Big data: Properties, Challenges, Data Processing Techniques using Hadoop Map Reduce and Distributed File Systems

Mayuri K P
Department of Computer Science & Engineering
Sir M Visvesvaraya Institute of Technology, Bangalore, India

C Padmini
Department of Information Science & Engineering
Sir M Visvesvaraya Institute of Technology, Bangalore, India

Abstract

Big data has drawn huge attention from researchers in information sciences, Healthcare, Business, Policy and decision makers in government and enterprises. In addition social, scientific and engineering applications have created large amounts of both structured and unstructured information which needs to be processed, analyzed and linked. This paper is aimed to demonstrate a close- up view about big data characteristics, challenges and issues in adopting and accepting big data technology and its processing techniques. Several solutions to the big data problem have emerged which introduce the MapReduce environment and it is available as open-source in Hadoop. Hadoop's distributed processing, MapReduce algorithms and overall architecture are a major step towards achieving the promised benefits of big data.

Keywords: Big data, Hadoop, HDFS, MapReduce

I. PROPERTIES/ CHARACTERISTICS

1) Variety:
Handles both the structure and unstructured data from the resources like WebPages, Emails, sensor devices, social media sites. Collected from either active or passive devices, these data is difficult to handle by the existing traditional analytic systems.
2) Volume:
The data capacity ranges from tera bytes to peta bytes at present and is supposed to increase to zeta bytes nearby in future. This enormous amount of data is produced every day through social networking sites and is highly unmanageable by the existing traditional systems.
3) Velocity:
Velocity in big data is not only limited with the speed of the incoming data, but also about the speed at which data flows. Traditional systems cannot efficiently perform data analytics on the high speed data movement.
4) Variability:
Variability generally considers about the inconsistencies of the data flow, this leads to a challenge in maintaining heavy data loads that is produced from social media, when certain events occurred.
5) Validity:
Validity means using the correct and accurate data for the intended use. Valid data is the key for making the right decisions.
6) Veracity:
Veracity refers to biases, noise and abnormality in data. It is the data that is being stored and mined, meaningful to the problem that is being analyzed.
7) Volatility:
Volatility refers how long the data is valid and how long should it be stored. Deletion of data is required that the data is no longer relevant to the current analysis.
8) Complexity:
Includes the issues regarding the data cleansing, linking and transformation of data across systems coming from various sources. Data can quickly spiral out of control because of multiple data linkages and hierarchies existing among data.

II. LITERATURE REVIEW

The authors [1] focused on the big data technology along with its importance in the modern world and existing projects which are effective and important in changing the concept of science into big science and society too. They also discussed about the good big data practices to be followed. The authors [2] pointed out a review about the big data mining and the issues and challenges with emphasis on the distinguished features of big data. They also discussed some methods to deal with big data. The authors [3] focused on HDFS and MapReduce architecture and processing techniques in a distributed environment in listing the benefits of the big data. The authors [4] pointed out the details on difficulties in data capture, data storage, data analysis and

visualization. This paper aimed about to demonstrate a close-up view about big data, including big data applications, big data opportunities and challenges. The authors also discussed about several underlying methodologies to handle the data deluge, for example granular computing, cloud computing, bio-inspire computing and quantum computing. The authors [5] discussed about the general background of big data and then focused on Hadoop platform using MapReduce algorithm which provide the environment to implement applications in distributed environment and it can capable of handling node failure. The authors [6] were aimed at studying big data security at the environmental level, along with the probing of built-in protections and the Achilles heel of these systems, and also embarking on a journey to assess a few issues that we are dealing with today in procuring contemporary big data and proceeds to propose security solutions and commercially accessible techniques to address the same. They also addressed about the security issues of the big data. The authors [7] talks about the benefits like stability & robustness, a rich set of features and compatibility with traditional analytics applications. During their study of the performance exhibited by a commercial cluster file system in Map/Reduce workloads and its comparison with a distributed file system, they observe, that a significant amount of time is spent during the copy phase of the Map/Reduce model after map task finishes. In Hadoop platform, the input and output data of Map/Reduce jobs are stored in HDFS, with intermediate data generated by Map tasks are stored in the local file system of the Mapper nodes and are copied (shuffled) via HTTP to Reducer nodes. The time taken to copy this intermediate map outputs increase proportionately to the size of the data. However, since in case of a clustered file system, all the nodes see all the data, this copy phase can be avoided by keeping the intermediate data in the clustered file system as well and directly reading it from there by the reducer nodes. This endeavor will completely eliminate the copy phase after map is over and bound to give a significant boost to overall performance of Map/Reduce jobs. The author [8] discussed about a variety of system architectures have been implemented for data-intensive computing and large-scale data analysis applications including parallel and distributed relational database management systems which have been available to run on shared nothing clusters of processing nodes for more than two decades. However most data growth is with data in unstructured form and new processing paradigms with more flexible data models were needed. The reference [9] provides a link: https://wiki.apache.org/ which comprises the details regarding how actually the Hadoop Map and Reduce phases are combined and the other information about the developer code details etc. The authors [10] reports the experimental work on big data problem and its optimal solution using Hadoop cluster, Hadoop Distributed File System (HDFS) for storage and using parallel processing to process large data sets using Map Reduce programming framework. They have done prototype implementation of Hadoop cluster, HDFS storage and Map Reduce framework for processing large data sets by considering prototype of big data application scenarios. The results obtained from various experiments indicate favourable results to address big data problem.

