

Apache Hadoop for processing image files using Sequence file

¹Dr. E. Laxmi Lydia,²Hima Bindu Gogineni , ³Ravva Ravi, ⁴G Sandhya,⁵G. Jose Moses

¹Professor, Vignan's Institute of Information Technology (A), Department of Computer Science and Engineering, Visakhapatnam, Andhra Pradesh, India.elaxmi2002@yahoo.com

²Assistant Professor,Department of Computer Applications, Vignan's Institute of Information and Technology

³Assistant Professor,Vignan's Institute of engineering for Women

⁴Assistant professor,Vignan's institute of engineering for women

⁵Professor, Computer Science and Engineering,Raghu Engineering College (Autonomous), Viskahpatnam (Andhra Pradesh), India.

Article Info

Volume 82

Page Number: 1232 - 1244

Publication Issue:

January-February 2020

Abstract

MapReduce as an advanced approach to process data in a distributed manner by taking advantage of the Hadoop framework which is an open-source for employing a tremendous volume of data. Data available in multimedia at excessive quantity in the progressive world allows new demand for processing and storage. Hadoop working as a distributed computational framework as open-source to all available data considers the processing of different forms of data (such as images) on a thriving organization of calculating nodes by providing essential associations. This will accept loads and plenty of images files and used to abolish replicate files from the feasible appropriate data. Compressed binary format data or encrypted binary format data, in particular, cannot be partitioned but can only be read as the distinguished consecutive flow of data. Practicing such files as input to a MapReduce task (Job), such that the process of single mapper allowing performing till the entire file gets processed, provoking a conceivable large performance potency accomplishment. The paper intends to present a suitable splittable format of data using SequenceFile and MD5 algorithm results to enhance the effectiveness of image processing.

Article History

Article Received: 14 March 2019

Revised: 27 May 2019

Accepted: 16 October 2019

Publication: 06 January 2020

Keywords: Sequence File, MapReduce, Distributed Processing, Image files, Hadoop

INTRODUCTION

The endless allowance of image data has to turn into substantially as backward due to the increase of convivial networking, scrutinized cameras, and sources from satellite images. The indicated multimedia data is not unnatural for the progress. Another field too processes

enormous data on this planet, mentioned as Big Data technology grabbed its attended among top vital technologies[1]. Big Data points to the tremendous determinations of data collection, later examine which also leads to vigorous times and manages to exploit necessary database management mechanisms[2]. With the result of the

developed evaluation of consistent authorization of data and because of the expanding measure of data that is getting to be accessible consistently and accelerated modernized expansion. The size of massive data cannot be absolutely mentioned as it increases day after day. Datasets with multi-terabytes data vary with thousand gigabytes for one terabyte to multi-petabytes data varies with thousand terabytes for one petabyte are designed as Big data. Maintain large volumes of data may provide helpful information and also it appends some predicament in the storage of data which costs depending on the security measures and its privacy measures [3]. Working on the major issues that utilize a high stockpiling limit within multiprocessor systems is the primary measure. The smart way to increase data size, successful processing requirements and applying various programming maintaining quality and algorithms in Hadoop can abundantly deal with new demands [3]. Message Passing Interface(MPI) is an interface within distributed systems that follows a successive procedure for processing large-scale voluminous data (image data).

Future what a distributed system needs are parallel data processing. To execute and process the parallel computations [14] of algorithms, that completely depends on system programmer and developer. Ahead of the absence of specialists and professional developers they were not usually perceived [4] Big Data.

One of the most prominent big organization i.e, GOOGLE that handles Big Data has introduced a programming model (MapReduce)[5]. The introduced model is a process in distributed systems. This also helps in maintain secure administration of distributed intact. The organizations deal with the primary existence of uniform issues in the introduced model in forming a cluster and its computations. The demand to set up the

systems obligate with energy, moderate systems, substantial duration, imperative hardware and software. These preconditioned companies and enterprises are unreasonable for fewer and average size [4]. The interference of distributed computing was defined by the observations of the customers with the basic production of hardware and software benefits. Its requirement of the sources are based on the volume of computing nodes and simply lease them based on the run time of the algorithm and its execution time. The generated PDF files day after day in 2007 by New York times are identified. Amazon cloud computing has converted eleven million photographs with four terabytes within 24 hours into PDF format with the help of a hundred nodes. The accomplishment of the task was continued by the employment of various applicable algorithms and general systems [6]. A model named MapReduce was identified in this paper on the basis of the latest distributed processing. This model is expulsion for an open-source Hadoop framework that specifically holds the entire work process of image processing, continued through distributed processing systems based on the framework pros and cons. Considering the efforts of the framework, the MapReduce model has conferred the distributed processing through Hadoop for all the image through image processing [17]. Through this process, the readers are advised to use cloud computing and various computations over online and computer processing for applicants beyond the number of various contemplations and resources. The applicants for the allowance of resource utilization, pay the cost. The cutting of edge data novelty was provided by the advert to source number [7]. Some of the most implemented frameworks are promoted in image processing as a factor of Hadoop. HIPI (Hadoop Image Processing Interface) [8] is a structure that precisely expected to allow image processing in Hadoop.

