CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The growth of World Wide Web (WWW) has been phenomenal in the last few decades. The reliance on the web systems for day to day activities has been increased tremendously in the recent years. WWW has become an integral part of the life of many people. Web services are frequently used for many purposes like mail communication, gathering information about various products and services, education, research, social interactions, financial transactions, games, music and entertainment etc. The explosive growth of the web has imposed a heavy demand on networking resources and web servers.

One of the most significant parameter which determines the quality of web access systems is the network bandwidth. As the demand for the web services is increasing steadily, the demand for the network bandwidth has grown rapidly and the issues like network traffic has also increased to a larger extent. It is well known that the bandwidth depends on the network infrastructures and the web performance depends on the web traffic. The greater the web traffic, the larger the latency in accessing the web objects. Poor web infrastructures in many developing countries render persistent web traffic and higher latencies. Higher bandwidth alone will not solve the problem, more than that technological understanding of the underlying pattern of web traffic, interactions among the clients, proxy and servers provides
interesting insights in improving the web access and overall web performance.

1.2 WEB CACHING

Web browsing dominates today's internet. More than two thirds of the traffic on the Internet today is generated by the web. The number of available web pages has increased enormously; the number of web users has also increased manifold and sensitive information on internet decreases the performance and security of web.

Although the bandwidth of the connections has increased manifold in the last decade, the user perceived latencies is still a dominant issue in navigating the web due to overloaded elements (network, servers, switches, or intermediate hardware), long message transference times, and the Round Trip Time (RTT). Consequently, the reduction of the users perceived latency when browsing the web is considered as an important research issue.

A number of techniques have been proposed to reduce the user’s perceived latency in the web systems. Web caching, replication, and prefetching are the most popular techniques and widely implemented in achieving the latency savings. It is very common to observe that big companies usually implement web replication by using CDNs (Content Delivery Networks) (Rabinovich & Spatscheck 2003) to reduce their websites access time, but this solution is expensive and many small companies and organizations cannot afford it.

“A cache is a copy of a portion of the data from the data provider to a smaller and faster storage device (cache) interposed between the data consumer and the data provider, so that future data accesses can be resolved from the cache with less cost” (Du 2005). It is important to note that caching
is a logical entity instead of a physical entity. A cache may have different physical presentations, but they serve the same logical functionality. In a storage hierarchy as shown in Figure 1.1, each storage layer has a different physical form, but they all function in the same way of serving as the cache for the layer above them. Several studies have shown that the performance of the web system can be improved by web caching ((Duska et al 1997, Kroeger et al 1997, Caceres et al 1998).

![Figure 1.1 Storage hierarchy](image)

**Figure 1.1 Storage hierarchy**

Wang (1999) identified several advantages of using web caching which are listed below:

- Web caching reduces bandwidth consumption, thereby decreases network traffic and reduces network congestion
- The frequently accessed web objects are stored in the nearby proxy caches and fetched from them instead of remote data servers, thereby minimizing the transmission delay.
• The web objects that are not cached can also be retrieved relatively faster because of the reduction in network traffic.

• Web caching reduces the workload of the remote web server by distributing data among the proxy caches over the wide area network.

• The robustness of the web service is greatly enhanced by web caching. For example, if the remote server is not available due to the remote server’s crash or network partitioning, the client can obtain a cached copy at the proxy, thus assuring continuous service.

1.2.1 Classification of Web Cache

Web caching can be deployed at three levels - client level, proxy level and original server level as shown in Figure 1.2.

• Browser cache

It is often located in the client side. Browser cache stores the web pages in cache memory. It enables the user to retrieve the recently used web pages instantaneously from its local storage.

• Proxy cache

It is found in the proxy server which located between client machines and origin servers. The principle of working of proxy server cache is same as that of browser cache, but it is operated in a much larger scale. Unlike the browser cache, the proxy cache serves several users of the order of hundreds or thousands at the same time. When a request is received, the proxy server checks its cache. If the object is available, it sends the object to the client. If the object is not available, or it has expired, the proxy server will
request the object from the origin server and send it to the client. The object will be stored in the proxy’s local cache for future requests.

- **Server cache:**

  In a server cache, the web pages are stored in the origin server side. The advantage of the server cache is that it greatly reduces the need for redundant computations or database retrievals. Thus, the server load can be reduced if the origin server cache is employed.

  According to (Mookerjee & Tan 2003), client-side caching (also called as browser caching) is more economical and effective than server or proxy caching.

