A Novel Approach for Prefetching of Web Pages through Clustering of Web Users to Reduce the Web Latency

G.T. Raju[1] and M.V. Sudhamani[2]

[1] Dept. of Computer Science and Engineering
RNS Institute of Technology, Bangalore-98
gtraju1990@yahoo.com
[2] Dept. of Information Science and Engineering
RNS Institute of Technology, Bangalore-98
mvsudha_raj@hotmail.com

Abstract. Web users are experiencing a long latency while retrieving the Web pages due to the amount of network traffic increased with the WWW expansion. Potential sources of latency are the Web servers' heavy load, network congestion, low bandwidth, bandwidth underutilization, and propagation delay. To solve the latency problem, prefetching technique that predicts the destination pages for user community has become critical to save the communication overhead. Prefetching means fetching of Web pages before the users request them so as to reduce the user perceived latency. A novel Cluster and Prefetch (CPF) approach is proposed in this paper. Experimental results shows that the CPF approach effectively reduces the user perceived latency without wasting the network resources with high prediction accuracy.

1 Introduction

Prefetching technique is motivated by the fact that, in general, once a user goes to a Web site; he/she generally browses around for several pages before leaving for another site. Since the user follows hyperlinks upon his/her interests, it is likely that links are not followed uniformly. It is possible to either predict each user's interest using cookies or mine a consensus of interests with some confidence from access log files recorded by the Web server. This information not only is valuable for the Web administrator to eliminate uninterested pages, or balance load among the servers, but also can help to improve Web-browsing time. Most prefetching techniques predict the Web page requests for individual user. These techniques can easily overload the network when there are large numbers of users. To overcome this, Cluster and Prefetch (CPF) approach is proposed in this paper. This approach uses the ART1 NN clustering algorithm for clustering the Web users. The prototype vector of each cluster gives a generalized representation of the Web pages that are most frequently requested by all the members of that cluster. Whenever a host connects to the server or a proxy, the proposed prefetching strategy returns the Web pages for the cluster to which the host belongs to. Advantage of the CPF approach is that the better network resource utilization by prefetching the Web pages for a user community rather than a single user, thereby improving the Web browsing time of user.

Aswatha Kumar M. et al. (Eds.): Proceedings of ICAdC, AISC 174, pp. 983–989.
springerlink.com © Springer India 2013

2 Analytical Model of Web Prefetching

The objective of analytical model for web prefetching is to study the perceived latency in retrieving a web page by a browser/user with the given web traffic parameters such as – *number of users* and the *bandwidth of access link* etc.,

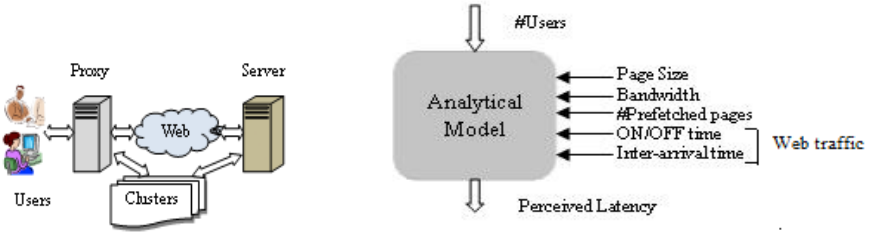

Fig. 1. (a) Perspective of Web Prefetching (b) Analytical Model of Web Prefetching

Perspective of Web prefetching and its analytical model are as shown in Fig. 1. When a user request for a page at particular time instance, the proxy identifies the web user and the cluster to which user belongs to. If the page is already prefetched by the prefetcher and is consistent with the original page on the remote server, the proxy sends the page to the user (Hit). If the proxy does not have a copy of the requested page, then the proxy prefetches the pages represented by the prototype vector of the cluster to which the user belongs to, sends the requested page to user (Miss) and keeps a copy in the cache. What is interested here is the page delivering latency or the response time which is defined as the time interval from the browser clicking an object to the requested object being displayed on the monitor. The response time depends on various parameters such as: *Web traffic (ON time, OFF time, Inter-arrival time), Page size , Number of prefetched objects, Number of users, and the Bandwidth of access link.*

Web traffic is modeled as *ON-OFF* process which is shown in Fig. 2, with *ON state* corresponding to the request and downloading time of the objects and the *OFF state* corresponding to the inactive time (viewing time).

