Iraqi Journal of Information Technology. V.9 N.4. 2019

Big Data Techniques: A Survey

Jamal N. Hasoon[1]
Department of Computer Science
University of Almutansiryah
jamal.hasoon@uomustansiriyah.edu.iq

Assist. Prof. Dr. Rehab Hassan[2]
Department of Computer Science
University of Technology
110019@uotechnology.edu.iq

Abstract

Big data refer to the large volume of data, it can be analyzed for strategic Developing and better decisions. Big Data applications exaggerate in near few years because a traditional data techniques be limited specification. Various types of distributions and technologies used to suffer the Big Data challenges are developed. A survey of recent technologies are review for Big Data. The main technologies features are studied enable to extract knowledge from Big Data. Such distributions have some limitations and may differ in offerings and capacities. The used technologies face the increasing multi-streams and Big Data challenges. In this work review the big data technologies and challenge.

Keywords

Big Data, Map, Reduce, Shuffle, Hadoop, HDFS, YARN, Internet of Things (IoT)

1. Introduction

The term big data refer to the size of data that are grown exponentially. Big data could characterized by 5Vs [1]: first V refer to the volume of data (tables, files, transaction, or records), second V referred to the velocity of processed data should be (stream, processes, near to real time, real time, or batch), third V referred to the value of data (hypothetical, correlations, event, or statistical), fourth V referred to veracity of data (accountability, availability, reputation, authenticity or trustworthiness), and variety of data (probabilistic, multifactor, unstructured or structured). The term Big Data describe a huge and complex dataset with respect to traditional method [2]; the internet, social media and other giant companies find many difficulties with the increasing of size of datasets. Operational big data and Analytical big data are two types of big data available [3]. Recently and in next few years faster growing in data, each human about 1.7 megabyte per second in 2020 (over 50 billion smart connected devices in the world).

Many optimization algorithms dealing with complex real-world problems [4] shown in feature extraction, complex function optimization, transport, engineering, bioinformatics, Data Mining, and many others. There are many application for Big Data: ***Wireless Sensor Network (Smart Grid);*** in real time, monitoring Smart grids operations achieved through multiple connections among smart meters, sensors, control centers and other infrastructures [5], ***E-health;*** connected health platforms are already used to personalize health services [6], ***Internet of Things (IoT);*** is a field of big data applications. Because of the high variety of objects, the applications of IoT are continuously evolving [7], ***Public utilities;*** such as water supply organizations are placing sensors in the pipelines to monitor flow of water in the complex water supply networks [8], and ***Transportation and logistics Services;*** Many public road transport companies are using RFID (Radiofrequency Identification) and GPS to track buses and explore interesting data to improve their services... For instance, data collected about the number of passengers using the buses in different routes are used to optimize bus routes and the frequency of trips [9]. Various Big Data projects tools used for: ***Data integration;*** [10] support data uploading and system integration by using own Data-Loader, ***Distributed storage***; distributions based on HDFS or own file system, ***Centralized management***; [11] parallel resource management for workflow to services for system control and coordination, ***Rapid and interactive analysis***; [12] machine learning tools in some distributions offers to support scalable advanced analytics, ***Security;*** [13] virtual private cloud instances, VPN IPsec for encrypted connections, network access control, and hardware isolation but still problem of how to find a balance between security and privacy rules while extracting value and knowledge from continuous streams, ***Visualization;*** [14] for example reports, dashboards, graphs, or Big Sheets to support decision making, ***Cloud computing services***; [15] offer many cloud services important for web applications that need important computing resources.

