Impact Factor (SJIF): 4.542

International Journal of Advance Research in Engineering, Science & Technology

e-ISSN: 2393-9877, p-ISSN: 2394-2444

Volume 4, Issue 12, December-2017

An Efficient Way to Detect the Duplicate data in Clouds by TRE Mechanism

Naushin Parveen N Sheikh[1], Sagar M. Bhakre[2]

[1]M. Tech scholar, Department of Computer Science & Engineering, Ballarpur Institute of Technology, Ballarpur, Maharashtra, India

[2]Assistant Professor, Department of Computer Science & Engineering, Ballarpur Institute of Technology, Ballarpur, Maharashtra, India

Abstract —In this paper, we display PACK (Predictive ACKs), a novel end-to-end movement repetition disposal (TRE) framework, intended for distributed computing clients. Cloud-based TRE needs to apply a reasonable utilization of cloud assets so the data transfer capacity cost decrease joined with the extra cost of TRE calculation and capacity would be streamlined. PACK's principle preference is its capacity of offloading the cloud-server TRE push to end customers, in this way limiting the preparing costs prompted by the TRE calculation. Dissimilar to past arrangements, PACK does not require the server to ceaselessly keep up customers' status. This makes PACK exceptionally appropriate for unavoidable calculation situations that consolidate customer versatility and server relocation to keep up cloud flexibility. PACK depends on a novel TRE method, which enables the customer to utilize recently got lumps to recognize beforehand got piece chains, which thus can be utilized as solid indicators to future transmitted pieces. We introduce a completely utilitarian PACK execution, straightforward to all TCP-based applications and system gadgets. At long last, we break down PACK benefits for cloud clients, utilizing movement follows from different sources.

Keywords- Caching, Cloud Computing, Network Optimization, Traffic Redundancy Elimination

I. INTRODUCTION

Distributed computing offers its clients a prudent and helpful pay-as-you-go benefit demonstrate, referred to likewise as utilization based estimating. Cloud customers1 pay just for the genuine utilization of figuring assets, stockpiling, and transmission capacity, as indicated by their evolving needs, using the cloud's versatile and flexible computational abilities. Specifically, information exchange costs (i.e., transmission capacity) is an imperative issue when endeavoring to limit costs. Subsequently, cloud clients, applying a sensible utilization of the cloud's assets, are roused to utilize different activity decrease systems, specifically movement excess disposal (TRE), for diminishing data transfer capacity costs. Movement excess stems from normal end-clients' exercises, for example, over and over getting to, downloading, transferring (i.e., reinforcement), appropriating, and altering the same or comparable data things (records, information, Web, and video). TRE is utilized to dispense with the transmission of excess substance and, in this manner, to fundamentally decrease the system cost. In generally normal

TRE arrangements, both the sender and the beneficiary inspect and look at marks of information lumps, parsed by the information content, preceding their transmission. At the point when repetitive lumps are distinguished, the sender replaces the transmission of each excess piece with its solid mark. Business TRE arrangements are famous at big business organizes, and include the sending of at least two restrictive convention, state synchronized center boxes at both the intranet section purposes of server farms and branch workplaces, taking out dreary activity between them (e.g., Cisco, Riverbed, Quantum, Juniper, Blue Coat, Expand Networks, and F5). While exclusive center boxes are mainstream point arrangements inside endeavors, they are not as alluring in a cloud domain. Cloud suppliers can't profit by an innovation whose objective is to diminish client transmission capacity bills, and in this manner are not prone to put resources into one. The ascent of "on-request" work spaces, meeting rooms, and work-from-home arrangements segregates the laborers from their workplaces. In such a dynamic workplace, settled point arrangements that require a customer side and a server-side center box match end up noticeably inadequate. Then again, cloud-side flexibility inspires work dissemination among servers and movement among server farms. Along these lines, it is usually concurred that a widespread, programming based, end-to-end TRE is critical in the present unavoidable condition. This empowers the utilization of a standard convention stack and makes a TRE inside end-to-end secured movement (e.g., SSL) conceivable. Current end-to-end TRE arrangements are sender-based. For the situation where the cloud server is the sender, these arrangements require that the server persistently keep up customers' status. We appear here that cloud flexibility requires another TRE arrangement. In the first place, cloud stack adjusting and control enhancements may prompt a server-side process and information movement condition, in which TRE arrangements that require full synchronization between the server and the customer are difficult to finish or may lose effectiveness because of lost synchronization. Second, the notoriety of rich media that devour high data transfer capacity rouses content dispersion arrange (CDN) arrangements, in which the administration point for settled and versatile clients may change progressively as indicated by the relative administration point areas and burdens. Besides, if a conclusion to-end arrangement is utilized, its extra computational and capacity costs at the cloud side ought to be weighed against its transfer speed sparing increases. Plainly, a TRE arrangement that puts the vast majority of its computational exertion on the cloud side2may swing to be less savvy than the one that use the consolidated customer side abilities. Given a conclusion to-end arrangement, we have found through our investigations that sender-based end-to-end TRE arrangements add a significant load to the servers, which may destroy the cloud cost sparing tended to by the TRE in any case. Our examinations additionally demonstrate that present end-to-end arrangements likewise experience the ill effects of the necessity to keep up end-to-end synchronization that may bring about debased TRE proficiency.

II. LITERATURE REVIEW

Ankita Kotalwar , Dr. Sadhana Chidrawar "Expectation Analysis for Cloud Bandwidth and Cost Reduction" International Journal of Advance Research in Computer Science and Management Studies Volume 3, Issue 9, September 2015.

In this paper, we introduce PACK (Predictive ACKs), a novel end-to-end activity repetition disposal (TRE) framework, intended for distributed computing clients. Cloud based TRE needs to apply a sensible utilization of cloud assets so the data transmission cost diminishment joined with the extra cost of TRE calculation and capacity would be improved. PACK's principle favorable position is its capacity of offloading the cloud-server TRE push to end-customers, hence limiting the handling costs initiated by the TRE calculation. Not at all like past arrangements, PACK does not require the server to constantly keep up customers' status. This makes PACK extremely reasonable for inescapable calculation situations that join customer portability and server movement to keep up cloud versatility. PACK depends on a novel TRE procedure, which enables the customer to utilize recently got lumps to recognize already got piece chains, which thus can be utilized as solid indicators to future transmitted lumps. We display a completely practical PACK usage, straightforward to all TCP-based applications and net-work gadgets. At last, we break down PACK benefits for cloud clients, utilizing activity follows from different sources.

T. Navin Kumar, K. V. Manoj Kumar, M. R. Srinivasulu and Mrs. Girija "Cost Reduction Technique Using Cloud Computing" International Journal of Technical Research and Application Volume 3, Issue 2 (Mar-Apr 2015)

In this paper, we are utilizing distributed computing for wiping out movement excess and diminishing expense for an advantage of cloud clients. Here, we present the method as transmission capacity expectation through synchronization over client and server. The client's data transmission were anticipated by the server and continue with the affirmation procedure, server of cloud give the transfer speed to the end client of which client needs. From this, cloud giving distinctive transmission capacity to various client which consequently dispensing with end-to-end activity excess and cost useful for an each client getting to cloud. So effortlessly cloud client acquires their undertaking and pay just for the use in the cloud.

