

A Novel Approach for Transaction Management in Heterogeneous Distributed Real Time Replicated Database Systems

¹Mr.Dheeraj Bhimrao Lokhande, ²Prof.Dhainje Prakash B.

¹Research student, ²Professor
Department of Computer Science and Engineering
Solapur University, Solapur
Maharashtra, India
dheerajlokhande1985@gmail.com

Abstract: Today, Cloud computing is most widely used system due to number of benefits to end users. In IT (Information Technology) organizations, cloud computing is most vital domain to work. There are different types of services such as SaaS, IaaS and PaaS provided by cloud computing based on the end users needs. Rather than using own resources for data storage and Management, organizations started to utilize cloud computing data storage systems. Cloud storage (eg. Amazon S3) is emerging as a popular service and most of the enterprises shifts their data workloads to the cloud.

The popularity of Cloud computing system increased dramatically because it rent computing resources, bill on a pay-as-you-go basis and multiplex many users on same physical infrastructure. Cloud users are provided an illusion of infinite computing resources so that on demand resource consumption rate can be increased or decreased.

Cloud computing offers the vision of a virtually infinite pool of computing, storage and networking resources where applications can be deployed. It is the popular solution for on-demand and dynamic resource provisioning. The provisioning & maintenance of cloud resources are done with the help of resource management techniques (RM). The RM techniques are responsible to keep the track of free resources and assign the resources from free pool to incoming tasks.

Along with the growing demands of modern applications and workloads, cloud computing has gained prominence throughout the IT industry. Modern applications are growing along various dimensions such as the number of users, complexity, data size and many people are connecting to the internet through various devices.

Web Applications which deals with the cloud storage systems contain heterogeneous workload to deploy on the cloud. This heterogeneous workload needs to be handled carefully otherwise problems like long scheduling delay, starvation for low priority tasks may occur which can hurt the performance of the application significantly. To handle this workload characterization is much more important. In workload characterization, heterogeneous workload is divided into multiple task classes with similar characteristics in terms of resource and performance objectives.

It is important to consider heterogeneity of workload. The workload typically consists of diverse applications with different priorities and resource requirements. Failure to consider heterogeneous workload, it will lead to long scheduling delay, starvation by affecting performance of the application.

Modern applications faces challenges such as workload characterization, resource allocation and security. To meet these needs we propose a new framework which interacts with the heterogeneous database systems and provides workload characterization by using K-means clustering algorithm, resource allocation by providing virtual machines through requesting to the cloud and security by using AES algorithm for encryption and decryption.

In this research work, transactions are taken in form of heterogeneous files from users as input and deploy the files efficiently at cloud storage system as output. Transactional operation like insert, update, delete are performed on users data at cloud storage systems. Here, we are attempting to design novel approach for secure, energy efficiency, scalable transaction management system based on terminologies and methods of resource scheduling in heterogeneous distributed database systems.

We address "the transaction management problem", which is to interact with the heterogeneous database systems and to allocate and schedule computing resources with workload characterization by providing security to user's data.

Performance Evaluation of the proposed system is done based on fluctuating workloads and in terms of Throughput, Energy Efficiency, CPU Utilization, Scheduling Time, Response Time. The performance of the system is measured and compared with Container Based Scheduling(CBS), Dynamic Power-Saving Resource allocation(DPRA) mechanism and SLA-Aware Energy resource management(SLA) technique.

The Experimental results of the proposed system shows effective performance of the system. Results shows that proposed system outperforms existing techniques. The implemented environment is easy to use and the input to the system are given for execution in a normal way without changing or restructuring the code.

Keywords:-

Cloud Computing, Cloud Storage, Workload Characterization, Energy Efficiency, Resource Management , Resource Allocation

2) Literature Review:-

In this section, the study on all recent transaction management and data security methods is presented.

2.1 Task/Transaction Management Methods:-

In this research, the problem of transaction management is represented by means of task scheduling problem, therefore our studied methods listed below are presented their works recently on efficient task or resource scheduling in cloud computing systems for different types of transactions.

In [3], author proposed Heterogeneity-Aware dynamic capacity provisioning method for cloud data centres. Specifically, they first used the K-means clustering algorithm in order to divide workload into distinct task classes with similar characteristics in terms of resource and performance requirements. Then they introduced a method that dynamically adjusting the number of machines to minimize total energy consumption and scheduling delay. Simulations were conducted using traces from a Google's compute cluster demonstrate Harmony can reduce energy by 28 percent compared to heterogeneity-oblivious solutions. Figure 2 is showing the architecture proposed by author in [3].

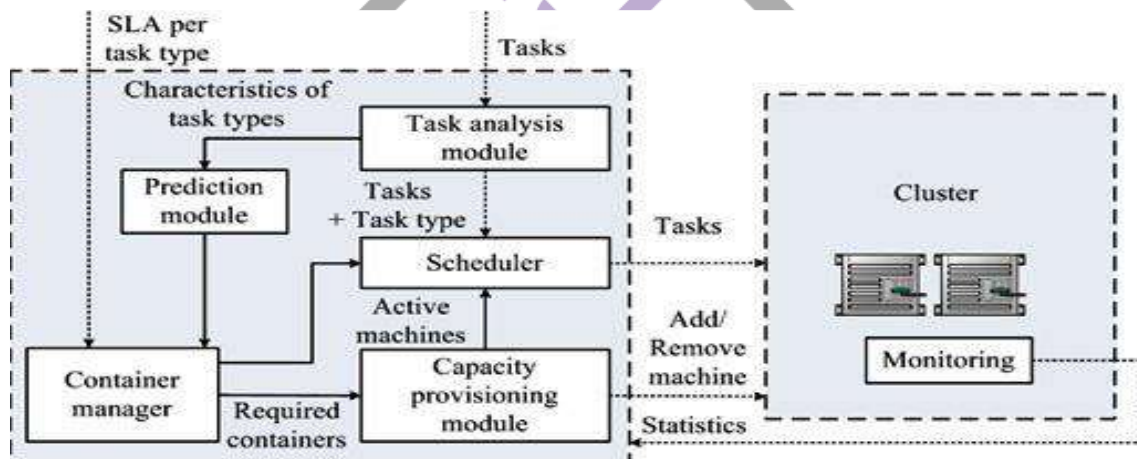


Figure 2.1: System Architecture Proposed in [3]

