IJCTA, 9(10), 2016, pp. 4233-4240
© International Science Press

A Study of distributed systems in real-time applications

***Pratheeba. B *Prathilothamai. M**

Abstract : In Real-Time Applications, handling data is a challenging task due to the huge volume and high velocity of data. In order to handle Big Data in real time, Distributed Frameworks like Hadoop and Apache Spark are introduced. We investigate the tradeoff between speed Vs. storage of Hadoop and Apache Spark. We have proposed a system which will help public in emergency situation during travelling on the road. As a result of this survey, we have finalized Apache Spark is suitable for our system.

Keywords : Big Data, Distributed Framework, Hadoop, Spark, Real-Time Applications.

1. INTRODUCTION

Now-a-days real-time applications are using Hadoop or Apache Spark to manage Big Data. In this paper, we have analyzed some of the various real-time applications and have compared the performances along with the pros and cons of the distributed system (Hadoop or Spark) used for it. The following applications are considered for the performance and efficiency analysis of Hadoop Vs Spark.

Considering the healthcare industry, the whole of data related to the health and well-being of the patients form “big data”[1]. “The Elderly Health Monitoring Platform Based on Spark”[2] explains how Apache Spark can serve better than Hadoop in monitoring the health of the elderly. By almost 2 to 4 times, the operational and computing time of Hadoop being greater than that of Apache Spark, will result in a delay in catering to the emergency needs of the elderly which might even turn fatal. But, the Apache Spark platform is being more expensive than Hadoop might not be affordable to many of the elderly people who in most cases might no longer be earning on a regular basis. “Large-Scale Multimodal Mining for Healthcare with MapReduce”[3] shows how the MapReduce Framework can be used to collect and analyze large volumes of patient data in order to quicken diagnoses and help in the future clinical decision-makings. This system requires large memory and thus the disk-based storage framework of Hadoop is preferred over in-memory storage of Apache Spark. This usually suffers a drawback of increased time consumption during data retrieval.

Effectualness combined together with the ease-of-use of the MapReduce process in parallel to a large number of data analysis algorithms is one reason why the Hadoop and MapReduce framework has become an integral part of the bioinformatics community[4]. The Big Data technology when applied in the field of bioinformatics and life sciences, such as the prediction of the structure and function of various proteins, has the capacity to bring into existence numerous killer applications[5]. Ensuring no single point of failure and quick processing in parallel, bioinformatics data storage has also been done using Hadoop[6].

In the field of network security, Hadoop can help overcome problems that hinder the prevention of Distributed Denial of Services[7]. Big Data also finds its place in routing protocols[8]. When compared to Relational Database Management System(RDBMS), the data management framework of Hadoop has greater data upload and query

* Department of Computer Science and Engineering, Amrita School of Engineering, Coimbatore Amrita Vishwa Vidyapeetham, Amrita University, India. bpratheeba.12@gmail.com, m_prathilothamai@cb.amrita.edu.

performance that is insensitive of data size[9]. At Facebook too, for the storing and analysis of large data sets, Hadoop has been conventionally used along with Hive[10].

In the field of marketing, Big Data comes with its own issues and challenges[11].” Efficient Processing and Utilizing Big Data in E-Commerce”[12] suggests a new architecture that uses Hadoop framework to aid E-Commerce Businesses in fixing the prices of products based on buyer statistics and helps improve customer online experience through improved recommendations. The Hadoop framework is opted since with each day the number of online users is fast increasing and volumes of data is being generated everyday due to which disk-based storage is required. To compensate for the computational overhead, a marking scheme is suggested as described in a later section(I C). Even still, if highly speeded query results are to be obtained, a switching from Hadoop to the Spark framework has to be ensured. In “Real time monitoring of system counters using MapReduce Framework for effective analysis”[13], the MapReduce framework is shown to be implemented in real-time processing with the use of Apache Spark. The MapReduce framework is mostly suited for offline batch processing. The monitoring of Information Technology(IT) systems to report any fall in the health of application servers requires quicker processing of data in order to ensure faster recovery. There is thus a need for non-batch processing. Interactive and real-time queries have to be efficiently and quickly answered. Therefore the bringing in of Apache Spark in the real-time monitoring of system counters becomes a necessity.