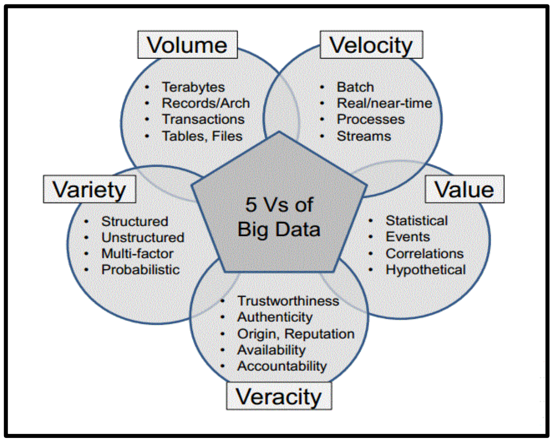

Figure 1: the 5V of big data []

2. Big Data management

Big data challenge is how to collect, integrate and store, with less hardware and software requirements, tremendous data sets generated from distributed sources [16]. Another challenge is Big Data management. It is crucial to efficiently manage Big Data in order to facilitate the extraction of reliable insight and to optimize expenses. Indeed, a good data management is the foundation for Big Data analytics. Big Data management means to clean data for reliability, to aggregate data coming from different sources and to encode data for security and privacy. It means also to ensure efficient Big Data storage and a role-based access to multiple distributed endpoints [17]. In other words, Big Data management goal is to ensure reliable data that is easily accessible, manageable, properly stored and secured.

3. Hadoop

Hadoop is the name given to a new software framework designed to use basic hardware to manage big data [18]. It is open sourced, meaning users can change, tweak or improve the software as needed. Basically, it provides the software to turn large pieces of relatively simplistic hardware into a big data network. Hadoop makes it possible to run applications on systems with thousands of commodity hardware nodes [19], and to handle thousands of terabytes of data. Its distributed file system facilitates rapid data transfer rates among nodes and allows the system to continue operating in case of a node failure. This approach lowers the risk of catastrophic system failure and unexpected data loss, even if a significant number of nodes become inoperative.

