VOL 2 (2018) NO 3

e-ISSN : 2549-9904
ISSN : 2549-9610

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

Enhanced Big Data Platform for Visualization of Employee Data

Mr.Manishankar S [#], S. Sathayanarayana [#]

[#] *Department of Computer Science, School of Computer Science & Engineering, Bharathiar University, India*
E-mail: manishankar1988@gmail.com

***Abstract*— In this Digital world storage area capacity required for an Enterprise is quite huge, and processing that Big Data is one of the major challenging areas in today's information technology. As the heterogeneous data from the various sources grow rapidly, there should be some proficient way for data storage for each enterprise. Most of the Enterprises have a tendency to migrate their data in to servers with high processing capability to handle variety and voluminous data. Major problem that arises in such big data servers of an Enterprise is the process involved in segregating data according to their types. In this research, an efficient methodology is proposed which handles the segregation of data inside a server with multi valued distribution-based clustering. These clustering-based solutions provide an efficient visualization of varying data in the server and also a separate visualization of employee data too. The paper discusses about the simulation of the clustering technique with respect to an Enterprise data and visualization of file storage structure and categorization of data, also it gives a picture of performance of the Big data server.**

***Keywords*— Big Data, Data storage servers, Data Analytics, Clusterization, Visualization.**

I. INTRODUCTION

The most required thing for an Enterprise is the recurring data that increases day by day and storing the data in a well-arranged manner so that an efficient and fast retrieval of information can be there for the later requirement of the users [1]. Storage and easy retrieval is usually not a difficult task in case of small amount of data. But when we consider the big data within an organization which consist of huge and varied amount of data it would be highly tedious and more time-consuming task [2]. Clustering is one of the general methods used for mining the data. Clusterization is the technique of grouping the item to one cluster where the items in one group are different from items in another group.

Clusters are formed with calculating smaller distances between data instances, dense area of the data space, intervals amongst instances and varying statistical distributions [3]. Grouping up the similar kind of data is useful in data management. We are all familiar with the difficulty in finding some particular item from a box contains jumbled items. To get one particular item from such a box with some desired property like its colour, size is quiet weary task. Considering a situation where the cluster or group of items based on one of the properties like colour, size etc. while storing the items in the box, there arises a need of a separate storage space for each group of similar items. At this point we can get our desired item from the box very easily since the diversification of the data according to the criteria is taken care.

Big Data has a huge built structure and is characterized by four typical features: data volume, escalating velocity mounting mixture of data types and structures, and rising unpredictability of data [4]. Data are is produced from various sources of information systems. Data sources include data warehouse, data marts, and servers, from Adhoc-sensor nodes. Heterogeneous files supplied from sources are distinct in their attributes making it difficult to store it in a good structure for easier retrieval with less time for searching for the items in the storage space. A system is being developed which helps in identifying the data property, priority, user control levels. Storing is done with creation of a zone based structure space where appropriate files are stored that leads to an advantage of retrieving the file for the users as per their customized requirement. Role based user profiling is an important criterion while partitioning the files. The abstraction of the view level of the file or data is correlated with the profile of the user. The file structure details are stored in the cloud and every entry is stored with corresponding entries of the list stored in the cloud server. The retrieval is based on unique requirement originated by the user with specific attribute generated. In such a situation the files of user desired format retrieval is a heavy task for the server from the jumbled storage. At the interface level, an algorithm that separates the file type extensions. The output is passed to the next higher level where user access control parameters are summated to the file type extension value. A hierarchical indexed based file table is constructed with values of keywords and priority level of the user. This table is designed with the help of a key function which calculates index values based on values passed from supporting algorithms [5].

Analysis of the file structure can be understood from the visualization that is plotted using cluster values obtained from the system. Visualization is a technique provided for monitoring the stored file's amount in the storage space. This aids the server manager with efficient utilization of resources in case of an assorted data. Usually pictorial representation like charts enhances the understanding level of the file.

