Reducing User Latency in Web Prefetching Using Integrated Techniques

Submitted by

Naveed Ahmad
(407-FBAS / MSCS / S08)

Supervised by

Owais Ahmed Malik
Assistant Professor,
School of Electrical Engineering and Computer Science,
NUST, H-12, Islamabad.

Co-Supervised by

Asim Munir
Assistant Professor,
Department of Computer Science,
International Islamic University, Islamabad.

Department of Computer Science

Faculty of Basic & Applied Sciences,

International Islamic University, Islamabad.

(2011)

With the Name of

Allah,

The most merciful and compassionate the most gracious and beneficent whose help and guidance we always solicit at every step, and every moment.

A dissertation submitted to the

Department of Computer Science

International Islamic University, Islamabad

As a partial fulfillment of the requirements for

The award of the degree of

Masters of Science in Computer Science

DEDICATION

To my **parents** who are like cool shade in the noontide of my life, particularly to my **mother** whose hands get tired of praying for my success and to those who pray for me and encouraged me throughout my educational career.

ACKNOWLEDGMENT

I am grateful to almighty ALLAH, WHO is merciful and beneficent, and WHO enabled me to work on this research successfully. I offer heartiest "DROOD-O-SALAM" to our holly Prophet MUHAMMAD (Sally Allah-o-Alaihe Wa Sallam). I wish to express my deepest gratitude to my supervisor, **Mr. Owais Ahmed Malik**, for his continual and essential support, esteemed supervision, encouragement and guidance throughout the course of my research. Without his sincere guidance this work would never have been completed. I would like to sincerely thank him for the crucial help he offered me throughout a series of discussions and meetings. I am also grateful to my co-supervisor, **Mr. Asim Munir** who assisted me through useful discussions on every problem I have faced during my research. His comments during our meetings were always constructive and encouraging. I would further like to thank all the fellow research students in the Web Semantics Group for creating an ideal working environment.

I would like to thank to Dr. Amir Hayat who is the HoD of Computing in NUST- School of Electrical Engineering and Computer Science for giving me his valuable time whenever I have problems during the course of my research thesis. His motivational talks and meetings provided me a guideline for identifying my research problem. I would also highly thankful to Dr. Khalid Latif and Dr. Sharifuallah Khan of NUST- School of Electrical Engineering and Computer Science (SEECS) for allowing me to attend the Research Group Meeting at their institute.

Finally, the credit of this research goes to my parents and to all of my family members for their unconditional support and encouragement throughout my study.

Naveed Ahmad

407-FBAS/MSCS/S08

Declaration

I hereby declare that this thesis, neither as a whole nor as a part thereof has been copied out from any source. It is further declared that no portion of the work presented in this report has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

Naveed Ahmad

407-FBAS/MSCS/S08

Abstract

Web caching and Web Prefetching are the areas for research in Web Mining. Web Prefetching improves the performance of the Web Caching techniques due to prediction of the user pages in advance before the user requests. Both techniques provide the web pages local to the user; they provide the resource i.e. web pages for user's ease and access. Web caching is limited due to its size. Web prefetching is the process of accessing the web objects before the user's request. When a client requests for a web page, before accessing the web page a prediction is made for accessing that web page. All the web objects are brought from server to the client. The access to the web objects are on the basis of the data prefetched from the server. This research focuses on when a client prefetched the page from the server then how can we improve the overall performance of web prefetching mechanism by using the network bandwidth. The proposed mechanism provided the pages locally available to a user or group of users by utilizing bandwidth of the network. The server contains an algorithm for the prediction of web pages.

Table of Contents

List of Figures

List of Tables

CHAPTER # 1

Introduction to Web Mining and Web Prefetching

1. Introduction to Web Mining

Web mining is the application of data mining techniques which is used to discover the unknown patterns of web usage [1]. Web mining is used to evaluate the customer's behavior over the web in particular the parts of the web which are frequently accessed by the users. The data of the web is analyzed from the user's view. The users of a web server view the pages that are important to them. This kind of information is taken from the web data by using different techniques of data mining.

Our work is based on some concepts which are defined as follows [2]. A single user is a person that accesses the file from one or more files by the browser. A web page is the collection of all the files that are seen from the user's screen at any point. A web page contains several files at any time for a user. The web page contains frames, graphics and media. The clicks stream for a user is a series of pages that a user takes from a site.

Proxy server is computer which stores the web pages that are requested by the user. When a user requests for web page, the request is seen by the proxy server. If the web pages requested by the users are found in proxy server these pages are given to the users and if the web page requested by the user is not contained in proxy server then the request of user is entertained by web server [3]. A session is the time for user that accessed a part of a web site. From the server point of view a session is the set of pages that contains the information of users who visited the web page.

1.1 Types of Web Mining

Web mining is generally divided into three types [1] [4] [5]:

- Web Content Mining
- Web Structure Mining
- Web Usage Mining

The types of Web mining is illustrated in the figure below:

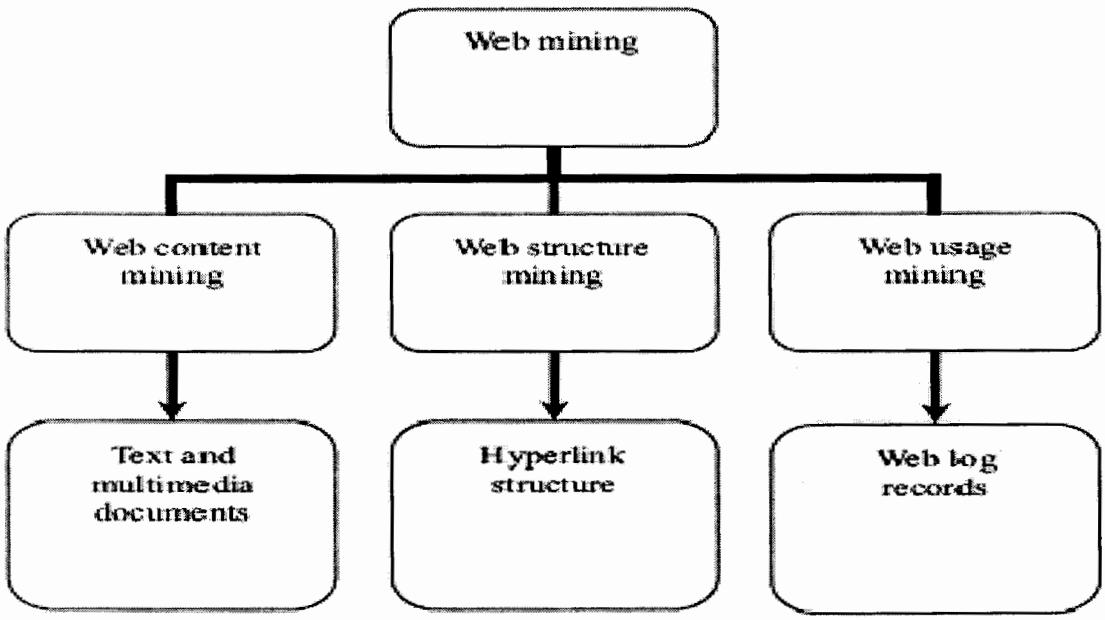

Figure: 1.1 Categories of Web Mining [1]

1.1.1 Web Content Mining

It is the set of techniques which are used for helping the users to find out the documents to fulfill the certain criteria. Web content mining is also called as text mining. Content mining is used for scanning and mining web documents, text, pictures and graphs of the Web page for the user in order to provide relevant query search. The scanning of the web page is completed when the cluster of the web pages is made. There is a huge amount of data available on the internet. The clustering of similar web pages provides the relevant information for search engine query [1] [4] [5]. Text mining reduces the irrelevant information for the search engine because it mines the information according to the user's request of a web page.

1.1.2 Web Structure Mining

The web structure mining is the mining of collection of contents such as HTML and XML Tags and how different web pages are interlinked by the other web pages. For example link based clustered analysis technique's objective is to cluster the web objects i.e. the similar

web objects are grouped together and dissimilar web object are grouped in different groups [1] [4][5].

1.1.3 Web Usage Mining

The web usage mining is the type of web mining in which we study the techniques to model/mine the user behavior i.e. which parts of the web are visited most frequently [1] [4] [5].Web usage mining contained the information such as how a web page is used, the date, time and IP addresses and page reference [1] [4] [5]. Mining of the data in the web logs is an example of web usage mining because we analyze the behavior of users i.e. what kind of information they get from the web, the information about web that can be seen by server point of view, which data of web can be accessed by its user. It help us to improve the parts of the web which is used most of the user.

1.2 Applications of Web Mining

There are many useful benefits of Web Mining.

1. The main uses of web mining are to gather, manage, classify and give the important information which is available on the WWW to the users who request it.

2. One application of web mining is used in e-commerce and e-services. In e-commerce and e-services web mining gives the advantage to business. Consumer of web services is analyzed by using different techniques of web mining. This helps out in finding the usage of the web and behavior of the web services users. Companies can understand the user demands. In this way they can take advantage of web mining techniques.

In e-commerce customer's requirements and risk to loss the customer is evaluated through web mining techniques and this gives benefits to the company [5].

1.3 Web Prefetching

Web caching and Web prefetching are two important areas of web mining [3] [6]. The number of web users has been increased continuously due to the published data available on the websites. Due to increasing number of users for accessing the data will increase in server loads and network traffic.

Web caching is one of techniques used for the client, server and proxies to help users in fast access to the web documents. Web caching gives us the fast access to web document because the cache keeps the most recently used documents in the cache [3]. Cache can be deployed to client, server or proxy depending for the purpose for which it is used. Web cache reduces latency because cache keeps the data/web pages local to the users. If the documents are present in the client's cache, then the documents will be accessed faster and this will reduce the network traffic of overall system.

Web prefetching is the process of accessing the web objects before the user's request [6]. When a client requests for a web page, before accessing the web page a prediction is made for accessing that web page. All the web objects are brought from server to the client. The access to the web objects are on the basis of the data prefetched from the server.

The cache has the limited size to access the web pages that are recently stored in the cache. The repeated access to same pages will be performed on cache. Prefetching is classified into server initiated prefetching, client initiated prefetching and proxy initiated prefetching and hierarchical prefetching [6].

1.4 Problem Statement

To improve the performance using prefetching mechanism, a technique is used for caching [7]. In this mechanism, the prediction web object/ web page is based on the current and previous request to the same web object.

In [8], a method for prediction of the web pages is made at the proxy server. The proxy contained a prediction engine. In the prediction engine, proxy server logs are maintained. The proxy server log records all the users/clients that send request to the server that are kept on the web log which is used to form mineable warehouse. The warehouse is used to track the user activity. In model predictive web prefetching is defined at the proxy server for access time optimization of user.

Web prediction engine can be placed at client, server and the proxy. In order to reduce the user's latency, the best architecture for the prediction engine is to locate it at proxy. This means that architecture plays an important role in the web prefetching. If the prediction engine is at proxy, then the traffic between server and proxy is increased [9].

When a client requests a web page from the server, how can we improve the overall performance of web Prefetching mechanism by using the network bandwidth?

The problem statement clearly defines that when a user/group of users access the pages from the server, how can we reduce the latency of the user or group of users by web prefetching mechanism? The mechanism provided the pages locally available to the user or group of users by utilizing bandwidth of the network.

1.5 Contribution

We have taken Sequential Rank based Selection algorithm [30] and modified it for proposed model technique in "Reducing User Latency in Web Prefetching Using Integrated Techniques" and results proved that this technique has provided higher availability of resources with accuracy. The proposed technique has provided resources with highest availability for cluster formation and these resources are returned to the user for predictive prefetching.

On the basis of model, we have modified Sequential Rank based selection algorithm for achieving the following objectives:

1. To develop a system without any threshold value for improving the over all performance of the system.
2. To select clusters for users to availability of resources.
3. Provide server resource usage information.

The data sets are taken from four different web servers. The data contained International Islamic University, NUST University, Hazara University and Twitter web servers. Analysis of results proved that proposed technique is efficient in web page prediction. The proposed technique improved overall performance of web prefetching system.

CHAPTER # 2

Literature Review

2. Literature Review

To minimize the latency of the user for accessing the web pages, web caching is used. Web caching reduces network traffic. The network traffic may increase due to the increase of data transmission and exchange of information over the network [3]. The exchange of messages and data increases the network traffic and contacting every time to the server increases the server load. To overcome this problem (the delay for the users to access the web documents), Web caching is used. Caching reduces the latency in client's view.

Using caching with server, clients and proxies reduces the latency to access the web document, as cache stores the recently accessed documents. The frequent access to the similar documents is performed on the basis of cache.

The classification of caching according to different architectures is described in [3] is explained as follows.

2.1 Proxy Caching

In proxy caching architecture, proxy server has a cache that stored the document of home server in its cache. When a client requests for the document, proxy server will reply back to the client if the requested documents are present in the client's cache. The only one cache is used for the whole network in proxy caching architecture. Single cache for the entire network causes the scalability problem i-e; we cannot increase the cache/network. Proxy coaching's disadvantage is the scalability issue [3]. The load balancing cannot be done in the single proxy server caching.

Types of proxy caching

- Reverse Proxy Caching
- Transparent Caching

2.1.1 Reverse Proxy Caching

In reverse proxy caching, Cache is used with the server for storing the contents of server. It helps the server in web hosting i-e; mapping a domain to a single site. Reverse proxy server is independent from client side.

2.1.2 Transparent Caching

Transparent caching is deployed at the proxy server. Transparent caching removed the disadvantages of load balancing because transparent caching system has more than one cache at proxy server. If a cache has more data than the size of a cache, the extra data will be given to other cache in the system for load balancing. For implementing the transparent caching, Routers or switches with multiple caching are used at proxy server.

2.2 Adaptive Web Caching

Adaptive caching is used in the network of distributed cache system. In distributed cache system, the cache can joins the group and leave group of cache system on the content demand by the users [3]. Adaptive caching uses the Content Group Management Protocol (CGMP) and the Content Routing Protocol (CRP) for the leaving and joins the groups of cache system.

2.3 Push Caching

Push caching is used for the keeping the information close to the user who requested for it. It is different from the adaptive caching because push caching is for the user's information/data whereas adaptive cache is used for keeping the content of information.

2.4 Active Caching

Active caching is used in dynamic approach. In dynamic approach the data in cache is changed consistently, it is difficult to allocate all the information in cache due to limited size of cache.

All these caching architectures are used for the user's easiness and convenience in different networks design for reducing latency of users and load sharing among cache. Caching uses network bandwidth for the availability of web documents. It reduced server load and network traffic of overall system. The data is locally available for client [3].

2.5 Framework of Prefetching

A technique of web logging is used for improving web prefetching [7]. This mechanism improved in reducing network traffic using prefetching system. The mechanism mine data from for web logs.

This technique enhanced the web caching and web prefetching for accessing the web documents. The whole procedure of web mining is operated on the basis of web logs. This is similar to server initiated approach. In server initiated approach, server keeps the web logs of each of its client. Log contains the probability of each web object to be prefetched.

On the basis of weblog, prediction of future access of page is done. The prediction about the web object/ web page is base on the current and previous request of a user to the same web object/web page.

The most frequent pages/documents from the web logs are mined because many web server logs can keep the logs of the user's behavior. By using an algorithm, the probability of future request of web pages is calculated. The documents with highest probabilities are predicted for users. In the real environment, this technique has reduced the latency of users [7].

2.6 Web Prefetching at Proxy Level

A method is used for web prefetching [8]. In this method, prediction of web pages is made at the proxy server. The proxy contained a framework known as prediction engine. In the prediction engine, the proxy server logs are maintained. The proxy server log records the entire client's request. These requests are sent to the server. The requests generated by user forms mineable warehouse at proxy server log. The warehouse is used to track the user's activity. In this way predictive web prefetching is made at the proxy server for access time optimization of users.

The predicted web pages from the server are loaded into the proxy server while user is busy in performing some other tasks using the data mining rules. The web caching is one of the ways that helps to minimize the network usage, server loads and enhances in reducing the average latencies experienced by its users.

Web latency can be reduced, if the prediction of pages is made by software at proxy server. When a user is viewing current page and their contents are prefetched in advance. This method of accessing the pages early before the user's request is called predictive web prefetching. Data mining techniques are used at proxy server for web prefetching; the method optimized the access time of access of web server. In the proxy server, the data mining techniques computed the web pages that are to be requested by a user [8].

There is a proxy server between the client and server. The proxy server has the following objective used in the proposed model.

- Reduce Latency
- Filter Unwanted Requests

2.6.1 Reduce Latency

There are millions of users accessing the web pages of a server. When a proxy server is used, the user's requests are checked in proxy server. If the web pages are in the proxy server, then pages are sent back to the user as a reply. Due to the proxy server, every time the client is not contacted with the server for the same pages requested by user.

Proxy server keeps the same pages as they are requested by the users. If the pages are not in the proxy server, the request of the user for the web pages is sent to the web server. The web server sent web pages are reply back to the user through proxy server. For example two users A and B request for a web page C. Assume that A has requested for the web page and got the page C first. When user B requests for the Page C, Page C will be return to user B from proxy server. This will reduce latency because the web page is present in proxy server.

2.6.2 Filter Unwanted Requests

Proxy server could prevent the visit to specific websites to its users. For example a company can restrict its employees to see certain web sites. The privilege to access the web pages is done through proxy servers. The predicted web pages are brought into the proxy server and stored temporarily in the proxy servers.

When requests for the web pages are made by the user, these pages are available in the cache of the proxy server. Prediction prefetch engine will compute the probabilities of documents at the proxy server for prefetching.

2.6.3 Prediction at Proxy Server Logs

All the records of web users who access the web pages are stored in the web logs. Web logs are used as a mineable warehouse. These web logs of proxy server are for the purpose of keeping the track of limited users. The weblogs are filtered by removing the irrelevant like image

files that do not take part in the pattern recognition. User identification and path completion are also done in the proxy server logs.

The path completion is processing of web server logs when some important access of web page is not recorded in the proxy server logs. After overall cleaning of the proxy server log, a data mart is created. The data mart is used for prefetching of the web page in the proposed model in [8].

2.7 Architecture for Web Prefetching

In [9] a different architecture of web prefetching is discussed. The architecture is compared with the other architectures. There are three main components of the architecture for the prediction of web pages. These are client, server and the proxy for the prediction of web objects.

The proposed generic web prefetching model of architecture consists of the following components.

- Generic Web Architecture
- Prediction Engine
- Prefetching Engine

2.7.1 Generic Web Architecture

The proposed architecture is based on the generic web architecture. The generic web architecture consists of two main parts.

- **User/ clients agents**

The user/clients agents are the software for the user to accesses the web pages.

- **Web Server**

When a user demands for a web page, it simply writes down the url or by clicking the hyperlink of a previously downloaded page. Finally the whole page is displayed.

- **Proxy Server**

A proxy can be used for a group of users to access the web page of server. If the proxy is used between the client and server, the server load and the network traffic between proxy and server is reduced. Proxy keeps the previous accessed pages by the users in its cache.

2.7.2 Prediction Engine

The prediction engine can be placed at any of the part of prefetching architecture. The main thing is to place the prediction engine in such a way that it reduces the latency experienced by the user. The user's behavior on the previous access of web pages can found through the prediction engine. The prediction engine preprocessed the previous web pages and predicted the new hints of pages for the users. The result produced by the prediction is a list of hints. The hint list is used as prediction of pages for the users in near future [9].

The results are taken by locating the prediction engine at client, server and proxy [9]. SFAR is value of object Session First Accessed Ratio. When a user accessed first time, the object can not be predicted. This is called SFAR (Scene First Access Ratio). When object is never scene before the prediction engine, it is called FSR (First Scene Ration). FSR is the ratio of number of different objects to total number of requests and SFAR is the ratio of number of session to number of accesses. Session is the time of accessed of object by user. On the basis of values of SFAR and FSR, performance is measured by locating the predictor at different position of client, server and proxy architectures. The results [9] are given below:

	Single Predictor			**Collaborative Predictor**	
Factor	**Client**	**Proxy**	**Server**	**Client-server**	**Proxy-Server**
SFAR(object) **FSR(object)** **FSFR(object)**	1.3% 41.6% -	1.3% 16.3% -	45% 0.4% -	1.3% 0.4% 5%	1.3% 0.4% 0.8%
Maximum Prediction	58.4%	83.7%	55%	95%	98.7%
SFAR(Latency) **FSR** **SFSR**	28% 25.6% -	2.8% 32.6% -	63.4% 1.7% -	2.8% 1.7% 4.6%	2.8% 1.7% 0.7%
Maximum Reduced	54.5%	67.4%	36.6%	95.7%	97.2%

Table 2.1: Comparisons of results by locating Predictor at different position [9]

Web prediction engine can be placed at client, server and the proxy. The experimental results in [9] showed that the best architecture for locating the predictor is proxy server.

