Creating an Efficient Prefetching Mechanism by Leveraging Rule Based Agents

Jyoti[1], A. K. Sharma[2], Amit Goel[3]

[1]Sr. Lecturer, Dept of CE, YMCA Univ. of Science and Tech., Haryana, INDIA
[2]Prof. & Head, Dept of CE, YMCA Univ. of Science and Tech., Haryana, INDIA
[3]Manager, Evalueserve, Gurgaon, Haryana, INDIA
justjyoti.verma@gmail.com, ashokkale2@rediffmail.com, goelamit1@yahoo.com

Abstract: Prefetching and caching are two well-known approaches for improving the performance of the Web and have become essential components of the Web infrastructure. But without their careful usage, both can result in the depletion of the performance which they could render complementing each other's drawbacks. In recent years, agents have become a very popular paradigm in computing because of their flexibility, modularity and general applicability to a wide range of problems. This paper provides a novel approach wherein agents have been introduced between client machines and proxy server to help the clients in getting the prefetched documents of their interests thereby balancing both the caching and prefetching.

Keywords: Prefetching, agents, web mining, proxy server

1. Introduction

The Web has evolved rapidly from a simple information-sharing mechanism offering only static text and images to a rich assortment of dynamic and interactive services, such as video/audio conferencing, e-commerce, and distance learning. The explosive growth of the Web has imposed a heavy demand on networking resources and Web servers. Users often experience long and unpredictable delays when retrieving Web pages from remote sites [1]. Hence, an obvious solution in order to improve the quality of Web services would be the increase of bandwidth, but such a choice involves increasing economic cost. However, the higher bandwidth would solve temporarily the problems since it would ease the users to create more and more resource-hungry applications, bunching again the network. Therefore, the network limitations will remain or worsen unless effective software solutions are also provided. The authors of [2] have proposed a methodology to incorporate a Predictive Prefetching Engine (PPE) at the proxy level that helps in creating a database of rules that are extracted by applying the various data mining techniques at diverse levels on the proxy log. The current paper extends the work of [2] by introducing agents between clients and the proxy servers that will help in the triggering of rules for prefetching the web documents according to the client's requirements. The organization of the paper is as follows. The next three subsections give a brief outline of the web caching, web prefetching and the agents in general. Section II outlays the various factors that acted as motivation for this work. Section III discusses the proposed work followed by the conclusion and references.

1.1. The Web Caching Approach

Caching proved itself as an important technique to optimize the way the Web is used [3]. In particular, Web caching is implemented by proxy server applications developed to support many users. Proxy applications act as an intermediate between Web users and servers. Users make their connection to proxy applications running on their hosts. The proxy connects the server and relays data between the user and the server. At each request, the proxy server is contacted first to find whether it has a valid copy of the requested object. If the proxy has the requested object this is considered as a cache hit or otherwise a cache miss occurs and the proxy must forward the request on behalf of the user. Upon receiving a new object, the proxy services a copy to the end-user and keeps another copy to its local storage. From the above discussion follows that Web caching reduces bandwidth consumption, network congestion, and network traffic because it stores the frequently requested content closer to users. Also, because it delivers cached objects from proxy servers, it reduces external latency (the time it takes to transfer objects from the origin server to proxy servers). Finally, caching improves reliability because users can obtain a cached copy even if the remote server is unavailable. As far as concerned the performance of a Web proxy caching scheme, it is mainly dependent on the cache replacement algorithm [4] (identify the objects to be replaced in a cache upon a request arrival) which has been enhanced by the underlying proxy server. However, cache hit rates have not improved much with these schemes. Particularly, a Web caching scheme has three significant drawbacks:

- If the proxy is not properly updated, a user might receive stale data,
- as the number of users grows, origin servers typically become bottlenecks.

- Finally, several factors diminish the ideal effectiveness of Web caching. The obvious factors are the limited system resources of cache servers (i.e., memory space, disk storage, I/O bandwidth, processing power, and networking resources).

However, even if the cache space is unlimited, there are significant problems that cannot be avoided by such an approach. Specifically, large caches are not a solution because, the problem of updating such a huge collection of Web objects is unmanageable.

Therefore, we must resort to an approach, which will predict the future users' requests and retain in cache the most valuable objects.

