Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IMPACT FACTOR: 6.017

IJCSMC, Vol. 5, Issue. 12, December 2016, pg.92 – 102

An Improved Performance Evaluation on Large-Scale Data using MapReduce Technique

P.Selvaramalakshmi [a], Dr.S. Hari Ganesh [b], Dr. S.Florence Tushabe [a,b,*]

[a]Research Scholar, Bishop Heber College (Autonomous), Tiruchirappalli, Tamil Nadu, India
Email: lakshmibhabuu@gmail.com
[b]Asst. Professor, Dept. of Computer Science, H.H. The Rajah's College (Autonomous), Pudukkottai, Tamilnadu, India
Email:hariganesh17@gmail.com
[a,b,*]Associate. Professor, Dept. of Computer Science, UTAMU, Kampala, Uganda

***Abstract:** In a day-to-day life, the capacity of data increased enormously with time. The growth of data which will be unmanageable in social networking sites like Facebook, Twitter. In the past two years the data flow can increase in zettabyte. To handle big data there are number of applications has been developed. However, analyzing big data is a very challenging task today. Big Data refers to initiatives and technologies that involve data that is too diverse, high-speed-changing or huge for conventional technologies, skills and infrastructure to address efficiently. The current infrastructure to handle big data is not efficient because of data capacity. The processing of big data problem can be solved by using MapReduce method. The efficient implementation of MapReduce model requires parallel processing and networked attached storage. Hadoop and Hadoop Distributed File System (HDFS) by apache are commonly used for storing and managing big data. In this research work we suggest different methods for catering to the problems in hand through MapReduce framework over HDFS. MapReduce method has been studied at in this paper which is required for implementing Big Data analysis using Hadoop Distributed File System.*

***Keywords:** MapReduce; Big Data; Zettabyte; Hadoop; Hadoop Distributed File System.*

1. Introduction

Today Internet Services are most popular computer applications with millions of users. Internet Services such as e-commerce websites and social networks manage with huge volumes of data [2]. These services generate large volume of data from millions of users every data, which is potential gold mine for understanding access patterns and increasing ad revenue. The Internet conveys large amounts of data from traditional sources like forms, research institutions and surveys and government organizations [5].

The exponential growth and availability of structured and unstructured data can be described by using the popular term called Big data. Big data analysis is the method of examining data to discover unknown relationships, hidden patterns and other useful information that can be used to make better decisions [11]. Big data analysis not only helps you to understand the information contained within the data, but it will also help to recognize the data that is most significant to the business and future business decisions. The traditional database and software techniques are not efficient to process Big data. Today Analyzing big data is a very challenging task [6].

MapReduce is one of the programming model that efficiently processing the analysis of big data on a huge number of commodity machines [7]. Efficiently process the number of WebPages collected from all over the world the MapReduce model was developed for the back end of Google's search engine to enable a large number of servers [1].The developers to analyze big data the middleware and an Application Programming Interface provided by MapReduce Framework [6]. MapReduce framework using a distributed file system (DFS) to initially partition the data into multiple machines using and it is represented as (key, value) pairs. The calculation is carried out by two user defined functions one is map and another one is reduce function. The map procedure takes as input a (key, value) pair and produces a list of new (key, value) pairs as output. The reduce function takes map function output as the input and then return the output of new list of (key, value) pairs. The MapReduce mechanism hidden the complexity involved in developing a system that works on many servers from the developer [12].

Hadoop Map Reduce is a technique which analysis big data. Hadoop applications use Hadoop Distributed File System (HDFS) for primary storage. Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware [1]. It is inspired by the Google File System. HDFS designed to hold very large volume of data (terabytes or petabytes or even zettabytes), and provide high throughput access to this information [2].

Objective of study

The three main challenges being faced in big data are volume, variety and velocity [8]. Volume refers to the quantity of data to be processed, variety is the capability to handle different types of data and velocity indicates the speed at which the data are processed. The big data capture, control and process the large volume of data in a well-organized way [9]. Today, large volumes of data are in an unstructured manner. It is very difficult to perform the process in unstructured data. As a result the data required to be structured in order to perform some tasks. The data can be structured and the produce the result based on user choice by using Hadoop Map Reduce and combined filtering method [4].