“A Recommend System for Modelling Large-Scale Advertising”[14]uses the data collected in the form of online likes, shares, adds to cart etc. to build an effective advertising framework. In this case, real-time iterative jobs have to be performed over a distributed dataset that has all the information collected by the online users. Spark is thus preferred over Hadoop not only to serve this purpose but also because of its feature of storing the intermediate and final results in memory instead of on hard disks. The training of the data collected is performed using the Spark Platform. Though Spark caters to the computational efficiency in terms of time, as mentioned earlier, it suffers from the drawback of being memory intensive. “Log Analysis in Cloud Computing Environment with Hadoop and Spark”[15] understands the pros and cons of both Hadoop and Apache Spark thereby performing an integration of both to ensure effectiveness in the analysis of logs collected from users of various system applications. Such integration definitely faces the issue of being cost-intensive. Also, recent studies show the feasibility of conducting analytical and programming operations at ease on configured Big Data platforms[16]. In “Platfora Method for High Data Delivery in Large Datasets”, algorithm of Platfora for processing of information using the Map Reduce framework and the Distributed Computing abilities of Hadoop have been shown to reduce the complexity[17]. In “Efficiency of Stream Processing Engines for Processing BIGDATA Streams”, Apache Spark along with Apache Flink have been said to be the most efficient engines for Stream Processing[18].

2. USAGE OF APPROPRIATE DISTRIBUTED FRAMEWORK IN REAL-TIME APPLICATIONS

In this section, we will be discussing two different types of Distributed Framework, Hadoop and Apache Spark, used in various Real-time Applications.

A. The Elderly Health Monitoring Platform Based on Spark

As discussed by Min Dong et al., health monitoring begins with the collection of information about the location *i.e.*, the latitude and longitude details, attitude, surrounding details like that of temperature, pressure and altitude, and human body status of elderly persons. This data is obtained with the help of smart wearable equipments. Through the use of smart phones, other wireless networking equipment and protocols like the Bluetooth, gathered data is sent to a remote server. Data compression takes place in order to ensure ease of caching and based on the time interval predefined, the data is sent to the server with the help of the ‘Push To Receive mode’ of Zero Message Queue(ZeroMQ), which is a light-weighted messaging protocol.

With the help of Spark technology, paralleled computing takes place and analysis is done. Furthermore, for real-time feedback, Spark Streaming is used to analyze quickly the data online in order to provide the results to hospitals or emergency care units so that the elderly are taken care of spontaneously in the case of accidents like falls. Spark Streaming provides a lot of support including a socket that runs on the Transmission Control Protocol(TCP) and other types of data input. The output can be sent to a Heterogeneous Distributed File System(HDFS) or to any other database that is supported by Hadoop. In Spark Streaming, using ZeroMQ's 'Pull Mode' of operation, the pushed data is acquired, the data that has already been processed gets saved in HBase and the analysed data in My Structured Query Language(MySQL). Online detection of attitude is performed with the help of SMV or the Signal Magnitude Vector. To get a better understanding of the status of health of the elderly, K-means algorithm is used for clustering. Spark being in-memory, is much better for K-means when compared to Hadoop. ZeroMQ's 'Publisher Mode' of operation is then used to publish the analyzed data to the necessary people or to the medical institutes. Relatives of the elderly can enable the 'Subscription mode' of ZeroMQ to get these data. SMS alerts can also be sent to the relatives of the elderly or to the doctor.

The CPU of the entire machine used in their study is Intel G2030, 4GB memory. As data is stored in memory, the iteration speed is quickened in Spark. The results of their study show that, no matter if the number of Cores is 2, 4 or 12, the time consumed by Hadoop for performing the computation is more than the time taken by Spark for its operation, Hadoop taking about 2 to 4 times of that of Spark.