1.2. The Web Prefetching Approach

Prefetching attempts to overcome these limitations by pro-actively fetching content before users actually request it [5]. Web prefetching is the process of deducing user's future requests for Web objects by locating popular requested objects into the cache prior to an explicit request for them. Unlike Web data caching, which exploits the temporal locality, the Web prefetching schemes are based on the spatial locality of Web objects. In particular, the temporal locality refers to repeated users' accesses to the same object within short time periods, whereas, the spatial one refers to users' requests where accesses to some objects frequently entail accesses to certain other objects. Typically, the main benefits of employing prefetching are that it prevents bandwidth underutilization and reduces the latency. Therefore, bottlenecks and traffic jams on the Web are bypassed and objects are transferred faster. Thus, the proxies may effectively serve more users' requests, reducing the workload from the origin servers. Consequently, the origin servers are protected from the flash crowd events as a significant part of the Web traffic is dispersed over the proxy servers. On the other hand, the main drawback of systems which have enhanced prefetching policies is that some prefetched objects may not be eventually requested by the users. In such a case, the prefetching scheme increases the network traffic as well as the Web servers' load. In order to overcome this limitation, high accuracy prediction models have been used [6]. From the above it occurs that caching and prefetching complement each other in order to reduce the noticeable response time perceived by users [7].

1.3. The Agents

The details of mobile agents and their employment in a distributed environment can be found elsewhere [8, 9]. Here, a brief introduction of mobile agents and their role in a network-based information system has been discussed. The term mobile agent is often context-dependent and has two separate and distinct concepts: *mobility* and *agency*. The term *agency* implies having the same characteristics as that of an agent. These are self contained and identifiable computer programs that can move within the network from node to node and act on behalf of a user or other entity. These can halt execution from a host without human interruption ([10]). The current network environment is based on the traditional client/server paradigm as shown in Fig.1.

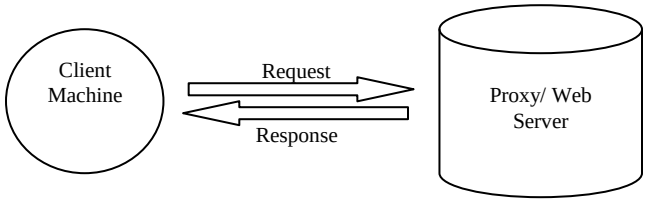

Fig. 1 Client/ Server Communication

However, in the case of mobile agents employed in a network, the service provision/utilization can be distributed in nature and is dynamically configured according to changing network performance metrics like congestion and user demand for service provision ([11]).

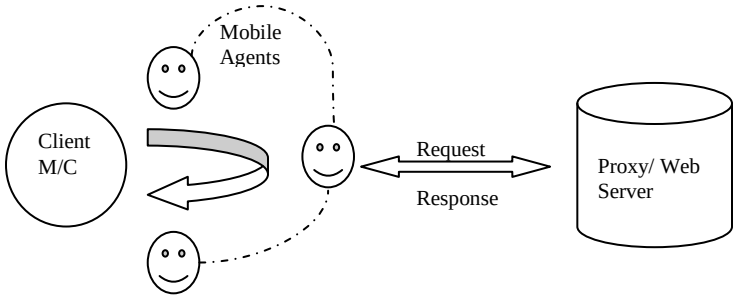

Fig. 2 Mobile Agent Communication

Mobile agents are typically suited to applications requiring structuring and coordinating wide area network and distributed services involving a large number of remote real time interactions. They can decide how best to handle a user's request based on past data or history of a similar request. These programs are therefore capable of learning from user behavior to some extent. Fig. 2 shows how agents can act as the intermediary between the clients and the proxy to meet the client's needs.

2. Motivation

Despite of the many efforts done by the industry and the research community to improve the World Wide Web performance, the web latency perceived by the users is still a perennial issue to reduce. The rising up of the Web architecture techniques such as web caching, prefetching and replication have became an important solution to reduce the user perceived latency. These techniques make use of the temporal, spatial and geographical locality properties of web objects to improve the web performance [12]. In the open