CONSEQUENCE OF DISTRIBUTED FRAMEWORK IN DATA PROCESSING

A large amount of unstructured data has been processed for storage and analysis through an open-source design through Hadoop[8]. The importance of this framework shows the origin of processing data through search engines such as google and yahoo. These search engines were in need of improved and advanced models and tools for processing web page searching and indexing. Hadoop framework is exclusively designed for parallel processing of data at 'n' number of computer nodes. The data size varied from petabyte and Exabyte within the processing of data in clusters that are efficiently distributed horizontally. Apache foundation was the designer and developer of the Hadoop source in 2005. It was the name of the elephant toy of Doug Cutting Son and Mike Cafarella who were working at Yahoo. This was originally designed to bolster provision for the Nutch internet pioneer enterprise. The Apache Hadoop design process was made depending on the modules as mentioned below [9]:

- This structure consists of uniform libraries and utilities necessary by more alternative Hadoop modules.
- This structure consists of a distributed file system named Hadoop Disturbed File System (HDFS) which results in data on stock machines, determining exact excessive bandwidth over the cluster.
- It maintains a resource management system called Hadoop Yarn which is subjected to govern the resource computations in clusters and practicing them for setting up of user's functional applications.
- It is a parallel programming model to process large-scale data known as Hadoop MapReduce

FUNCTIONAL WORK MECHANISM IN HADOOP

Tremendous data files from the sources transfer the log files through social networks and more data sources are disintegrated through a distributed network in this framework. Distribution, procurement, and improving immense files on the Hadoop cluster is undertaken by its diffused documented texture called HDFS. To increase the potency of this structured design, multiple numbers of computing nodes are allowed to process distributed documents [15]. Accordingly, if any disturbance or collision of work in between the process, it gets quit and recovered back through the three computational nodes in HDFS [10]. Firstly, the name node managing computational node that allows and makes a record of all sharing files and deposit the region of every part. Secondly, Infrequent analysis of machine nodes and managing the file system in Hadoop decides their removal of unnecessary data. File blocks are the components to store the data i.e, Information to Hadoop PCs known as information node. Another node named Management node holds and progress the entire available data from the information node-set. Thirdly, the resultant node that helps to provide replication of nodes and manages accordingly. Thus, even though the nodes get lost, information within the nodes won't be lost. The layout of the file managing system of Hadoop is represented in figure 1. Ensuring the distribution of information Hadoop system, MapReduce sector [16], process and analysis will be completed [11]. Figure 2 represents the entire work progress by considering the fundamental steps. The applicant addresses the challenging issues to the nodes and executes them using (job tracker). suggestions were often made through addressing than making of java accent. Through the working of Job tracker, it checks the files according to the necessity mentioned by the applicants.

Furthermore, the management node identifies the working computational machines that are included in the cluster. Consequently, the requests were given to each and every node that exists in the cluster. These nodes are classified in terms of tasks known as task trackers. They evaluate the data and perform the given task by the job tracker parallel by executing the Map limit [12]. Once the work is executed by the task tracker, the same machine node saves the result. Obviously, one single node will be using all the data appealed by the results. All the obtained traditional results from the task tracker to the job tracker delivers

the content to the reduce function of these nodes.

For this reason, it handles the outputs of the applicants which is shown in the previous figure through nodes. Immediately, the outputs are executed successfully by MapReduce and the accurate adaption of the results is identified through Big Data analysts. This approach of administering data was implemented rightfully on the issues or acknowledged planning for data analysis and perhaps found on exchanging the concluded data within relational databases or data repositories [13].