2.7.3 Prefetching Engine

Prefetching engine can be located at client, server and proxy server. The goal of the [9] is to locate the prefetching engine in such a way that it reduced the user latency. The function of prefetching engine is to take hint provided by the prediction engine. The best position for the prefetching engine placement is to locate it near to the clients.

A mechanism [10] for web prefetching is to optimize the web server using the page rank prefetching for clustered accesses. In the mechanism the rank page algorithm calculates the importance of a page or document to be prefetched. The importance of page to be prefetched depends upon the next page and previous page. The pages that interlinked to each other are made a cluster. The clustered web pages are ordered according to the rank algorithm [10].

When prefetching is used with hierarchical cache based system, it reduces the latency of the user's request for web page. Prefetching the hierarchical cache system improves the performance of overall system [11].

2.8 Latency in Web Prefetching

In [12], a technique is defined for maximum benefits that a user can achieve from web prefetching in real environment. The technique used in the paper reduced the user latency experienced by the user. In the technique a predictor is made for predicting the web pages. In the previous work no such techniques was introduced that works in the real web environment.

The functionality of the predictor is to predict the accurate page/documents. In the prefetching mechanism, users request for page/document that they required. For prefetching a user web browser provides hints that are given by the prediction engine.

Web Prefetching was not used before due to two reasons.

1. Limited bandwidth of Network for user
2. Number of protocol for web prefetching

Now these two problems are solved for the availability of data local to the user. The web server contained an algorithm. The algorithm predicts the hints on the bases of user request.

Predictor predicts the pages for user by preprocessing the previous files of server. There are two components of architecture proposed in the reference [12].

- Prediction Engine
- Prefetcher

2.8.1 Prediction Engine

In the prediction engine the preprocessing of user's behavior is done on the previous accessed pages at server side. The algorithm is in the prediction engine that is used to predict hints for the user. The maximum accuracy of the web prefetching depends upon the hint provided by prediction engine. If the prediction engine predicts the correct web object, then it will improve to reduce the latency.

2.8.2 Prefetcher

The prefetcher is at the client side. The hints provided by the prediction engine are prefetched by the prefetcher. In real environment web prefetching can reduce page latency experienced by user is [12] up to 52%. The percentage is calculated by simulating users in the framework. These clients performed requests to a simulated web server with a prediction engine that included a perfect prediction algorithm according to architecture. This implemented predictor is perfect because it always provides precise hints and this can be done only in a testing framework, by reading the client requests in advance from the trace file.

A model is proposed in [13] for the estimation of the network traffic to calculate extra loads at the server caused by prefetching.

In the reference [14], a mechanism of prefetching is explained. In the method prediction about the web document is made at prefetch. The method is efficient because the method did not increase the basic cost of prefetching mechanism and the method proposed in the reference [14] reduced the user latency.

The web prefetching techniques would not require extra resources because it involved the speculation prediction of some web documents. The model speculated the web document correctly; it will reduce the user latency. The prediction at prefetch techniques prefetched the web object when the user is busy for getting the web pages. During this idle time between the

two requests of users, the web browser will obtained extra web page hint provided by the server. The server side has a prediction engine.

When a user requests for web pages, the algorithm at the server predicted extra pages after preprocessing at the server. The proposed architecture is function like Firefox Mozilla where user surf for web pages and extra hints of web pages are predicted at the browser side. On the backend service, the server web pages are predicted at the server side and these pages are sent to the web browser on user's request [14]. The experiment proved that the user latency is reduced 14% on increasing the network traffic just 8% [14].

2.9 Intelligent Web Prefetcher

In [15], a new model for web caching and prefetching is proposed. In this method, the web mining is used to discover the pattern in the web structure. In the web structure, the web pages are interlinked with one another. The user behavior to access these pages for the similar link made a pattern. This kind of pattern is helped in finding the user behavior. In the model, the correlations of the web pages can be found through the association rules. The correlations of the web pages are well enough to predict a pattern for the knowledge discovery in the web.

In the proposed model some of the children of pages are prefetched. These prefetched pages are used for web browser as prediction. The proxy server is used in the proposed model. The proxy server has the following module for preprocessing of the data [15].

- Web log Module
- Structure Module
- User Session Cache
- Prefetcher Module
- Proxy Application Module
- Prefetching Queue

It is assumed that the web server keeps the full cooperation with the proxy server.

2.9.1 Web log Module

From the web mining module, the server logs are cleaned and preprocessed for getting the data in a format that is required by a web miner to generate the association rules [15]. Association rules are formed from the web miner that has a range of support and confidence. This module is operated at specific time of Intervals to formulate the association rules.

2.9.2 Structure Module

The structure module makes the site graph. It contains the usage count of each page of a web. The usage of page is counted after a specific time intervals. It also keeps the information whether to cache a page or not [15].

2.9.3 User Session Cache

Cache is used to store the session of each user and when the session ends, the session of user is also deleted. Information of each page usage is kept in table of the cache [15].

2.9.4 Prefetcher Module

The prefetcher module consists of a requests history list, a prefetching queue, and a prefetching thread. Depending upon the prefetching scheme, a prefetching queue is constructed for storing a list of the documents/web pages that are required to be prefetched. The prefetcher thread uses a url from the prefetching queue and finds out either the page is in the cache [15].

If web page is not in the cache, prefetcher thread takes the page from the main server and stores in the cache. The history list of each user session, the prefetching queue and the prefetcher thread are deleted after a specific time intervals [15].

2.9.5 Proxy Application Module

The proxy application module responses to the requests made by the user. It receives the user request and finds out whether the page requested by the user is in the cache or not. If the page is found in the cache, the proxy application module sends it back the page to the user who requested it [15]. If the requested page by the user is not present in the proxy server cache, the proxy server sent request to server and takes the page from the server and sends pages back the to user. Proxy application module stores the cached page to its cache after sending back it to user

[15]. It stores url of the user requests in history list of user and the algorithm updates the prefetching queue in the next step.

2.9.6 Prefetching Queue

A prefetching queue is contains for urls to stores according to the prefetching scheme. The history list is for storing requests made by the user for a Session [15]. In association rules prefetching scheme, all rules present in the request's history list are recognized. The each recognized rule is stored in the prefetching queue. The model made the prefetching intelligent [15].

In the reference [16], a new technique of web prefetching is described. In this techniques graph algorithm is used for prediction of web pages. The double dependency graph algorithm is based on dependency graph algorithm. Implementation of algorithm proved that it reduced user's latency comparing with the dependency graph algorithm. The functionality of double dependency graph is given below.

In figure 2.1 Html1, Html2, Image1 and Image2 are the interlinked web objects with arc weights w=1, 0.5. Suppose a user accessed a page Html1; the double dependency graph [16] will construct a graph. This construction of graph is based on the dependent neighbor nodes. The prediction of web object will be saved in data depending upon threshold value. Here the threshold value is 2. The user can stored two web objects which are interlinked with htlml1 because the threshold is set as two. The value for the prediction depends on counting the arc weights and the dependency of interlinked objects.

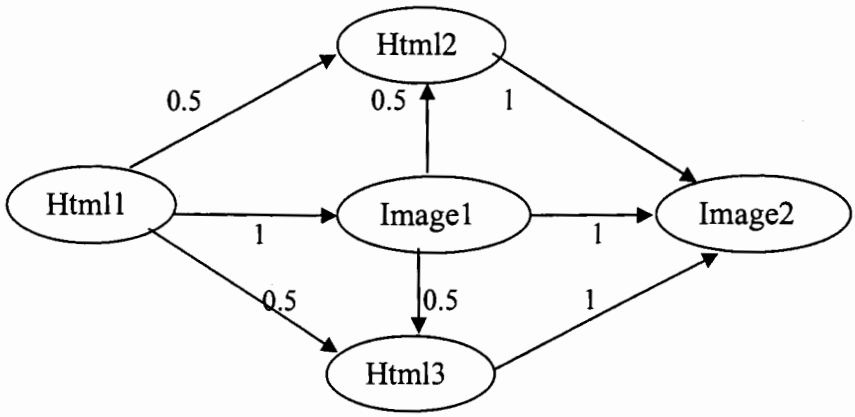

Figure 2.1: State of the graph of the data dependency graph algorithm with weight

Html1, Image1, Html2 and Image2 by accessed by one user; Html1, Image1, Html3 and Image2 accessed by other user. All the web objects information is stored by double dependency graph algorithm.

The Double Dependency Graph (DDG) algorithm keeps information about the dependency among web pages accessed by the user. It keeps the information of the dependency of objects to the same page and dependent to an object of another page [16].

2.10 Web Prefetch Performance Evaluation

In the reference [17], we studied a framework for implementing web prefetching in real environment. The proposal in the paper is based on to compare two algorithms for prediction in web prefetching. The comparison of the algorithms made in [17] provided the evaluation of performance in real environment. The name of the proposal which is used for performance evaluation of two algorithms is Delfos proposal.

In the real environment, Delfos is a technique used for measuring the performance of algorithm. The proposal of this paper compared the two well known algorithms. The Delfos current version is integration of apache 2 web server and Mozilla web browser. Any web server or any web client is suitable for working with Delfos [17].

The architecture of the framework is divided in three parts.

- Web Client
- Web Server
- Prediction Engine

2.10.1 Web Client

In the client a web browser is used such as in the Firefox Mozilla and a tool is used for capturing the session of user [17].

2.10.2 Web Server

The web log file of server is used as an input for the algorithm. The web log file provided the useful information. The information provided by the web logs is statistical. The statistics can be used to calculate the performance, prediction and accuracy or usefulness of algorithms [17].

2.10.3 Prediction Engine

The prediction engine is provided hint for the users. The prediction engine contained the algorithm for the preprocessing of data. The input data required for the prediction is the web logs files. This input data for the prediction engine is called training data set. The training data set is the web logs of the servers. The prediction engine after preprocessing the training data set would make a prediction about the hint of pages for the users. It is not necessary to delay for user to access the web server training data [17].

The comparison of two algorithms is made in the experiment that is prediction by partial match dependency graph algorithm. The behavior of prediction by partial match algorithm in the experiment is traced in the following way [17].

The prediction by partial match algorithm will build a tree when it received a request from user and stored a list of contexts for each session of user. For providing the hints of pages to user, it takes both the tree and the context of list of a session [17].

For example, when a request is made a new child node to all the nodes is added in the contexts list and the occurrence of child node is updated if there is already similar node available. Each node consists of two values: url and number of occurrences. When a session ends the list of contexts is deleted [17].

In the data dependency graph algorithm, if a user requests for the web page, the algorithm builds a dependency graph of its accessed page. In the graph the accessed pattern of the pages are stored. The graph keeps the node for each page that is accessed by the user. An Arc from A to B is drawn in the graph, if A is accessed before the B in w is look ahead window size [17]. The w is also called as weight/cost of the arc. If user accessed a page frequently then the occurrences of that page with w eight arcs is more than the other. The use of similar kind of experiment is helpful in benchmarking the web prefetching mechanism.

2.11 Referrer Graph Algorithm for Web Prediction

In the reference [18], a new algorithm is implemented known as referrer graph for the prediction in the web. The algorithm consumed much less memory and resources with efficient and accurate results as compared to the other algorithms.

In the present time, there is higher availability of bandwidth connections for the users. They face latency when navigating a web is increased due to increases of overloaded network traffic, transfer of long messages from server to the clients.

When a user request for a web page, based on the user request the Referrer Graph algorithm construct a graph. The procedure for construction of a graph is given as:

Each requested web page by the user is represented by a node in the graph. For each web page request of user that reports its referrer, an arc or arrow is built from the previous (referred) node to the current node in the graph. The Referrer Graph takes the relationship in account for a session and construction of Referrer Graph for a website depends upon the web structure.

The prediction of a web page in the referrer Graph is made on the base of direct reference from one page to the other. For each user request if the node contains already in the graph, then its occurrence is increased. For creating a node in the graph, an arc with occurrence one is set from successor to predecessor node [18].

Example

The example given below describes the client's session; the graph is constructed by running the Referrer Graph algorithm. At the end, hints are provided by the prediction algorithm for the user based on probability. In the figure given below, the client requested for the pages is for a session. This table is constructed for client in one session after processing the Referrer Graph algorithm [18].

The nodes of the graph are labeled with urls and their probabilities of occurrence are labeled with arc from one node to the other. The probability of each request is calculated through the learning process. The learning process is an algorithm that is used to take out the probability of each node of the graph when a user requests for web pages in a session [18].

The hierarchy consists on the parent node known as p.html, c1.gif, c2.pdf, c3.html; c4.html, d1.jpg, d2.png d3.html and e.jpg are sub nodes of the tree hierarchy.

The figure: 2.2 depicted the Referrer Graph construction for one user session.

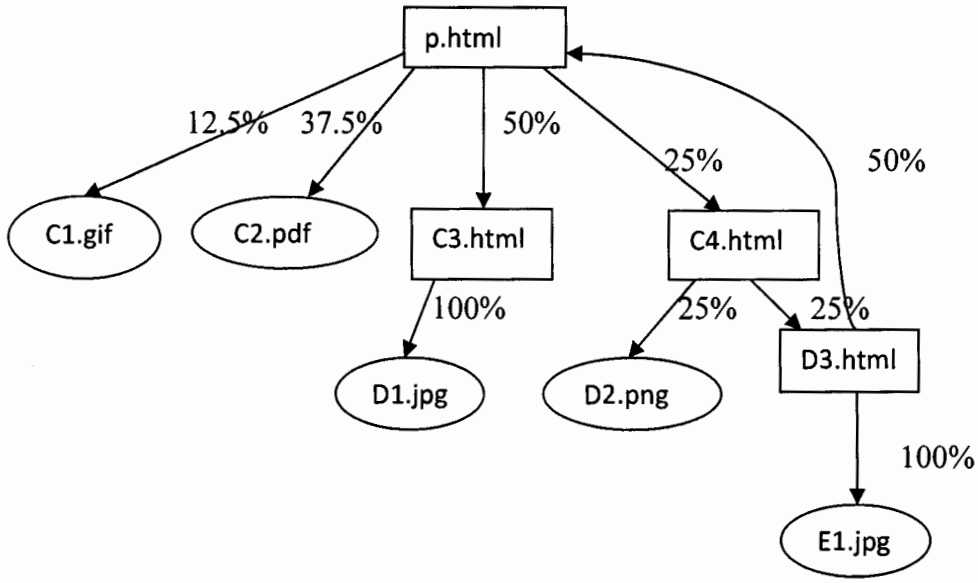

Figure: 2.2 Referrer Graph algorithms after simple navigation by the user for a session [18]

Client Session for the Web Request

The nodes of the graph stored by the Referrer Graph algorithm are given as:

URI requested	URI of Referrer
P.html	-
C1.gif	P.html
C2.pdf	P.html
C4.html	P.html
D2.png	C4.html
D3.html	C4.html
E3.jpg	D3.html
P.html	-
C1.gif	P.html
C3.html	P.html
D1.gif	C3.html
P.html	www.search.com
C1.gif	P.htmlC4.html

P.html	-
D2.png	C4.html
D3.html	C4.html
E.jpg	D3.html
P.html	D3.html

If the Client Request for URI, the hints are

URI	**Hints URI and Probability**
P.html	C4.html 50%, C3.html 25%, D2.png 50%, D1.jpg 25%
C4.html	D3.html 100%, E.jpg 100%
D3.html	P1.html 50%, C1.gif 37.5%, C2.pdf 12.5%
C3.html	-

The example showed the working of RG algorithm

This example showed the how the probability is counted for each page of a web when a user accessed the web pages during one navigational session [18].

The experiment to implement the referrer graph algorithm is carried out through a set of simulated clients with prefetching ability. The clients made requests to the simulated server. The prediction engine consists of different algorithms that provide the hint for the simulated client's requests. The latency of the user is reduced and the comparing with other algorithms, Referrer Graph algorithm consumed less computational and memory space [18].

2.12 Comparison of Algorithms

In the reference [19], the network user faced latency due to the increase of network traffic. It is important to reduce the latency of user. One way of reducing the latency is web caching, web caching has limited benefits. For reducing the user latency another technique is web prefetching. It reduces the user's access time while getting documents from the server but it required the bandwidth and increase of traffic. For the web prefetching to reduce the user latency, the important is that to prefetch the document in advance or not.

This paper described the comparison of different algorithms. The comparison of web prefetching algorithms is based on the consumption of memory and the hit ratio. The hit ratio is the documents that are required by the users. The comparison will be taken these two points into the account [19].

2.12.1 Prefetching by Caching

One of the ways is to cache the entire pages from server to client, but this technique is not good because it consumed 100% of the memory [19] and also caused increase in network traffic. In the real environment, it is not possible to implement it.

2.12.2 Prefetching by Popularity

Another way to prefetch the documents is prefetching by popularity. In this method the top ten documents of a web server are prefetched by the proxy server. These pages are available at proxy for the user when they demand for it. When this technique is implemented, the result showed that it reduced the 40% of the client accessed the pages at the cost of 60% increased in network traffic [19]. The results are achieved through a simulation of client server environment.

2.12.3 Prefetching by Life Time

When an object is changed consecutively for time interval, it is least frequently changes, then it is said to be longer time interval for that object [19]. By using this scheme the network traffic cost is minimized but user latency is large because it produced low hit ratio of accessed pages.

2.12.4 Prefetching by Good Fetch

This technique balanced a page access frequency and a page change frequency. If a user prefetched an object/page before it is referred. The user demand for the same page as it is prefetched; it is termed as Good Fetch. The algorithm calculates the probability of Good Fetch of an object and prefetched those objects whose frequency is more than others [19].

2.12.5 Prefetching by APL

In this algorithm for prefetching of web pages of user depends upon the object life time, object probability of being accessed and user request rate. We select the object which has the highest value of life time, probability and user request rate. It means that that object will be prefetched in the user cache.

By comparison of the techniques, it is seen that prefetching by popularity got the highest hit ratio, but it utilized high bandwidth. Prefetching by Good-Fetch and APL are to balance the prefetching by popularity because it utilized the low network traffic and prefetching hit ratio is

low as compared to popularity prefetching. By Lifetime fetches the long interval live web objects, replicates the objects in the user cache that required the least bandwidth as compared to other prefetching algorithm [19].

In the reference [20], Web prefetching is a useful technique for reducing the user latency and network traffic for the web services. Web caching is different from the web caching because web caching make use of the temporal locality where as web prefetching is a predictive technique in which the web pages are predicted for the future request of user .In the reference [20], integration of these two techniques that is web caching and web prefetching caused improvement of the web infrastructure.

In the reference [20], we present a clustering based prefetching technique in which a graph based clustering algorithm knows the clusters of correlated web pages that are built on the user access patterns at proxy server.

2.13 Clustering In Web Prefetching

The clustering is done by preprocess the log files. The users that requested same objects are grouped under the same group. A graph is made for each user request from one page to the other. The requests made by each user groups are presented by a weighted directed Web graph G (u, v); where u is the web page and v edge is the set of user's transitions from one web page to the others page [20]. The weight of each edge is proportional to the number of transitions in the set. The example shows the construction of graph.

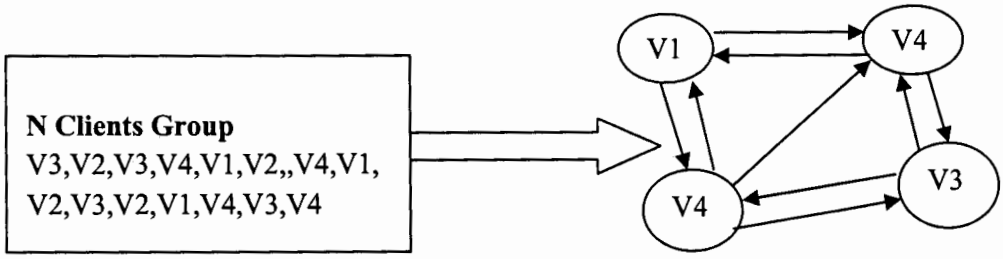

Figure 2.3: User navigational Graph

If web pages are increased, the resulting web graph that has been made on the user access pattern is unmanageable because the node of the graph increases. The association rules are made to take the transition of graph with minimum support and confidence.