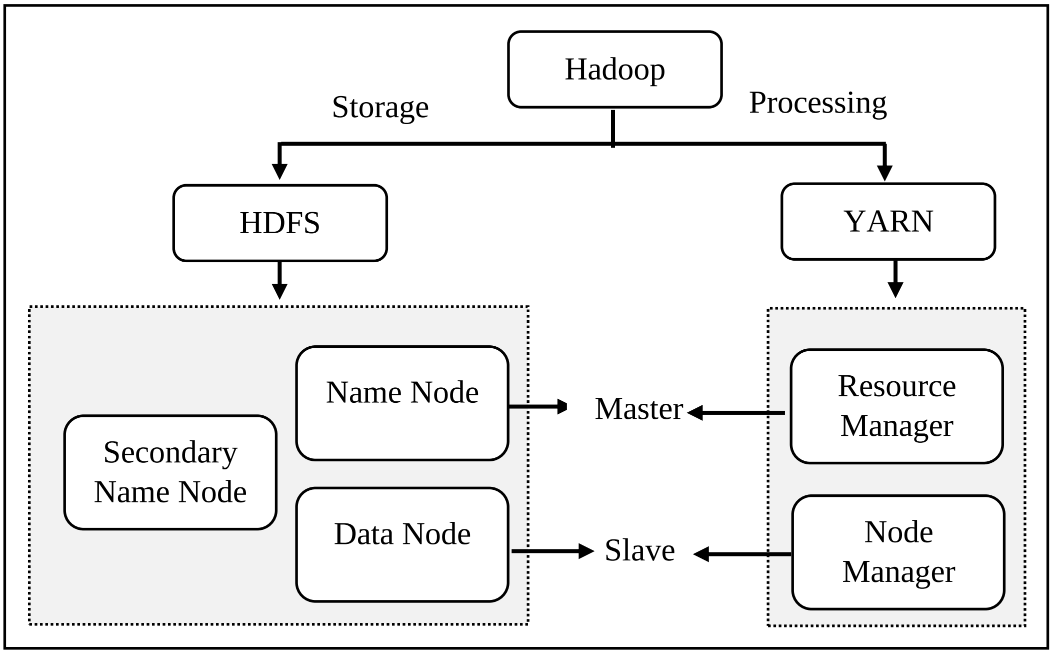

Figure 2: Hadoop framework

3.1 Hadoop Distributed File System (HDFS)

It provides a managing pools of big data and supporting analyzing. HDFS supports the transferring of data between compute nodes [20]. It was coupled with MapReduce for data processing by separating blocks and distributes them in a cluster to different nodes for efficient parallel processing and fault-tolerant. HDFS replicates each piece of data multiple times and distributes the copies to individual nodes, placing at least one copy on a different server rack than the others. As a result, the data on nodes that crash can be found elsewhere within a cluster. This ensures that processing can continue while data is recovered. HDFS uses master/slave architecture. In its initial incarnation, each Hadoop cluster consisted of a single Name Node that managed file system operations and supporting Data Nodes that managed data storage on individual compute nodes. The HDFS elements combine to support applications with large data sets. This master node "data chunking" architecture takes as its design guides elements from Google File System (GFS), a proprietary file system outlined in in Google technical papers, as well as IBM's General Parallel File System (GPFS), a format that boosts I/O by striping blocks of data over multiple disks, writing blocks in parallel. While HDFS is not Portable Operating System Interface model-compliant, it echoes POSIX design style in some aspects.

3.2 YARN

Enterprises using Hadoop should consider using 10GbE, bonded Ethernet and redundant top-of-rack switches to mitigate risk in the event of failure. A file is broken into 64MB chunks by default and distributed across Data Nodes. Each chunk has a default replication factor of 3, meaning there will be 3 copies of the data at any given time. Hadoop is “Rack Aware” and HDFS has replicated chunks on nodes on different racks. Job-Tracker assign tasks to nodes closest to the data depending on the location of nodes and helps the Name-Node determine the ‘closest’ chunk to a client during reads [21].

MapReduce 2.0 has two components YARN that has cluster resource management capabilities and MapReduce. In MapReduce 2.0, the Job-Tracker is divided into three services: Resource-Manager, a persistent YARN service that receives and runs applications on the cluster. A MapReduce job is an application. Job-History-Server, to provide information about completed jobs Application Master, to manage each MapReduce job and is terminated when the job completes. Also, the Task-Tracker has been replaced with the Node-Manager, a YARN service that manages resources and deployment on a node. Node-Manager is responsible for launching containers that could either be a map or reduce task. This new architecture breaks Job-Tracker model by allowing a new Resource-Manager to manage resource usage across applications, with Application-Masters taking the responsibility of managing the execution of jobs. This change removes a bottleneck and lets Hadoop clusters scale up to larger configurations than 4000 nodes. This architecture also allows simultaneous execution of a variety of programming models such as graph processing, iterative processing, machine learning, and general cluster computing, including the traditional MapReduce.

3.2.1 MapReduce

MapReduce is a framework for processing parallelizable problems across large datasets using a large number of computers (nodes), collectively referred to as a cluster (if all nodes are on the same local network and use similar hardware) or a grid (if the nodes are shared across geographically and administratively distributed systems, and use more heterogeneous hardware) [22]. Processing can occur on data stored either in a filesystem (unstructured) or in a database (structured). MapReduce can take advantage of the locality of data, processing it near the place it is stored in order to minimize communication overhead.

Map step: Each worker node applies the map function to the local data, and writes the output to a temporary storage. A master node ensures that only one copy of redundant input data is processed, **Shuffle and Sort step:** Worker nodes redistribute data based on the output keys, such that all data belonging to one key is located on the same worker node, and **Reduce step:** Worker nodes now process each group of output data, per key, in parallel. MapReduce allows for distributed processing of the map and reduction operations. Provided that each mapping operation is independent of the others, all maps can be performed in parallel though in practice this is limited by the number of independent data sources and/or the number of CPUs near each source. Similarly, a set of 'reducers' can perform the reduction phase, provided that all outputs of the map operation that share the same key are presented to the same reducer at the same time, or that the reduction function is associative. MapReduce can be applied to significantly larger datasets than "commodity" servers can handle a large server farm can use MapReduce to sort a petabyte of data in only a few hours [23]. The parallelism also offers some possibility of recovering from partial failure of servers or storage during the operation: if one mapper or reducer fails, the work can be rescheduled assuming the input data is still available. Another way to look at MapReduce is as a 5-step parallel and distributed computation [24]: **first;** Prepare the Map input (each processor have key work on, and the input data associated with that key value), **second;** Run the user-provided Map code (is run exactly once for each key value, generating output), **Third;** Shuffle the Map output to the Reduce processors (provides that processor with all the Map-generated data associated with that key value), **Fourth;** Run the user-provided Reduce code (run exactly once for each key value), **Fifth;** Produce the final output (collects all the Reduce output, and sorts it to produce the final outcome).