In [4], author introduced novel cost efficient resource provisioning method based on MapReduce cloud service framework called Cura. The Cura has a number of unique benefits introduced by author. Firstly, Cura was designed to provide a cost-effective solution to efficiently handle MapReduce production workloads that have a significant amount of interactive jobs. Secondly, unlike other services that require customers to decide the resources to be used for the jobs, Cura leverages MapReduce profiling to automatically create the best cluster configuration for the jobs. While the existing models allow only a per-job resource optimization for the jobs, Cura implemented a globally efficient resource allocation scheme that significantly reduces the resource usage cost in the cloud. Thirdly, Cura leverages unique optimization opportunities when dealing with workloads that can withstand some slack. By effectively multiplexing the available cloud resources among the jobs based on the job requirements, Cura achieved significantly lower resource usage costs for the jobs. Cura's core resource management schemes included cost-aware resource provisioning, VM-aware scheduling and online virtual machine reconfiguration.

In [5], bee colony based method for efficient load balancing was introduced, which was based on the foraging behavior of honey bees in order to balance load across VMs. In this method, tasks removed from over loaded VMs are treated as honey bees and under loaded VMs are the food sources. Additionally this approach considered the priorities of tasks in the waiting queues of VMs and tries to achieve minimum response time and reduced number of task migrations. The practical results were showing that there is significant improvement in the Quality of Service (QoS) using this approach.

In [6], author introduced an Energy-aware Resource Allocation method, named *EnReal*, to address the challenges of energy efficient resource allocation in cloud computing. Basically, they leveraged the dynamic deployment of virtual machines for scientific workflow executions. Specifically, an energy consumption model was presented for applications deployed across cloud computing platforms, and a corresponding energy-aware resource allocation algorithm was proposed for virtual machine scheduling to accomplish scientific workflow executions. The experimental analysis of this method was done by using cloudsims simulator.

In [7], another energy efficient resource allocation method introduced. Author proposed DPRA (dynamic power-saving resource allocation) method in order to improve energy efficiency. This DPRA mechanism not only considered the energy consumption of physical machine (PM) and virtual machine (VM) but also newly tackled the energy efficiency ratio of air conditioner. Moreover, the least squares regression method was utilized to forecast PM's resource utilization for allocating VM and eliminating VM migrations. Author evaluated the DPRA method against three other previous methods using simulations.

In [8], the recent workload and resource management algorithm in order to achieve the efficient resource allocation in cloud computing proposed. They introduced a proactive and cost-aware autoscaling solution in order to address previous methods problems by combining a predictive model, cost model, and a smart killing feature. They utilized a workload prediction mechanism based on time-series forecasting and machine-learning techniques. Their practical results showing that their ensemble method outperforms individual techniques, as well as some of the popularly used ensemble models, when it comes to accuracy. Additionally they designed greedy heuristic-scaling algorithm in order to address the resource allocation problem considering both the QoS and cost factors. This is one of the finest methods that we studied. Figure 3 demonstrating the architecture designed by author for this method.

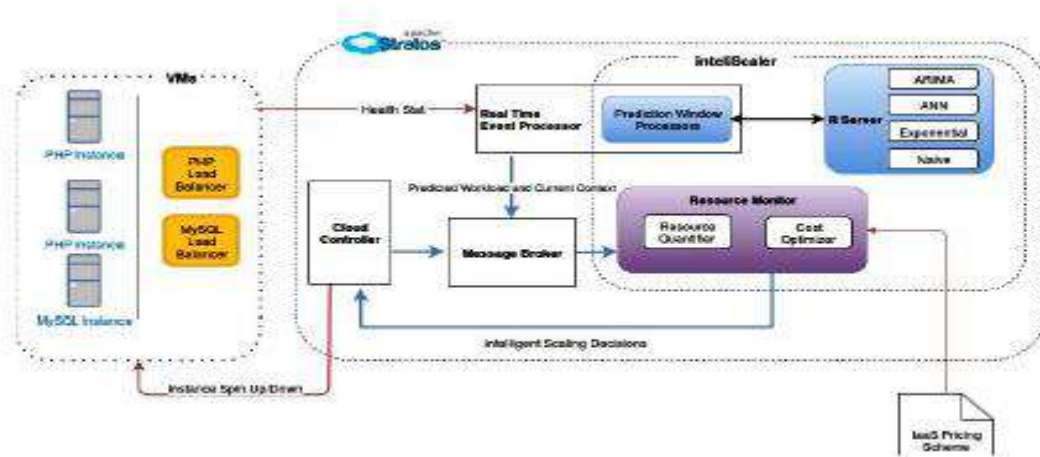


Figure 2.2: System Architecture Proposed in [8]

In [9], author presented workflow approach for efficient resource scheduling in cloud computing systems. First they designed the workflow framework and resource model based on queue.

3) Research Gap:-

Cloud Computing is emerged as an extremely successful paradigm for deploying web applications. The major reasons for the successful and widespread adoption of cloud infrastructures are scalability, elasticity, pay-per-use pricing and economics of scale.

Since the majority of cloud applications are data driven, database management systems(DBMS)is an integral technology component in the overall service architecture. Cloud storage systems are maintaining distributed database systems.

Modern cloud computing systems usually provide a highly scalable and fault-tolerant database that sacrifices other feature. Often, these systems may not support transactions at all or else restrict transactions to one data item each.