![Classification of Web Cache](image)

**Figure 1.2 Classification of Web Cache**

### 1.2.2 Cache Replacement Algorithms

Generally, cache storage has limited capacity and, as a result, cache storage will ultimately become full as users continuously browse the internet. Thus, it enables the storage of the new object in the cache only when the
existing object is removed. A cache replacement policy will determine which object will be removed so that enough space is created for the new object (Jarukasemratana & Murata 2013). As a general rule, objects that will not be used again should be removed from the cache first, so as to use the limited cache space in the most efficient way. Many cache replacement policies have been developed and each posses their own algorithms for selecting the objects to be removed from the cache. The overall aim of cache replacement policies is to increase cache hit rate.

1.2.3 Properties of WWW Caching System

This section highlights the important properties of the web caching system. The list of properties and the desirable features of the efficient web caching system is discussed here.

- Fast access

  Access latency is important in determining the quality of web service. A desirable caching system should have reduced web access latency. The average latency of a web caching should be lower than that without employing a caching system.

- Robustness

  It is a measure of quality of web service which guarantees the availability of web service to web user. A robust system has to ensure three things. First, a proxy crash should not down the entire system. This can be taken care of by eliminating the single point failure in the caching system. Second, the caching system should fall back gracefully in case of failures. Third, the design of the caching system should ensure that it recovers easily from a failure.
- **Transparency**

  From the user perceptiveness, a web caching system should be transparent and noticeable results in the web caching system should be faster response and higher availability.

- **Scalability**

  The scalability of the web cache should match the rapidity of the growth of internet. The scalable caching scheme would accommodate the increasing size and density of network in a better manner. The protocol employed in the caching systems has to be lightweight if scalability is considered.

- **Efficiency**

  Efficiency of WWW caching depends on overhead that web caching system imposes on the network. The caching system should not adopt any scheme which leads to under utilization of critical resources in network.

- **Adaptivity**

  The caching system need to be adapted to the dynamic changing of the user demand and the network environment. The adaptivity involves several aspects: cache management, cache routing, proxy placement, etc. This is essential to achieve optimal performance.

- **Stability**

  The caching schemes used in web caching system should not introduce instabilities into the network.
• **Load balancing**

   It is desirable that the caching scheme distributes the load evenly through the entire network. A single proxy or server should not be overloaded and thereby degrading the performance of the network.

• **Ability to deal with heterogeneity**

   The hardware and software architectures are evolving as the network grows. An efficient web caching scheme has to adapt to a range of network architectures.

• **Simplicity**

   Simplicity is highly desirable as simpler schemes are easier to implement and likely to be accepted as international standards.

1.3 **PREFETCHING**

   Although caching has several advantages in web based systems, its benefit diminishes as web documents become more dynamic (Douglis et al 1997). Sometimes the cached object may be stale at the time of its request. The reason may be the passive nature of most of the web caching systems in use today. Web prefetching techniques are orthogonal to caching and replication techniques, so that they can be applied together to achieve a better web performance. The function of the prefetching is to overcome the limitations of passive caching by proactively fetching the objects before it is being requested. Prefetching scheme predicts the demand for the web objects that are more likely to be accessed in the near future and fetches them even before it is actually demanded (Balamash et al 2007).
Prefetching schemes employs a prediction engine which decides which objects from the predicted hints are going to be prefetched. The prediction engine uses the information depending on the elements of web architecture like client, proxy and server. When the prediction engine is located at the client, only one user access pattern is used to perform predictions (Duchamp 1999).

When the prediction engine is located at the proxy, the prediction engine uses the multiuser and multi-server information at the proxy server to perform the predictions (Bouras et al 2004, Teng et al 2005).

When the prediction engine is located at the server side, it takes advantage of the multi-user accesses to the same website to makes predictions (Domenech 2006a, Padmanabhan 1996). Finally, the predictions can be performed by several elements in collaboration (Domenech 2006b).

1.3.1 Classification of Prefetching Algorithms

According to the type of information used to make predictions, the prediction algorithms can be classified into two main groups as past access based algorithms and web content based prediction algorithms (Domenech 2006b). Theses classifications are shown in Figure 1.3.

A. Past access based prediction algorithms

In the past access based prediction algorithms, the future accesses are predicted based on the previous access patterns. These algorithms can be further distinguished into Markov models based prediction algorithms (Padmanabhan & Mogul 1996) and data mining techniques based prediction algorithms (Yang et al 2003). The advantage of Markov models is that they provide high precision in predictions. However, they are expensive in terms
of intensive computation and memory consumption. The resource consumption of data mining based algorithms is also comparatively higher.

**B. Web content based prediction algorithms**

In web content based prediction algorithms, information from the analysis of web content is used to make predictions.