2.13.1 Use of Web Cluster Algorithm

The web cluster algorithm is used to cluster the inter website pages. The user group is represented access pattern of weighted graph. The graph is partition by filtering some of the edges of the graph. The algorithm used for making the web cluster is Breath for Search (BFS). The algorithm traverses each node of the graph and keeps only those nodes of the graph that are accessed by the user through navigation [20]. The scheme of cluster is depends upon the content of cluster that the user of the group accessed.

The cluster prefetch scheme is described in the following steps.

1. A user request for a web page/web object of the site
2. The proxy server recognized the user that request for a web page according to its IP address and proxy server assigns it to one of the group of clients. In this model, the cluster of the web objects is known by the proposed prefetching scheme.

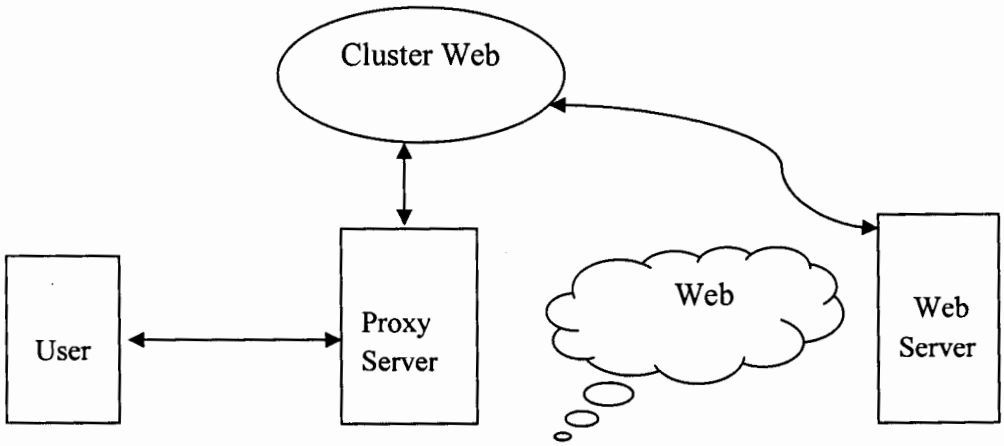

Figure: 2.4 Proposed Model for Clust Prefetch in [20]

3. The proxy prefetched all the web objects that exist in the cluster which is selected in previous step. The objects/pages are kept in the proxy server cache according to cache replacement policy.
4. The proxy sent the requested object to the user.

2.13.2 Benefits of Proposed Framework

The performance of prefetching is increased if the clusters are managed properly.

1. The cluster web prefetching scheme improved significantly the network performance.
2. The Clust Prefetch is a realistic scheme that can be implemented to any Web cache environment.
3. The algorithm that made cluster of the user request, the number of clusters does not impact the efficiency of the propose prefetching scheme because it made use of support and confidence value.
4. The proposed scheme in the reference [20] is adaptive, because it changes in the web user`s patterns of web.
5. The proposed scheme is made the web data clusters, that can be used for keeping the track of recent past request of the users.

2.14 Prediction by Popularity

A new technique of web prefetching is described in the reference [21] that improved the web prefetching at Wide Area Network (WAN). For improving the performance of web prefetching web logs server are used. The algorithm used for prefetching is "N Next Most Popular" approach for the prediction about the next page is made on preprocessing the web logs data.

2.14.1 N Next Most Popular Method

In N Next Most Popular approach, for the prediction of future request of the user a profile of user access is built at each user side. The data of the cache log is used for building the profile of user access.

The analysis of cache log data is used for the number of time a user request /visits a page and the sequence of request made by user for that page. In this way future behavior of a user and its accessed to the page is known. By the preprocessing the log data, the ranking of web pages accessed by the user and its frequency of accessed is known [21].

For implementing the N Next Most Popular approach, a list is built for every client in the WAN. This list contains the pages and it is used by the algorithm to make prediction about the

future request of user. For prediction of pages, decision algorithm is used in each client based list.

In [22] mechanism, sequential and non sequential pattern discovery of web usage is discussed. For the traversing of the node the depth for search traversing is used. Three models AR (Association Rule), SP (Sequential Pattern) and CSP (Contiguous Sequential Pattern) are used for the comparisons.

In [23], a method for weighted suffix trees is described. The user navigational prediction is online in a website or in the cache system. The method consume constant amount of space and relatively less memory space.

Web caching and web prefetching [24] is the field that improved the performance of web access in wireless environment. In this paper, the author described an algorithm for the integration of web caching and web prefetching in wireless local area network. Integrating web caching and web prefetching based on the prediction in sequence mining and it is implemented in web caching system. The experiment is carried out on Once Easy Cache system.

In [25] mobile small form factors; mobile handsets are the longer variable of the network causes latency for the user in web related activities in the mobile networks. The paper described designing and prototyping implementation of mobile interaction optimization system. This system improved mobile communication in the web based communication. The predicted user interaction sequences are used to optimize in reducing the amount of user input and user wait time using techniques of interaction short cuts, automatic text copying and form-filling, page pre-fetching.

In [26], addressed a problem of personalized of information. Transforming related to the web which is based on user profiling. Different approaches for user profiling are developed for personalizing a user context of web. There are three kinds of approaches content, collaborative and web usage filtration.

In [27], the apriori algorithm, hash tree and fuzzy are described for the web usage mining. Hash tree and Fuzzy enhanced Apriori algorithm to give the solution Crisp Boundary problem with higher optimized efficiency as compared to other algorithms. Apriori algorithm modification structure is generated by the use of hash tree algorithm. Crisp boundary problem in

the combined algorithm is overcome by our modified association. Apriori hash tree fuzzy algorithm has increased efficiency.

In [28], Web usage mining is the area of web data mining techniques for discovering usage patterns for web data. Three phase of web usage are discussed in [28] for knowing the need of web pattern in web based applications. These phases of web usage mining are preprocessing, pattern discovery and pattern analysis.

In [29], a model for an integrated web caching and web prefetching is proposed. The issues of prefetching aggressiveness for replacement policy and increased network traffic are addressed in the framework. The major of issue aggressiveness in web prefetching is solved through the prediction model based on statistical correlation between web objects.

CHAPTER # 3

Research Approach for Mining Web Logs

3 Research Approach for Mining Web Logs

Our solution to propose technique is based on the following four phases of web usage mining [1] [28] as given below:

- Data Collection
- Preprocessing
- Pattern Discovery
- Pattern analysis

3.1 Data Collection

For discovering useful knowledge from large databases [1] [28], the data set used for mining is very important. In the web usage mining, the data set could be collected in several ways. The data set can be collected from the server, client and proxy or from the operational database of the organization. This data set is based on the events like user's clicks, user's queries and user's checkout [22]. The data gathering for the web usage mining may be at server level data collection, client level collection, proxy level collection, server's monitors, network monitors and e commerce application monitors.

In our proposed model the web logs of the servers are the data that is taken from the web server for four different websites.

3.2 Preprocessing

The data set gathered from various sources is transformed for discovering the useful pattern [1]. This phase is called preprocessing [1] [28]. The preprocessing can be done for any of the three applications of Web Mining (Usage Preprocessing, Content Preprocessing and Structure Preprocessing) [22].

In the second phase we filter the web logs of the servers and remove the unwanted information from the web logs data. For example the data that do not take part into the discovery

Of hidden pattern such as jpg files is removed that contain pictures from the weblogs and then preprocessed it through algorithm.

3.3 Pattern Discovery

This phase is a data mining phase because useful hidden data/pattern is discovered in this stage [1] [28]. From the web logs after the preprocessing by an algorithm used in the model, useful hidden patterns are discovered which are the most frequently occurred Urls accessed by different users. We used modified techniques of sequential rank based selection algorithm [30] for selecting a url for prefetching purposes.

3.4 Pattern Analysis

After the pattern discovery [1] [28], the last step is to compare our results with existing technique in [7]. For analysis purpose different graphs are drawn. The results of existing algorithm and the proposed framework techniques are compared by plotting the graphs for analysis purposes.

The analysis of results is based on the three factors.

- Comparison of results in the form of snapshots
- Graphical comparison of results for analysis of pattern of resources both the techniques
- Web usage information of proposed techniques

At the end, results proved that the proposed model techniques showed higher availability of resources for the user's web resource usage.

CHAPTER # 4

Proposed Model Solution

4. Proposed Model Solution

The proposed architecture/system is consisting on client-server model.

4.1 Client Side

The client has a prefetch engine which has a set of association rules. The client side prefetch engine stores web object that is prefetched by the server side prediction engine or web log. For example, based on client's request, pages predicted by the prediction engine are stored in prefetch engine on the client side. A queue is maintained in prefetch engine that stores the predicted pages.

4.2 Server Side

The server side consists of two modules namely prediction engine and web log. Whenever a client sends a request for a web page/web object both modules will predict extra pages which will be stored in the prefetch engine.

4.3 Prediction Engine

Prediction engine has defined algorithm for predicting web pages/web objects. In the proposed model, we used sequential rank base selection enhanced technique and compare the result with the other existing algorithm [7]. The sequential rank base selection [30] selects only a web object from cluster grouping and sends it to the prefetch engine.

Proposed Architecture diagram

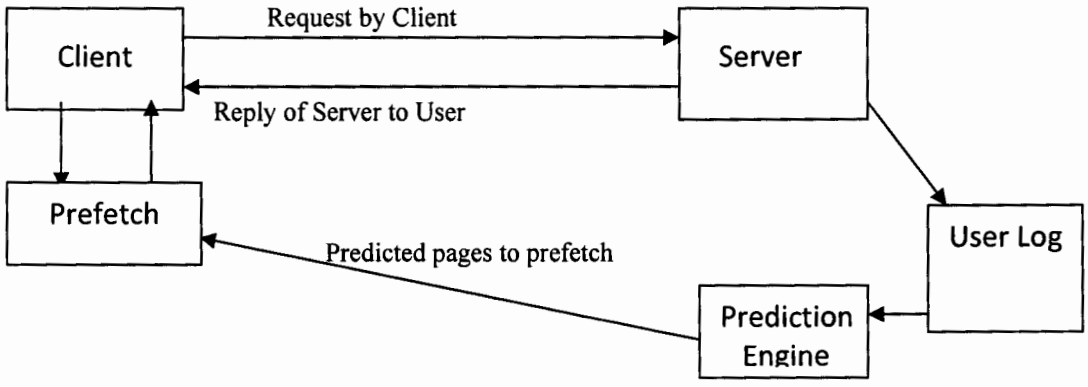

Figure 4.1: Framework for Web Prefetching

4.4 Web Logging

Whenever a client requests, the server maintains a log of this request. The server maintains the log contains of each client.

4.5 Flow of Data during the Prefetching Mechanism

The figure given below represents the client, server and prediction mechanism. The server side contains the data for the prefetching of user. When user requests for a web page from server, the server will return that page and extra resource for availability of users. For the prediction of pages, an algorithm is used to calculate probability. The flow of data during the prefetching mechanism is given below.

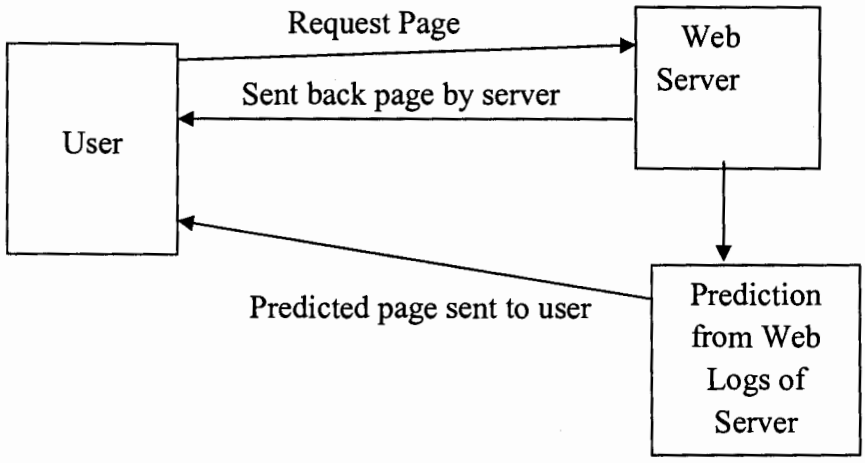

Figure 4.2: Flow of data between user and server during web prefetching mechanism

When a user requests for web page to a server, the server will send the page back to the user as reply and the prediction Engine predicts a page for prefetching purposes for user.

4.6 Advantage of Proposed Framework

- The proposed model reduces the user latency by prefetching the pages local to the user. The page will be available local to users.
- Web Usage Information

To achieve the web usage advantage, we have modified the sequential rank based selection algorithm [30]. From the modified technique; we have generated the clustering of different pages of web log data set. From the clustering of different pages and by counting the frequency of each similar page in clustered data set would generate the web usage information of server's used resources by the users.

4.7 Implementation of Proposed Model Technique

The implementation of the idea is based on the proposed technique. The existing model in the reference [7] is an algorithm. We will compare the results of algorithm in reference [7] and proposed technique by us based on [30]. The comparison of results showed that our proposed technique is efficient in web page prediction.

First I implemented the algorithm in the reference [7].The data set we have used is the web logs of the servers. The data set consists of information about the web logs of a server. The web logs consist of the information about user session, IP addresses, the link that user accessed and referrer urls.

In the experiment, we have taken the data set of web logs of four web servers. The data sets are the web logs of International Islamic University Islamabad, Twibuzz, Hazara University Mansehra and NUST University web servers. The experiment is done in a controlled environment in which we have taken the results of both existing algorithm [7] and proposed technique.

By the analysis of graphs and results for existing algorithm and proposed technique proved that our method is efficient in web page prediction and occupies less space as compared to the algorithm used in [7]. The proposed technique by us for web prefetching in real environment also gives the information about the web usage of web site of a server.

4.8 Proposed Technique

Steps to mine the data in proposed architecture are given below:

Method:

Input: Web log data Set

Out Put: Clustering/Similar Pages, Max Cluster, Median

I= 1; //Initialize Variable

Pages P= {P1, P2… PK} //Initialize Variable for storing web page

Data Set S= {Cli… Cln} //Data set for storing Clusters of Pages

For (K=2; k<= n; K++) // variable k denotes the 2nd Page in the weblog

If (Pk==Pi) //p1=p2 // Mapping of String

Then Update into Relevant Cluster; // Cluster in the relevant data set is updated by putting Pk in Cli

Update Data Set;

Else

K=K+1 // i=2

Cli = Pk

Update S, by putting Cli in S

Repeat for Loop

Return: Data Set S // Out put updated Version Set of Clusters

End

Go to

Out Put: Max Cluster

Input= CL= {Cl1, cl2, Cl3 …. Cl n}

K= the cluster that is searched by using sequential Rank base Selection algorithm

q= Constant (division of S)

M=median of S

Cluster Selection by Sequential Rank Based Selection(S, k) [30]

1. If |S|<q

   Then SORT |S| and return K Cluster

   Else

   Divide |S| into q subsequence and sort each subsequence and find Median of each S or q.

2. Find the Median "M" of "S"

3. Create 3 Subsequences according to Median of "S"

   L: Clusters of S that are < m

   E: Cluster of S that are =m

   G: Clusters of S that are >m

4. If |L|>= k

   Then return Select (L, K)

   Else

   If |L|+ |E|>=K

   Then return m

   Else return Select (G, K-|L|-|E|)

   End IF

   Return Max Cluster K

   Return Cluster Data Set

Working of Model Technique:

1. Finding Frequent Pages from web logs data

The association Rule is to first find out most frequent pages in the web logs of a server accessed by user's i-e, the same pages requested by different users are stored in the clustered. Identify these pages by their Urls and separate them.

2. Clustering of Pages

The next step is to cluster the different pages of weblog server accessed by web users.

The same pages are clustered in a group according to their URL address. Different pages are clustered in different groups according to their groups and URL address.

3. Calculate the Frequency of a page accessed by the users

From the web logs server data, count the frequently accessed Web server pages/urls by users and put them in the relevant group.

4. Select a Page/URL for Prediction

Run the Sequential Rank based Selection Techniques to result out the prediction of a page for a user in clusters grouping.

The prediction is made on clustering of a page how much time the users will access that page from the weblog. The frequency of the web page is the number of times a user has accessed that web page from weblogs for each cluster. Selected web page and its clustered of that page output as result. It also displayed list of the clustered of each page accessed by different users.

The implementation of the proposed techniques can be viewed by two ways.

- **Single User**

When a user accessed different pages of a web site, the proposed techniques will make a table of clustered pages/urls accessed by a user. Different pages/urls accessed by single user are clustered according to their group to which they belong and the based on frequent accessed is output as result.

The technique gives prediction of a page/url to prefetch for web prefetching mechanism for a single user.

- **Multiple User**

When multiple users of a web site accessed the pages of website, the proposed model techniques will give the information of web usage of different users of a website.

This kind of informative model is useful for the business purposes because the proposed model gives information about which parts of a website are most frequently used by different users.

4.9 Conceptual Model for Implementation

Before Implementation, The model is understood as:

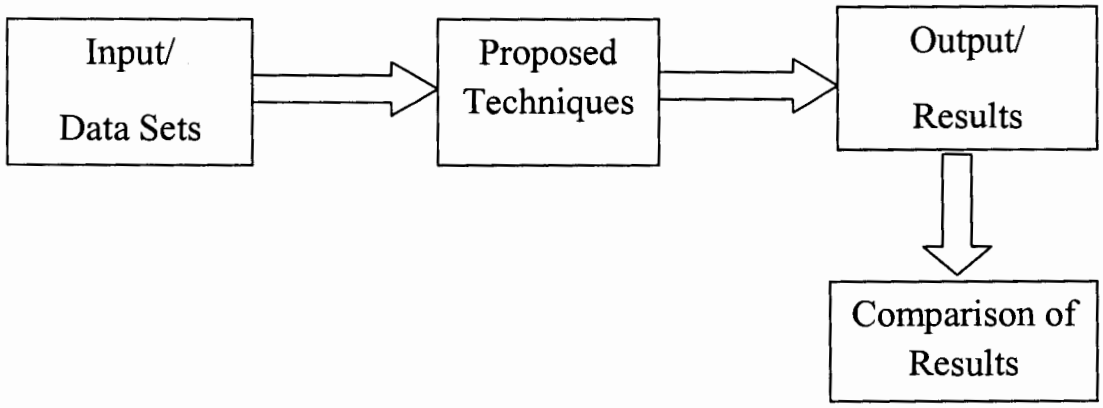

Figure: 4.4 Concept of Implementation

Our implementation will be in this scenario, the input will be the web log data file of four web servers. The existing algorithm and proposed techniques will run and calculate the predicted web pages. At the end, results proved the efficiency of our proposed techniques.