Recently techniques to support multi-item transactions in these types of systems have been successfully developed but have focused on transactions across homogenous data stores. However, applications often need to store different data in different storage systems perhaps for legacy or interoperability reasons. So it is difficult to perform transaction management across heterogeneous distributed database systems.

Cloud storage systems are most suitable and popular platforms for deployment for modern web applications which contains heterogeneous workloads. Analysis of these heterogeneous workloads is necessary because resource requirement may be different, different priorities, different performance objectives. Neglecting heterogeneity of workload web applications may get hurt in terms of performance due to long scheduling delay, starvation etc. so heterogeneous workload characterization i.e. task scheduling is overlooked in previous approaches.

Resource allocation is factor which should be done in such way to match the workload or task demand. After characterization of workload suitable resource should be provided so as to deploy on the cloud.

While dealing with the cloud computing, modern web application should protect users data being misused travelling over the internet. So web applications should provide security measures to protect the data over cloud storage systems.

4) Problem statement:-

“To develop hybrid framework for secure and energy efficient transaction management by using suitable algorithms”

The problem statement aim is to develop a novel framework which can interact with the heterogeneous database systems and cloud storage system. In this research is that after analyzing heterogeneity of the workload, workload characterization and task scheduling is performed. According to scheduling resources are allocated to different tasks and security techniques i.e. encryption and decryption are performed on user data for protection over cloud and traveling over the internet. This framework provides user public auditing techniques such as insert, update, delete, download. so that user can manipulate their data over heterogeneous databases and cloud storage systems.

5) Methodology:-

Modern application consists of diverse workload in term of different priorities, performance and resource requirements. While dealing with modern web based application workload analysis is the most important factor which needs to be considered. The modern web applications which deals with cloud face problem like long task scheduling delay, starvation which can significantly hurt the performance of the application. These problems occurs due to not considering workload analysis. Workload analysis can be very efficient by which energy efficient resource allocation is performed. Characterization of workload is performed by analyzing the workload. Task classification is important factor through which performance of the application can be improved. The improvement in the performance of the application can be achieved by identifying the requirements of the application such as performance and resource requirement.

Task scheduling delay is a primary concern for modern web applications for several reasons:-

- (1) A user may need to immediately scale up an application to accommodate a surge in demand and hence requires the resource request to be satisfied as soon as possible.
- (2) Even for lower-priority requests (e.g., background applications), long scheduling delay can lead to starvation, which can significantly hurt the performance of these applications.

In practice, however, there is often a tradeoff between energy savings and scheduling delay. Even though turning off a large number of machines can achieve high energy savings, at the same time, it reduces service capacity and hence leads to high scheduling delay.

Failure to consider the heterogeneity of the workloads will lead to both sub-optimal energy-savings and long scheduling delays, due to incompatibility between workload requirements and the resources offered by the provisioned machines.

In the context of workload scheduling in data centers, a metric of particular importance is scheduling delay which is the time a request waits in the scheduling queue before it is scheduled on a machine. Characterizing workload in production clouds has received much attention in recent years, as both scheduler design and capacity upgrade require a careful understanding of the workload characteristics in terms of arrival rate, requirements, and duration.

In this research work, after analyzing the heterogeneous workload task classification is performed by using k-means clustering algorithm. Using standard K-means clustering, we show that the heterogeneous workload can be divided into multiple task classes with similar characteristics in terms of resource and performance objectives. After workload analysis it can be found that workload composition are highly heterogeneous and dynamic over time.

In this research work, heterogeneity of the workload is considered. Specifically, we first use the K-means clustering algorithm to divide workload into distinct task classes with similar characteristics in terms of resource and performance requirements.

The proposed framework which is designed considers the heterogeneous nature of workload and perform following three operation:-

- 1) Dividing tasks into task classes using the K-means algorithm. whose main objective is to understand workload characteristics
- 2) Once workload characterization is obtained then allocate the resource i.e. Virtual machines to the characterized task
- 3) Perform security operations on the task before uploading on cloud storage systems.

To explain the framework, Amazon AWS cloud is used. The work mainly focuses on workload characterization using clustering of the input workload, allocate the resource by using AWS EC2 and performing security operations i.e. encryption and decryption.

The basic model of the framework is shown in the following figure.

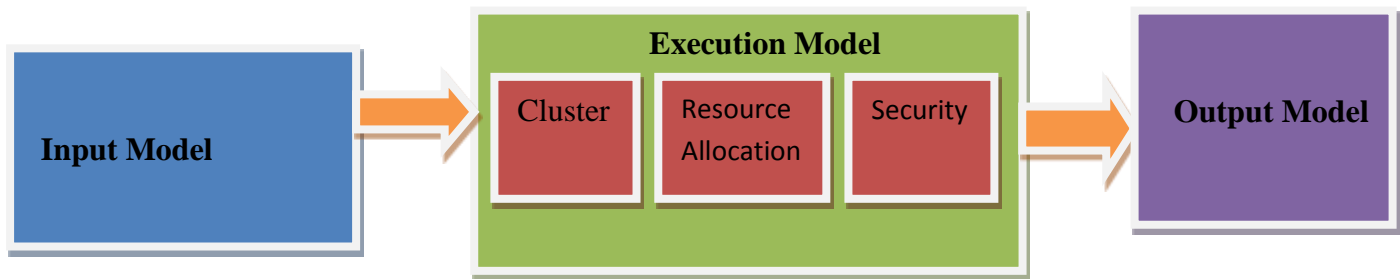


Figure 6.1 :- Basic Model of Proposed framework

The proposed framework contains three important models:-

- 1) Input model
- 2) Execution model
 - 1) Clustering
 - 2) Resource Allocation
 - 3) Security i.e. Encryption and Decryption
- 3) Output model

The framework communicates with Amazon AWS cloud to use the various services provided by the Amazon. The services are:-

- 1) **Elastic Compute cloud (EC2):-**
EC2 is used to allocate the resources i.e. Virtual Machines
- 2) **DynamoDB:-**
DynamoDB is used to create cloud database which stores important information regarding users and files.
- 3) **Simple Storage System (S3):-**
The framework performs encryption operation on the files and these encrypted files are stored on S3. So for storage S3 is used where a special space is created for framework used called buffer.

