ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 2, February 2015

A Novel Decision Tree Based Web Prefetching Scheme

C.Anuradha, Sundararajan.M, Arulselvi S

Assistant Professor, Dept. of CSE, Ramanujam Centre for Computational Intelligence, Bharath University, Chennai, Tamil Nadu, India

Director, Research Center for Computing and Communication, Bharath University, Chennai, Tamil Nadu, India

Co-Director, Research Center for Computing and Communication, Bharath University, Tamil Nadu, India

ABSTRACT: Web prefetching fetches objects and stores them in advance, hoping that the prefetched objects are likely to be accessed in the near future.It is used to effectively mitigate the user perceived latency when accessing the web pages. In this paper we propose a prefetching scheme that uses a decision tree approach to compute the probability values of anchor texts present in the web page so that effective predictions can be made on the user accesses. We discuss elaborately about the various algorithms that are currently used in web prefetching. Predictions can be effective when user has long browsing sessions. The proposed method can effectively prefetch objects that are used to minimize user access latencies and also to maximize hit rates.

KEYWORDS: anchor links, prefetching, prediction, tokens

I. INTRODUCTION

Many factors contribute to a less-than-speedy web experience, including heterogeneous network connectivity, real-world distances, and congestion due to unexpected network demand. Web caching has been proposed as a technology that helps reduce network usage and server loads and improve average latencies experienced by the user. When it is successful, prefetching web objects into local caches can be used to further reduce latencies and even to shift network loads from peak to non-peak periods. One example is a proxy server that intercepts the requests from the clients and serves the clients with the requested objects if it has the objects stored in it; if the proxy server does not have the requested objects, it then fetches those objects from the web server and caches them, and serves the clients from its cache. Another example is local caching that is implemented in web browsers.

Previous research on web prefetching focused on using the history of client access patterns to make predictions. The access patterns were represented using URL graph and based on graph traversal done by search algorithm the prefetching predictions are computed. These schemes suffer from the drawback of not able to prefetch web objects that are newly created or never visited before. To overcome these limitations, keyword based semantic prefetching approach [6, 16] was introduced. It could predict the web objects to satisfy the future requests based on semantic preferences of past retrieved web objects. Neural networks was trained using the keyword set to make predictions, and the research was motivated by the fact that the web user's surfing behavior was often guided by the keywords in anchor texts of URL that refer to a web object. The anchor link (URL) represents relationship between two web documents or two parts of the same web document. Anchor text provides relevant descriptive or contextual information to users about the contents related to a particular anchor link.

In this paper we present the semantic prefetching scheme for making the predictions by applying decision tree algorithm to compute the probability value of anchor texts present in a web page. The DT algorithm is a classification and regression algorithm provided by Microsoft SQL Server Analysis Services (SSAS) for use in predictive modeling of both discrete and continuous attributes. DTs are one of the most popular methods that are used for Data Mining purposes. In this paper, we focus on extracting anchor links from a displayed web page, convert the anchor text

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 2, February 2015

associated with each anchor link into set to tokens (keywords), apply decision tree algorithm over the tokens to compute probability, generate preference list based on the probability to perform prefetching of web objects. When the user access an anchor link, now the system tries to satisfy the user request using web objects stored in the prefetching cache. Performance of the proposed scheme is evaluated by observing the browsing patterns of users in different web sessions. The results show that the hit rate improves when a user is involved in longer browsing sessions and access the web links relevant to a particular topic of interest.

The paper is organized as follows: Section 2 presents the related work carried out in web prefetching domain. Section 3 provides overview of the scheme used to achieve semantic prefetching and explains the implementation of prefetching scheme. Section 4 deals with the experimental setup and presents the evaluation results. Section 5 concludes the paper with remarks on future work.

II. RELATED WORK

Prefetching has been applied to a variety of systems to hide communication latency . Crovella and Barford had analyzed the effect of prefetching on the network performance by considering network delay as the primary cost factor. A simple transport rate controlled mechanism was proposed to improve the network performance. The usage of anchor text to index URL's in Google search engine was suggested by Brin and Page. The research was focused on effective usage of additional information present in the hypertext. Chakrabarti et al designed and evaluated an and line spacing. Several techniques were designed to be used at client -side, server-side and hybrid client/server for enhancing the delivery of Web pages to the user. User's browsing behavior was analyzed to identify specific interest on a domain for supporting services like Web personalization and web prefetching. The effectiveness of using link-based or content-based ranking method in finding the Web sites was analyzed]and the results indicated that anchor texts were highly useful in site finding. A text analysis method was discussed that used text in and around the hypertext anchors of selected Web pages to determine the users interest in accessing the Web pages. A keyword-based semantic prefetching approach was proposed that applied neural networks to predict the future.