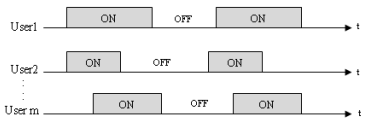

Fig. 2. ON-OFF Web traffic model

ON state is initiated by user click on the hyper link and the page is downloaded during this state. It is possible that the *ON state* lasts over multiple web request periods when the downloading of the last embedded object and the next HTML object overlaps. For example this can happen when the user requests a new object in the middle of the object download. *ON state* is found to follow a Weibull distribution whose probability density function is given by

$$f(x) = \frac{\beta}{\delta}\left(\frac{x}{\delta}\right)^{\beta-1} \exp\left[-\left(\frac{x}{\delta}\right)^{\beta}\right] \quad for\ x > 0 \tag{4}$$

Where β is the shape parameter and δ is the scale parameter. Let x_2 be the continuous random variable represents the ON time with β=0.77 and δ = $e^{4.4}$. Intervals between adjacent requests to follow another Weibull distribution with β=0.5 and δ=1.5. Let x_2 be the continuous random variable represents inter-arrival time.

OFF state is the user viewing time or time that the user is away from the system. OFF state is found to follow Pareto distribution whose probability density function is given by

$$f(x) = \alpha k^{\alpha} x^{-\alpha-1} \tag{5}$$

Where α is the shape parameter and *k* is the scale parameter. Let x_3 be the continuous random variable represents the OFF time with α=0.58 and *k*=60.

Page size to follow another Pareto distribution with α= 1.3 and k=300. Let x_4 be the continuous random variable represents *page size*. Number of *prefetched objects* to follow Weibull distribution with β=0.5 and δ=1.5 represented by continuous random variable x_5 and number of *non-prefetched objects* to follow another Weibull distribution with β=0.9 and δ =4 represented by another continuous random variable x_6.

Let *m* be the *number of users* (250, 500, and 1000) and *B* be the *bandwidth* of the access link (512kbps, 2048kbps, and 4056kbps). Let *c* be a constant whose value is given by $c = B/m$.

The joint probability density function of the *response time (latency)* with prefetching is modeled as

$$f_{X_1X_2X_3X_4X_5}(x_1,x_2,x_3,x_4,x_5) = c[6.986\times10^{-4}(net)x_4^{2.1}x_5^{-0.5}e(-0.816x_5^{0.5})] \tag{6}$$

Where $$net = [2.59\times10^{-2}x_1^{-0.23}e(-0.03x_1^{0.77}) + 0.4078x_2^{-0.5}e(-0.816x_2^{0.5}) + 6.99x_3^{-1.58}] \tag{7}$$

The joint probability density function $f_{X_1X_2X_3X_4X_5}(x_1,x_2,x_3,x_4,x_5)$ satisfies the following properties:

(1) $f_{X_1X_2X_3X_4X_5}(x_1,x_2,x_3,x_4,x_5) \geq 0$ for all x_1, x_2, x_3, x_4, x_5

(2) $$\int_0^{86400}\int_{x_1}^{86400}\int_0^{86400}\int_0^{1048}\int_0^{1000} f_{X_1X_2X_3X_4X_5}(x_1,x_2,x_3,x_4,x_5)\,dx_1\,dx_2\,dx_3\,dx_4\,dx_5 = 1$$

(3) For any region *R* of p-dimensional space

$$P([X_1,X_2,X_3,X_4,X_5] \in R) = \iint_R \int\int\int f_{X_1X_2X_3X_4X_5}(x_1,x_2,x_3,x_4,x_5)dx_1dx_2dx_3dx_4dx_5$$

Similarly, the joint probability density function of the *response time (latency)* without prefetching is modeled as

$$f_{X_1X_2X_3X_4X_6}(x_1,x_2,x_3,x_4,x_6) = c[4.4246\times10^{-4}(net)x_4^{2.1}x_6^{-0.1}e(-0.287x_6^{0.9})] \tag{8}$$

The joint probability density function $f_{X_1X_2X_3X_4X_6}(x_1,x_2,x_3,x_4,x_6)$ satisfies the following properties:

(1) $f_{X_1X_2X_3X_4X_6}(x_1,x_2,x_3,x_4,x_6) \geq 0$ for all x_1,x_2,x_3,x_4,x_6

(2) $\int_0^{86400}\int_{x_1}^{86400}\int_0^{86400}\int_0^{1048}\int_0^{1000} f_{X_1X_2X_3X_4X_6}(x_1,x_2,x_3,x_4,x_6)dx_1dx_2dx_3dx_4dx_6 = 1$

(3) For any region *R* of p-dimensional space

$$P([X_1,X_2,X_3,X_4,X_6] \in R) = \iiint_R \int\int\int f_{X_1X_2X_3X_4X_6}(x_1,x_2,x_3,x_4,x_6)dx_1dx_2dx_3dx_4dx_6$$

Experiments have been conducted to show the comparative analysis on the user perceived latency from the analytical model and the actual trace for varying number of users and given bandwidth. Results presented in section 4.2 shows that the response time from the model and the trace is fairly well matched.