4. Scheduling in Hadoop

First in first out (FIFO) is the default scheduling algorithm in Hadoop. Research work is being taking place in scheduling a job in Hadoop [25]: **FIFO scheduler**; A job is first partitioned into individual tasks, and then loaded into the queue and assigned to free slots on Task-Tracker nodes. Each job would use the whole cluster, so jobs had to wait for their turn, **Fair Scheduler**; was developed at Facebook to manage access to their Hadoop cluster [26]. The Fair Scheduler [27] aims to give every user a fair share of the cluster capacity over time. Users may assign jobs to pools, with each pool allocated a guaranteed minimum number of Map and Reduce slots [28]. Free slots in idle pools may be allocated to other pools, while excess capacity within a pool is shared among jobs. The Fair Scheduler supports preemption, so if a pool has not received its fair share for a certain period of time, then the scheduler will kill tasks in pools running over capacity in order to give the slots to the pool running under capacity, **Capacity Scheduler** [29]; developed at Yahoo addresses a usage scenario where the number of users is large, and there is a need to ensure a fair allocation of computation resources amongst users. The Capacity Scheduler allocates jobs based on the submitting user to queues with configurable numbers of Map and Reduce slots [30]. Queues that contain jobs are given their configured capacity, while free capacity in a queue is shared among other queues. Within a queue, scheduling operates on a modified priority queue basis with specific user limits, with priorities adjusted based on the time a job was submitted, and the priority setting allocated to that user and class of job, **Longest Approximate Time to End (LATE)** [31]; all tasks should be finished for completion of the entire job. The scheduler tries to detect a slow running task to launch another equivalent task as a backup which is termed as speculative execution of tasks. If the backup copy completes faster, the overall job performance is improved. Speculative execution is an optimization but not a feature to ensure reliability of jobs, **Delay Scheduling**; [32] is a solution that temporarily relaxes fairness to improve locality by asking jobs to wait for a scheduling opportunity on a node with local data. When a node requests a task, if the head-of-line job cannot launch a local task, it is skipped and looked at subsequent jobs. However, if a job has been skipped long enough, non-local tasks are allowed to launch to avoid starvation. The key insight behind delay scheduling is that although the first slot we consider giving to a job is unlikely to have data for it, tasks finish so quickly that some slot with data for it will free up in the next few seconds. **Dynamic Priority Scheduling**; [33] supports capacity distribution dynamically among concurrent users based on priorities of the users. Automated capacity allocation and redistribution is supported in a regulated task slot resource market. This approach allows users to get Map or Reduce slot on a proportional share basis per time unit. These time slots can be configured and called as allocation interval. It is typically set to somewhere between 25 seconds and 1 minute, **Deadline Constraint Scheduler;** [34] addresses the issue of deadlines but focuses on increasing system utilization. Dead line depend on a job execution cost model that considers various parameters like map and reduce runtimes [35], input data sizes, data distribution, and Constraint-Based Hadoop Scheduler that takes user deadlines as part of its input, and finally **Resource Aware Scheduling** [36] Scheduling in Hadoop is centralized, and worker initiated. Scheduling decisions are taken by a master node, called the Job-Tracker, whereas the worker nodes, called Task-Trackers are responsible for task execution. The Job-Tracker maintains a queue of currently running jobs, states of Task-Trackers in a cluster, and list of tasks allocated to each Task-Tracker. Each Task Tracker node is currently configured with a maximum number of available computation slots.