The rest of the paper is devised as follows: Section 2 deals with the fundamental concept of Hadoop Distributed File System (HDFS) and MapReduced framework, Section 3 describes the proposed work, Section 4 describes the Hadoop Map reduce method, Section 5 illustrates the Collaborative filtering and Recommendation generation, Section 6 explain the performance evaluation and Section 7 reaches the conclusion of this paper.

2. Related Work

2.1 *Fundamental Concept of Mapreduce:*

MapReduce framework has the syntax of map function and reduces functions. MapReduce technique allows distributed procedure for Map/Reduction functions. MapReduce [4] is an easy programming model for processing large volume of data sets in parallel.

The basic concept of MapReduce is to divide a task into subtasks, process the sub-tasks in parallel, and finally combined the results of the subtasks to form the final output that can be shown in Figure 1.

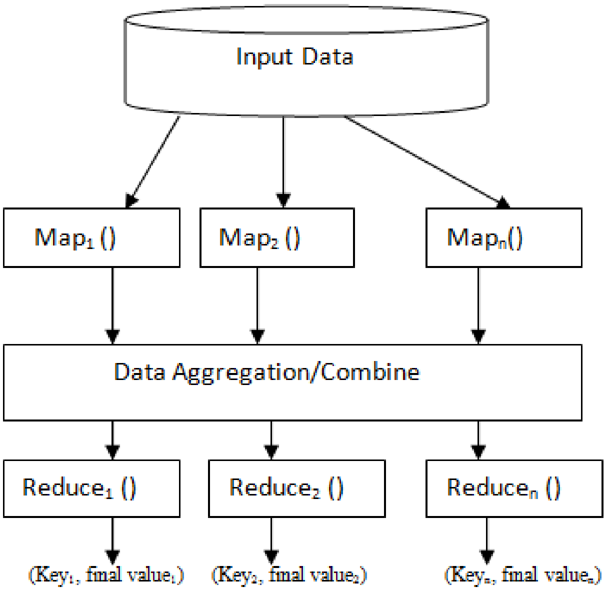

Fig. 1. MapReduce Architecture

The programmers do not need to be worried about the implementation details of parallel processing in MapReduce framework because its programming model is automatically parallelized, so that the programmers can write only two functions: map and reduce [15]. The map function reads the data input in parallel and distributes the output data to the reducers. The reduce function takes output from the map function and then produce a list containing all the values output with that key [8].

2.1 Programming Model of MapReduce in Parallel Computing:

MapReduce program uses Functional programming to analyze big data. The programming model of MapReduce is executed in the Apache Hadoop project. Hadoop can make hundreds of nodes that process and compute tera-bytes of data working simultaneously. Hadoop was inspired by Google's MapReduce and GFS to process large volume of data set for information retrieval and analysis [5].

Hadoop [3] contains File storage and Distributed processing system components. The file storage component is called “Hadoop distributed file system (HDFS)”. It provides reliable, scalable and low cost storage. Hadoop applications use Hadoop Distributed File System (HDFS) as the primary storage system. HDFS creates multiple copies of data blocks and distributes them on compute nodes. HDFS stores files across a set of servers in a cluster. Hadoop Distributed File System ensures availability of data by repeatedly monitoring the servers in a cluster and the blocks that they manage [4].

The second most important components of Hadoop are Distributed data processing system called "MapReduce" [10]. Both Hadoop distributed File System and the MapReduce frameworks are running on the identical set of nodes. The applications specify the input and output position and supply map and reduce function through implementations of suitable interfaces and abstract classes [6].

Figure 2 shows master/slave architecture of Hadoop Distributed File System. An HDFS [4] contain a Master server called NameNode which manages the file system namespace and control the clients to access the files. In addition, there are a set of DataNodes which control the storage attached to the nodes that they run on it. HDFS exposes a file system namespace and allows user data to be stored in files and then the file is divided into one or more segments and these segments are stored as a set of DataNodes [6].

