Reducing Latency For Mobile Devices

by

Mohammed Farhan Khan
17141024
Md. Murad Ali Iskander
19201140
Ishmam Raiyan Rouf
16201110
Ekram Wasi Shatil
16301072

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
January 2021

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

farhan

Mohammed Farhan Khan
17141024

Md. Murad

Md. Murad Ali Iskander
19201140

Ishmam R. Rouf

Ishmam Raiyan Rouf
16201110

Shatil

Ekram Wasi Shatil
16301072

Approval

The thesis/project titled "Reducing Latency For Mobile Devices" submitted by

1. Mohammed Farhan Khan (17141024)
2. Md. Murad Ali Iskander (19201140)
3. Ishmam Raiyan Rouf (16201110)
4. Ekram Wasi Shatil (16301072)

Of Fall, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 8, 2021.

Examining Committee:

Supervisor:
(Member)

Hossain Arif

Hossain Arif
Assistant Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Prof. Mahbub Majumdar
Chairperson
Dept. of Computer Science & Engineering
Brac University

Mahbubul Alam Majumdar
Professor and Dean
Department of Computer Science and Engineering
Brac University

Abstract/ Executive Summary

The massive development of the web has caused the servers to be overloaded and causes heavy amounts of network traffic. A Lot of users around the world are concurrently accessing the web, hence due to insufficient bandwidth bottleneck is created. To make web service better, rapid time of response is necessary to maintain the user's connection to the web, and therefore, many ways need to be created to diminish the latency of connecting to web pages. In this case, web caching offers an effective approach by temporarily saving files close by, minimizing time of access, yet restricting the device to a degree. Web prefetching, which is a more effective strategy, is designed to fetch pages beforehand, additionally reducing latency. In the past, several Web prefetching methods have been proposed, incorporating different web and data mining techniques. To analyze proxy server data Web Use Mining has been integrated, and to get and to get a perception into user's web browsing trends, and then create regulations. Thus, we propose a Link analysis method, with a modified weighted Hits based algorithm that mainly focuses on two traits of how similar and popular the page is.These are integrated with other rules to rank the pages to be perfected. After the experimentation results reveal that the modified Hits algorithm can discover more relevant pages compared to Arc and Hits.

Keywords: Prefetching, bandwidth, Latency, proxy, Hits, Arc.

Dedication

Dedicated to every single soul who works hard to bring a change to the world of science.

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.

Secondly, to our advisor Hossain Arif sir for his kind support and advice in our work. He helped us whenever we needed help.

Thirdly, to Jon Kleinberg, researcher and inventor of the Hyperlink-Induced Topic Search algorithm, who helped us in our research by providing necessary resources.

And finally to our parents without their throughout support it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

List of Figures

List of Tables

List of acronyms

This list describes several acronyms that will be later used within the body of the document

ARC	Adaptive Replacement Cache
CSS	Cascading Style Sheet
DNS	Dynamic Name System
GIF	Graphics Interchange Format
GSP	Generalized Sequential Pattern
HITS	Hyperlink Induced Topic Search
HTTP	Hyper Text Transfer Protocol
IBM	International Business Machine
IDE	Integrated Development Environment
IHITS	Improved Hyperlink induced Topic Search
IP	Internet Protocol
JPG	Joint Photographic Expert Group
LRU	Least Recently Used
OS	Operating System
PLWAP	Pre-order Linked Location Encoded Web Access Pattern
PPE	Packet Processing Engine
STG	Shot Transition Graph
TCP	Transfer Control Protocol
WAP	Web Access Pattern
WASD	Web Get to Sequence Database

Chapter 1

Introduction

1.1 Motivation

To allow portable computers to access the internet quickly a significant quantity of research has been done. Limiting bandwidth enables mobile consumers to have high communication costs. However, this expense may be minimized as servers use mobile customers to relay data the same way they use it. The cost of broadcasting standard interest data is independent of the number of consumers who receive it. However, broadcast environments create high latency for consumers, as opposed to on-demand data service. This occurs because broadcast produces a unidirectional stream of airborne data, and consumers have to stand by for the relevant self-interest data to show, causing network congestion.

1.2 Objective

The aim of our work is to reduce the latency with the help of different techniques of the broadcast data and allow predictive prefetching to be done by the client's devices. The smart technique is implemented at the server, and placing the data items that will be requested the most is placed close to each other. The aim of this method is to take the one by one requested data items and decrease the latency to access the pages. At the client-side, client cache hit ratio is improved by predicting the data items that the client might request later. Predicting the prefetching list decreases the delay of access by enlarging the cache hit ratio utilization. Over time, the broadcast request issued is stored in a history where valuable information is gathered about the trends of broadcast requests and their time of issue. The history of the broadcast is stored by the order they were requested and the series of items submitted by customers. Our methods are focused on studying the history of broadcasting using data mining methods. These approaches are used in broadcast history to extract valuable data. These techniques are used to find useful information from the histories. Finding similarities and sequences in every data item by analyzing a vast collection of data is done though data mining research.

1.3 Thesis Overview

The subsequent sections of the thesis have been organized as follows. Chapter 2 is the literature review which contains related existing approaches along with the problem statement relevant to our proposed model. Chapter 3 includes all the background information related to our work and how we are using the resources to get the desired output, along with a review of all the. Section 3.1 contains introduction to Page rank and how its overall process works. Section 3.2 is the section discussing Markov model which is mainly used to find subsequent states of a graph. Section 3.3 is elaborates different approaches used in page rank. Section 3.4 explains another algorithm of page rank which is a simple way of ranking pages. Section 3.5 explains hits algorithm and how it works. In chapter 4, we have described the proposed model of our works along with relevant graphs and table. It includes the overall models relevant to our solution algorithm. The predicted results and relevant discussions are showed in chapter 5. Lastly, the summary of the report and conclusion as well as some future work-plan is done in chapter 6. In future work part, we will talk about our future ambition and working plan to upgrade our proposed model for better performance.

Chapter 2

Literature Review

Dongshan and Junyi et al [1], Extensive research in both academics and industries have been the focus of the gathering of information about user behavior on the Internet. Several methods to gathering data on user Web behavior have been introduced over the past few years. Some of these are: (1) updated web browsers, (2) logs of web servers, (3) logs of web proxies, and (4) monitoring of packets. These strategies has its own advantages, but most suffer from significant limitations with regard to the details that can be recorded in detailed. Until now, the only technique used to gather data for estimating different latency components was the method of packet monitoring, which used complex algorithms and external hardware to generate the HTTP trace. This method has been incorporated in a patented commercial solution quite recently. However, according to A. Singh and A. K. Singh [2], web caching and pre-fetching are approached by the usage of mining process through sequential data, which improves the efficiency of the proxy server. The squid proxy server provides access to web logs that provide information about the navigation pattern of the user and helps to enhance the efficiency of the server's web capture and pre-fetching. The Pre-order Linked Location Encoded Web Access Pattern (PLWAP) algorithm is used to locate the user's frequent web object access, evaluate access log file browsing history, and compare results using page replacement algorithms. Previous papers also implemented web pre-fetching, which includes a cluster of closely connected web pages. All web pages are called simultaneously. The writers have sought to fix it. But two log files were used for the experimental work to carry out. One for data formation and another for checking the results collected. And results were directly influenced by the choice of files.

Kumar et al [3] proposed Web Usage Mining as integration in order to analyze the proxy server and cached logs and also getting insight into the proxy server user access habits and to create rules. Also, to rank the pages to be pre-fetched, a process called the Relation analysis method is suggested to be combined with these rules. The authors have suggested the operational methods in three steps, considering web pages that are likely to be viewed by users. 1) Preprocessing phase, in which web based using of mining essentially uses log files for access to the input dataset. Such files primarily store users browsing habits in commonly used log formats or mixed log-based formats. The Preprocessing phase operates in the following stages (i) Data Cleaning, removing unnecessary data that is not needed for mining purposes. (ii) User identity, recognizing individual users. (iii) Session recognition, using a 30-minute time-out duration, requests made by each user are split into the meetings.

(iv) Route Completion, for the precise tracking of sequential transactions. Then, 2) Rule Creator Process, association rules creation, rough-set algorithm of clustering steps, k-order Markov model and all associated rule mining are implemented subsequently. The phase with measures such as (i) Rough Set Clustering assists in decision-making process of classifying Data. Usage of the principles of goal set, classes of equivalence, the upper and the lower approximations are also found. (ii) k-order Markov Model & Association Rule Mining, Markov analyses is applied after the clustered set is obtained. The basic assumption is that all the subsequent based states depend on the k-order states in the Markov model which were previously. Finally, 3) Rule Selector & Page Rating Process, which is the selecting of suitable rules and regulations of high valued pages after the association rules have been obtained. This is proposed as the most achieving by assigning significant scores to each connected page using a HITS-based algorithm.

Liao et al [4] proposed for distributed file systems, a server-side pre-fetching technique. This paper also sought to strengthen data prefetching on using the combined client identification data and storage side which accesses client logs to enhance data prefetching. To direct read block data in advance, the prefetching scheme will predict certain block addresses of read operations that will be in future and push it to the corresponding client so that per-client block can be analyzed to access server history. The authors also used piggy-backed that identifies clients to distinguish server-side block-level to access streams, using a horizontal graph of visibility approach to convert selected sequence of block access in the linked network. The incoming I/O based requests piggyback client information on the storage servers so that it is possible to anticipate future accesses and prefetch block data completely on the server side without any interaction with the client file system or the program. J. Liao [4] suggested using the horizontal graph of visibility technique in turning a time series of blocking access events into a linked graph, using the Tarjan algorithm, which is also a graph theory algorithm to locate vertices in a linked graph which are cut, to identify patterns with block access. The X-step algorithm of pattern matching seeks access patterns that are matching to the background of blocked access observed in return quickly and reliably predict future access. Awareness about mapping of the logical I/O requests to actual, physical access blocks on storage servers on client machines that have to know about certain client file system and application details. Block access case history can be positioned in a client-specific sense such as servers can anticipate and store potential block access all requests and then proactively push the perfect data to the correct systems of client file. The client file system also gives an I/O based requests to the block data writing and reading storage servers, showing the average response time for the five block requests to be selected and also given request. Knowledge of logical I/O requests mapped to specific physical access blocks on the storage servers on client filesystems and application details.

Phan et al [5] used SMP, which is similarity-based on SOAP a multicast protocol. According to them, high-volume transactions and electronic communication require bandwidth-efficient communication, such as multimedia or mobile applications. The SOAP mechanism produces an awfully large amount of traffic once there are similar service operations. SOAP network traffic could be minimized by using SMP to boost efficiency. Similarity-based SOAP multicasting (SMP) also no dependent on low-level multicasting on SOAP unicast, so no network configuration is urgent which are quite complexed. Duplicate parts are reused for various clients for extremely similar messages, rather than producing messages for different customers with duplicated identical pieces, which are sent from the source as a result of dramatically reducing network traffic. (i) Similarity measurement (ii) SOAP message indexing (iii) SMP message structure (iv) SMP routing are four main components of SOAP. Analytical studies have shown that SOAP multicast has defeated conventional unicast or multicast, and (i) total traffic network and (ii) average time to response is where the two requirements. SMP uses unicast route path and inserts it inside the message into a set of clients. SMP may also send numerous messages together, which demonstrates performance. By calculating the similarity between two XML schemas, the similarity between messages was determined. Message templates of two SOAP are compared on the basis of a standard XML node between the two message XML trees. To allow the sending of SMP, a network of enabled routers based on SMP which has to be displayed between the server and also between the client. Each router parses the SMP message header along a multicast tree, partitions the client addresses into bunches based on the following hop of each client, duplicates SMP messages if necessary, and progresses along the following hop to match SMP messages. Yang [6] suggested a system of SOAP indexing to ensure the rapid merging and splitting of SMP messages. Each of the XML node is also marked with a location and presented with a compact frame including 1) The ID of the node information type, 2) the node's position within the message, and 3) the value of the node. The Dijkstra algorithm is used, and most messages follow the shortest paths.

