ISSN No. 0976-5697

Volume 8, No. 5, May – June 2017

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

# AN ANALYTIC AND COMPARATIVE STUDY OF MAPREDUCE- A SYSTEMATIC REVIEW

Er. Kamalpreet Kaur
Department of computer engineering
Punjabi University
Patiala, Punjab, India

Er. Gurjit Singh Bhathal
(AssitentProfessor)
Department of computer engineering
Punjabi University
Patiala, Punjab, India

Dr. Gaurav Gupta
(AssitentProfessor)
Department of computer engineering
Punjabi University
Patiala, Punjab, India

***Abstract-*** In information technology, rapidly increasing in the growth rate of big data generates various issues regarding storing, analysis and processing of large amount data which is collected from various distributed sources. Such problems are tackled by Hadoop framework. Because of it provides the various services, tools and techniques which are useful to store the different format data and also provides various techniques for processing the huge data sets. A MapReduce technique is used to parallel processing of large distributed data. In this survey paper, we collect the many research papers or articles from various conferences and journal's since 2005-17. These all studies are collected from various digital libraries to find out the performances of MapReduce in the research field.

***Keywords:*** MapReduce; Hadoop; distributed computation; parallel processing; Survey

## I. INTRODUCTION

In recent years, continuous digital information is generated from various sources, which increase the computational capabilities and processing tools. So there is one of the largest challenges to software system to provide a mechanism for storage, computing power or valuable information is retrieved from large scale datasets. All the digital information is gathered from social media, web services, private-public organizations, and business intelligence companies and educational and health field, daily reaching petabytes scale data set. [9] Sometimes valuable information is not properly discovered by existing system. Because of, mostly data are stored in different data formats, using another language and so many, which are incompatible. Big companies like Google; yahoo holds a vast amount of data sets. Data from these companies, is not just evaluated by their applications or services provided by them, but also by their data sets or huge amount of information kept. Because of this meaningful information is further used in future applications. The big data concept is used to collect the very large scaled data sets which may not be processed by traditional approaches. Some challenges are involved in processing a big data as follows: capturing or storing of data, analysis of data and suitable infrastructure for processing a huge amount datasets within a reasonable time period. Such problem an acceptable solution is the use of parallel and distributed computational model which extract the relevant information from big data. Such kind of computational models is accomplished by using clusters of commodity of hardware, computational capacity at very low cost.

A MapReduce programming model is most auspicious solution for storing and processing of big data. [12] MapReduce includes the features of fault-tolerance, scalability, easy processing of large data sets. Initially Google implements technique of MapReduce, after that a Hadoop framework adopt that MapReduce techniques, then Doug cutting started their working on a MapReduce implementation.[2] A map-reduce is a parallel distribution computational model used for large data sets. In map-reduce program has two Functions. In first Map function a complex problem is divided into several parts in a key/ value pair and these sub-problems are solved directly, then assigned to cluster of working nodes. [7] The problems are solved and independently from each other. The final step of MapReduce program is to combine the solution of these various sub-problems and produces a final single result of large data sets. In simple words MapReduce is a parallel distributed computational model, is used for processing a large amount of datasets and also performing a computation on distributed data sets, which resides on cluster of systems. [7] Cluster computing is more suitable for distributed system environment. Because it provides a high throughput in a distributed manner and also provides a super computer power, storage and network communication [5] A Hadoop framework is open to source ecosystem, provides various applications and services which are useful to store, manipulate and extract the valuable information from big data in many ways. A Hadoop framework provides a parallel processing technique known as map reduce and distributed file system through HDFS for big data. [15] We can say a Hadoop framework handles all the problems of big data exploration and utilization. All these factors have made MapReduce very popular in both academia and in industry. Research communities are early adopter MapReduce, provides the evolution and development features to the framework. Over the previous years, MapReduce technique received lots of contributions from researchers and most of them are published in journals and conferences. [6]

This paper provides an overview of MapReduce to understand the modernistic research achievements for researchers and students. We present a systematic literature review on map reduce research. The main objective of this paper involves the reviewing literatures on MapReduce to find out the quantitatively evaluate the trend of MapReduce related research since 2006-20017. For this study, we collect the some articles from journals and conferences, published by various digital libraries to know the improvements in the map reduce framework for big data.