III. BIG DATA CHALLENGES AND ISSUES

Each phase of big data analysis such as data acquisition and recording, information extraction and cleansing, data integration, aggregation and query processing introduces challenges [2]. For the effective decision making and to uncover the hidden patterns and correlations big data challenges include technical, analytical, privacy and security challenges, data storage and processing etc.

A. Technical challenges:

Technical challenges include fault tolerance, heterogeneous and incompleteness, scalability and complexity, quality of data.

1) Fault tolerance:

with the new technologies like cloud computing and Big data, the damage should be in some acceptable threshold level whenever the failure occurs, rather than restarting the whole task from the scratch. One way to increase the fault tolerance in big data is dividing the whole computation into individual tasks and assigning these tasks into different nodes for computation. If something happens, that particular task can be restarted. This method cannot be applicable for tasks which might be recursive in nature, because the output of previous work will be applied to the next work as input. In such cases dividing the whole process into individual tasks become cumbersome process, and can be avoided by applying checkpoints, which keeps track of system state at certain interval of time. If any failure occurs the computation can be restarted from the last check point maintained.

2) Heterogeneous and Incompleteness:

Data can be both structured and unstructured, 80% of the data generated by organizations are unstructured. These data are highly dynamic and does not have particular format. They can be in the form of images, PDF document, E-mail attachments, medical records, graphics, voice mails etc. in the case of complicated heterogeneous mixed data, the data has several rules, patterns and the pattern properties vary greatly. Transforming this data to structured format for later analysis is a major challenge, so new technologies required for dealing with such type of data. Structured data is always organized into highly mechanized and manageable way. It shows well integration with data base, but unstructured data is completely raw and unorganized. Converting all this unstructured into structured data one is also not feasible digging through unstructured data is cumbersome and costly [1]. Incomplete data creates uncertainties during data analysis and it should be managed. Incomplete data refers to the missing of data field values for some samples. The missing values can be caused by different realities such as malfunction of a sensor node or some systematic policies to intentionally skip some values [2]. Data imputation is an established research field which seeks to impute missing values in order to produce improved models.

3) Scale and complexity:

Managing the rapidly increasing volume of data is a challenging issue. Due to the scalability and complexity of data that need to be analyzed. The data organization, retrieval and modeling were also become challenges, and traditional software tools are not enough for managing large volumes of data [2]. The data processing solutions used in earlier were not supporting parallelism across nodes within a cluster. But now the concern has shifted to parallelism within a single node. In past techniques of parallel data processing across nodes are not capable of handling intra node parallelism. The scalability issue of big data has led to aggregating multiple disparate workloads with varying performance goals into very large clusters. This requires a high level of resource sharing, also requires dealing with the system failures in an efficient manner, which occurs more frequently if operating on large clusters [1].

4) Quality of Data:

For the better results and conclusions to be drawn, big data mainly focuses on quality data storage rather than having very large irrelevant data [1]. Data can be processed to improve the data quality with the techniques including data cleaning, data integration, transformation and data reduction [4]. These techniques can also be applied to remove noise and correct inconsistencies.