Cohen and Kaplan [7] as means of minimizing web-based latency, spoke of using pre-fetching of the text. As a synchronized solution, they suggested pre-transfer prefetching techniques and demonstrated their potential for a substantial decrease in long hold-up times. They viewed prefetching as a distinctive complement to other latency reduction elements, such as document caching, record prefetching, persistent HTTP, and HTTP requirement pipelining. User-perceived lag is the biggest concern they have tackled, which is irritating to all right now. With the flow of time, anyone who uses the internet wants to feel real-time when sharing data, documents, etc. So, following strategies have been proposed: (i) Pre-resolving, removing DNS query time (ii) Pre-connecting the first request to practice a secure connection, and (iii) Pre-warming, which is effectively a request to the server before the original request. Y. Xu and Y. Han [8], understood that waits encountered by high-bandwidth network customers are regulated by the setup phase prior to the actual transmission of information. So, they decided to find out how to reduce the link setup time so that the data could be transmitted faster than before. One of the drawbacks of this approach they have seen is that there is a lot of overhead on network bandwidth for document prefetching. Other than the content itself, other entities may be prefetched using a prefetched TCP. This includes cache validations, identification of out-of-date links, recognition of redirected links and follow-up, and Meta data prefetching. A side effect of these HTTP exchanges is server "warning".

Chapter 3

Background Study

3.1 Page Rank

The Page Rank algorithm is the algorithm that Google uses to rate search engine web pages [9]. This algorithm was named after the researcher named Larry. Page Rank algorithm works to evaluate an uncomfortable assessment of how vital the web is by counting the number and quality of network joins to a webpage. The underlying suspicion is that other websites are going to receive more connections from more critical websites. The estimation of the Page Rank algorithm yields a probability dissemination used to speak to the possibility that a person will arrive at some given page haphazardly clicking on joins. For collections of reports of some measure, PageRank will estimate. In a few academic papers, it is agreed that the dispersion at the beginning of the computational planning is similarly divided between all documents within the array. The PageRank measurements involve a few passes, called "iterations", to change incorrect PageRank values through the array to more accurately match the hypothetical approximation.

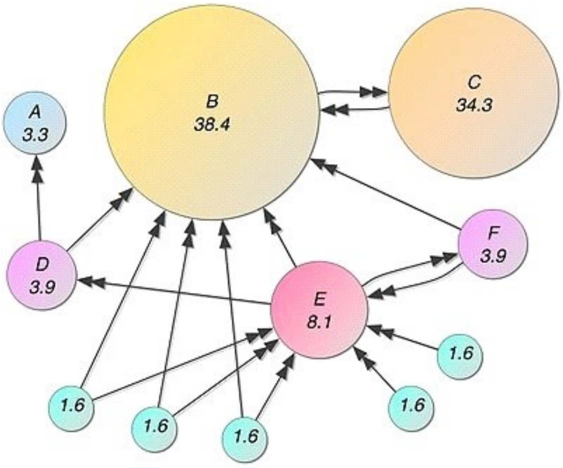


Figure 3.1: Pagerank expressed network [10]

Page Rank expressed networks as percentages. From the figure 3.1 we can see that C has more percentage then E though it has more connections then C [10]. Here C gets high value because it has an important page which contains high value. In the off probability that web surfers starting on an irregular page has an 85 percent chance of selecting an arbitrary interface from the page they are actually traveling to, and a

15 percent chance of jumping from the whole web to a page picked irregularly, then they would hit Web Page E with a 8.1 percent of that time. Without damping, all other pages will have Page Rank zero with the exception of A, B and C but under damping, Page A effectively links to all other pages on the site, despite the fact that it has no active links to its argument.

3.2 Markov Model

Markov model is used to changing systems arbitrarily [11] [12]. It is expected that future states depend as it were on the current state, not on the occasions that happened before it. For the most part, this presumption empowers thinking and computation with the model that would something else be recalcitrant. For this reason, within the areas of prescient demonstrating and probabilistic estimating, it is alluring for a given demonstrate to display the Markov property. By doling out probabilities for the movements inside the chain, the observable stage finalizes the Markov chain. In terms of relative frequencies, the probabilities speak to the predicted consumption. At that point, test cases are chosen by the Markov model as going through. The advantages of Markov models are that this is widespread and as long as it captures the organizational behavior, the generated groupings look like a test of the actual usage. Another value of it is a systematic stochastic model is based on it.

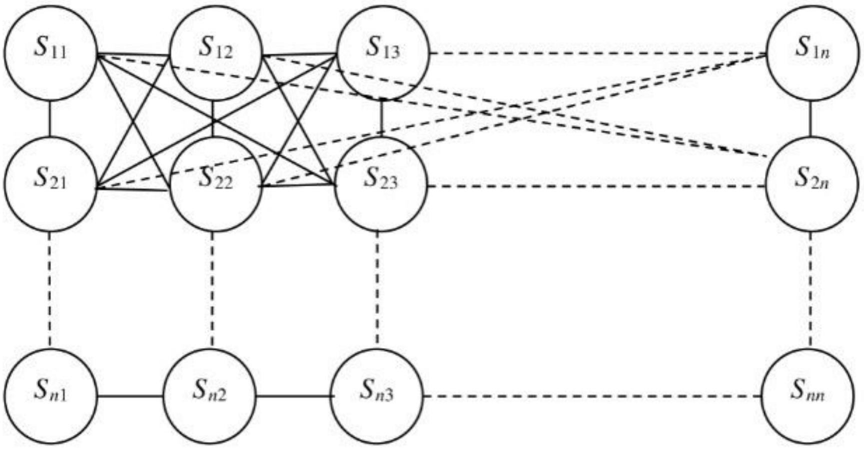


Figure 3.2: Concept of Markov model [13]

Markov Demonstrate's basic idea is to predict a further collection of web pages based on the results of the client's previous history [13]. Web anticipation, based on his prior sessions, the following action or actions compares to expecting that the client will go to the following collection of web sites. The client's past behavior is contrasted to the previous collection of web pages that have been browsed as of now. The Kth-order Markov demonstrates in Web page expectation as the chance that a client can access the K+1th page as the client has been past the Kth web pages requested.

Consider Sn1, Sn2, Sn3...Snn be the set of web pages gone to by the client and s is the arrangement of pages we ought to foresee Sn+1$^{\text{th}}$ page, so it is given as:
Pr (Sn+1=s | S1=s1, S2=s2.......Sn=sn)

For first order Markov model:

Pr (Sn2=snn | Sn1=s21).

(2)For second order Markov model:

Pr (Sn3=snn | S2=s22, S1=s21)

The Markov model's basic preferences are its skill and implementation in constructing a model. Building various orders of the Markov display can be effectively seen to be direct with the approximation of the planning set given. The main concept is to use the data structure to build and keeps track of the configuration of each design along with its probability.

3.3 Various Prefetching Approach

Web has advanced to supply complex user specific energetic administrations [7]. These administrations have put a great demand on the constrained arrange foundation that ensures advantages for these administrations. In terms of higher latency, the higher demand for content and the small agreement basis have resulted in unpleaseant experience for web customers. Adding network capacity and increasing the entire transmission speed of the network is the simplest solution to this issue. However, such an arrangement empowers the creation of applications that devour the next bandwidth arrangement and hence have a richer user encounter that induces network congestion. The techniques used to enhance intermediary server execution are web caching system and pre-fetching system. On the basis of log analysis and the implementation of knowledge mining techniques, the different approaches proposed for these procedures can be grouped according to categories.

3.3.1 Clustering Based Approach

The assumption of web accesses demonstrates with high precision by recommending a technique named as a cut-and-pick approach based on critical suspicion that the same customer usually visits multiple similar sites that form clusters [14]. As indicated by the author in the Network Proxy log, the fundamental challenge is to share identifying facts of and client as the intermediary log includes data from more than one source jointly, the message is interleaved respect to time.

The developer provides a lot like a sophisticated transaction approach that separates facts between exchanges by determining the borders between logs and then by choosing the proper grouping of references between them. In this method, to separate the similarly similar client IP address websites, the author generates a STG from the client access design obtained from the preprocessed intermediary log. The basic web page information is disposed of and used like it was website details during the creation of the STG. At that point, we prune the less connected websites for each client IP by using the related run of display mining and thereby obtain a cluster of similarly related sites for each client.

A session-based approach is used in this approach, which means that the time gap between two subsequent customer IP specifications is not more than edge (tout) and each request is coordinated on the basis of time stamp of the log segment. Finally, the creator compares past approaches with its outcome (known as the settled time interim & server approach) and appears to be productive with his suggested cut-and-pick method.

Another method focused on the Rough Set Clustering principle that classifies loose, doubtful or missing data in terms of knowledge gathered from past work. The developer believes that the important web log sessions should get to data as they were those in which the client invested the most severe time. In this way, the developer proposes pruning the less important session on the basis of dynamic measurement of the threshold of each client exchange. The speaker addresses that, as in such meetings, the client has been through the most drastic number of pages in further clustering. They demonstrated a calculation called Unpleasant Set of Clustering algorithm that uses the notion of rough sets to determine equivalence set of sessions at that stage to discover Lower Estimate & Upper Guess of sessions by using clustering technique to differentiate client session log. The lower estimate covers all sessions that certainly have a position in the cluster and the higher guess includes pages that might belong to the cluster.

Client session clustering is performed on the basis of lower and upper approximation clusters. The data obtained in the session was used to foresee the pre-fetching of another page. For session time down, a 30-minute session edge is introduced. At the intermediate server, the authors suggest the idea of PPE that will take client requests as input and matches that ask for existing hard set clusters and then opt to pre-fetch the page for that user. Another method which explores the almost ubiquity of web pages is subject to adjustment in terms of time and thus it is extremely difficult to pre-characterize a well-known web pages and therefore estimate a cap of notoriety, so a chart-based approach is more feasible in web page clustering.

3.3.2 Sequential Data Mining based Approach

Successive mining approach to the site access pattern of mine visit from the crude Web server information log [15]. The preprocessed web logs are orchestrated in this method in the access turn of the personal client arrangement that exists in the access sequence database known as the Web Get to Sequence Database (WASD), which is later used to consecutive mining.

An information system called WAP (Web Access Pattern) was designed to register compactly for the disposal of the sweeping help count for groupings and corresponding counts. In addition, the WAP tree effectively retains connections of traversing prefixes with regard to postfix architecture. A map known as the WAP (Network Access pattern) tree is then constructed by filtering the WASD to mine and visit subsequence of the web access pattern for each node (Client IP) recursively but only two times. To start with, it determines the set of visit occasions and the next WAP-mine filter generates an information structure, called WAP tree, to log all tally occasions for further mining, using visit occasions. At this stage WAP mines the WAP trees recur ably explores the Web Get to Build The WAP trees.