The rest of paper id organized in various sections. First section presenting the research method and discussed about a strategies use to develop this systematic review. The next section discusses about the main objective of this study and then data collection. Furthers section of this systematic review represents the selection of studies and contribution of all the studies in our research. Last section defines the conclusion of the systematic review study.

## II. RESEARCH METHOD

The main focus of this survey paper on the present status of Hadoop MapReduce research; thus contents analysis covering various digital library database. To generate this study or survey paper, we follow some strategies.

- Generate a search string using Boolean operators.
- Same search string use to collect the documents from IEEE, Science direct, and Springer libraries
- Only journal and conference articles are contributed from 2005 to 2017.
- Some selected papers are discussed in the study, which is relevant to search string.

## III. RESEARCH OBJECTIVES

This survey aims to investigate MapReduce research performance in the big data field for the period of 2005 to 2017. To achieve this aim, following objectives of research must be accomplished.

- To study about the data flow of MapReduce, its processing when a job is submitted.
- To discuss distributed computing and parallel processing of large data sets using Hadoop MapReduce.
- To evaluate MapReduce research performance across various digital libraries;
- To discuss some of related MapReduce improvement based on the selected papers from the literature.

## IV. DATA COLLECTION

The data collected in this study were obtained from the database of various digital libraries. These digital libraries are most popular academic database for research, analysis and literature reviews. We collect the various research papers from these libraries which are published from 2005 to 2017. The keyword is used such as (“MapReduce”, “map reduces”, “Hadoop”, “Big data”, “Parallel processing”, and “distributed computing”**).** By using a term of big data in our research, it returned many documents; some documents were beyond the scope of this survey paper. Thus, these topics were limited by using Boolean operators AND, OR, and Not. With the help of these operators, we create a search string which is used to collect all the required documents from digital libraries. The database covers only journals and conference articles published since 2005 to 2017.

## V. SEARCH STRATEGIE

A first question arises in research is the search strategies and search string was defined. We defined the search scope and consulted with various digital libraries.[8]

A. ***Research keyword-*** these keywords are identified by the research questions. These keywords are identified were they written exactly as defined by its developers.

B. ***Search string-*** some keyword which is built based on the research string is used according to specific need of digital libraries. The search string used to obtain the initial results of “MapReduce” reviews. This type of expression may be used in almost all digital libraries.

C. ***Source-*** basically four digital libraries are selected for this survey paper from its databases all studies of journal and conferences articles are collected. With the search string defined, we chose the following digital libraries as sources:

- IEEE Xplore
- Science Direct
- Springer’s
- ACM Digital Library

## VI. SELECTION OF STUDIES

Studies selected for further analysis on this systematic review must be published as full papers in Journals or in Conferences indexed by the selected digital libraries. After obtaining the results from the digital libraries, all studies have to be analyzed individually to confirm the relevance in the context of our review. To select or discard studies, inclusion and exclusion criteria were defined as follows. Table 1 shows the stages involved in the selection of papers.

**Table 1 Search strategies**

| | |
|---|---|
| Step 1 | Apply search string to gathering the results. |
| Step 2 | Refine the research papers, excludes duplicate papers and invalid papers |
| Step 3 | Apply the inclusion/exclusion criteria for paper title, keywords, abstract, introduction and conclusion. |
| Step 4 | Review the selected studies, inclusion/exclusion to the text. |

***Inclusion criteria-*** The inclusion of Studies is based on its relevance to the research. First, we analyzed title, keywords, abstract, its introduction and conclusion of the studies obtained from the initial search in the libraries. If at any time one of the inclusion standards was broken, the study was discarded. The inclusion criteria were the following:

- Studies concepts related to the performance of MapReduce platform and its algorithms.
- Studies have discussed the distributed file system for large data sets.
- Studies have also defined the parallel processing of big data using MapReduce.
- Studies published in Journals and/or Conferences.

**Table 2 Selected Studies**

| Digital Libraries | Step 2 | Step 3 | Step 4 |
|---|---|---|---|
| IEEE | 292 | 28 | 7 |
| Science direct | 341 | 12 | 6 |
| Springer | 246 | 14 | 4 |
| ACM | 50 | 3 | 1 |

***Exclusion criteria-*** Exclusion of studies was made by the analysis of title, keywords, abstract, introduction and conclusions when necessary, observing the following standard:

- Studies published in Workshops, books since they normally represent less mature work that later, after refinement, are improved and published in conferences and journals;
- Studies that do not have the complete full text available at the source;
- Studies that do not answer or are irrelevant to the research questions;
- Repeated studies that were published in more than one source;
- Short papers, talks, demonstrations, or tutorials;
- Similar studies published by the same authors.

