**“An Energy Efficiency of Cloud Based Services Using EaaS Transcoding of the Multimedia Data”**

A Thesis submitted

To

**Rashtrasant Tukadoji Maharaj Nagpur University,**

**Nagpur, India**

In partial fulfillment

For the award of the degree

**Master of Technology**

**In**

**Computer Science & Engineering**



***Submitted by****Ms. Harshal P. Ganvir*

**Under the Guidance of**

*Prof. Pravin G. Kulurkar*

**Department of Computer Science & Engineering**

Vidarbha Institute of Technology,

Uti, Umrer Road, Nagpur-441209

2015-2016

©Vidarbha Institute of Technology, Nagpur

**ACKNOWLEDGEMENT**

The written word has an unfortunate tendency to convert genuine gratitude into a stilled formality. But this is only way by which I can record my feelings permanently. Many people have contributed their time and efforts in helping me complete my dissertation work successfully.

I would like to express my deepest gratitude to my guide **Prof. Pravin G. Kulurkar** whose continuous guidance and motivation helped me to achieve greater heights.

I also express my gratitude to **Dr. Pradeep Agnihotri** (Principal, V.I.T. Nagpur) for their constant encouragement and motivation. I am also thankful to **Prof. Pravin Kulurkar** (H.O.D. Computer Science and Engineering Department) for their support and help which encouraged me a lot in completing this project.

And last but not the least I would like to thank my parents, batch mates, friends and all those people who have directly or indirectly helped me in carrying out my work successfully.

**Harshal P. Ganvir**

VIDARBHA INSTITUTE OF TECHNOLOGY, NAGPUR

**Department of Computer Science & Engineering**

Session 2015-16

**C E R T I F I C A T E**

This is to certify that the project titled **“An Energy Efficiency of Cloud Based Services Using EaaS Transcoding of the Multimedia Data”**has been successfully completed by **Harshal P. Ganvir** under the guidance of **Prof. Pravin G. Kulurkar** in recognition to the partial fulfillment for the degree of Master of Technology in Computer Science and Engineering Department, Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur.

**Prof. Pravin G. Kulurkar**

Project Guide & H.O.D

Department of Computer Science & Engineering

**Dr. Pradeep Agnihotri**

Principal

Vidarbha Institute of Technology, Nagpur

**DECLARATION**

I hereby declare that the thesis titled **“An Energy Efficiency of Cloud Based Services Using EaaS Transcoding of the Multimedia Data”** submitted here in has been carried out by me in the Department of Computer Science and Engineering of **Vidarbha Institute of Technology, Nagpur**. The work is original and has not been submitted earlier as a whole or in part for the award of any degree / diploma at this or any other Institution / University.

**Name of Candidate:** Harshal P. Ganvir

**Enrollment No:**…………………..

**Date:**

**Place:** Nagpur

**CERTIFICATE BY THE EXAMINERS**

The thesis entitled **“An Energy Efficiency of Cloud Based Services Using EaaS Transcoding of the Multimedia Data”** Processing submitted **Harshal P. Ganvir** Roll No.\_\_\_\_\_\_\_\_\_\_, Enrollment No………….has been examined by the undersigned as a part of the examination and is hereby recommended for the award of the degree of Master of Technology in the faculty of Computer Science and Engineering of Vidarbha Institute of Technology, Nagpur.

……………………………….. ………………………………..

Internal Examiner External Examiner

Date Date

**INDEX**

|  |  |  |
| --- | --- | --- |
| **CHAPTER**  **NO.** | **NAME OF CHAPTER** | **PAGE NO.** |
|  | Abstract | i |
| List of Figure | ii |
| List of Tables | iii |
| List of abbreviations | iv |
| **CHAPTER 1** | **INTRODUCTION** | 01-08 |
| 1.1 | Overview | 01 |
| 1.2 | Outlier Detection | 02 |
| 1.3 | Support Vector Data Description | 03 |
| 1.4 | Difficulties in Outlier Detection | 05 |
| 1.5 | Existing Algorithm for Outlier Detection in Categorical Dataset | 05 |
| 1.5.1 | Greedy Algorithm | 05 |
| 1.5.2 | AVF (Attribute Value Frequency) | 06 |
| 1.5.3 | NAVF (Normally Distributed Attribute Value Frequency) | 06 |
| 1.5.4 | ROAD (Ranking-Based Outlier Analysis and Detection Algorithm) | 06 |
| 1.6 | Applications of Outlier Detection | 06 |
| 1.7 | Thesis Objective | 08 |
| **CHAPTER 2** | **LITERATURE SURVEY** | 09-13 |
| 1.1 | Overview | 09 |
| 1.2 | Detailed of LOF, COF, and INFLO Algorithms | 10 |
| 1.2.1 | LOF (Local Outlier Factor) approach | 10 |
| 1.2.2 | COF (Connectivity-based Outlier Factor) approach | 10 |
| 1.2.3 | INFLO (INFLuential measure of Outlierness by symmetric relationship) approach | 11 |
| 1.3 | Comparative Study | 12 |
| **CHAPTER 3** | **EXISTING METHODOLOGY** | 14-21 |
| 3.1 | Greedy Algorithm | 13 |
| 3.2 | AVF (Attribute Value Frequency) | 14 |
| 3.3 | NAVF (Normally Distributed Attribute Value Frequency) | 15 |
| 3.4 | ROAD (Ranking-Based Outlier Analysis and Detection Algorithm) | 16 |
| 3.5 | Statistical based | 18 |
| 3.6 | Distance Based | 19 |
| 3.7 | Kernel K-Means Clustering-Based Method | 19 |
| 3.8 | Kernel LOF-Based Method | 20 |
| **CHAPTER 4** | **PROPOSED METHODOLOGY** | 22-25 |
| 4.1 | Proposed Methodology | 22 |
| **CHAPTER 5** | **IMPLEMENTATION OF PROPOSED METHODOLOGY** | 26-34 |
| 5.1 | Data | 24 |
| 5.2 | Data Processing | 25 |
| 5.3 | Algorithm Implementation | 26 |
| 5.3.1 | NAVF (Normally Distributed Attribute Value Frequency) | 26 |
| 5.3.2 | ROAD (Ranking-Based Outlier Analysis and Detection Algorithm) | 29 |
| 5.3.3 | Hybrid Algorithm | 31 |
| **CHAPTER 6** | **RESULT ANALYSIS AND DISCUSSION** | 35-42 |
| 6.1 | Comparison with standard design / Standard Result/ Expected Result | 32 |
| 6.2 | Experimental Outcomes | 33 |
| 6.3 | Comparison with standard design / Standard Result/ Expected Result | 36 |
| 6.3.1 | Comparison of NAVF with Our New Algorithm (Hybrid) | 36 |
| 6.3.2 | Comparison of RANK with Our New Algorithm (Hybrid) | 38 |
| 6.4 | Comparison with Existing algorithm | 40 |
| **CHAPTER 7** | **CONCLUSION & FUTURE SCOPE** | 43 |
| **CHAPTER 8** | **REFERENCES** | 44-46 |
|  | **Annexure** |  |
|  | **Publication** |  |

**ABSTRACT**

Network-based cloud computing is rapidly expanding all over as an alternative to conventional office-based computing. Cloud computing has become widespread and the energy consumption of the network and computing resources will grow cloud. This  
happens at a time when there is increasing attention being paid to the need to manage energy consumption across the entire information and communications technology (ICT) sector. Also data center energy use have much attention, as there has been less attention paid to the energy consumption of the transmission and switching networks. This paper, presents an analysis of energy consumption in cloud computing. The analysis will consider both public and private clouds. We show that energy consumption in transport and switching can be a significant percentage of total energy consumption in cloud computing. Cloud computing provides more energy efficiency and use of computing power. Computing tasks are of low intensity or infrequent. Thus, under some circumstances cloud computing may consume more energy than conventional computing where each user performs all computing on their own personal computer (PC).

The De-duplication method best suited to protect data in cloud. This process De-duplicates data both across backups and within backups and does not require any knowledge of the backup data format. The job can be system allocation can be performed for the batch jobs with the sequence of job allocation. And the content similarity is used for the de-duplications process and filtering the De-duplication content. In the time interval, the job can be finished with the effective resources then allocation can be in the order sequences. The included automates filtering, to help an analyst in cloud with similar content by designating of Data duplication can be easily removed by the content similarity algorithm. The workloads can be categorized as per the order of the job work load can be assigned. The scheduling can be maintained as per the sequence of the job within the time interval the particular job can be executed.