Creators view on algorithm called PLWAP in another similar approach that utilizes a pre-order associated, position-coded WAP tree form. In addition, it removes the need to recursively re-construct WAP trees in the midst of sequential mining carried out by the WAP tree process, thus minimizing execution time. Compared to the WAP tree in which usual postfix groupings are discovered, the PLWAP approach finds common prefix arrangements to begin with. The concept is to discover a frequent design by constantly discovering its typical frequent sequences in the design starting with the main visit case.

Using location codes, it is able to recognize the descendants and kin hubs of the most experienced parent hub inside the header table on the postfix root set of a visit portion that is tested for concatenating the subsequence of the prefix in case it is visited in addition to root set agreements. In the event that all the bolsters considering elder parent hubs of its entire postfix root are more significant than the least back or equal to, a part is visited.

At that point, the developer eventually compares the different algorithms such as GSP and WAP using PLWAP which takes the runtime of PLWAP and verifies if it is much smaller than the methods used earlier.

3.3.3 Web Caching and Pre-fetching Approach

Web caching can be a well-known tool for the application of the Web-based system by storing Web objects that are expected to be used in the near future in an environment closer to the user [16]. The Internet caching instruments are actualized at three levels: client level, intermediary level and unique server level. Essentially, proxy servers play the key parts between clients and web locales in reducing of the response time of client demands and sparing of arrange transfer speed. Subsequently, for achieving superior reaction time, a proficient caching approach ought to be built in a proxy server. The cache substitution is the center or heart of the net caching consequently, the plan of proficient cache substitution calculations is vital for caching components achievement. Hence, cache substitution calculations are moreover called web caching algorithms. Due to the impediment of cache space, a cleverly mechanism is required to oversee the Internet cache content effectively. The conventional caching policies are not productive within the Web caching since they consider just one calculate and overlook other components that have effect on the effectiveness of the Net caching. In these caching approaches, most prevalent objects get the foremost demands, whereas a huge parcel of objects, which are put away within the cache, are never asked once more. This is called cache pollution problem. In this manner, numerous Web cache substitution policies have been proposed.

The internet prefetching could be a hot investigate point that has picked up expanding consideration in recent a long time. The cache prefetching gets a few web objects some time recently clients actually ask it. Hence, the cache prefetching makes a difference on lessening the user- perceived idleness. Numerous considers have appeared that the combination of caching and prefetching pairs the execution compared to single caching. Concurring to, a combination of web caching and prefetching can potentially move forward idleness up to 60%, though web caching alone makes strides the latency up to 26%. Be that as it may, the most disadvantage of frameworks upgraded with prefetching approach is that the clients may not inevitably ask a few perfected objects. In such a case, the prefetching conspire increments the arrange traffic as well as the Internet servers' stack. In addition, the cache space isn't utilized ideally. Subsequently, the prefetching approach ought to be planned carefully in arrange to overcome these impediments. In any case, the most disadvantage of frameworks improved with prefetching arrangement is that the clients may not in the long run ask a few prefetched objects. In such a case, the prefetching conspire increments the arrange traffic as well as the Net servers' stack. Besides, the cache space isn't utilized ideally. Hence, the prefetching approach ought to be outlined carefully in arrange to overcome these impediments.

3.4 Adaptive Replacement Cache

Adaptive Replacement Cache (ARC) is used for replace pages to get better performance [17] [18]. It is a page replacement algorithm. IBM Almaden Research Center developed this algorithm in 2006. This algorithm is basically used for both keep tracking of frequently used pages and recently used pages.

LRU (least recently used) algorithm maintain a list of recently used pages and sort them to make a further move but in terms of ARC algorithm it keeps track of most viewed pages as well as recently viewed pages and make sort of the connections for further use. LRU algorithm makes a list of recent viewed pages and keep them to the top in the priority list and other entries goes down to the list. To overcome this problem ARC was introduced. ARC algorithm is used for basically keep tracking both the most viewed pages as well as recent viewed pages. By part of the cache

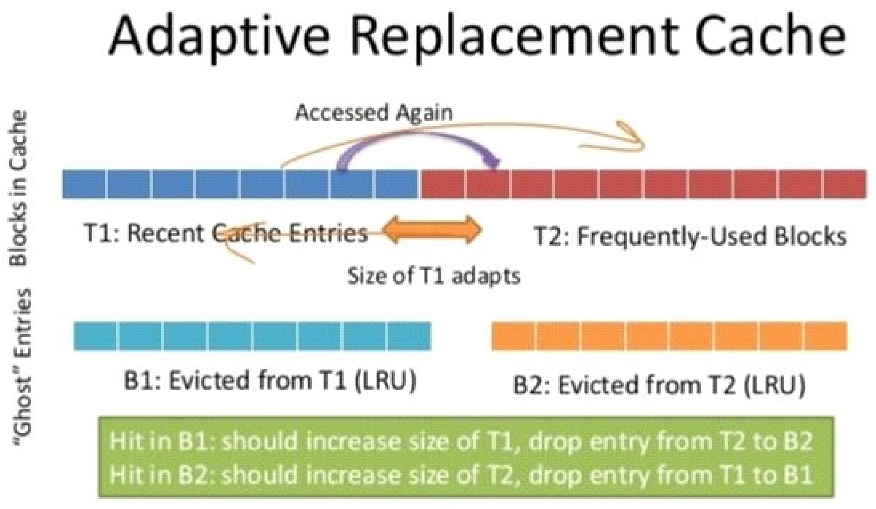

Adaptive Replacement Cache
Accessed Again
Blocks in Cache
"Ghost" Entries
T1: Recent Cache Entries
T2: Frequently-Used Blocks
Size of T1 adapts
B1: Evicted from T1 (LRU)
B2: Evicted from T2 (LRU)
Hit in B1: should increase size of T1, drop entry from T2 to B2
Hit in B2: should increase size of T2, drop entry from T1 to B1

Figure 3.3: Adaptive Replacement Cache [19]

register, ARC algorithm helps to improves the fundamental LRU technique into two records, R1 and R2, one used in recently used pages and another one is used for most visited pages [19]. Each of these, in essence, is extended with redacted list (B1 or B2) added to the bottom of the two documents. By keeping documents of the past of late deleted cache passages, these apparition records serve as scorecards, and the algorithm implements ghost hits to respond to subsequent changes in asset use. Notice that the documents of the apparition contain metadata as it is (keys for the passages) and not the details of the asset itself. Four LRU documents type out the unified cache registry. T1, used for new entries, T2, used for regular entries. B1, recently evicted unwanted entries from the T1 cache, but still registered. B2, for identical unwanted entries, but evicted from T2.

R1 together with B1 are referred to together as L1, which is a collective history with single comparisons later on. In comparison, L2 is the mix of new passages T2 and that B2 joins T1, to the cleared out and are continuously forced out to the cleared out, separated from T1 to B1 in the long term, and finally dropped out at last. Any segment in T1, which is referenced again, receives another chance, and reaches L2, fair to the central proper! Marker. Marker. It is driven outward from there once again, taking T2 to B2. This can be rehashed uncertainly by passages in L2 which receives another blow before they eventually drop out on the far right to B2.Entries that enter the cache of (T1, T2) will trigger this! To pass towards the marker of the goal∧. If there is no free space inside the cache, this marker also determines if an entry will be removed from either T1 or T2. The calculation of T1, moving ∧ to the right, would be increased by hits in B1. In T2, the final segment is removed to B2. Hits in B2 shrink T1, pushing ∧ down to the cleaned out. At present, the final passage in T1 is forced into B1.

3.5 Hyperlink Induced Topic Search

A connection analysis algorithm developed by Jon Kleinberg which marks web pages [20] is the Hyperlink Induced Subject Search (HITS) Algorithm. For a basic search for web link constructs, this algorithm was used to explore and rate the relevant web site pages. HITS is used as the hubs and authorities seen in Figure 3.4 to create a recursive connection between web pages.

Authorities: The sets of extremely applicable web site pages are called as Roots, provided the query to a search engine. They are future Authorities.

Hubs: Hubs are named pages that are not really relevant and points another page in the Root. Therefore, an Authority is a website that has many hubs connected to it, while a Hub is also connected to authority as it is connected via various other web pages.

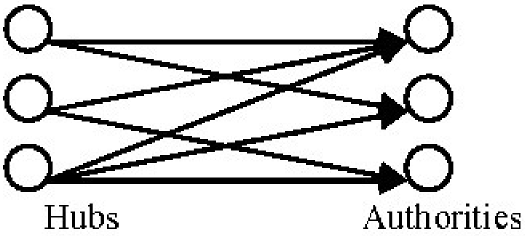


Figure 3.4: Hubs and Authorities link

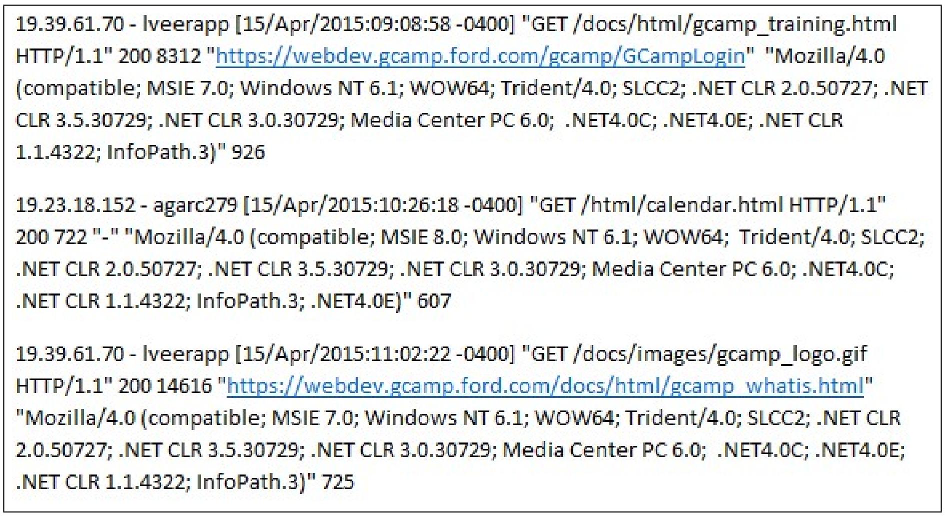
Chapter 4

Proposed Model

The proposed process workflow is divided into three phases:

4.1 Preprocessing Phase

For the input dataset access log files are used for web usage mining. These files mainly store users browsing patterns in different log formats of mixed and common. A snippet of the proxy server entries is shown in Fig. 4.1.



```
19.39.61.70 - lveerapp [15/Apr/2015:09:08:58 -0400] "GET /docs/html/gcamp_training.html
HTTP/1.1" 200 8312 "https://webdev.gcamp.ford.com/gcamp/GCampLogin"  "Mozilla/4.0
(compatible; MSIE 7.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET
CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0;  .NET4.0C; .NET4.0E; .NET CLR
1.1.4322; InfoPath.3)" 926

19.23.18.152 - agarc279 [15/Apr/2015:10:26:18 -0400] "GET /html/calendar.html HTTP/1.1"
200 722 "-" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64;  Trident/4.0; SLCC2;
.NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C;
.NET CLR 1.1.4322; InfoPath.3; .NET4.0E)" 607

19.39.61.70 - lveerapp [15/Apr/2015:11:02:22 -0400] "GET /docs/images/gcamp_logo.gif
HTTP/1.1" 200 14616 "https://webdev.gcamp.ford.com/docs/html/gcamp_whatis.html"
"Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR
2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0;  .NET4.0C; .NET4.0E;
.NET CLR 1.1.4322; InfoPath.3)" 725
```

Figure 4.1: Proxy Log

The following fields correspond to each entry:

- host
- rfc931
- user_name
- date : time
- requests
- status
- byte
- referrer

- user_agent
- cookies

The steps of the preprocessing phase is shown:

4.1.1 Data Cleansing:

In this stage, information such as jpg,gif and css files which are unnecessary and unrelated are removed and it is first for mining.