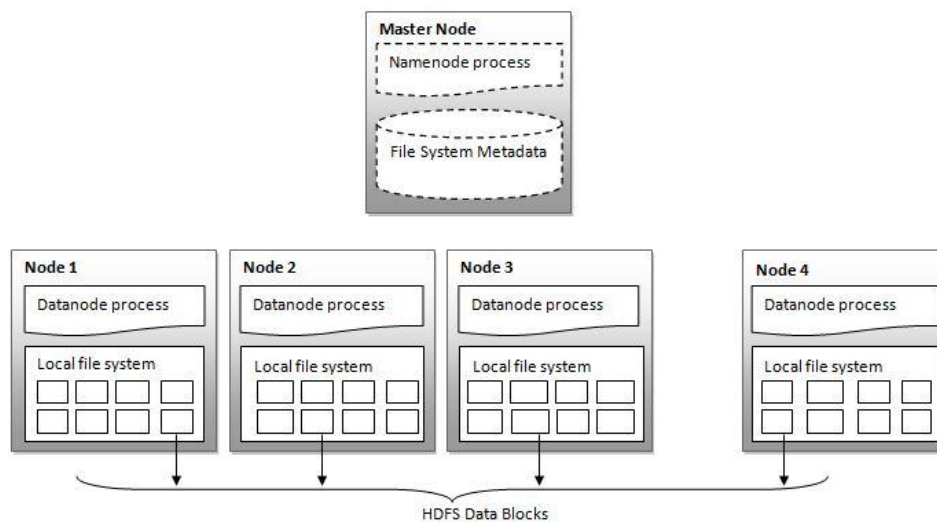
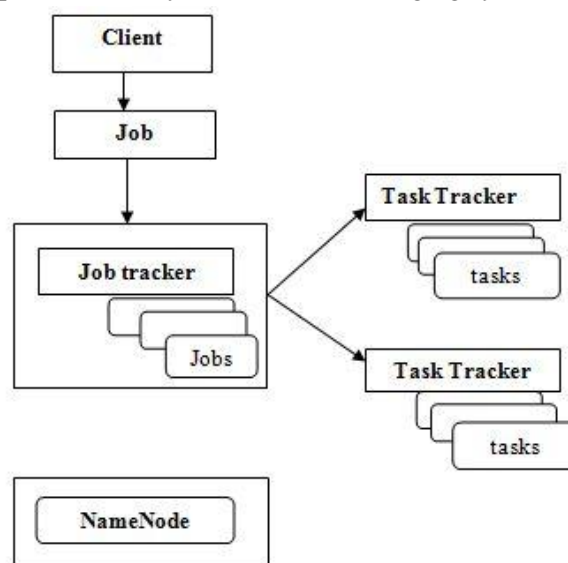
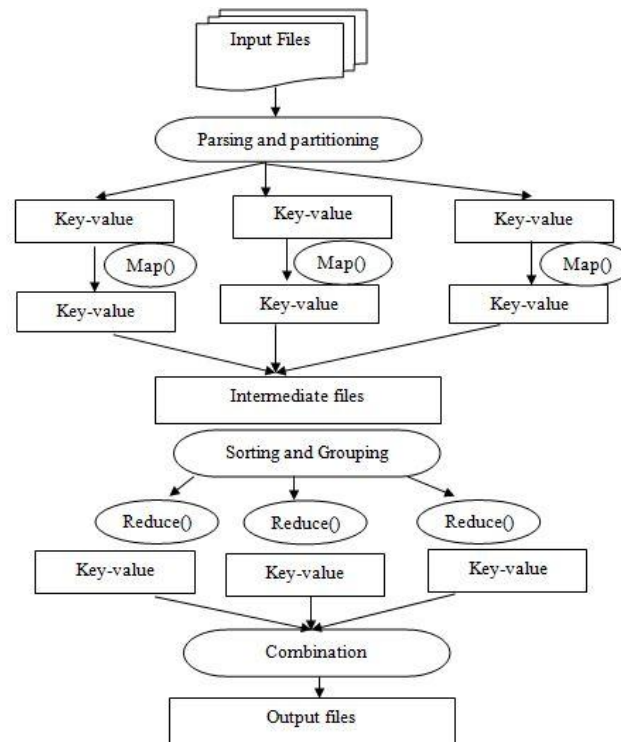


Fig.1: Represents the layout of the file managing system of Hadoop



(a) Representation of HDFS Namenode and JobTracker



(b) operational structure of MapReduce algorithm using Map() and Reduce() functions
Fig.2: (a),(b) work progress by considering the fundamental steps

PROBLEM STATEMENT

Based on the availability of tremendous data on the internet, where images [18] files nowadays are very large in volume. These images are enormous with different sorts of documents and consume large amounts of disk utilization and sometimes poor while downloading. For such image data, the documents are pointed using a term named after compression. The strategy of the compression approach is lossless and sometimes lossy. Additional simplification of the term, disparate contrast images are gathered. All the data that is unique are recovered. The

Lossless and lossy constraints are conditions that illustrate the compression. The archived data is kept uncompressed, every single bit of the lossless compression, data has been recorded initially.

Furthermore, images [19] files have different parts and collections. It also contains duplicate files that are necessarily expelled from the authorized data. The data that has been

encrypted or compressed cannot be further partitioned and must be analyzed as a separate consecutive flow of data. Taking advantage of such files MapReduce's job involves a single mapper to perform and assemble the entire document. Possibly, the performance hit will be expansive. Following are some of the usual image formats for consideration:

- The joint photographic expert grouplabeled as JPG
- Bitmap image labeled as BMP
- Portable Document format labeled as PDF
- Graphic Interchange Format labeled as GIF
- The joint photographic expert group labeled as JPEG
- Tag index File format labeled as TIFF

PROPOSED APPROACH

This proposed approach will accept loads and plenty of images files [20] and used to abolish

replicate files from the feasible appropriate data. Compressed binary format data or encrypted binary format data, in particular, cannot be partitioned but can only be read as the distinguished consecutive flow of data. Practicing such files as input to a MapReduce task (Job), such that the process of single mapper allowing performing till the entire file gets processed, provoking a conceivable large performance potency accomplishment. In such circumstances, it is desirable to allow a disruptable format mostly referred to as SequenceFile, also, there is another process to avert if it cannot acquire the file in the

disparate format and handle step by modifying it into a splittable setup. Therefore, to enforce this approach MD5 algorithm is carried out.