The framework utilizes MySQL database to store user and file related information. In MySQL, 'transaction' database is created which is collection of various tables.

The framework uses k-means clustering algorithm for workload characterization and for establishing task classes. These classes are formed based on type of the file, size of the file etc. Based on that two clusters are created. These clusters are allocated resources i.e. virtual machines by using Amazon EC2 cloud.

6) Execution Environment:-

We implemented the transaction management framework in java Netbeans IDE Environment. It contains many files such .jsp files, .java files and .servlet files. The framework is developed to work with Amazon AWS cloud. The framework takes heterogeneous files such as text files, word files, presentation files, excel files, image files, audio files, video files, pdf files etc. from user specified path. By using clustering algorithm, the framework creates two clusters of these files. These clusters are allocated resources by using resource allocation algorithm to upload the files to cloud storage systems.

The various services provided by AWS cloud is used in this research work such as EC2 for allocating the virtual machines, DynamoDB for creating database which is a collection of various tables and S3 to allocate the separate space for each user and store users encrypted files. The framework uses Advanced Encryption Standard (AES) algorithm to encrypt and decrypt the user specified files.

Along with Dynamodb database, the framework also works with MySQL database. To use MySQL database, SQLyog Community 13.1.1 32-bit tool is used. SQLyog community 13.1.1 is a GUI tool for RDBMS MySQL. It is used to create various tables which contains information related to users, files and other information regarding the execution of the framework. The framework separates every user by providing separate space for storage by using Amazon AWS S3.

7) Execution Flow of the framework:-

The framework is implemented in Java NetBeans IDE environment. The framework works with Amazon AWS Cloud services such as EC2, DynamoDB and S3. EC2 provides required resources i.e Virtual machines by which users files are uploaded on AWS cloud storage.

Dynamodb provides the database for storing users registration details and file related information. S3 is used for creating separate buffer for the framework which contains or provides separate space for each user to store uploaded files.

The Framework also works with another database i.e. MySQL which contains 'transaction' table for user registration information along with that it maintains file information details as well as values of graph table.

The web pages required for User Registration, Login, update, delete etc. are created in web pages module which are in .jsp format. The input files to the framework is taken from user specified path. The aws module contain all the necessary files required to communicated with cloud. The clustering of the files is performed by using cluster.java file.

The resource allocation request to Amazon AWS cloud is sent through EC2 module which includes necessary security group required for allocation of Virtual Machines (VM). Request.java file contains specification of virtual machines. MySQL database is connected to the framework through GetMyConnection.java file.

In this research work, the heterogeneous types files are taken as input which are shown in the following table.

Sr.No	Type of File	Sr.No.	Type of File
1	Text file (.txt)	5	Pdf file(.pdf)
2	Word file (.doc)	6	Image file (.jpeg)
3	Excel file(.xlsx)	7	Audio file (.mp3)
4	Presentation file(.ppt)	8	Video File (.mp4)

Table 7.1:- Heterogeneous nature of files

The transaction management framework is developed by using algorithms such as k-means clustering algorithm, resource allocation algorithm and security algorithm. The framework considers the heterogeneous nature of the workload which is the issue which is not considered in previous approaches. The previous methods does not considers the importance of workload characterization or task classification.

8) Experimental setup :-

The framework is implemented in Netbeans IDE 8.2 environment. It contains different files for clustering, resource allocation and security. The framework asks users about registration details. All this information regarding user is kept in both the databases i.e MySQL database and in Amazon AWS DynamoDB database. So MySQL server 5.1 is installed by using mysql-essential-5.1.54-win32 setup file. To use MySQL database graphically, SQLyog community 13.1.1 -32 bit is used.

The Amazon AWS Cloud is connected to the framework by using credential file which includes aws_access_key_id and aws_secret_access_key i.e access id and security key to access AWS cloud.

The performance of the framework is measured on computer system of core i3 supporting NetBeans environment with windows 10 operating system. The input files to the framework is taken from user specified path. The input files are of heterogeneous type with includes .doc files, .ppt files, .txt files, image and video files. The framework takes the input files in the range 01,05,10,15,20,25,30. The framework clusters these files, allocate resources and perform encryption and decryption on users data.

9) Experimental Methodology:-

The framework is implemented completely in Java Netbeans IDE environment. The experiment is conducted on a system with core i3 machine on the windows 10 platform. Connection with Amazon AWS cloud is made by using user Id and password. Then various services such as EC2, DynamoDB, S3 is set to be used by the framework to use and store necessary information in appropriate place.

At the same time MySQL database is used by using SQLyog Community Software. The SQLyog Community contains necessary information about the framework. It is a tables which contains important information regarding the framework such as user registration details, Generated key's for the files, files in the encrypted formats etc.

The input files given to the framework are varied in range 01, 05, 10, 15, 20, 25 and 30. According clustering is performed on user input and cluster are created by user file type, file size. These clusters are allocated to virtual machines for uploading. By the request through coding i.e dynamically, Amazon AWS EC2 service, provides the two virtual machines required for the framework to place files on Amazon S3 Cloud.

The database named DynamoDB which is present on the cloud contains number of table to store important information regarding the files. It contains information like uploaded files, Encrypted files, different keys of different files, information about deleted files in the various tables.

The storage service of Amazon S3 cloud used to create separate buffer for the framework. Each user on the framework is allocated separate space to store their files. The framework takes number of files from user as input then encrypts the files and place the files on AWS S3 storage on appropriate user's folder or space.