5. Hadoop distributions

IT communities work to enrich Hadoop tools, infrastructure, and services by Sharing Big Data inventions with open source modules. The other side users may various versions of Hadoop platform may composed and each one has its own specifies may cause incompatibility platform. Technologies from different sources and its combination may suffered a hidden risks. To solve these risks, many IT Vendors have developed their own modules and packaged them into distributions. Most of Hadoop distributions have been enhanced gradually including various services such as: coordination services, distributed storage systems, interactive searching tools, resource management, advanced intelligence analysis tools, etc. some distribution as follow [37] and [38]:

5.1. Cloudera

One of the most used Hadoop distributions, it offers many benefits such as unified batch processing, centralized administration tool, role-based access control, and interactive SQL. In Cloudera explanations can be integrated with various range of infrastructure and can handle workloads with different data formats. Cloudera browsing and querying data in Hadoop in easy way and realize a real-time interactive querying. Cloudera offers a flexible model for supporting structured and unstructured data. It has been confirmed that in comparison to Hive-QL (Hive Query Language), it has disadvantages such that it is not suitable for querying streaming data such as streaming video or continuous sensor data. In addition to that, all joins operations are performed in memory that is limited by the smallest memory node present in the cluster. Cloudera robustness can be affected by the single point failure during query execution. Indeed, it quits the entire query if any host that is executing the query fails. Cloudera Enterprise RTQ does not support internal indexing for files and does not allow to delete individual rows. Cloudera wants to be like the commercial company Hadoop which was founded by Hadoop experts from Facebook, Google, Oracle, and Yahoo. If their platform is largely based on Apache's Hadoop, It is complemented with home components primarily used for cluster management. The purpose of Cloudera's economical model is the sale of licenses as well as support and training. Hence, Cloudera offers a fully open source version of their platform (Apache 2.0 license).

5.2. Hortonworks Data Platform (HDP)

Is built on Apache Hadoop to handle Big Data storage, querying and processing. It has the advantage of being a rapid, cost-effective and scalable solution. It provides several services for management, monitoring and data integration.in addition to that, HDP has been positioned as a key integration platform since it provides open source management tools and supports connections with some BI platforms. HDP ensures a distributed storage through HDFS and the non-relational database Hbase. It allows a distributed data processing based on the MapReduce, querying data through Hue and running scripts using Pig. HDP includes Oozie to manage and schedule workflows, as well as Hcatalog to handle Metadata services. Many tools are also available in HDP, including webHDFS, Sqoop, Talend Open Source, Ambari and Zookeeper.

5.3. Amazon Elastic MapReduce (EMR) is a web-based service built on Hadoop framework. It has the benefit of providing an easy, rapid and effective processing of huge data sets. it simplifies running Hadoop and related Big Data applications on AWS. It removes the cost and complexity of managing Hadoop installation. In addition, it allows resizing on demand the Amazon clusters by extending or shrinking resources. Thus, it is possible to easily extract valuable insight from big data sources without caring about the Hadoop complexity. This solution is popular in many industries and supports different goals such as log analysis, web indexing, data warehousing, machine learning, financial analysis, scientific simulation, and bioinformatics. It can handle many data source and types, including clickstream logs, scientific data, etc. Another advantage is that users can connect EMR to several tools like S3 for HDFS, backup recovery for HBase, Dynamo support for Hive. It includes many interesting free components such us Pig and Zookeeper.

5.4. MapR MapR is a commercial distribution for Hadoop designed for enterprises. It has been enhanced to provide a better reliability, performance and ease of use of Big Data storage, processing and especially analysis with machine learning algorithms. It provides a set of components and projects that can be integrated to a wide range of Hadoop ecosystem. MapR does not use HDFS. Indeed, MapR has developed it owns MapR File Systems (MapRFS) in order to increase performance and enable easy backups. The MapR-FS has the advantage of being compatible with NFS. Thus, data can be easily transferred between them. MapR is based on the standard Hadoop programming model.