3 The CPF Model

The architecture of the CPF model is as shown in Fig. 3. The feature extractor module extracts each client's feature vector. ART1 clustering module identifies the group to which the client belongs to and returns that group's prototype vector. The prefetching module prefetches the URLs that are most frequently accessed by all the members (hosts) of that cluster represented by a prototype vector. The proxy server responds to the client with prefetched URLs. The prefetching accuracy is measured by predicting the URLs for each member of the cluster and then the prediction is verified with access logs recorded for the next *t* days (prediction period). The pseudo code for CPF approach is given in Fig. 4.

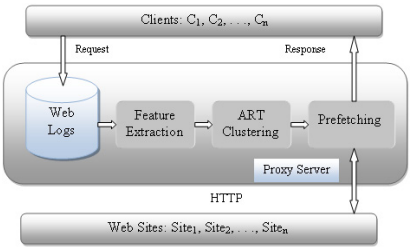

Fig. 3. Architecture of CPF Approach

```
CPF (Host_Id)
 {
        //Takes input: Host_Id of the host that request a URL
        //Cluster the hosts using ART Neural Network Clustering Algorithm;
  ART1_Clustering(P,ρ);
        /* P is the Array of Pattern vectors and ρ is the Vigilance Parameter. Let 'n' is the
number of clusters and C1, C2,….,Cn are the clusters represented by the prototype vectors.
The prototype vector for the kth cluster is of the form Tk = ( tk1, tk2, …, tkm) where tkj for
j=1,2,..m are the top-down weights corresponding to node k in layer F2 of the ART neural
network. */
   Initialize Count=0;
   Repeat for each cluster Ck of the n clusters
   If ( Host_ID is a member of cluster Ck)
   {
    Repeat for j = 1,2,….m
    If ( tkj = 1) {
     Prefetched_URLs[Count] = URLj
     Count++; }
   }
   Return Prefetched_URLs[] ;
 } // End of CPF prefetching scheme
```

Fig. 4. Pseudo code for CPF approach

4 Experimental Results

4.1 Performance of CPF Approach

Let *n* be the number of URLs prefetched, *k* be the number of URLs requested from the prefetched URLs, and *m* be the number of URLs requested by the user. Two parameters are used to assess the performance of CPF prefetching scheme:

1. *Hits*: The number of URLs requested from the prefetched URLs
2. *Accuracy*: The ratio of *Hits* to the number of prefetched URLs

Prediction accuracy is computed as

$$\frac{\sum_{r=0}^{k} URL_r}{\sum_{j=1}^{n} prefetched\_URL_j} \tag{9}$$

To verify the accuracy, URLs for each host are prefetched and compared with the predicted URLs over the next *t* days where *t* is the prediction period. Table 1 presents the results obtained by executing the CPF pseudo code on NASA Web log files for 6 days (1/Aug/1995 to 6/Aug/1995).

Table 1. Results of CPF approach

Cluster Id	Users in Clusters	User Id	No. of Requests	Number of URLs Prefetched	Hits	Prediction Accuracy
C1	U1,U2,U3, U4, U5	1	162	36	35	97.22
		2	184		33	91.66
		3	132		34	94.44
		4	168		35	97.22
		5	190		34	94.44
C2	U6,U7,U8, U9	6	200	62	60	96.77
		7	135		56	90.32
		8	146		58	93.54
		9	202		61	98.38
C3	U10,U11, U12	10	181	28	27	96.42
		11	126		26	92.85
		12	0		-	100
C4	U13,U14, U15, U16	13	186	24	22	91.67
		14	54		20	83.3
		15	147		22	91.67
		16	85		21	87.5

Fig. 5 shows the Web traffic (the number of URLs requested by each host/users). It is observed from the Fig. 6 that, the prediction accuracy ranges from 83.33 to 98.38%. The average prediction accuracy is 93.16% excluding the deviated one**.** Experimental results show that, the proposed CPF approach has very high prediction accuracy compared to other approaches [1,2,3,4].