B. Large-Scale Multimodal Mining for Healthcare with MapReduce

Fei Wang et al., discusses a health support system, Advanced Analytics for Information Management (AALIM), that collects medical data, like those concerning the type of illness and the medications prescribed for the particular diagnosis, from different patients. By holistically presenting this wide range of data collected, decision-making in the case of a similar health condition diagnosed at some point in the future can be made more efficient. In short, this system provides a summary of the different possible diagnoses, medications associated with them, and other demographic information.

Considering the fact that it takes approximately 90 minutes per person, the existing AALIM system can process the data of about 20 patients every 24 hours. This is good in an environment with few patients. But when the scope is increased and the data of millions of patients is to be analyzed in the case of large health institutions, the existing system will fail to serve efficiently. This is where Hadoop comes into the picture. The MapReduce Framework is used to perform efficient parallel computing and provide the analyzed results quick enough to help future patients based on the analysis of historical data collected from similar health scenarios of patients from the past.

Electrocardiogram(EKG) processing pipeline is the center of study as chosen by Fei Wang et al.,. The process begins with the conversion of data of patients that is positioned in a local directory and structured using the tree data structure, to flat files that are made up of binary key-value pairs. These files are called 'sequence files'. The intermediate results of maps are also saved with the help of these files. The entire path of the file present in the data directory is made the key. The value has the content of the file. As every stage in the pipeline is implemented, a definition for a per-EKG or per-record execution is stated. The obtainment of features happens in the Map operation and the Reduce operation puts all the obtained outputs together. Since the pipeline is serially dependent, for the Map operation, the complete path of every EKG is assigned to the key thereby ensuring that there is no dependency not only within but also among various patients.

The use of files along with the various concepts of networking to exchange data by Hadoop creates some added costs. Thus, many stages in the pipeline are usually clubbed together making one Map operation due to which the voluminous data sent over the network is made small. The Map task once again creates the folders containing the EKG, with the same pattern as the serial implementation. It then constructs a picture of the EKG waveform. Using this, the channels of the EKG, the heart rates and the related reports are obtained. After this, with the help of these heart rates is constructed a consensus rate, through which extraction of structural characteristics from the EKG waveform and identification of diseases is performed. The patient correspondence ûles are used to perform a comparative study with other rate estimates.

Once processing gets completed, the results are aggregated into another per-patient file. The requirement of combining the freshly arriving outputs with the already present records of the database is efficiently dealt with by the Map task through the help of a hash function to find out the entries that require alteration.

The research of Fei Wang et al., was experimented on a group containing 17450 12-channel EKG waveforms got from 1377 cardiac patients belonging to a huge network of a hospital, every EKG channel having 2.5 seconds of recordings of signal (rate of sampling being 300 per seconds). MapReduce was run on a 9 servers-containing cluster. Every server having four 64-bit 2.2 GHz Advanced Micro Devices(AMD) cores. A 4.79 times enhancement of speed was obtained using six nodes and a 7.22 times enhancement of speed when three more nodes were added.

Thus the MapReduce framework of Hadoop can help improve the health care facilities available to people based on efficient analysis of the voluminous data collected from people from the past with similar health conditions. Further, this kind of analysis can help find clinically more relevant and less costly means to diagnose and treat the medically-ill, a leaner, quicker, more targeted Research and Development pipeline of medications and medical equipments, study of patterns in diseases and tracking of disease outbreaks and transmission to provide for better public health surveillance and hasten response, quicker development of more accurately targeted vaccines and many more. Apache Spark is not preferred in this application because large volumes of data when handled through it results in the consumption of very large amounts of memory due the fact that Spark framework is in-memory.

C. Efficient Processing and Utilizing Big Data in E-Commerce

In this study, a new platform architecture for the processing of Big Data in the field of E-Commerce is proposed by Xiaohui Pan. As per the usual Hadoop framework, the input data is broken down into blocks. Marking of these blocks is then performed with the help of time labels and application relationship labels which are very essential in any analysis related to E-commerce.