4.1.2 Identification of user:

With the use of IP address, user agent and referrer entries Specific users are identified by the IP address. To a large extent, the user identification phase depends on the connection type of the user and other conditions.

4.1.3 Identification of Session:

Using a time-out duration of 30 minutes requests made by each user are broken down into sessions.

4.1.4 Path completion:

This phase is done to accurately record the sequential transactions. Path that are not registered explicitly and hard to find out in the log entries are detected here.

4.2 Rule Creator Phase

To create association rules the steps are rough-set clustering, then k-order markov model and similarity- popularity based hits model are applied subsequently.

4.2.1 Rough Set clustering:

The goal of clustering is to separate the data into points of several groups in such a way that the data which are similar are put in same group, hence forming groups containing several similar data. A multitude of groups more specifically clusters of data are formed in such a way. It was done to find the initial rough clustered set by applying the clustering method. Because of uncertainty, to further classify the decision process rough set clustering helps. The goal-set definitions, data in same groups, upper and lower estimation are used here. User sessions which are clustered are obtained.

4.2.2 k-order Markov Model:

Markov tests are applied after the rough set from the clustered data is obtained. In the Markov model, the basic presumption is that the following states depend on the prior k states. The frequency of each page is then calculated and with that value the probability $\mathrm{p_i}$ of $\mathrm{S^k{}_j}$ state of k number of pages is calculated:

$$P(p_{\mathrm{i}}|S^{\mathrm{k}}{}_{\mathrm{j}}) = \frac{Frequency(< S^{\mathrm{k}}{}_{\mathrm{j}}/p_{\mathrm{i}} >)}{Frequency(S^{\mathrm{k}}{}_{\mathrm{j}})} \tag{4.1}$$

The user session history that satisfies the states is then collected. For each state, association rules are built and stored

4.2.3 Rule Selector & Hits Algorithm phase:

Preference of applicable regulation and ranking of pages is performed after the session history is obtained which is link of pages. This is achieved using an algorithm based on HITS. The proposed algorithm is based on a modified version of weighted hits algorithm, the weights to edges are set considering how popular and similar the pages are.

The query of similarity of webpage $\mathrm{p_i}$ to webpage $\mathrm{p_j}$ is the similar value of π of page i on query set Q and π of page j on query set Q. It is shown as:

$$Similarity(\pi_Q p_i, \pi_Q p_j) = \begin{cases} (1+S_i)*(1+S_j) \ , & if \ i \to j \\ 0 \ , & otherwise \end{cases} \tag{4.2}$$

$\mathrm{S_i}$ shows the similar index of I on query set Q and $\mathrm{S_j}$ shows the similar index of j on query set Q. In addition, To describe the target document an text is found as the anchor, which describe the content of the page, and it summarizes the page with a high degree of precision the subject of the target content.

The popular value is considered of Link(i,j) define the condition the chance the user will switch from page i to page j via link(i,j). Two type of link(i,j) for popular according to its path. $\mathrm{W_{out}}$ is noted as the popular value considering number of out-links. That calculates the number of links going out for page i and the number of links going out for all page reference pages.

$$W^{out}_{(j,i)} = \frac{O_i}{\sum_{p \in R(j)} O_p} \tag{4.3}$$

Number of out-links of page i and page j is represented by $\mathrm{O_i}$ and $\mathrm{O_p}$, respectively.

The referrer page list of page which is j is denoted by R(j). Win is noted as the number of inlinks. $\mathrm{W_{in}}$ is determined considering the number of in-links of page i and all the number of pages in i inlinks.

$$W^{in}_{(j,i)} = \frac{I_i}{\sum_{p \in R(j)} I_p} \tag{4.4}$$

Where number of in-links of page I is represented by Ii and page j Ip. R(j) shows the Reference list of page j.

In Figure 4.2a, page A points page C and page D. The inlinks and outlinks of these two pages are $I_{\mathrm{C}} = 2, I_{\mathrm{E}} = 2, O_{\mathrm{C}} = 3, O_{\mathrm{E}} = 3$. Therefore, $W^{\mathrm{out}}(A, C) = O_{\mathrm{C}}/(O_{\mathrm{C}} + O_{\mathrm{E}}) = 3/5$ and $W^{\mathrm{in}}(A, C) = I_{\mathrm{C}}/(I_{\mathrm{C}} + I_{\mathrm{E}}) = 2/4$.

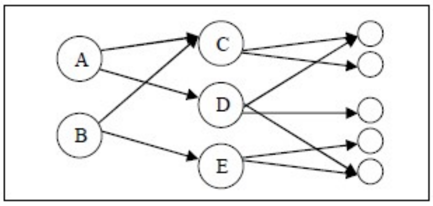


Figure 4.2a: Web Links

According to our model, link(h,$\mathrm{a_{inlink}}$)(i=1,2,3) in Figure 4.2 a and is more popular than link($\mathrm{h_{inlink}}$,a)(i=1,2,3) in Figure 4.2 b.

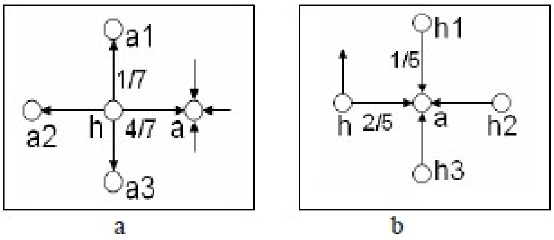
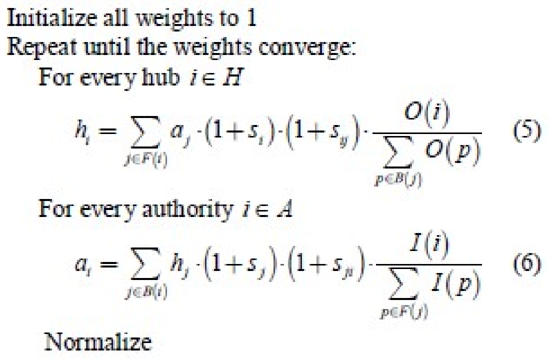


Figure 4.2b: How to distribute weight with popularity

Using the query induced popular and similar algorithm is combined, improved weighted hits-based algorithm is provided:



```
Initialize all weights to 1
Repeat until the weights converge:
  For every hub $i \in H$
    $h_i = \sum_{j \in F(i)} a_j \cdot (1+s_i) \cdot (1+s_{ij}) \cdot \frac{O(i)}{\sum_{p \in B(j)} O(p)}$    (5)
  For every authority $i \in A$
    $a_i = \sum_{j \in B(i)} h_j \cdot (1+s_j) \cdot (1+s_{ji}) \cdot \frac{I(i)}{\sum_{p \in F(j)} I(p)}$    (6)
  Normalize
```

(4.5)

Then by using this algorithm, score is assigned to each page. Then link analysis is performed and pages are ranked.

4.3 Page selector phase

A base of knowledge of all the links is created. When a request is received, the prefetching mechanism is run. Our modified Hits algorithm matches the rules and the pages are scored. The web pages likely searched by the user is then loaded into the cache.

Chapter 5

Result Analysis

5.1 Result

Since it is hard to work with a proxy server in the current condition, hence we have successfully evaluated our algorithm- in an IDE on a well specified computer. The hardware device that we used in this purpose had the following:

_Hardware - Intel Core i5-7200U 2.5 GHz Processor, 64-bit CPU, 1TB Hard Drive, 8GB DDR4 Memory

_Software - Eclipse IDE

_Environment – 8.1.6 Language - Java

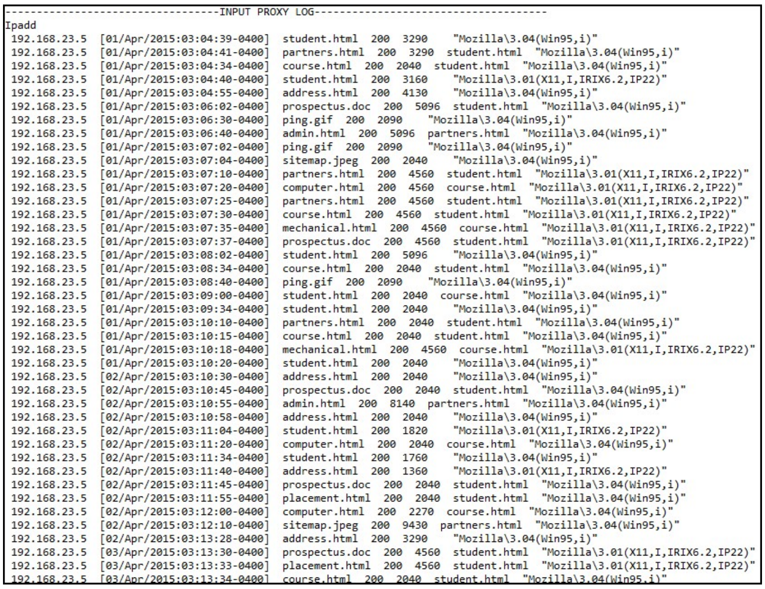
_OS - Windows 10 Education

_Network Bandwidth - 40 Mbps

We formulized the algorithm using the Eclipse IDE in Java. And used the data found with proxy server dataset.

