Reducer phase is the combination of shuffle and reduce and the input of reducer's job is to process the data and produces the output of map function that is all data point creates a new set of output.

(ii) New cluster centre is calculated using

$$V_i = \left(\frac{1}{C_i}\right) \sum_{j=1}^{C_l} X_j \tag{2}$$

C_i = No. of data points in the ith cluster

(iii) The reducer phase calculates the new centre using data points that are stored in HDFS.

```
Function Reduce is
Let V ⟶ Ø
Calculate new cluster center X uniformly
V= V ∪ {X,C }
for i=1 ; i< n ;i++
 If |V|< n
$V_i = \left(\frac{1}{C_i}\right)\sum_{j=1}^{c_i} X_j$
for i< n do
reduce ( num, data)
data =A[i] , a list of counts
int result =0
set i = i+1
end for
for each data A[i] in values
result (cluster)
repeat
end function
```

Algorithm 3. Algorithm for Reducer Phase (Layer 3)

Figure 2 shows the proposed method flow chart. In this system the input data is processed and partitioned to calculate the distance between the data points. The next step is the formation of clusters from which the cluster centre is calculated. The shuffling and sorting is also done at this stage which performs multiple tasks on different machine to check whether the same numbers are combined together before sending them to reduce tasks. In the last stage the sorted output is given to the input of the reduce function where computation of centre co-ordinate new cluster is done before the process is ended.

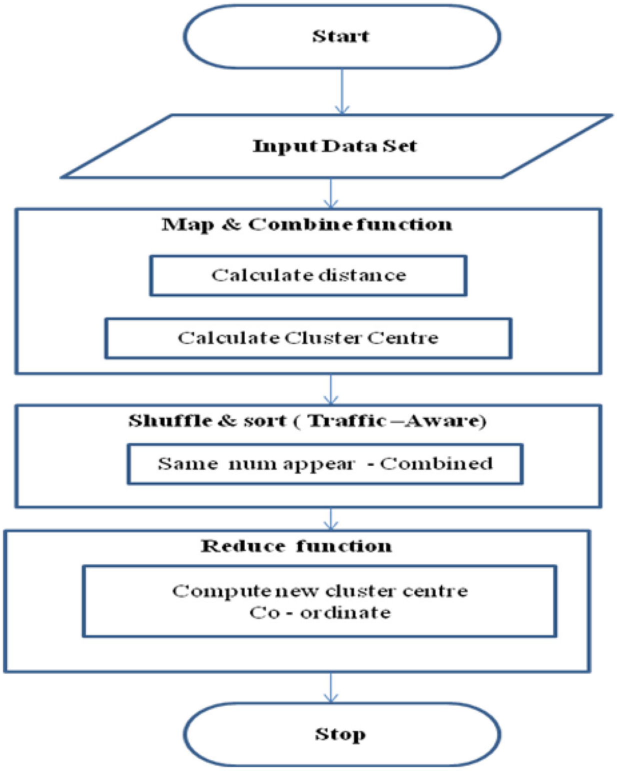


Fig. 2 Proposed layers 3 traffic aware clustering system flow chart

3.3 Data partitioning

The data partitioning [18] is done in the Map task function. The proposed system includes the logical block addressing clusters in Map function at this stage because multiple tasks content for the same slot wastes the time for re allocating the data. The sorted list in each list contains the block of the same partition and they are split into different array list then a new a new cluster repeated data into the partition. Figure 3 shows the data partitioning.

4 Implementation results

The system is developed in JAVA and deployed in Hadoop framework. The results obtained from the implementation of layers 3 traffic aware clustering algorithm are compared with bisecting K-means algorithm and K-Means parallel algorithm using the same data sets. The data sets are “Reuters 21578” [19], “Online 20 Newsgroup” [20], “Web Ace” [21], “TREC” [22]. The algorithm is performed on the basis of two parameters namely network traffic related execution time and performance accuracy. Table 1. Illustrates the comparison of execution time of various methods using different data sets.