III. IMPLEMENTATION OF PROPOSED METHODOLOGY

A. Modules

There are 6 types of modules in this project as follows:-

1. Register Module
2. Login Module
3. Upload Module
4. Client Module
5. Query Details Module
6. Pack Module

B. Module Description

1 Register Module:

Client has to register in the registration module by providing username, password and email and contact details.

2 Login Module:

After registering client logins into his account by using the username and password which was given at the time of the registration.

3 Upload Module:

This module is used to choose a file and upload a file.

4 Client Module:

The functionalities of client are search category, search query.

5 Query Details Module:

This module is used to display the queries.

6 Pack Module:

This module consists of registration, login and upload file details.

C. Introduction of Technologies Used

About Java

Initially the language was called as “oak” but it was renamed as “java” in 1995.The primary motivation of this language was the need for a platform-independent (i.e. architecture neutral) language that could be used to create software to be embedded in various consumer electronic devices.

- Java is a programmer’s language
- Java is cohesive and consistent
- Except for those constraint imposed by the Internet environment. Java gives the programmer, full control

Finally Java is to Internet Programming where c was to System Programming.

Java Architecture

Java design gives a versatile, strong, high performing condition for improvement. Java gives convenientce by accumulating the byte codes for the Java Virtual Machine, which is then translated on every stage by the run-time condition. Java is a dynamic framework, ready to stack code when required from a machine in a similar room or over the planet.

Assemblage of code

When you arrange the code, the Java compiler makes machine code (called byte code) for a theoretical machine called Java Virtual Machine (JVM). The JVM should execute the byte code. The JVM is made for the defeating the issue of likelihood. The code is composed and gathered for one machine and translated on all machines .This machine is called Java Virtual Machine.

Compiling and interpreting java source code.

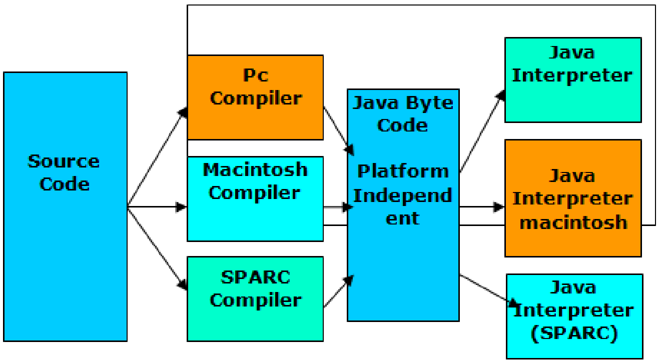


Figure 1 Compiling & Interpreting java source code

During run-time the Java interpreter tricks the byte code file into thinking that it is running on a Java Virtual Machine. In reality this could be an Intel Pentium windows 95 or sun SPARCstation running Solaris or Apple Macintosh running system and all could receive code from any computer through internet and run the Applets.

AWT and Swings

AWT

Graphical User Interface

The UI is that piece of a program that collaborates with the client of the program. GUI is a sort of UI that enables clients to interface with electronic gadgets with pictures as opposed to content orders. A class library is given by the Java programming dialect which is known as Abstract Window Toolkit (AWT) for composing graphical projects. The Abstract Window Toolkit (AWT) contains a few graphical gadgets which can be added and situated to the show region with a format administrator. As the Java programming dialect, the AWT isn't stage autonomous. AWT utilizes framework peers protest for building graphical gadgets. A typical arrangement of devices is given by the AWT to graphical UI outline. The execution of the UI components gave by the AWT is finished utilizing each stage's local GUI toolbox. One of the AWT's importance is that the look and feel of every stage can be protected.

Parts:

A graphical UI is worked of graphical components called segments. A segment is a protest having a graphical portrayal that can be shown on the screen and that can connect with the client. Parts enable the client to connect with the program and give the contribution to the program. In the AWT, all UI parts are examples of class Component or one of its subtypes. Regular segments incorporate such things as catches, scrollbars, and content fields.

Types of Components:

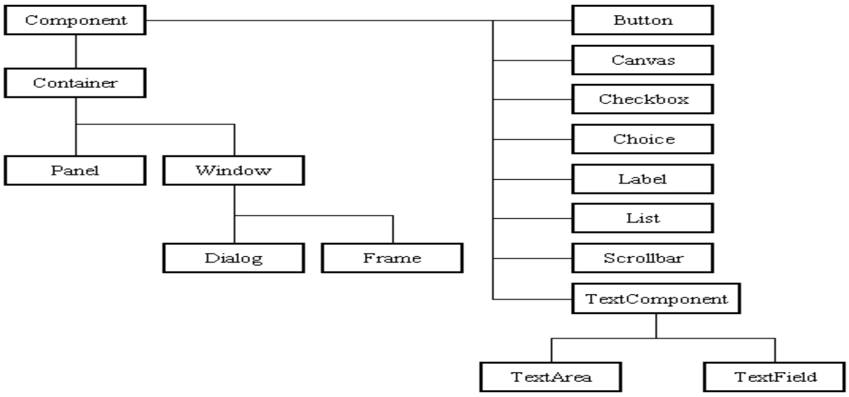


Figure 2 Types of components

Before proceeding ahead, first we need to know what containers are. After learning containers we learn all components in detail.

Containers

Components do not stand alone, but rather are found within containers. In order to make components visible, we need to add all components to the container. Containers contain and control the layout of components. In the AWT, all containers are instances of class Container or one of its subtypes. Components must fit completely within the container that contains them. For adding components to the container we will use add () method.

Types of containers

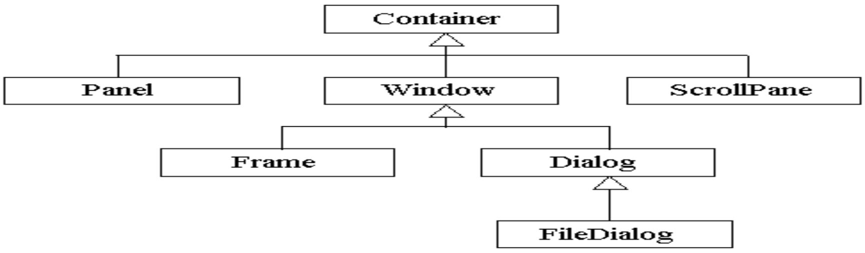


Figure 3 Types of Container

Basic GUI Logic:

The GUI application or applet is created in three steps. These are:

- Add components to Container objects to make your GUI.
- Then you need to setup event handlers for the user interaction with GUI.
- Explicitly display the GUI for application.

A new thread is started by the interpreter for user interaction when an AWT GUI is displayed. When any event is received by this new thread such as click of a mouse, pressing of key etc. then one of the event handlers is called by the new thread set up for GUI. One important point to note here is that the event handler code is executed within the thread.