**LIST OF FIGURE**

|  |  |  |  |
| --- | --- | --- | --- |
| **SR.NO.** | **FIGURE**  **NO.** | **TITLE OF FIGURE** | **PAGE NO** |
| 1. | 1.4 | Types of service models |  |
| 2. | 4.5 | Experimental Analysis and Results |  |
| 3. | 5.1 | System Architecture |  |
| 4. | 5.3 | Load balancing in the Cloud |  |

**LIST OF TABLE**

|  |  |  |  |
| --- | --- | --- | --- |
| **SR.NO.** | **TABLE NO.** | **TITLE OF TABLE** | **PAGE NO.** |
| 1. | 2.1 | Literature Survey | 12 |
| 2. | 6.1 | Comparison with existing system | 42 |

**LIST OF ABBREVIATION**

|  |  |  |
| --- | --- | --- |
| **Sr.No.** | **Abbreviation** | |
| 1. | SAAS | Software as a Service |
| 2. | IAAS | Infrastructure as a Service |
| 3. | PAAS | Platform as a Service |
| 4. | JMS | Java Messaging Service |
| 5. | PVFS | A Parallel File System Clusters |

**INTRODUCTION**

1. **Overview**

The increasing number of availability of high-speed Internet and corporate IP connections is enabling the delivery of new network-based services. Internet-based mail services have been operating for many years so service offerings have recently expanded to include network-based storage and network-based computing. These new services are offered both to corporate and individual users. The cloud computing service model involves the provision to a service provider, of large pools of high performance computing resources and high-capacity storage devices which are shared among end users. There are many cloud service models, generally end users subscribing to the service have their data hosted by the service and also have computing resources allocated. The service provider’s may also extend to the software applications required by the end user. To be successful, the cloud service model may also requires a high-speed network to provide connection between end user and service provider’s infrastructure. Cloud computing offers an overall financial benefit, in that end users share a large, centrally managed pool of storage and the computing resources, rather than owning and managing their own systems. Often using existing data centers as a basis, cloud service providers invest in the necessary infrastructure and management systems, and in return receive a time-based from end users. Since at any one time, substantial number of the end users are inactive, the benefits of the economies of scale and from statistical multiplexing, and also receives a regular incoming stream from the investment by means of service subscription. The end user sees convenience benefits from data and services available from any location, also from having data backups centrally managed, from the availability of increased capacity when needed and from usage-based charging. Important is the last point for many users in that it averts the need for a large one off investment in hardware, sized to suit maximum demand, and also requires upgrading for every few years. We present an overview of energy consumption in cloud computing and compare this to energy consumption in conventional computing. It is the energy consumed when the same task is carried out on a consumer personal computer (PC) that is connected to the Internet but does not use cloud computing method. We consider both

public and private clouds that include energy consumption in switching and transmission, data processing and data storage. Specifically, we present a network-based model of user computing equipment, and a model of the processing and storage functions in data centers. We examine a variety of cloud computing service scenarios in terms of energy efficiency. Our approach is to view cloud computing as an analog of a classical supply chain logistics problem, which considers the energy consumption or cost of processing, storing, and transporting physical items. The difference in our case, the items are bits of data. We explore a number of practical examples in which users customers outsource their computing and storage needs to a public cloud or private cloud. Three cloud computing services are considered, including storage as a service, processing as a service and software as a service. As the name implies, storage as a service allows users to store data in the cloud. Processing as a service gives users the ability to outsource selected computationally intensive tasks to the cloud. Software as a service combines these two services and allows users to outsource all their computing to the cloud and use only a very-low-processing-power terminal at home. We showed that energy consumption in transport and switching can be a significant percentage of total energy consumption. Cloud computing can enable more energy-efficient use of computing power. However, we show that under some circumstances cloud computing can consume more energy than conventional computing on a local PC. Our conclusion is that cloud computing can offer significant energy saving techniques such as virtualization and consolidation of servers and advanced cooling systems.

**1.2 Cloud Computing Description**

Cloud computing is the phrase used to describe different scenarios in which computing resource is delivered as a service over a network connection. Cloud computing is therefore a type of computing that relies on sharing a pool of physical and/or virtual resources, rather than deploying local or personal hardware and software. It is somewhat synonymous with the term ‘utility computing’ as users are able to tap into a supply of computing resource rather than manage the equipment needed to generate it themselves; much in the same way as a consumer tapping into the national electricity supply, instead of running their own generator. One of the key characteristics of cloud computing is the flexibility that it offers and one of the ways that flexibility is offered is through scalability. This refers to the ability of a system to adapt and scale to changes in workload. Cloud technology allows for the automatic provision and deprovision of resource as and when it is necessary, thus ensuring that the level of resource available is as closely matched to current demand as possible. This is a defining characteristic that differentiates it from other computing models where resource is delivered in blocks (e.g., individual servers, downloaded software applications), usually with fixed capacities and upfront costs. With cloud computing, the end user usually pays only for the resource they use and so avoids the inefficiencies and expense of any unused capacity.

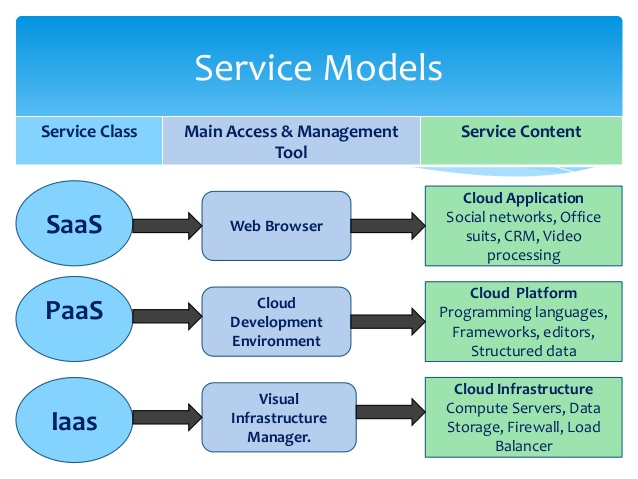
Now days with the huge increasing of population and the using of technology, it leads to many problems. The growth in technology is increasing the amount of storage or communication and technique devices. There are many data sources like cameras, mobiles, tablets and computers etc. Sometimes one person has more than one device and storing data from all of them or he/she wants to use his/her data on any device that have at any time. The old technology was to connect the devices to each other physically and start transferring data between them. The need for new technology that should be easier and faster appears. The scientist and researchers start thinking for solution until they found the cloud storage. Many clouds are offered from different brands. After the cloud become popular and used from a huge number of users many problems are appearing again with cloud. Some of these problems are external and other internal. Security was one of the big challenges of the cloud service providers. The implement and applied algorithms to keep the customer's information secure. The data duplication is one of the big challenges in cloud computing. The duplicated data effects on the storage and the performance of the cloud. Previous researches show that about 90% of the data that are stored in cloud backup are duplicated. The researchers start studying for deduplication techniques to optimize the storage. There is more than one way to deduplicate the data. Either by analyzing the data within the uploading and see whether it matches anything that is stored in storage, if it matches just ignore it or by uploading the data, then apply an algorithm to analyze and check if any data are matched keep one and delete the others. This study focus on the first technique which ignore uploading the duplicated data. Hash algorithm is used to check the data if it is matched ignore the uploading else count it as a unique and continue uploading to store it.

**1.3 Energy Efficiency in Cloud Computing:**

Energy efficiency is increasingly important for future information and communication technologies (ICT), because the increased usage of ICT, together with increasing energy costs and the need to reduce green house gas emissions call for energy-efficient technologies that decrease the overall energy consumption of computation, storage and communications. Cloud computing has recently received considerable attention, as a promising approach for delivering ICT services by improving the utilization of data centre resources. In principle, cloud computing can be an inherently energy-efficient technology for ICT provided that its potential for significant energy savings that have so far focused on hardware aspects, can be fully explored with respect to system operation and networking aspects.

Cloud computing is a fundamentally more efficient way to operate compute infrastructure. The increases in efficiency driven by the cloud are many but a strong primary driver is increased utilization. All companies have to provision their compute infrastructure for peak usage. But, they only monetize the actual usage which goes up and down over time. Cloud computing gets an easy win on this dimension. When non-correlated workloads from a diverse set of customers are hosted in the cloud, the peak to average flattens dramatically.

**1.4 Types of service models :**



1. **Storage as a Service:**

SaaS is a [software](http://www.webopedia.com/TERM/S/software.html) delivery method that provides access to software and its functions remotely as a Web-based service. Software as a Service allows organizations to access business functionality at a cost typically less than paying for licensed [applications](http://www.webopedia.com/TERM/A/application_software.html) since SaaS pricing is based on a monthly fee. Also, because the software is hosted remotely, users don't need to invest in additional [hardware](http://www.webopedia.com/TERM/H/hardware.html). Software as a Service removes the need for organizations to handle the installation, set-up and often daily upkeep and maintenance. Software as a Service may also be referred to as simply hosted applications. With SaaS, cloud providers host and manage the software application and underlying infrastructure and handle any maintenance, like software upgrades and security patching. Users connect to the application over the Internet, usually with a web browser on their phone, tablet or PC.

# [Platform as a Service](http://www.interoute.com/vdc):

PaaS is a category of cloud computing that provides a platform and environment to allow developers to build applications and services over the internet. PaaS services are hosted in the cloud and accessed by users simply via their web browser. Platform as a Service allows users to create software applications using tools supplied by the provider. PaaS services can consist of preconfigured features that customers can subscribe to; they can choose to include the features that meet their requirements while discarding those that do not. Consequently, packages can vary from offering simple point-and-click frameworks where no client side hosting expertise is required to supplying the infrastructure options for advanced development. Services are constantly updated, with existing features upgraded and additional features added. PaaS providers can assist developers from the conception of their original ideas to the creation of applications, and through to testing and deployment. This is all achieved in a managed mechanism. As with most cloud offerings, PaaS services are generally paid for on a subscription basis with clients ultimately paying just for what they use. Clients also benefit from the economies of scale that arise from the sharing of the underlying physical infrastructure between users, and that results in lower costs.

1. [**Infrastructure as a Service**](http://www.interoute.com/vdc)**:**

IaaS is one of the three fundamental service models of cloud computing alongside Platform as a Service (PaaS) and Software as a Service (SaaS). As with all cloud computing services it provides access to computing resource in a virtualised environment, “the Cloud”, across a public connection, usually the internet. In the case of IaaS the computing resource provided is specifically that of virtualised hardware, in other words, computing infrastructure. The definition includes such offerings as virtual server space, network connections, bandwidth, IP addresses and load balancers. Physically, the pool of hardware resource is pulled from a multitude of servers and networks usually distributed across numerous data centers, all of which the cloud provider is responsible for maintaining. The client, on the other hand, is given access to the virtualised components in order to build their own IT platforms. In common with the other two forms of cloud hosting, IaaS can be utilised by enterprise customers to create cost effective and easily scalable IT solutions where the complexities and expenses of managing the underlying hardware are outsourced to the cloud provider.