PPM models were commonly used in Web prefetching for predicting the user‟s next request by extracting useful knowledge from historical user requests. Factors such as page access frequency, prediction feedback, context length and conditional probability influence the performance of PPM models in prefetching. An online PPM model based on non compact suffix tree was implemented that used maximum entropy principle to improve the prefetching performance. A novel PPM model based on stochastic gradient descent was proposed that defined a target function to describe a node‟s prediction capability and then selected a node with maximum function value to predict the next most probable page.

Markov models were effectively used in Web prefetching by utilizing the information gathered from Web logs. Different techniques were presented for intelligently selecting the parts of different order Markov models to create a new model with reduced state complexity and improved prediction accuracy. Three schemes of pruning (support, confidence and error) were presented to prune the states of All-K th order markov model. A Markov–Knapsack approach was proposed that combined Multi-Markov Web-application centric prefetch model with a Knapsack Web object selector for enhancing the Web page rendering performance. An integration model was designed to combine clustering, association rules and Markov models to achieve better prediction accuracy with minimal state space complexity. Markov tree was used for effective page predictions and cache prefetching, which used the training data set to construct the tree structure for representing the Web page access patterns of users. Domain ontology provides useful semantic information to be used in next page prediction systems. Two methods were discussed to integrate the semantic information into Markov models for prediction. The methods allowed low order Markov models to make intelligent accurate predictions with less complexity than the higher order models. An approach for Web page prediction through linear regression was proposed that depended on the transition probability and ranking of links in the current Web page for the prediction accuracy.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 2, February 2015

III. SEMANTIC PREFETCHING SCHEME

The proposed scheme uses the scheme proposed by [1][2]. Our work is on the algorithm being used. In this paper, the decision tree algorithm is used for classification. This scheme is responsible for making efficient predictions of web objects to be prefetched for satisfying the user's future requests with low latency. It is based on the concept of client-side prefetching, where the client directly prefetches web documents from the server and stores it in local cache to service user requests. It significantly reduces the latency when servicing user requests, since there is no network latency for retrieving locally cached documents. The prefetching scheme consists of the following components: Tokenizer, Prediction unit, Prefetching unit and the Prefetching cache.

3.1 Tokenizer

When user is viewing a web page, Tokenizer parses the web page to extract anchor links (URL) and its associated anchor text. It then identifies the tokens (keywords) from each anchor text of a link. A token is considered as the meaningful word within anchor text of a URL. When a user clicks anchor link in a web page, then tokenizer moves the tokens of that particular anchor text into user token repository. The repository has collection of tokens with their frequencies, where token frequency indicates the number of times a particular token is seen in the anchor text of links selected by the user. When a token occurs for the first time, new entry is created in the repository with initial count value as 1. For the existing tokens its count value gets incremented. The user token repository is used by the prediction unit to compute probability values of anchor links that are not accessed by the user in a web page.

3.2 Prediction Unit

It is responsible for computing the probability value of each anchor link using decision tree . The advantages of using decision tree algorithm for computing the probability values are: a) simple mechanism to compute values for the specified data b)requires minimal storage, since it stores only token counts and c) incremental update whenever new data is processed. Anchor text associated with each link is taken and thetokens from it are compared with tokens stored in the user token repository to compute the probability value. The anchor links are then arranged based on the probability value to be given as input to the prefetching unit.

3.3 Prefetching Unit

The web objects that are required to satisfy the user requests with low latency are retrieved by the prefetching unit from the web server and stored in the local cache. The selection of web objects for prefetching is based on the preference list enerated as output by the prediction unit. Prefetching is normally performed when the client is viewing a web page (i.e. 'idle' time). Prefetch requests are given low priority than the regular user requests, so whenever user makes a request the prefetching unit suspends any ongoing prefetch activity. It is possible to prefetch only limited number of links at any time
because of small time period to perform prefetch before user makes a new selection.

3.4 Prefetching Cache

The web documents that are prefetched from the server are stored in the prefetching cache to satisfy the user's future requests. To eliminate the caching impact due to temporal locality exhibited in the user access patterns, prefetching cache is managed separately from the browser's in-built cache. When new web documents need to be stored in the prefetching cache, it selects documents that are not accessed for a long time and purge it whenever there is insufficient space in the cache.