Figure 4 shows the performance using execution time of the proposed algorithm compared with four other algorithms namely Basic K-means, K-Means Parallel, Bisecting K-Means and DB-scan with the same data sets. The proposed algorithm formulates the network traffic minimization and facilitates to construct three layer clustering algorithm. In the intermediate layer, a potential aggregator is created at each machine so that aggregation of data can be done from all mappers. Thus the proposed system efficiently reduces the network traffic cost.

Figure 5 shows the system accuracy of the proposed system tested using the data in Table 2 and compared with various algorithms namely Bisecting K-Means, K-Means Parallel, Basic K-Means, DB-Scan on the same data sets. Accuracy is calculated using this formula

$$\text{Accuracy} = \sum\nolimits_i P_i \left(max_j \left(\frac{P_{ij}}{P_j} \right) \right) \tag{3}$$

Accuracy is mainly calculated for the quality of clusters and Fig. 5 shows the comparison of the various algorithms. The

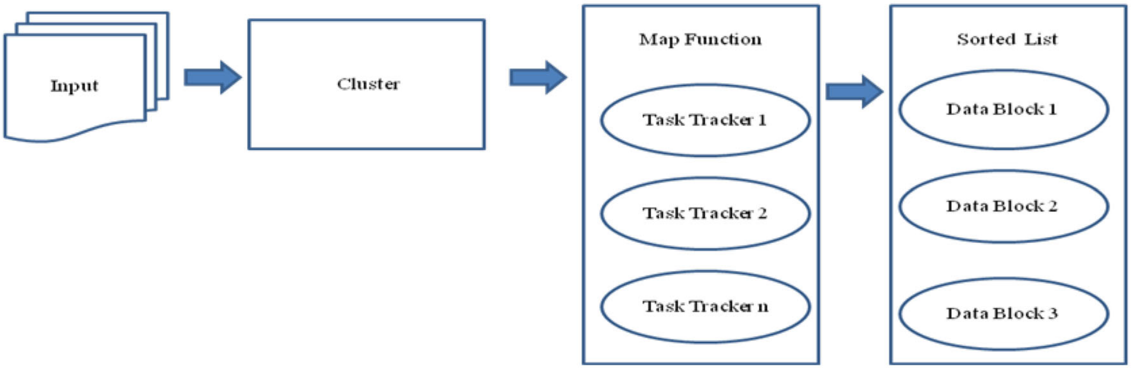


Fig. 3 Data partitioning

Table 1 Comparison of various methods with four different data sets execution time

Data sets	Basic K-means [13]	K-means parallel [23]	Bisecting K-means [24]	DB-scan [4]	Proposed method
Reuters-21578 [19]	43.5136	41.231	41.0123	43.2154	35.1325
20 Newsgroup [20]	35.3689	34.1265	32.1021	36.1235	33.1234
Web Ace [21]	25.3218	26.3521	20.1251	24.3654	19.3251
TREC [22]	15.6619	29.3214	22.3654	19.3256	14.1645

