# Implementation of Image Processing System using Handover Technique with Map Reduce Based on Big Data in the Cloud Environment

Mehraj Ali and John Kumar
Department of Computer Application, Thiagarajar College of Engineering, India

**Abstract**: *Cloud computing is the one of the emerging techniques to process the big data. Cloud computing is also, known as service on demand. Large set or large volume of data is known as big data. Processing big data (MRI images and DICOM images) normally takes more time. Hard tasks such as handling big data can be solved by using the concepts of hadoop. Enhancing the hadoop concept will help the user to process the large set of images. The Hadoop Distributed File System (HDFS) and Map Reduce are the two default main functions which is used to enhance hadoop. HDFS is a hadoop file storing system, which is used for storing and retrieving the data. Map Reduce is the combination of two functions namely maps and reduces. Map is the process of splitting the inputs and reduce is the process of integrating the output of map's input. Recently, medical experts experienced problems like machine failure and fault tolerance while processing the result for the scanned data. A unique optimized time scheduling algorithm, called Dynamic Handover Reduce Function (DHRF) algorithm is introduced in the reduce function. Enhancement of hadoop and cloud and introduction of DHRF helps to overcome the processing risks, to get optimized result with less waiting time and reduction in error percentage of the output image.*

**Keywords**: *Cloud computing, big data, HDFS, map reduce, DHRF algorithm.*

*Received September 13, 2013; accepted March 20, 2014; published online April 1, 2015*

## 1. Introduction

Cloud computing is the sought after field nowadays in information technology. Cloud computing is a package comprising of server and client machines. Cloud computing processes the data in the distributed and parallel modes. Cloud computing is also, known as service on demand. The services of the cloud computing enables end users to pay and obtain required data from the service providers like IBM, Amazon and Intel among others. In this proposed work, an enhanced cloud tool called Intel (a product of Intel) is utilised. Enhancing the concept of hadoop over the cloud computing gives the better result in the process of computing big data. The hadoop enhances the Hadoop Distributed File System (HDFS) and map reduce functions in it. The map reduce concept will execute the complicated tasks very easily, with simple requirements of machines. Google first introduced the concept of map reduce programming model [4, 11]. map reduce concept has few basic functions like master, slave, job manager, job node, etc. The master functions supervise the execution of map and reduce operations.

The image processing techniques like grayscale, sobel edge detection, gaussian blur and fast corner_9 detection are also, enhanced in the proposed work. Presently, this regular set of work is made with the other corner detection method and scheduling algorithm for 2D to 3D data processing [9, 11]. In the proposed work, it has been proved that, there is another better corner method, improved Sum of Absolute Differences (SAD) matching and an optimized scheduling algorithm, which could benefit the client in the useful manner. Dynamic Handover Reduce Function (DHRF) algorithm has proved that, it works better than the existing algorithm in the reduce function. JPEG files can be viewed and opened mostly by many image viewers. Some image formats may get deleted while compressing. Even some reduction in the quality of image may occur while compressing. In the proposed model, a template has been made in such a way that, it accepts the input data in any format. An attempt has been made to show that all the accepted input is to be compressed to that of the fixed frame size. The raw data formats are converted to the fixed frame size and then the data compression is done to a fixed scale. The output will be a better one with high flexibility, less waiting time and less error percentage. Mostly the medical data will be in the DICOM format and rarely in the .JPEG format. As an outcome of this work the output templates received will be in the .JPEG format.