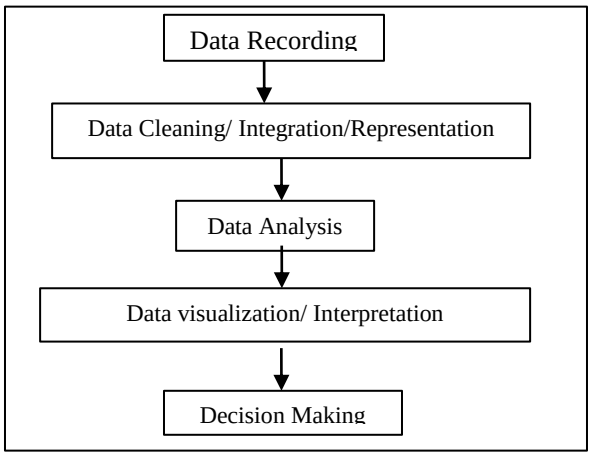

Fig.1: Processing Techniques for Data Quality.

The data is captured through ubiquitous information sensing mobile devices, remote sensing, software logs, cameras, microphones, RFID readers, WSN's and so on. The world's technological capacity to store information has roughly doubled about every 3 years since 1980's. These valuable data are captured at high cost. The preprocessing of data is necessary before it is stored, such as data cleansing, transforming and cataloguing. After these processing, the data is available for higher level online data mining functions. Scalability is the major issue to be addressed by these online data mining functions, for the purpose to analyze the big data the analytical methods: sampling, online and multi- resolution techniques are required. After data analysis the complex data sets need to be visualized more effectively by using different graphs. After all, a decision has to be taken here regarding whether big data will lead to an end of theory or whether it can help us to make better decisions on large number of big data techniques and technologies have been developed or under developing.

B. Storage and Processing challenges:

Social media sites along with the sensor devices are themselves a great contributor for producing the large amount of data. Uploading this huge amount of data to the cloud does not solve the problem. More over this data is changing so rapidly which will make hard to upload in real-time. At the same time the cloud's distributed nature is also problematic for big data analysis. Processing of large amount of data also takes large amount of time. To find suitable elements whole dataset need to be scanned, this is somewhat not possible. Thus building up indexes right in the beginning while collecting and storing the data is a good practice, and reduces the processing time considerably. The problem is that each index structure is designed to support only some classes of criteria.

C. Security and Privacy Challenges:

Trustworthiness of each data source need to be verified and techniques should be explored for identifying the maliciously inserted data. Any security measure used for big data should meet the basic requirements:

- It should not compromise essential big data characteristics.
- It should not compromise the basic functionality of the cluster.
- It must address a security threat to big data environments or big data stored within the cluster.

Releasing or modification of unauthorized data and denial of resources are the categories of security violations. Techniques like authentication, authorization, encryption and audit- trails can be used for the security of the big data [2]. Some of them

include authentication methods, file encryption, implementing access control, key management, and logging and secure communication.

IV. BIG DATA PROCESSING TECHNIQUES

A. *HADOOP*

MapReduce and Hadoop are the most widely used models today for big data processing. Hadoop is an open source large scale data processing framework that supports distributed processing of large chunks of using simple programming models. The Hadoop framework application works in an environment that provides distributed storage and computation across clusters of computers. Hadoop is designed to scale up from single server to thousands of machines each offering local computation storage. Hadoop has two major layers namely:

1) Processing/ computation Layer (MapReduce)
2) Storage Layer (Hadoop DFS)

Hadoop consists of distributed file system, data storage and analytics platforms and a layer that handles parallel computation, workflow and configuration administration [6].

B. *Hadoop MapReduce*

MapReduce is a parallel programming model for distributed application devised at GOOGLE for efficient processing of multi-terabyte data-sets, on large clusters of commodity hardware in a reliable, fault-tolerant manner. The MapReduce algorithm contains two important tasks, namely Map and Reduce.

The Map or Mapper's job is to process the input data; The Map takes a set of data and converts it into another set of data, where individual elements are broken down into Tuples (Key/value pairs). Generally the input data is in the form of file or Directory and is stored in the Hadoop File System. The input file is passed line by line to the mapper function. The mapper processes these data and creates several small chunks of data.