## VII. CHARACTERIZATION OF THE SELECTED STUDIES

This systematic review follows based on (Kitchen ham and Chartres's methodology (2007)). This section presents the characterization of the selected studies according to our rules. Selected studies were obtained from four digital libraries: IEEEXplore, Springer's, ACM, and Science Direct. [6] Table 2 shows the numbers of studies selected from each digital library. Our final selection resulted in 18 studies from the ACM Digital Library, the IEEE Xplore Digital Library, and Science Direct Digital Library. Some more studies are also used to in the research context. Table .3 shows the year wise published papers from various digital libraries.

**Table 3 Studies by year**

| Years | IEEE | Science Direct | Springer's | ACM Digital lib. |
|---|---|---|---|---|
| 2005 | 0 | 0 | 0 | 0 |
| 2006 | 1 | 0 | 2 | 0 |
| 2007 | 1 | 0 | 1 | 0 |
| 2008 | 3 | 0 | 1 | 0 |
| 2009 | 4 | 0 | 5 | 0 |
| 2010 | 1 | 1 | 5 | 1 |
| 2011 | 9 | 13 | 7 | 2 |
| 2012 | 11 | 19 | 17 | 6 |
| 2013 | 34 | 25 | 21 | 5 |
| 2014 | 48 | 39 | 37 | 13 |
| 2015 | 91 | 125 | 53 | 8 |
| 2016 | 88 | 98 | 61 | 13 |
| 2017 | 9 | 21 | 47 | 4 |

## VIII. HADOOP MAPREDUCE

MapReduce is a parallel programming model that was designed to process or produced a large data set of clusters of computers. Google implements technique of MapReduce. Google introduced that how large data sets are split into fixed sized blocks and processed. After that a Hadoop framework adopts that MapReduce techniques, then Doug cutting started their working on a MapReduce implementation. MapReduce program is basically based on divide- conquer method. A map-reduce program splits the input data sets into various independent parts. Map-reduce are mostly used by Google, Yahoo and other many web organizations. In map-reduce program has two steps. In a first step a complex problem is divided into several parts in a key/ value pair and these sub-problems are solved directly then assigned to cluster of working nodes. The problems are solved and independently from each other. The final step of MapReduce program is to combine the solution of these various sub-problems and produces a final single result of large data sets. [11] There are two functions which are followed by the map-reduce program.

- ***Map function-*** In a Hadoop framework map function done its job into two nodes that are master node and worker node. A programmer gives the input and master node takes the input data and split into various sub-parts after that master node distribute these sub problems to worker nodes.
- ***Reduce function-*** Reduce function helps to collect the several problems with its solution and combined them in a specific way to form a final output. In reduce function they provide a list of key/values.[9]

***Distributed File system-*** Distributed file system is an client/server application in which data id reside on server and data accessed and processed by the client on their local system. The DFS makes easy to share information and files among users on a network in a controlled and authorized way. The client users can share files and store data just like they are storing the information locally with the permission of the server. However, the servers have full control over the data and give access control to the clients. [3]

**Table 4 Comparison of GFS and HDFS [1]**

| Properties | | Google file system | Hadoop Distributed file system |
|---|---|---|---|
| Implementation | Operation System | Linux | Cross-Platform |
| | License | Google (Proprietary) | Apache |
| | Written in | C, C++ | JAVA |
| | Developers | Google | Yahoo, but is open source for the community |
| Architecture | Nodes | Master node and chunk Server | Name node and data node |
| | Architecture Paradigm | Master node stores the file and location of data and make the decision regarding Storage on chuck server | Complete view of the file system is available for the Name Node. |
| | Hardware utilization | Commodity Servers and Hardware | Commodity Servers and Hardware |
| | Nodes Communication | Master Nodes convey the commands to chunk server | Name node transfers the commands to Data node |
| | Chunk Servers or Data Nodes | Chunks Stores the files on local file system after processing at user level | Chunks Stores the files on local file system after processing at user level. |
| Operation | Write Operations | Along with append operation, even random offset writes and record appends are performed | Supports at the end of document |
| | Default block size | 64 MB default, but it can be altered by the user. | 128 MB default, but the user can alter the size |
| | Deletion | Unique garbage collection process. The resources of deleted files are not reclaimed immediately and are renamed in the hidden namespace which are further deleted if they are found existing for 3 days of regular scan. | Deleted files are renamed into a particular folder and are then removed with the help of the garbage collection process. |