4.10 Example to Implement Techniques

Given a web log server data set of Hazara university server's website known as www.hu.edu.pk, in the weblog server data, the IP address indicates different domains that made a request for a page from a web server. The session is the time in web logs of a server when the user made the requests for that web page. Session is actually the time when user is accessed that web page. In a domain, the domain users and other domain users can access the web page from a web server. The domain users are identified from their IP addresses in the web logs server data

The next is the page/url address when a user gets that web page and the last address is the referrer address. Referrer address is the address in the web log, when a user requested for the

web page from one page for the next of a web site. The browser information about a user from which it requested is also indicated by the browser type in web log data such as Firefox Mozilla. The given below is the table:

IP Address	Time		URL
192.168.10.9	[14/May/2010:00:10:34	0700]	GET www.mail.hu.edu.pk/HTTP/1.1
192.168.10.9	[14/May/2010:00:10:35	0700]	GET www.hu.edu.pk/search_degree.php/HTTP/1.1
192.168.10.9	[13/May/20010:00:10:36	0700]	GET www.hu.edu.pk/viewfaculty.php?id=6/HTTP/1.1
192.168.10.9	[13/May/2010:00:10:43	0700]	GET www.mail.hu.edu.pk/HTTP/1.1
192.168.10.9	[13/May/2010:00:35:33	0700]	GET www.hu.edu.pk/lib.php/HTTP/1.1
192.168.10.9	[13/May/2010:00:36:20	0700]	GET www.hu.edu.pk/abt_museum.php/HTTP/1.1
192.168.10.9	[13/May/2010:00:37:09	0700]	GET www.mail.hu.edu.pk/HTTP/1.1
192.168.10.9	[13/May/2010:00:40:56	0700]	GET www.hu.edu.pk/search_degree.php/HTTP/1.1
192.168.10.9	[13/May/2010:00:40:58	0700]	GET www.hu.edu.pk/lib.php/HTTP/1.1
192.168.10.9	[13/May/2010:00:59:04	0700]	GET www.hu.edu.pk/math.php/HTTP/1.1
192.168.10.9	[13/May/2010:00:59:08	0700]	GET www.hu.edu.pk/search_degree.php/HTTP/1.1
192.168.10.9	[13/May/2010:01:29:37	0700]	GET www.hu.edu.pk/math.php/HTTP/1.1
192.168.10.9	[13/May/2010:01:29:41	0700]	GET www.hu.edu.pk/search_degree.php/HTTP/1.1
192.168.10.9	[13/May/2010:01:42:42	0700]	GET www.hu.edu.pk/math.php/HTTP/1.1
192.168.10.9	[13/May/2010:20:53:42	0700]	GET www.hu.edu.pk/nassef_fund.php/HTTP/1.1
192.168.10.9	[13/May/2010:20:53:49	0700]	GET www.hu.edu.pk/math.php/HTTP/1.1

Figure: 4.4 Structure of web log of server

In the above example, the web log of the server contains the IP address of the domain's user who have made request for web pages, the second column contains the session that is actually time when user accessed that web pages. The third column contains the urls of the pages that are got by the users. The each row in the table is called as web object. The embedded object table which is initially used known as training data set.

- **Finding the Frequent page**

For implementation, we find out the frequent pages/urls accessed by the users.

- **Clustering of Similar URLs/Pages and Calculating Frequency**

Occurrence of each similar url is assigned a value to make a cluster. When a single user accessed different urls, they are clustered according to their group. Count the frequency in a single cluster for web page/url. In this way calculating the frequency is easy in each cluster. The figure given below depicts the occurrence of each url.

P.No	URL in Name	Cluster of Each Page
1	GET www.mail.hu.edu.pk/HTTP/1.1	3
2	GET www.hu.edu.pk/search_degree.php/HTTP/1.1	4
3	GET www.hu.edu.pk/viewfaculty.php?id=6/HTTP/1.1	1
4	GET www.hu.edu.pk/lib.php/HTTP/1.1	2
5	GET www.hu.edu.pk/abt_museum.php/HTTP/1.1	1
6	GET www.hu.edu.pk/math.php/HTTP/1.1	4
7	GET www.hu.edu.pk/nassef_fund.php/HTTP/1.1	1

Table: 4.5 Cluster in web log of a server

In this way the frequency of a web page is calculated as in above table.

The GETwww.mail.hu.edu.pk/HTTP/1.1 is requested by the user three times, so its occurrence is three because it is found three times in the web log of Hazara University web server in cluster. The url Getwww.hu.edu.pk/search_degree.php/HTTP/1.1 and Getwww.hu.edu.pk/math.php/ HTTP/1.1 are requested by the user four times, so their cluster contains occurrence is four in the cluster. The Get: www.hu.edu.pk/viewfaculaty.php, id=http 1.1, Get: www.hu.edu.pk/abt museuj.php/HTTP 1.1 and Get: www.hu.edu.pk/nassed_fund.php/HTTP 1.1 is one in the cluster.

- **Select a Page for Prediction**

Run the proposed techniques to result out the prediction of a page for a user in cluster grouping. The selection of a page from the cluster data will predict a page at the web log of a server.

4.10.2 Implementation of proposed Technique

Before implementation, we rank the pages/urls according to similarity. The similar pages from the web logs that make a cluster are stored in sorted way. The median is based on the cluster. One cluster of same page is actually is the rank of that page. Different pages will form different cluster. If a page is accessed by the user most frequently in the cluster, its probability to access the web page will be high. So the probability of that page for the prediction is more.

- **Software Used for Implementation**

The implementation of our concept can be done in Java, Mat lab and PHP languages. We used the Visual Studio C#.net and MS Excel and Note Pad for implementation of proposed framework technique. The data sets for proposed framework technique are taken from servers. These are the web server of International Islamic University web server, National University of science and Technology web server, Hazara University web server and Tiwibuzz web server.

- **Data of Web Logs Servers**

The data sets taken web servers are in notepad format. The data in the notepad is seen to be raw. In order change the data sets into understandable format. We have converted the data sets of server into MS Excel tables. The procedure for changing the data of web logs data file from Notepad into MS Excel table is given below:

- **Change the Data of the Web log Server file in the Excel Table**

The web log data file of server is in unreadable format in Note Pad. The data file of the web server is seen to be raw because it is not easy to understand the web server data file in Note Pad for anyone. For our convenient we have converted web data files into MS excel table. The procedure of saving a web data file is given:

First open the MS Excel

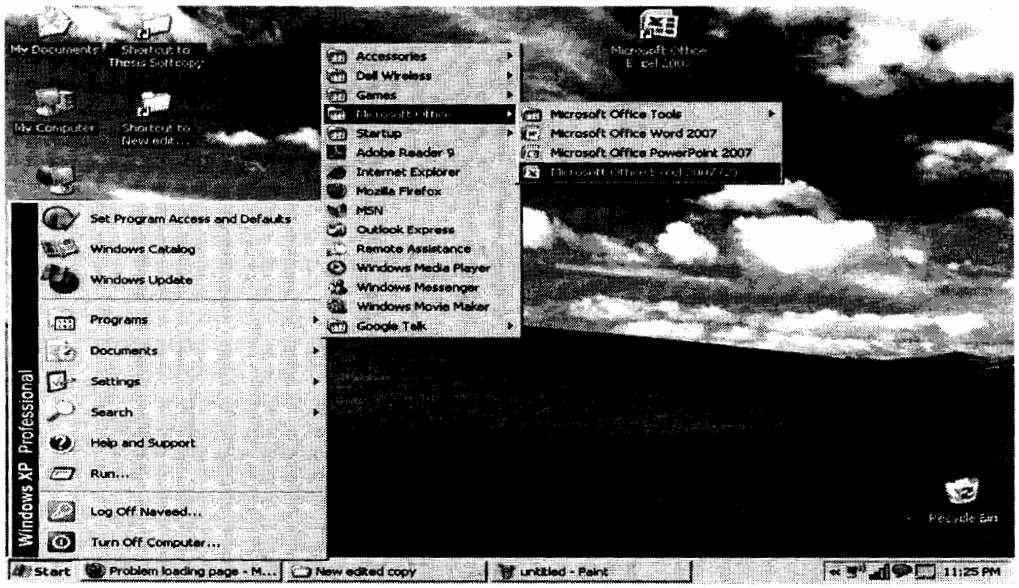

Figure: 4.10.1 Open web log data set File Saving in MS Excel

- **Browse the path of web server data file name in MS Excel**

Click on Home and Click on Open File

Give path of data file

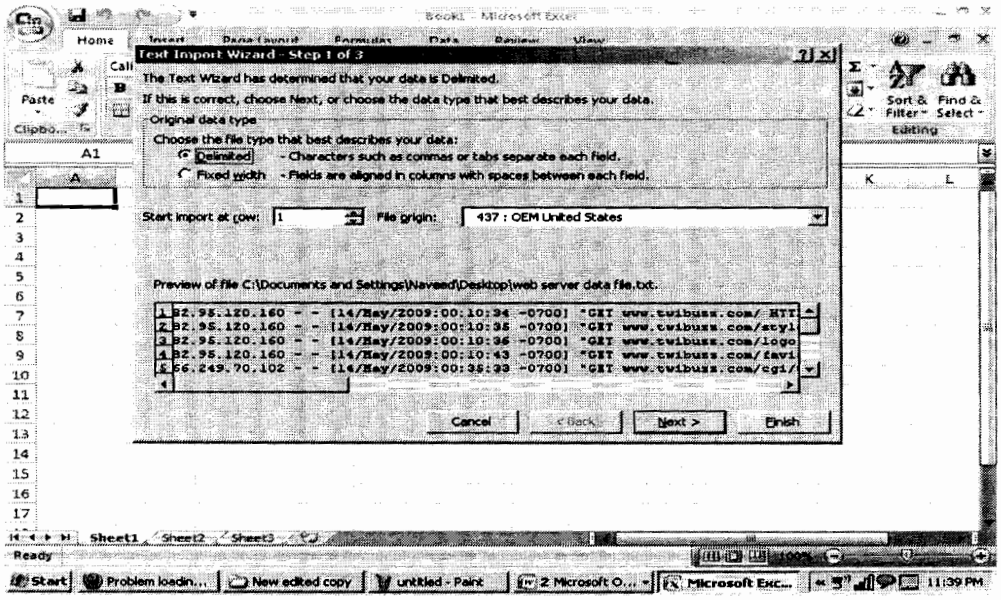

Figure: 4.10.2 Web log data file saving in MS Excel

Selected the space delimited and then clicked next and Finish. Then Save the web server data file.

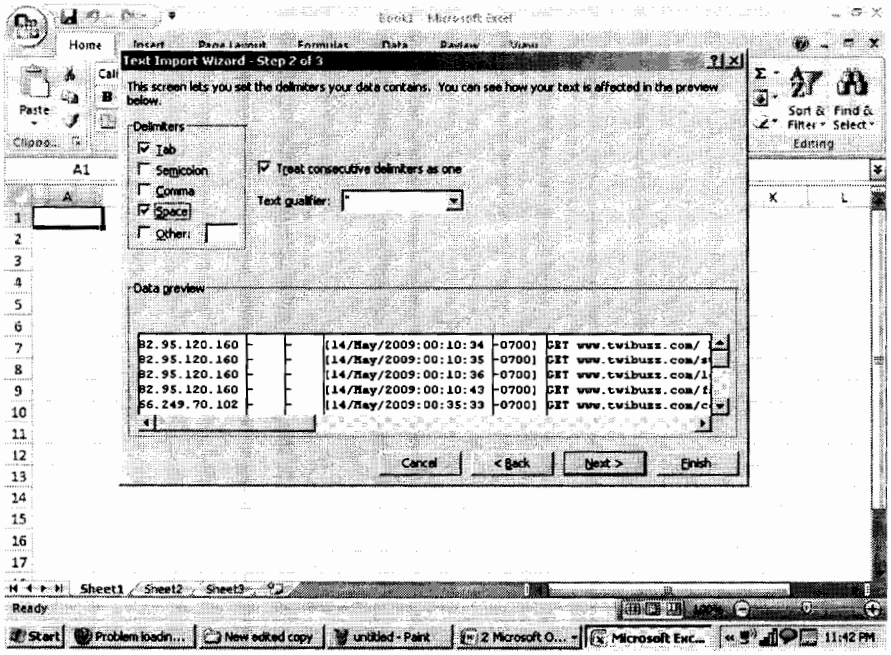

Figure: 4.10.3 Web log data File Saving in MS Excel

After Finish the data look in logical way. Then save the in the name web server data file. This procedure is repeated for all the four data set file of web servers.

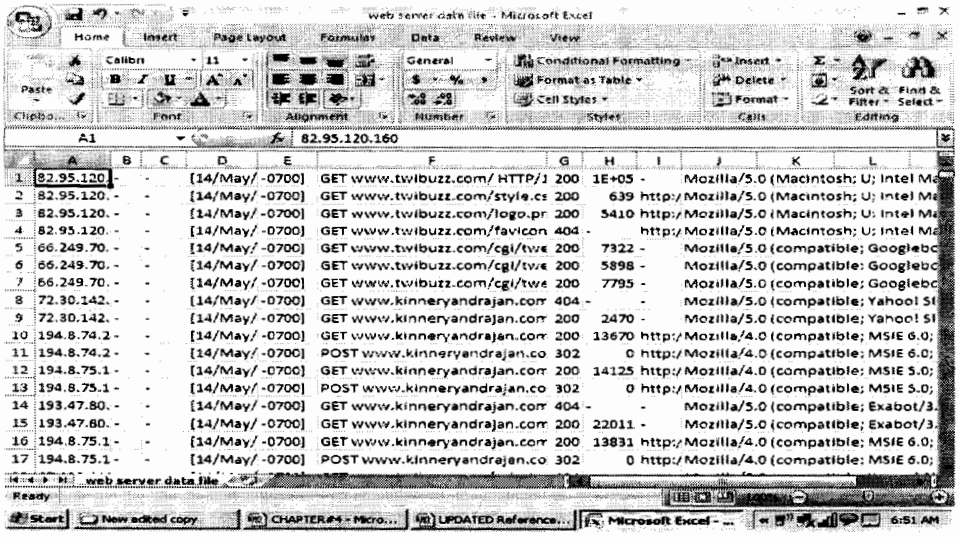

Figure: 4.10.4 Web log data File Saving in MS Excel

The first column denotes IP address, second and third column used as separator of the data. Each row of the table is called objects. The table is also known as training data sets. The training data set of all the four servers file are tested on the existing algorithm and proposed model techniques.

- **Finding the Frequent of Page and Rule For Mining in Proposed Techniques**

Web log contains the urls of different user in the web log file data. These urls are accessed by different users when they made request for web pages. The frequent of the same pages accessed by the users are stored in a table. The frequent urls data table acts as rules for Mining the web logs data. One url accessed by the different user represents a cluster for that url.

- **Prediction of page in the cluster**

Run the Sequential Rank based Selection algorithm to choose url from different urls with its frequent number accessed by different users. The median is calculated according to the steps given in the algorithm and one url is selected for prefetching.

Four data files of four different web servers are used as input for existing algorithm and proposed framework technique.

Let P1, P2, P3, P4... P_{n-1}, P_n are the web pages that contained in the weblogs of the server. The pages are accessed by a user. As logs contained the information about which pages that are accessed by the users. The prediction to prefetch a page is based on clustering of those pages. Our implementation contained one to one mapping of a user.

The step four in proposed technique is the selection of a cluster from set of cluster. The Sequential Rank based Selection will select used for a web page for prefetching purposes from the cluster data. The algorithm is used for only to predicts a Url/web page with highest accessed by the user from each cluster data set.

4.11 Existing Algorithm

The algorithm used in the reference [7] is given below:

Input W: access sequence of current session; hc: confidence threshold;

T: prediction rule table; EOT of T Outputs S: a set of predicted documents with probability

Method:

```
S = Empty set
For i=W.length down to 1
  {
C = rules in T with the form: W -> X (confx)
For each rule in C
If confx>= hc
 Then
{
 Add X {(confx) into S;
For each embedded object Oi of X in table E,
Add O| (conf,} into S;
}
IF S is not empty then return S;
W = W with the first letter removed;
}
Return empty set;
```

The algorithm has the list of sequence of current sessions of web pages

Confidence Threshold is the value of the web page which is maximum probability of a web page to be predicted.

T for EOT is a table which contains the rule for the prediction of web pages; S is the output with the probability of all the web pages that are to be predicted by the algorithm.

First S is empty, by the rules the some urls are predicted by running the algorithm. The W is the variable which removes the prediction of the pages by the algorithm from S.

At first when algorithm runs, it will take the values previous session and current session according to the rule. The rules are defined initially. If the values according to the rules are greater than or equal to the threshold, the algorithm will take that value. This value of predicted pages from EOT will be brought into the Variable S. After running the algorithm the S will be set as empty.

CHAPTER # 5

Conclusion & Results

5.1 Comparison of Results

The data sets of International Islamic University web server, Hazara university web server, National University web server and Twibuzz web logs server are taken. This data set is tested on the existing algorithm [7] and the proposed framework technique.

Our results are compared on basis of following factors:

1. Results of both Existing algorithm/Proposed technique for same web server data in form of Snapshots

The results of existing techniques are taken for the same input web log data set. The Interfaces shows the existing solution and proposed model technique solution.

2. Graphical Representation of results produced by existing algorithm and proposed framework techniques.

For same input of four weblogs from the server, the results of the web log data set of International Islamic University, Web log data set of Hazara University web server, Web log data set of National University of Science and Technology and web log data set of Twitter Twibuzz web server are given in figure 5.3, 5.6, 5.10 and 5.14.

3. Web Usage Information

The clustering of the pages in the proposed framework techniques and the calculation of the frequency from each clustered group has produced the web usage information of the server.

The four data sets of web servers are used for both the existing algorithm [7] and framework at server side proposed by us as input are given below:

- Web log Data Set of International Islamic University Web server
- Web log Data Set of Hazara University Web server
- Web log Data Set of National University of Science and Technology Web server
- Web log Data Set of Twitter Twibuzz Web server

5.1.1 Web Log of International Islamic University Web Server

The data set of International Islamic University web log server contains the records/web objects of three thousand. The data set is taken from different users in the same domain. The result of the data set of International Islamic University web server for the existing algorithm in the reference [7] is given below:

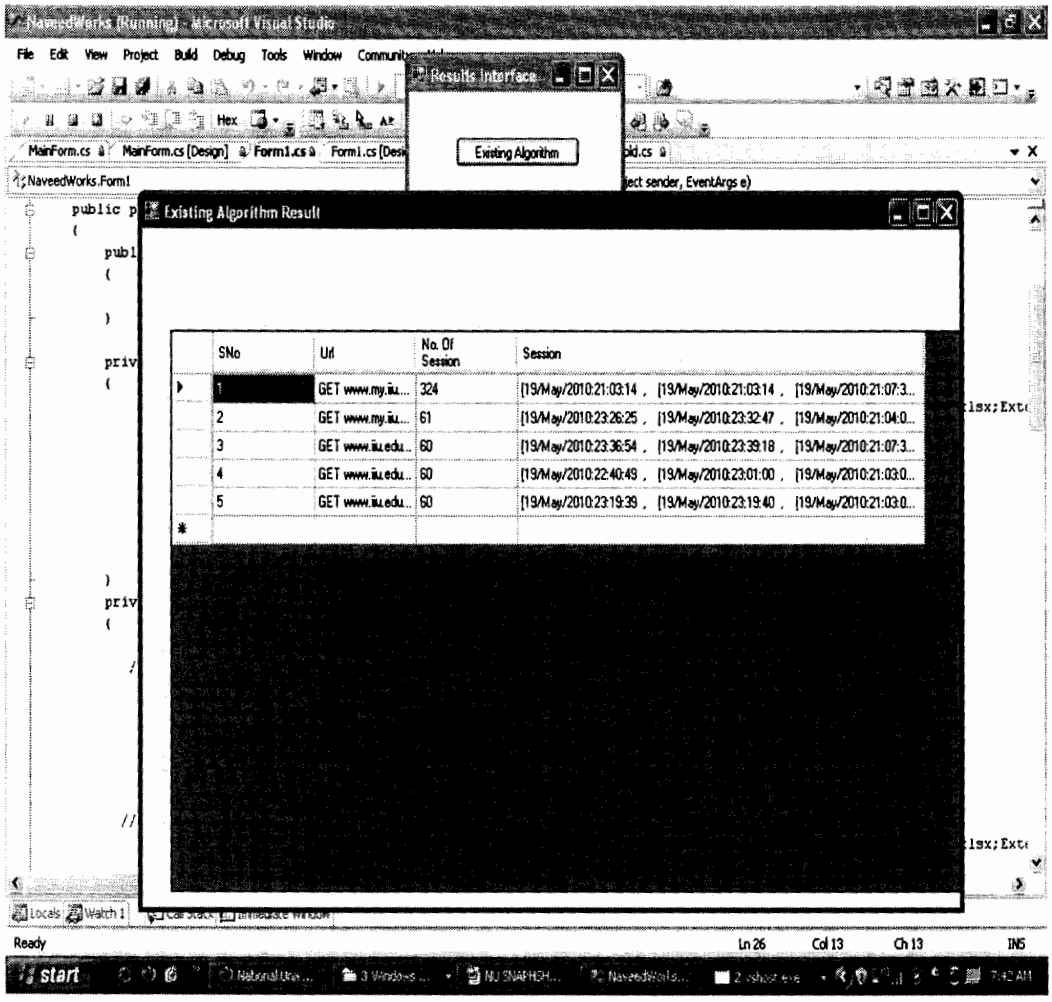

Figure: 5.1 Results of existing algorithm using International Islamic University data set

In this experiment the threshold is to show only first five web objects prediction made by existing algorithm in the reference [7]. In the algorithm the session column is used as input of each url.