If Dataset A and Dataset B are connected in a query, it would imply that some data blocks in A and some of them in B are related. To mark such relationships, the application relationship labels are used. The data blocks that are closely associated will then be stored in a particular data node.

Once the data blocks are processed, the routing table for the Big Data is obtained containing the information relating to the positioning of the data blocks in the data nodes. This architecture, eliminates the need for the shuffle stage in Hadoop. Also, the computational efficiency is improved, mainly that of the computation of relation query.

This architecture of Hadoop can find its applications in adjusting prices of products by companies based on user demand, helping serve customer expectation through recommendations and advertising, analysis of business statistics, and management of supply chains.

In the architecture proposed, storage is done with the help of disk arrays. On top of the storage layer, resides the platform infrastructure layer that has a high-performance cluster that runs a Hadoop framework. On top of this is the Big Data Layer that is used to denote that the data sources could be many and varied. On top of this resides the Processing and Analyzing Modules Layer where the MapReduce programs actually run. Then comes the Business Views and Models Layer where based on the Big Data application, additional processing through MapReduce or custom a customized code in Java might be incorporated to construct an intermediary data structure like that of a statistical model, a flat file, or a relational table that can be used for extra analysis or to be queried by a traditional tool that is SQL-based. Finally comes the Application and Visualization Layer that is used to run the Visualization Applications to view the resultant data in an efficient way.

D. Real Time Monitoring of System Counters Using MapReduce Framework for Effective Analysis

Sandeep B. Aher and Poonam D. Lambhate propose a system that offers efficient monitoring to different users in an IT system like the Monitoring team and the Performance Testing team. To ensure real-time monitoring of systems to report any problems immediately to the appropriate team, Spark’s in-memory techniques are used instead of Hadoop’s disk-based MapReduce model that occurs in two steps.

The applications using Spark run on a cluster as sets of processes that are independent. They are coordinated by the Spark Context Object. In order to function on the cluster, this object links to different flavors of cluster managers who distribute resources across different applications. When the connection is achieved, acquisition of executors on nodes in the cluster takes place. These are the processes that run the computations and ensure the storing of data for the application. Then the application code is sent to the executors in the form of Python files or JAR files. The Spark Context then sends tasks that the executors are required to perform.

Thus the process begins through the collection of data by the data collection agent on each and every server. Then the data is processed in adherence to the criteria of input got from the application service manager. After this, the data is grouped based on the process ID and the received data is put into the format of an array. The output can be visualized using graph tools.

In the research performed by Sandeep B. Aher and Poonam D. Lambhate, the experimental setup had Windows 7 on desktop machines. The frequency of data collection was initially at regular intervals of a second and went on up to real-time data collection.

This system achieved real-time data monitoring, specifically, the process-level free memory (MB) monitoring using the Apache Spark Platform. This can be used to trigger reporting of system errors immediately after its occurrence to ensure necessary actions be taken in order to facilitate quick recovery and reduced loss for the IT Systems.

The MapReduce Framework in this setup is only to ensure scalability and fault-tolerance and not for the actual ‘Map’ and ‘Reduce’ functionalities.

E. A Recommend System for Modelling Large-Scale Advertising

Internet has lead to the production of a vast amount of digital information. E-Commerce only adds to this huge load of data. Advertising over the Internet plays a major role in the profits made by the virtual business organizations. This paper proposes a System for the modeling of large-scale advertising using the Spark framework.

Digital information that can help in advertising is collected with the help of click logs, product comments, social media posts, online shopping records and many more. Based on this information, different products or services can be recommended to different sets of users based on the similarities in their needs or their online behaviors.

The system recommended by Jing Ma and Yueming Lu, uses the historical data of likes, buys, adds to cart etc., to decide the user’s attributes such as background, preferences and motives. These features are then subdivided into blocks. A distributed Collaborative Filtering(CF) Algorithm is implemented to handle these sub-blocks and process it in parallel. This algorithm can learn the optimization parameters by the training the system with the help of history data. Weights are provided to different actions like liking, clicking, adding to favorites, adding to the cart and buying. Based on these history data, predictions are made for the current users and different products or services are recommended to them.