Implementation of fast corner_9 method has proved that it can give the users a better result than harries corner method. Also, the implementation of improved SAD reduces the error percentage when compared to the existing method. The computation of small files is proved to be better in the existing system. The concept of big-data is being used in the proposed work, so that it could manage the input into fixed frame size. Patel *et* *al.* [6] reports the experimental work on big data problem and its optimal solution using hadoop cluster, HDFS for storage and using parallel processing to process large data sets using map reduce programming template. In the reported work, the word junks meaning intermediate data will be sent to the reduce function. In the reduce function, the proposed DHRF algorithm is added to give the result. Big data chunks with different size and sequence is computed in each node, so that transfer of a chunk is overlapped with the computation of the previous chunk in the node, as much as possible [1, 2, 3]. These junks are computed in the reduce functions. The data transfer delay can be comparable or even higher than the time required to compute the data [7, 8, 10]. To overcome the problem of transfer delay from the existing system, a novel optimization scheduling algorithm has been implemented in our study. In the existing module, processing large data is by a large number of small files, which exhibits better performance [5]. In this work, handling of large set of data has been implemented with the enhanced tool.

Figure 1 shows format by which the inputs in the various formats are stored in the data container. The job manager function is use to assign work for the servers. These inputs of the data will be stored in HDFS to start the map reduce function. Each job will be assigned as a task.

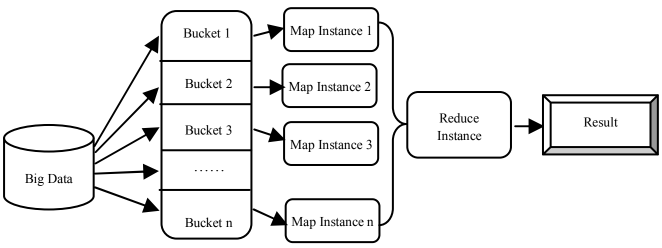



Figure 1. Architecture of hadoop for map reduce function.

The hadoop's characteristic is to split the data and distribute to the hosts to compute. This work will be done simultaneously in the parallel mode. This is known as distributed cum parallel computing. Figure 2 shows the modules in the map reduce function. The task depends upon the strength and the storage of the computing system.

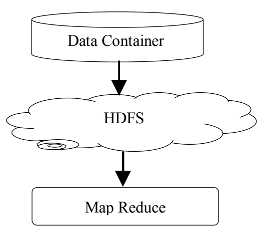



Figure 2. Modules in map reduce function.

## 2. Data Processing Procedures

In this work, the process comprises of few data processing techniques as shown in Figure 3. Such as: Gray scale, sobel, Gaussian blur, fast corner and SAD matching to find the difference between two data. The gray scale method is used because, when an image is converted to gray scale, the image's quality will be improved. Sobel method is used to find the edges of the images. Gaussian blur is used to blur the image, so that it will be useful for fast corner_9 method to detect the corners of the image. The enhancement of these methods results in better quality of the image. Hence, these are surveyed as the best. The given equation finds the solution for the harris corner method to find the image patch area with the argument ($u$, $v$). Where ($u$, $v$) denotes the image patch point and while processing, ($x$, $y$) get shifted to ($u$, $v$), where $w$ is the center point on the ($x$, $y$).

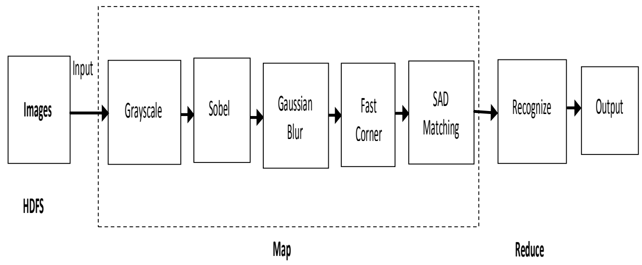



Figure 3. Data processing technique.

$$E(u,v)=\sum_{x,y} w(x,y)[I(x+u,y+v)-I(x,y)]^{2} \qquad (1)$$

In harris corner, a square mask with point $p(x, y)$ was established. Figure 4 shows the output of harris corner method.

When the mask of greyscale value is higher than the threshold, the point $P$ is defined as the corner. Here, $E$ denotes the patch area, $I$ denote the given image.

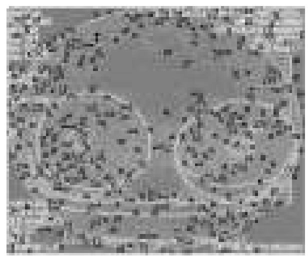


Figure 4. Harris corner method.