**LITERATURE SURVEY**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Title of the paper | Author | Major observations / findings |
| 1 | “A Survey On Reducing Energy Sprawl in Cloud Computing” | Malathi.P (July 2015) | Author in this paper have studied various techniques for reducing the energy sprawl in cloud computing. The two amplifying factors are virtualization and task consolidation plays a vital role in optimizing energy consumption. Thus, this survey can be used to enhance the energy consumption models by designing energy prediction models, energy optimization models and energy consumption monitors for the cloud system. |
| 2 | “Energy Efficiency Model For Cloud Computing.” | Arindam Banerjee, Prateek Agrawal and N. Ch. S. N. Iyengar(2013) | Author in this paper have studied the need of power consumption and energy efficiency in cloud computing model. It has been shown that there are few major components of cloud architecture which are responsible for high amount of power dissipation in cloud. Finally we have shown the future research direction and the continuity of this work for next level implementation. |
| 3 | “Energy Efficient Allocation of Virtual Machines in Cloud Data Centers.” | Anton Beloglazov and Rajkumar Buyya (2010) | Author in this paper have studied and have proposed and evaluated heuristics for dynamic reallocation of VMs to minimize energy consumption, while providing reliable QoS. The obtained results show that the technique of dynamic reallocation of VMs and switching off the idle servers brings substantial energy savings and is applicable to real-world. |
| 4 | “On the Reliability and Energy Efficiency in Cloud Computing.” | Yogesh Sharma, Bahman Javadi, Weisheng Si (2015) | Author in this paper have studied to identify the need of a reliability-aware and energy-aware resource provisioning policy to improve the availability of the services of cloud by reducing the energy consumption. |
| 5 | “Techniques for Efficient Keyword Search in Cloud Computing.” | Raakesh, Varun Raj G , Krishna R, Mr.L.Maria, Michael Visuwasam March(2015) | Author in this paper have studied the notion of authorized data decompression was proposed to protect the data security by including differential privileges of users in the duplicate check and presented several new de-duplication constructions supporting authorized duplicate check in hybrid cloud. |
| 6 | “Dynamic job Scheduling in Cloud Computing based on horizontal load  Balancing” | Mousumi Paul, Debabrata Samanta, Goutam Sanyal | Author in this paper have studied the load balancing mechanism in the central middleware which reduces the overhead of scheduling on a single middleware by partitioning the job queue. |

**Table 2.1: Literature Survey**

**EXISTING METHODOLOGY**

**3.1 The Effectiveness of Deduplication on Virtual Machine Disk Images:**

Virtualization is becoming widely deployed in servers to efficiently provide many logically separate execution environments while reducing the need for physical servers. While this approach saves physical CPU resources, it still consumes large amounts of storage because each virtual machine (VM) instance requires its own multi-gigabyte disk image. Moreover, existing systems do not support ad hoc block sharing between disk images, instead relying on techniques such as overlays to build multiple VMs from a single “base” image. Instead, we propose the use of De-duplication to both reduce the total storage required for VM disk images and increase the ability of VMs to share disk blocks. To test the effectiveness of De-duplication, we conducted extensive evaluations on different sets of virtual machine disk images with different chunking strategies. Our experiments found that the amount of stored data grows very slowly after the first few virtual disk images if only the locale or software con- figuration is changed, with the rate of compression suffering when different versions of an operating system or different operating systems are included. We also show that fixed length chunks work well, achieving nearly the same compression rate as variable-length chunks. Finally, we show that simply identifying zero-filled blocks, even in ready-to use virtual machine disk images available online can provide significant savings in storage.

**3.2 An Empirical Analysis of Similarity in Virtual Machine Images:**

To efficiently design De-duplication, caching and other management mechanisms for virtual machine (VM) images in Infrastructure as a Service (IaaS) clouds, it is essential to understand the level and pattern of similarity among VM images in real world (IaaS) environments. This paper empirically analyzes the similarity within and between 525 VM images from a production (IaaS) cloud. Besides presenting the overall level of content similarity, we have also discovered interesting insights on multiple factors affecting the similarity pattern, including the image creation time and the location in the image’s address space. Moreover, we found that similarities between pairs of images exhibit high variance, and an image is very likely to be more similar to a small subset of images than all other images in the repository. Groups of data chunks often appear in the same image. These image and chunk “clusters” can help predict future data accesses, and therefore provide important hints to cache placement, eviction, and prefetching.

**3.3 Efficiently Storing Virtual Machine Backups:**

Physical level backups offer increased performance in terms of throughput and scalability as compared to logical backup models, while still maintaining logical consistency. As the trend toward virtualization grows, virtual machine backups (a form of physical backup) are even more important, while becoming easier to perform. The downside is that physical backup generally requires more storage, because of file system meta-data and unallocated blocks. De-duplication is becoming widely accepted and many believe that it will favor logical backup, but this has not been well studied and the relative cost of physical vs. logical on de duplicating storage is not known. In this paper, we take a data-driven approach using user data to quantify the storage costs and contributing factors of physical backups over numerous generations. Based on our analysis, we show how physical backups can be as storage efficient as logical backups, while also giving good backup performance.

**3.4 Allow Bandwidth Network File System:**

Users rarely consider running network file systems over slow or wide-area networks, as the performance would be unacceptable and the bandwidth consumption too high. Nonetheless, efficient remote file access would often be desirable over such networks particularly when high latency makes remote login sessions unresponsive. Rather than run interactive programs such as editors remotely, users could run the programs locally and manipulate remote files through the file system. To do so, however, would require a network file system that consumes less bandwidth than most current file systems.

**3.5 Avoiding the Disk Bottleneck in the Data Domain Deduplication File System:**

Disk-based De-duplication storage has emerged as the new-generation storage system for enterprise data protection to replace tape libraries. De-duplication removes redundant data segments to compress data into a highly compact form and makes it economical to store backups on disk instead of tape. A crucial requirement for enterprise data protection is high throughput, typically over 100 MB/sec, which enables backups to complete quickly. A significant challenge is to identify and eliminate duplicate data segments at this rate on a low-cost system that cannot afford enough RAM to store an index of the stored segments and may be forced to access an on-disk index for every input segment. This paper describes three techniques employed in the production Data Domain De-duplication file system to relieve the disk bottleneck. These techniques include:

(1) The Summary Vector, a compact in-memory data structure for identifying new segments.

(2) Stream-Informed Segment Layout, a data layout method to improve on-disk locality for sequentially accessed segments.

(3) Locality Preserved Caching, which maintains the locality of the fingerprints of duplicate segments to achieve high cache hit ratios.

Together, they can remove 99% of the disk accesses for De-duplication of real world workloads. These techniques enable a modern two-socket dual-core system to run at 90% CPU utilization with only one shelf of 15 disks and achieve 100 MB/sec for single-stream throughput and 210 MB/sec for multi-stream throughput.

**3.6 Building a High-Performance Deduplication System:**

Modern De-duplication has become quite effective at eliminating duplicates in data, thus multiplying the effective capacity of disk-based backup systems, and enabling them as realistic tape replacements. Despite these improvements, single-node raw capacity is still mostly limited to tens or a few hundreds of terabytes, forcing users to resort to complex and costly multi-node systems, which usually only allow them to scale to single digit petabytes. As the opportunities for De-duplication efficiency optimizations become scarce, we are challenged with the task of designing De-duplication systems that will effectively address the capacity, throughput, management and energy requirements of the peta scale age. In this paper present our high-performance De-duplication prototype, designed from the ground up to optimize overall single-node performance, by making the best possible use of a node’s resources, and achieve three important goals: scale to large capacity, provide good De-duplication efficiency, and near raw disk throughput. Instead of trying to improve duplicate detection algorithms, we focus on system design aspects and introduce novel mechanisms that combine with careful implementations of known system engineering techniques. In particular, we improve single node scalability by introducing progressive sampled indexing and grouped mark and sweep, and also optimize throughput by utilizing an event-driven, multi-threaded client/server interaction model. Our prototype implementation is able to scale to billions of stored objects, with high throughput, and very little or no degradation of De-duplication efficiency.

**3.7 Ceph: A Scalable, High-Performance Distributed File System**:

Ceph, a distributed file system that provides excellent performance, reliability, and scalability. Ceph maximizes the separation between data and metadata management by replacing allocation tables with a pseudo random data distribution function (CRUSH) designed for heterogeneous and dynamic clusters of unreliable object storage devices (OSDs). We leverage device intelligence by distributing data replication, failure detection and recovery to semi-autonomous OSDs running a specialized local object file system. A dynamic distributed metadata cluster provides extremely efficient metadata management and seamlessly adapts to a wide range of general purpose and scientific computing file system workloads. Performance measurements under a variety of workloads show that Ceph has excellent I/O performance and scalable metadata management, supporting more than 250,000 metadata operations per second.

**3.8 PVFS: A Parallel File System for Linux Clusters:**

As Linux clusters have matured as platforms for low cost, high-performance parallel computing, software packages to provide many key services have emerged, especially in areas such as message passing and networking. One area devoid of support, however are critical for high performance I/O on such clusters. We have developed a parallel file system for Linux clusters, called the Parallel Virtual File System (PVFS). PVFS is intended both as a high-performance parallel file system that anyone can download and use and as a tool for pursuing further research in parallel I/O and parallel file systems for Linux clusters. PVFS and present performance results on the Chiba City cluster at Argonne. The provide performance results for a workload of concurrent reads and writes for various numbers of compute nodes, I/O nodes, and I/O request sizes. We also present performance results for MPI-IO on PVFS. We compare the I/O performance when using a Myrinet network versus a fast ethernet network for I/O-related communication in PVFS. We obtained read and write bandwidths as high as 700 Mbytes/sec with Myrinet and 225 Mbytes/sec with fast ethernet.