DESCRIPTION ON SEQUENCE FILE

The sequence file consists of the binary format of key/value pairs described as flat files. input/output formats of all these sequence files are largely used in MapReduce. Inwardly rather privately, the sequence file stores the temporary outputs of the MapReduce map functions. figure3 illustrates the Sequence file procedure.

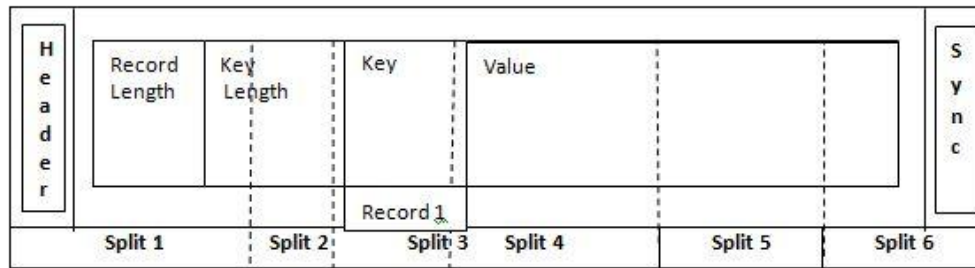


Fig.3: Illustration the Sequence file procedure

IMPLEMENTATION OF SEQUENCE FILES IN HADOOP

Use of image files with a tremendous measure of data, the best idea to process and achieve better results are obtained only through sequence files using hadoop[13]. Hadoop distinguishes the data using two prominent applicational functions such as a map and reduce. Map files consistently can be analyzed. sequence files in particular orderly the data files and allows files to get partitioned by map-reduce. Thus, this will choose the large volumes of files to perform multiple parallel tasks by splitting the work. Appropriate use of sequence files provides Hadoop an attempt to handle its parade. Nevertheless, the chunks of data are adequately worthwhile. Figure 4 represents the work progression of transforming image files with the periods of map-reduce..

configurations and maintains the quality, filename, image substance in HDFS in Bytes Writable format. The processed binary file stores the information at every stage rather than using all the bytes in the file. Reading and writing of binaries use usual ByteWritable. This would perform the process in a significant approach but the sequence of files was still questionable for image files as it is hard to illustrate in a simpler manner. The following are some of the ways that work on image files using sequence files in the notorious approach.

Considering the key/value pairs, the sequence file uses content as key in map file for

Whenever the image files are acquired its most conceptual procedure is to generate a sequence file. The arrangement of image pre-processing is continually allowed to perform sequence files. For this reason, sequence files are joined to this and are associated similarly, wherever the images files got struck and not retrieved properly at the same time. This entire process

and gained through class org.apache.hadoop.io.SequenceFile.Writer.

Some of the files such as tar files, the images are processed by implementing a tool written by Stuart Sierra which improves sequence files. Promoting the jobs to map-reduce by capturing all image files and storing image files as a sequence file. The input is provided as HDFS and output will be achieved through a sequence file. A quick question glides are implemented through psyche before initiating the separation of codes. The sequence file keeps charge of the image file data that has been loaded into the memory in bytes format. There are two particular perspectives to obtain accurate files when the memory has lots of duplicate image files while making a sequence file.

- In the image processing, an image copy finder and single uncomplicated map-reduce jobs allow performing sequence files with a peculiar end goal to function correctly.
- Every image needs to be functioned autonomously before changing the image into a sequence file. The image copy finder and the separation of tasks show a sequence file to identify copies of files and structure of sequence files.

GENERATION OF JOBS IN SEQUENCE FILE THROUGH PRE-PROCESSING

The target of a sequence file is to maintain information about key and vales that holds filename and BytesWritable.

For instance:

hdfs://localhost:8022/customer/elevy/smallArc
hiveImages/WonderWoman.jpg,

therefore, map/lessen task will concentrate exclusively a map in order that determines the read single file on the image and compose it to a sequence file through SequenceFileOuputFormat. This system named FileSystem that accepts oppose as a specificassembled object to access the HDFS file and FSDataInputStream with a unique decision intention to inspect from it. The byte array is an essential design to the society as a bytewritable. Considering the final arrangement of the task is SequenceFileOutputFormat class and this is made to sequence file using the map. This exists in the BinaryFilesToHadoopSequenceFile.java that accomplishes this job ahead.