In this system, efficient training of models and storage of valuable results plays a very important role. This is where distributed computing and storage comes into play. Spark is used for this as it comes with an easy to operate package containing a variety of recommendation algorithms including the CF Algorithm. For the performing of an iterative job over a distributed dataset, Spark is preferred over Hadoop. Spark is also preferred to Hadoop because of its feature of storing the intermediate and final results in memory instead of on hard disks. Thus, the training of the data collected is performed using the Spark Platform.

This training is also done on a standalone platform for the sake of comparison. With one lakh of training samples, standalone takes 6s whereas the Spark platform takes less than a second to give out the results.

F. Log Analysis in Cloud Computing Environment with Hadoop and Spark

To analyze a log means to collect information related to the amount of users accessing a system, the contents that are being accessed by various users and other behaviors of the users, the status of the system being accessed and lots more, from the vast data available. Various standalone platforms are available for the analysis of logs. But these platforms often suffer from issues related to scalability. To handle huge volumes of data, the distributed cloud computing framework of Hadoop, which is open source, provides a scalable cloud computing platform for log analysis. For data analysis, a tool named Hive is used and for data storage, a distributed file system, HDFS, is used. Hadoop though suitable for batch processing, suffers from some drawbacks like its disk-based storage mechanism which makes it unsuitable for interactive querying and iterative applications where quick responses are required.

In the paper by Xiuqin Lin et al.,, a method to integrate Hadoop and Spark is proposed. This is to make sure that Batch processing and scalability is possible in terms of large disk storage, at the same time ensuring the feasibility of interactive ad-hoc querying of data from logs and running iterative algorithms. Spark is an in-memory cloud computing system, Shark, compatible with Hive, is the system constructed on it for analyzing data. Sophisticated functions that are used in analysis as well as SQL queries can be run with Shark.

The integration of Hadoop and Spark results in using HDFS as the platform for stable storing of data and MapReduce operation to ensure stability in terms of batch processing. Spark is used for its in-memory calculations. For the implementation of classification or clustering algorithms like K-means, a data mining module (DM) is incorporated.

The proposed method works as follows. Logs are collected and sent to an Extract-Transform-Load(ETL) module. This ETL module pre-processes the incoming data and loads it into the data warehousing. This pre-processing includes deletion of redundant data and conversion of data into suitable format. This is followed by a log analysis module that analyses the incoming logs according to the statistical demands of the users. The resultant data is loaded into both the rational databases - Hive and Shark. Hive is used for batch operations like data statistics, while Shark is used to respond to interactive queries. There also exists a Query and Display System that allows users to input the queries and get the results displayed.

Hadoop uses ‘heartbeat’ to communicate the decisions regarding to scheduling which creates a 5-10s delay. Spark being event-driven consumes only 5ms for the same.

Balancing the pros and cons of both Hadoop and Shark, an efficient Log Analysis System has been developed. The experiment was tested in a cluster of 6 nodes for K-means and PageRank data re-use interactive algorithm using 2 CPU core along with 4G memory for each of the nodes. It was found that Spark’s performance was 40 times more than that of Hadoop. When the quantity of data was increased, the performance of Spark came down though it was still about 8 times more than that of Hadoop.

This paper also states that Shark is more suitable for queries that result in fewer results and Hive is preferred for its stability when the size of the results does not vary much from the size of the input.