The rules are made for calculating the sessions for same url. In this way the user's sessions are counted against one web object. The threshold value is set to show only the number

of url's session for topmost five values that has the maximum probability to prefetch for a user. For a user five values above threshold are predicted by algorithm.

Data Predicted by Existing Algorithm

P.No	Page Address	Session
1	GET www.my.iiu.edu.pk/Libraries/CentralLibrary/tabid/266/Default.aspx	324
2	GET www.my.iiu.edu.pk/About/Administration/RectorOffice/tabid/74/Default.aspx	61
3	GET www.iiu.edu.pk/events/admission/spring2009/admission_international.html//	60
4	GET www.iiu.edu.pk/News/may_10.html#dawah_academy/HTTP/1.1	60
5	GET www.iiu.edu.pk/News/may_10.html#library/HTTP/1.1	60

Table: 5.1 List of results output of existing algorithm using Islamic University data set

The algorithm shows the analysis of top most five predicted web objects/pages.

The results produced for same web log server input used in our proposed techniques is given as:

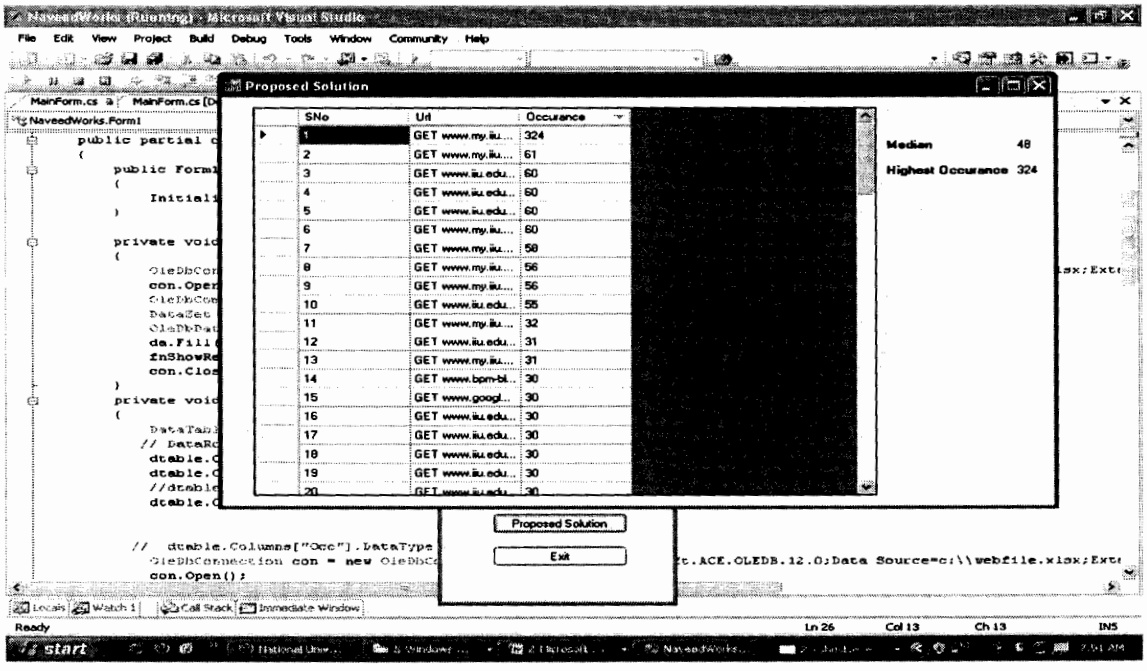

Figure: 5.2 Results of Proposed Techniques using International Islamic University data set

The techniques proposed by us showed the following results with the probability for each page usage in each cluster. The predicted page to prefetch for a user is the one that has highest accessed by the user in each cluster. The additional information is useful in sense that it provides the usage of each web page of a web server.

The results produced by proposed model techniques are given below:

P. No	Page Address	Cluster
1	GET www.my.iiu.edu.pk/Libraries/CentralLibrary/tabid/266/Default.aspx	324
2	GET www.my.iiu.edu.pk/About/Administration/RectorOffice/tabid/74/Default.aspx	61
3	GET www.iiu.edu.pk/events/admission/spring2009/admission_international.html/HTTP/1.1	60
4	GET www.iiu.edu.pk/News/may_10.html#dawah_academy/HTTP/1.1	60
5	GET www.iiu.edu.pk/News/may_10.html#library/HTTP/1.1	60
6	GETwww.my.iiu.edu.pk/Faculties/Shariaandlaw/tabid/189/Default.aspx/HTTP/1.1	60
7	GET www.my.iiu.edu.pk/Academies/DawahAcademy/tabid/318/Default.aspx/HTTP/1.1	58
8	GET www.my.iiu.edu.pk/Academies/Shariah/tabid/326/Default.aspx/HTTP/1.1	56
9	GET www.my.iiu.edu.pk/Libraries/DawahLibrary/tabid/246/Default.aspx/HTTP/1.1	56
10	GET www.iiu.edu.pk/events/admission/fall2010/default.htm/HTTP/1.1	55
11	GET www.my.iiu.edu.pk/Academies/Shariah/FacultyStaff/tabid/328/Default.aspx/1.1	32
12	GET www.iiu.edu.pk/iiu_videos_1.htm/HTTP/S1.1	31
13	GETwww.my.iiu.edu.pk/Faculties/BasicAppliedSciences/Departments/ EnvrionmentalScience/tabid/159/Default.aspx/HTTP/1.1	31
14	GET www.bpm-blogpage.blogspot.com/HTTP/1.1	30
15	GETwww.google.com/a/iiu.edu.pk/ServiceLogin?service= =2/HTTP/1.1	30
16	GETwww.iiu.edu.pk/downloads/announcement/WelfareFund_loan_2010.pdf/HTTP/1.1	30
17	GET www.iiu.edu.pk/events/admission/spring2009/listofprograms_offered.htm/HTTP/1.1	30
18	GET www.iiu.edu.pk/events/newsletter/default.html/HTTP/1.1	30
19	GET www.iiu.edu.pk/events/seminar/eng_conference/call_for_paper.html/HTTP/1.1	30
20	GET www.iiu.edu.pk/iiu_videos_5.htm/HTTP/1.1	30

21	GET www.iiu.edu.pk/News/may_10.html#competition/HTTP/1.1	30
22	GET www.iiu.edu.pk/News/may_10.html#iiu_students/HTTP/1.1	30
23	GET www.iiu.rozee.pk/HTTP/1.1	30

Table: 5.2 Lists of results using Islamic university data set

The difference between Proposed Technique from the clustering and existing algorithm is that modified sequential rank base selection techniques predicted one page at time and the rule defined by us produced the results of each page's probabilities. In existing algorithm only those pages are predicted which has threshold value equal or above on the basis of session.

When a user requests for page, the modified sequential Rank base Selection algorithm techniques will prefetch only a page for the user in advance from the cluster data.

- **Graphical Comparison**

The graph is constructed on the basis of results produced by the existing algorithm and proposed framework techniques. The existing algorithm predicted the session. The session is time, when the user accessed the pages. On the basis of session the algorithm in the existing techniques predicted the pages. These sessions are counted for each user request. The pages are representing along x-axis by the numbers.

The pages predicted in existing algorithm are represented by a line of existing algorithm in the graph. The proposed model techniques prediction of pages is represented by line result output framework. The proposed framework techniques provide higher availability of resources over the existing algorithm.

To view the data sets produced by proposed techniques and existing algorithm can be seen in the graph. In the results produced by existing and proposed model techniques, the web page position is according to page number as predicted by the existing and proposed model techniques.

Applying the data set produced by both algorithms as input for graph, we got:

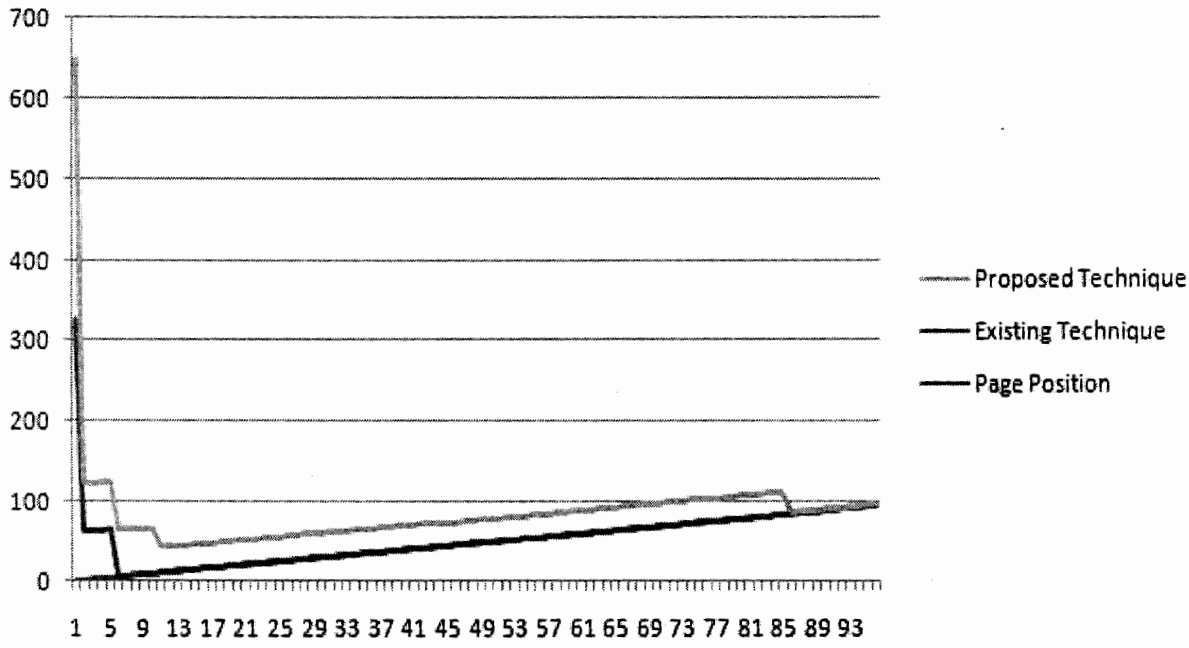

Figure: 5.3 Comparisons of results for International Islamic University web data set

X-axis represents the urls/pages and y-axis represents the users accessed these urls.

The graphical comparisons of the existing algorithm and proposed techniques for weblogs data set of Hazara University showed that proposed technique has provided higher availability of resources for the users over the existing scheme in web prefetching mechanism. The graph showed that prediction pages produced by proposed technique are equal or greater than existing algorithm.

Calculating Reduced Latency in Existing Technique and Proposed Model Techniques Performance:

Percentage $_{\text{(Web Object)}}$ = (Each Accessed Session/Total number of session in weblog)*100

The session is the Time in which a user request. This is recorded in the weblog data file. The total session is calculated by the existing algorithm [7], the number of request's session made by the user in the weblogs.

Percentage $_{\text{(Web Object)}}$ = (Each Predicted Cluster Frequency/Total Objects in web logs)*100

Overall Performance= Total sum of all percentage (Existing techniques and proposed model technique)

Summary of Results

Page No	Percentage of Existing Technique	Percentage of Proposed Model Technique	Page No	Percentage of Proposed Model Technique
1	10.8%	10.8%	13	1.03%
2	2.03%	2.03%	14	0.46667%
3	2%	2%	15	0.46667%
4	2%	2%	16	0.46667%
5	2%	2%	17	0.46667%
6	-	2%	18	0.46667%
7	-	1.93%	19	0.46667%
8	-	1.86%	20	0.46667%
9	-	1.86%	21	0.46667%
10	-	1.83%	22	0.46667%
11	-	1.06%	23	0.46667%
12	-	1.03%	-	-
Over all Performance	**18.3%**	**-**	**-**	**36.96%**

Table: 5.2 Performance of Existing and Proposed Techniques using Islamic university data set

The result proved that proposed technique is efficient and improved the overall performance of web prefetching system.

5.1.2 Web Log Data Set of Hazara University web server

The web log of the Hazara University contained the thirty five hundred records/web objects. The weblogs is taken from Hazara University web server. The results of the both proposed technique for server and existing algorithm is given by the snapshots. The web log of Hazara university server is used as input for the both proposed technique and existing algorithm. The results of the algorithm in [7] are given below:

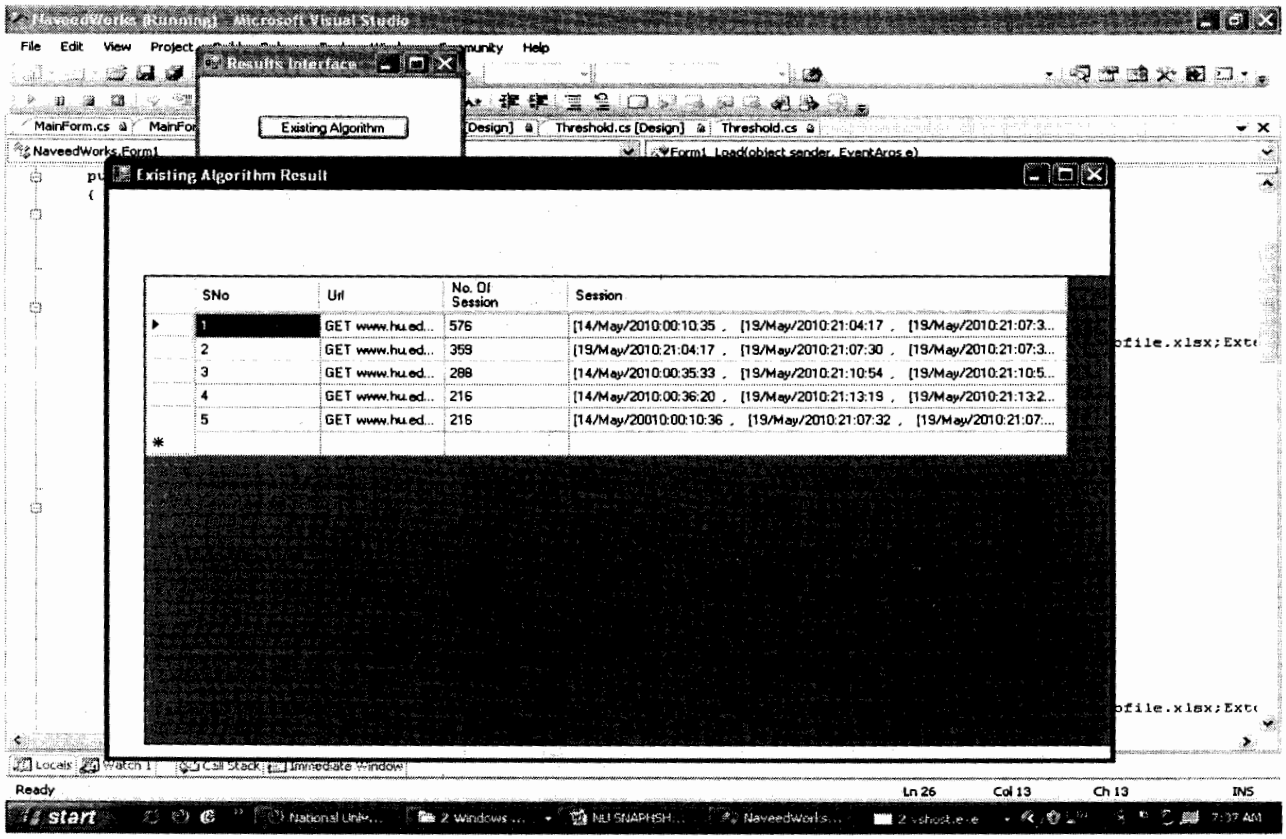

Figure: 5.4 Results of existing algorithm using Hazara university data set

The results shown in figure are number of sessions for each page. The counting the session produced the total number of sessions contained for web prefetching. The results of existing algorithm are given below:

P.No	Page Address	Session
1	GET www.hu.edu.pk/search_degree.php/HTTP/1.1	576
2	GET www.hu.edu.pk/index.php/HTTP/1.1	359
3	GET www.hu.edu.pk/lib.php/HTTP/1.1	288
4	GET www.hu.edu.pk/abt_museum.php/HTTP/1.1	216
5	GET www.hu.edu.pk/viewfaculty.php?id=6/HTTP/1.1	216

Table: 5.4 List of results of existing algorithm

The results of proposed technique for the same input are given below. The association Rule define for proposed technique showed the frequency for each page`s usage by the user in each cluster. The algorithm produced the result for with highest probability.

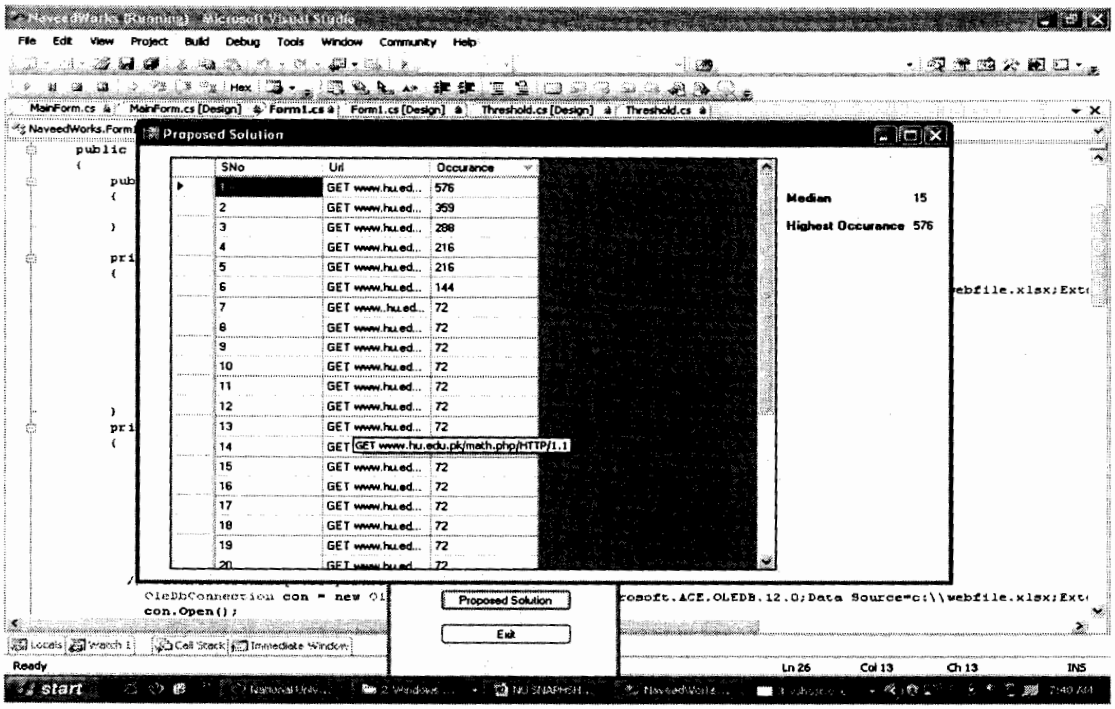

Figure: 5.5 Results of proposed techniques

The results of proposed techniques are given below:

P.No	Page Address	Cluster
1	GET www.hu.edu.pk/search_degree.php/HTTP/1.1	576
2	GET www.hu.edu.pk/index.php/HTTP/1.1	359
3	GET www.hu.edu.pk/lib.php/HTTP/1.1	288
4	GET www.hu.edu.pk/abt_museum.php/HTTP/1.1	216
5	GET www.hu.edu.pk/viewfaculty.php?id=6/HTTP/1.1	216
6	GET www.hu.edu.pk/campus_museum.php/HTTP/1.1	144
7	GET www.hu.edu.pk/vcpage.php/HTTP/1.1	72
8	GET www.hu.edu.pk/b_chem.php/HTTP/1.1	72

9	GET www.hu.edu.pk/clht.php/HTTP/1.1	72
10	GET www.hu.edu.pk/feepay.php/HTTP/1.1	72
11	GET www.hu.edu.pk/history.php/HTTP/1.1	72
12	GET www.hu.edu.pk/hu_events.php/HTTP/1.1	72
13	GET www.hu.edu.pk/huadministration.php/HTTP/1.1	72
14	GET www.hu.edu.pk/math.php/HTTP/1.1	72
15	GET www.hu.edu.pk/nassef_fund.php/HTTP/1.1	72
16	GET www.hu.edu.pk/news_con_artical.php/HTTP/1.1	72
17	GET www.hu.edu.pk/prog.php/HTTP/1.1	72
18	GET www.hu.edu.pk/res_planning.php/HTTP/1.1	72
19	GET www.hu.edu.pk/s_criteria.php/HTTP/1.1	72
20	GET www.hu.edu.pk/search_mamsc09.php/HTTP/1.1	72
21	GET www.hu.edu.pk/view_urdunews.php/HTTP/1.1	72
22	GET www.hu.edu.pk/viewfaculty.php?id=11/HTTP/1.1	72
23	GET www.hu.edu.pk/viewfaculty.php?id=12/HTTP/1.1	72
24	GET www.hu.edu.pk/viewfaculty.php?id=16/HTTP/1.1	72
25	GET www.hu.edu.pk/viewfaculty.php?id=18/HTTP/1.1	72
26	GET www.hu.edu.pk/viewfaculty.php?id=21/HTTP/1.1	72
27	GET www.hu.edu.pk/viewfaculty.php?id=23/HTTP/1.1	72
28	GET www.hu.edu.pk/viewfaculty.php?id=5/HTTP/1.1	72
29	GET www.hu.edu.pk/viewfaculty.php?id=9/HTTP/1.1	72
30	GET www.mail.hu.edu.pk/HTTP/1.1	72
31	**Total Cluster**	30

Table: 5.5 Results of proposed techniques by using Hazara University data set

- **Graphical Comparison**

The graph is constructed on the basis of results produced by the existing algorithm and proposed framework techniques. The existing algorithm predicted the session. The session is time when the user accessed the pages. On the basis of session the algorithm in the existing techniques predicted the page.