# 3. Methodology

## 3.1. Fast Corner_9

In the presence of various corner detection methods include harris corner method, Susan, Zheng and Harr. Among these methods, fast corner_9 is used due to higher clarity when compared with other corner detection method. This fast corner detection has come from moravec Sum of Squared Differences (SSD) and the harris corner detection (second derivative of SSD). By using the non minimal suppression it is determined. This is calculated by subtracting the original value, by comparing $p$ value and with $t$ value. This has been implemented on client machine with the configuration Intel (R) Core 2duo, 4GB RAM and 2.93GHz processor and the Server with the configuration of IBM X 3400 M3 Server, Intel Xeon E 5507 (Quad

Core), 2.26GHz Processor, 146GB DDR3-1333MHz ECC RAM, 146GB 10K SAS hard disk drive.

$$v = min \begin{cases} \sum (p - pixelvalue)n, if \ \frac{1}{n}(value - p) < t \\ \sum (pixelvalue - p)n, if \ \frac{1}{n}(p - value) > t \end{cases} \quad (2)$$

Where $v$ is the corner value, $p$ denotes greyscale value, $t$ denotes the Threshold value and $n$ denotes to the non-maximum suppression.

### 3.2. Sum of Absolute Differences

SAD is a mathematical term that has been enhanced here to find between differences the two $P$ blocks of the processed data.

$$SAD = \sum (u, v) \in w \ [A_1(u, v) - A_2(x+u, y+v)] \quad (3)$$

Where $A_1$ and $A_2$ are the two images. Like the harris corner method, here also, ($u$, $v$) denotes the image patch area and while processing ($x$, $y$) gets shifted from ($u$, $v$). Where, $w$ refers to the fraction of image points. Before the implementation of SAD there is an existence of SSD to find the difference. As SSD is an old, it isn't able to produce a clear result like SAD. The Relationship between the intersection points detected by harris corner detection method and labels the corresponding points for computers to judge the corresponding locations of intersection. In relationship between the corners, the interpolations are used to judge the location of the corner while being photographed to simulate the data.

## 4. Dynamic Handover Reduce Function Algorithm

Figure 5 shows the Illustration of DHRF algorithm. After the installation of Intel Manager is over, next the hadoop set up has to be done in the system. Now, the system can work on the map reduce functions and use the facility of the HDFS. After the successful installation of the Intel Manager and hadoop and its content, the proposed DHRF algorithm has to be inserted in the reduce function. Since, the map reduce is an open source, it can be edited and modified according to the user's need. When a task is applied to the nodes on the cluster, the map function starts its job of splitting the data. The task node assigns the job for the each node, and also it supervises the job node and its functions. When the job assigned by the task node gets over, the output of the map function is ready with the intermediate data. In the proposed work DHRF algorithm receives the map result and finally, recognizes the labeling for the output. In this map, as defined earlier the four methods of processing have been involved. The master node monitors all the functions of the map function. The task of the map function is the hardest task, which manages the server and client machines.

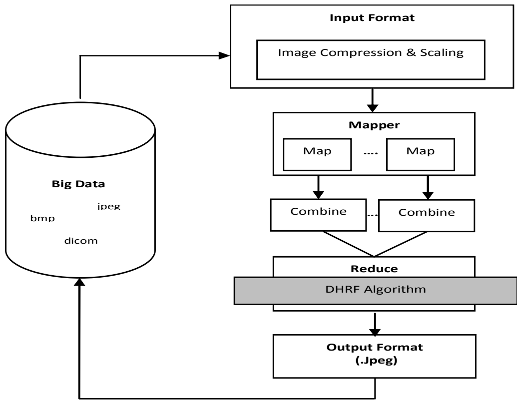



Figure 5. Illustration of DHRF Algorithm.