3.5 Implementing Repositories

The repositories (user-accessed and predicted-unused) are implemented as a table with three fields: token, token-count and last-updated time. Token field is used to store the tokens
extracted from the anchor texts. Token-count field indicates the number of times each token is updated in the repository. Lastupdated time field indicates the time when the token was last added to the repository. Both the repositories are of fixed size and new tokens are added into it by eliminating the old tokens whenever the repositories reach their maximum limit. Based on the last-updated time field, the tokens are selected for elimination from the repositories. The repository

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 2, February 2015

size should be selected carefully in order to avoid the legitimate tokens from being eliminated and to prevent the trivial tokens from occupying the allotted space. We set the size of each repository to be 100.

USER TOKEN REPOSITORY

Token	Count
academics	10
admissions	4
applied	1
computer	3
courses	5
department	1
science	7
web	3
students	5
organization	2
test	7

3.6 Computing Prediction Values

The anchor text is represented as set of tokens arranged in a specific order.
Anchor text = {T1, T2, T3 . . . Tn}, n = number of tokens

The steps in prefetching scheme are explained as follows:
(1) User requests a web page by providing its URL in the web browser.
(2) The requested web page is retrieved and displayed to the user.
(3) The displayed web page is scanned to extract the list of anchor links and its associated anchor text for evaluation.
(4) The anchor text associated with each link is processed to generate the set of tokens, where 'token' corresponds to individual words in the anchor text.
(5) The tokens of an anchor text are added to the user token repository when the user access particular anchor link.
(6) For each token its count value is maintained in the user token repository.
(7) The probability of each anchor link is computed by applying decision tree on the anchor text (i.e. set of tokens) with reference to the tokens maintained in user token repository.
(8) Anchor links are sorted based on the computed probability value and placed in the preference list.
(9) Prefetching unit takes anchor links from the preference list and retrieves web objects from the web server, which are then placed in the prefetch cache.
(10) When the user clicks an anchor link in a web page, the system first verifies its availability in the prefetch cache.
(11) If it is available in the prefetch cache, then web page will be displayed.
(12) If web objects are not available in the prefetch cache, then it is retrieved from the web server and displayed to the user.

3.7 Hint List

Based on the computed prediction value, the hyperlinks are selected to create a hint list. The hint list is managed using a priority queue that arranges the hyperlinks according to its prediction value. Prefetching engine uses the hyperlinks stored in the hint list to prefetch the Web objects during browser"s idle time period. When user visits a new Web page, the contents of hint list will be cleared and then populated with new set of hyperlinks to perform prefetching. It helps to prevent prefetching of irrelevant hyperlinks during a browsing session as shown in Figure1.

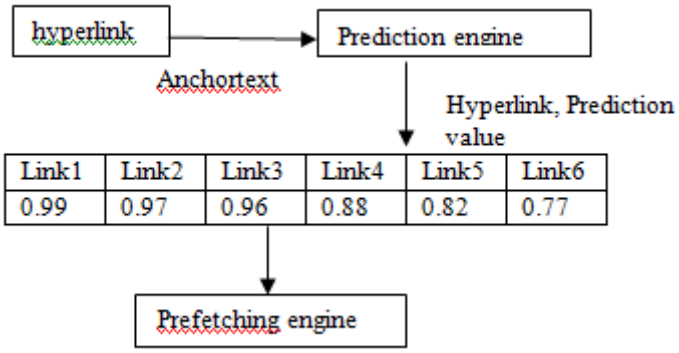
ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 2, February 2015

Figure 1



3.8 PREFETCHING ENGINE

It uses the hyperlinks listed in the hint list to prefetch the Web objects during browser idle time to avoid interference with regular user requests. When a user requested Web object is available in the prefetch cache, then it is served quickly with minimal retrieval time. The prefetching engine will not retrieve all the predicted Web objects from the server due to the following reasons: a) Lack of idle time due to faster navigation between the Web pages by users b) Few predicted objects already exists in the regular cache and c) Few predicted objects already demand requested by the users. The prediction engine needs to generate useful hint list to avoid wastage of user and server resources that may lead to performance degradation. Prefetch requests are given low priority than the regular user requests to allow the Web browser to utilize the entire available bandwidth for satisfying the user requests. Whenever user initiates any page loading activity in the browser, prefetching activity gets terminated and the remaining information in the hint list will be discarded.