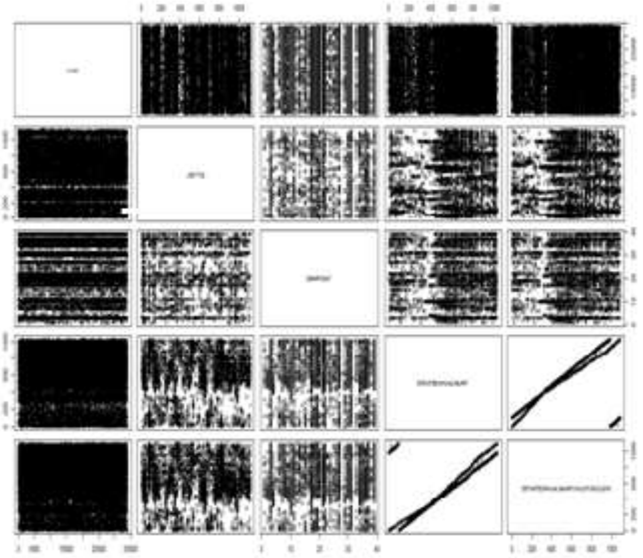

Fig 1. Visualization of file clustering in existing models (in R)

II. EXISTING TRENDS OF RESEARCH

First, Files and data is stored in the servers in the form of raw bits and bytes and no procedure to segregate the data according to the format they occur. The file format may be in various types like audio, video, image, text and multimedia. In the existing real-world data warehouse, crawler is used in moving among diverse data which doesn't have any order while storing is done. [6] Due to structure of the data stored, there is no proper visualization on the analysis of data. No special parameter is considered to categorize data in the present servers in the organizations. All data that is being uploaded will be stored together into a single repository. This results in efficient retrieval of data. This storage without segregation has higher effect on performance as the organization size and data warehouse size increases. More over there is no visualization provision that is given to the administrator to manage data efficiently. There is no adequate algorithm to process the division of header files. Hence it is difficult to compartmentalize the storage structure. When it comes to a file structure with huge number of file tables, the schema at the physical level will be much more complex to handle for a huge data. In case of big data servers Petabytes of data are input with varied formats and consumes enormous processing time. Considering a system like Mongodb which is a semi structured data for managing document oriented information. Here the data are encapsulated into a standardized format and then a function is given to associate each document with the internal structure. Mongodb drivers use GridFS function which divides files into chunks and store them into separate documents[7]. It is a distributed file system that provides multiple copies of files among separate machines to provide balanced load and fault tolerant .Map reduce concept is the basic need for the processing large data. Map reduce walkthrough with reducer and partitioner spends a longer time in splitting files into chunks[8], and more over chunk retrieval and processing function are associated with delay and overhead[9]. In the context of an organization where there is huge diverse data, the handling process similar to that of mongodb is a tedious task. In the existing system there is a lack of an algorithm which performs adequate organizing of file structure according to extension of file types. Always a optimized storage function can be a better solution.[10]confirm that you have the correct template for your paper size.

Hadoop based analytic approach has a growth in its dimension now a day [11]. It gives a scalable platform to improve the results in analytics by faster and better computations [12]. It also offers a merging of various analytic platform together [13].it has a much-improved performance capability offered with a single node single cluster as well as multi node multi cluster [14]. Big data has MAP-REDUCE conceptualization improving the data storage in a high scale [15].

III. PROPOSED SYSTEM MODEL

The system is created with an aim to process large diverse data of an organization. The preliminary step is to create a file storage structure which imparts balanced load and easier retrieval. As a part of pre-processing of input, the various properties of data are being analyzed and parameter wise they are stored into table. For the above step, there is a requirement for an interface which interacts with user and gains information about extension, size, type and required user preferences. There is a level of unsupervised learning algorithm which is involved in scrutinizing data. According to the customized requirement of the organization; the file table is created with a zone based vector space in which zones are formed according to the parameters analyzed from the information processing. It is already proven that when data is hierarchically organized there is a need for clustering which helps mapping data to corresponding groups for easier handling of information. Various parameters for clustering is considered based on organizational storage policy of data. File table have associated value [6]s of keywords and priority levels of the user. A mapping function is being formulated which consists of keywords formed from parameters to map the storage space to corresponding cluster they belong in a hierarchical way. The keyword and index types are organized with the help of an algorithm that comes as a part of preprocessing and analysis. Inside the data warehouse each user is provided with a separate profile account based on verifying credentials. The correlation task is carry forwarded to map user profile with the file policy. The algorithm designed segregates the file extension types and a dynamic symbolic clustering is performed. It accepts a huge amount of data from various sources. The data can be of heterogeneous format. The data is stored on the server with a clustered manner and the system will provide visualization for monitoring the data to the database administrator.