5.5. IBM InfoSphere BigInsights is designed to simplify the use of Hadoop in the enterprise environment. It has the required potential to fulfill enterprise needs in terms of Big Data storage, processing, advanced analysis and visualization. The Basic Edition of IBM InfoSphere BigInsights includes HDFS, Hbase, MapReduce, Hive, Mahout, Oozie, Pig, ZooKeeper, Hue, and several other open source tools. IBM InfoSphere BigInsights Enterprise Edition provides additional important services: performance capabilities, reliability feature, built-in resiliency, security management and optimized fault-tolerance. It supports advanced Big Data analysis through adaptive algorithms (e.g., for text processing). In addition, IBM provides a data access layer that can be connected to different data sources (like DB2, Streams, dataStage, JDBC, etc.). It also leverages IBM Infosphere Streams, another tool belonging to the Infosphere set. This IBM distribution has other advantages: first, the possibility to directly store data streams into BigInsights clusters. Second, it supports real-time analytics on data streams. This is achieved through a sink adapter and a source adapter to read data from clusters. IBM facilitates also visualization through Dashboards and Big Sheets (a spreadsheet-like interface for manipulating data in clusters).

5.6. GreenPlum's Pivotal HD provides advanced database services (HAWQ) with several components, including its own parallel relational database. The platform combines an SQL query engine that provides Massively Parallel Processing (MPP), as well as the power of the Hadoop parallel processing framework. Thus, the Pivotal HD solution can process and analyze disparate large sources with different data formats. The platform is designed to optimize native querying and to ensure dynamic pipelining. In addition, Hadoop Virtualization Extensions (HVE) tool supports the distribution of the computational work across many virtual servers. Free features are also available for resource and workflow management through Yarn and Zookeeper. To support an easy management and administration, the platform provides a command center to configure, deploy, monitor and manage Big Data applications. For easier data integration, Pivotal HD proposes its own DataLoader besides the open source components Sqoop and Flume.

5.7. Oracle Big Data appliance combines, in one system, the power of optimized industry-standards hardware, Oracle software experience as well as the advantages of Apache Hadoop's open source components. Thus, this solution includes the open source distribution of Cloudera CDH and Cloudera Manager. Oracle Big Data Appliance is presented as a complete solution that provides many advantages: scalable storage, distributed computing, convenient user interface, end-to-end administration, easy-to-deploy system and other features. It supports also the management of intensive Big Data projects. The Oracle appliance lies on the power of the Oracle Exadata Database Machine as well as the Oracle Exalytics Business Intelligence Machine. The data is loaded into the Oracle NoSQL database. It provides Big Data connectors for high-performance and efficient connectivity. It includes also an open source oracle distribution of R to support advanced analysis The Oracle Big Data Enterprise can be deployed using Oracle Linux and Oracle Java Hotspot virtual machine Hotspot.

5.8. Windows Azure HDInsight is a cloud platform developed by Microsoft and powered by Apache Hadoop framework. It is designed for Big Data management on the cloud to store, process and analysis any type of large data sources. It provides simplicity, convenient management tools, and open source services for Cloud Big Data projects. Furthermore, it simplifies the processing and intensive analysis of large data sets in a convenient way. It integrates several Microsoft tools such as PowerPivot, Power View and BI features.

Conclusion

Current Big Data platforms are supported by various processing, analytical tools as well as dynamic visualization. Such platforms enable to extract knowledge and value from complex dynamic environment. They also support decision making through recommendations and automatic detection of anomalies, abnormal behavior or new trends. In this paper, we have studied Big Data characteristics and deeply discussed the challenges raised by Big Data computing systems. In addition to that, we have explained the value of Big Data mining in several domains. Besides, we have focused on the components and technologies used in each layer of Big Data platforms. Different technologies and distributions have been also compared in terms of their capabilities, advantages and limits. We have also categorized Big Data systems based on their features and services provided to final users. Thus, this paper provides a detailed insight into the architecture, strategies and practices that are currently followed in Big Data computing. In spite of the important developments in Big Data field, we can notice through our comparison of various technologies that many short comings exist. Most of the time, they are related to adopted architectures and techniques. Thus, further work needs to be carried out in several areas such as data organization, domain specific tools and platform tools in order to create next generation Big Data infrastructures. Hence, technological issues in many Big Data areas can be further studied and constitute an important research topic

References

[1] Ahmed Oussous, Fatima-Zahra Benjelloun, Ayoub Ait Lahcen, and Samir Belfkih, "***Big Data technologies: A survey***", Journal of King Saud University, Computer and Information Sciences, 2017.