![Classification of prediction algorithms](image)

**Figure 1.3 Classification of prediction algorithms**

1.4 **COOPERATIVE CACHING**

Cooperative caching is one in which multiple caching nodes share and cooperatively manages the cached contents (Yu 2005). In a cooperative caching system, caches deployed in different client systems can cooperate with one another to serve requests from clients in their networks. When a cache receives a request for a web object that is not stored locally, it tries to locate whether another cache in the cache group which has the requested object. If any of them has the object, the object is directly served to the requesting client. Thus cooperative caching brings coordination among all the client caches as shown in Figure 1.4. Dahlin et al (1994) have proposed various approaches for efficient cooperative caching which included Direct...
Client Cooperation, Greedy Forwarding, Centrally Coordinated Caching, N-Chance Forwarding, Hash-Distributed Caching and weighted LRU algorithm.

![Cooperative web caching system](image)

**Figure 1.4 Cooperative web caching system**

1.5 OBJECTIVES

In this section, the main objectives of this thesis are presented.

- To improve the efficiency of retrieved information by using co-operative clients and transparent proxy caching system in a hybrid caching architecture.

- To generate training dataset using Access Log Manager (ALM) to be used by proxy server using machine learning technique like Support Vector Machine (SVM).

- To develop a proxy server cache manager by utilizing efficient techniques like Web Navigational Graph (WNG), graph mining and using modified algorithms for clustering and prefetching the web objects.
- Design a Client Cache Manager (CCM) responsible for effective web object sharing among cooperative clients using Dynamic Hash Table (DHT) based techniques.

- Validate the proposed system using training data sets by employing standard performance metrics.

1.6 PROPOSED SYSTEM ARCHITECTURE

The proposed information retrieval system consists of three major sub-systems which include access log manager with SVM classifier, cluster based proxy server cache manager (PCM) and DHT based co-operative client cache manager system (CCM). The system architecture of the present work is shown in Figure 1.5.

![Figure 1.5 System architecture for information retrieval](image-url)
In the proposed architecture, a single machine has been configured as a server machine by running the proxy server cache application and clients are connected to it. When the application is executed by the client for the first time, the proxy authentication module authenticates the client. The proxy server maintains database of all active clients. Once all the clients are connected to the network, they all join together through a chord protocol for effective peer to peer (P2P) communication.

1.6.1 Access Log Manager (ALM) with SVM Classifier

Web access log is used to record the user’s interactions with the web pages. It plays an important role in predicting the user access pattern, pre-fetching as well as caching of web data. The training data is generated by ALM. Web pages that are referred by various clients are identified from the log file. The web log file contents are pre-processed and trained using the features recency, frequency, retrieval time and size of web object. In order to create a training dataset information is extracted from the traces of log file. SVM classifier is used to identify the log data which is nominated for further processing.

1.6.2 Cluster based Proxy Cache Manager (PCM)

According to the sample data generated by ALM, the proxy server creates clusters using Web Navigational Graph (WGN) and association rule mining techniques. These clusters are maintained by the proxy cache manager. WNG shows the navigations made between the various web objects by each user. Association rule mining technique is used to create sub graphs. The edges are filtered out by their weight. The connectivity between pages is determined by support and confidence parameters. A clustering algorithm takes the contents of WNG as input. A threshold value is fixed for support
and confidence parameters, and the edges which have values less than this threshold are removed.

### 1.6.3 Dynamic Hash Table (DHT) based Cooperative Client Cache Manager System (CCM)

Cooperative caching is supported by a Sharable Object Space (SOS) implemented as a part of the physical memory of the cluster nodes. The client cache manager is responsible for caching web objects and sharing the object through SOS. With cooperative caching a web object being requested can be served from a client node’s local memory or from a SOS web cache as a cache hit. The search is done in proxy cache clusters if there is a miss still. Chord protocol is used for sharing of SOS objects as a peer to peer distributed hash table. The hashing, routing and resource searching algorithms are used for object sharing in co-operative client environment.

Hashing algorithm is used to join a client in the chord system. It generates unique keys for its sharable web objects using Secure Hashing Algorithm1(SHA1) consistent algorithm and then send them to their successor node in the chord ring. Routing algorithm is used to identify the successor node whose key values are less than or equal to the object key. Resource searching algorithm enables the searching of object in local browser web cache.

A hybrid algorithm was developed which handles the movement of web objects out of SOS if there is no enough space to store new object in SOS. The hybrid algorithm calculates a key for each object based on access frequency, recency, size and access latency. The objects with low key value will have high priority to evict first. The deleted object notification is updated in both proxy cache and in the key value pair table of the client who is
holding that object key. This will avoid the “false hit” in case of object non existence status.

Each client maintains a cluster of objects based on specific domains using the single link method. It reflects the user access pattern. A client post a request which is already maintained by some other client, the destination client send more number of objects instead of a single object to the target client because the resultant object is maintained within a cluster of destination client. This type of prediction and prefetching improves the performance by reducing the user perceived latency and response time.

1.7 PERFORMANCE METRICS

The list of performance metrics most commonly used in evaluation of information retrieval system are listed in the following prose.

1.7.1 Web Caching Performance Metrics

Some of the most commonly used metrics for evaluating performance of the web caching systems are give below.