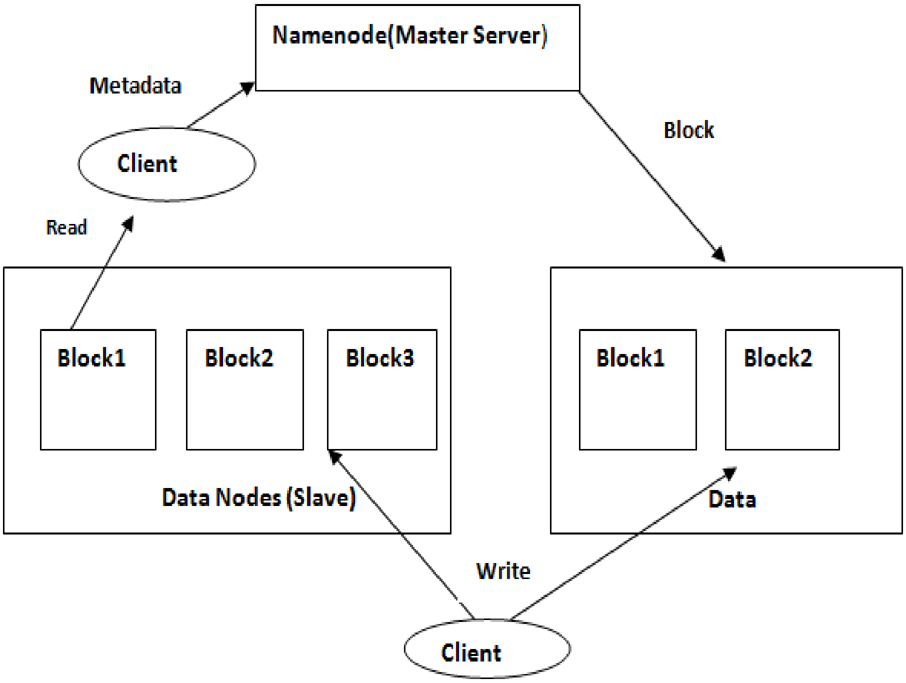

Fig. 2. HDFS Master/Slave Architecture

3. Proposed System

Processing large volume of unstructured data requires structural arrangement of the given data. Hadoop is binary well-suited with Map reduce. Map Reduce is a shuffling plan to perform filtering and aggregation of data analysis operations. Map is a filtering method [7] used for filtering the datasets and Reduce is a method used for aggregation of data sets. The Hadoop distributed system for the map reduce job is illustrated in Figure 3.

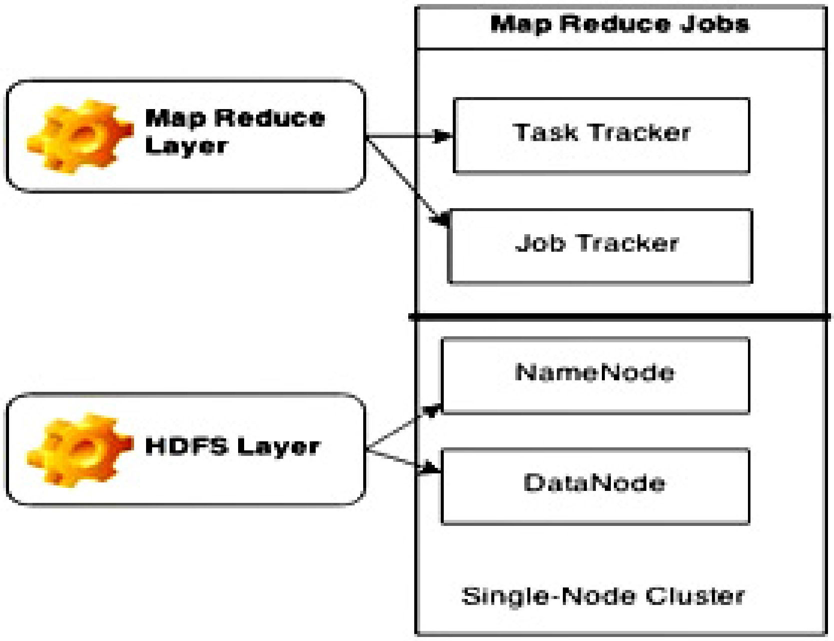

Fig. 3. Map reduce Jobs

3.1. Volume, Velocity and Variety

The Volume, Variety and Velocity are basic requirements for grouping the large volume of data before the actual structuring process is done. Large volume of data is represented by means of Big Data analytics [2]. Every day the number of people interacting with the social media such as Twitter and Facebook, is increasing drastically. The data comes from any sources like databases and excel sheets structured data and unstructured data is in various other forms like pictures, audio, video. So that it is important to analyze the large volume of data. The Velocity specify the speed at which the data are processed. The term variety refers to handle structured data or unstructured data.