**PROPOSED METHODOLOGY**

The De-duplication method best suited to protect data in cloud. This process De-duplicates data both across backups and within backups and does not require any knowledge of the backup data format. The job can be system allocation can be performed for the batch jobs with the sequence of job allocation. And the content similarity is used for the de-duplications process and filtering the De-duplication content. In the time interval, the job can be finished with the effective resources then allocation can be in the order sequences. The included automates filtering, to help an analyst in cloud with similar content by designating of Data duplication can be easily removed by the content similarity algorithm. The workloads can be categorized as per the order of the job work load can be assigned. The scheduling can be maintained as per the sequence of the job within the time interval the particular job can be executed.

**4.1 User Registration and Cloud Access:**

Access users only to have authentication process before registration, Authentication process is always occurred prior to mobility management process included location registrations and service delivery, and it also ensures network resources are accessed by authorized clients and prevents resources from any illegal client or damage. Before the registration of cloud services to ensure whether the client is an authenticated or not to access cloud server. We can ensure the information stored in the cloud is used judiciously by the responsible stakeholders as per the service level agreements.

**4.2 Indexing the Cloud Data:**

The based on requirements to prepare the dataset in avoid de-duplication content. Indexing is nothing but consists of structured and unstructured format. Unstructured format is an unarranged format. Sparse Indexing is based on the reference format and capturing the repeated words queries. Indexing converts the unarranged format into structured arranged format. This may be avoid the problem of delay during searching. Sparse Indexing are used to quickly locate data without having to search every database based on the queries is accessed.

**4.3 Finding & Avoiding Similarity:**

Content similarity detection is typically performed by means of De-duplication, which is broadly classified into static and content defined. Static approaches split the input data in to equally sized chunks, which are then compared among each other. In order to identify and eliminate duplicates. While simple and fast, static approaches suffer from misalignment issues (i.e insertions or deletions lead to the impossibility to detect duplicates).Comparison phase quantifies the degree of similarity between indexing pairs belonging to the same data. And blocking the De-duplication chunks using novel techniques. Novel technique strategy aimed at reducing the user labeling effort in large scale De-duplication tasks.

**4.4 Allocating the Workloads Job Management Based on Content Similarity:**

The resource can be allocated based on the dependencies of the each job. Based on the dependencies the resource can be allocated. The Content Similarity is a statistical methods to categorize a De-duplication and blocked the adjustable levels of granularity. We cultrate the data set, so that it contains only one representation of each sequence for quantifying and comparative studies. The included automates filtering, to help an analyst in cloud with similar content by designating of Data duplication can be easily removed by the content similarity algorithm.

**4.5 Experimental Analysis and Results:**

Implementation is often used in the tech world to describe the interactions of elements in programming languages. In Java, where the word is frequently used, to implement is to recognize and use an element of code or a programming resource that is written into the program. One aspect of implementing an interface that can cause confusion is the requirement that to implement an interface, a class must implement all of the methods of that interface. This can lead to error messages due to insufficient implementation of methods. In general, the syntactical requirements of implementation and other tasks can be a burden for developers, and mastering this is part of becoming an in-depth user.

Checksum Algorithm

Files Properties

File Name

File Size

Login

Registration

User

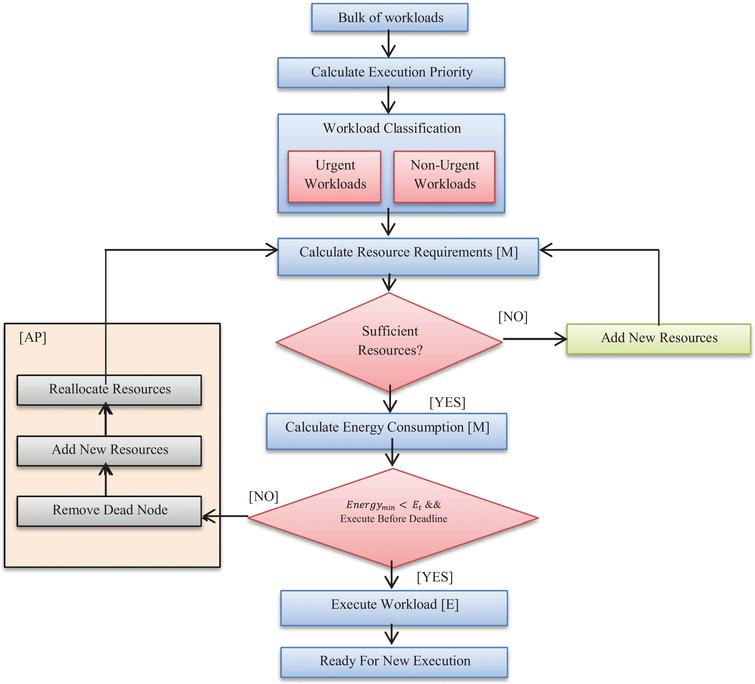
Load Files for Content Matching

File Path

File Size

Avoid Chunks and Duplicate Content

**IMPLEMENTATION OF PROPOSED METHODOLOGY**



**5.1 Indexing**

In our model, prior to searching for a document, an encrypted index of the corpus must be generated by the trusted client-side server. The index is then encrypted and sent to the untrusted cloud server. Searching takes place by running SQL queries over the encrypted index. For illustration purposes, a small representation of such an index based on unencrypted form in and the corresponding partially encrypted versio. Each row in the encrypted index table corresponds to one document \_ 2 corpus. Each row contain two columns: an arbitrarily assigned unique document id (ID) and a specialized data structure that contains truncated symmetric-key encrypted keywords associated with encrypted versions of the keyword’s location in \_ (Word Vectors). In addition, this data string contains an offset that is used to map the truncated encrypted keyword with its full version (stored on the trust client-side server).

**5.2 Keyword Truncation**

A main attack point on the proposed scheme thus far is that it is highly susceptible to statistical frequency analysis attacks. If each keyword were encrypted individually using a deterministic encryption method, a nosey cloud provider could compare the encrypted index with a language probability table to estimate which words map to which encrypted words. To combat this problem we truncate the encrypted words to a predefined number of bits.

**5.3 Load balancing in the Cloud**

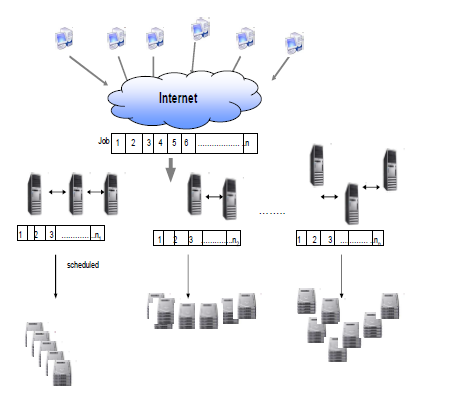
Load balancing is a process of reassigning the total load to the individual nodes of the collective system to make resource utilization effective and to improve the response time of the job, simultaneously removing a condition in which some of the nodes are over loaded while some others are under loaded. Server load balancing addresses several requirements that are becoming increasingly important in networks:

• Increased scalability

• High performance

• High availability and disaster recovery

First the requests or job coming from the user side are stored in a job pool or the central



**5.4 Job assignment using Lexi – search approach**

After the partitioning the job queue, scheduling is done by the Lexi – search approach in the local middleware. Task scheduling is a kind of transportation problem. There is a set J {1,2,3,……..,n} of n jobs and a set I {1,2,3,……..,m} of m available resources in which jobs will be assigned for execution. This has some restrictions that one job will be allocated to only one resource and each resource has to do only one job. All the resources start doing the jobs simultaneously but a resource doing more than one job has to do them one after the other. Let Pi,j be the probabilistic measurement or credit of a job to be assigned to a particular resource.

This Pi,j is calculated as follows:

a i,j X e i,j

P i,j =

Σ a i,j X Σ e i,j

a i,j is the availability of a resource j to be free after executing task i . a i,j is also computed as the sum of arrival time of a task and the execution time of task i to be executed on j resource. a i,j = arrival time of task i + e i,j [ **e i,j** = execution time of task i on resource j**]**

The aim is to find that assignments of the resources to the jobs for which the corresponding time of completion of all jobs is the minimum. If the decision variable **x i,j** (i,j) є *I* X *J* takes the value 1 when the i th resource execute the j th job and 0 otherwise. So the mathematical formulation of the above problem is :



Since number of resource is less than the number of jobs, we call this problem an *Imbalanced* *Time Minimizing Assignment Problem (****ITMAP****).* Clearly it always has a feasible solution. An assignment, X = *{xij},* is one which satisfies

(1)and (3), and *T(X)* is the corresponding time of completion of the jobs. An assignment is called a feasible assignment if (2) is also satisfied. This assignment **xi, j** can also be represented by a row vector as follows:

w = (il, i2,..., in), ……………………….**(4)**

where, all ij's are not distinct, clearly [w[=I,J[=n]]]. Thus, the assignment represented by (4) implies that the jth job is done by the ij th person, j = 1,2,..., n. Each assignment in its vector form (4) can be thought of as a word, w, of length n, with letters ij's from the set I. Let W = {w} be the set of all feasible words of length n. Then for a feasible word, say w, given by (4), the corresponding feasible assignment Xw = { **xi, j** w } is given by :

Xw i,j = 1, j=1,2,…………….,n.