IV. Methodology

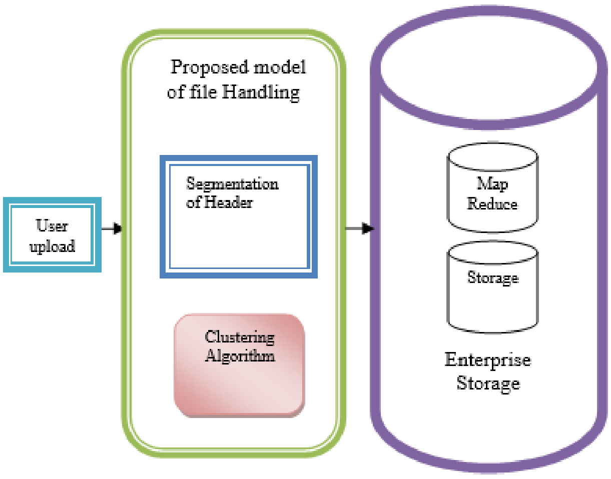

Fig 2. Architecture of the System

The architecture is composed of various modules as depicted in the above diagram. The first module is the client which consists of an API that facilitates the user provision to upload his/her private and public data that he maintains for the organization. The file is fed into a preprocessing module where header of each file is separated and analyzed, threshold value for file is generated and this leads to the segregation of files according to types. The clustering sub module takes the input from the analyzed header where each parameter that led to the division of file is recorded and with the help of an efficient clustering algorithm categorized into sectors. These sector values are mapped into corresponding storage area for the user in correspondence with the map reduce processing.

The system is composed of a large data repository [16] designed for an organizational storage purpose. The repository is designed in a way to accumulate files of various formats. Each user is provided with a separate zone based storage structure which is again forms hierarchical order. Each hierarchical level of order is indexed in the file table structure. The file table is composed of i-node number, type, category, storage time stamp and special index key for traversing through the file table. The index key is formed as a combination of vector space id and i-node number. Time stamp is generated with addition and updating of each file. This keeps the track of transaction of records. A function is derived for correlating file table and the storage structure space. Each zone represents user profile. A value associated with user profile provides the vector space id. The user profile may be a set of security credentials and file policy details which ease the design of sectoring of warehouse. User profile withholds elementary details such as username and password. Each user is given a storage sector which makes the retrieval possibly faster. The storage sector of each user is again divided into sub sectors each composing of various type and categorical data stored by the user. The policy details are framed when the user is being added as a part of the organization which provides the details of file access and retrieval of file structure. File policy details consists of file mode , file path and file group. File group may be individual or group users. The initial division of the data takes place from analyzing the policy of organization in the file system. This step consists of building the file system structure which is distributed into individual files and shared files. Each individual file structure consists of sectors which specify type and category of user data.

There is an Application Programming Interface which helps in taking the input from the user which support the decision structure in categorizing user files. Each time when the input is taken certain constraints are considered taking recommendations from the user[17]. These parameters are fed into a spread sheet where each column and row depicts the instance of data according to users threshold values. The variations in data types are being recorded and machine is under the learning process to generate a threshold value. These values alleviate the task of segregation between data items of similar category. The threshold values are fed into a clustering algorithm which categorize them into different clusters and then a mapping function correlates them to corresponding sector id of the warehouse. If the sector is not existing then it is created. As initially the user registers, the user's private data sector is created according to their profile. It is compartmentalized with respect to the general rules and later customized according to the requirement. The file table is updated with the sector ids as and when it is created. It denotes the status of transaction values supplemented with the time stamp.

ALGORITHM

```
Big Data Cluster Algorithm (D, F, T, C, I, N)

D: Data F: File T: Type C: Cluster  I: Values N: Total number of files of same type

Step 1: Choose a File F.

Step 2: Pre-process F to read the file header for information about File Type.

Step3: Store the file type into value T.

Step 4: Create a data folder D mapped with type T

Step5: calculate the value for the centroid of cluster taking

    $ΔG=(ΔT+ΔD)/N$                                   (1)

Step 6: Divide the Data according to centroid value with help of a clustering algorithm into cluster C\

Step 7: Calculate the distance between the centroid and the value of the file mapped on to the space given by the formula

    $Ii= \sum i=1n (X-Xi)^{2}$                       (2)

Step 8: Calculate value for each cluster where

        Cluster Id is I

Step 9: Use the map reduce function to store each cluster C mapped onto data folder D.

Step 10: Generate the visualization
```

We have developed a web application where we implemented our new concept to store the big data securely and efficiently. For this we have developed a JSPweb application where the admin and multiple users of the system can login simultaneously from different system. Our actual data which are uploaded by the users will be stored in Big Data server; here we are using MongoDB as our main database. We have also used a ordinary RDBMS to store the properties of the data uploaded by the users for the pre-processing steps. These properties are extracted from the actual data. So our system is a combination of jsp, one ordinary RDBMS and MongoDB. [18]. The functionalities are as given below. First of all administrator have login to the system to add the users to register with name, user ID and password. By using this user ID and Password user can login to the system.