Administrator named 'PKG' is created by the framework which identifies the authenticated users on network. only after Administrator permission, each user is able to work with the files on the network and user can download file, update the file or delete the file.

The user can not delete the file directly on network. The delete request is first sent to Administrator. The administrator verifies the authenticity of the user and then delete permission is given to user.

The performance of the framework is checked by using parameters like throughput, CPU utilization, Energy Efficiency, Scheduling Time and Response Time. The framework is compared with Container Based Scheduling Technique (CBS), Dynamic Power Saving Resource Allocation (DPRA) method and SLA-Aware Energy Efficient Resource Management for Cloud Environments (SLA) method.

10) Results and Discussion:-

The framework deals with heterogeneous workload by clustering the workload, allocating the resources to the clusters and proving security to user data. The framework is compared with Container Based Scheduling(CBS), Dynamic Provisioning Resource Allocation(DPRA) and SLA-Aware Energy Efficient Resource Management for Cloud Environments(SLA). The main parameters considered for evaluating the performance of the framework are as follows:-

- 1) Throughput
- 2) CPU utilization
- 3) Energy Efficiency
- 4) Scheduling Time
- 5) Response Time

The results after executing the files in the range starting from input files 01,05,15,20,25,30 are collected in tabular form for CBS,DPRA ,SLA and proposed work .The input files for the framework can be from 1 to 30.so framework can upload 30 files with encryption on cloud storage system.

All the evaluating parameters are shown in tabular form and graphical form. Both form i.e tabular and graphical form contains values for files in the order for:-

1) Throughput:-

It is a measure of how much amount of time is required to upload an average of files on the cloud storage systems.

It is calculated as:-

Throughput value = Total file size / Number of files;

2) CPU utilization:-

It is a measure of how much percentage of CPU is utilized by the framework for uploading the files. For calculating this value OperatingSystemMXBean () method is used to calculate the usage of CPU.

It is calculated as:

CPU Utilization= usage. cpuuti();

3) Energy Efficiency:-

Energy Efficiency is calculated for encryption process of files. It is a measure of change in battery life divided by the number of files. It gives the battery life consumed in percentage.

It is a measure of the amount of battery life of the system consumed for encryption of number of files used by the framework. It is calculated in percnatage.

It is calculated as:-

Energy Efficiency = Change in battery life / Number of files;

4) Scheduling Time:-

It is the time a file has to wait for encryption. It is calculated in seconds.

It is calculated as:-

Scheduling Time = Delay Time – Encryption Time

Delay Time is the total time to upload the given files.

Encryption Time is the time required to encrypt a file.

5) Response Time:-

It is the total time taken by the framework from start to end. It is calculated in seconds.

It is calculated as:-

Response Time= stop time – start time

11) Comparison of proposed system with existing systems:-

In this section, tables and graphs are shown of the evaluation parameters. The lines represented in the graph contains following information.

As shown in the graph, The Green line belongs to SLA technique.

The Red line represents the CBS method.

The Blue line represents the DPRA technique.

The Yellow line represents the proposed work.

1) Throughput table and its corresponding Graph:-

Throughput				
No of Files	CBS	DPRA	SLA	Proposed
1	34.8	37.8	29.8	41.0
5	33.0	34.9	25.8	45.0
10	32.5	40.2	31.5	49.8
15	30.0	33.0	20.0	34.0
20	32.53333333333333	34.53333333333333	25.53333333333333	37.53333333333333
25	30.2	33.2	27.2	35.2
30	42.13333333333333	46.13333333333333	40.13333333333333	56.13333333333333