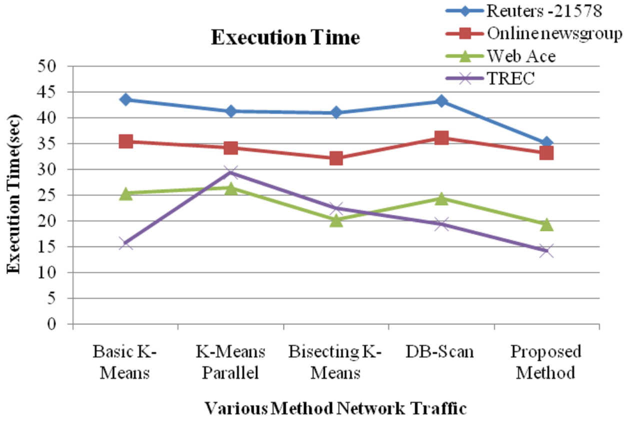


Fig. 4 Performance of the algorithms using execution Time

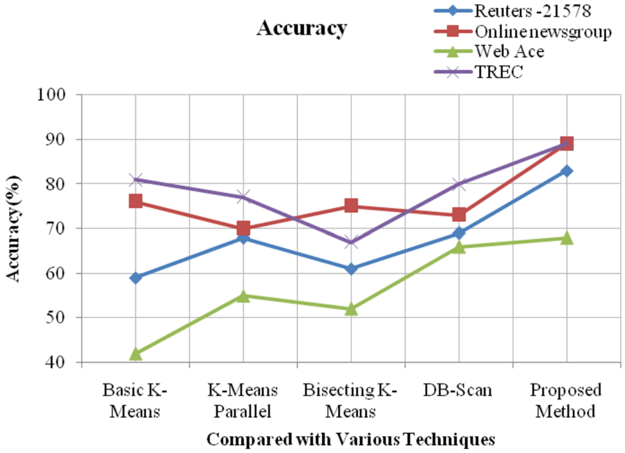


Fig. 5 Comparisons of various techniques with accuracy

performance graph shown in Figs. 4 and 5 clearly demonstrates that the proposed algorithm better in performance than other algorithms.

5 Conclusion

In this system, layers 3 traffic aware clustering algorithm is proposed as an efficient traffic aware partition and aggregate to minimize the network traffic cost. Big data provides background of various clustering technique which can be used to analyze data and design three layered structure to solve the problem. The comparison the various algorithms like Bisecting K-Means, K-Means Parallel, Basic K-Means, DB-Scan with the proposed system were done on the same data sets to calculate their execution time and accuracy. The implementation results had been tested using four data sets and the obtained results demonstrates that the proposed system is more accurate. Same data groups from multiple tasks on different machines were combined before sending them to reduce tasks. The usage of the proposed algorithm shows better performance by reducing the network traffic using partition, aggregation and reduces the network traffic.

Table 2 Comparison of various techniques accuracy %

Data sets	Basic K-means [13]	K-means parallel [23]	Bisecting K-means [24]	DB-scan [4]	Proposed method
Reuters-21578 [19]	59	68	61	69	83
20 Newsgroup [20]	76	70	75	73	89
Web Ace [21]	42	55	52	66	68
TREC [22]	81	77	67	80	89