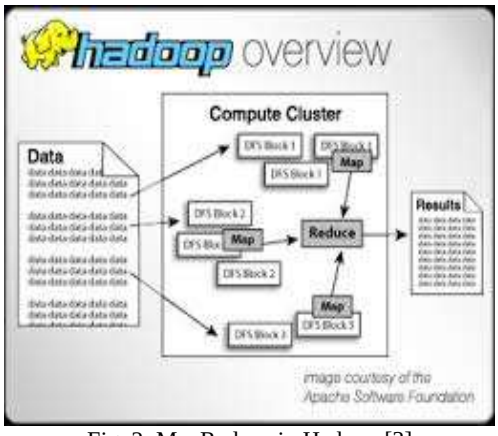

Fig. 2: MapReduce in Hadoop [3]

After data is loaded into clusters in Hadoop, it is distributed to all the nodes. The HDFS then splits the data into sets which allow management by individual nodes within the cluster. Hadoop follows the policy of "Moving Computation to the Data". In according to the application logic, the data is broken into formats. Hadoop programming framework is record- oriented. A node in the cluster processes a subset of records by a process, which is then scheduled using the location information in the file system. The computation is moved to the closest location of the availability of the data. Unnecessary data transfers are avoided since much of the information is read from the locally available disk system. Due to this performance is greatly enhanced because of the high data locality.

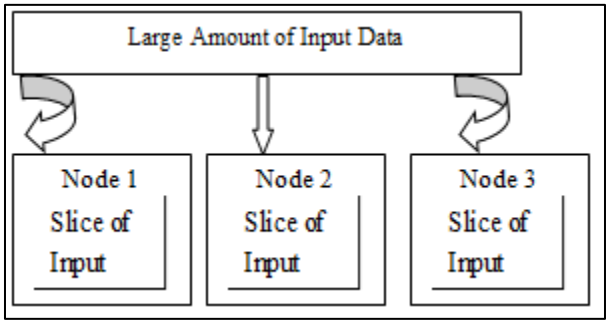

Fig.3. Moving Computation to Data [8]

There are a Master (Job Tracker) and a number of slaves (Task Tracker) in the MapReduce framework. The Master node is in charge of job scheduling and task distribution for the slaves. The slaves implement the tasks exactly as assigned by the master. As long as the systems start to run, the master node keeps monitoring all the data nodes. If there is a data node failed to execute the related task, the master node will ask the data node or another node to re-execute the failed tasks.

Secondly Reduce task takes the output from a Map as an input and combines those data tuples into a smaller set of tuples. As the name implies the Reduce task is always performed after the Map job. This stage is the combination of the shuffle stage and the Reduce stage. The reducer's job is to process the data that comes from the mapper. After processing it produces a new set of output, which will be stored in the HDFS.

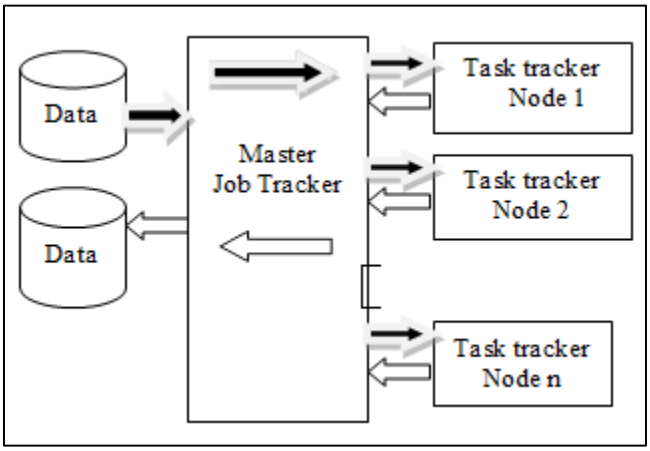

Fig. 4: Map/Reduce overview: Solid arrows are for Map flows and feint arrows are for reduce flows. [4]

In practice, applications specify that input files and output locations and submit their Map and Reduce functions via interactions of client interfaces. These parameters are important to construct a job configuration. After that the Hadoop job client submits the job and configuration to the job tracker. Once job tracker receives all the information, it will distribute the software/configuration to the task trackers, schedule tasks and monitor them, provide status and diagnostic information to the job client. From the fore going we know that coordination plays a very important role in Hadoop, it ensures the performance of a Hadoop job.

During a MapReduce job, Hadoop sends the Map and Reduce tasks [7] to the appropriate servers in the cluster. The framework manages the activities like task issuing, verifying the task completion and copying data around the cluster between the nodes. Most of the computing takes place on nodes with data on local disks that reduces the network traffic. After completion of the given tasks, the cluster collects and reduces the data to form an appropriate result, and sends it back to the Hadoop server. The MapReduce framework handles task scheduling, monitoring and failures. The main leverage of MapReduce is the tasks of similar nature are grouped together, so that the same type of data is placed on the same nodes. Doing this saves the synchronizing overhead which might have been caused if tasks were grouped in a random order [3].