Table 1: Throughput Table

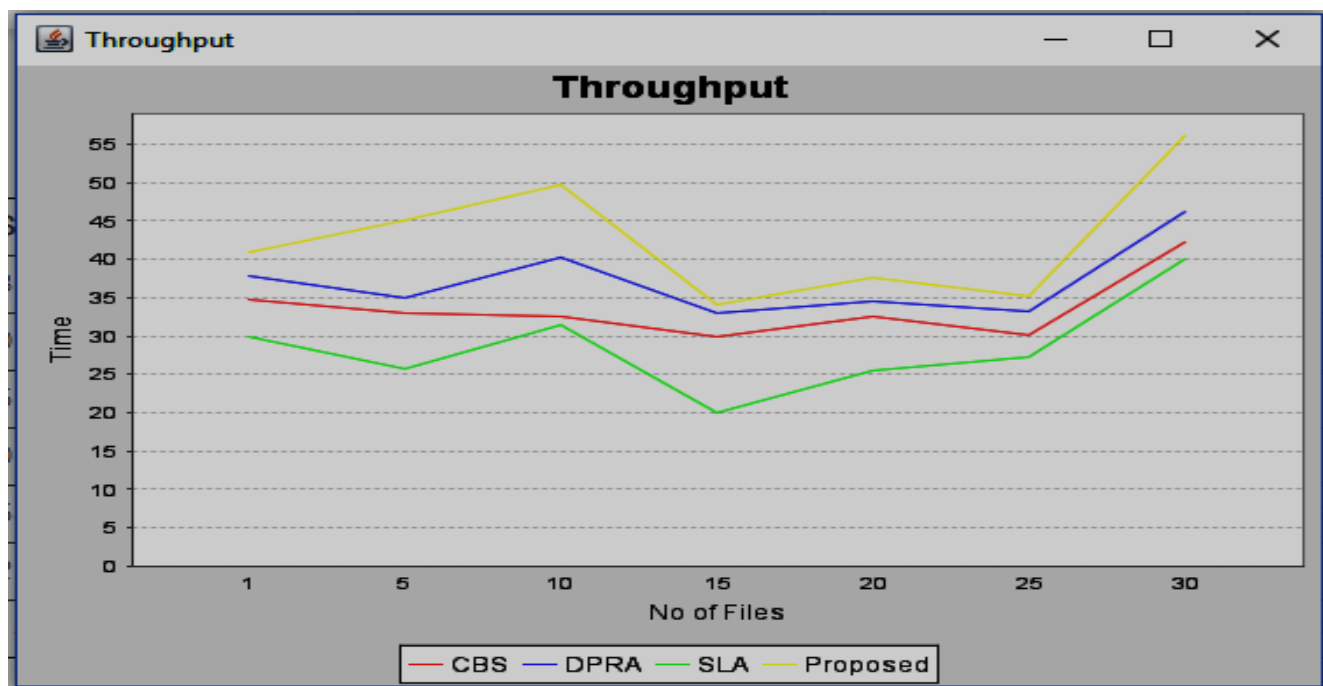


Figure 1:- Throughput Graph

2) CPU Utilization table and corresponding Graph:-

CPU Utilization				
No of Files	CBS	DPRA	SLA	Proposed
1	75.7662532271742	60.6201706016467	84.5783550026281	44.5
5	59.4642312943613	68.8886741375679	77.8137782422531	54.8
10	70.2896897098159	55.2034043270667	68.499180325764	50.2
15	73.3066442914394	63.4667155606033	82.341939305287	52.3
20	69.0106642552488	74.26709581871	58.0926088246559	56.0
25	64.4315840219083	61.5799613557383	73.1052172255203	50.0
30	75.9188608029918	47.0303201436986	75.4475134971304	45.0

Table 2 CPU Utilization

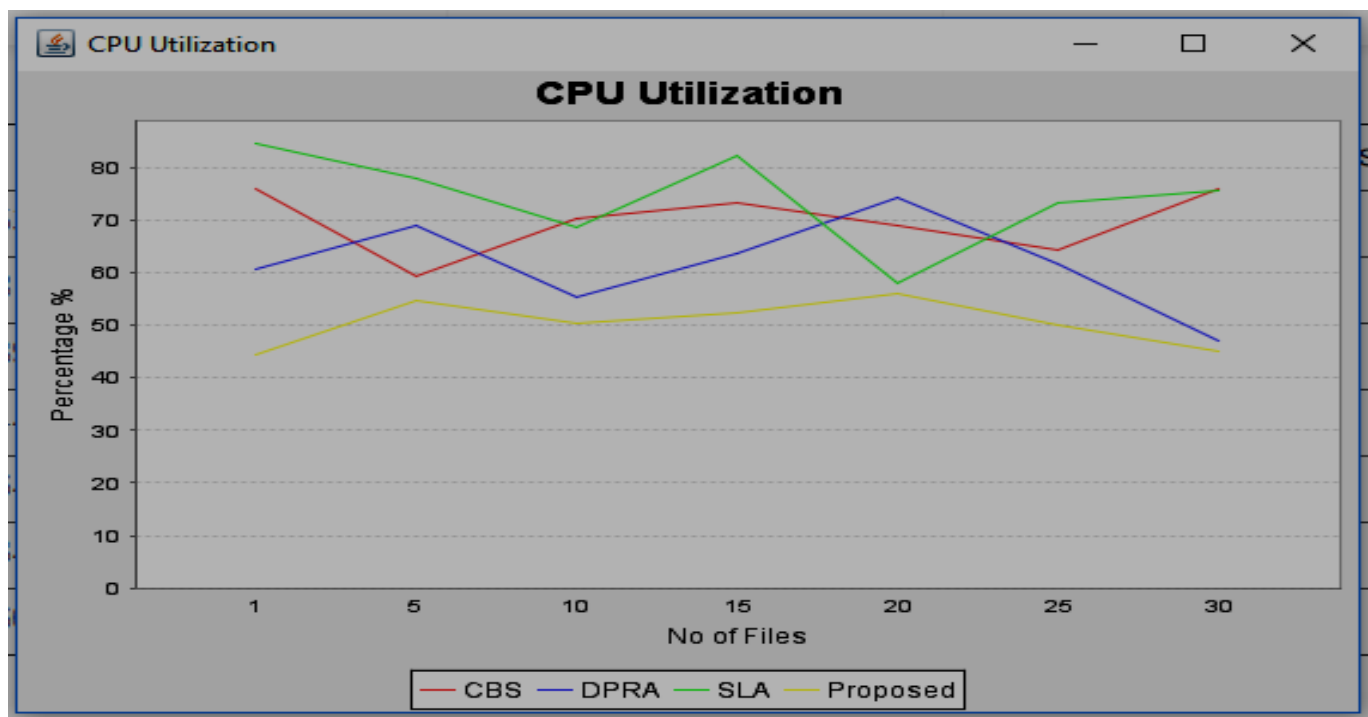


Figure 2 :CPU Utilization Graph

3) Energy Efficiency table and corresponding Graph :-

Energy Efficiency				
No of Files	CBS	DPRA	SLA	Proposed
1	1.0	1.2	1.5	0.5
5	1.2	1.8	2.0	1.0
10	2.0	2.7	2.8	1.5
15	2.8	3.0	3.5	2.5
20	3.2	3.3	3.8	3.0
25	3.6	3.5	4.0	3.1
30	3.7	3.8	4.8	3.2