A. Hit Ratio (HR)

It is the ratio of total no of request served by the cache to the total number of request given.

\[
HR = \frac{\text{No.of request served by the cache}}{\text{Total No.of request given}} \quad (1.1)
\]
B. **Byte Hit Ratio (BHR)**

It is the ratio of bytes served by the cache to the total number of bytes required.

\[
HR = \frac{\text{No. of bytes served by the cache}}{\text{Total No. of bytes required}} \quad (1.2)
\]

### 1.7.2 Prefetch Metrics

Prefetch metrics are aimed at quantifying the performance provided by the prediction algorithm. Domènech et al (2006) synthesized the literature on web prefetching performance metrics and the most commonly used metrics are highlighted here.

**A. Precision \((P_c)\)**

Precision is a measure of the ratio of a good prediction to the number of predictions (Equation 1.3). Precision can be considered as a theoretical index, which just evaluates the algorithm without considering physical system restrictions like cache, network or time restrictions.

\[
P_c = \frac{\text{Good Predictions}}{\text{Predictions}} \quad (1.3)
\]

Precision can also be measured based on performance by measuring the ratio between the number of prefetched objects and prefetch hits as shown in Equation 1.4.

\[
P_c = \frac{\text{Prefetch Hits}}{\text{Prefetches}} \quad (1.4)
\]
B. **Recall (R_c)**

Recall measures the percentage of requested objects that were previously prefetched. The recall quantifies the weight of the predicted (Equation 1.5) or prefetched objects (Equation 1.6) over the amount of objects requested by the user. Recall is also sometimes referred to as usefulness, hit ratio, or accuracy.

\[
P_c = \frac{\text{Good Predictions}}{\text{User Requests}} \quad (1.5)
\]

\[
R_c = \frac{\text{Prefetch Hits}}{\text{User Requests}} \quad (1.6)
\]

C. **Applicability**

Applicability can be defined as the ratio of the number of predictions to the number of requests (Bonino et al 2003).

\[
\text{Applicability} = \frac{\text{Predictions}}{\text{User Requests}} \quad (1.7)
\]

D. **Byte Precision (P_{cB}).**

Byte precision measures the percentage of predicted (or prefetched) bytes that are subsequently requested.

\[
P_{cB} = \frac{\text{Good Predictions}}{\text{Predictions}} \quad (1.8)
\]
E. **Byte Recall (R_{cB})**

Byte recall measures the percentage of demanded bytes that were previously predicted (or prefetched). It quantifies how many accurate predictions are made, measured in transferred bytes.

\[
R_{cB} = \frac{\text{Good Predictions}}{\text{User Requests}}
\]  

(1.9)

1.8 **THESIS ORGANIZATION**

The first chapter begins with an introduction to the web systems and traffic in the web access. A short introduction to caching, prefetching and cooperative caching is provided. Then the objectives of the thesis, architecture of the proposed system is described. Various performance metrics are highlighted in the final section. The rest of this thesis is organized as follows.

Chapter 2 reviews the key literature based on information retrieval involving web objects. The chapter begins with an introduction to web caching which is the base of the present thesis work. Caching algorithms are then introduced. A detailed synthesis of various studies based on caching schemes is also reported. Then, prefetching, a more closely related technique to caching is discussed. A number of studies based on web prefetching techniques is also reviewed. Various studies on effectively integrating web caching and prefetching is highlighted. Finally cooperative caching which is the central theme of the thesis is discussed. A number of studies based on cooperative caching is reviewed in detail. An overall summary of the literature review is included in the final section.
Chapter 3 presents the functions and features of access log manager. It describes the method of generating training data set using a log file. The procedure in data pre-processing, cleaning is described in detail. The method of employing machine learning algorithm to identify the pattern in the data using SVM classifier is elaborated.

Chapter 4 discusses clustering in proxy cache management. It details the procedure in the construction of web navigational graph, cluster creation, prediction and prefetching technique. A hybrid Combined Recency and Frequency (CRF) algorithm for cache management is also given.

Chapter 5 explains the Cooperative Client cache management system. It describes the dynamic hash table based chord network creation, finger table & Key value pair table creation. Number of algorithms is also provided in the chapter like Node Join algorithm, Hashing algorithm, Routing algorithm, Resource search and Node exit algorithm. It describes the query integrator which acts as an interface between the user and the proposed system.

Chapter 6 presents the experimental results of the study. The first part explains the dataset used in the experiment and simulation setup in detail. In the next section, validation of the SVM classifier is presented, followed by results of the comparison of hybrid algorithm with other algorithms is discussed. The results of the prefetching technique and effectiveness of cooperative caching is also displayed.

Chapter 7 provides overall conclusion about the thesis. A separate section on future scope of work is also provided.