Xwi,j = 0, (*i*, *j*)Î*IXJ* -{(i,j) : *j*Î *J* }

The value of *T(*Xw*)* for this feasible assignment corresponding to w is**

For this purpose we have defined:

**5.4.1 Alphabet matrix:**

It is an m × n matrix formed by the positions of the elements of the given m × n matrix {**P i,j** }of credit. The jth column *of AB* consists of the positions of the entries in the jth column of the matrix { **P i,j** } when they are arranged in the non-decreasing order of their values. Let *ab(y,j)* stands for the y th entry in the jth column *of AB.* Therefore, *ab(1,j)* corresponds to the smallest entry in the jth column of the matrix *{* **P i,j** *}* that is, all the words in W can be systematically generated by considering the elements of the jth column of *AB* in the jth position (j -- 1,2,..., n) of a word, i.e., *ij E {ab(q,j), q = 1,2,..., m}.*



**5.4.2 Partial Word (Pw):**

Pw = (il, i2,..., it), *r* ≤ *n,* represents a partial word. A partial assignment corresponding to it consists of assigning the jth job to the ijth resource, j = 1,2,..., r (jobs r + 1, r + 2 , . . . , n are still to be assigned). Pw defines a block of words each of which has first r letters as il, i2 . . . . , it. In this sense Pw is called the leader of this block of words. If a partial word is such that I/[ > n - IPwl, then clearly this partial word cannot contain a feasible word, where I[[ is the index set of unassigned persons. Such a partial word is called an *infeasible partial word.* On the other hand, | I | ≤ n - |Pw| then Pw is called a *feasible* *partial word.* Contribution to the objective function T(.) by the partial assignment, say XPw, corresponding to Pw is given by

**

Clearly, for a word ‘w’ whose leader is Pw , we have *T(*Xw*) ≥ T(*Xpw*)*

**5.5 N Notation:**

To starting upper bound on the value of the objective function T(.).

Js J-{j1,j2,......js-1} (Clearly J1=J) \_| I | index set of unsigned person.

Tu updated upper bound on the value o the objective function T(.).Φ empty set.

**5.6 Upper bound and objective function T(.) evaluation :**

For each iÎI , find min(Pi,j = Piji) (say) and set Xiji =1, iÎI ,Then each of the m person is assigned to unique job in the set (j1,j2,....jm). For allocation of the remaing jobs in Jm+1, proceed as follows :



**(Say)**

Then allocate job Jm+k to resource im+k ,k=1,2,…(n-m).

‘T0’ will be given by

T0 =max(*j J* Î Σ (Pij : Xij =1) )

This heusistic will provide the starting upper bound on the value of T(.) quite close to its

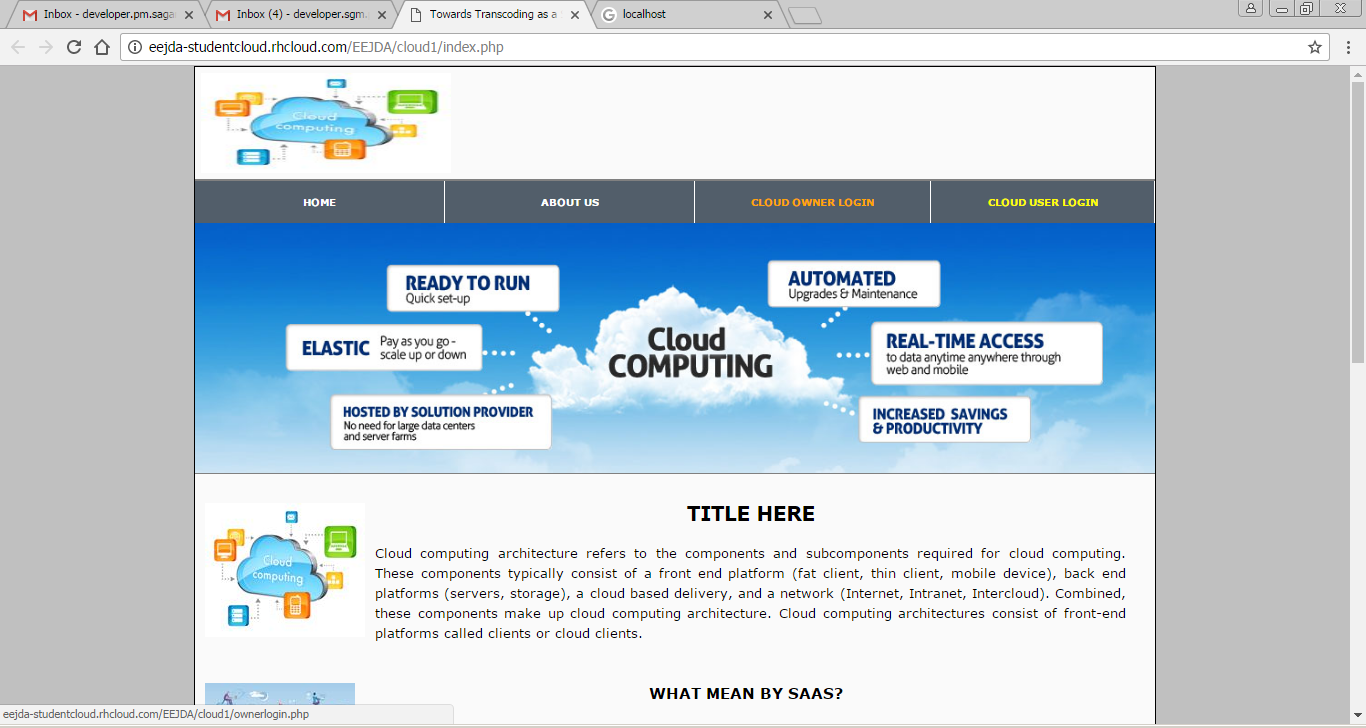
optima value. A feasible assignment then obtained is



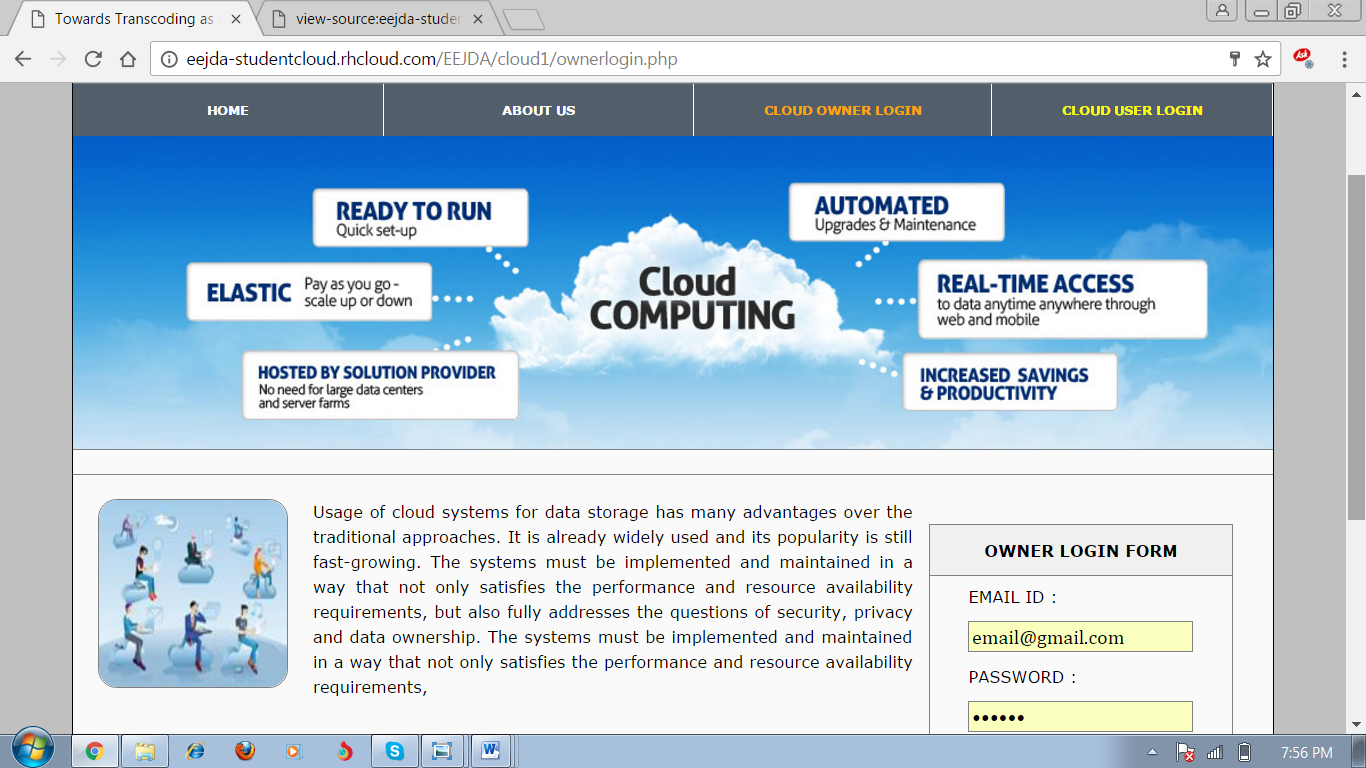
Let the feasible word corresponding to this feasible assignment be w = (ab(y1,1), ab(y2,1),........ ab(yn,1)). Therefore the above feasible assignment can then be given as