After the successful login all the registered users can get their account where many features are available including upload their data of any type to the server. The additional features are, the users can make the data into 2 broad categories protected and public. A file uploaded with protected option that file will be available only for the user, whereas publically uploaded file will be shared with other users. And one more additional feature is user can make their own categories like entertainment, sports and educational etc. The files uploaded under these categories will be shown to the user in the future in this category only. And the last feature as common for all the service providing web site we give a feedback portal for the users.

Now let us look what is happening at the backend, when a user upload a file from their system that file will be taken to the server as it is, After the file reaching to the server pre-processing starts. As many users may login and upload the files simultaneously to the system there will be a bulk amount of data. In the pre-processing step every files accessed separately and extract it properties like file format, file size, path etc. These data will be stored in an ordinary RDBMS, here we use MS Access. The stored file properties are used to cluster the files based on their type, i.e. each type of files will be gathered together. After this clusterization on the file properties system will generate a dedicated collection in Mongo DB for each type. If there is no such dedicated collection for a particular type system will create new and store the file in server as chunks of data. If the collection of that particular type already exists in the server then the data will be stored in that collection as chunks. As the data in the server are stored in the form of chunks there is very less probability for data getting corrupted or modified by itself. As the data stored in the server in a clustered manner the efficiency of data retrieval would be high.

Now let's come to the admin portal, The main responsibility of the admin is to maintained the server properly. So he should be able to monitor the server in very easy way. For this we are providing the admin with visualization, where the amounts of data uploaded by the users are represented in a graphical way. By monitoring this the admin can take the decision to keep a separate storage server for any particular type of data which is updated heavily by many users. And from the security perspective the admin can easy track any malware or threat files stored in the server. There will be a separate indication for the newly arrived and isolated file in the visualization. So that admin can take appropriate decision to handle this file. and the last feature is to review the feedback from the users to improve the service

V. Implimentation Details

The Big data processing model derived here is a specific model for an organization whose data processing is quite huge, which can be considered as prototype for Big data. The major aim of the company is storage of the files and other data that each employee generates for their company work in to a distributed server. As we know that in a company there will be varying data generated as user's profiles vary, and as server grows distributed it will be difficult for the retrieval of data, processing capacity gets lower causing system to crash even. Analysis of company requirement was carried out and details of the data required where collected, the plotting of data to a distributed server was analyzed and found to be feasible for deploying in to a server. Choice of the server made and a Mongo Db server as setup and it was connected to the storage application .user interface was such way that he could give preferences and choices of his request. We have developed a web application where we implemented our new concept to store the big data securely and efficiently. For this we have developed a jsp web application where the admin and multiple users of the system can login simultaneously from different system.

Our actual data which are uploaded by the users will be stored in Big Data server; here we are using MongoDB as our main database. We have also used a ordinary RDBMS to store the properties of the data uploaded by the users for the pre-processing steps. These properties are extracted from the actual data. So our system is a combination of JSP, one ordinary RDBMS and MongoDB. The functionalities are as given below. First of all administrator have login to the system to add the users to register with name, user ID and password. By using this user ID and Password user can login to the system.

After the successful login all the registered users can get their account where many features are available including upload their data of any type to the server. The additional features are, the users can make the data into 2 broad categories protected and public. A file uploaded with protected option that file will be available only for the user, whereas publically uploaded file will be shared with other users. And one more additional feature is user can make their own categories like entertainment, sports and educational etc. The files uploaded under these categories will be shown to the user in the future in these categories only. And the last feature as common for all the service providing web site we give a feedback portal for the users.