5.2 Analysis

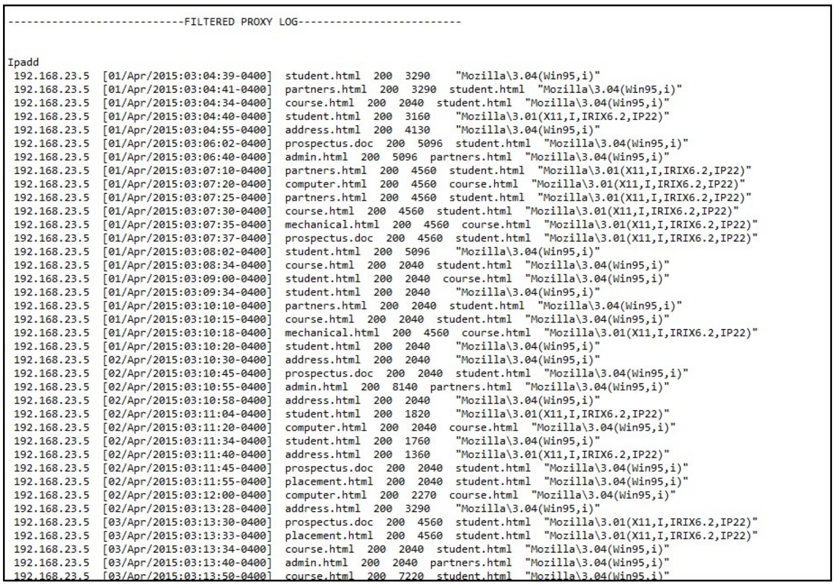
The implementation of the proposed method was carried out on a sample proxy log that was adapted beforehand to represent the appropriate preprocessing values. A snapshot of the proxy data log is given in figure 5.1.



```
--------------------------------INPUT PROXY LOG-----------------------------------
Ipadd
 192.168.23.5  [01/Apr/2015:03:04:39-0400]  student.html  200  3290    "Mozilla\3.04(Win95,i)"
 192.168.23.5  [01/Apr/2015:03:04:41-0400]  partners.html  200  3290  student.html  "Mozilla\3.04(Win95,i)"
 192.168.23.5  [01/Apr/2015:03:04:34-0400]  course.html  200  2040  student.html  "Mozilla\3.04(Win95,i)"
 192.168.23.5  [01/Apr/2015:03:04:40-0400]  student.html  200  3160    "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
 192.168.23.5  [01/Apr/2015:03:04:55-0400]  address.html  200  4130    "Mozilla\3.04(Win95,i)"
 192.168.23.5  [01/Apr/2015:03:06:02-0400]  prospectus.doc  200  5096  student.html  "Mozilla\3.04(Win95,i)"
 192.168.23.5  [01/Apr/2015:03:06:30-0400]  ping.gif  200  2090    "Mozilla\3.04(Win95,i)"
 192.168.23.5  [01/Apr/2015:03:06:40-0400]  admin.html  200  5096  partners.html  "Mozilla\3.04(Win95,i)"
 192.168.23.5  [01/Apr/2015:03:07:02-0400]  ping.gif  200  2090    "Mozilla\3.04(Win95,i)"
 192.168.23.5  [01/Apr/2015:03:07:04-0400]  sitemap.jpeg  200  2040    "Mozilla\3.04(Win95,i)"
 192.168.23.5  [01/Apr/2015:03:07:10-0400]  partners.html  200  4560  student.html  "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
 192.168.23.5  [01/Apr/2015:03:07:20-0400]  computer.html  200  4560  course.html  "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
 192.168.23.5  [01/Apr/2015:03:07:25-0400]  partners.html  200  4560  student.html  "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
 192.168.23.5  [01/Apr/2015:03:07:30-0400]  course.html  200  4560  student.html  "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
 192.168.23.5  [01/Apr/2015:03:07:35-0400]  mechanical.html  200  4560  course.html  "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
 192.168.23.5  [01/Apr/2015:03:07:37-0400]  prospectus.doc  200  4560  student.html  "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
 192.168.23.5  [01/Apr/2015:03:08:02-0400]  student.html  200  5096    "Mozilla\3.04(Win95,i)"
 192.168.23.5  [01/Apr/2015:03:08:34-0400]  course.html  200  2040  student.html  "Mozilla\3.04(Win95,i)"
 192.168.23.5  [01/Apr/2015:03:08:40-0400]  ping.gif  200  2090    "Mozilla\3.04(Win95,i)"
 192.168.23.5  [01/Apr/2015:03:09:00-0400]  student.html  200  2040  course.html  "Mozilla\3.04(Win95,i)"
 192.168.23.5  [01/Apr/2015:03:09:34-0400]  student.html  200  2040    "Mozilla\3.04(Win95,i)"
 192.168.23.5  [01/Apr/2015:03:10:10-0400]  partners.html  200  2040  student.html  "Mozilla\3.04(Win95,i)"
 192.168.23.5  [01/Apr/2015:03:10:15-0400]  course.html  200  2040  student.html  "Mozilla\3.04(Win95,i)"
 192.168.23.5  [01/Apr/2015:03:10:18-0400]  mechanical.html  200  4560  course.html  "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
 192.168.23.5  [01/Apr/2015:03:10:20-0400]  student.html  200  2040    "Mozilla\3.04(Win95,i)"
 192.168.23.5  [02/Apr/2015:03:10:30-0400]  address.html  200  2040    "Mozilla\3.04(Win95,i)"
 192.168.23.5  [02/Apr/2015:03:10:45-0400]  prospectus.doc  200  2040  student.html  "Mozilla\3.04(Win95,i)"
 192.168.23.5  [02/Apr/2015:03:10:55-0400]  admin.html  200  8140  partners.html  "Mozilla\3.04(Win95,i)"
 192.168.23.5  [02/Apr/2015:03:10:58-0400]  address.html  200  2040    "Mozilla\3.04(Win95,i)"
 192.168.23.5  [02/Apr/2015:03:11:04-0400]  student.html  200  1820    "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
 192.168.23.5  [02/Apr/2015:03:11:20-0400]  computer.html  200  2040  course.html  "Mozilla\3.04(Win95,i)"
 192.168.23.5  [02/Apr/2015:03:11:34-0400]  student.html  200  1760    "Mozilla\3.04(Win95,i)"
 192.168.23.5  [02/Apr/2015:03:11:40-0400]  address.html  200  1360    "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
 192.168.23.5  [02/Apr/2015:03:11:45-0400]  prospectus.doc  200  2040  student.html  "Mozilla\3.04(Win95,i)"
 192.168.23.5  [02/Apr/2015:03:11:55-0400]  placement.html  200  2040  student.html  "Mozilla\3.04(Win95,i)"
 192.168.23.5  [02/Apr/2015:03:12:00-0400]  computer.html  200  2270  course.html  "Mozilla\3.04(Win95,i)"
 192.168.23.5  [02/Apr/2015:03:12:10-0400]  sitemap.jpeg  200  9430  partners.html  "Mozilla\3.04(Win95,i)"
 192.168.23.5  [02/Apr/2015:03:13:28-0400]  address.html  200  3290    "Mozilla\3.04(Win95,i)"
 192.168.23.5  [03/Apr/2015:03:13:30-0400]  prospectus.doc  200  4560  student.html  "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
 192.168.23.5  [03/Apr/2015:03:13:33-0400]  placement.html  200  4560  student.html  "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
 192.168.23.5  [03/Apr/2015:03:13:34-0400]  course.html  200  2040  student.html  "Mozilla\3.04(Win95,i)"
```

Figure 5.1: Snapshot of proxy log

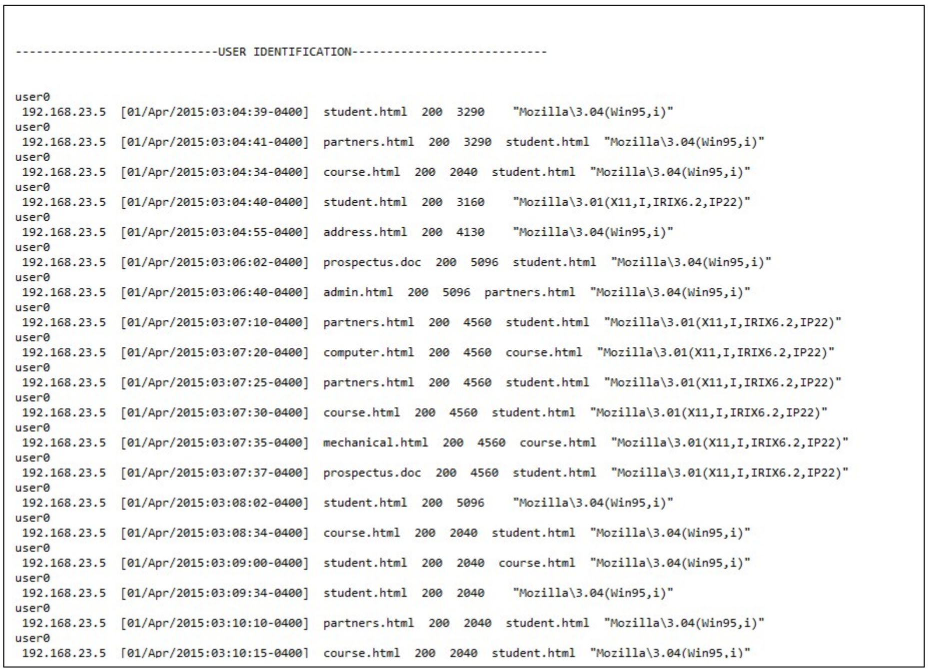
Form the proxy log in figure 5.1, irrelevant data which isn't needed is removed such as gifs and jpeg, i.e "ping.gif" shown in figure 5.2 which has no gif or jpeg.



```
---------------------------FILTERED PROXY LOG--------------------------


Ipadd
 192.168.23.5  [01/Apr/2015:03:04:39-0400]  student.html  200  3290    "Mozilla\3.04(Win95,i)"
 192.168.23.5  [01/Apr/2015:03:04:41-0400]  partners.html  200  3290  student.html  "Mozilla\3.04(Win95,i)"
 192.168.23.5  [01/Apr/2015:03:04:34-0400]  course.html  200  2040  student.html  "Mozilla\3.04(Win95,i)"
 192.168.23.5  [01/Apr/2015:03:04:40-0400]  student.html  200  3160    "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
 192.168.23.5  [01/Apr/2015:03:04:55-0400]  address.html  200  4130    "Mozilla\3.04(Win95,i)"
 192.168.23.5  [01/Apr/2015:03:06:02-0400]  prospectus.doc  200  5096  student.html  "Mozilla\3.04(Win95,i)"
 192.168.23.5  [01/Apr/2015:03:06:40-0400]  admin.html  200  5096  partners.html  "Mozilla\3.04(Win95,i)"
 192.168.23.5  [01/Apr/2015:03:07:10-0400]  partners.html  200  4560  student.html  "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
 192.168.23.5  [01/Apr/2015:03:07:20-0400]  computer.html  200  4560  course.html  "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
 192.168.23.5  [01/Apr/2015:03:07:25-0400]  partners.html  200  4560  student.html  "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
 192.168.23.5  [01/Apr/2015:03:07:30-0400]  course.html  200  4560  student.html  "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
 192.168.23.5  [01/Apr/2015:03:07:35-0400]  mechanical.html  200  4560  course.html  "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
 192.168.23.5  [01/Apr/2015:03:07:37-0400]  prospectus.doc  200  4560  student.html  "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
 192.168.23.5  [01/Apr/2015:03:08:02-0400]  student.html  200  5096    "Mozilla\3.04(Win95,i)"
 192.168.23.5  [01/Apr/2015:03:08:34-0400]  course.html  200  2040  student.html  "Mozilla\3.04(Win95,i)"
 192.168.23.5  [01/Apr/2015:03:09:00-0400]  student.html  200  2040  course.html  "Mozilla\3.04(Win95,i)"
 192.168.23.5  [01/Apr/2015:03:09:34-0400]  student.html  200  2040    "Mozilla\3.04(Win95,i)"
 192.168.23.5  [01/Apr/2015:03:10:10-0400]  partners.html  200  2040  student.html  "Mozilla\3.04(Win95,i)"
 192.168.23.5  [01/Apr/2015:03:10:15-0400]  course.html  200  2040  student.html  "Mozilla\3.04(Win95,i)"
 192.168.23.5  [01/Apr/2015:03:10:18-0400]  mechanical.html  200  4560  course.html  "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
 192.168.23.5  [01/Apr/2015:03:10:20-0400]  student.html  200  2040    "Mozilla\3.04(Win95,i)"
 192.168.23.5  [02/Apr/2015:03:10:30-0400]  address.html  200  2040    "Mozilla\3.04(Win95,i)"
 192.168.23.5  [02/Apr/2015:03:10:45-0400]  prospectus.doc  200  2040  student.html  "Mozilla\3.04(Win95,i)"
 192.168.23.5  [02/Apr/2015:03:10:55-0400]  admin.html  200  8140  partners.html  "Mozilla\3.04(Win95,i)"
 192.168.23.5  [02/Apr/2015:03:10:58-0400]  address.html  200  2040    "Mozilla\3.04(Win95,i)"
 192.168.23.5  [02/Apr/2015:03:11:04-0400]  student.html  200  1820    "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
 192.168.23.5  [02/Apr/2015:03:11:20-0400]  computer.html  200  2040  course.html  "Mozilla\3.04(Win95,i)"
 192.168.23.5  [02/Apr/2015:03:11:34-0400]  student.html  200  1760    "Mozilla\3.04(Win95,i)"
 192.168.23.5  [02/Apr/2015:03:11:40-0400]  address.html  200  1360    "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
 192.168.23.5  [02/Apr/2015:03:11:45-0400]  prospectus.doc  200  2040  student.html  "Mozilla\3.04(Win95,i)"
 192.168.23.5  [02/Apr/2015:03:11:55-0400]  placement.html  200  2040  student.html  "Mozilla\3.04(Win95,i)"
 192.168.23.5  [02/Apr/2015:03:12:00-0400]  computer.html  200  2270  course.html  "Mozilla\3.04(Win95,i)"
 192.168.23.5  [02/Apr/2015:03:13:28-0400]  address.html  200  3290    "Mozilla\3.04(Win95,i)"
 192.168.23.5  [03/Apr/2015:03:13:30-0400]  prospectus.doc  200  4560  student.html  "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
 192.168.23.5  [03/Apr/2015:03:13:33-0400]  placement.html  200  4560  student.html  "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
 192.168.23.5  [03/Apr/2015:03:13:34-0400]  course.html  200  2040  student.html  "Mozilla\3.04(Win95,i)"
 192.168.23.5  [03/Apr/2015:03:13:40-0400]  admin.html  200  2040  partners.html  "Mozilla\3.04(Win95,i)"
 192.168.23.5  [03/Apr/2015:03:13:50-0400]  course.html  200  7220  student.html  "Mozilla\3.04(Win95,i)"
```