References

1. Vadivel, M.: Enhancing map-reduce framework for big data with hierarchical clustering. Int. J. Innov. Res. Comput. Commun. Eng. **2**(Special Issue 1) (2014)
2. Vidya, P.: Big data hadoop: aggregation techniques. Int. J. Sci. Res. (IJSR). ISSN (Online): 2319–7064 (2014)
3. Lena T. Ibrahim, Rosilah Hassan, Ahmad, K., Asat, A.N.: A study on improvement of internet traffic measurement and analysis using Hadoop system. In: The 5th International Conference on Electrical Engineering and Informatics, 10–11 Aug, 2015, Bali, Indonesia (2015)
4. Dhanalakshmi, R., Mohamed Jakkariya, S., Mangaiarkarasi, S.: Aggregation methodology on map reduce for big data applications by using traffic-aware partition algorithm. Int. J. Innov. Res. Comput. Commun. Eng. **4**(2) (2016)
5. Ahammad Fahad, S.K., Alam, M.M.: A modified K-means algorithm for big data clustering. In: IJCSET April 2016, vol. 6, Issue 4, pp. 129–132 (2016). www.ijcset.net
6. Abubaker, M., Ashour, W.: Efficient data clustering algorithms: improvements over K means. Int. J. Intell. Syst. Appl. **03**, 37–49 (2013)
7. Shah, N., Mahajan, S.: Document clustering: a detailed review. Int. J. Appl. Inf. Syst. (IJAIS) **4**, 30–38 (2012)
8. Krishna Mohan, K.V.N., Prem Sai Reddy, K., Geetha Sri, K., Prabhu Deva, A., Sundarababu, M.: Efficient big data processing in Hadoop MapReduce. Int. J. Adv. Res. Comput. Sci. Softw. Eng. **6**(3) (2016)
9. Sripada, S.C., Sreenivasa Rao, M.: Comparison of purity and entropy of K-means clustering and fuzzy C means clustering. Indian J. Comput. Sci. Eng. (IJCSE) **2**(3) (2011)
10. Ke, H., Guo, S., Guo, M.: On traffic-aware partition and aggregation in map reduce for big data applications. IEEE Trans. Parallel Distrib. Syst. Cit Inf. (2015). https://doi.org/10.1109/TPDS.2015.2419671
11. Gawande, P., Shaikh, N.: Improving network traffic in MapReduce for big data applications. In: International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT). IEEE (2016)
12. Suganya, G.: An efficient network traffic classification based on unknown and anomaly flow detection mechanism. Int. J. Comput. Trends Technol. (IJCTT) **10**(4) (2014)
13. Singh, A., Yadav, A., Rana, A.: K-means with three different distance metrics. Int. J. Comput. Appl. **67**(10), 0975–8887 (2013)
14. Li, T., Ma, S., Ogihara, M.: Entropy-based criterion in categorical clustering. In: Proceedings of the 21st International Conference on Machine Learning, Banff, Canada (2004)
15. Shim, K: Map reduce algorithms for big data analysis. In: The 38th International Conference on Very Large Data Bases, August 27th 31^{st} 2012, Istanbul, Turkey, Proceedings of the VLDB Endowment, vol. 5(12) (2012)
16. Vijayalakshmi, G.: Large scale optimization to minimize network traffic using map reduce in big data applications. In: 2016 International Conference on Computation of Power, Energy Information and Communication (ICCPEIC)
17. Reddy, Y.D., Sajin, A.P.: An efficient traffic-aware partition and aggregation for big data applications using map-reduce. Indian J. Sci. Technol. ISSN (Print): 0974–6846 (2016). https://doi.org/10.17485/ijst/2016/v9i10/88981
18. Neelakandan, S., Divyabharathi, S., Rahini, S.: Large scale optimization to minimize network traffic using mapreduce in big data applications. In: 2016 International Conference on Computation of Power, Energy Information and Communication (ICCPEIC)
19. https://archive.ics.uci.edu/ml/datasets/Reuters-21578+Text+Categorization+Collection
20. http://qwone.com/~jason/20Newsgroups/
21. https://catalog.ldc.upenn.edu/LDC2006T06
22. http://trec.nist.gov/data.html
23. Ping, Z.H.O.U., Jingsheng, L.E.I., Wenjun, Y.E.: Large-scale data sets clustering based on MapReduce and Hadoop. J. Comput. Inf. Syst. **7**(16), 5956–5963 (2011)
24. Steinbach, M., Karypis, G., Kumar, V: A comparison of document clustering techniques. KDD workshop on text mining, vol. 400(1) (2000)

G. Venkatesh has completed his under graduation B.C.A. from Urumu Dhanalakshmi College, Bharathidasan University, Trichy, his post graduation M.C.A. from Urumu Dhanalakshmi College, Bharathidasan University, Trichy, B.Ed. from Oxford College of Education, Teacher Education University, Trichy and M.Ed. from Jeevan College of Education, Teacher Education University, Trichy, Second Class M.Phil. (Computer Science) from Jamal Mohamed College (Autonomous), Bharathidasan University, Trichy. He is presently research scholar at Sri S Ramasamy Naidu Memorial College (Affiliated to Madurai Kamarajar University), Sattur–626203, Virudhunagar District. His current area of interest includes big data analysis and Image processing.

K. Arunesh has received Ph.D. degree in Computer Science from the Bharathidasan University, Tiruchirappalli, Tamil Nadu, India. He is currently an Associate Professor of Computer Science and Dean, Academic Affairs at Sri.S.R.N.M College, Sattur. His current research interests include Knowledge Discovery in Databases, Big Data Analytics, Data Mining and Recommender Systems. He has published widely in leading journals and conference proceedings, and served program committees as conference Chair person and Convener. He is currently on the editorial boards and serves as reviewer of leading journals and conferences.