Swings

About Swings

Swing is important to develop Java programs with a graphical user interface (GUI). There are many components which are used for the building of GUI in Swing. The Swing Toolkit consists of many components for the building of GUI. These components are also helpful in providing interactivity to Java applications. Following are components which are included in Swing toolkit:

- list controls
- buttons
- labels
- tree controls
- table controls

All AWT adaptable segments can be taken care of by the Java Swing. Swing toolbox contains much a greater number of segments than the basic part tool compartment. It is special to some other toolbox in the way that it underpins incorporated internationalization, a profoundly adjustable content bundle, rich fix bolster and so on. Not just this you can likewise make your own look and feel utilizing Swing other than the ones that are upheld by it. The tweaked look and feel can be made utilizing Synth which is exceptionally composed. Not to overlook that Swing likewise contains the essential UI, for example, adjustable painting, occasion dealing with, simplified and so on.

The Java Foundation Classes (JFC) which bolsters numerous more highlights essential to a GUI program involves Swing too. The highlights which are upheld by Java Foundation Classes (JFC) are the capacity to make a program that can work in various dialects, the capacity to include rich designs usefulness and so on. There are a few parts contained in Swing toolbox, for example, check boxes, catches, tables, content and so forth. Some extremely straightforward parts additionally give advanced usefulness. For example, content fields give designed content info or secret key field conduct. Besides, the record programs and exchanges can be utilized by one's need and can even be altered.

IV. EXPERIMENTAL RESULT

Our examination utilizing a wide arrangement of substance sorts demonstrates that PACK meets the normal style objectives and has clear advantages over sender - based TRE, especially once the cloud calculation esteem and buffering necessities are essential. More-finished, PACK forces additional exertion on the sender just if repetition is abused, so lessening the cloud general esteem. Two captivating future augmentations will give additional edges to the PACK thought. To start with, our execution keeps up chains by keeping for Associate in nursing piece exclusively the last found sub-sequent lump in a LRU design. A striking augmentation to the present work is that the connected science investigation of chains of pieces that will change various possibilities in each the lump arrange and the comparing expectations. The framework may allow making more than one expectation at any given moment, and it's sufficient that one among them are right to boom activity disposal. A moment promising course is that the method of operation change of the half breed sender– beneficiary approach upheld shared decisions de-rived from collector's energy or server's esteem changes.

Steps and screen shots for result

1. First run the cloud application
2. Run the client application

   2.1 Result of traffic volume and detect redundancy:

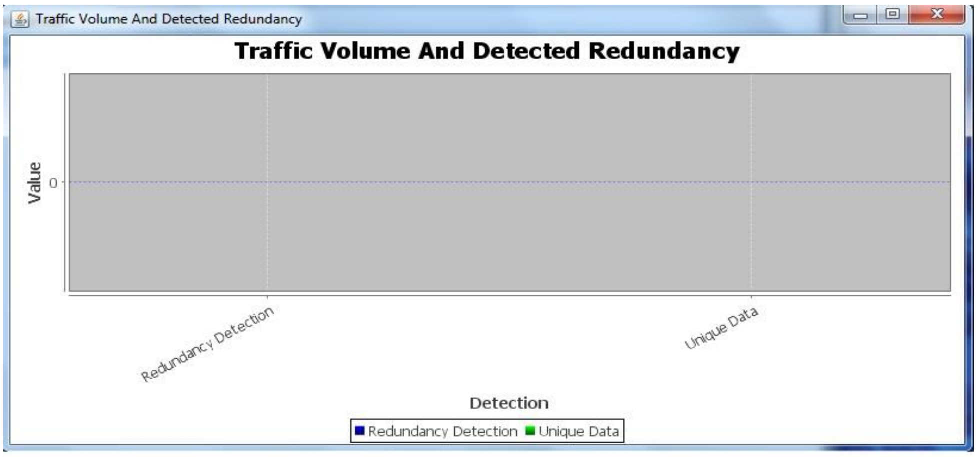


Figure. 4 Result

3. Click on new user screen

   3.1 Login Screen:

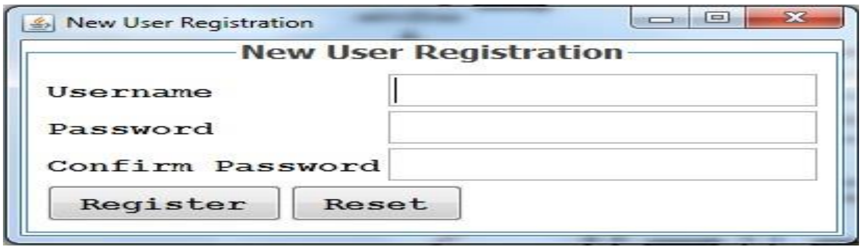
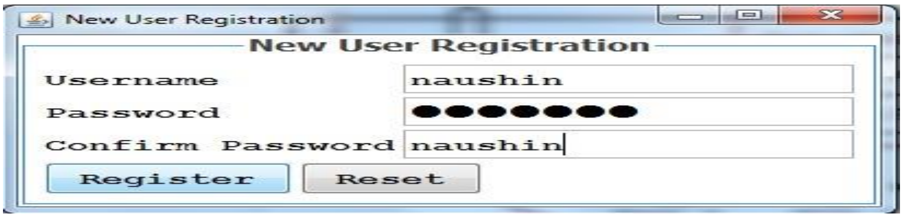