Figure 5.2: Filtered proxy log

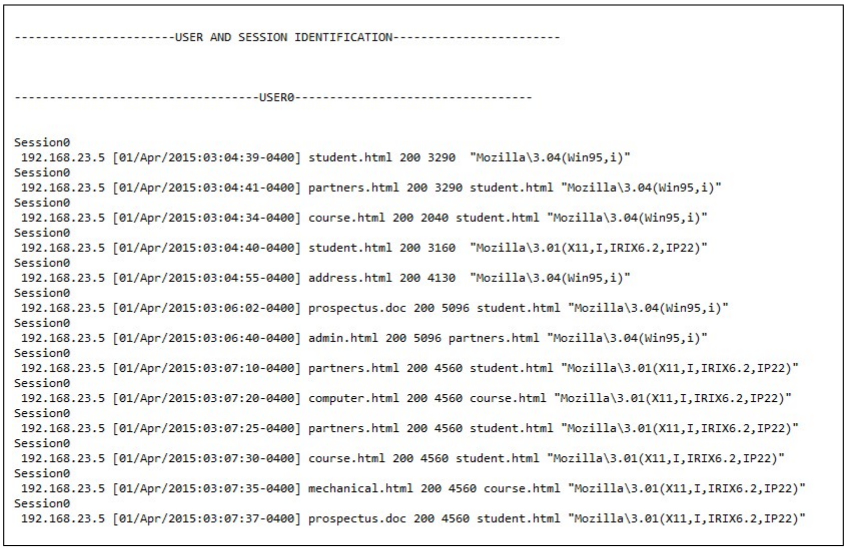
After filtering the proxy log, users are pinpointed. This is to individualize each user and hence get a clear usage history for specific user and their usage patterns. In figure 5.3 where user 0 is shown and like that more user are found out.



```
----------------------------USER IDENTIFICATION---------------------------


user0
 192.168.23.5  [01/Apr/2015:03:04:39-0400]  student.html  200  3290    "Mozilla\3.04(Win95,i)"
user0
 192.168.23.5  [01/Apr/2015:03:04:41-0400]  partners.html  200  3290  student.html  "Mozilla\3.04(Win95,i)"
user0
 192.168.23.5  [01/Apr/2015:03:04:34-0400]  course.html  200  2040  student.html  "Mozilla\3.04(Win95,i)"
user0
 192.168.23.5  [01/Apr/2015:03:04:40-0400]  student.html  200  3160    "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
user0
 192.168.23.5  [01/Apr/2015:03:04:55-0400]  address.html  200  4130    "Mozilla\3.04(Win95,i)"
user0
 192.168.23.5  [01/Apr/2015:03:06:02-0400]  prospectus.doc  200  5096  student.html  "Mozilla\3.04(Win95,i)"
user0
 192.168.23.5  [01/Apr/2015:03:06:40-0400]  admin.html  200  5096  partners.html  "Mozilla\3.04(Win95,i)"
user0
 192.168.23.5  [01/Apr/2015:03:07:10-0400]  partners.html  200  4560  student.html  "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
user0
 192.168.23.5  [01/Apr/2015:03:07:20-0400]  computer.html  200  4560  course.html  "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
user0
 192.168.23.5  [01/Apr/2015:03:07:25-0400]  partners.html  200  4560  student.html  "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
user0
 192.168.23.5  [01/Apr/2015:03:07:30-0400]  course.html  200  4560  student.html  "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
user0
 192.168.23.5  [01/Apr/2015:03:07:35-0400]  mechanical.html  200  4560  course.html  "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
user0
 192.168.23.5  [01/Apr/2015:03:07:37-0400]  prospectus.doc  200  4560  student.html  "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
user0
 192.168.23.5  [01/Apr/2015:03:08:02-0400]  student.html  200  5096    "Mozilla\3.04(Win95,i)"
user0
 192.168.23.5  [01/Apr/2015:03:08:34-0400]  course.html  200  2040  student.html  "Mozilla\3.04(Win95,i)"
user0
 192.168.23.5  [01/Apr/2015:03:09:00-0400]  student.html  200  2040  course.html  "Mozilla\3.04(Win95,i)"
user0
 192.168.23.5  [01/Apr/2015:03:09:34-0400]  student.html  200  2040    "Mozilla\3.04(Win95,i)"
user0
 192.168.23.5  [01/Apr/2015:03:10:10-0400]  partners.html  200  2040  student.html  "Mozilla\3.04(Win95,i)"
user0
 192.168.23.5  [01/Apr/2015:03:10:15-0400]  course.html  200  2040  student.html  "Mozilla\3.04(Win95,i)"
```

Figure 5.3: User Identification

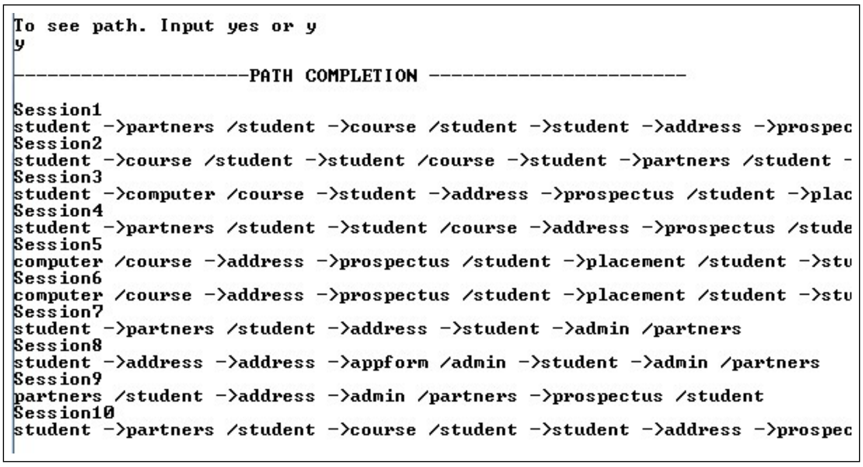
Next, sessions are found out for each user. To see in usage pattern in particular session. As shown in figure 5.4, all website visited by session 0 subsequently there are more sessions.



```
----------------------USER AND SESSION IDENTIFICATION-----------------------


----------------------------------USER0---------------------------------


Session0
 192.168.23.5 [01/Apr/2015:03:04:39-0400] student.html 200 3290  "Mozilla\3.04(Win95,i)"
Session0
 192.168.23.5 [01/Apr/2015:03:04:41-0400] partners.html 200 3290 student.html "Mozilla\3.04(Win95,i)"
Session0
 192.168.23.5 [01/Apr/2015:03:04:34-0400] course.html 200 2040 student.html "Mozilla\3.04(Win95,i)"
Session0
 192.168.23.5 [01/Apr/2015:03:04:40-0400] student.html 200 3160  "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
Session0
 192.168.23.5 [01/Apr/2015:03:04:55-0400] address.html 200 4130  "Mozilla\3.04(Win95,i)"
Session0
 192.168.23.5 [01/Apr/2015:03:06:02-0400] prospectus.doc 200 5096 student.html "Mozilla\3.04(Win95,i)"
Session0
 192.168.23.5 [01/Apr/2015:03:06:40-0400] admin.html 200 5096 partners.html "Mozilla\3.04(Win95,i)"
Session0
 192.168.23.5 [01/Apr/2015:03:07:10-0400] partners.html 200 4560 student.html "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
Session0
 192.168.23.5 [01/Apr/2015:03:07:20-0400] computer.html 200 4560 course.html "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
Session0
 192.168.23.5 [01/Apr/2015:03:07:25-0400] partners.html 200 4560 student.html "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
Session0
 192.168.23.5 [01/Apr/2015:03:07:30-0400] course.html 200 4560 student.html "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
Session0
 192.168.23.5 [01/Apr/2015:03:07:35-0400] mechanical.html 200 4560 course.html "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
Session0
 192.168.23.5 [01/Apr/2015:03:07:37-0400] prospectus.doc 200 4560 student.html "Mozilla\3.01(X11,I,IRIX6.2,IP22)"
```

Figure 5.4: Session Identification

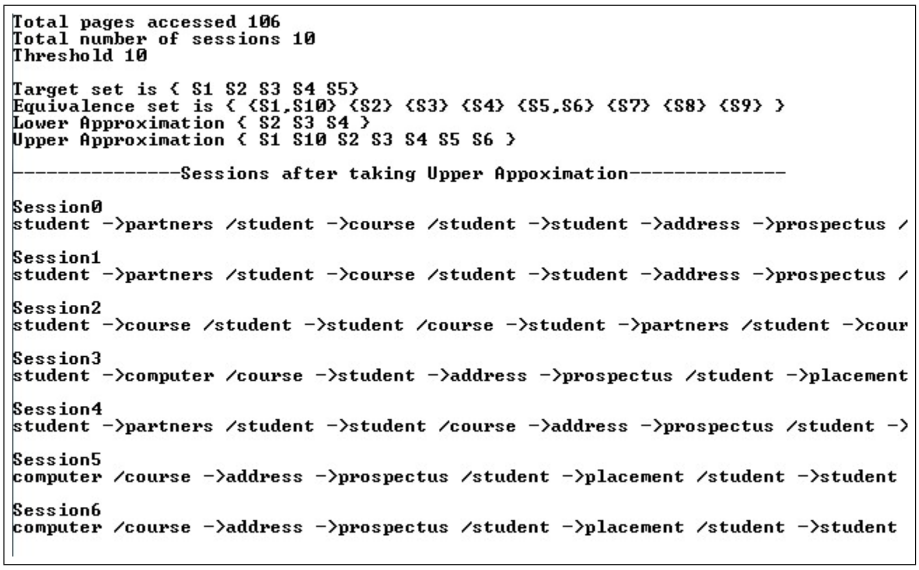
To access the path which might not be directly seen or accessible path completion is done. To see which pages connect to one after another. This is done for every session that is why we needed to find the sessions. In figure 5.5, we can see for session 1 the all pages visited.



```
To see path. Input yes or y
y
----------------------PATH COMPLETION ------------------------

Session1
student ->partners /student ->course /student ->student ->address ->prospec
Session2
student ->course /student ->student /course ->student ->partners /student -
Session3
student ->computer /course ->student ->address ->prospectus /student ->plac
Session4
student ->partners /student ->student /course ->address ->prospectus /stude
Session5
computer /course ->address ->prospectus /student ->placement /student ->stu
Session6
computer /course ->address ->prospectus /student ->placement /student ->stu
Session7
student ->partners /student ->address ->student ->admin /partners
Session8
student ->address ->address ->appform /admin ->student ->admin /partners
Session9
partners /student ->address ->admin /partners ->prospectus /student
Session10
student ->partners /student ->course /student ->student ->address ->prospec
```