*Algorithm* 1: DHRF.

```
BufferedImage res= new
BufferedImage(width,height,BufferedImage.TYPE_BYTE_GRA
Y);
 // Initialize the image process
byte[] bytesCompressed=
compressor.compress(imagetoCompress);
Deflater deflater= new Deflater();
deflater.setInput(bytesToCompress);
// Produce the data compression
BufferedImage resizedImage= new
BufferedImage(IMG_WIDTH, IMG_HEIGHT, type);
Graphics2D g= resizedImage.createGraphics();
g.drawImage(originalImage, 0, 0, IMG_WIDTH,
IMG_HEIGHT, null);
g.dispose();
// put the data into scaling
static{
        URL.setURLStreamHandlerFactory(new
        FsUrlStreamHandlerFactory());
       }
// write the map reduce structure
 in= new URL(PATHTOBEMAPPED).openStream();
IOUtils.copyBytes(in, System.out, 2, false);
// set the server to handle mapper
FSDataOutputStream out= fileSystem.create(path);
InputStream in= new BufferedInputStream(new
FileInputStream(new File(source)));
// mark data into HDFS of hadoop
Process the image until completing the grayscale,sobel,
guassian, fast corner, SAD matching of the image.
// operate the data process until the data processed map(in_key,
in_val) -> list(out_key, intermediate_val)
reduce(out_key, list(intermediate_val)) -> list(out_value)
// Set the map reduce Operation
FileSystem fs= file.getFileSystem(context.getConfiguration());
FSDataOutputStream fileOut = fs.create(new
Path("your_hdfs_filename"));
// write the data mapper
reduce(WritableComparable, Iterator, OutputCollector,
Reporter)
continue until reducer task is complete
  // send mapper output data to reducer
JobConf.setNumReduceTasks(int)
 // set small unit value to the task and reducer wait queue
 interrupt.task
store the result(image).
```

The map reduces function handles the data to produce the desired output as shown in Figure 4. Finally, map results are sent to reduce operation. The implementation of DHRF algorithm focuses on reduces function integrating the task and allots the process to produce the result in .jpeg format.

## 5. Proposed Work

To obtain an optimized result from the existing image processing techniques, the proposed work implements hadoop and cloud computing using INTEL Manager. A set up of ten machines configured with Intel (R) Core 2 duo, 4GB RAM and 2.93GHz processor is used in this study. In these machines, INTEL Manager a Cloud tool is installed. INTEL Manager has basically Master and Slave in it. Since, hadoop is an open source; it can be modified according to our need. So, the editing is done in the reduce function, by adding the proposed DHRF algorithm. In this work, an enhanced cloud tool called Intel Manager is utilized. The coding or the application of image processing techniques is installed on the INTEL Manager, to run the proposed experiment. The main advantage of this enhanced Intel Manager is it works on the Normal Window XP (64bit) infrastructure. In this work Intel Manager which is open source software available as private cloud cum hybrid cloud is used.

### 5.1. Performance Analysis and Evaluation

Previously, an analysis was done using a different algorithm with four servers. Analysis shows, the time consistency by designing an optimized scheduling algorithm. The proposed DHRF algorithm reduces the time and error percentage using the reduce function. The DHRF algorithm has to be coded with java and the input has to be given in jar file format. When the start option is selected, the operation gets started. The same operation can be paused and stopped. There will be a screen with two segmentations. First, segment shows the given input which is to be processed. Second, one shows the output. Once the operation gets over, it automatically shows the output. Figure 6 explains the process.

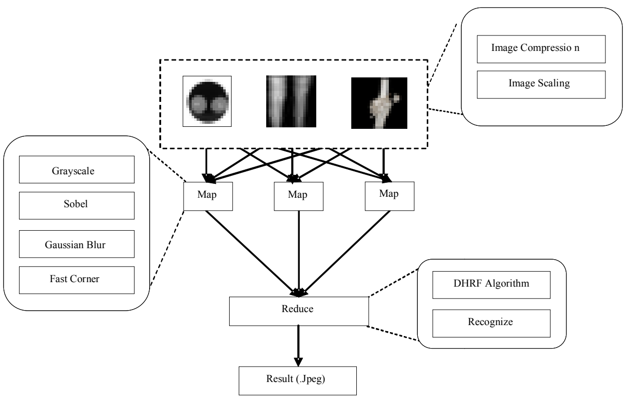



Figure 6. Work flow and sequence of DHRF algorithm.