[2] James Moyne and Jimmy Iskandar, "***Big Data Analytics for Smart Manufacturing: Case Studies in Semiconductor Manufacturing***", Licensee MDPI, 2017.

[3] Bhashyam Ramesh, "***Big Data Architecture***", Springer India, 2015.

[4] Kuchipudi Sravanthi, Tatireddy Subba Reddy, "***Applications of Big data in Various Fields***", (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (5), 2015.

[5] Salman Kahrobaee, Sohrab Asgarpoor, and Wei Qiao, "***Optimum Sizing of Distributed Generation and Storage Capacity in Smart Households***", IEEE TRANSACTIONS ON SMART GRID, VOL. 4, NO. 4, 2013.

[6] Wullianallur Raghupathi1 and Viju Raghupathi, "***Big data analytics in healthcare: promise and potential***", Raghupathi and Raghupathi Health Information Science and Systems 2014.

[7] Shiann Ming Wu, Tsung-chun Chen, Yenchun Jim Wu, and Miltiadis Lytras, "***Smart Cities in Taiwan: A Perspective on Big Data Applications***", Licensee MDPI, 2018.

[8] K. Thompsona, R. Kadiyalab, "***Leveraging Big Data to Improve Water System Operations***", ELSEVIER, Procedia Engineering 89, 2014.

[9] Naoufel Cheikhrouhou, Paul de Vrieze, Emanuele Giovannetti, Shaofeng Liu, Ying Xie, Lai Xu, and Hongnian Yu, "***Big Data Empowered Logistics Services Platform***" available at: https://www.researchgate.net/publication/308902135, 2016.

[10] Branka Mikavicaa, Aleksandra Kostic-Ljubisavljevica, and Vesna Radonjic Đogatovica, "***Big Data: Challenges and Opportunities in Logistics Systems***", Logistics International conference, Belgrade, Serbia, 2015.

[11] B. Arputhamary and L. Arockiam, "***A Review on Big Data Integration***", International Journal of Computer Applications, Advanced Computing and Communication Techniques for High Performance Applications, 2014.

[12] Khalid Adam Ismail, Mohammed Adam Ibrahim Fakharaldien, Jasni Mohamed Zain, and Mazlina Abdul Majid, "***Data Analysis and Storage***", Proceedings of the 2015 International Conference on Operations Excellence and Service Engineering Orlando, Florida, USA, 2015.

[13] Amir Gandomi, Murtaza Haider, "***Beyond the hype: Big data concepts, methods, and analytics*** ", ELSEVIER, International Journal of Information Management Volume 35, Issue 2, 2015.

[14] Ali Gholami and Erwin Laure, "***Big Data Security and Privacy Issues in the Cloud***", International Journal of Network Security & Its Applications (IJNSA) Vol.8, No.1, 2016.

[15] Lidong Wang1, Guanghui Wang, Cheryl Ann Alexander, "***Big Data and Visualization: Methods, Challenges and Technology Progress***", Digital Technologies, 2015, Vol. 1, No. 1, 2015.

[16] Nabeel Zanoon, Abdullah Al-Haj, Sufian M Khwaldeh, "***Cloud Computing and Big Data is there a Relation between the Two: A Study***", International Journal of Applied Engineering Research Volume 12, Number 17, 2017.

[17] Rogerio Rossi & Kechi Hirama, "***Characterizing Big Data Management***", Issues in Informing Science and Information Technology Volume 12, 2015.