Figure 5.5: Path Completion

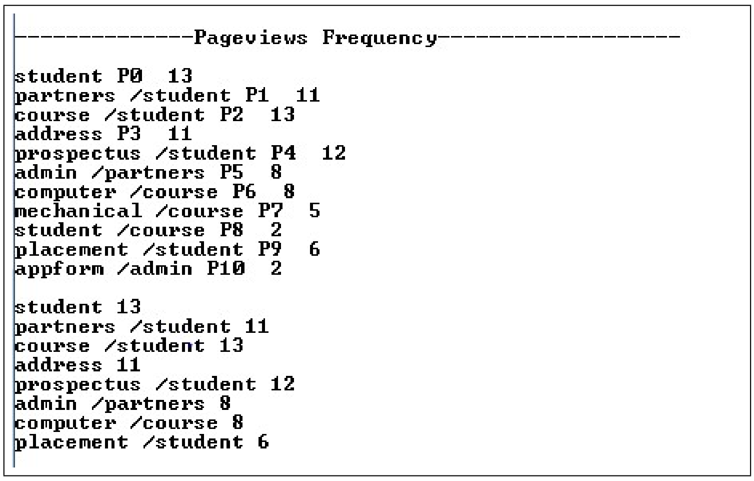
In the next step, Rough clustering is done to make an initial group of the data points, like this many sets of groups are formed shown in figure 5.6, where all the similar data are grouped together for instance s2, s3, s4 are similar.



```
Total pages accessed 106
Total number of sessions 10
Threshold 10

Target set is { S1 S2 S3 S4 S5}
Equivalence set is { {S1,S10} {S2} {S3} {S4} {S5,S6} {S7} {S8} {S9} }
Lower Approximation { S2 S3 S4 }
Upper Approximation { S1 S10 S2 S3 S4 S5 S6 }

---------------Sessions after taking Upper Appoximation--------------

Session0
student ->partners /student ->course /student ->student ->address ->prospectus /

Session1
student ->partners /student ->course /student ->student ->address ->prospectus /

Session2
student ->course /student ->student /course ->student ->partners /student ->cour

Session3
student ->computer /course ->student ->address ->prospectus /student ->placement

Session4
student ->partners /student ->student /course ->address ->prospectus /student ->

Session5
computer /course ->address ->prospectus /student ->placement /student ->student

Session6
computer /course ->address ->prospectus /student ->placement /student ->student
```

Figure 5.6: Clustering Step

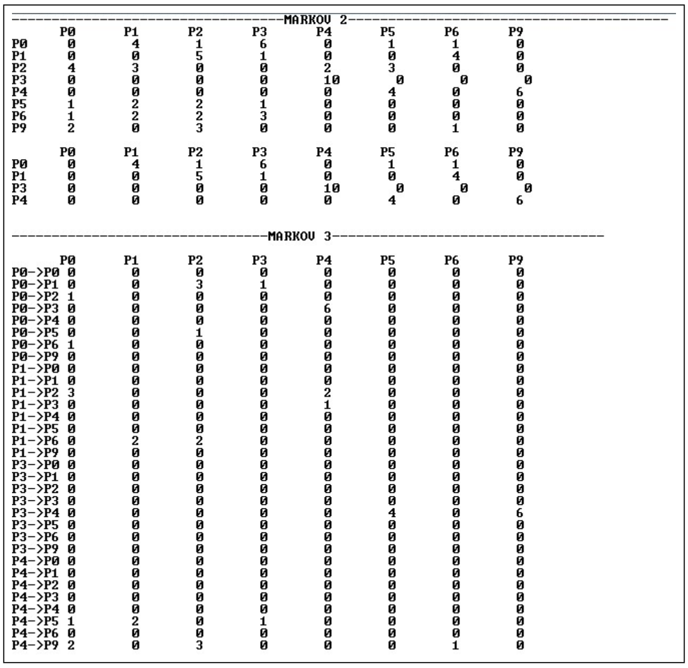
Frequency of web page viewed of each page is calculated to show how many times it has been accessed in figure 5.7



```
--------------Pageviews Frequency--------------------

student P0  13
partners /student P1  11
course /student P2  13
address P3  11
prospectus /student P4  12
admin /partners P5  8
computer /course P6  8
mechanical /course P7  5
student /course P8  2
placement /student P9  6
appform /admin P10  2

student 13
partners /student 11
course /student 13
address 11
prospectus /student 12
admin /partners 8
computer /course 8
placement /student 6
```

Figure 5.7: Page View Frequency

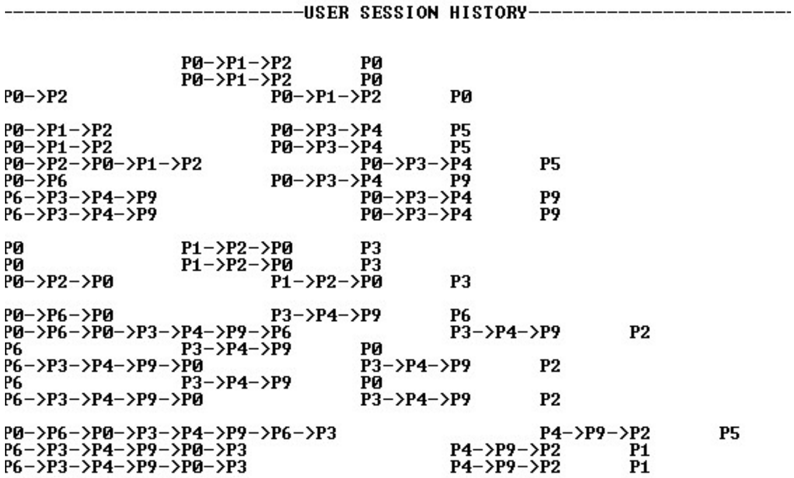
From the frequency count that is collected for every page is used in the Markov model. To find the subsequent state of the pages. From the figure 5.8, we have calculated the matrix for transitions of the pages from p0 going to p0, p0 going to p1, p0 going to p2 and so on for all pages, if p1 is visited from p0 4 times that is recorded. Again Markov model is run to clarify the transition of pages.



```
----------------------------------MARKOV 2----------------------------------------
       P0       P1       P2       P3       P4       P5       P6       P9
P0      0        4        1        6        0        1        1        0
P1      0        0        5        1        0        0        4        0
P2      4        3        0        0        2        3        0        0
P3      0        0        0        0        10        0        0        0
P4      0        0        0        0        0        4        0        6
P5      1        2        2        1        0        0        0        0
P6      1        2        2        3        0        0        0        0
P9      2        0        3        0        0        0        1        0

       P0       P1       P2       P3       P4       P5       P6       P9
P0      0        4        1        6        0        1        1        0
P1      0        0        5        1        0        0        4        0
P3      0        0        0        0        10        0        0        0
P4      0        0        0        0        0        4        0        6


--------------------------------MARKOV 3----------------------------------

       P0       P1       P2       P3       P4       P5       P6       P9
P0->P0 0        0        0        0        0        0        0        0
P0->P1 0        0        3        1        0        0        0        0
P0->P2 1        0        0        0        0        0        0        0
P0->P3 0        0        0        0        6        0        0        0
P0->P4 0        0        0        0        0        0        0        0
P0->P5 0        0        1        0        0        0        0        0
P0->P6 1        0        0        0        0        0        0        0
P0->P9 0        0        0        0        0        0        0        0
P1->P0 0        0        0        0        0        0        0        0
P1->P1 0        0        0        0        0        0        0        0
P1->P2 3        0        0        0        2        0        0        0
P1->P3 0        0        0        0        1        0        0        0
P1->P4 0        0        0        0        0        0        0        0
P1->P5 0        0        0        0        0        0        0        0
P1->P6 0        2        2        0        0        0        0        0
P1->P9 0        0        0        0        0        0        0        0
P3->P0 0        0        0        0        0        0        0        0
P3->P1 0        0        0        0        0        0        0        0
P3->P2 0        0        0        0        0        0        0        0
P3->P3 0        0        0        0        0        0        0        0
P3->P4 0        0        0        0        0        4        0        6
P3->P5 0        0        0        0        0        0        0        0
P3->P6 0        0        0        0        0        0        0        0
P3->P9 0        0        0        0        0        0        0        0
P4->P0 0        0        0        0        0        0        0        0
P4->P1 0        0        0        0        0        0        0        0
P4->P2 0        0        0        0        0        0        0        0
P4->P3 0        0        0        0        0        0        0        0
P4->P4 0        0        0        0        0        0        0        0
P4->P5 1        2        0        1        0        0        0        0
P4->P6 0        0        0        0        0        0        0        0
P4->P9 2        0        3        0        0        0        1        0
```

Figure 5.8: Using Markov Model

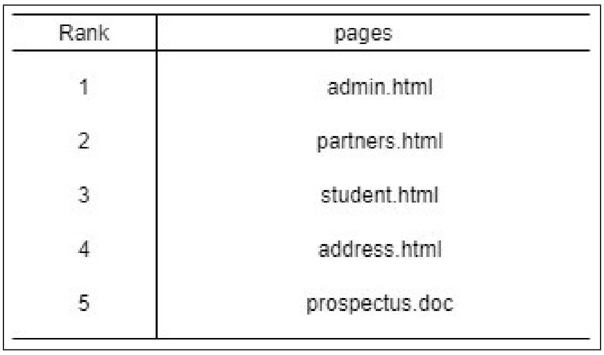
Session history of each state is gathered from Markov model, shown in figure 5.9, which shows the path of pages visited for every sessions.



```
---------------------------USER SESSION HISTORY------------------------


                P0->P1->P2      P0
                P0->P1->P2      P0
P0->P2                  P0->P1->P2      P0

P0->P1->P2              P0->P3->P4      P5
P0->P1->P2              P0->P3->P4      P5
P0->P2->P0->P1->P2              P0->P3->P4      P5
P0->P6                  P0->P3->P4      P9
P6->P3->P4->P9                  P0->P3->P4      P9
P6->P3->P4->P9                  P0->P3->P4      P9

P0              P1->P2->P0      P3
P0              P1->P2->P0      P3
P0->P2->P0              P1->P2->P0      P3

P0->P6->P0              P3->P4->P9      P6
P0->P6->P0->P3->P4->P9->P6              P3->P4->P9      P2
P6              P3->P4->P9      P0
P6->P3->P4->P9->P0              P3->P4->P9      P2
P6              P3->P4->P9      P0
P6->P3->P4->P9->P0              P3->P4->P9      P2

P0->P6->P0->P3->P4->P9->P6->P3                  P4->P9->P2      P5
P6->P3->P4->P9->P0->P3                  P4->P9->P2      P1
P6->P3->P4->P9->P0->P3                  P4->P9->P2      P1
```

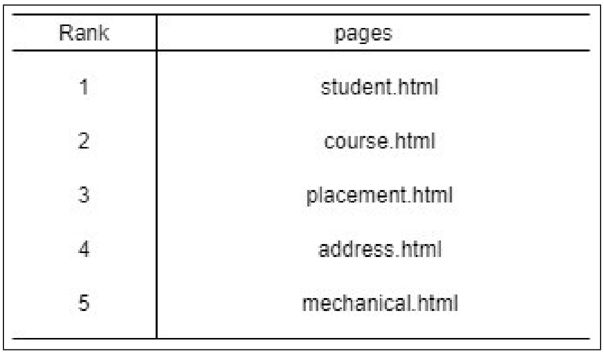
Figure 5.9: User Session History

From the session history shown in figure 5.9, query which are found for each pages and with the path and links from session history, hubs and authorities are calculated with the improved hits algorithm considering popularity and relevance and the pages are given a score. After that with the hubs and authorities calculated we searched for the word Student with 2 popular page ranking methods Hits, Arc and our improved Hits. The main purpose of searching a word with the algorithms is to rank the pages by assigning a score to each page, which is done by the improved hits algorithm, arc and hits. The results are shown in the tables below.