The major advantage of MapReduce is that it is easy to scale data processing over multiple computing nodes. Under the MapReduce model, the data processing primitives are called mappers and reducers. Decomposing a data processing application into mappers and reducers is sometimes non-trivial. But once we write an application in MapReduce form, scaling the application to run over hundreds or even thousands of machines in a cluster is merely a configuration challenge [10]. This simple scalability is what has attracted many programmers to use the MapReduce model.

V. HADOOP DISTRIBUTED FILE SYSTEM

HDFS is based on Google File System (GFS). It's highly fault tolerant and is designed to be deployed on low- cost hardware. It provides high throughput access to application data and is suitable for applications having large data sets. It provides easier access though it stores large amount of data. HDFS follows the master- slave architecture [5]. Which includes the elements called Name node, Data node and Blocks. These elements contain built-in servers in them, which help the users to easily check the status of the cluster. HDFS provides file permissions and authentication.

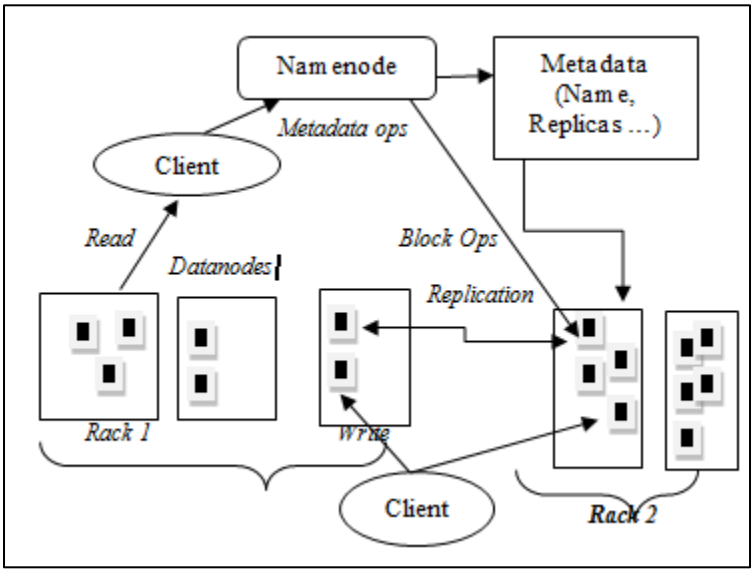

Fig. 5: HDFS Architecture. [7]

A. Namenode:

It is the commodity hardware that contains the GNU/Linux operating system and the Namenode software. It is software that can be run on commodity hardware. The system having the Namenode acts as a Master server and it does the tasks including: manages the file system namespace. Regulates clients access to files and it also executes file system operations such as renaming, closing and opening files and directories.

B. Blocks and Datanode:

The file system of HDFS is block-structured, in which files are broken down into small units of size that is specified. These units or blocks can be stored through a loop or clusters of multiple data storage computing capability. The computing systems in each cluster are called Datanodes.

The default block size is 64MB, but it can be increased as per the need to change in HDFS configuration. A file can consist multiple blocks and it is not necessary that they are stored on same machine, as the decision where each block will be stored is randomly selected [3]. If multiple machines are needed in serving a file, then a file could become unavailable even if a single machine in the cluster is lost. HDFS handles this issue by replicating each block across multiple systems which is set to 3 as default.

The file system Namespace: HDFS supports an empherical file structure. Files are stored in the directories created either by user or an application. The Namenode handles the filesystem namespace. It records alternations and its associated properties. A number can also be specified for replicas of a file by the application which must be maintained by the HDFS which is defined as the replication factor and the information stored in the Namenode.

C. Data Replication:

HDFS is programmed to manage last file stored in large cultures of data mines/structures while ensuring reliability. This is managed by storing files in a sequence of blocks which are the same size, with the last block being an exception. These blocks are then replicated to test fault tolerance in which size of the block and the replications are configurable. An application can then custom specify the number of copies of a file.

HDFS runs across the nodes in a Hadoop cluster and together connects the file systems on many input and output data nodes to make them into one big filesystem. The present Hadoop system consists of Hadoop kernel, MapReduce, the Hadoop Distributed File System and a number of related components such as Apache Hive, HBase, Oozie, Pig and Zookeeper [9]. These tools are used for handling the velocity and heterogeneity of data.