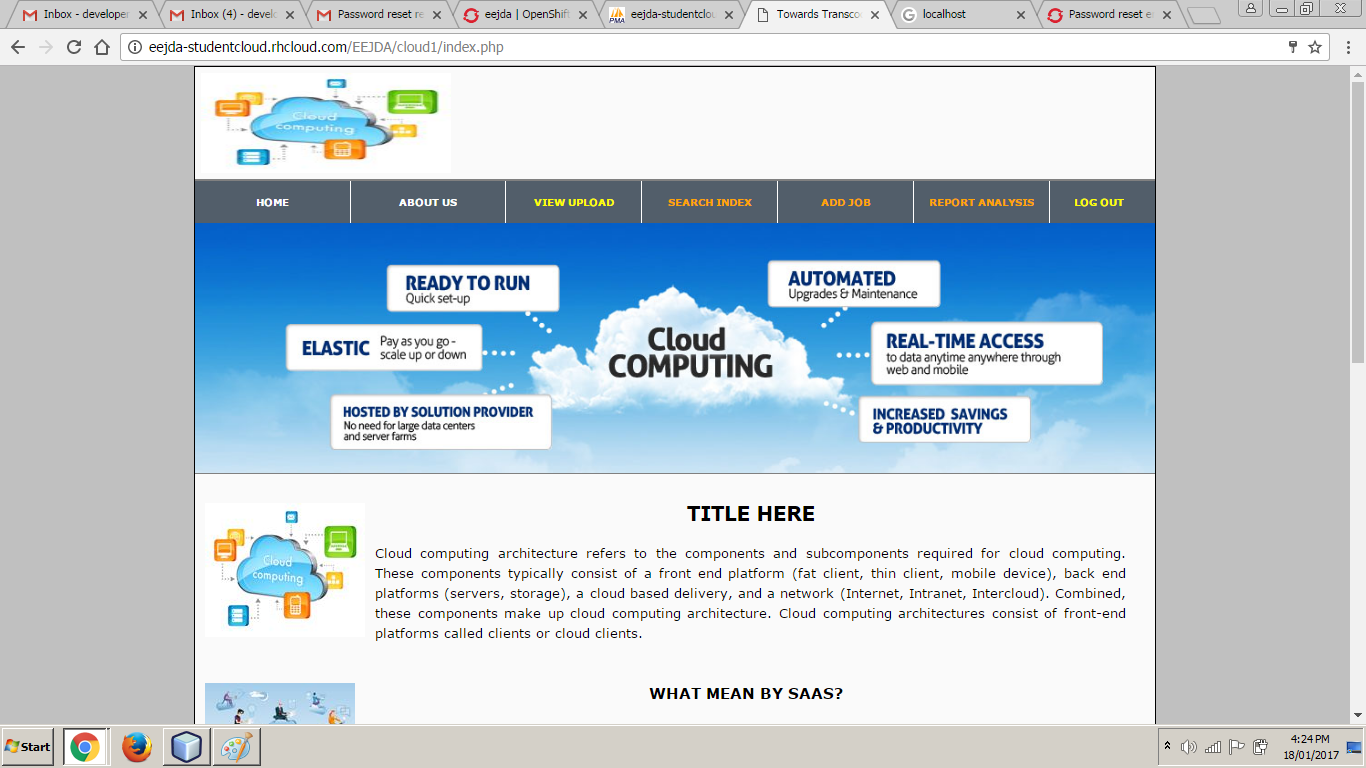
**RESULT ANALYSIS AND DISCUSSION**

******

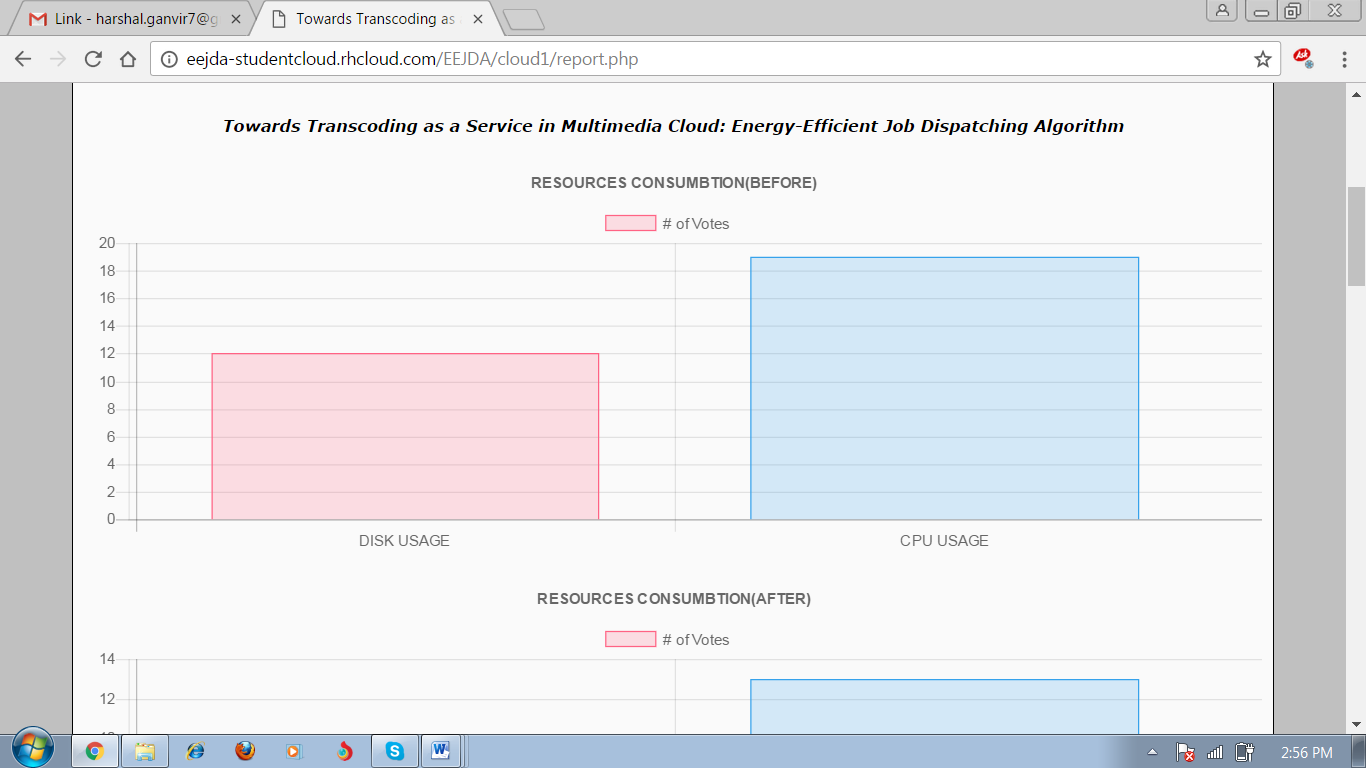
**Figure 6.1 Cloud Owner Side**



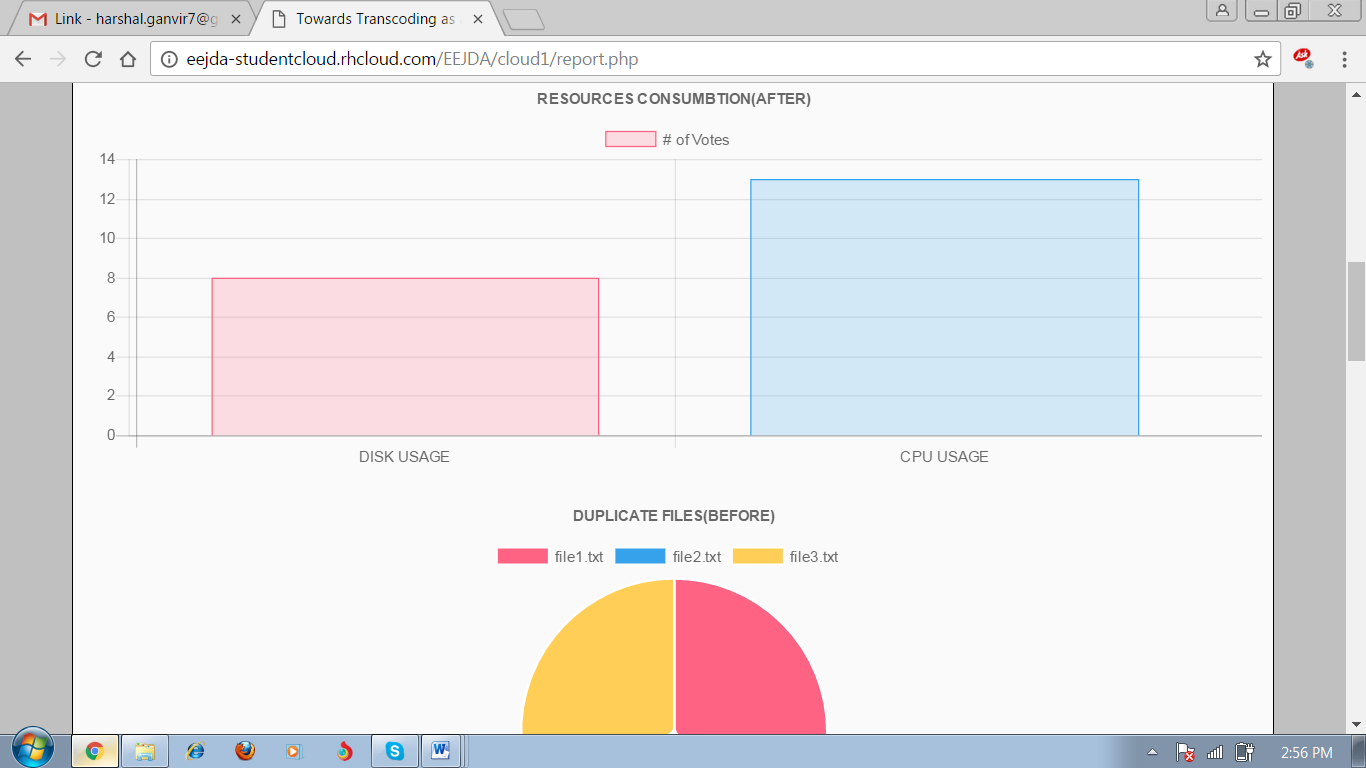
**Figure 6.2 Cloud Owner Login**



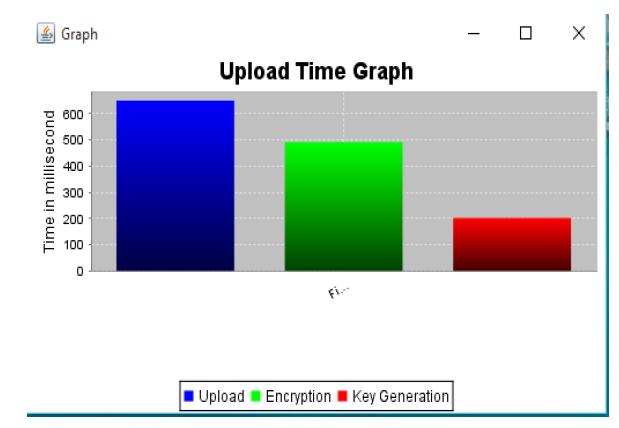
**Figure 6.3 Homepage of Cloud owner**



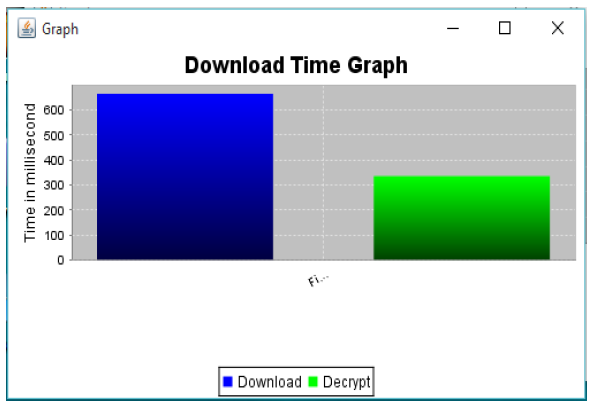
**Figure 6.4 Report 1**



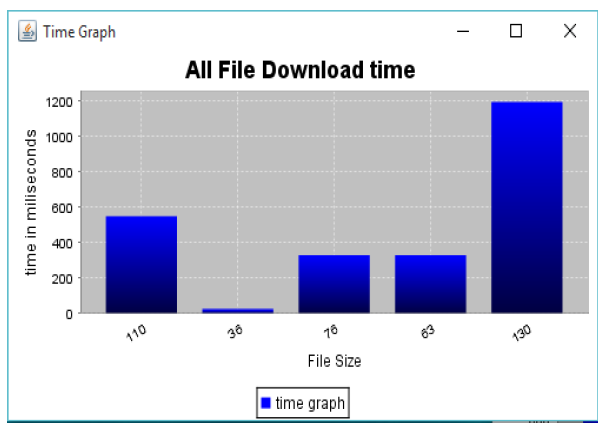
**Figure 6.5 Report 2**



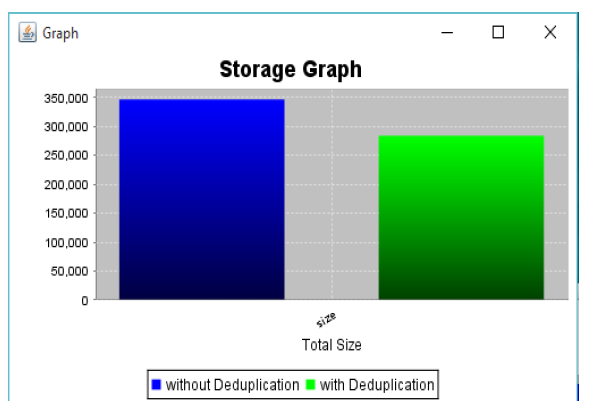
**Figure 6.6 Report 3**



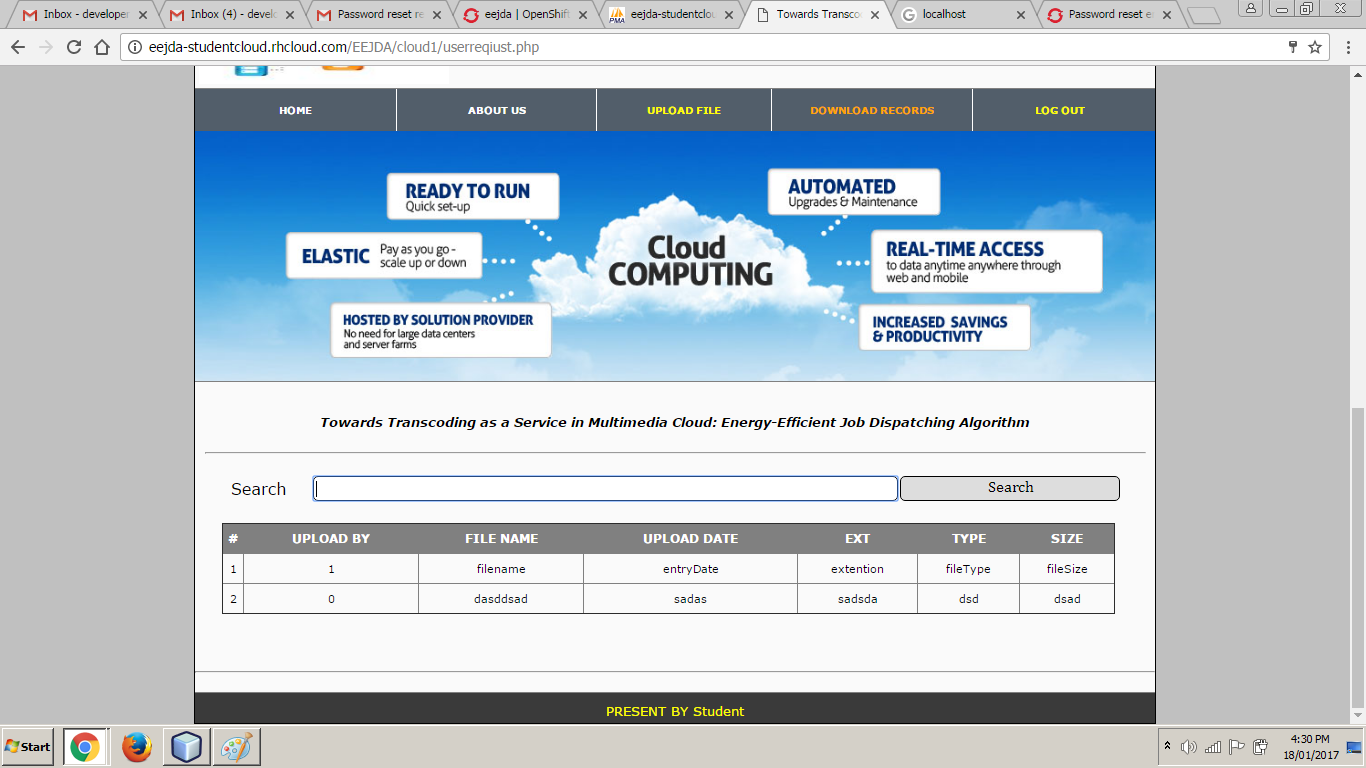
**Figure 6.7 Report 4**



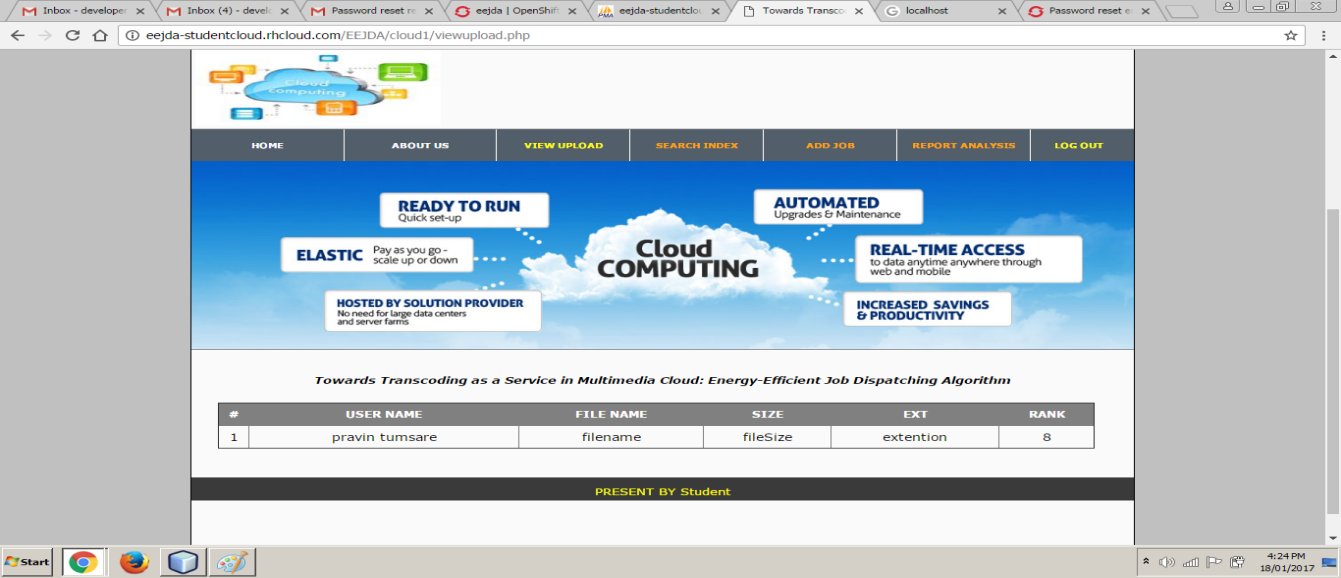
**Figure 6.8 Report 4**

****

**Figure 6.9 Report 5**



**Figure 6.8 User Homepage**

**Figure 6.9 View Upload**

**CONCLUSION & FUTURE SCOPE**

We explored the impact of many factors on the effectiveness of De-duplication. We showed that package installation and language localization have little impact on De-duplication ratio. However, factors such as the base operating system. The Linux distribution can have a major impact on De-duplication effectiveness. Thus, we recommend that hosting centers suggest “preferred” operating system distributions for their users to ensure maximal space savings. If this preference is followed subsequent user activity will have little impact on De-duplication effectiveness. We found that, in general, 40% is approximately the highest De-duplication ratio if no obviously similar VMs are involved. In future work we plan to explore several promising avenues. First, we did not explore what happens when the groups are not operating simultaneously and/or access common content at different times. How to leverage and anticipate such De-synchronizations can provide further potential for improvement. Second, our approach treats all chunks individually, both in terms of advertisements and exchanges. Thus, it would be interesting to understand and exploit correlations between chunks.