3.2. Processing Big Data from social sites

Data items from Social websites having some weakness that has to be processed and used powerfully. The data coming out from a single user's Facebook or Twitter or Google accounts helps us to track the user's interests in different aspect [13]. These data items can be collected and then store them in a repository for our future use. The repository having the ability for avoiding the data crash or loss.

Big data classification can be a long and complex process[12,13]. Predictability is one of important property in Big data field which can be used to predict the data to the user based on their choice. We can merge or organize the data items based on their property. First we are arranging the data items and then sort the data items based on any one of their property. The third step is to generate a database which consists of the processed data. Finally the comparison operation is performed with other users information in the database when we processing the new user data. When the user's interests or any other property matches with any other users in the database, then based on their interests we show the subsequent information to the new user. The new user’s processed data to be added into the database. As a result, now the interests of the users are checked and the pattern which matches exactly to the new user is considered.

4. Hadoop Mapreduced Implementation Method

In big data analysis, to run applications on systems that involves thousands of nodes containing terabytes of data [1]. Hadoop is an open-source Map reduce implementation designed for large clusters. Hadoop makes it possible because the HDFS is the primary storage system used by Hadoop applications. Hadoop Distributed File System contains a single master node called the JobTracker and many slave nodes called the TaskTrackers [3]. A single point failure doesn't affect the entire system. It also helps us to continue operation even when there is node failure.

The uncertainty in the data should be removed when we process the data set obtained from the social sites like Twitter or Facebook by using Hadoop MapReduced implementation method. It has been done by arranging the data item based on their types either structured or unstructured and then the arranged data items are sorted for better understanding [8].

The MapReduce processes the data item and then arrange data item based on user requirements. There are two types of HDFS nodes , one is DataNode and another one is NameNode. The DataNode stores the data blocks of the files in HDFS and NameNode that have metadata [8].

4.1. MapReduce

MapReduce allow for distributed processing of the Map/Reduction functions. The MapReduce processing infrastructure contains an “input split” that allows each data block to be broken into separate records.

A) Input Reader

The *input* reader divides the input data into proper size and the framework allocates one split to every map function. The input reader reads data from established storage and produces key/value pairs as an output.

B) Map Function

The *map* function process a key/value pair to produce another key/value pair. A number of map functions running in parallel on the data that is divided across the cluster, create a set of intermediate key/value pairs.

C) Compare Function

The input for every reduces is pulled from the machine and stored using the applications *comparison* function.

D) Partition Function

The *partition* function is specified the key and the number of reducers and returns the indexes of the required reduce. It is significant to choose a partition function that provides an approximately consistent distribution of data reducers allocated more than their share of data for load-balancing process to complete.

E) Reduce Function

The *reduce* function merge all the intermediate values that are related with the same intermediate key.

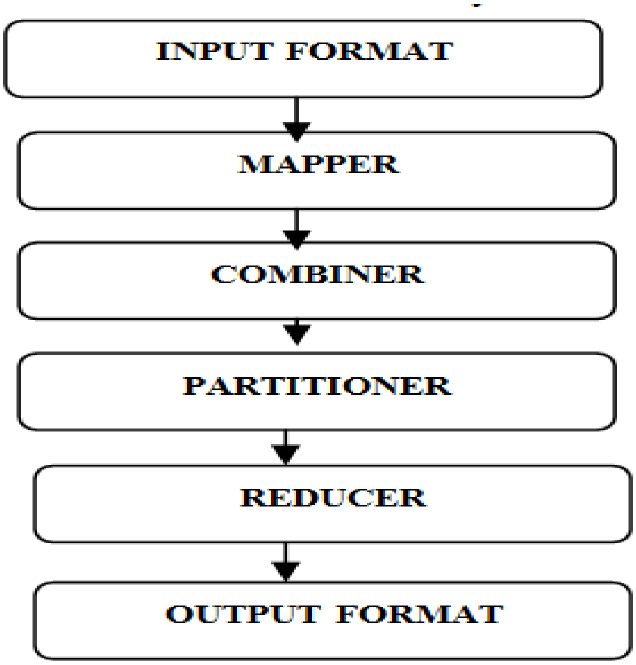

Fig. 4. MapReduce programming model

In case of a single node failure in MapReduce framework, the map tasks and incomplete reduce tasks will be re-executed instead of executing the complete map and reduce tasks. We can accomplish the minimum execution time. The Map Join Reduce method is used for the processing the heterogeneous data items.