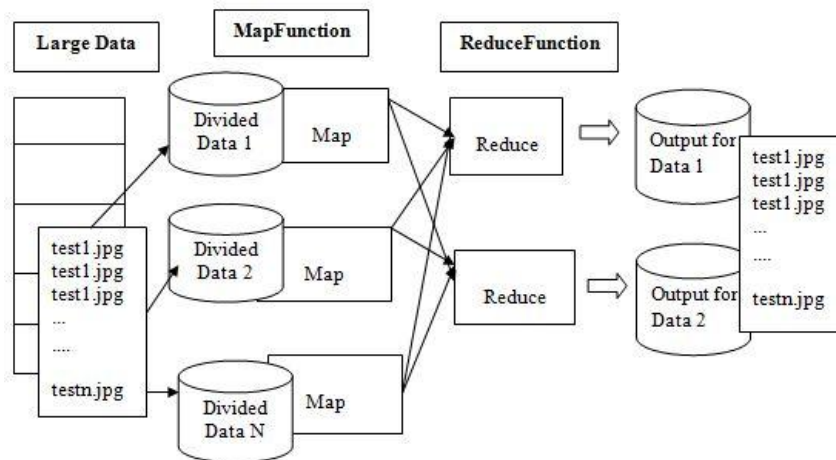


Fig.4: MapReduce algorithm using Map() and Reduce() functions

BINARY FILE

The term binary is defined as 0's and 1's. The file that holds binary digits is referred to as a Binary file which is computer-readable nevertheless not coherent. The impressive text data can also be stored in ASCII format. The possibility to develop binary files using Hadoop demonstrate with checking all the images. The image replication developer maintains the circulation of various limits of smaller files as an improvement for a Hadoop job and present to investigate the binary form of data in a map/reduce task. This is a common acceptable process of binary files in the machine and in algorithms. Hadoop demonstrates the best example through word count program text data to the binary form of output data. Hadoop is an excellent framework that allows massive files although the propagation of the processing on the minor consultant. This will provide optimized solutions through processing and contribute effective results.

Officially, providing services to the tremendous data, HDFS is supported. It reasonably overheads the huge data on all image files to achieve focal points like partitions of data into blocks on different node

machines and also provide a substitute for every individual file with a less replicates. Based on the processing time, this can be affected too. As per the frequent change in the data, file is consigned to its responsible map operation. In this way, massive large files equip an extensive supply of map tasks. Maintaining the map tasks, applications can run without any overload by allowing heaps. With a reasonable justice, MultiFileInputSplit, are used instead of reusing the JVM machines.

PROCEDURE FOR MD5 ALGORITHM

MD5(message digest)an innovative method was introduced by Professor L. Rivest in 1991.With the agreement of RFC 1321, “the algorithm accepts the input information has irrational length and delivers result as 128-bit “message digest” for the given input.

This algorithm is usually an applicational aspect of digital signatures. In the case of large documents are protected and compressed through encryption algorithms such as RSA. Allowing two keys for privacy. One is the public key and the other is the private key, using the cryptosystem. The following Figure 5 represents the structure of the MD5 algorithm.

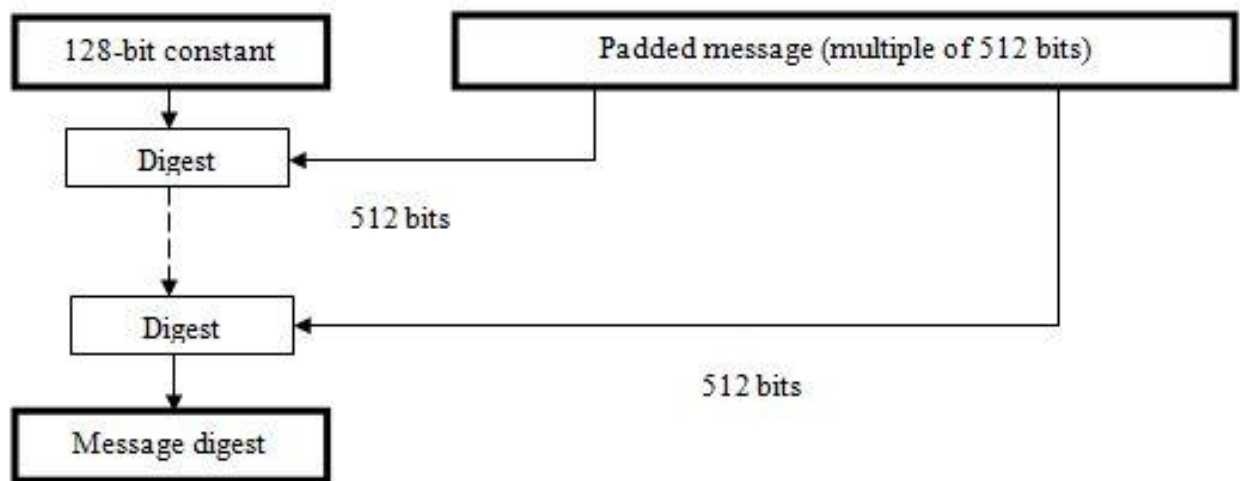


Fig.5: represents the structure of the MD5 algorithm.