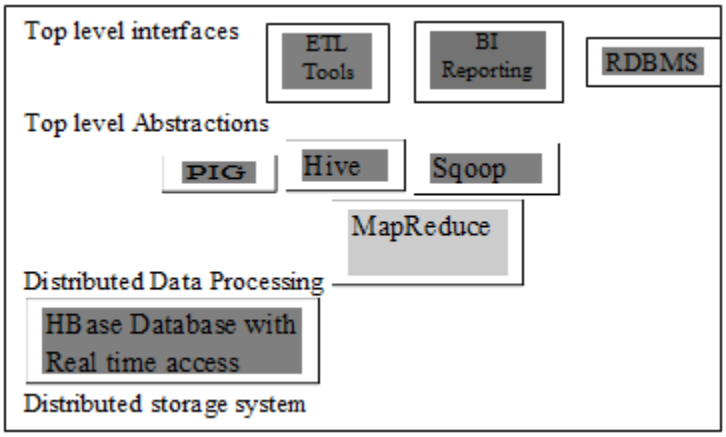

Fig. 6: Hadoop Architecture Tools.[6]

- HDFS: A highly fault-tolerant distributed file system that is responsible for storing data on clusters.
- MapReduce: A Powerful parallel programming technique for distributed processing of vast amount of data on clusters.
- HBase: A column oriented distributed NoSQL database for random read/write access
- Pig: A high level data Programming language for analyzing data of Hadoop computation.
- Hive: A Data warehousing application that provides a SQL like access and relational model.
- Sqoop: A Project for transferring/importing data between relational databases and Hadoop.
- Oozie: An orchestration and workflow management for dependent Hadoop jobs.

VI. CONCLUSION AND FUTURE WORK:

The amount of data is growing exponentially worldwide due to the explosion of social networking sites, search and retrieval engines, and media sharing sites, stock trading sites, news sources and so on. Big data analysis is becoming indispensable for automatic discovering of intelligence that is involved in the frequently occurring patterns and hidden rules. In this paper we discussed about properties/ characteristics, issues and challenges of big data and big data processing techniques using Hadoop. Big data analysis tools like MapReduce over Hadoop and HDFS which helps organizations to better understand their customers and market place and to take better decisions and also helps researchers and scientists to extract useful knowledge out of big data. We also discussed some Hadoop components which are to support the processing of large data sets in distributed computing environments. In future we can use some clustering techniques and check the performance by implementing it in Hadoop.

REFERENCES

[1] Avita Katal, Mohammed Wazid, R H Goudar "Big Data: Issues, Challenges, Tools and Good Practices", 978-1-4799- 0192-0/13© 2013 IEEE.
[2] Jaseena K U & Julie M David "Issues, challenges, and Solutions: Big Data Mining", Pp. 131-140, 2014. © CS & IT- CSCP 2014.
[3] Praveen kumar & Dr Vijay Singh Rathore "Efficient Capabilities of Processing of Big Data using Hadoop MapReduce", IJARCCE, Vol. 3, Issue 6, June 2014.
[4] C L Philip Chen & Chun-Yang Zhang "Data- intensive applications, challenges, techniques and Technologies: A Survey on Big Data, 0020-0255, © 2014 Elsevier Inc.
[5] Rotsnarani Sethy & Mrutyunjaya Panda "Big data analysis using Hadoop: A Survey", IJARCSSE vol. 5, Issue 7, July 2015, ISSN: 2277 128 X.
[6] Priya P Sharma, Chandrakant P Navdeti, (2014) "Securing Big data Hadoop: A Review of security Issues, threats and solution", IJCSIT, 5(2), PP2126-2131.
[7] Mukerjee A, Datta J, Jorapur R, Singhvi R, Haloi S, Akram , "Shared disk Big Data Analytics with apache Hadoop", 2012, 18-22.
[8] Humbetov S, "Data Intensive Computing with MapReduce and Hadoop", in Proc 2012 AICT, IEEE, 6th International conference PP.5.
[9] Hadoop Tutorial, Apache Software Foundation, 2014, Available: http:// hadoop.apache.org/
[10] Aditya B Patel, Manashvi Birla and Ushma Nair, "Addressing big data problem using Hadoop and MapReduce", in Proc. 2012 Nirma University International Conference on Engineering. pp. 1-5.