[18] Raghavendra Kune1, Pramod Kumar Konugurthi, Arun Agarwal, Raghavendra Rao Chillarige, and Rajkumar Buyya, "***The Anatomy of Big Data Computing***", Cloud Computing and Distributed Systems (CLOUDS) Lab, The University of Melbourne, Australia, 2016.

[19] Zahid Javed, Tariq Shahzad, Muhammad Tehseen Qureshi, Badarqa Shakoor and Fozia Mushtaq, Review: Big Data and Hadoop (Big Data and its application), available at: "https://www.researchgate.net", 2016.

[20] Rotsnarani Sethy, Mrutyunjaya Panda, "Big Data Analysis using Hadoop: A Survey", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, Issue 7, 2015.

[21] Xiaoyi Lu, Fan Liang, Bing Wang, Li Zha, and Zhiwei Xu, "***DataMPI: Extending MPI to Hadoop-like Big Data Computing***", IEEE 28th International Parallel & Distributed Processing Symposium, 2014.

[22] N.Kiruthika, and T.K.P.Rajagopal, "***Study on Hadoop and MapReduce Framework***", International Journal of Modern Trends in Engineering and Research (IJMTER), Volume 02, Issue 04, 2015.

[23] Madhavi Vaidya, "***Parallel Processing of cluster by Map Reduce***", International Journal of Distributed and Parallel Systems (IJDPS) Vol.3, No.1, 2012.

[24] Kyong-Ha Lee, Hyunsik Choi, and Bongki Moon, "***Parallel Data Processing with MapReduce: A Survey***", SIGMOD Record, December, Vol. 40, No. 4, 2011.

[25] R.Sreedhar, D. Umamaheshwari, "***Big-Data Processing With Privacy Preserving Map-Reduce Cloud***", International Journal of Innovative Research in Science, Engineering and Technology, Volume 3, Special Issue 1, 2014.

[26] Rakesh Varma, "***Survey on MapReduce and Scheduling Algorithms in Hadoop***", International Journal of Science and Research (IJSR), Volume 4 Issue 2, 2015.

[27] Ms. Anjana Sharma, "***Hadoop MapReduce Scheduling Algorithms: A Survey***", International Journal of Computer Science and Mobile Computing, Vol. 4, Issue. 12, 2015.

[28] Harshitha R, Rekha G S, Dr. H S Guruprasad, "***A Survey on Scheduling Techniques in Hadoop***", IJEDR Volume 3, Issue 1, 2014.

[29] Prachi Srivastva, Hemant Kr Singh, and Shafeeque Ahmad, "***Job Attentive Scheduling Algorithm in Hadoop***", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 7, Issue 4, 2017.

[30] Somya Singh, Neetu Narayan, and Gaurav Raj, "***Survey on Data Processing and Scheduling in Hadoop***", International Journal of Computer Applications, Volume 119 – No.22, 2015.

[31] Divya S, Kanya Rajesh R, Rini Mary Nithila I, and Vinothini M, "***Big Data Analysis and Its Scheduling Policy – Hadoop***", IOSR Journal of Computer Engineering (IOSR-JCE), Volume 17, Issue 1, 2015.

[32] Nagina, Dr. Sunita Dhingra, "***Scheduling Algorithms in Big Data: A Survey***", International Journal Of Engineering And Computer Science, Volume 5 Issue 8 2016.

[33] M. Brahmwar, M. Kumar and G. Sikka, "***Tolhit: A Scheduling Algorithm for Hadoop Cluster***", Twelfth International Multi-Conference on Information Processing, 2016.

[34] Hadi Yazdanpanah, Amin Shouraki, and Abbas Ali Abshirini, "***A Comprehensive View of MapReduce Aware Scheduling Algorithms in Cloud Environments***", International Journal of Computer Applications, Volume 127 – No.6, 2015.

[35] Allae Erraissi, Abdessamad Belangour, and Abderrahim Tragha, "***A Big Data Hadoop building blocks comparative study***", International Journal of Computer Trends and Technology (IJCTT),Volume 48 Number 1, 2017.