Figure 5 Login Screen

4. Click for new user:
New User Screen:



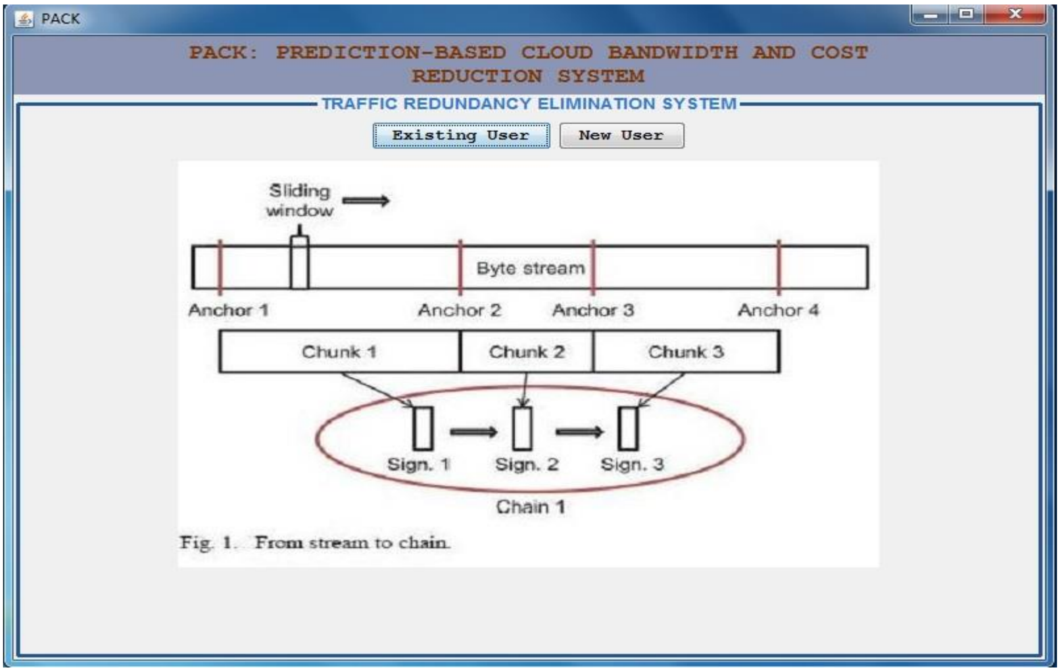
New User Registration

New User Registration

Username naushin
Password ●●●●●●●
Confirm Password naushin

Figure: 6 New User Registration

4.1 Click on existing user to login:
Login screen



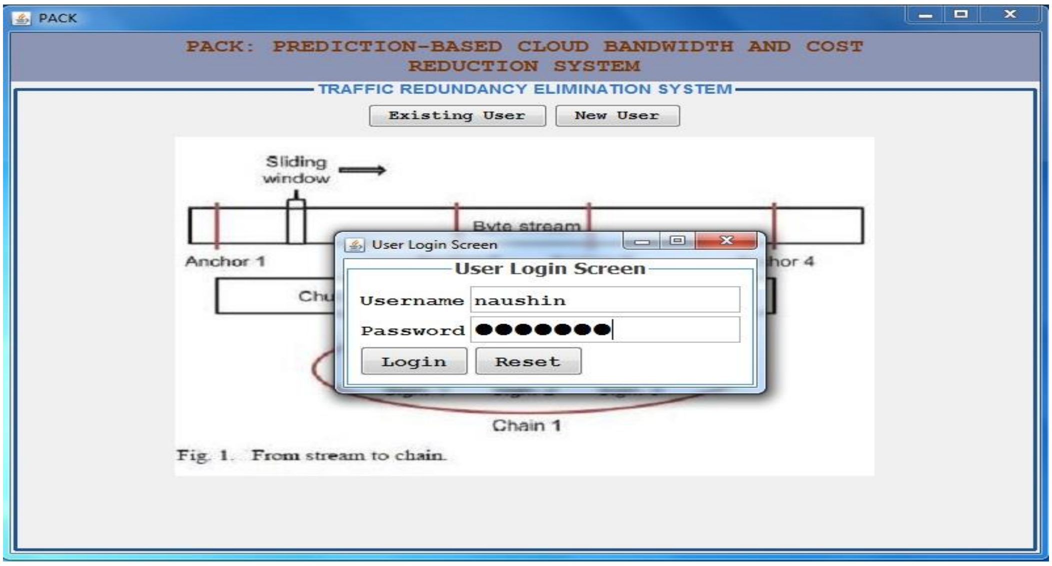
PACK

PACK: PREDICTION-BASED CLOUD BANDWIDTH AND COST REDUCTION SYSTEM

TRAFFIC REDUNDANCY ELIMINATION SYSTEM

Figure 7 Login Screen

4.2 Login screen after successful login



PACK

PACK: PREDICTION-BASED CLOUD BANDWIDTH AND COST REDUCTION SYSTEM

TRAFFIC REDUNDANCY ELIMINATION SYSTEM

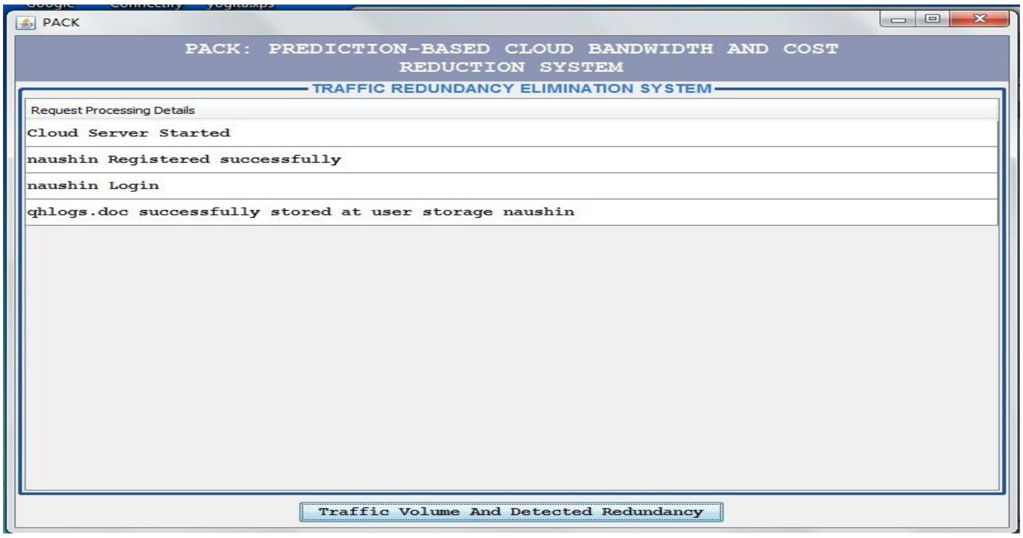
User Login Screen

User Login Screen

Username naushin
Password ●●●●●●●

Figure 8 Login Screen

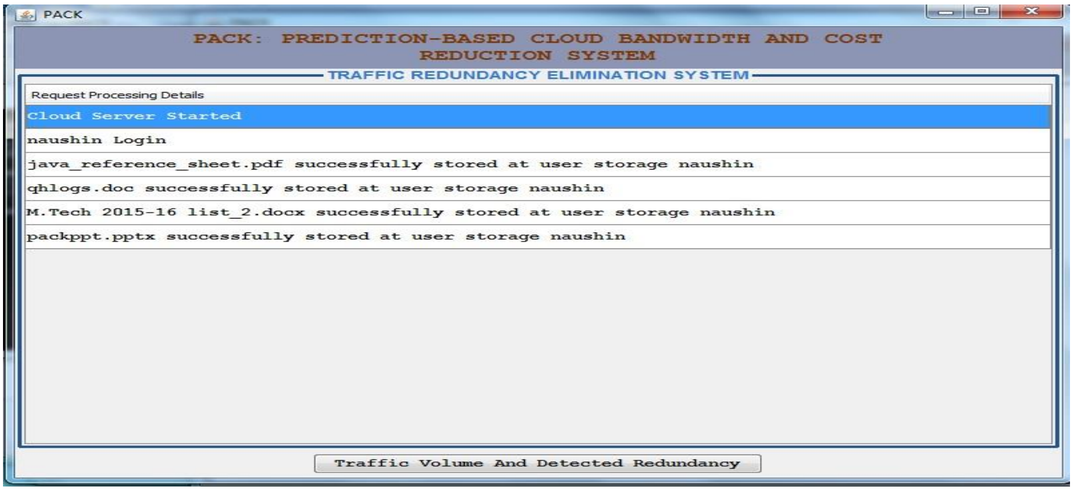
5. Click for registered User
6. upload the data into the server
7. After uploading the files into Cloud Server