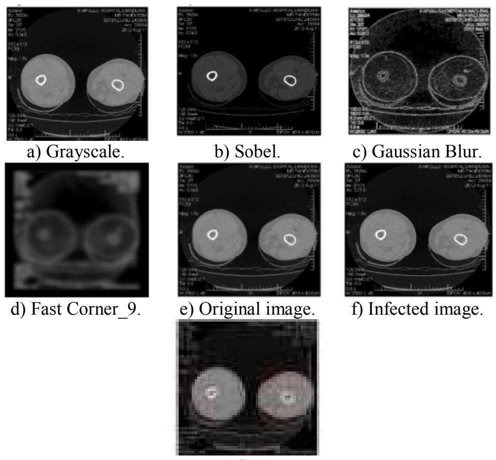
The task node, automatically selects the server to do the map function. Then, the output of the map function will be taken as the input for reducer function. The reducer function is to integrate the input of map before the map function. The implementation of DHRF algorithm will work on the reduce function and will perform the scheduling process. That is to reduce the waiting time by comparing DHRF algorithm and existing DSRF algorithm. The DHRF algorithm is designed mainly to reduce the time and to decrease the waiting time.

## 6. Implementation

### 6.1. Structure

The map function structure shows the flow of process that takes place while processing the map function. The original input is shown in Figure 7-a. The process of converting the original image to Grayscale is shown in Figure 7-b, which means the original image, is converted into Black and White format. The conversion of Grayscale to Sobel edge detection is shown in Figure 7-c, which reduces the edge's noise in the image.

Finally, the conversion of Sobel edge detection to Gaussian Blur is shown in Figure 7-d, which blurs the image for further processing. The differences between the original images Figure 7-e and the infected image Figure 7-f are found by using SAD matching. The result of SAD matching is shown in Figure 7-g.



a) Grayscale. b) Sobel. c) Gaussian Blur.
d) Fast Corner_9. e) Original image. f) Infected image.
g) SAD.

Figure 7. Results (image) obtained from data processing techniques.

### 6.2. Implementation Environment

This session clearly explains the task of hadoop map reduce function. First, the cloud tool INTEL Manager is installed on the machines. Basically, hadoop map reduce runs in Ubuntu OS. INTEL Manager is the private and hybrid cloud, which easily interacts with the Amazon EC2. Here, an attempt is made to run in Windows Operating System (64bit). Windows operating is a comfortable environment and user friendly than Ubuntu to run and process the INTEL Manager. This application runs in Windows Operating System to process the data for map reduce function. The concept of processing the software/application in an infra structure is called as the Infra structure As A Service (IAAS). This helps in executing the data processing technique. Windows Operating System is the platform which supports the application called Intel Manager on which, the deployment of map reduces function works. The process has the flexibility with a start and stop option on this platform. The users are allowed to select the services which they need to start with. The coding should be inserted as a jar file to start the process.

## 7. Result and Discussions

The DHRF algorithm has proved that, it reduces the time complexity while processing the reduce function. The proposed method produces the better result, when the result of proposed SAD matching is compared with the existing SAD matching.

The main advantage of the fast corner_9 method during the execution was, the output of the data is marked with better quality to detect the corners of the image. The output of fast corner_9 method is shown in Figure 8. While comparing with existing, the proposed fast corner_9 method shows the corner more clearly. During the comparison, it is proved that, the method applied in the proposed algorithm works better than the existing system. Whatever the size or format of images may be, the result will be produced in the pre defined format. The result occupies less memory space when compared with the size of the input for storage.