IV. EVALUATION

Performance of the proposed prefetching scheme cannot be evaluated efficiently using trace based simulations. It is due to the fact that semantic technique tries to capture individual user interests in a single browsing session, but the web traces could not give a comprehensive view of client's interests. An efficient way to capture the required information would be to extract user interests at the client side. It is done by capturing the user interests using information obtained through user's browsing patterns. The user accesses web pages in two ways: a) pass URL request directly in the web browser and b) navigate across the web pages using anchor links embedded in a web page. In our semantic prefetching approach, we extracted user interests by exploiting the navigational behavior (web page access using anchor links) exhibited by the user. The performance of the proposed scheme is tested by performing user interest based browsing, where an individual user is assigned the task of obtaining information about aparticular topic of his interest. For the evaluation purpose we used an open source browser-cxbrowser developed in C#, using which the users will access the web pages. The proposed prefetching scheme is implemented as an add-on to the browser. It provides facility for the user to configure prefetching settings based on his requirements.The user can enable or disable prefetching during a browsing session. The user requests in a browsing session are maintained in a log file for performance analysis. The results are taken by monitoring the browsing pattern of two different users - User_A and User_B. User_A takes the role of a prospective student visiting several university websites for gathering information related to his topic of interest. User_B takes the role of a person visiting news portals for gathering information of his interest. The websites considered for evaluation possess identical semantic structure and content.

The anchor links to be prefetched are predicted by applying decision tree approach. The prefetch operation takes place only during user's idle time (i.e. when user not browsing), so depending on the user's access pattern the number of links prefetched will vary dynamically. In Figure 3, x-axis denotes the pages accessed by user in a browsing session and y-axis denotes the number of anchor links that are predicted and prefetched. When the user initially starts a browsing session, the user token repository will be empty and it cannot make any predictions of web objects. As the user starts browsing web pages by clicking anchor links, then user token repository will be filled with tokens based on which it

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 2, February 2015

starts making the predictions. If the user moves from current page to next page quickly, then less number of web objects is prefetched. In some cases the user spends long time in a particular web page, which increases the idle time causing more number of links to be prefetched. When a web page contains only small amount of information that are relevant to user interests, then minimum predictions are generated for that web page.

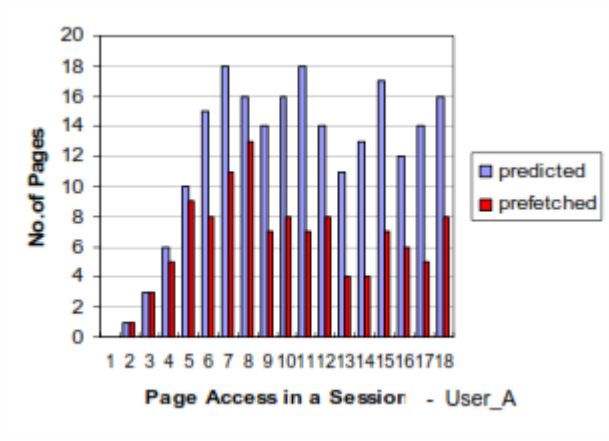
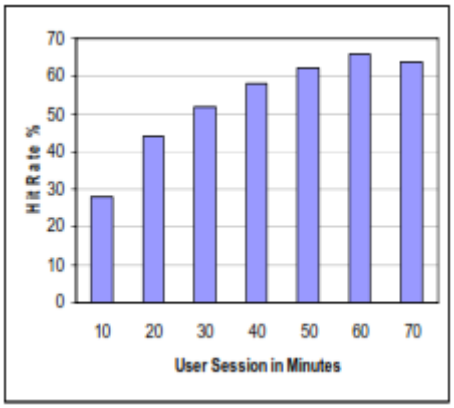


Figure 2



ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 2, February 2015

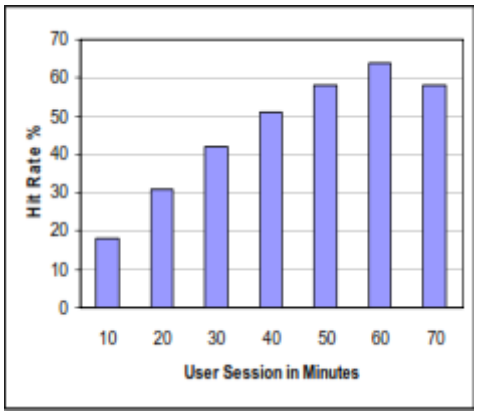


Fig 3

V. CONCLUSION

In this paper we have discussed Web prefetching scheme that
used decision tree approach to compute the prediction value of hyperlinks, which was used to decide the Web objects to be prefetched. Information available in the user-accessed and predicted-unused repositories was used to compute the prediction value of hyperlinks, which improved the prediction accuracy and minimized user perceived latency. It generated effective predictions during the browsing sessions, when user visited Web pages seeking information relevant to specific topic of interest. The proposed scheme was experimentally evaluated by observing the results over several user browsing sessions. Results indicate that the proposed scheme provides good hit rate and precision accuracy, when compared to other existing algorithms.