4.2. Processing flow of MapReduce jobs

The Map job has three sub tasks such as Mapper, Combiner and Partitioner. Mapper performs the mapping of data, Combiner combines the mapped data and Partitioner divide the data into small clusters, after which the shuffling key/value of map job to unique reduce job is completed [15]. Reduce job has two subtasks namely joiner and reducer. The joiner performs the joining of the intermediate results from the map jobs and reducer subtask is used for performing aggregation. After the map and reduce jobs, the final result is stored in Hadoop Distributed File System [HDFS]. Figure 5 shows the execution flow of MapReduce process.

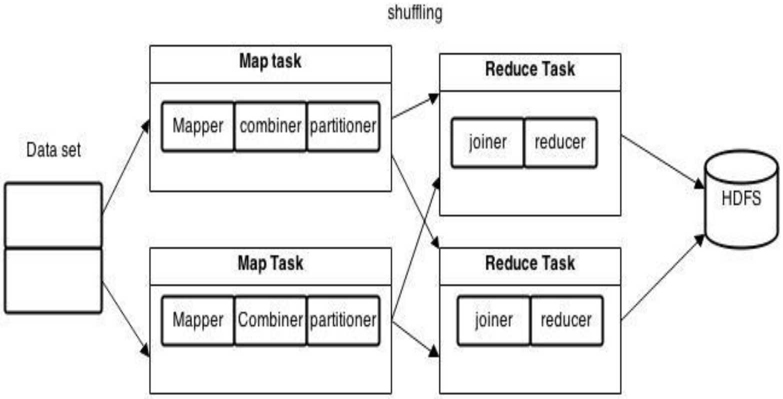

Fig. 5. Execution Flow of a MapReduce process.

4.3. Datasets

Map reduce is applied in Twitter Datasets which contains information like user name, location, department, tweet and expressions. Hence, the Twitter dataset is specified as input to map-reduce method for the modification and structuring.

4.4. Running Hadoop MapReduce

The Hadoop MapReduce program is given with the twitter data sets as input and the MapReduce method is run for the 'N' number of data in the dataset. This method can be executed for inputs of any size. Hadoop MapReduce program model can supports efficient and fast processing of the data, by which unstructured data on any volume can be structured effectively [1]. In order to avoid File already exist exception, the output file generated by this MapReduce method has to be removed every time before running it.

4.5. Hadoop Distributed File System Results

The output generated from Hadoop distributed File System (HDFS) for a MapReduce job can be used to store the final results of map reduce process and the result can be viewed by browsing the file system in the NameNode log. The JobDetails log produces the jobs that are finished while processing of the MapReduce procedure. The information about the cluster summary, distributed file system used, capacity of the file systems and also the number of live nodes and dead nodes stored in NameNode log [3]. The details of NameNode and JobTracker are accomplished as the outcome of the execution of Map Reduce task. The output directory of the file system and the output of the map reduce job can be discovered by using NameNode log.

Job Details log: Log have information regarding the type of jobs and reports the user whether the job is finished or running or killed.

4.6. Structuring of the unstructured data

After the successful implementation of the MapReduce program, the resulting data set is structured with a particular order based on the user requirements.

5. Prediction Based On Collaborative Filtering

An efficient processing of queries in the big data and to maintain the effective storage structure the collaborative filtering method is applied after the data is structured by using MapReduce technique. Predicting the requirement of an individual user by finding the similarity among past data of all users and the data of the present user can be done by using Collaborative filtering method.