In more details, Hadoop MapReduce jobs are divided into various subsets of map task and reduce task that process into distributed manner on a cluster of computers. Each task works on subsets of data sets or input sets, it has been assigned so that the load is spread on numbers of computers. The map task usually performs the load, transform and filter data like functions as well as reducing task is responsible for handling and manage the single output. As there map task and reduce task in Hadoop framework is defined by various phases. The output of map task is an intermediate key and values are input to reduce task. [5]

a) ***Record reader-*** the record reader translates the subsets of input generated by input format into records. In this phase of map task data is converted into records and then pass the data records to mapper in the form of keys and value pair. [7] Usually key represents the positional information or data and values represent the block of data that creates a record.

b) ***Mapper-*** in mapper phase a user's code or information is executed on key and value pair which is generated by record phase. One or more than one key value pair produced by record reader known as an intermediate code pair. In MapReduce, the decision of making keys and values is constant to finish a job. There is key defines what data will be together and value represented the related information to analysis in reduce task.
c) ***Partitioner-*** in this phase of map task, partitioner takes input from a mapper phase in the form of intermediate keys and value pair and splits them up into group of keys a value pair known as a shard. Partitoner produced one shard to one reducer. It distributed the keys and value pair over the reducer, but still ensures that key with the same values in the different mapper end up at the same reducer. Partitoner perform the modulus operation by the numbers of reducer. At last partitioner phase pull the mapper key and value to reduce.
d) ***Shuffle-*** reduce task is started with the shuffle phase. In this phase of shuffling, output files of partitioner phase is downloading on the local machine, where a reducer is running. Individual data sets are sorted by key into one large data list. In a simple way, all the data pieces are sorted into keys with group of relevant values that can be easily iterated in the reducer task. Sorted of keys and specifying lists of values are controlled by the developer.
e) ***Reducer-*** the reducer task takes the input from shuffle phase. It takes grouped data as input and perform a reduce function one per key with their list of values. The reduce function is passed to keys and all the values which are associated with that key. [14] The reduce function produces less keys and value pairs as compare to map function they generate a more than zero key value pair to the final output. In reducer phase data can be aggregated, filtered and combined in many ways.
f) ***Output-*** final output is produced by the final key value pair of reduce function and output files is written out by record writer. At the end of MapReduce function a final output of data is written out to HDFS.

In Hadoop MapReduce architecture, the client first sends a job to the Name Node. The job can be sent either using Hadoop Query language, such as Hive or by writing a job source code. Before that, the data source files should be uploaded to the HDFS by dividing the Big Data into blocks that have the same size of data, usually 64 or 128 MB for each block. Then, these blocks of data sets are distributed among various Data Nodes within the cluster. A source file of MapReduce code, now a task has a name of the data file in HDFS and the name of the file where the results will be stored in. Hadoop architecture follows the concept of write-once and read-many, it does not allow any changes once a source file is stored in HDFS. Each sub task can access the data from all blocks. [3] Therefore, bandwidth and latency of the network is not a big problem in the cloud, where data is written one time and read many times. Many iterative computations utilize the architecture efficiently as the computations need to access same data many times. Therefore Hadoop MapReduce technique allows various jobs of the same data sets accessible independently. Several research groups have also presented solutions about data locality to address the issue of latency while reading data from Data Nodes. [14]

## IX. BIG DATA

A big data term refers to a massive amount of digital information which is amassing from various governments & private organizations, science &medical fields, IT, business, finance and various social networks. A big data involves all the information from street sensors, mobile phones and all the system-to-system communications which have some value and also useful to human beings. Big data computing is an emanating data science ensample of multifaceted information about scientific discovery & business analytics data over a very large scale – information. There are multiple dimensions which define the term of big data that are data source, processing, analysis, storing infrastructure & security. Big data is most popular to describe the continuous growth of data and availability of data both Structured (relation data), Semi-structured (XML data), Unstructured (images, audio, videos etc.). [10] But now a day's big data are defined by other terms that are SAP, which is described below: Store/capture- How to capture or store the data? Analysis- How to analysis the big data? Processing - How to process it? Big data are described by 4 V's that are as follows.[13] This definition of big data is originally coined by Doug Laney refer to challenge to data management.