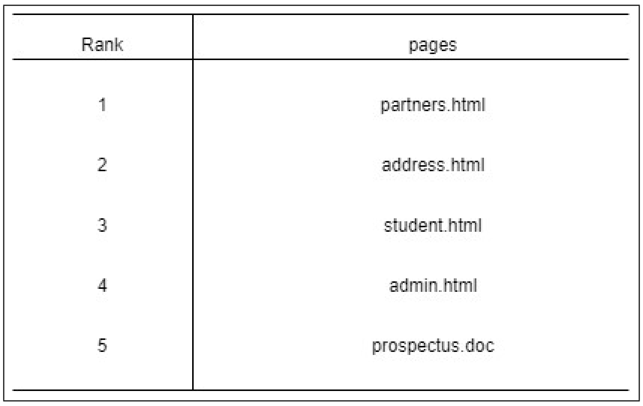
Rank	pages
1	admin.html
2	partners.html
3	student.html
4	address.html
5	prospectus.doc

Table 5.1: Rank list On Student with hits algorithm



Rank	pages
1	student.html
2	course.html
3	placement.html
4	address.html
5	mechanical.html

Table 5.2: Rank list on Student with improved hits algorithm.



Rank	pages
1	partners.html
2	address.html
3	student.html
4	admin.html
5	prospectus.doc

Table 5.3: Rank list on Student with Arc algorithm.

The table consists of the ranked pages and shown the descending order of which pages with the score calculated from Hits, Arc and improved Hits. So considering table 5.1, 5.2 and 5.3 we can see only table 5.2 with the improved hits put the correct page "Student" in the first rank which was our search query, from this we can determine that relevant pages will be shown more frequently and topic drift will not occur. Furthermore for the second rank it is the "courses" page. While on Arc it is "address" and on Hits its "partners" thus showing after students the popular page courses only shows in improved Hits model. From this above tables we can determine our model does better compared to the two in this instance. We also searched this query 10 times to get a definitive result.

In addition to "student" we searched "admin" and "course" with Hits, improved HITS and Arc on the same data set. And we took 10 results for each query to determine the outcome and made a comparison which is shown in table 5.4.

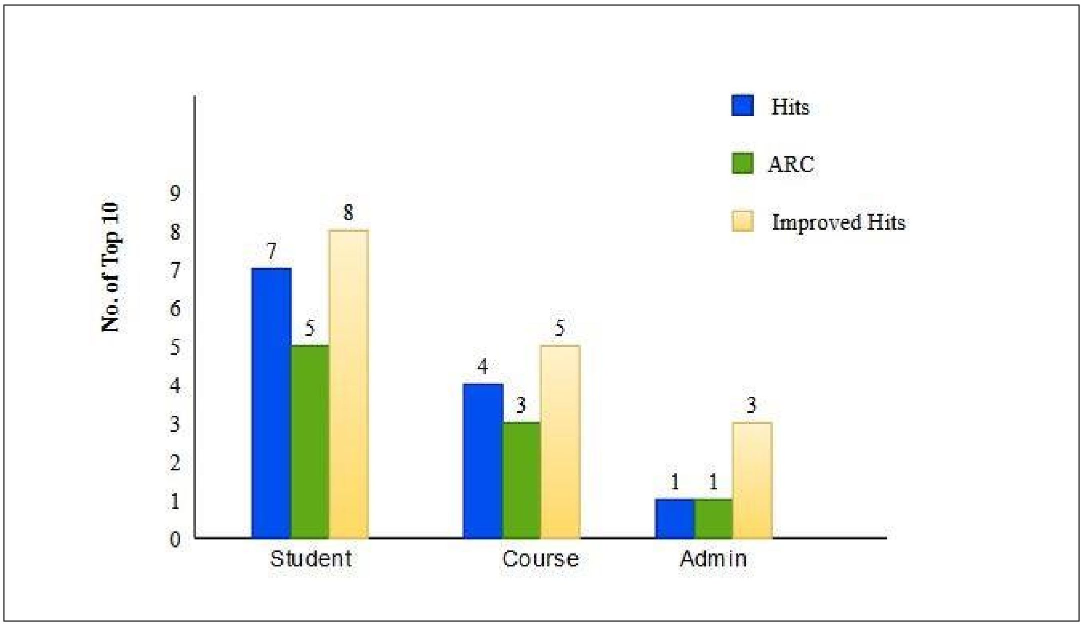


Table 5.4: Proportion of Result

From table 5.4 we can see, the query Student, 7 of the top 10 matches. 5 of the top 10 results returned by Arc matches. The top 8 results returned by improved HITS are matched, 8 of them are relevant. To the query course, 4 of the top 10 results returned by Hits are matched. 3 of the top 10 results returned by Arc are same. 5 of the top 10 results returned by improved HITS are matched. To the query admin, 1 of the top 10 results returned by Hits is matched. 1 of the top 10 results returned by Arc is same. 3 of the top 10 results returned by improved HITS are relevant From Table 2. Taking values from table 5.4 and calculating the percentage of each query result matched/ 10, our method gives the outcome of 10% to 50% better compared to Hits and Arc. Hence we determine our method is better.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Amount of people accessing the internet is increasing by Millions of day by day; hence the bottleneck is created by insufficient bandwidth is a major problem. Reducing latency is the ultimate solution to this problem. Hence our improved Hits model significantly reduces the latency by implementing a new perfecting method and storing it in the cache for browsers to use. Although certain limitations and errors might occur due to not implementing real servers and we plan to solve in the near future

6.2 Future Work

Our future plan for the improved hits algorithm is to work with a better dataset; we used a limited dataset from a proxy server due to limits for Covid-19. Therefore we plan to use a dataset with more unique and vast values containing similar queries. Furthermore, we had to test data from a proxy server and implement it in an IDE mimicking the actual server. In the future, we plan to test it on a real sever to get more accurate results

Bibliography

[1] Xing Dongshan and Shen Junyi, "A new markov model for web access prediction," *Computing in Science Engineering*, vol. 4, no. 6, pp. 34–39, 2002. DOI: 10.1109/MCISE.2002.1046594.

[2] A. Singh and A. K. Singh, "Web pre-fetching at proxy server using sequential data mining," in *2012 Third International Conference on Computer and Communication Technology*, 2012, pp. 20–25. DOI: 10.1109/ICCCT.2012.14.

[3] P. Kumar, S. Kadambari, and S. Rawat, "Prefetching web pages for improving user access latency using integrated web usage mining," in *2015 Communication, Control and Intelligent Systems (CCIS)*, 2015, pp. 401–405. DOI: 10.1109/CCIntelS.2015.7437949.

[4] J. Liao, F. Trahay, B. Gerofi, and Y. Ishikawa, "Prefetching on storage servers through mining access patterns on blocks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 9, pp. 2698–2710, 2016. DOI: 10.1109/TPDS.2015.2496595.

[5] K. A. Phan, Z. Tari, and P. Bertok, "Similarity-based soap multicast protocol to reduce bandwith and latency in web services," *IEEE Transactions on Services Computing*, vol. 1, no. 2, pp. 88–103, 2008. DOI: 10.1109/TSC.2008.8.

[6] W. Yang, "An improved hits algorithm based on analysis of web page links and web content similarity," in *2016 International Conference on Cyberworlds (CW)*, 2016, pp. 147–150. DOI: 10.1109/CW.2016.30.

[7] E. Cohen and H. Kaplan, "Prefetching the means for document transfer: A new approach for reducing web latency," in *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, vol. 2, 2000, 854–863 vol.2. DOI: 10.1109/INFCOM.2000.832260.

[8] Y. Xu and Y. Han, "Alirs: A high scalability and high cache hit ratio replacement algorithm," in *2011 International Conference on Computational and Information Sciences*, 2011, pp. 66–70. DOI: 10.1109/ICCIS.2011.66.

[9] R. R. Raluca Tanase, 2009. [Online]. Available: http://pi.math.cornell.edu/~mec/Winter2009/RalucaRemus/Lecture3/lecture3.html.

[10] *PageRank*, en, Page Version ID: 998926215, Jan. 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=PageRank&oldid=998926215 (visited on 01/07/2021).

[11] P. K. Kang, M. Dentz, T. Le Borgne, and R. Juanes, "Spatial markov model of anomalous transport through random lattice networks," *Phys. Rev. Lett.*, vol. 107, p. 180 602, 18 Oct. 2011. DOI: 10.1103/PhysRevLett.107.180602.

[12] L. Rabiner and B. Juang, “An introduction to hidden markov models,” *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4–16, 1986. DOI: 10.1109/MASSP.1986.1165342.

[13] K. M. Sagayam and D. J. Hemanth, “A probabilistic model for state sequence analysis in hidden markov model for hand gesture recognition,” *Comput. Intell.*, vol. 35, no. 1, pp. 59–81, 2019. DOI: 10.1111/coin.12188. [Online]. Available: https://doi.org/10.1111/coin.12188.

[14] K. M. A. Patel and P. Thakral, “The best clustering algorithms in data mining,” in *2016 International Conference on Communication and Signal Processing (ICCSP)*, 2016, pp. 2042–2046. DOI: 10.1109/ICCSP.2016.7754534.

[15] Jianliang Xu, Jiangchuan Liu, Bo Li, and Xiaohua Jia, “Caching and prefetching for web content distribution,” *Computing in Science Engineering*, vol. 6, no. 4, pp. 54–59, 2004. DOI: 10.1109/MCSE.2004.5.

[16] N. Megiddo and D. S. Modha, “Outperforming lru with an adaptive replacement cache algorithm,” *Computer*, vol. 37, no. 4, pp. 58–65, 2004. DOI: 10.1109/MC.2004.1297303.

[17] Z. Li, D. Liu, and H. Bi, “Crfp: A novel adaptive replacement policy combined the lru and lfu policies,” in *2008 IEEE 8th International Conference on Computer and Information Technology Workshops*, 2008, pp. 72–79. DOI: 10.1109/CIT.2008.Workshops.22.

[18] N. Megiddo and D. Modha, “Arc: A self-tuning, low overhead replacement cache,” in *FAST*, 2003.

[19] *Modern file systems*, https://www.slideshare.net/DavidEvansUVa/flash-modern-file-systems, (Accessed on 01/08/2021).

[20] L. Yan, Y. Wei, Z. Gui, and Y. Chen, “Research on pagerank and hyperlink-induced topic search in web structure mining,” in *2011 International Conference on Internet Technology and Applications*, 2011, pp. 1–4. DOI: 10.1109/ITAP.2011.6006308.