Table 3Energy Efficiency

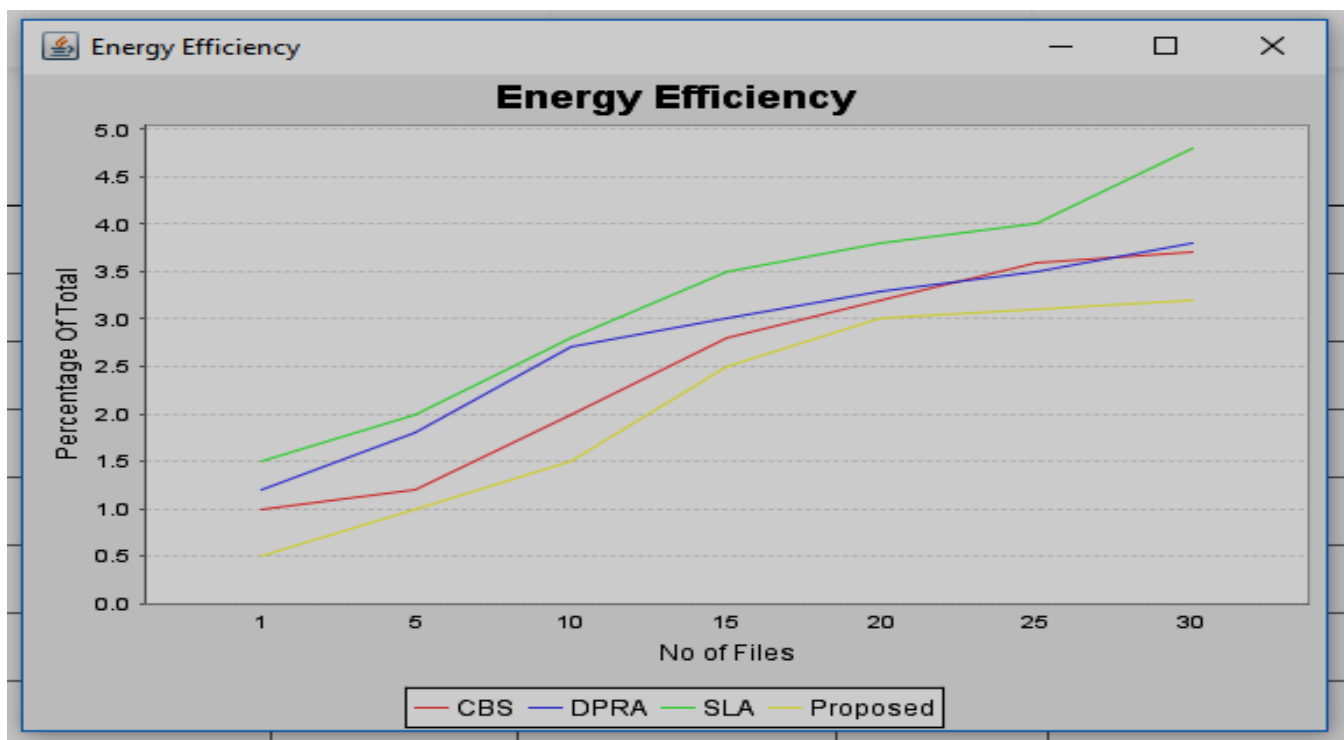


Figure 3:- Energy Efficiency Graph

4) Scheduling Time table and corresponding Graph:-

Scheduling Time				
No of Files	CBS	DPRA	SLA	Proposed
1	28.265672626493	18.0456179413985	36.0334378604978	12.0
5	48.1918102960254	35.5735725858699	62.0	32.0
10	74.2238972529987	64.5223128264684	83.1485414101388	53.0
15	106.366729124045	96.5856476539691	115.352382321887	85.0
20	139.009055404584	129.342637852055	148.032282704401	118.0
25	166.17859428492	156.36583232438	175.1225367804	145.0
30	200.399130350597	190.132971724377	209.144261963723	179.0