a) ***Volume-*** volume refers to the amount or quantity of data that is generated by various resources. The size of the data may be into terabytes or petabytes or evenly into Exabyte.
b) ***Velocity-*** velocity defines the transfer rate of data. Velocity represents the speed of generating the data from its resources. There is fast mechanisms are required for processing, analyzing the data sets.
c) ***Value-*** values refers to a process of generating a huge amount of hidden values from a large data sets or information.
d) ***Variety-*** variety explains the different type of data coming from resources. There are structured data, unstructured data and semi-structured data which have a proper format to store an unstructured data

A Big data concept relates to Cloud computing because it provides distributed resources, by combining them to solve a complex, large-scale computation and to achieve higher throughput. By cloud computing we can use distributed resources at anywhere, any time through the internet. Cloud computing provides infrastructure as a services, platforms as a service and software as a service. It provides a middleware platform, application development, database and application servers, users can use the software's everywhere through internet. For examples Google doc, Google apps, etc. some organizations which enables the cloud computing only pay for resources and services they use. There are some challenges or problems in big data that are as store, searching, processing and analysis and presentation. [12] The traditional approach to store and process the data on a computer system is that the information or data will be stored in relational databases like Oracle, MySQL, etc. The main disadvantage of this traditional approach is that it is not compatible for large amount of data sets. To fulfill the above challenges or to solve the problems, big data have a solution that is Hadoop.

**Table 5 Comparison between Traditional database and big data issues [16, 17]**

| S. No. | Property | Traditional database | Big data solution |
|---|---|---|---|
| 1 | Data format | Structured or semi structured data | Multi-structured data |
| 2 | Data volume | Data into gigabytes. | The data range is terabytes to petabytes or evenly to Exabyte. |
| 3 | Data generating speed | Transaction-oriented | Data growth speed is high due to web and sensors etc. |
| 4 | Programming language | Query language | Data intensive computational languages like MapReduce and NoSQL. |
| 5 | Data store | Relational databases | HDFS |
| 6 | Data source | Centralized data | Distributed data |
| 7 | The theorem applies | CAP theorem with ACID properties | CAP theorem with BASE properties. |

## X. APACHE HADOOP OVERVIEW

Hadoop is open-source software. Hadoop framework is a registered trademark of Apache foundation which is written in the Java programming language. Mike cafarella and Duge cutting started Hadoop in 2005. [2] The Apache Hadoop framework is wide popular for parallel distributed processing of large data sets. Hadoop has inspired by the Google File system and Google's MapReduce distributed computing environment. [9] In big data computation a Hadoop framework provides various tools for batch processing and stream processing. Hadoop is more reliable and scalable for large number of data and cluster of computer systems. [15] The Apache Hadoop framework is basically designed to scale up from thousands of machines. Each machine offers to process and storage of data and also implemented to detect and handle the hardware failures as well as failures at the application layer.[4] There are various tools like Hbase, Hive, Pig, Spark, S4, Mahout, Avro and oozier etc. In the Hadoop framework different modules are presented, which helps in processing of large data that are as follows.

a) ***Hadoop Common-*** Apache Hadoop Common provides a set of common utilities to other Hadoop modules. [2] It has shared libraries that include Java implementations, I/O utilities, and error detection. Also included are interfaces and tools for configuration of permission of users, authentication, service-level agreement, and data confidentiality and awareness to rack.

b) ***MapReduce-*** MapReduce is a parallel processing model. MapReduce was designed to process large data sets. It divides he input into several parts then processes it and produce a single result. [18]

c) ***Hadoop Yarn-*** Apache Hadoop yarn "*Yet another Resource Negotiator*" is a resource management platform. They provide the job scheduling in clusters. Initially, Hadoop and MapReduce were tightly coupled, with MapReduce responsible for both cluster resource management and data processing. But now YARN has taken over the resource management responsibilities, allowing a separation between that infrastructure and the programming model.

d) ***HDFS-*** Hadoop distributed file system is used to store a distributed program file and that provide the high throughput access to application data. [15] HDFS replications the data in various storage nodes therefore users can access the distributed data concurrently. [15] HDFS run on top of local file system of a cluster of Hadoop and store very large data files which is suitable for streaming data access. HDFS has become an important tool for managing resources of big data and supporting big data analytics and also provides high fault-tolerance. HDFS is designed to be deployed on low-cost hardware and it introduced Master/Slave architecture. [13] HDFS has two nodes- master node (name node) and slave node (data node).