PACK
PACK: PREDICTION-BASED CLOUD BANDWIDTH AND COST REDUCTION SYSTEM
TRAFFIC REDUNDANCY ELIMINATION SYSTEM
Traffic Volume And Detected Redundancy

```
Request Processing Details
Cloud Server Started
naushin Registered successfully
naushin Login
qhlogs.doc successfully stored at user storage naushin
```

Figure 9 Uploading to Cloud Server

7.1 Uploading into Server


PACK
PACK: PREDICTION-BASED CLOUD BANDWIDTH AND COST REDUCTION SYSTEM
TRAFFIC REDUNDANCY ELIMINATION SYSTEM
Traffic Volume And Detected Redundancy

```
Request Processing Details
Cloud Server Started
naushin Login
java_reference_sheet.pdf successfully stored at user storage naushin
qhlogs.doc successfully stored at user storage naushin
M.Tech 2015-16 list_2.docx successfully stored at user storage naushin
packppt.pptx successfully stored at user storage naushin
```

Figure 10 Uploading into Server

1. For User Operation

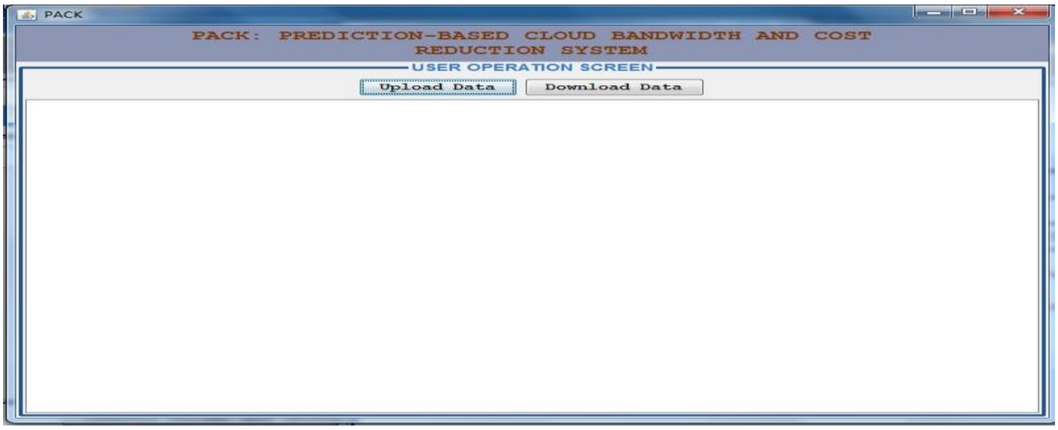
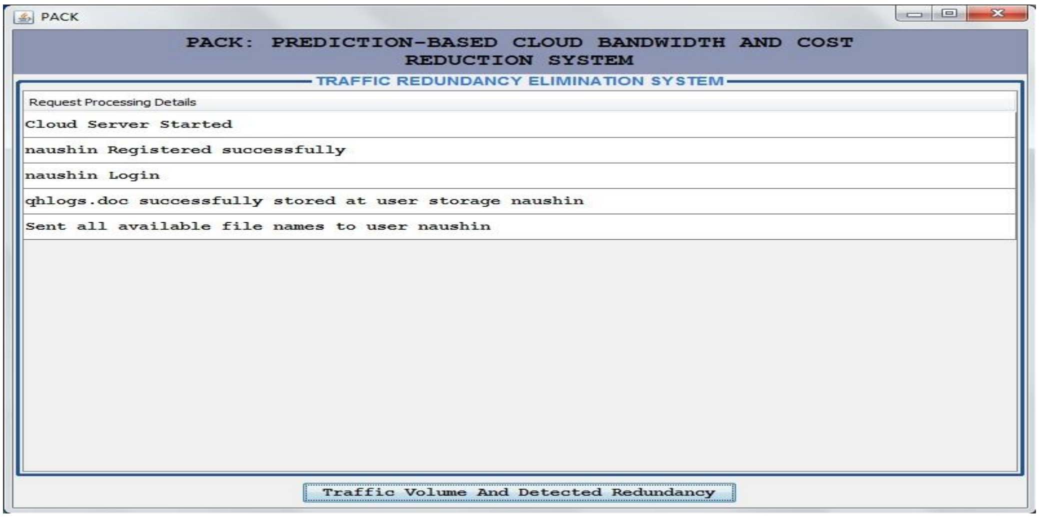

PACK
PACK: PREDICTION-BASED CLOUD BANDWIDTH AND COST REDUCTION SYSTEM
USER OPERATION SCREEN
Upload Data
Download Data

Figure 11 User Operation Screen

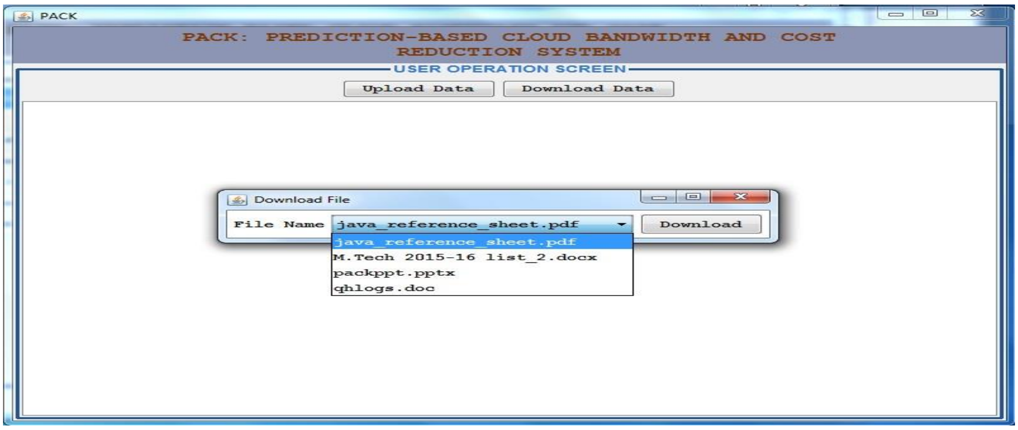
2. The uploaded files will be stored at cloud server :


PACK
PACK: PREDICTION-BASED CLOUD BANDWIDTH AND COST REDUCTION SYSTEM
TRAFFIC REDUNDANCY ELIMINATION SYSTEM
Request Processing Details
Traffic Volume And Detected Redundancy

```
Cloud Server Started
naushin Registered successfully
naushin Login
qhlogs.doc successfully stored at user storage naushin
Sent all available file names to user naushin
```