APPLICATIONAL EXECUTION OF STEPS

Computing the padding bits: The input information is “padded” (developed) so that its dimension (in bits) corresponds to $448 \bmod 512$. The process of Padding is steadily operated, regardless although the length of the message is $448 \bmod 512$. It is accomplished by an individually “1” bit is connected to the padding message that obtains to be balanced to $448 \bmod 512$. partly 512 bits are added with a condition not less than one bit

Accumulation of length: the length of the message was defined by 64-bit, added to the

output of the initial step and it is not more significant than 2^{64} . Due to the insignificant-form of 64 bits are applied. The outcome of the message (ensuing to stuffing with bits, b) acquire a length that has a straight forward diversified multiple of 512 bits and it is given by 16 (32-bit) words of information message.

Input MD buffer

Evaluate the message digest by considering four-word buffer (P,Q,R,S). Here, every P,Q,R,S word is saved under the 32-bit register. All the mentioned registers are promoted to the succeeding conditions in hexadecimal, insignificant-form bytes initially:

Word	P	01	23	45	67
Word	Q	89	Ab	Cd	Ef
Word	R	Fe	Dc	Ba	98
Word	S	76	54	32	10

Generate the entire message in 16-word blocks, all the considered 4 functions are indicated as three input and executes one word as output (all four consists of 32-bit).

$$F(M,N,O) = MN \text{ or not } (M) O$$

$$G(M,N,O) = MO \text{ or } N \text{ not } (O)$$

$$H(M,N,O) = M \text{ xor } N \text{ xor } O$$

$$I(M,N,O) = N \text{ xor } (M \text{ OR not } (O))$$

Effectiveness of the selected algorithms

Approaches	Size of the key(bits)	Assumed speed (Kbytes/sec.)	PRB Optimized (Kbytes/sec.)
TEA algorithm	128	700	-
DES algorithm	56	350	7746
Triple DES algorithm	112	120	2842
IDEA algorithm	128	700	4469
RSA algorithm	512	7	-
SHA algorithm	160	750	25162
MD5 algorithm	128	1740	62425

PERFORMED WORK

Apache foundation layed an open-source platform for researchers, who found great advantage in using Hadoop. Appealing to the MapReduce designs, regardless of Hadoop, we can additionally operate on the twister framework[14], phoenix framework [15] or more Mapreduce technical structure [16].

Considering two major structures that belong to MapReduce design for particular principles. Twister is applied to figure out iteratively and phoenix is implemented to perform multi-processor frameworks over shared memory. These are suitable for fitting decisions in Hadoop. Yet, the overall concept of this paper

lies behind Hadoop, it's high subsistence and its preference forexistence in multi-purpose.

PROPOSED FRAMEWORK

HDFS is used to store image files and organize them accordingly by eliminating the text. The input data is been partitioned into blocks and circulated over the Hadoop cluster. All the binary files are functioned using read/write

operations in Hadoop. Subsequently, the unspecified data that can be converted were placed into HDFS. This allows the flexibility that is captivating for magnificent data arrangement. Allowing videos from CCTV's from social occasions were taken and placed into HDFS storage. Following is the flowchart diagram and its steps accomplished outcomes for the considered expression:

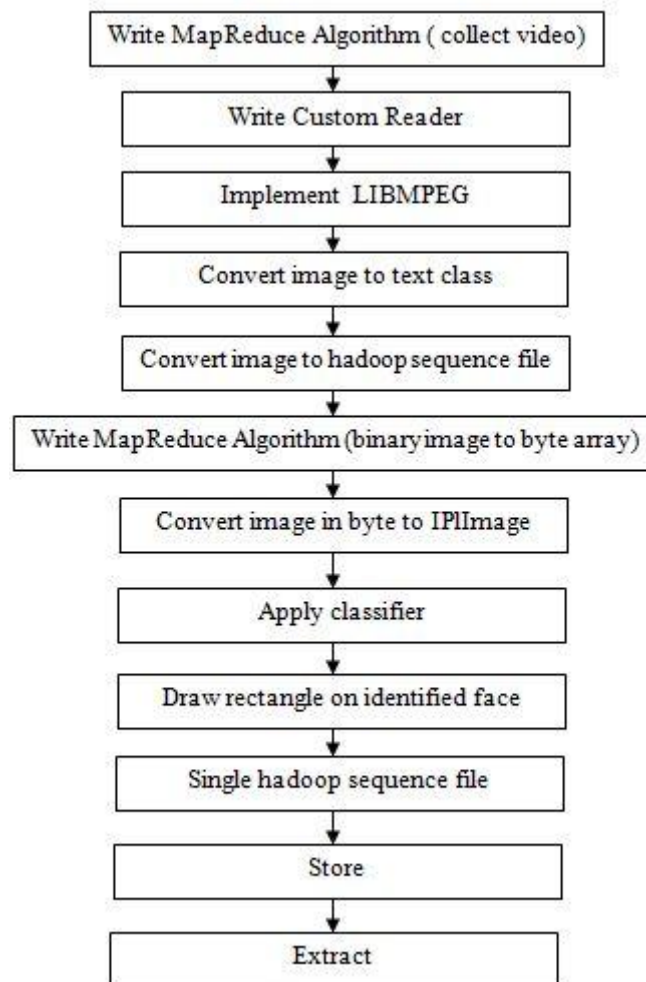


Fig.6: Flowchart for the proposed work

1. Gather the video sequence level data that is ready to use and write the vast chunk data into the map-reduce algorithm and accumulate it as a reconciled file in HDFS.
2. To read data, write custom recorders, then pictures from a particular group are verified by the previously occurred output file of the video file.
3. To concentrate image frames, enable LIBMPEG device in ritual RecordReader
4. Conversion of text class from binary data to writable.
5. Transform images into hadoop sequence file.
6. Read binary image from the sequence file into a byte array using the

MapReduce algorithm which performs map function and reduces function.