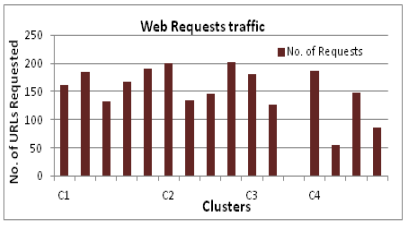

Fig. 5. Web Requests traffic

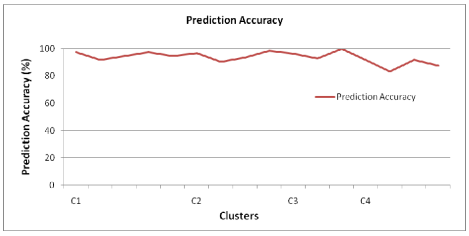

Fig. 6. Prediction Accuracy

4.2 Comparative Analysis

Comparative analysis has been made on the response time (latency) obtained from the analytical model and the trace collected from NASA Web site. The model parameters are derived from the trace data. To validate the model, the two metrics related to *user perceived latency* and *the traffic increase* are used. *Latency per page ratio* is the ratio of the latency that prefetching achieves to the latency with no prefetching. Lower the *latency ratio* value better the *performance*. *Traffic increase* denotes the bytes transferred through the network when prefetching is employed divided by the bytes transferred when prefetching is not employed. Lower the values of *traffic increase* better the *performance*. Cumulative Distribution Function (CDF) comparison of latency with and without prefetching for the trace and model with the given bandwidth is shown in Fig. 7. Average Latency Ratio v/s Traffic Increase is shown in Fig. 8. A summary of prefetching techniques is provided in Table 2.

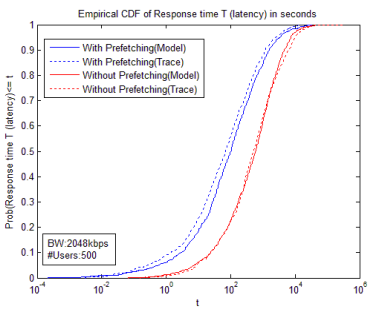

Fig. 7. CDF comparison of Response time

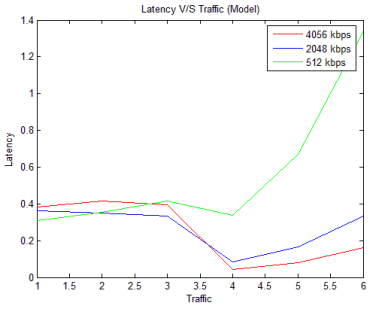

Fig. 8. Average Latency Ratio v/s Traffic Increase

Table 2. Summary of prefetching techniques

Reference	Method	Single User	Multiple Users	Prediction Accuracy
[1]	User profiles, weighted directed graph	Yes	-	50-75%
[2]	ANN, Keywords in Anchor Text	Yes	-	60-70%
[3]	PPM algorithm	Yes	-	40-73%
[4]	Top-10 prefetching approach	Yes	-	60%
[5]	Directed Graph	Yes	-	70-75%
[6]	Intelligent adaptive NN predictor	Yes		80%
Proposed method	***CPF method (thru clustering)***	***Yes***	***Yes***	***83-98%***

5 Conclusions

CPF approach that showed its usefulness in reasonable utilization of network resources through prefetching of Web pages for a community of users instead of a single user with an average prediction accuracy of 93.16% has been presented. Though the CPF approach results in substantial increase of network traffic, it effectively reduces the user perceived latency. Future research directions in this regard concern with the development of adaptive predictive systems that use hybrid approach such as use of statistical, neural, and Bayesian learning algorithms.

References

1. Loon, T.S., Bhargavan, V.: Alleviating the latency and bandwidth problem in WWW browsing. In: Proc. of the USENIX Symposium on Internet Technologies and Systems, USITS 1997 (1997)
2. Ibrahim, T., Xu, C.Z.: Neural Nets based Predictive prefetching to tolerate WWW latency. In: Proc. of the 20th International Conference on Distributed Computing Systems. IEEE, Taipei (2000)
3. Fan, L., Cao, P., Jacobson, Q.: Web prefetching between low-bandwidth clients and proxies: Potential and performance. In: Proc. of the Joint International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS 1999, Atlanta, GA (1999)
4. Markatos, E.P., Chronaki, C.E.: A Top-10 approach to prefetching on the Web. In: Proc. of the 8th Annual Conference of the Internet Society, INET 1998, Geneva, Switzerland (1998)
5. Padmanabhan, V.N., Mogul, J.C.: Using predictive prefetching to improve WWW latency. Proc. of ACM Computer Communication Review 26(3), 23–36 (1996)
6. Tian, W., Choi, B., Phoha, V.V.: An Adaptive Web cache access predictor using Neural Network. In: Proc. of the 15th International Conference on Industrial and Engineering Applications of AI and Expert Systems, Cairns, Australia, pp. 450–459 (2002)