Figure 12 Uploading Data Screen

3. Client can download the data from cloud, to download click on Download Data.

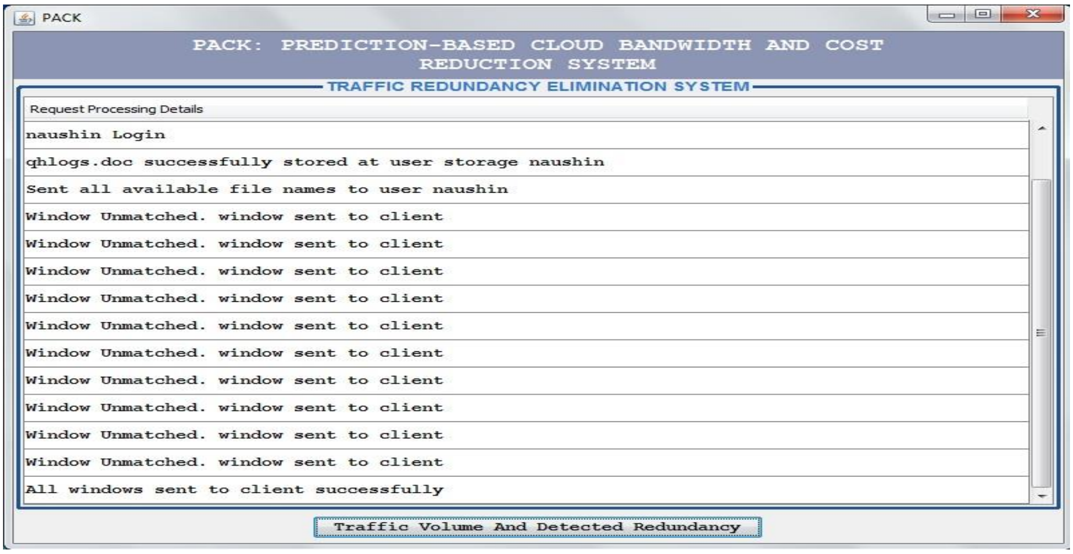


File Name: java_reference_sheet.pdf

```
java_reference_sheet.pdf
M.Tech 2015-16 list_2.docx
packppt.pptx
qhlogs.doc
```

Figure 13 Downloading From Server Screen

4. For Downloading in Chunk:


PACK
PACK: PREDICTION-BASED CLOUD BANDWIDTH AND COST REDUCTION SYSTEM
TRAFFIC REDUNDANCY ELIMINATION SYSTEM
Request Processing Details
Traffic Volume And Detected Redundancy

```
naushin Login
qhlogs.doc successfully stored at user storage naushin
Sent all available file names to user naushin
Window Unmatched. window sent to client
Window Unmatched. window sent to client
Window Unmatched. window sent to client
Window Unmatched. window sent to client
Window Unmatched. window sent to client
Window Unmatched. window sent to client
Window Unmatched. window sent to client
Window Unmatched. window sent to client
Window Unmatched. window sent to client
Window Unmatched. window sent to client
All windows sent to client successfully
```

Figure 14 Downloading in Chunk Screen

5. Unique Data

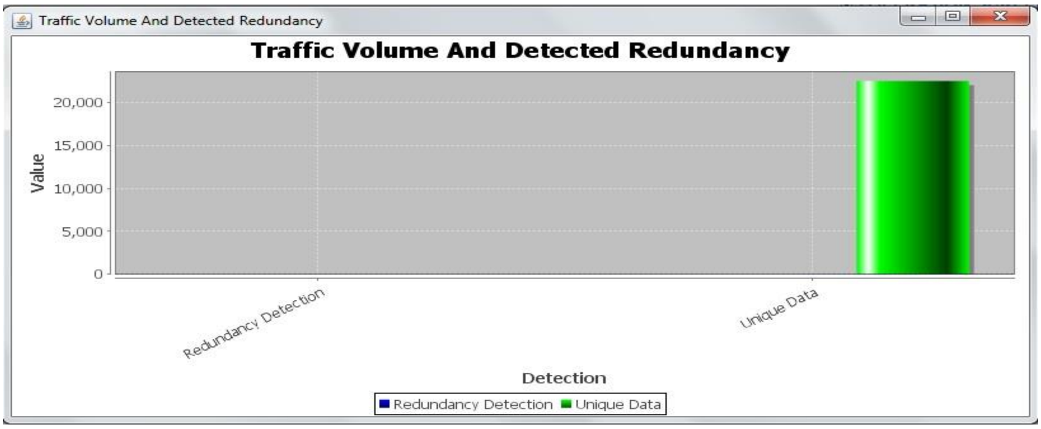
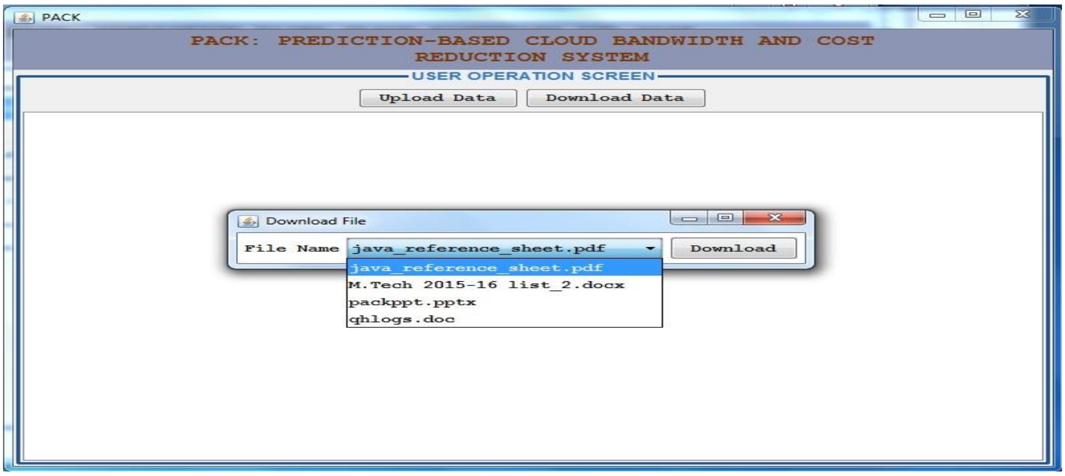