Now lets look what is happening at the backend, when a user upload a file from their system that file will be taken to the server as it is, After the file reaching to the server pre-processing starts. As many users may login and upload the files simultaneously to the system there will be a bulk amount of data. In the pre-processing step every file accessed separately and extract it properties like file format, file size, path etc. These data will be stored in an ordinary RDBMS, here we use MS Access. The stored file properties are used to cluster the files based on their type, i.e. each type of files will be gathered together. After this cauterization on the file properties system will generate a dedicated collection in Mongo DB for each type. If there is no such dedicated collection for a particular type system will create new and store the file in server as chunks of data. If the collection of that particular type is existing in the server then the data will be stored in that collection as chunks. As the data in the server are stored in the form of chunks there is very less probability for data getting corrupted or modified by itself. As the data stored in the server in a clustered manner the efficiency of data retrieval would be high.

The data details are analyzed and stored in to a CSV file format and fed in to mining module which is helpful for cauterization of the data. So it helps the segregation of the files and data easily, here we use a learning algorithm as we have explained in methodology part, this helps in giving a better structure to the data storage. the many files which fall in to a cluster are put in to single track of storage sector. the storage structure is really dynamic as we consider the change in user preference as we take in starting. Each user's storage structure is completely different as they want they can model it.

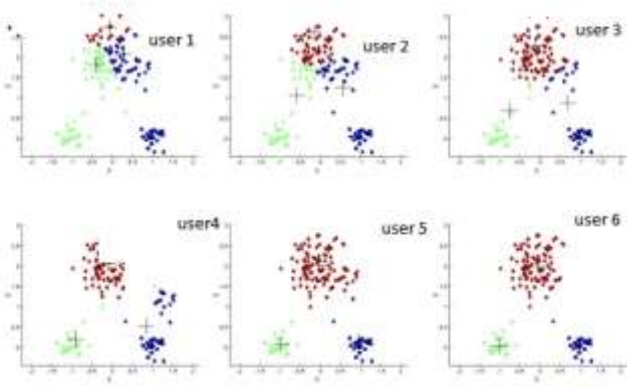

Fig 3. Cluster of Data profile of various user.

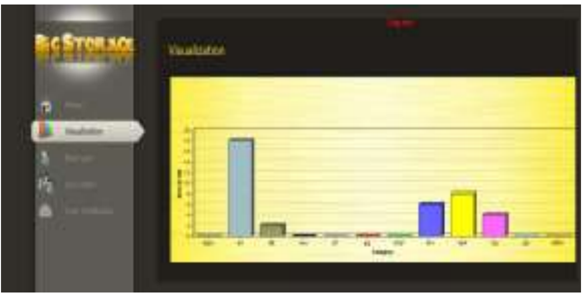

Fig 4. Single user storage profile

Now let us come to the admin portal, The main responsibility of the admin is to maintain the server properly. So he should be able to monitor the server in very easy way. For this we are providing the admin with visualization, where the amount of data uploaded by the users is represented in a graphical way. By monitoring this the admin can take the decision to keep a separate storage server for any particular type of data which is updated heavily by many users. And from the security perspective the admin can easy track any malware or threat files stored in the server. There will be a separate indication for the newly arrived and isolated file in the visualization. So that admin can take appropriate decision to handle this file. and the last feature is to review the feedback from the users to improve the service

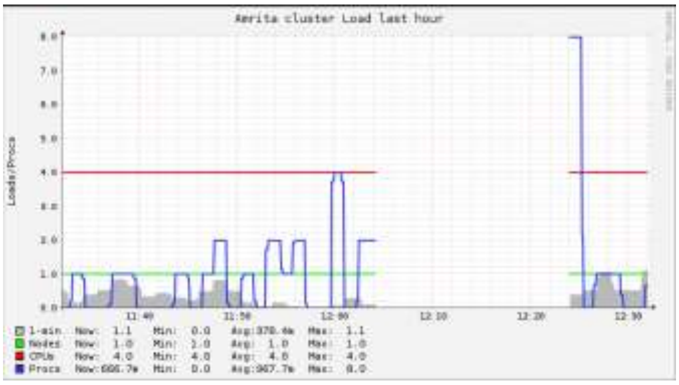

Fig 5. Load of the Big data system reduced

The load of the big data system is observed to be reduced due to the effect of the cluster management system which is powered by the clustering algorithm, which divides data in to varies types and does a job-oriented data analysis. Thus, system has visible improvement when compared to existing YARN based schedulers.

VI. Conclusion And Future Works

In this paper, we are proposing a method for storing the big data in an efficient manner by making use of clusterization technique. The data of same type will be stored together. This will help to improve the efficiency of retrieval of files for the user. Visualization will be helpful. For a user and admin, we are giving a freedom to plan storage structure of their own need with increasingly flexible Big data model, also there is a provision to perform analytics on the information stored also. The result also shows there is a significant improvement in the Big data platform in running many tasks. The clusterization model can be varied and made more efficient it can be considered as the future work. With our present system we are considering only the format of the file for the clusterization, but to increase the efficiency we can make one more level of content-based filtering and retrieval.