- A name node manages the hierarchy of the file system. A master node allows file system operations like modifications, closing and opening of files, access time and so on.
- Data node performs read-write operations on a file system. It allows performing block creation, deletion and updating operation to its users but according to name node instructions.

## XI. CONCLUSION

A MapReduce is a parallel processing paradigm is used for processing a vast amount of distributed data sets. A MapReduce independently communicated to Hadoop distributed file system which is used for storing the distributed sets and different type of data. So we can say a Hadoop framework is one of the most suitable solutions for big data's problems. By using the MapReduce programming model, various algorithms are used to parallel processing of large data sets to optimize the throughput. MapReduce uses various algorithms to perform its actual function. The aim of the survey paper to measure the performance of MapReduce in the research field.

## REFERENCES

[1] Ameya Daphalapurkar, M. S. (2014). Mapreduce & Comparison of HDFS And GFS. *International Journal Of Engineering And Computer Science, 3,* 8321-8325.

[2] *Apache Hadoop Foundation*. (n.d.). Retrieved JULY 2016, from http://hadoop.apache.org/

[3] Chih Fong Tsai, W.-C. L.-W. (2016, September 2016). Big data mining with parallel computing: A comparison of distributed and MapReduce methodologies. *The Journal of Systems and Software, 122,* 83-99. doi:http://dx.doi.org/10.1016/j.jss.2016.09.007

[4] Dawei Jiang, B. C. (2010). The Performance of MapReduce: An Indepth. *3*, pp. 472-483. ACM.

[5] Ibrahim Abaker, T. H. (2016). MapReduce: Review and open challenges. (pp. 389–422). Springers. doi:DOI 10.1007/s11192-016-1945-y

[6] Ivanilton Polato, R. R. (2014). A comprehensive view of Hadoop research—A systematic literature review. *Journal of Network and Computer Applications, 46*. doi:DOI: 10.1016/j.jnca.2014.07.022

[7] Jianjiang Li, Y. L. (2017). Map-Balance-Reduce: An improved parallel programming model for load balancing of MapReduce. *Future Generation Computer Systems*. doi:http://dx.doi.org/10.1016/j.future.2017.03.013

[8] Kitchenham. (2007). *Guidelines for performing Systematic Literature Reviews in Software Engineering.*

[9] Mohd Rehan Ghazi, D. G. (2015). Hadoop, MapReduce and HDFS: A Developers Perspective. (S. Direct, Ed.) *Procedia Computer Science, 48,* 45-50. doi:10.1016/j.procs.2015.04.108

[10] Praveen Murthy, A. B. (2014, September). Big Data Taxonomy. 1-33. Retrieved from http://cloudsecurityalliance.org/research/big-data/

[11] Ren Li, H. H. (2016). MapReduce Parallel Programming Model:A State-of-the-Art Survey. *44*, pp. 832–866. Springers. doi:DOI 10.1007/s10766-015-0395-0

[12] Salvador García*, S. R.-G. (2016). Big data preprocessing: methods and prospects. (pp. 1-9). IEEE. doi:DOI 10.1186/s41044-016-0014-0

[13] Seema Maitrey, C. J. (2015). MapReduce: Simplified Data Analysis of Big Data. *the 3rd International Conference on Recent Trends in Computing . 57*, pp. 563-571. Elsevier. doi:doi: 10.1016/j.procs.2015.07.392

[14] Sherif Sakr, A. L. (October 2013). The Family of MapReduce and Large-Scale Data Processing Systems. *ACM Computing Surveys*, *46,* p. 44. doi:http://dx.doi.org/10.1145/2522968.2522979

[15] Taylor, R. C. (2010). An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics. *11*, pp. 1-6. Springers.

[16] Triguero, I., Daniel, P., bacardit, j., Garcia, s., & herrera, f. (2015). MRPR: A MapReduce solution for prototype reduction in big data classification. *Neurocomputing*, 331-3345. doi:10.1016/j.neucom.2014.04.078

[17] *Tutorial Ponits*. (n.d.). Retrieved july 2016, from www.tutorialpoints.com: http://www.tutorialspoint.com/hadoop/

[18] Wonhee Cho, E. C. (2017). Big data pre-processing methods with vehicle driving data using MapReduce techniques. Springers. doi:DOI 10.1007/s11227-017-2014-x