6. Comparison and Performance Evaluation

MapReduce Access Pattern (MRAP) performs multiple sequential reads per map task, whereas in MapReduce read performs only single read for a single map task. MRAP restructuring eliminates multiple map reduce phases to improve the access patterns for data restructuring. However, in Collaborative Filtering restructuring method, the prediction is based on the user's suggestions to produce a Recommendation System. The Figure 6 shows the comparison between the existing and proposed Systems. Table 1 demonstrate the comparison between the Existing and Proposed Systems.

Table 1. Comparative Analysis between Existing And Proposed Systems

Features	Existing System	Proposed System
Jobs Performed	Hash Algorithm	Map Reduce Tasks
Data Restructuring	MRAP Tasks	Collaborative filtering
Recommendations Based on	MRAP Restructuring	User's Prediction
Sentiment Analysis	Emoticons and tagging	Emotion Score

For filtering and aggregation of jobs the Map Reduce method takes less time than Hash algorithm .The hash algorithm takes more time because it has to compute the hash values for both map and reduce functions and store it in a hash table. Whenever it executes the job, the hash value is retrieved from the hash table and combines it with the number of tasks it carry out and generate the result. However, the MapReduce technique need not calculate the hash value and just it execute the mapping and reducing of the data. The collaborative filtering technique takes less time for finding the tweet of the users illustrated in Figure 7. MRAP takes more time because it is a more complex operation to create map and reduce model. The time complexity as a whole becomes O (log n).The Hash algorithm takes extra space than the MapReduce because it needs a lot of space for storing the hash values and hash tables that can be illustrated in figure 8.However, the MapReduce does not require the space to be allocated individually for hash table and hash values. The MRAP requires a large quantity of space to be owned for storing the large amount of map and reduce access patterns. The collaborative filtering does not need space for access patterns, but it occupies very low space for storing the recommendations. The space complexity is O (n).