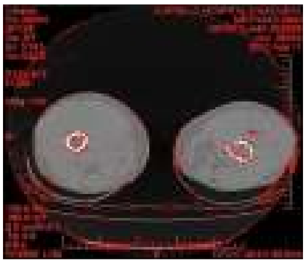


Figure 8. Fast corner_9 method.

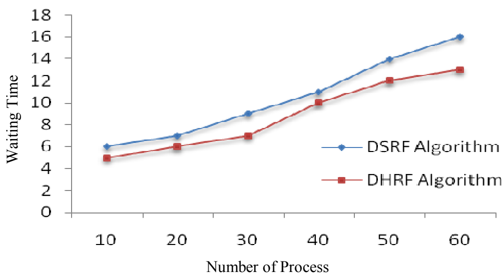



Figure 9. Waiting time of DSRF versus DHRF algorithm.

Figure 9 above shows the graphical representation between the proposed DHRF algorithm and existing DSRF algorithm, which clearly shows the number of users attempting the process and the time (waiting time) taken for the process to complete. From the graph we can clearly understand the proposed method is far better than the existing algorithm. The line for proposed DHRF algorithm falls below the line of DSRF algorithm in the Figure 10 shows the reduction in waiting time which is the main objective in the proposed work.

Figure 10 shows the graphical representation between the percentage of repeatability and corner of frames. When harris corner method is compared with the fast corner method, the proposed fast corner method has high deviation point. This comparison is executed with standard system configuration for both the proposed works and existing methods. This graph is made to show the number of attempts repeated to match the Corners of the frame.

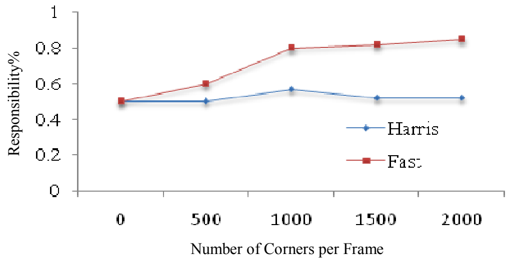



Figure 10. Number of Frames for harris versus fast method.

Figure 11 shows the graphical representation between the percentage repeatability and the Noise standard deviation between the existing harris corner detection method and proposed faster corner edge detection method. Since, the deviations in the Graphs positively shows that, the proposed method is far better and improved when compared with the existing work. As defined the input images are compressed and scaled. Figure 12 shows the graphical representation of SAD difference existing SAD with proposed SAD. This shows the proposed SAD matching is proved to be better than existing system.

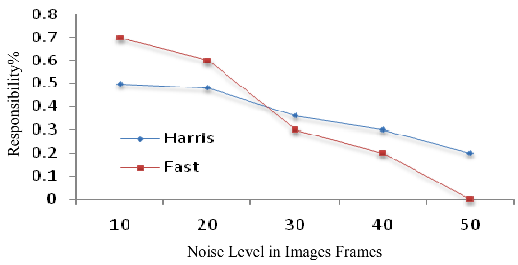



Figure 11. Noise level observed from harris versus fast method.

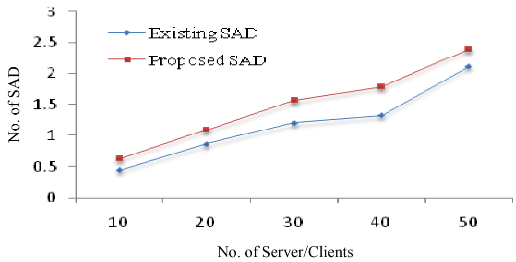



Figure 12. SAD difference between existing versus proposed method.

All the four graphs, shows the newly implemented algorithm and the enhanced techniques are better when compared with the existing technique. The minute variations in the graph matters and proved to be much better when compared with existing. The servers and the clients are shown in the graph by incorporating the readings according to the process, techniques and algorithm respectively. The proposed harris corner method and the SAD matching play a major role for the enhancement of the output image in this work. The graphs are valued in percentage, so that the results are obtained much accurately.