3. USAGE OF DISTRIBUTED FRAMEWORK (HADOOP VS APACHE SPARK) IN VARIOUS REAL-TIME APPLICATIONS

Table 1 : Comparison of Hadoop Vs Apache Spark

S.No	*Title of Paper*	*Application*	*Distributed Computing Framework Used*	*Comments*
1.	The Elderly Health Monitoring Platform Based On Spark	Health Monitoring of Elderly	Apache Spark	1. efficient in real-time emergency handling. 2. memory-intensive.
2	Large-Scale Multimodal Mining for Healthcare with MapReduce	Clinical decision-making	Hadoop	1. Voluminous data storage. 2.Increased time consumption due to disk-retrieval.
3	Efficient Processing and Utilizing Big Data in E Commerce	E-Commerce Data Analytics	Hadoop	1.Large amount of data generated. 2.does not require quick processing.
4	Real time monitoring of system counters using MapReduce Framework for effective analysis	System Monitoring	Apache Spark and Hadoop	1. Interactive Querying required. 2. Quick response essential. 3. Scalable and fault-tolerant because of MapReduce.
5	A Recommend System For Modelling Large-Scale Advertising	Commercial Advertising	Apache Spark	1. iterative jobs performed overdistributed dataset. 2. intermediate and final results to be stored in memory. 3. memory-intensive.

S.No	*Title of Paper*	*Application*	*Distributed Computing Framework Used*	*Comments*
6	Log Analysis In Cloud Computing Environment With Hadoop And Spark	Analysis of Logs	Hadoop and Apache Spark	1. HDFS - to ensure stability in storing of data and MapReduce to ensure stability in batch processing. 2. Spark for calculations that are performed in memory. 3. Shark data analysis for queries that result in fewer results and Hive for its stability when size of the results does not vary much from the size of the input.

4. CONCLUSION

We have analyzed various real-time applications which use distributed frameworks Hadoop or Apache Spark as listed in Table 1. Hadoop being less memory-intensive is built around scalability resulting in the drawback of reduced speed. To handle static data and applications like that of E-Commerce, which analyze large volumes of data to find best ways to attract and help customers, Hadoop is preferred. Hadoop is also recommended in various research systems as the computational speed is not critical in these scenarios. The processing framework of Apache Spark is constructed keeping in mind the speed, user-friendliness and the sophistication of analysis of Big Data. In medical applications where the speed at which the query results are to be obtained is critical as in the case of emergency health condition handling, Apache Spark is preferred. People authentication through Apache Spark can help use voluminous data present in documents like Aadhar cards very effectively and with less time consumption. This can even serve to control the entry of terrorists into native territories. Disaster Prediction through Apache Spark can ensure that warnings reach the public at an early period. Thus, different applications can be developed with increased efficiency by making the right choice of distributed frameworks. From this survey, we have found that Apache Spark is the right one for our road traffic prediction system.

5. FUTURE WORK

In “Timely Prediction of Road Traffic Congestion Using Ontology”[19], Prathilothamai et al., have suggested a method for effective traffic prediction on the roads by the parallel processing of data obtained from ultrasonic, passive infrared sensors and video cameras. The PIR sensor detects motion and the ultrasonic sensor is used to obtain the distance between the vehicles. Thus, the count of vehicles in the inbound and outbound direction along with their speed is got. Through background subtraction and the use of the Gaussian Mixture Model, the count of vehicles on the road and the distance between them is obtained from the data collected by the video camera. Euclidean distance has been used to find the minimum distance between vehicles.

Through Parallel Computing of the obtained information, and with the help of Ontology, efficiency in the prediction of traffic congestion has been achieved. The Map Reduce framework of Hadoop has been used for parallel computing in the above application.

From our study, it has been observed that the use of Apache Spark could ensure quicker results than that of Hadoop. Therefore, if Apache Spark be used in place of Hadoop, Road Traffic Congestion could be predicted much faster than that stated in the above research. The quicker the results of prediction be given out, the faster the public be informed about the possible traffic congestion. This could help guide the public accordingly which helps save not just a lot of time but also a large amount of fuel and also it help the emergency vehicles can reach their destination in time. Thus, the implementation of “Timely Prediction of Road Traffic Congestion Using Ontology” using Apache Spark could help us move a step away from traffic and a step towards a less pollutant environment.