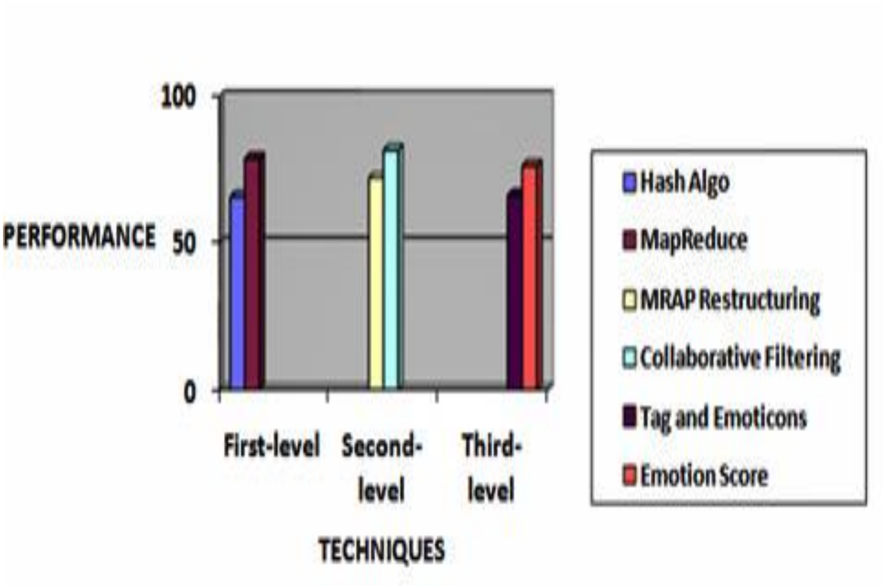

Fig. 6. Performance comparison between existing and proposed systems

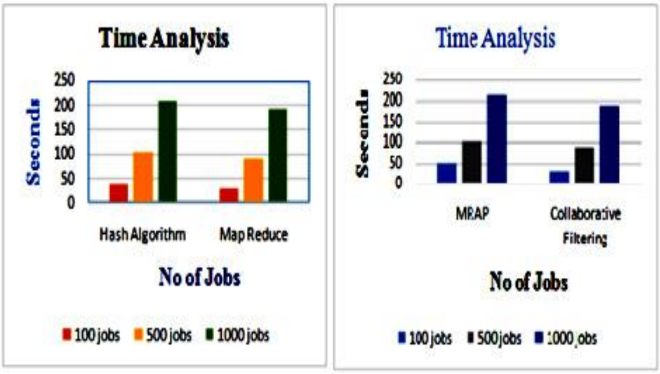

Fig. 7. Time complexity analysis between existing and proposed systems

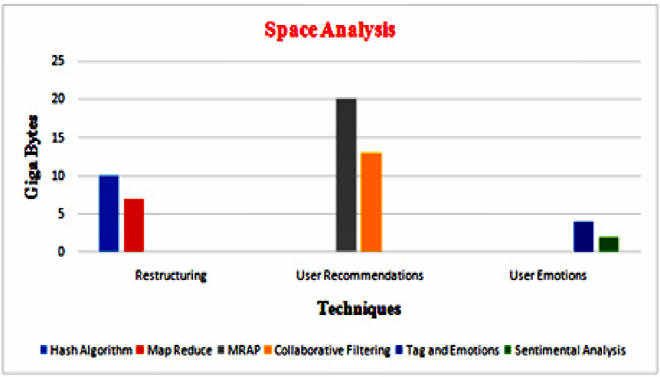

Fig. 8. Space complexity analysis between existing and proposed systems

7. Conclusion and Future Work

MapReduce framework is the most efficient method for processing large amount of data and the application of collaborative filtering provides recommendation creation for any number of data provided as input. MapReduce over Hadoop and HDFS, guarantees faster advances in many scientific disciplines and improving the profitability and success of many enterprises. This paper uses the MapReduce framework for efficient analysis of big data and for solving challenging data processing problems on large scale datasets in various domains.

In future the developed method can be enhanced and the recommendation generation process can be even more efficient and optimized efficiently.

References

[1] Hadoop,"PoweredbyHadoop,"http://wiki.apache.org/hadoop/PoweredBy.

[2] Bakshi, K. (2012) Considerations for Big Data: Architecture and Approach. *IEEE Aerospace Conference,(pp.1-7).Big Sky,USA.*

[3] Hadoop Tutorial, YahooInc., https://developer.yahoo.com/hadoop /tutorial/index.html

[4] Apache: Apache Hadoop, http://hadoop.apache.org.

[5] Hadoop Distributed File System (HDFS), http://hortonworks.com/hadoop/hdfs/.

[6] Afrati, F.N. & Ullman, J.D. (2011) Optimizing Multiway Joins in a Map-Reduce Environment. IEEE *Transactions on Knowledge and Data Engineering*, 23(9), 1282-1298.

[7] Y. Yuan, Y. Wu, X. Feng, J. Li, G. Yang, W. Zheng, "VDB-MR: MapReduce-based distributed data integration using virtual database", Future Generation Computer Systems 26 (2010) 1418–1425.

[8] Hadoop: open source implementation of MapReduce, <http://hadoop.apache.org/mapreduce/>.

[9] R. Baraglia, G. D. F. Morales, and C. Lucchese. Document similarity self-join with MapReduce. In ICDM, 2010.

[10] Y. He, H. Tan, W. Luo, H. Mao, D. Ma, S. Feng, and J. Fan. Mr-dbscan: An efficient parallel density-based clustering algorithm using MapReduce. In ICPADS, 2011.

[11] Jianqing Fan1, Fang Han and Han Liu, Challenges of Big Data analysis, National Science Review Advance Access published February, 2014.

[12] Lee, D., Kim J-S. & Maeng, S. (2013) A Large-scale incremental processing with MapReduce. *FutureGeneration Computer System*, *36*, pp 66-79.

[13] VinayakBorkar, Michael J. Carey, Chen Li, Inside "Big Data Management": Ogres, Onions, or Parfaits?, EDBT/ICDT 2012 Joint Conference Berlin, Germany,2012 ACM 2012, pp 3-14.

[14] Jiang, D., Tung, A. & Chen, G. (2011) MAP-JOIN-REDUCE: Toward Scalable and Efficient Data Analysis on Large Clusters. *IEEE Transactions on Knowledge and Data Engineering*, 23(9), 1299-1311.

[15] Gu, R., Yang, X., Yan, J., Sun, Y., Wang, B., Yuan, C. & Huang, Y. (2014) SHadoop: Improving MapReduce Performance by Optimizing Job Execution Mechanism in Hadoop Clusters. Journal of Parallel and Distributed Computing, 74(3), 2166-2179.