Figure 15 Unique data Screen

5.1 Downloading from Server-2:

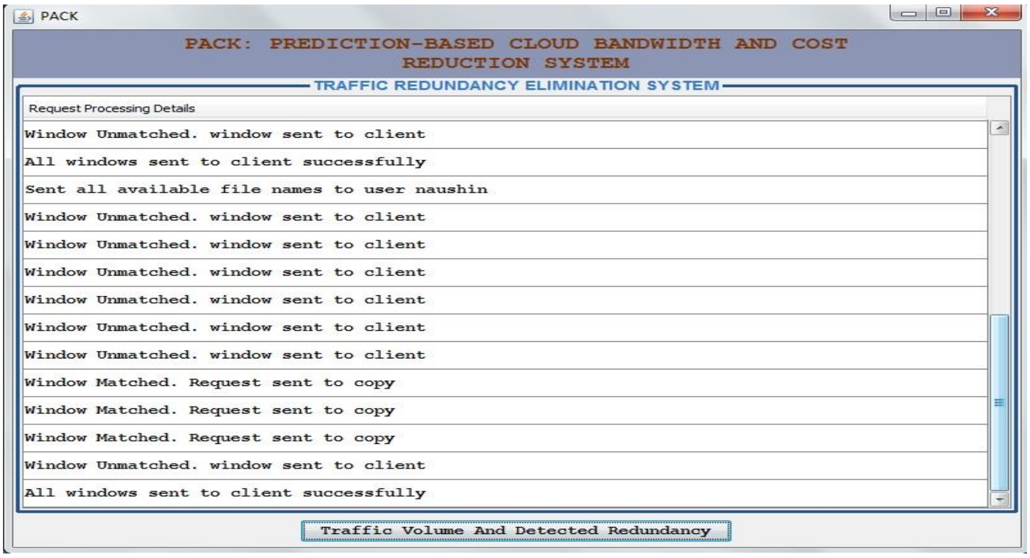


File Name: java_reference_sheet.pdf

java_reference_sheet.pdf
M.Tech 2015-16 list_2.docx
packppt.pptx
qhlogs.doc

Figure 16 Downloading from Server 2

6. Matched Data:


PACK
PACK: PREDICTION-BASED CLOUD BANDWIDTH AND COST REDUCTION SYSTEM
TRAFFIC REDUNDANCY ELIMINATION SYSTEM
Traffic Volume And Detected Redundancy

```
Request Processing Details
Window Unmatched. window sent to client
All windows sent to client successfully
Sent all available file names to user naushin
Window Unmatched. window sent to client
Window Unmatched. window sent to client
Window Unmatched. window sent to client
Window Unmatched. window sent to client
Window Unmatched. window sent to client
Window Unmatched. window sent to client
Window Matched. Request sent to copy
Window Matched. Request sent to copy
Window Matched. Request sent to copy
Window Unmatched. window sent to client
All windows sent to client successfully
```

Figure 17 Matched Data

7. Redundancy Result

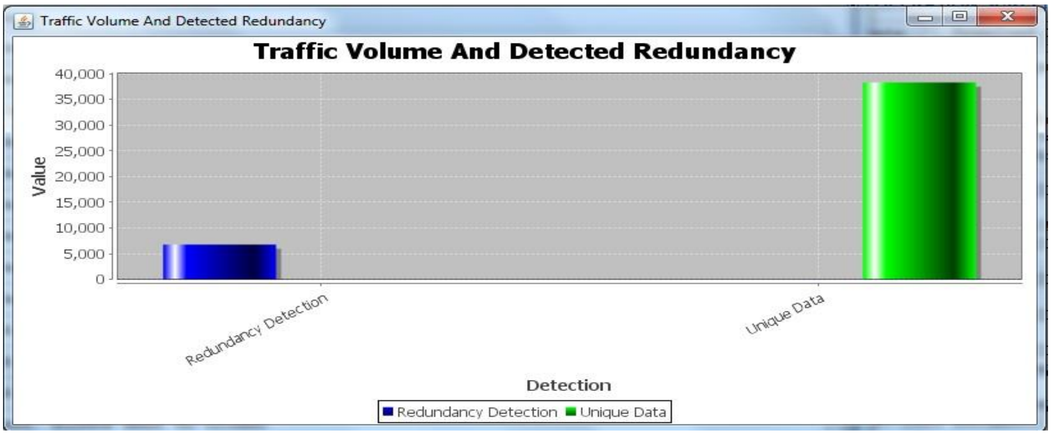


Figure 18 Traffic Reduction Elimination Screen

8. Downloading Server 3

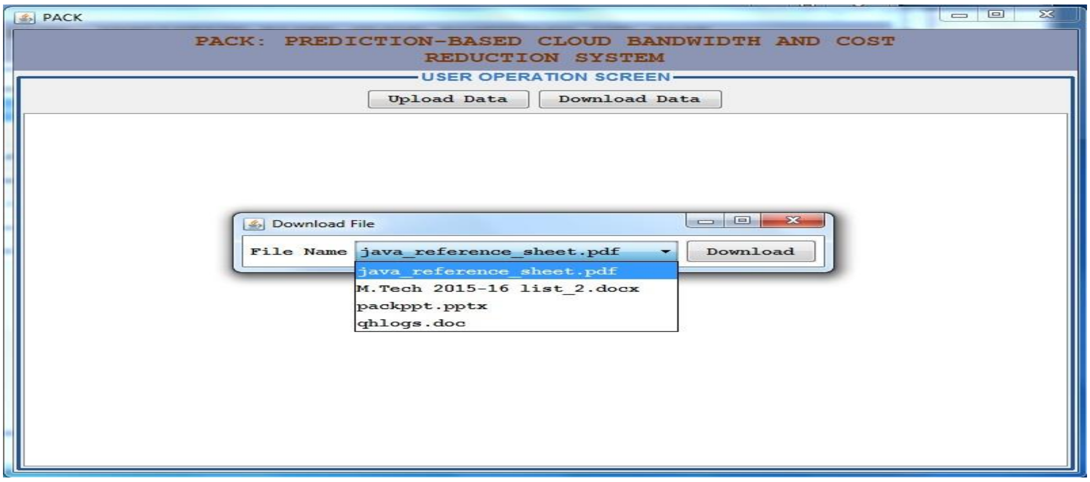
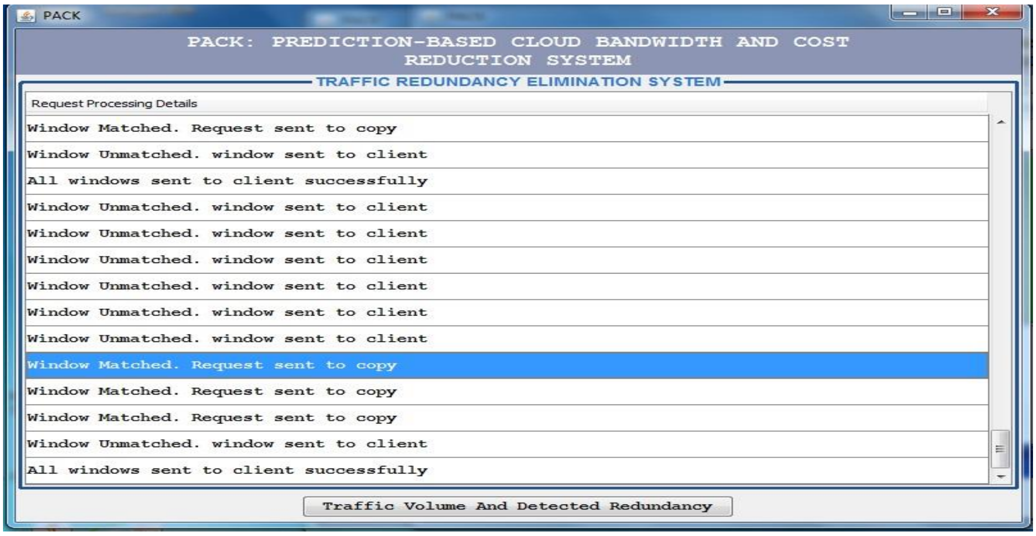