REFERENCES

[1] P.Venketesh, Dr.R.Venkatesan , L.Arunprakash, International Journal of Computer Science and Applications, Vol. 7, No. 1, pp. 66 – 78, 2010
[2] Kaliyamurthie K.P., Udayakumar R., Parameswari D., Mugunthan S.N., 'Highly secured online voting system over network", Indian Journal of Science and Technology, ISSN : 0974-6846, 6(S6) (2013) pp.4831-4836.
[3] H. Chris Tseng, "Internet Applications with Fuzzy Logic and Neural Networks: A Survey", Journal of Engineering, Computing and Architecture, Volume 1, Issue 2, 2007
[4] Kiran Kumar T.V.U., Karthik B., 'Improving network life time using static cluster routing for wireless sensor networks", Indian Journal of Science and Technology, ISSN : 0974-6846, 6(S5) (2013) pp.4642-4647.
[5] J.C.Mogul, "Method for predictive prefetching of information over a communications network", Patent No.5,802,292, 1998
[6] Muruganantham S., Srivastha P.K., Khanaa, "Object based middleware for grid computing", Journal of Computer Science, ISSN : 1552-6607, 6(3) (2010) pp.336-340.
[7] N. Craswell, D. Hawking, S.E. Robertson, "Effective
Site Finding Using Link Anchor Information", Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval, 2001
[8] Khanaa V., Thooyamani K.P., Saravanan T., "Simulation of an all optical full adder using optical switch", Indian Journal of Science and Technology, ISSN : 0974-6846, 6(S6)(2013) pp.4733-4736.
[9] B.D.Davison, "Predicting web actions from HTML content", Proceedings of 13th ACM Conference on Hypertext and Hypermedia, 2002
[10] Shirley Gloria D.K., Immanuel B., Rangarajan K., "Parallel context-free string-token petri nets", International Journal of Pure and Applied Mathematics, ISSN : 1311-8080, 59(3) (2010) pp.275-289.
[11] T.I.Ibrahim, C.Xu, "Neural net based predictive pre-fetching to tolerate WWW latency", Proceedings of the 20 Th International Conference on Distributed Computing Systems, 2000
[12] R.Cooley, P.Tan, J.Srivastava, "Discovery of interesting usage patterns from Web data", Lecture Notes in Artificial Intelligence, 1836, pp.163-182, Springer, Berlin, 2000
[13] N. Craswell, D. Hawking, S.E. Robertson, "Effective Site Finding Using Link Anchor Information", Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval, 2001
[14] B.D.Davison, "Predicting web actions from HTML content", Proceedings of 13 ACM Conference on Hypertext and Hypermedia, 2002
[15] Z.Chen, L.Tao, J.Wang, L.Wenyin, W.Ma, "A unified framework for web link analysis"; Proceedings of 3rd International Conference on Web Information Systems Engineering, pp. 63-72, Singapore , 2002
[16] C.Ding, X.He, P.Husbands, H.Zha, H.Simon, "PageRank, HITS and a Unified Framework for Link Analysis"; Technical Report, 49372, Lawrence Berkeley National Laboratory, 2002

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 2, February 2015

[17] Nadav Eiron and Kevin S. McCurley, "Analysis of Anchor Text for Web Search",Proceedings of SIGIR, 2003
[18] J.Arul Hency Sheela, GC-MS Studies of the Plant Clematis Gouriana , International Journal of Innovative Research in Science, Engineering and Technology , ISSN: 2319-8753 ,pp 13514-13519 ,Vol. 3, Issue 6, June 2014.
[19] Jemima Daniel,Usage of Language,International Journal of Innovative Research in Science, Engineering and Technology,ISSN: 2319-8753, pp 7073-7075,Vol. 2, Issue 12, December 2013.
[20] Jemima Daniel,The Duality of Human Nature,International Journal of Innovative Research in Science, Engineering and Technology,ISSN: 2319-8753,pp 511-512,Vol. 2, Issue 2, February 2013.
[21] Jemima Daniel,Impact of E-Mail communication,International Journal of Innovative Research in Science, Engineering and Technology,ISSN: 2319-8753,pp 527-528,Vol. 2, Issue 2, February 2013.
[22] Jemima Daniel,The Enchanting World In Karnas`S Plays,International Journal of Innovative Research in Science, Engineering and Technology,ISSN: 2319-8753,pp 757-758,Vol. 2, Issue 3, March 2013.
[23] Jemima Daniel,Myth in Indian English Dramas,International Journal of Innovative Research in Science, Engineering and Technology,ISSN: 2319-8753,pp 1551-1555,Vol. 2, Issue 5, May 2013.