6. REFERENCES

1. Raghupathi V, Raghupathi W, “Big data analytics in healthcare : promise and potential”, Raghupathi and Raghupathi Health Information Science and Systems 2014, Vol. 2(3), pp. 1-10.
2. Liu H, Dong M, Pang N, Bi S, Zeng X, Huang X, Tang X, “The Elderly Health Monitoring Platform Based On Spark”, The 5th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, Shenyang, China, 2015 June 8-12, pp. 514-19.
3. Holder A, Beymer D, Shekita E J, Wang F,Xu L H, Mahmood T S, Ercegovac V, “Large-Scale Multimodal Mining for Healthcare with MapReduce”, ACM, 2010 Nov 11–12, pp. 479-83.
4. Taylor R C, “An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics”, Proceedings of the 11th Annual Bioinformatics Open Source Conference (BOSC), 2010.
5. O’Driscoll A, Daugelaite J, Sleator R D, “‘Big data’, Hadoop and cloud computing in genomics”, Journal of Biomedical Informatics, 2013 Oct,Vol. 46(5), pp. 774-81.
6. Jeong Y S,”Parallel Processing Scheme for Minimizing Computational and Communication Cost of Bioinformatics Data”,Indian Journal of Science and Technology,2015 July, Vol. 8(15), pp. 1-8.
7. Munetomo M, Mizukoshi M, “Distributed Denial of Services Attack Protection System with Genetic Algorithms on Hadoop Cluster Computing Framework”, IEEE, 2015 May 25-28, pp. 1575-80.
8. Park H W, Yeo I Y, Jang H, Kim N,”Study on the Impact of Big Data Traffic Caused by the Unstable Routing Protocol”,Indian Journal of Science and Technology, 2015 Mar, Vol. 8(S5), pp. 59-62.
9. Wang D, Yu H, “Research and Implementation of Massive Health Care Data Management and Analysis Based on Hadoop”, IEEE Fourth International Conference on Computational and Information Sciences, 2012 Aug 17-19, pp. 514-17.
10. Aiyer A, Menon A, Borthakur D, Molkov D, Kuang H, Gray J, Sarma J S, Muthukkaruppan K,Ranganathan K, Spiegelberg N, Schmidt R, Rash S, “Apache Hadoop Goes Realtime at Facebook “, Proceedings of the 2011 ACM SIGMOD International Conference on Management of data,2011, pp. 1071-80.
11. Mamlouk L, Segard O,”Big Data and Intrusiveness: Marketing Issues”,Indian Journal of Science and Technology,2015 Feb, Vol. 8(S4), 189-93.
12. Pan X, “Efficient processing and utilizing Big Data in E-commerce”, International Conference on Intelligent Systems Research and Mechatronics Engineering (ISRME), 2015, pp. 315-21.
13. Lambhate P D**,** Aher S B, “ Real time monitoring of system counters using MapReduce Framework for effective analysis “, International Journal of Current Engineering and Technology, 2015 Aug, Vol. 5(4), pp. 1-5.
14. Ma J, Lu Y, “A Recommend System For Modelling Large-Scale Advertising”, International Conference on Cyberspace Technology (CCT), IEEE, 2014 Nov 8-10, pp. 1-4.
15. Wu B, Wang P, Lin X, “Log Analysis In Cloud Computing Environment With Hadoop And Spark”, Proceedings of IEEE IC-BNMT, 2013 Nov 17-19, pp. 273-76.
16. Noh K S, Lee D S, “ Bigdata Platform Design and Implementation Model”, Indian Journal of Science and Technology, 2015 Aug, Vol.8(18), pp. 1-8.
17. V. S. Thiyagarajan, “Platfora Method for High Data Delivery in Large Datasets”, Indian Journal of Science and Technology, 2015 Dec, Vol. 8(33), pp. 1-13.
18. B.V.S Srikanth and V. Krishna Reddy, “Efficiency of Stream Processing Engines for Processing BIGDATA Streams”, Indian Journal of Science and Technology, 2016 April, Vol. 9(14), pp. 1-10.
19. Ms. M.Prathilothamai, Ms Marilakshmi S, Ms Nilu Majeed, Mr.V.Viswanathan, “Timely Prediction of Road Traffic Congestion Using Ontology “, Proceedings of the International Conference on Soft Computing Systems, Vol. 398, Advances in Intelligent Systems and Computing, pp. 331-44.