Figure 19 Downloading Server

9. Matched Data


PACK
PACK: PREDICTION-BASED CLOUD BANDWIDTH AND COST REDUCTION SYSTEM
TRAFFIC REDUNDANCY ELIMINATION SYSTEM
Request Processing Details
Traffic Volume And Detected Redundancy

```
Window Matched. Request sent to copy
Window Unmatched. window sent to client
All windows sent to client successfully
Window Unmatched. window sent to client
Window Unmatched. window sent to client
Window Unmatched. window sent to client
Window Unmatched. window sent to client
Window Unmatched. window sent to client
Window Unmatched. window sent to client
Window Matched. Request sent to copy
Window Matched. Request sent to copy
Window Matched. Request sent to copy
Window Unmatched. window sent to client
All windows sent to client successfully
```

Figure 20 Matched Data

10. Server side, click on traffic volume and detected redundancy to see how much the unique and redundancy data is there

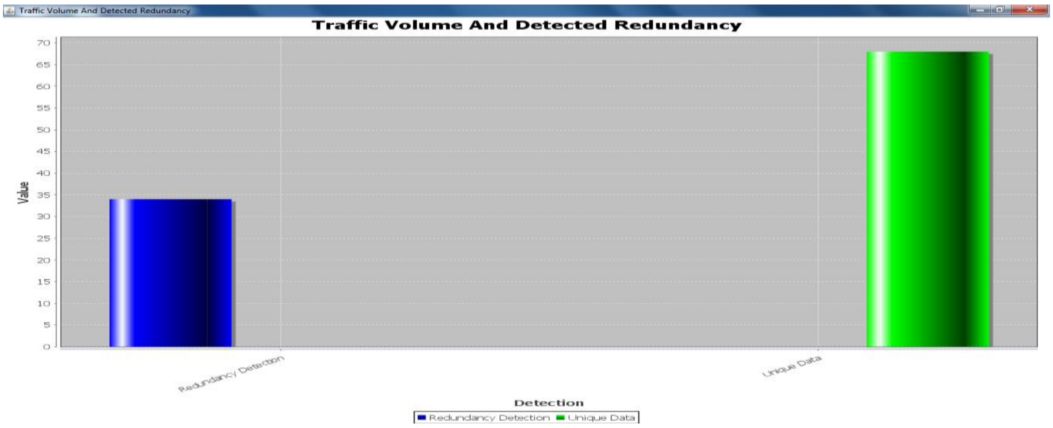

Traffic Volume And Detected Redundancy
Traffic Volume And Detected Redundancy
Value
70
65
60
55
50
45
40
35
30
25
20
15
10
5
0
Redundancy Detection
Unique Data
Detection
🟦 Redundancy Detection 🟩 Unique Data

Figure 21 Final Output Screen

V. CONCLUSION & FUTURE SCOPE

Distributed computing is relied upon to trigger popularity for TRE arrangements as the measure of information traded between the cloud and its clients is required to drastically increment. The cloud condition rethinks the TRE framework necessities, making exclusive center box arrangements insufficient. Thus, there is a rising requirement for a TRE arrangement that diminishes the cloud's operational cost while representing application latencies, client versatility, and cloud flexibility. In this paper, we have displayed PACK, a recipient based, cloud-accommodating, end-to-end TRE that depends on novel theoretical rule that decrease idleness and cloud operational cost. PACK does not require the server to constantly keep up customers' status, subsequently empowering cloud versatility and client portability while saving long haul repetition. Also, PACK is equipped for disposing of repetition in view of substance landing to the customer from different servers without applying a three-way handshake.

Table No. 1 Comparison between other TRE Mechanisms

Scheme	Window	Chunks	Speed
Sample Byte 8 Markers	1 Byte	32 Bytes	1,913 Mbps
Rabin Fingerprint	48 Bytes	8 KB	2,686 Mbps
Pack Chunking	48 Bytes	8 KB	3,259 Mbps

That PACK meets the normal outline objectives and has clear favorable circumstances over sender-based TRE, particularly when the cloud calculation cost and buffering prerequisites are essential. Also, PACK forces extra exertion on the sender just when repetition is abused, in this manner lessening the cloud general cost. Two intriguing future augmentations can give extra advantages to the PACK idea. To start with, our usage keeps up chains by keeping for any piece just the last watched resulting lump in a LRU design. An intriguing augmentation to this work is the measurable investigation of chains of lumps that would empower different potential outcomes in both the piece arrange and the comparing expectations. The framework may likewise permit making more than one expectation at any given moment, and it is sufficient that one of them will be right for effective movement disposal. A moment promising bearing is the method of operation improvement of the half and half sender– beneficiary approach in view of shared choices got from recipient's energy or server's cost changes.

REFERENCES

[1] E. Zohar, I. Cidon, and O. Mokryn, "The power of prediction: Cloud bandwidth and cost reduction," in *Proc. SIGCOMM*, 2011, pp. 86–97.

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

[3] U. Manber, "Finding similar files in a large file system," in *Proc. USENIX Winter Tech. Conf.*, 1994, pp. 1–10.

[4] N. T. Spring and D. Wetherall, "A protocol-independent technique for eliminating redundant network traffic,"in *Proc.SIGCOMM*, 2000, vol.30, pp. 87–95.

[5] A. Muthitacharoen, B. Chen, and D. Mazières, "A low-bandwidth network file system," in *Proc. SOSP*, 2001, pp. 174–187.

[6] E. Lev-Ran, I. Cidon, and I. Z. Ben-Shaul, "Method and apparatus for reducing network traffic over low bandwidth links," US Patent 7636767, Nov. 2009.

[7] S.Mccanne andM. Demmer, "Content-based segmentation scheme for data compression in storage and transmission including hierarchical segment representation," US Patent 6828925, Dec. 2004.

[8] R. Williams, “Method for partitioning a block of data into subblocks and for storing and communicating such subblocks,” US Patent 5990810, Nov. 1999.

[9] Juniper Networks, Sunnyvale, CA, USA, “Application acceleration,” 1996 [Online]. Available: http: //www.juniper.net/us/en/products-services/application-acceleration/

[10] Blue Coat Systems, Sunnyvale, CA, USA, “MACH5,” 1996 [Online]. Available: http://www.bluecoat.com/products/mach5

[11] U. Manber, “Finding similar files in a large file system,” in Proc. USENIX Winter Tech. Conf., 1994, pp.1–10.

[12] A. Muthitacharoen, B. Chen, and D. Mazières, “A low-bandwidth network file system,” in Proc. SOSP, 2001, pp. 174–187.

[13] A. Anand, C. Muthukrishnan, A. Akella, and R. Ramjee, “Redundancy in network traffic: Findings and implications,” in Proc. SIGMETRICS, 2009, pp. 37–48.

[14] A. Anand, A. Gupta, A. Akella, S. Seshan, and S. Shenker, “Packet caches on routers: The implications of universal redundant traffic elimination,” in Proc. SIGCOMM, 2008, pp. 219–230.

[15] A. Anand, V. Sekar, and A. Akella, “SmartRE: An architecture for coordinated network wide redundancy elimination,” in Proc. SIG-COMM, 2009, vol. 39, pp. 87–98.