7. Transform image in Byte manifestation to IPIImage (Image object in OpenCV)
8. Implement classification on the image using the classifier.
9. Choose the identified face by drawing a rectangle
10. Accumulate the identified images into one Hadoop sequence file.
11. Now save all the data in the sequence file to HDFS storage database
12. Derive recognized images from the sequence file.

CONCLUSION

The recent existence of an extensive quantity of perceptible data and its necessity for conversant and constraining transformation on restored management of allocated shared image processing schemes in image processing operations. In the coming era of technology, multiple different algorithms were designed to display the importance of the image processing field and realization of the large scale necessities in image processing fix to specific objectives of the outside environment. In this paper, the analysis of distributed processing functional operations and the programming modes are mentioned. Besides a set of works are examined with the Hadoop system as it is open-source. Its existence was lately identified but the processing models got advanced with better efficiency and also got some controversies. The occurrence, the IT society with its nonappearance of quality of lion's action, nonattendance of adequate expertise activity forces, and undesirable pitfall and effects on the report of its trait. Nevertheless, this disposing of data modes that manipulates MapReduce design and further distributed images surrounded by most powerful mechanical associations for image processing and set a consistency pattern to acknowledge distributed computing structures in the coming years

REFERENCES

1. Bakshi, K., "considerations for big data: Architecture and approach and aerospace conference- Big Sky", MT, 3-10 March 2012.
2. Miller, K.W, and Michael, "Big Data: New opportunities and New Challenges", Journal of IEEE Computer Society, 46, 22-24.
3. B., Tom, White, Y., Jimmy, L. And Davis, LS, "Web-Scale Computer Vision using MapReduce for Multimedia Data Mining " Proceedings of the 10th International workshop on multimedia data mining, Washington DC, 25-28 July 2010, 1-10.
4. Ghemawat, S, and Dean. J, "MapReduce: Simplified Data processing on Large Clusters", Communications of the ACM, 51, 107-114.
5. The New York Times Blog (2007), <http://open.blogs.nytimes.com/2007/11/01/self-service-prorated-super-computing-fun/>
6. Kalagiakos, P. (2011), "Cloud Computing Learning" Application of information and communication technologies (AICT), 5th international Conference, 12-14 October 2011.
7. C., Liu, L., Arietta, S., Lawrence, J Sweeney, HIPI: A Hadoop image processing interface for image-based MapReduce tasks. B.S. thesis, University of Virginia (2011).
8. Hadoop: <http://hadoop.apache.org/>
9. http://en.wikipedia.org/wiki/Apache_Hadoop
10. HDFS, <http://hadoop.apache.org/hdfs/>
11. MapReduce, <http://en.wikipedia.org/wiki/MapReduce>
12. Bhandarkar, M. (2010), MapReduce programming with Apache Hadoop, parallel & distributed processing (IPDPS) IEEE, 19-23 April 2010.
13. Dr. E. Laxmi Lydia, K. Vijaya Kumar, P. Amaranatha Reddy, D. Ramya, "Text Mining with Hadoop: Document Clustering with TF-IDF and Measuring Distance Using Euclidean", Journal of advanced research in

