ABSTRACT

The exponential growth in the usage of World Wide Web (WWW) for data dissemination has led to enormous web traffic and hence, web users experience considerable access delays. To reduce the access delay, it is better to keep the frequently used information by various web users in the cache memory because this helps the users to retrieve the information immediately. Apart from experiencing less access delay, caching process also helps in better bandwidth utilization, reduction of network congestion, and load reduction on the origin server. As web proxy cache size is limited, cache replacement methods are required to handle the cache content. The performance of web caching methods is analyzed using the widely used metrics, Hit Ratio (HR) and Byte Hit Ratio (BHR). HR is the percentage of the number of requests that is entirely served by the cache over the total number of requests. BHR is the percentage of the number of bytes that corresponds to the requests served by the cache over the total number of bytes requested. A high HR indicates the availability of the requested object in the cache most of the time and high BHR indicates reduced user-perceived latency and savings in bandwidth.

Web pre-fetching is the process of fetching the web objects from the origin server which has more likelihood of being used in future. The fetched contents are stored in the cache. Pre-fetching helps in increasing the cache hits and reducing the user-perceived latency. Combining pre-fetching and caching techniques result in experiencing less access delay and better bandwidth utilization than using those two techniques individually.

Web caching and web pre-fetching complement each other because web caching exploits temporal locality for predicting the repeated user's access to the same web objects and web pre-fetching exploits spatial locality in which the user’s access to certain objects entail access to certain other related objects. In caching the pre-fetched web pages, efficient cache replacement techniques have to be deployed to manage the cache content effectively. The traditional cache replacement techniques used often fail to increase the cache hit ratio substantially. Many works have been reported in the literature for web caching and pre-fetching of web pages. In this research, web caching and web-pre-fetching techniques have been combined together for higher efficiency. For pre-fetching of web objects, clustering technique is used. In clustering, both inter and intra clustering of web objects are considered. Inter-clustering deals with the grouping of the home pages of the various web sites browsed by the web users in the chosen time interval. Intra-clustering deals with the grouping of all the browsed pages of every web site, for various users in the chosen time interval. This process is repeated for all the web sites browsed by the users. For web caching, five different techniques, namely: Least-Recently-Used (LRU), Least-Frequently-Used (LFU), Support Vector Machine (SVM)-LRU, Bayesian network-LRU, and Neuro-fuzzy-LRU are used.

Web objects accessed by various users are identified from their log file. Web log file contents are pre-processed and classified as Class 0 or Class 1 based on the features, namely: recency, frequency, retrieval time, and size of web objects. The above said features are incorporated into the SVM classifier and it is trained. For each user, the web navigation graph (WNG) is constructed using a session time interval of thirty minutes. Each node in the WNG represents a website URL requested by the user and the edges in the graph indicate the number of transitions made by the user between the website URLs. An inter-site clustering algorithm gets the contents of WNGs as input and two parameters namely ‘support’ and ‘confidence’ are used to keep track of the frequently visited objects by the user. By fixing a threshold

value for these parameters, edges which have values less than the threshold will be removed. Then, Breadth First Search (BFS) algorithm is applied to identify the clusters for each client.

Cache memory is divided into short-term cache and long-term cache. Of the total cache size, two-thirds of the cache space is allotted to the short-term cache and one-third space is allotted to the long-term cache. When the user requested web object is neither found in the short-term cache nor in the long-term cache, then, the requested object is fetched from the origin server and it is transmitted to the user as well as placed into the short term cache. If the requested object is found in any one of that user's cluster(s), all the other web objects present in that cluster are loaded into the short-term cache by fetching them from the origin server during the browser idle time. On the other hand, when the user requested web object is found in the short-term cache (cache hit), it is given to the user and if that object is found in any of that user's cluster(s), the remaining web objects of that cluster that are not available in the short-term cache are pre-fetched from the origin server and are cached into the short-term cache during the browser idle time.

The access count of the requested web object found in the short-term cache is incremented by one. If this access count becomes greater than the threshold value chosen, then it is classified using the SVM classifier and it is moved either to the top or bottom of the long-term cache based on being classified as Class 1 or Class 0 respectively. If the requested web object is not available in the short-term cache and if found in the long-term cache, all the web objects including the requested one are re-classified by the SVM and are either moved to the top or bottom of the long-term cache. The requested web object is sent to the user and if it is found in any of that user's cluster(s), pre-fetching of other web objects (if any) found in that cluster are done. They are stored into the short-term cache. Least Recently Used (LRU) technique is used for the removal of web objects from the short-term cache if sufficient space is not available for caching a new web object.

The same process is repeated by using Bayesian-LRU and Neuro-fuzzy-LRU techniques separately for the classification of web objects found in the long-term cache and their performance in terms of HR and BHR are compared. It has been demonstrated that for the chosen sample dataset which contains the browsing pattern of many users and if LRU caching with pre-fetching through clustering (inter-site) is used, the average HR is 67% and the HR increases to 84% and 86%, if SVM-LFU and SVM-LRU caching with pre-fetching are used respectively. Considering BHR achieved, it is found to be 34%, 64% and 66% respectively for the above methods. For LFU caching with pre-fetching, the HR and BHR achieved is 47% and 26% respectively. When LRU, SVM-LRU, Bayesian-LRU and Neuro-fuzzy-LRU techniques are used separately for caching combined with clustering (inter-site and intra-site) technique for pre-fetching, HR is found to be 68.25%, 87.25%, 85.25% and 84.75% respectively. BHR achieved for the above methods is 36%, 68%, 66% and 65.5% respectively. Hence, SVM-LRU technique for caching combined with clustering technique for web pre-fetching results in better performance for the chosen dataset.