The page position is representing page numbering along x-axis according to prediction of pages by existing algorithm and proposed techniques. The pages predicted in existing algorithm are represented by a line of existing algorithm in x-y coordinate.

The proposed model techniques prediction of pages is represented by line result output framework in x-y coordinate. The proposed framework techniques provide higher availability of resources over the existing algorithm as shown in graph for the same for Hazara University data set. The results of both proposed technique for server side and existing algorithm can be shown by a graph.

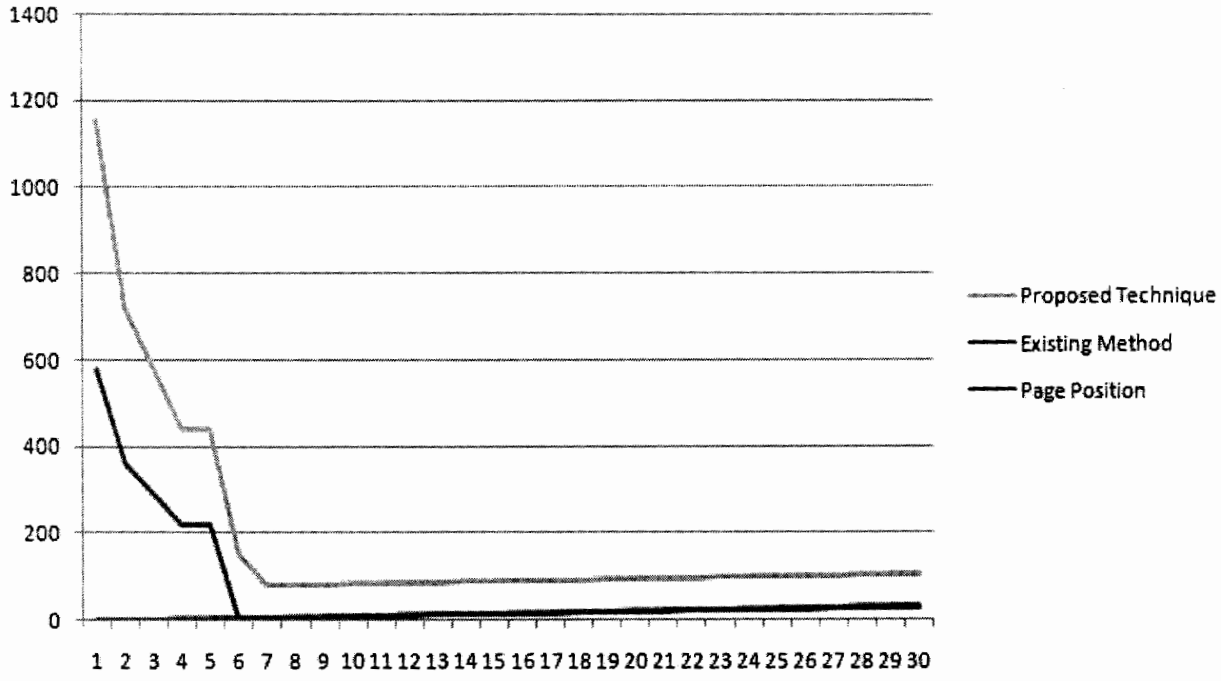

Figure: 5.6 Comparisons of results by using Hazara University data set

The graph values along x-axis are the pages/URLs and along y-axis the availability of the pages accessed for users.

The graph shows that proposed technique is efficient in prediction of web pages/records as compared to existing algorithm in the reference [7]. By applying the formula we got:

Summary of Results

Page No	Percentage of Existing	Percentage of Proposed Model	Page No	Percentage of Proposed Model
1	16.45%	16.45%	16	2.05%
2	10.25%	10.25%	17	2.05%
3	8.22%	8.22%	18	2.05%
4	6.17%	6.17%	19	2.05%
5	6.17%	6.17%	20	2.05%
6	-	4.11%	21	2.05%
7	-	2.05%	22	2.05%
8	-	2.05%	23	2.05%
9	-	2.05%	24	2.05%
10	-	2.05%	25	2.05%
11	-	2.05%	26	2.05%
12	-	2.05%	27	2.05%
13	-	2.05%	28	2.05%
14	-	2.05%	29	2.05%
15	-	2.05%	30	2.05%
Over all Performance	**47.26%**	-	-	**96.46%**

Table: 5.7 Performance of Existing and Proposed Techniques using Hazara university data set

5.1.3 Web log Data Set of National University of Science and Technology

The data set is taken from the web server of nust.edu.pk and it contains the web object of four thousands. The result of existing algorithm is given below:

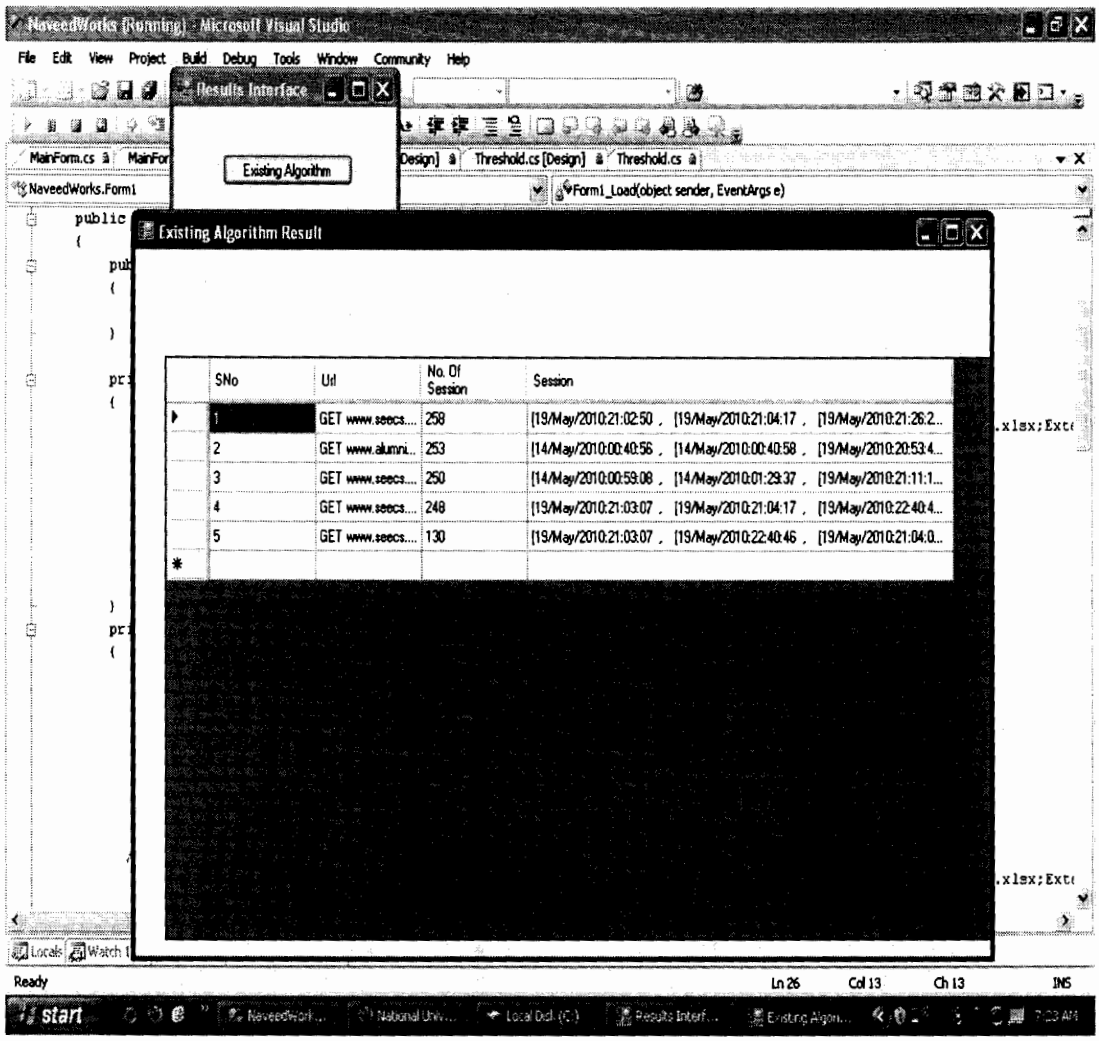

Figure: 5.8 Result of existing algorithm using NUST data set

The results data is given below:

P.No	Page Address	Session
1	GET www.seecs.nust.edu.pk/research_groups/industry_linkage.php/HTTP1.1	258
2	GET www.alumni.nust.edu.pk/AlumniHouse-FunctionsRepresentatives.aspx/HTTP1.1	253
3	GET www.seecs.nust.edu.pk/academics/doc/HTTP1.1	250
4	GET www.hu.edu.pk/abt_museum.php/HTTP/1.1	248
5	GET www.seecs.nust.edu.pk/academics/ee/index.php/HTTP1.1	130

Table: 5.8 Results of existing Algorithm by using NUST university data set

Output of proposed technique is given below in the form of snapshot

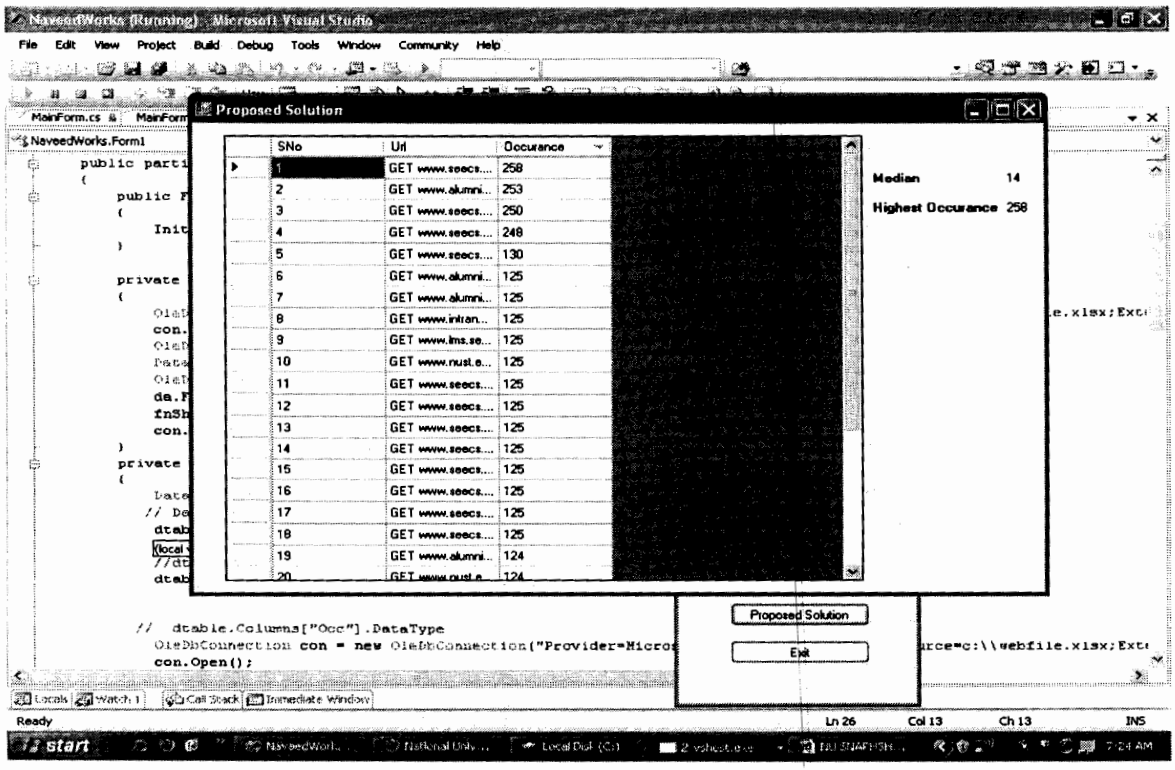

Figure: 5.9 Results of proposed Techniques by using NUST university data set

The results of proposed frame work for the same data set is given below:

P.No	Page Address	Cluster
1	GET www.seecs.nust.edu.pk/research_groups/industry_linkage.php/HTTP1.1	258
2	GET www.alumni.nust.edu.pk/AlumniHouse-FunctionsRepresentatives.aspx/HTTP1.1	253
3	GET www.seecs.nust.edu.pk/academics/doc/HTTP1.1	250
4	GET www.seecs.nust.edu.pk/about_niit/uc.php/HTTP1.1	248
5	GET www.seecs.nust.edu.pk/academics/ee/index.php/HTTP1.1	130
6	GET www.alumni.nust.edu.pk/AlumniHouse-Objectives.aspx/HTTP1.1	125
7	GET www.alumni.nust.edu.pk/Events.aspx/HTTP1.1	125
8	GETwww.intranet.seecs.edu.pk/index.php?title=Special:Userlogin&returntotitle=/HTTP1.1	125

9	GET www.lms.seecs.edu.pk/HTTP1.1	125
10	GET www.nust.edu.pk/usr/PhD-Prog-Medical-Sciences.aspx/HTTP1.1	125
11	GET www.seecs.nust.edu.pk/academics/degrees_offered.php/HTTP1.1	125
12	GET www.seecs.nust.edu.pk/academics/ee/hod.php/HTTP1.1	125
13	GET www.seecs.nust.edu.pk/campus_life/student_bodies/HTTP1.1	125
14	GET www.seecs.nust.edu.pk/library/index.php/HTTP1.1	125
15	GETwww.seecs.nust.edu.pk/Seminars_workshops/ Seminars_workshops.php/HTTP1.1	125
16	GET www.seecs.nust.edu.pk/student_resources/exam_section.php/HTTP1.1	125
17	GET www.seecs.nust.edu.pk/student_resources/exam_section/ ms_policy/index.php/HTTP1.1	125
18	GETwww.seecs.nust.edu.pk/student_resources/exam_section/ results/results.php/HTTP1.1	125
19	GET www.alumni.nust.edu.pk/HTTP1.1	120
20	GET www.nust.edu.pk/usr/PG-Prog-Mathematics-Prog-Wise.aspx/HTTP1.1	120
21	GET www.nust.edu.pk/usr/UG-Dates-to-Remember.aspx/HTTP1.1	120
22	GET www.seecs.nust.edu.pk/about_niit/downloads.php/HTTP1.1	120
23	GET www.seecs.nust.edu.pk/academics/ee/HTTP1.1	120
24	GET www.seecs.nust.edu.pk/admissions/scholarships.php/HTTP1.1	120
25	GET www.seecs.nust.edu.pk/its/HTTP1.1	120
26	GET www.seecs.nust.edu.pk/research_groups/research.php/HTTP1.1	115
27	GET www.nust.edu.pk/HTTP1.1/HTTP1.1	100
28	GET www.seecs.nust.edu.pk/academics/faculty_bs.php/HTTP1.1	50
29	Total Cluster	28

Table: 5.9 Results of Proposed Techniques by using NUST university data set

- **Graphical Comparison**

The graph is constructed on the basis of results produced by the existing algorithm and proposed framework techniques. The existing algorithm predicted the session. The session is time when the user accessed the pages. On the basis of session the algorithm in the existing techniques predicted the page.

In the graph, page position is representing page numbering along x-axis according to prediction of pages by existing algorithm and proposed techniques. The pages predicted in existing algorithm are represented by a line of Existing Algorithm in x-y coordinate. The proposed model techniques prediction of pages is represented by line Proposed Solution in x-y coordinate. The proposed framework techniques provide higher availability of resources over the existing algorithm as represented in graph for the same for NUST university data set.

The data sets results produced by proposed technique and existing algorithm can be seen in the graph. In the above results, it is given that the web page position is according to its occurrence.

Applying the data set results, produced by existing algorithms and proposed technique as input for graph we got:

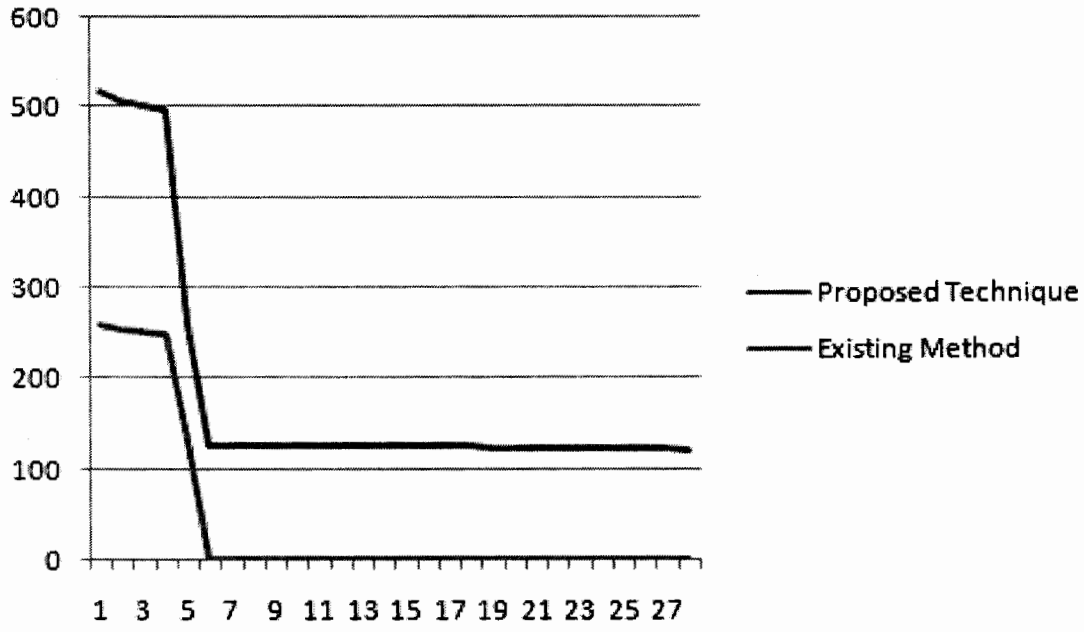

Figure: 5.10 Comparisons of results by using NUST university data set

- **Graphical Comparison**

The graph is constructed on the basis of results produced by the existing algorithm and proposed framework techniques. The existing algorithm predicted the session. The session is time when the user accessed the pages. On the basis of session the algorithm in the existing techniques predicted the page.