## 8. Conclusions

This work proves the images of various formats can be taken as input. The quality of the image is fine tuned with the proposed algorithm which has produced better result in the .jpeg format. This result shows that, whatever may be the format of the input, the result can be obtained in .jpeg format to give better improved quality of output with less waiting time and error percentage. In this work, the present algorithm has been implemented for the optimizing the result in the reduce function. In future, it is planned to incorporate some more modification on the map function, so that the results can be more accurate. This work implements four image processing techniques, where as in the future work, the comparison testing can be done by using less number of image processing techniques.

## References

[1] Gao K., Wang Q., and Xi L., "Reduct Algorithm based Execution Times Prediction in Knowledge Discovery Cloud Computing Environment," *the International Arab Journal of Information Technology*, vol .11, no. 3, pp. 268-275, 2013.

[2] Hao Y., "Fast Corner Detection-machine Learning for High Speed Corner Detection," available at: http://www.docin.com.cn/p-1429412129.html, last visited 2010.

[3] Jung G., Gnanasambandam N., Mukherjee T., "Synchronous Parallel Processing of Big-data Analytics Services to Optimize Performance in Federated Clouds, Cloud Computing," *in Proceedings of the 5th International Conference on Digital Object Identifier*, Honolulu, pp. 811-818, 2012.

[4] Kalagiakos P. and Karampelas P., "Cloud Computing Learning," *in Proceedings of the 5th International Conference Application of Information and Communication Technologies*, Baku, pp. 1-4, 2011.

[5] Lee H., Kim M., Her J., and Lee H., "Implementation of Mapreduce-based Image Conversion Module in Cloud Computing Environment," *in Proceedings of the International Conference on Information Network*, Bali, pp. 234-238, 2012.

[6] Malik S., Huet F., and Caromel D., "Cooperative Cloud Computing in Research and Academic Environment using Virtual Cloud, Emerging Technologies (ICET)," *in Proceedings of the International Conference on Digital Object Identifier*, Islamabad, pp. 1-7, 2012.

[7] Patel A., Birla M., and Nair U., "Addressing Big Data Problem using Hadoop and Map Reduce, Engineering (NUiCONE)," *in Proceedings of Nirma University International Conference on Engineering*, Ahmedabad, pp. 1-5, 2012.

[8] Sood K., "A Combined Approach to Ensure Data Security in Cloud Computing," *the Journal of Network and Computer Applications*, vol. 35, pp. 1831-1838, 2012.

[9] Srirama S., Jakovits P., and Vainikko E., "Adapting Scientific Computing Problems to Clouds using Mapreduce," *Future Generation Computer Systems*, vol. 28, no.1, pp. 184-192, 2012.

[10] Wang L., Tao J., Ranjan R., Marten H., Streit A., Chen J., Chen D., and G-Hadoop, "Mapreduce Across Distributed Data Centers for Data-Intensive Computing," *Future Generation Computer Systems*, vol. 29, no. 3, pp. 739-750, 2013.

[11] Wu T., Chen C., Kuo L., Lee W., and Chao H., "Cloud-Based Image Processing System with Priority-based Data Distribution Mechanism," *Computer Communications*, vol. 35, no. 15, pp. 1809-1818, 2012.

**Mehraj Ali** is a Full-Time PhD Research Scholar at Anna University, India. He has completed his MSc degree at Anna University, India. His Research areas includes cloud computing, big data, image processing etc., he has published articles in various National/International Conferences in his domain. He has also, published articles in peer reviewed Journals.

**John Kumar** obtained his PhD degree in Computer Applications from Anna University, India in 2010. He has started his teaching profession in the year 2003 to serve his parent Institution Thiagarajar College of Engineering, Madurai, where he obtained his Master Degree in Computer Applications. He has published several research papers in International and National Journals as well as conferences. His field of interest includes spatial data mining, cloud computing and image processing.