References

[1] D. Agrawal, S. Das, and A. El Abbadi, "Big data and cloud computing: current state and future opportunities," 14th Int. Conf. Extending Database Technol., pp. 530–533, 2011.

[2] S. Kaisler, F. Armour, and J. A. Espinosa, "Introduction to Big Data: Challenges, Opportunities, and Realities Minitrack," Proc. 47th Hawaii Int. Conf. Syst. Sci., pp. 728–728, 2014.

[3] R. Huang and W. Xu, "Performance evaluation of enabling logistic regression for big data with R," 2015 IEEE Int. Conf. Big Data (Big Data), pp. 2517–2524, 2015.

[4] V. Čančer, "Criteria weighting by using the 5Ws & H technique," Bus. Syst. Res., vol. 3, no. 2, pp. 41–48, 2012.

[5] X. Mo and H. Wang, "Asynchronous Index Strategy for high performance real-time big data stream storage," in Proceedings - 2012 3rd IEEE International Conference on Network Infrastructure and Digital Content, IC-NIDC 2012, 2012, pp. 232–236.

[6] [6] M. Mesiti and S. Valtolina, "Towards a {User}-{Friendly} {Loading} {System} for the {Analysis} of {Big} {Data} in the {Internet} of {Things}," Comput. {Software} {Applications} {Conference} {Workshops} ({COMPSACW}), 2014 {IEEE} 38th {International}, pp. 312–317, 2014.

[7] S. Gokuldev, A. Ashokan, and R. Rajeev, "A DTQ Scheduling Algorithm with Check pointing approach in Computational Grid," vol. 11, no. 9, pp. 6850–6855, 2016.

[8] A. B. Patel, M. Birla, and U. Nair, “Addressing big data problem using Hadoop and Map Reduce,” in 3rd Nirma University International Conference on Engineering, NUiCONE 2012, 2012.

[9] J. Yang and X. Li, “MapReduce based method for big data semantic clustering,” in Proceedings - 2013 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2013, 2013, pp. 2814–2819.

[10] D. Lazer, R. Kennedy, G. King, and A. Vespignani, “The parable of {Google Flue}: traps in big data analysis,” Science (80-.)., vol. 343, pp. 1203–1205, 2014.

[11] Q. Zhang, Z. Chen, A. Lv, L. Zhao, F. Liu, and J. Zou, “A universal storage architecture for big data in cloud environment,” in Proceedings - 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, GreenCom-iThings-CPSCom 2013, 2013, pp. 476–480.

[12] S. Singh and N. Singh, "Big Data analytics," 2012 International Conference on Communication, Information & Computing Technology (ICCICT), Mumbai 2012, pp.1-4.

[13] J. Zhou, Z. Li, Z. Zhang, B. Liang and F. Chen, "Visual Analytics of Relations of Multi-Attributes in Big Infrastructure Data," 2016 Big Data Visual Analytics (BDVA), Sydney, NSW, 2016, pp. 1-2.

[14] A. Saldhi, D. Yadav, D. Saksena, A. Goel, A. Saldhi and S. Indu, "Big data analysis using Hadoop cluster," 2014 IEEE International Conference on Computational Intelligence and Computing Research, Coimbatore, 2014, pp. 1-6.

[15] A. B. Patel, M. Birla and U. Nair, "Addressing big data problem using Hadoop and Map Reduce," 2012 Nirma University International Conference on Engineering (NUiCONE), Ahmedabad, 2012, pp. 1-5.

[16] T. Kurc, U. Catalyurek, Chialin Chang, A. Sussman and J. Saltz, "Visualization of large data sets with the Active Data Repository," in IEEE Computer Graphics and Applications, vol. 21, no. 4, pp. 24-33, Jul/Aug 2001.

[17] V. Hegde, P. Karthika, and M. G. Madhu, “Opinion mining and market analysis,” Int. J. Appl. Eng. Res., vol. 10, no. 10, pp. 25629–25636, 2015.

[18] H. Abbes and F. Gargouri, “Big Data Integration: A MongoDB Database and Modular Ontologies based Approach,” in Procedia Computer Science, 2016, vol. 96, pp. 446–455.