- dynamical & control systems, Vol.10,14-Special Issue, 2018.
14. Dr. E. Laxmi Lydia, B. Prasanna Kumar, D. Ramya, "Generation of dynamic energy management using data mining techniques basing on big data analytics issues in smart grids", International Journal of Engineering & Technology, 7(2.26), 2018, 85-89.
 15. E. Laxmi Lydia, P. Govindaswamy, SK. Lakshmanprabu, D. Ramya, "Document Clustering based on Text mining K-means algorithm using Euclidean distance similarity", Journal of advanced research in dynamical & control systems, vol. 10, 02-special issue, 2018.
 16. E. Laxmi Lydia, D. Ramya, "Text Mining with Lucene and Hadoop: Document clustering with updated rules of NMF Non-Negative Matrix Factorization", International Journal of Pure and Applied Mathematics, vol 118(7),191-198, 2018.
 17. Rajesh Dwivedi, Rahul Kumar, Ebenezer Jangam, "Dimensionality Reduction Technique on SIFT Feature vector for content-based image", Communications in computer and information, ScienceSpringer, Vol. 1035, Jul-19, ISSN:1865-0929.
 18. Tulasi Krishna Saija, Hemantha Kumar kalluri, "Gender Classification based on face images of the local binary pattern using support vector machine and backpropagation neural advances in modeling and analysis" Vol.20, No.1, pp.31-35, Aug-19, ISSN 1240-4543.
 19. Saija Tulasi Krishna, Hemantha Kumar Kalluri, "Deep Learning and transfer learning approaches for image classification", International Journal of Recent Technology and Engineering, Vol.7(5S4), Jun-19, ISSN:2277-3878.
 20. G. Keerthi, "Text and image classification using conventional machine learning to convolutional neural networks", Journal of advanced research in the dynamical and control system, Vol. 11 Issue 7, Nov-19.
 21. Elhoseny, M., Bian, G. B., Lakshmanprabu, S. K., Shankar, K., Singh, A. K., & Wu, W. (2019). Effective features to classify ovarian cancer data in internet of medical things. *Computer Networks*, 159, 147-156.
 22. Shankar, K., Elhoseny, M., Perumal, E., Ilayaraja, M., & Kumar, K. S. (2019). An Efficient Image Encryption Scheme Based on Signcryption Technique with Adaptive Elephant Herding Optimization. In *Cybersecurity and Secure Information Systems* (pp. 31-42). Springer, Cham.
 23. Elhoseny, M., & Shankar, K. (2020). Energy efficient optimal routing for communication in VANETs via clustering model. In *Emerging Technologies for Connected Internet of Vehicles and Intelligent Transportation System Networks* (pp. 1-14). Springer, Cham.
 24. Elhoseny, M., Shankar, K., & Uthayakumar, J. Intelligent Diagnostic Prediction and Classification System for Chronic Kidney Disease, *Nature Scientific Reports*, July 2019. Press. DOI: <https://doi.org/10.1038/s41598-019-46074-2>.
 25. Dutta, A. K., Elhoseny, M., Dahiya, V., & Shankar, K. (2019). An efficient hierarchical clustering protocol for multihop Internet of vehicles communication. *Transactions on Emerging Telecommunications Technologies*.
 26. Elhoseny, M., & Shankar, K. (2019). Optimal bilateral filter and Convolutional Neural Network based denoising method of medical image measurements. *Measurement*, 143, 125-135.
 27. Murugan, B. S., Elhoseny, M., Shankar, K., & Uthayakumar, J. (2019). Region-based scalable smart system for anomaly detection in pedestrian walkways. *Computers & Electrical Engineering*, 75, 146-160.
 28. Famila, S., Jawahar, A., Sariga, A., & Shankar, K. (2019). Improved artificial bee colony optimization based clustering algorithm for SMART sensor environments. *Peer-*

- to-Peer Networking and Applications, 1-9.
29. Lakshmanprabu, S. K., Shankar, K., Rani, S. S., Abdulhay, E., Arunkumar, N., Ramirez, G., & Uthayakumar, J. (2019). An effect of big data technology with ant colony optimization based routing in vehicular ad hoc networks: Towards smart cities. *Journal of cleaner production*, 217, 584-593.
30. Maheswari, P. U., Manickam, P., Kumar, K. S., Maselena, A., & Shankar, K. Bat optimization algorithm with fuzzy based PIT sharing (BF-PIT) algorithm for Named Data Networking (NDN). *Journal of Intelligent & Fuzzy Systems*, (Preprint), 1-8.
31. Shankar, K., Ilayaraja, M., & Kumar, K. S. (2018). Technological Solutions for Health Care Protection and Services Through Internet Of Things (IoT). *International Journal of Pure and Applied Mathematics*, 118(7), 277-283.
32. Lakshmanprabu, S. K., Shankar, K., Ilayaraja, M., Nasir, A. W., Vijayakumar, V., & Chilamkurti, N. (2019). Random forest for big data classification in the internet of things using optimal features. *International Journal of Machine Learning and Cybernetics*, 1-10.
33. Sankhwar, S., Gupta, D., Ramya, K. C., Rani, S. S., Shankar, K., & Lakshmanprabu, S. K. (2016). Improved grey wolf optimization-based feature subset selection with fuzzy neural classifier for financial crisis prediction. *Soft Computing*, 1-10.
34. Iswanto, I., Lydia, E. L., Shankar, K., Nguyen, P. T., Hashim, W., & Maselena, A. (2019). Identifying diseases and diagnosis using machine learning. *International Journal of Engineering and Advanced Technology*, 8(6 Special Issue 2), 978-981.
35. Lakshmanprabu, S. K., Mohanty, S. N., Krishnamoorthy, S., Uthayakumar, J., & Shankar, K. (2019). Online clinical decision support system using optimal deep neural networks. *Applied Soft Computing*, 81, 105487.
36. Shankar, K., Lakshmanprabu, S. K., Khanna, A., Tanwar, S., Rodrigues, J. J., & Roy, N. R. (2019). Alzheimer detection using Group Grey Wolf Optimization based features with convolutional classifier. *Computers & Electrical Engineering*, 77, 230-243.