Table 4 Scheduling Time table

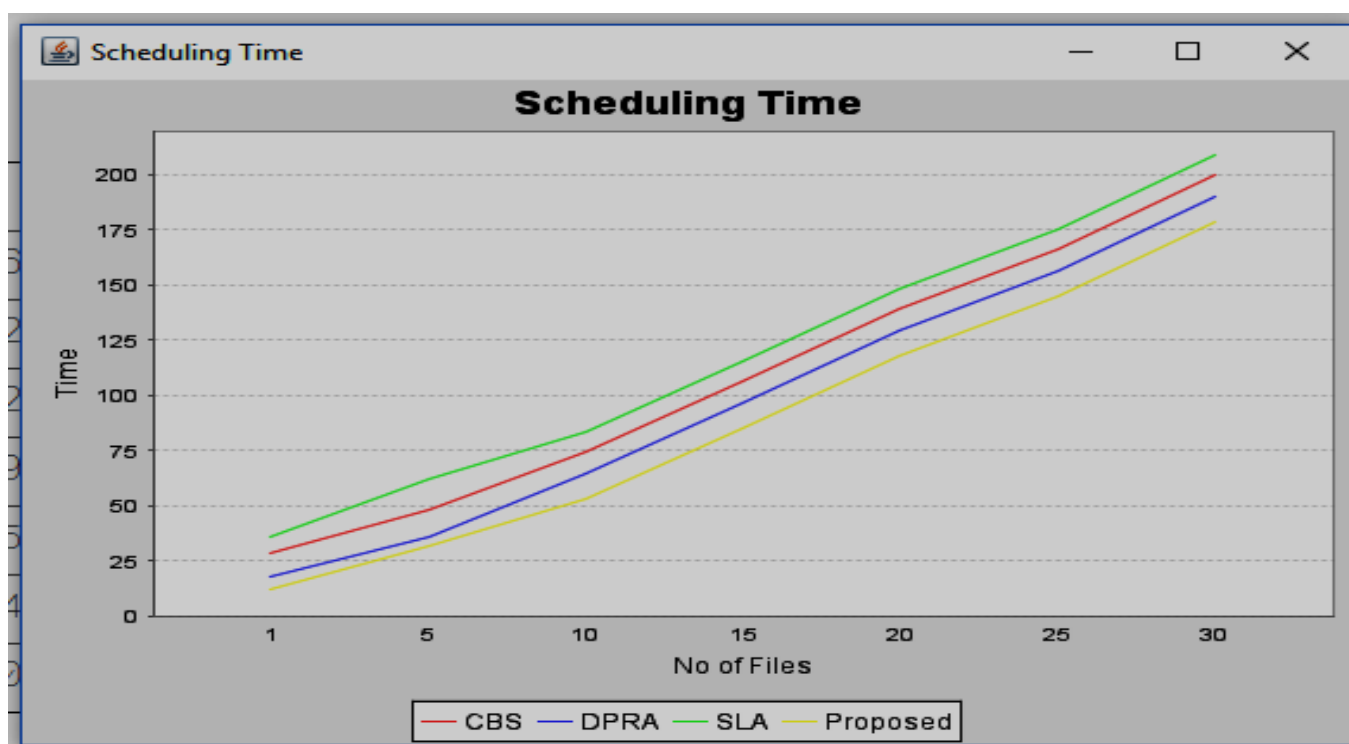


Figure 4:- Scheduling Time Graph

5) Response Time table and corresponding Graph:-

Response Time				
No of Files	CBS	DPRA	SLA	Proposed
1	47.147	57.147	67.147	22.147
5	77.163	52.163	82.163	39.163
10	143.897	143.897	128.897	88.897
15	117.753	137.753	177.753	102.753
20	245.158	245.158	216.158	196.158
25	226.047	225.947	226.447	202.447
30	300.812	274.812	312.812	254.812

Table 5 Response Time table

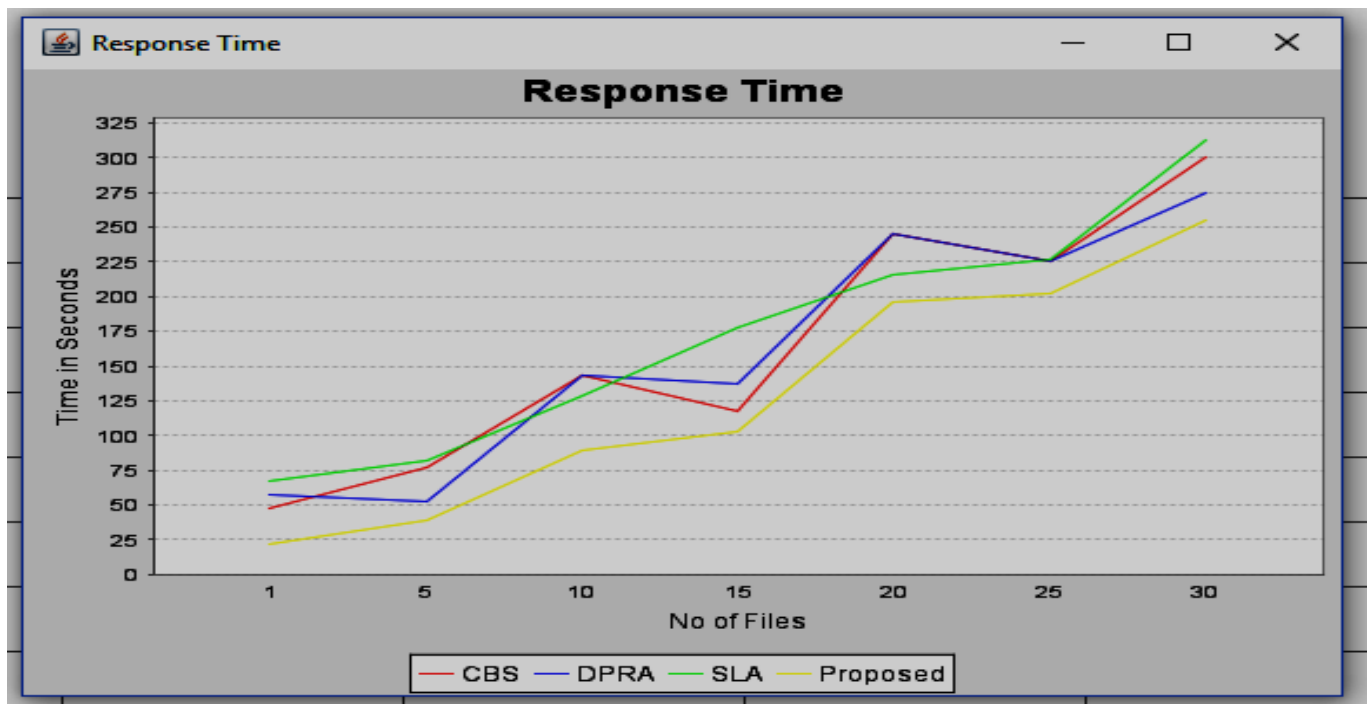


Figure 5:- Response Time Graph

References:-

- [1] B. Hayes, "Cloud computing," Commun. ACM, vol. 51, no. 7, p. 9, Jul. 2008.
- [2] R. G. Tiwari, S. B. Navathe, and G. J. Kulkarni, "Towards Transactional Data Management over the Cloud," 2010 Second Int. Symp. Data, Privacy, E-Commerce, pp. 100–106, Sep. 2010.
- [3] Qi Zhang, Mohamed FatenZhani, RaoufBoutaba, Joseph L. Hellerstein, "Dynamic Heterogeneity-Aware Resource Provisioning in the Cloud", IEEE TRANSACTIONS ON CLOUD COMPUTING, VOL. 2, NO. 1, JANUARY-MARCH 2014
- [4] BalajiPalanisamy, Aameek Singh, Ling Liu, "Cost-effective Resource Provisioning for MapReduce in a Cloud", IEEE Transactions on Parallel and Distributed Systems, 2014.
- [5] K R RemeshBabu, Amaya Anna Joy, Philip Samuel, "Load Balancing Of Tasks In Cloud Computing Environment Based On Bee Colony Algorithm", 5th International Conference on Advances in Computing and Communications, IEEE, 2015

Author Information:-



Author name is Mr.Dheeraj Bhimrao Lokhande. Currently, he is pursuing Ph.D. in computer science and Engineering from Solapur University, Solapur. He is completed his bachelor degree in information technology in 2008 and completed Master in computer science and engineering in 2012. He qualified GATE examination in 2008.