**REFERENCES**

[1] Cisco Visual Networking Index: Forecast and Methodology, 2012–2017, 2013.

[2] A. Vetro and C. W. Chen, “Rate-reduction transcoding design for wireless video streaming,” in *Proc. Int. Conf. Image Process.*, 2002, vol. 1, pp. I-29–I-32.

[3] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “A view of cloud computing,” *Commun. ACM*, vol. 53, no. 4, pp. 50–58, Apr. 2010.

[4] A. Garcia, H. Kalva, and B. Furht, “A study of transcoding on cloud environments for video content delivery,” in *Proc. ACM Multim. Worksh.* *Mobile Cloud Media Comput.*, 2010, pp. 13–18.

[5] Z. Li, Y. Huang, G. Liu, F. Wang, Z.-L. Zhang, and Y. Dai, “Cloud transcoder: Bridging the format and resolution gap between Internet videos and mobile devices,” in *Proc. 22nd Int. Workshop Netw. Oper. Syst.* *Support Digit. Audio Video*, 2012, pp.33–38.

[6] M. J. Neely, “Stochastic network optimization with application to communication

and queueing systems,” *Synthesis Lectures Commun. Netw.*, vol. 3, no. 1, pp. 1–211, 2010.

[7] J. Guo and L. N. Bhuyan, “Load balancing in a cluster-based web server for multimedia applications,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 17, no. 11, pp. 1321–1334, Nov. 2006.

[8] D. Seo, J. Kim, and I. Jung, “Load distribution algorithm based on transcoding time estimation for distributed transcoding servers,” in *Proc.* *ICISA*, 2010, pp. 1–8.

[9] A. Garcia and H. Kalva, “Cloud transcoding for mobile video content delivery,” in *Proc. IEEE ICCE*, 2011, pp. 379–380.

[10] S. Ko, S. Park, and H. Han, “Design analysis for real-time video transcoding on cloud systems,” in *Proc. 28th Annu. ACM Symp. Appl. Comput.*, 2013, pp. 1610–1615.

[11] Malathi.P “A Survey On Reducing Energy Sprawl in Cloud Computing.” Advanced Computational Intelligence: An International Journal (ACII), Vol.2, No.3, July 2015.

[12] Arindam Banerjee, Prateek Agrawal and N. Ch. S. N. Iyengar ”Energy Efficiency Model For Cloud Computing ” International Journal of Energy, Information and Communications Vol.4, Issue 6 (2013).

[13] Jayant Baliga, Robert W. A. Ayre, Kerry Hinton, and Rodney S.Tucker “Green Cloud Computing:Balancing Energy in Processing, Storage, and Transport.” 2010 IEEE 99, No. 1, January 2011 | Proceedings of the IEEE.

[14] Anton Beloglazov and Rajkumar Buyya (2010) “Energy Efficient Allocation of Virtual Machines in Cloud Data Centers “2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing.

[15] Yogesh Sharma, Bahman Javadi, Weisheng Si (2015) “On the Reliability and Energy Efficiency in Cloud Computing” Proceedings of the 13th Australasian Symposium on Parallel and Distributed Computing (AusPDC 2015), Sydney, Australia, 27 - 30 January 2015.

**ANNEXURE**

**CODING**

**Coding**

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

UI window = new UI();

window.frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the application.

\*/

public UI() {

initialize();

}

/\*\*

\* Initialize the contents of the frame.

\*/

private void initialize() {

frame = new JFrame()

frame.setBounds(100, 100, 692, 445);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

JTabbedPanetabbedPane = new JTabbedPane(JTabbedPane.TOP);

frame.getContentPane().add(tabbedPane, BorderLayout.CENTER);

JPanel panel = new JPanel();

tabbedPane.addTab("Preprocessing", null, panel, null);

GridBagLayoutgbl\_panel = new GridBagLayout();

gbl\_panel.columnWidths = new int[] { 0, 0 };

gbl\_panel.rowHeights = new int[] { 0, 0 };

gbl\_panel.columnWeights = new double[] { 1.0, Double.MIN\_VALUE };

gbl\_panel.rowWeights = new double[] { 1.0, Double.MIN\_VALUE };

panel.setLayout(gbl\_panel);

JPanel panel\_1 = new JPanel();

GridBagConstraints gbc\_panel\_1 = new GridBagConstraints();

gbc\_panel\_1.fill = GridBagConstraints.BOTH;

gbc\_panel\_1.gridx = 0;

gbc\_panel\_1.gridy = 0;

panel.add(panel\_1, gbc\_panel\_1);

GridBagLayout gbl\_panel\_1 = new GridBagLayout();

gbl\_panel\_1.columnWidths = new int[] { 0, 0, 0, 0 };

gbl\_panel\_1.rowHeights = new int[] { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };

gbl\_panel\_1.columnWeights = new double[] { 0.0, 1.0, 0.0,

Double.MIN\_VALUE };

gbl\_panel\_1.rowWeights = new double[] { 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,

0.0, 0.0, 0.0, Double.MIN\_VALUE };

panel\_1.setLayout(gbl\_panel\_1);

JLabellblSelectedFile = new JLabel("Data File for preprocessing: ");

GridBagConstraintsgbc\_lblSelectedFile = new GridBagConstraints();

gbc\_lblSelectedFile.insets = new Insets(0, 0, 5, 5);

gbc\_lblSelectedFile.anchor = GridBagConstraints.EAST;

gbc\_lblSelectedFile.gridx = 0;

gbc\_lblSelectedFile.gridy = 0;

panel\_1.add(lblSelectedFile, gbc\_lblSelectedFile);

dataFileForPreprocessingTextField = new JTextField();

dataFileForPreprocessingTextField.setEditable(false);

GridBagConstraintsgbc\_dataFileForPreprocessingTextField = new GridBagConstraints();

gbc\_dataFileForPreprocessingTextField.insets = new Insets(0, 0, 5, 5);

gbc\_dataFileForPreprocessingTextField.fill = GridBagConstraints.HORIZONTAL;

gbc\_dataFileForPreprocessingTextField.gridx = 1;

gbc\_dataFileForPreprocessingTextField.gridy = 0;

panel\_1.add(dataFileForPreprocessingTextField,

gbc\_dataFileForPreprocessingTextField);

dataFileForPreprocessingTextField.setColumns(10);

browsePreProcessingButton = new JButton("Browse");

browsePreProcessingButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEventae) {

JFileChooserfileChooser = new JFileChooser();

if (JFileChooser.APPROVE\_OPTION == fileChooser

.showOpenDialog(UI.this.frame)) {

selectedDataFile = fileChooser.getSelectedFile();

dataFileForPreprocessingTextField.setText(selectedDataFile

.getPath());

startPreProcessing.setEnabled(true);

}

System.out.println(selectedDataFile);

}

});

GridBagConstraintsgbc\_btnBrowse = new GridBagConstraints();

gbc\_btnBrowse.insets = new Insets(0, 0, 5, 0);

gbc\_btnBrowse.gridx = 2;

gbc\_btnBrowse.gridy = 0;

panel\_1.add(browsePreProcessingButton, gbc\_btnBrowse);

startPreProcessing = new Button("Start Preprocessing");

startPreProcessing.setEnabled(false);

startPreProcessing.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

startPreProcessing();

}

});

GridBagConstraintsgbc\_startPreProcessing = new GridBagConstraints();

gbc\_startPreProcessing.insets = new Insets(0, 0, 5, 5);

gbc\_startPreProcessing.gridx = 0;

gbc\_startPreProcessing.gridy = 1;

panel\_1.add(startPreProcessing, gbc\_startPreProcessing);

preProcessingProgressBar = new JProgressBar();

long time = (System.currentTimeMillis() - preprocessingStartTime) / 2000;

preProcessingProgressBar.setIndeterminate(false);

preprocessingStatusLable.setText("Completed hybrid preprocessing in "

+ time + " second(s)");

browsePreProcessingButton.setEnabled(true);

startPreProcessing.setEnabled(true);

return "done";

}

@Override

protected void done() {

super.done();

System.out.println("Done");

try {

System.out.println(get());

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (ExecutionException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

};

swingWorker.execute();

}

}

**PUBLICATION**

**PUBLICATION**

1. Harshal P. Ganvir, Prof. Pravin G.Kulurkar,”Review Paper on Outlier Detection Techniques”,International Journal of Computer Systems (ISSN: 2394-1065), Volume 02– Issue 11, November 2015
2. Harshal P. Ganvir, Prof.Pravin G.Kulurkar,”Efficient approach for Detection of Outlier using Hybrid Algorithm”,International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169 ,Volume: 4 Issue: 5 ,May 2016
3. Harshal P. Ganvir, Prof.Pravin G.Kulurkar,”Detection of Outlier in Uncategorical Dataset using Hybrid Algorithm”, International Journal Of Engineering And Computer Science ISSN: 2319-7242 Volume 5 Issues 6 June 2016, Page No. 17084-17087
4. Harshal P. Ganvir, ”Efficient approach for Detection of Outlier using Hybrid algorithm”, International Conference on Science & Technology for Sustainable Development(ICSTSD) ”,May 2016