In the graph, page position is representing page numbering along x-axis according to prediction of pages by existing algorithm and proposed techniques. The pages predicted in existing algorithm are represented by a line of Existing Algorithm in x-y coordinate. The proposed model techniques prediction of pages is represented by line Proposed Solution in x-y coordinate. The proposed framework techniques provide higher availability of resources over the existing algorithm as represented in graph for the same for NUST university data set.

The data sets results produced by proposed technique and existing algorithm can be seen in the graph. In the above results, it is given that the web page position is according to its occurrence.

Applying the data set results, produced by existing algorithms and proposed technique as input for graph we got:

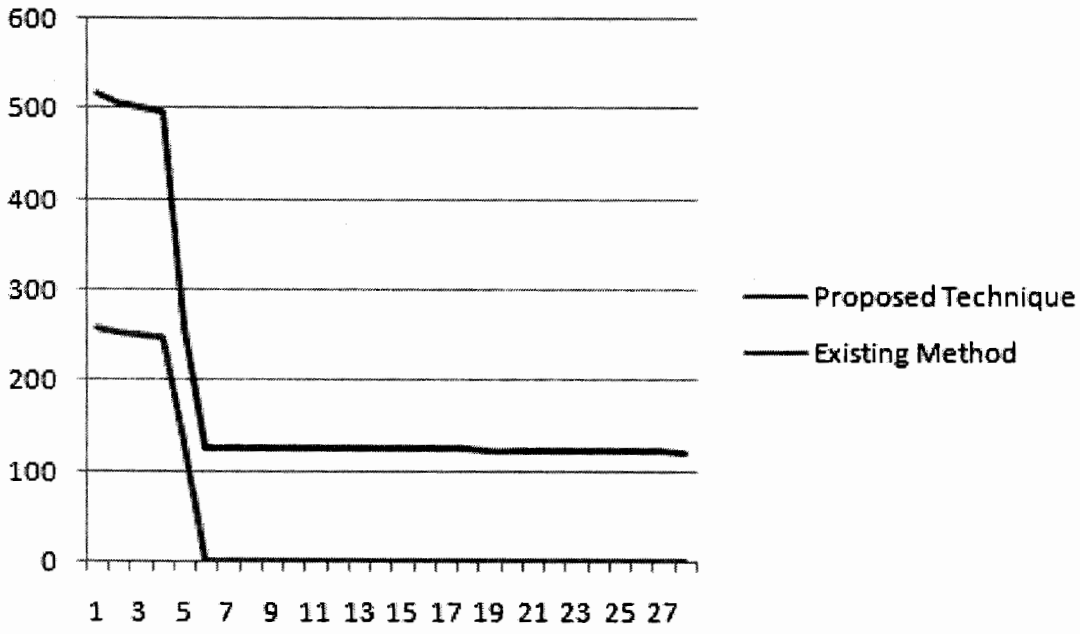

Figure: 5.10 Comparisons of results by using NUST university data set

X-axis represents the urls/pages; y-axis represents the users accessed these urls.

Proposed techniques showed the clustering of each page while the existing algorithm depends on the threshold value. By applying the formula for evaluating the performance we got:

Summary of Results

Page No	Percentage of Existing	Percentage of Proposed Model	Page No	Percentage of Proposed Model
1	6.45%	6.45%	15	3.125%
2	6.32%	6.32%	16	3.125%
3	6.25%	6.25%	17	3.125%
4	6.2%	6.2%	18	3.125%
5	3.25%	3.25%	19	3%
6	-	3.125%	20	3%
7	-	3.125%	21	3%
8	-	3.125%	22	3%
9	-	3.125%	23	3%
10	-	3.125%	24	3%
11	-	3.125%	25	3%
12	-	3.125%	26	2.8%
13	-	3.125%	27	2.5%
14	-	3.125%	28	1.25%
Performance	**28.47%**			**96.715%**

Table: 5.11 Performance of Existing and Proposed Techniques using NUST data set

5.1.4 Web log Data Set of Twitter Web server

The data set is taken from a web server of a site www.twibuzz.com. The data sets contained the web objects of twenty nine hundred and twenty. The result of existing algorithm for weblogs data of Twibuzz server in the form of snapshot is given below:

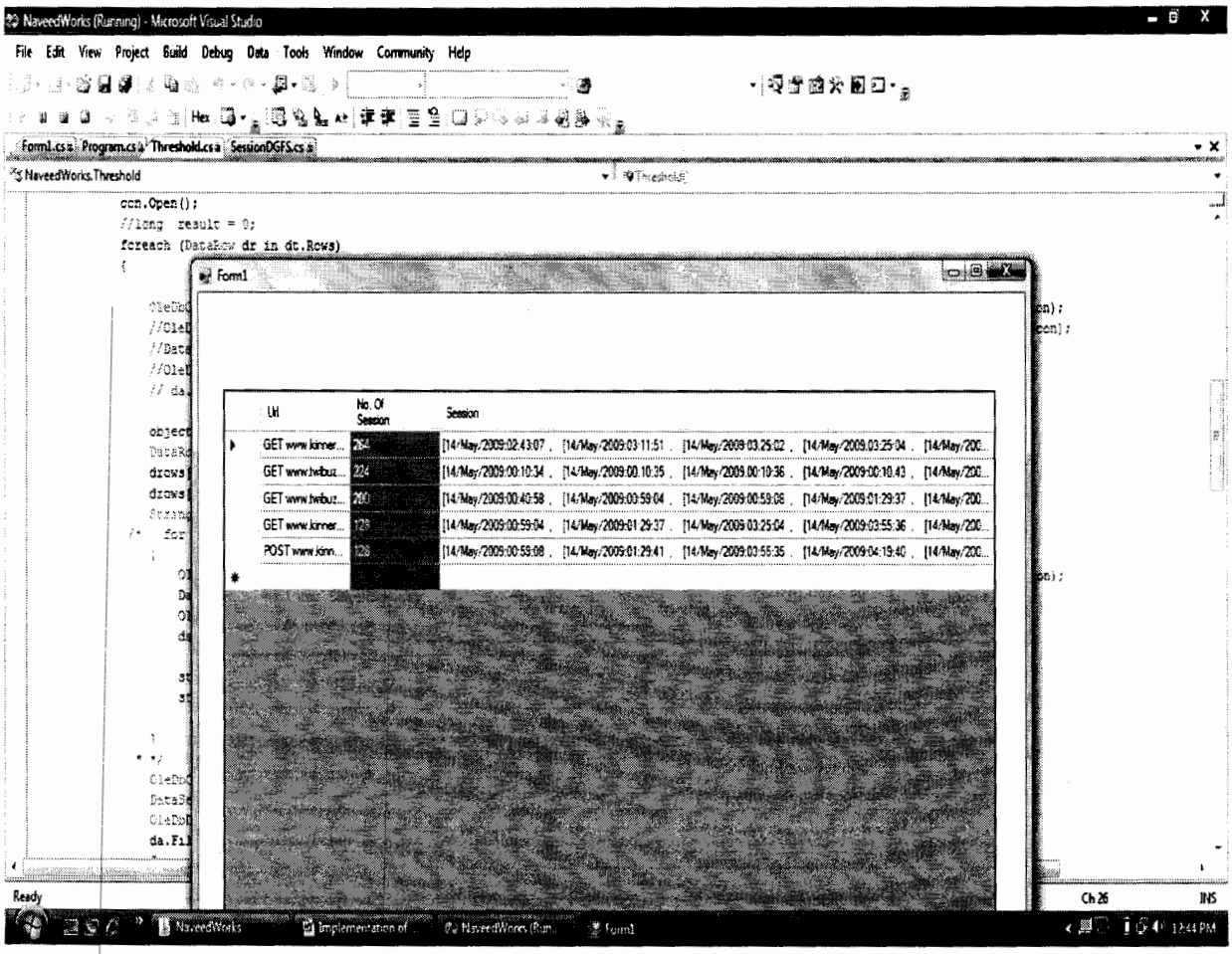

Figure: 5.12 Snapshot of existing algorithm by using Twibuzz data set

The results of proposed frame work for the same data set is given below:

P.No	Page Address	Session
1	GET www.kinneryandrajan.com/?page_id=2 HTTP/1.3	264
2	POST www.kinneryandrajan.com/wp-comments-post.php HTTP/1.3	224
3	GET www.kinneryandrajan.com/robots.txt HTTP/1.4	200
4	GET www.twibuzz.com/style.css HTTP/1.5	128
5	GET www.twibuzz.com/logo.png HTTP/1.5	128

Table: 5.12 Results of existing algorithm by using Twibuzz server data set

The data of web log tested on proposed technique is shown by the snapshot given below:

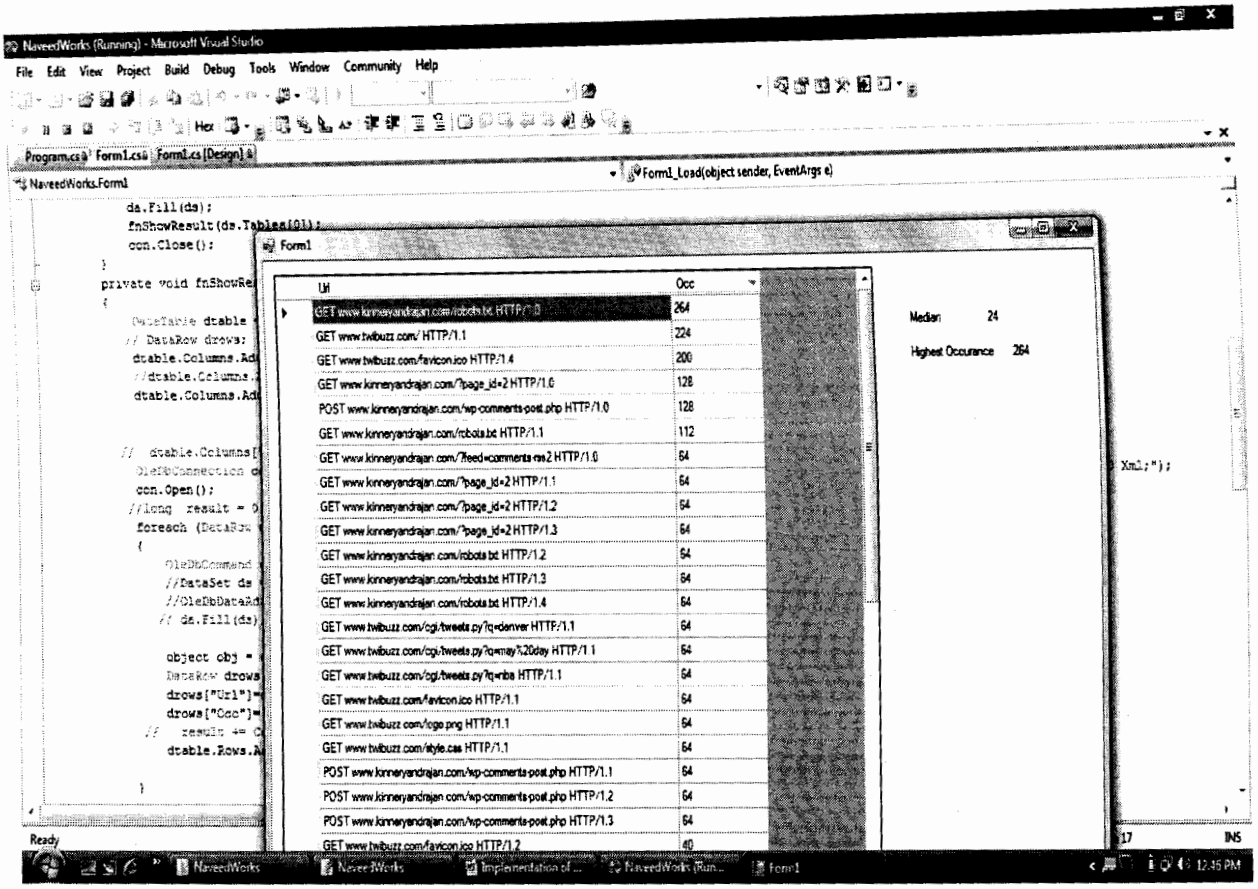

Figure: 5.13 Results of Proposed Techniques by using Twibuzz data set

The results are given below:

P.No	Page Address	Cluster
1	GET www.kinneryandrajan.com/?page_id=2 HTTP/1.3	264
2	POST www.kinneryandrajan.com/wp-comments-post.php HTTP/1.3	224
3	GET www.kinneryandrajan.com/robots.txt HTTP/1.4	200
4	GET www.twibuzz.com/style.css HTTP/1.5	128
5	GET www.twibuzz.com/logo.png HTTP/1.5	128
6	GET www.kinneryandrajan.com/?feed=comments-rss2 / home/HTTP/1.0	112
7	GET www.kinneryandrajan.com/id 3?page_id=2 HTTP/1.0	64
8	POST www.kinneryandrajan.com/name/wp-comments-post.php HTTP/1.0	64
9	GET www.kinneryandrajan.com/list/?page_id=2 HTTP/1.0	64

10	POST www.kinneryandrajan.com/wp-comments-post.php/evaluate/ HTTP/1.0	64
11	GET www.kinneryandrajan.com/robots.txt/pag_id=6 HTTP/1.1	64
12	GET www.twibuzz.com/style.css/ id_9 HTTP/1.1	64
13	GET www.twibuzz.com/logo.png /id_10/HTTP/1.1	64
14	GET www.twibuzz.com/favicon.ico/id_12 HTTP/1.1	64
15	GET www.twibuzz.com/cgi/tweets.py?q=denver/id _13 HTTP/1.1	64
16	GET www.twibuzz.com/cgi/tweets.py?q=nba/id_14 HTTP/1.1	64
17	GET www.twibuzz.com/cgi/tweets.py?q=may%20day/id_15 HTTP/1.1	64
18	GET www.kinneryandrajan.com/robots.txt/id_16 HTTP/1.0	64
19	GET www.kinneryandrajan.com/?feed=comments-rss2/id_17 HTTP/1.0	64
20	GET www.kinneryandrajan.com/?page_id=29 HTTP/1.0	64
21	POST www.kinneryandrajan.com/wp-comments-post.php/id_30 HTTP/1.0	64
22	GET www.kinneryandrajan.com/?page_id=23 HTTP/1.0	64
23	POST www.kinneryandrajan.com/wp-comments-post.php HTTP/1.0	40
24	GET www.kinneryandrajan.com/?feed=comments-rss27 HTTP/1.0	32
25	GET www.kinneryandrajan.com/?page_id=29 HTTP/1.0	32
26	POST www.kinneryandrajan.com/wp-comments-post.php/40 HTTP/1.0	32
27	GET www.kinneryandrajan.com/?page_id=60 HTTP/1.0	32
28	POST www.kinneryandrajan.com/wp-comments-post.php/70 HTTP/1.0	32
29	**Cluster**	**28**

Table: 5.13 Results of Proposed Techniques by using Twibuzz data set

Graphical Representation

The graph is constructed on the basis of results produced by the existing algorithm and proposed framework techniques. The existing algorithm predicted the session. The session is

time when the user accessed the pages. On the basis of session the algorithm in the existing techniques predicted the page.

The page position is representing page numbering along x-axis according to prediction of pages by existing algorithm and proposed techniques. The pages predicted in existing algorithm are represented by a line of existing algorithm in x-y coordinate. The proposed model techniques prediction of pages is represented by line result output framework in x-y coordinate. The proposed framework techniques provide higher availability of resources over the existing algorithm as shown in graph for the same for Tiwibuzz web server data set.

The given below is the graph comparison of data set of Twibuzz weblog server which is used as input for both algorithm and proposed technique at server side.

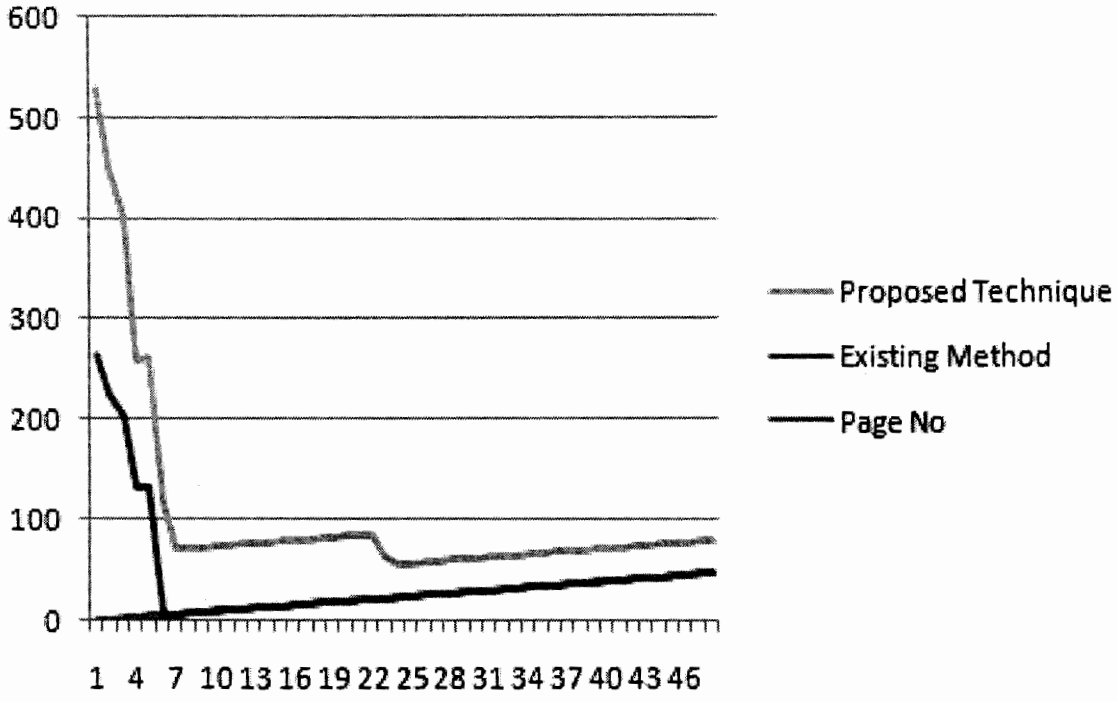

Figure: 5.14 Comparisons of results by using Twibuzz data set

X-axis represents the urls/pages and y-axis represents the users accessed these urls.

By analysis of graph, it is clearly seen that the results data set produced by our proposed framework technique for prediction of web page is efficient. By applying the formula we evaluated the performance.

Summary of Results

Page No	Percentage of Existing	Percentage of Proposed Model	Page No	Percentage of Proposed Model
1	9.04%	9.04%	15	2.19%
2	7.67%	7.67%	16	2.19%
3	6.84%	6.84%	17	2.19%
4	4.38%	4.38%	18	2.19%
5	4.38%	4.38%	19	2.19%
6	-	3.83%	20	2.19%
7	-	2.19%	21	2.19%
8	-	2.19%	22	2.19%
9	-	2.19%	23	1.36%
10	-	2.19%	24	1.09%
11	-	2.19%	25	1.09%
12	-	2.19%	26	1.09%
13	-	2.19%	27	1.09%
14	-	2.19%	28	1.09%
Performance	**32.31%**			**77.99%**

Table: 5.15 Comparisons of results by using Tiwibuzz data set

Overall Generalize Comparisons of Results:

The overall performance for the same data set of International Islamic university web server, National university web server, Hazara university web server and Tiwibuzz web server for existing techniques and proposed model technique are presented in the table given below:

Comparative Summary

Serial No.	Data Sets	Existing Technique	Proposed Model Techniques
1	IIUI web server	18.3%	36.96%
2	HU web server	47.26%	96.46%
3	NUST web server	28.47%	96.715%
4	Tiwibuz web server	32.31%	77.99%

Table: 5.16 Comparisons of all the data set results

The results proved that proposed model technique is efficient in web page prediction. The proposed model techniques improved overall performance of web prefetching system.

5.2 Advantages of our Proposed Techniques

- **Predicted the correct prefetching web object**

The graphical analysis showed that the proposed technique performed for the input data sets of three different web servers. Hence the method developed is efficient for all four data set of different web logs of web servers for prediction of web object for user.

- **Resource Usage**

The sequential Rank Based Selection algorithm predicted a page for prefetching purpose, so it utilizes less memory space of a user. When a user requests for one page, that page and prefetching page is given back to the user as a reply in the proposed model technique.

Suppose that a user access the web page, the proposed framework calculate the predicted pages for the available to the user. The sequential rank based Selection algorithm will select a page for the user. This page from a cluster will be selected by the proposed framework.

- **Web Page Usage**

The clustering of same page in a group can be viewed every page's usage of website by different users. This can be represented by the graph. The website usages of all the three web servers are represented by the graph by different users below:

- **Graphical representation of web usage**

The parts of International Islamic University and Hazara University web usage by the users are given below in the form of graph.

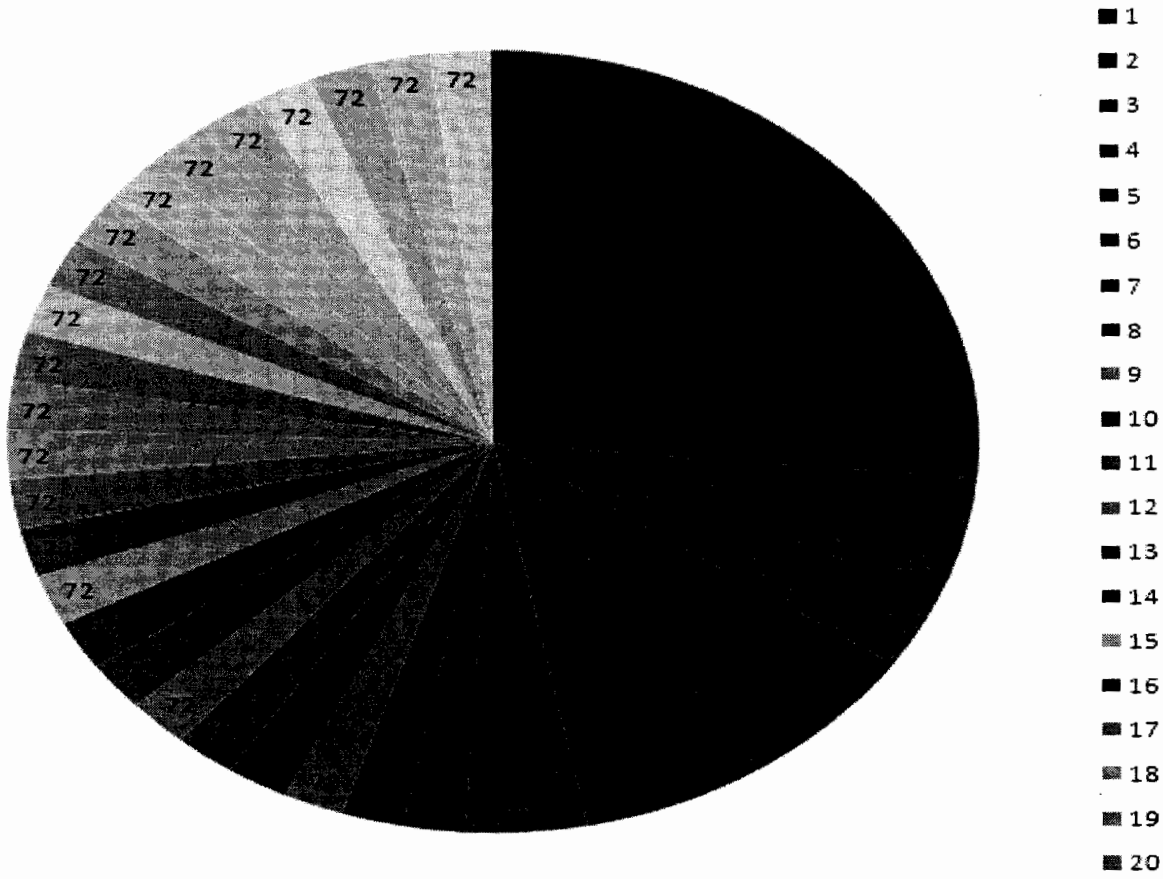

Figure: 5.15 Web usage of International Islamic University Website

Graphical representation of web users of IIUI Web server and grouping of user's behavior

Web usage of the server

Page No	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Usage of Page By users	576	359	288	216	216	144	72	72	72	12	72	72	72	72	72
Page No	**16**	**17**	**18**	**19**	**20**										
Usage of Page By users	72	72	72	72	72										

The proposed architecture shows the web usage behavior of different users of who have accessed different pages of web server. The graph given below represents the web usage behavior of different users who have accessed the web pages from Hazara University Web server.

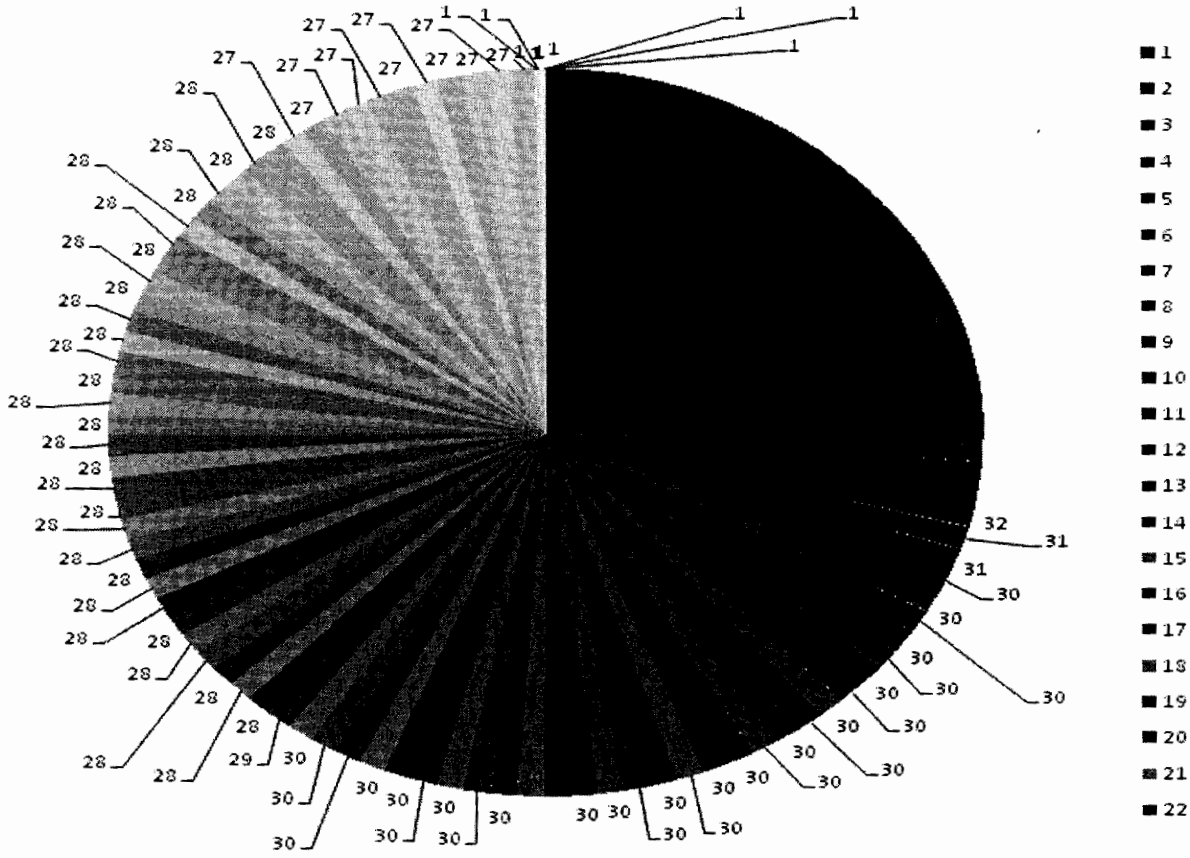

Figure: 5.16 Web usage of Hazara University Website

Web usage of the server

Page No	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Usage of Page By users	324	61	60	60	60	60	58	56	56	55	32	31	31	30	30
Page No	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
Usage of Page By users	30	30	30	30	30	30	28	28	28	27	27	27	1	1	1

The graph given below represents the web usage behavior of different users who have accessed the web pages from National University of Science and Technology Web server.

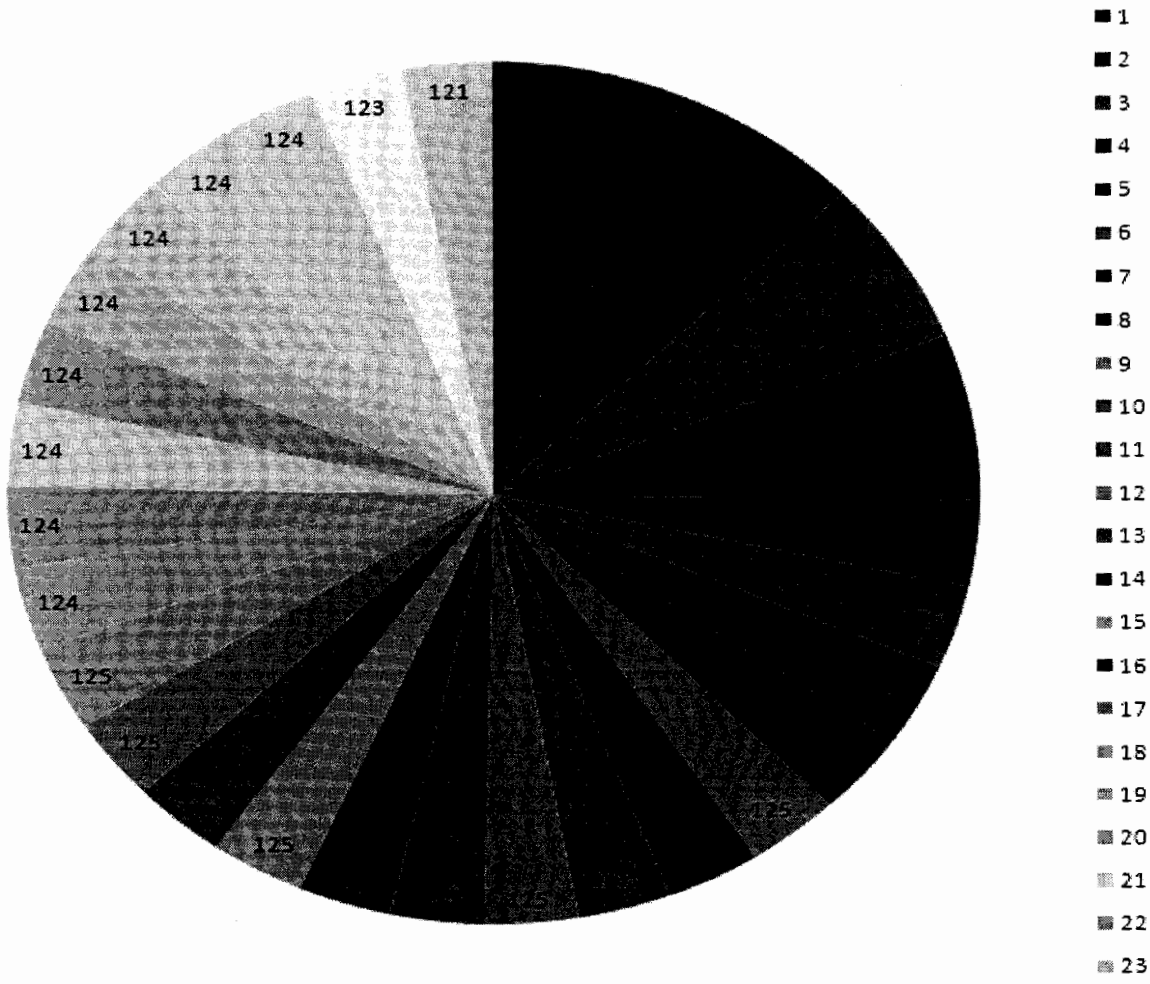

Figure: 5.17 Web usage of National University of a Science and Technology Website

Web usage of NUST university web server

Page No	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
Usage of Page By users	258	253	250	248	130	125	125	125	125	125	125	125	125
Page No	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**
Usage of Page By users	125	125	125	125	125	124	124	124	124	124	124	124	124
Page No	**27**	**28**											
Usage of Page By users	123	121											

5.3 Conclusion

When a client requests for a web page, before accessing the web page a prediction is made for accessing that web page. All the web objects are brought from server to the client. The access to the web objects are on the basis of the data prefetched from the server.

The data sets of three web logs of servers are tested on both existing algorithm and the model proposed by us. The results showed that proposed mechanism performed better than the algorithm existing for web page prediction. By implementation and graphical analysis, results showed that method outperformed the existing algorithm in web page prefetching mechanism.

Major advantage of the proposed technique is that It selects only one web page object of a website for prefetching purposes of user; hence consumed much less memory space of users and utilizes much less bandwidth of the network. The proposed technique reduced the user's latency due to the efficient prediction of web pages by the Sequential Rank Based Selection algorithm. Our model's implementation showed the web usage behavior of different users who have accessed the pages from a web server.

References

[1] Miguel Gomes da Costa Júnior, Zhiguo Gong, "Web Structure Mining: An Introduction", Proceedings of the IEEE International Conference on Information Acquisition, Hong Kong and Macau, China, June 2005.

[2] "User Instructions", http://water.usgs.gov/osw/streamstats/instructions.html, accessed on April 5, 2010.

[3] Greg Barish, Katia Obraczka, "World Wide Web Caching: Trends and Techniques", IEEE Communications Magazine, Information Science Institute University of Southern California, Los Angeles, USA, pp: 178-184, May 2000.

[4] "Web mining", http://searchcrm.techtarget.com/definition/Web-mining, accessed on April 15, 2010.

[5] "Types of Web mining", http://www.expertstown.com/web-mining-types/, accessed on August 10, 2010.

[6] O Kit Hong, Fiona Robert, P.Biuk Aghiai, "A Web Prefetching Model Based Content Analysis", http://www.sftw.umac.mo/~robertb/publications/MITC99/MITC99.pdf, accessed on July 10, 2009.

[7] Qiang Yang, Haining Henry Zhang, Tianyi Li, "Mining Web Logs for Prediction Models in WWW Caching and Prefetching", International Conference on Knowledge Discovery and Data mining in proceedings of the seventh ACM SIGKDD, San Francisco, California, USA, pp: 473-478, 2001.

[8] Jyoti Pandey, Amit Goel, Dr A K Sharma, "A framework for predictive web prefetching at proxy level using Data Mining", IJCSNS International Journal of Computer Science and Network Security, pp: 303-308, June 2008.

[9] Josep Dom`enech, Julio Sahuquillo, Jos´e Ana Gil, Ana Pont, "The Impact of the Web Prefetching Architecture on the Limits of Reducing User's Perceived Latency", In proceeding WI '06 of IEEE/WIC/ACM International Conference on Web Intelligence, Hong Kong, pp: 740–744, 2006.

[10] Victor Safronov, Manish Parashar, "Optimizing Web servers using Page rank prefetching for clustered accesses", Journal of Information Sciences, pp: 165-176, 17 March 2003.

[11] Dan Foygel, Dennis Strelow, "Reducing Web Latency with Hierarchical Cache based Prefetching", International Conference on Parallel Processing (ICPPW'00), Toronto, Canada, pp: 103, August 2000.

[12] B.dela Ossa, J.Sahuquillo, A. Pont, J. A.Gil, "An Empirical Study on Maximum Latency Saving in Web Prefetching", In proceedings of IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, pp: 556-559, Milan, Italy, 2009.

[13] Johann Marquez, Josep Dom`enech, Jos´e A.Gil, Ana Pon, "An intelligent technique for controlling web prefetching costs at the server side", In proceeding of International Conference on Web Intelligence and Intelligent Agent Technology, Sydney, Vol: 1, pp: 669-675, 2008.

[14] Joseph Domenech, Ana Pont, "A user focused evaluation of Web Prefetching Algorithms", Journal of Computer Communication, Vol: 10, Page No: 2113-2224, 2007.

[15] Ajay Bhushan Pandey, Jaideep Srivastava, Shashi, Shekhar, "A Web Proxy Server with an Intelligent Prefetcher for Dynamic Pages Using Association Rules", Technical Report, Department of Computer Science and Engineering, University of Minnesota, Minneapolis, USA. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.125.1597&rep=rep1&type=pdf, accessed on July 10, 2010.

[16] Josep Dom`enech, Jos´e A.Gil, Julio Sahuquillo, Ana Pont, "DDG: An Efficient Prefetching Algorithm for Current Web Generation", Hot web'06 1st IEEE Workshop, Issue: 14, November 2006, ISBN: 1-4244-0596-3, pp: 1-12, Boston.

[17] B.dela Ossa, J.A. Gil, J.Sahuquillo, A.Pont, "Web Prefetch Performance Evaluation in a Real Environment", IFIP/ACM Latin America Networking Conference, pp: 8, 11 October 2007.

[18] B.de la Ossa, A. Pont, J.Sahuquillo, J.A. Gil, "Referrer Graph: a low-cost web prediction algorithm", Proceedings of the 25th ACM Symposium on Applied Computing (ACM SAC 2010), Switzerland, pp: 831-838, 2010.

[19] Yingyin Jiang, Min-You, Wei Shu, "Web Prefetching: Costs, Benefits and Performance", Journal of Department of Electrical and Computer Engineering, University of New Mexico, WCW' Boulder, Colorado, Aug 15, 2002.

[20] George Pallis, Athena Vakali, Jaroslav Pokorny, "A clustering-based prefetching scheme on a Web cache environment", Journal of Computers and Electrical Engineering, ACM, Vol.34, Issue: 4, pp: 309-323, 2008.

[21] Christos Bouras, Agisilos Konidaris, "A Most Popular Approach of Predictive Prefetching on a WAN to Efficiently Improve WWW Response Times", In proceedings of Springer, Berlin, pp: 344-351, 2004.

[22] Bamshad Mobasher, Honghua Dai,Tao Luo, Miki Nakagawa, "Using Sequential and Non-Sequential Patterns in Predictive Web Usage Mining Tasks", In proceedings IEEE of International Conference of Data Mining(ICDM), USA, pp: 669-672, 2002.

[23] Costantinos Dimopoulos, Christos Makris, Yannis Panagis, Evangelos Theodoridis, Athanasios Tsakalidis, "A web page usage prediction scheme using sequence indexing and clustering techniques", 12th International Conference on Applications of Natural Language to Information Systems, Vol: 69, Issue: 4, April 2010, pp: 371-382, Greece, 2010.

[24] Beihong Jin, Sihua Tian, Chen Lin, Xin Ren, Yu Huang, "An Integrated Prefetching and Caching Scheme for Mobile Web Caching System", In proceeding of Software Engineering, Artificial Intelligence/Networking, and Parallel/Distributed Computing, Qingdao, China, pp: 522 -527, 2007.

[25] Dong Zhou, Ajay Chander, Hillview Avenue Palo Alto, "Optimizing User Interaction for Web-based Mobile Tasks", Proceeding WWW '10 Proceedings of the 19th international conference on World wide web, ACM, New York, USA, 2010.

[26] Miha Grčar, Dunja Mladenič, Marko Grobelnik, J.Stefan, Jamova, "User Profiling for the Web", Journal of Computer Science Information System, vol: 3, pp: 1-29, 2006.

[27] S.Veeramalai, N.Jaisankar, A.Kannan, "Efficient Web Log Mining Using Enhanced Apriori Algorithm with Hash Tree and Fuzzy", International journal of computer science & information technology (IJCSIT), Vol: 2, Issue: 4, August 2010.

[28] Jaideep Srivastava, Robert Cooleyz, Mukund Deshpande, Pang-Ning Tan, "Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data", ACM SIGKDD, New York, USA, Vol: 1, Issue: 2, January 2000.

[29] Qiang Yang, Zhen Zhang, "Model based Predictive Prefetching", In proceeding of International workshop of Database and Expert System Application, IEEE Computer Society, Washington, USA, pp: 291-295, ISBN: 0-7695-1230-5, 2000.

[30] Behrooz Parhami, "Introduction to Parallel Processing Algorithms and Architectures", pp: 111-112, Publisher: Kluwer Academic New York, ISBN: 0-306-45970-1, 2002.

[31] Margaret H.Dunham, "Data Mining: Introductory and Advanced Topics", Publisher: Prentice Hall, ISBN: 10-0130888923, pages: 315, 2002.

[32] Soumen Chakrabarti, "Mining the Web Discovering Knowledge from Hypertext Data", Publisher: Morgan-Kaufmann, ISBN: 1-55860-754-4, pages: 352, 2002.