literature, there are several proposals about the benefits of the above techniques applied at different elements of the generic Web architecture (i.e. clients, proxies, servers). In [13] authors suggests that the use of caching can reduce up to 26% the latency; also the use of prefetching can improve the web performance up to 57%. Furthermore, the combined use of caching and prefetching can reduce the latency perceived up to 60%. Nevertheless, authors in [14] take into account the current Web generation; point out a theoretical upper bound of 97% of latency reduction when prediction is done in a collaborative manner between proxies and servers. In [15], the authors present a non-interfering web prefetch system between clients and servers, unfortunately there is not a caching module in its architecture. An attempt to integrate web caching and prefetching was done by [16] in an interesting proposal where both techniques were applied only at the proxy server side and the used workload was generated by a synthetic workload generator. The authors in [17] present an extended study about a prefetching technique and its impact on the Proxy Cache Server in a real WAN environment (i.e. university campus). The later proposal contributes with many useful considerations (e.g. log analysis, session estimation, web object types) to take into account when prefetching is applied. In [2], authors have proposed a framework for extracting relevant web pages from WWW using data mining. They proposed a Predictive Prefetching Engine (PPE) that sits on the proxy server. Since a proxy server lies between a web browser and a web server, it is a potential tool that can be suitably employed to reduce the www latency i.e. it can intercepts all requests to the web server to see if it can fulfill the requests by itself. If not, then only it may forward the request to the web server. The job of PPE is to preprocess the proxy web log to perform the preprocessing tasks like reduction of search space, user and session identification and path completion. After preprocessing a cleaner version of log is formed called data mart. The next step applies data mining operations like rough set clustering so as to narrow down the look up into the log and thus reducing the complexity of the overall process. Following this is the rule generation phase which extracts the rules for prediction. The data mining operations like rough set clustering, markov and association rules if applied alone do not provide accuracy. Therefore, authors have carefully integrated the three operations to improve the prediction accuracy thereby making the repository of the rules that will help in prefetching of the desired documents [18]. The proposed work extends the effort of [2, 18] by introducing agents between clients and the proxy servers that will help in the triggering of rules for prefetching the web documents according to the client's requirements.

3. Proposed architecture

The proxy is chosen as the deployment location for PPE since the active involvement of the clients is not desirable as that would require the clients to involve actively which clients tend to avoid and the focus is to make the structure as transparent as possible. Our approach requires input from several clients and thus choosing a single client as a point of deployment would not have been beneficial. The web servers themselves are involved in several other tasks that require a lot of memory and processing time and including one more process would have affected the throughput of the web servers. The outcome of the PPE is the repository of rules of the form $D_i \Rightarrow D_j$ *or* $D_i \Rightarrow D_j \Rightarrow D_k$

These rules will be of no use if they are not fired according to the client's demands. To effectively fire the desired rules, a layer of agents has been introduced between the client machines and the Proxy's PPE. The proposed framework is shown in fig.3. For every client machine, there will be a dedicated intelligent agent providing its services. The agents will work in the following manner:

3.1. Prefetching Scheme

Given that we have a set of rules in the repository created and maintained by the PPE, the prefetching scheme works as follows:

1. Let the request be for document A.
2. The agent scans the rule database[1] for the rules of the form A→X for some document X.
3. The agent then scans the database for every rule or part of the rule which has X in its sequence (e.g. A→ Y→X →Z). The only exception to this scan would be in the case of X being the last document in the sequence.

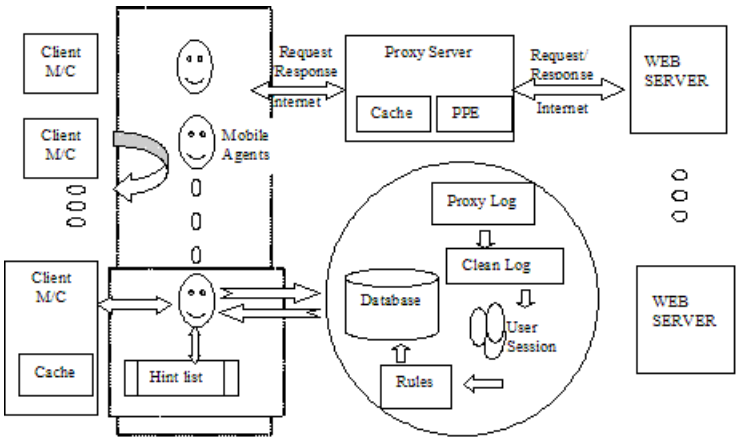

Fig.3 Proposed Architecture for prefetching the documents from the rule database of Proxy's PPE to the client's cache

4. As it scans, the agent brings all the documents that succeed X to the hint list maintained by the agent itself

[1]The rule database can be organized using some indexing scheme

and accordingly prefetch the corresponding web page from the proxy server to the client's cache.

5. The agent continues the scan and populates the hint list till such time the user requests for a web page which doesn't appear in the sequence. In such case, the agent cleans up the hint list and starts afresh (step 2).

documents from the proxy server to the client's cache. For performing the above said task, agents have been introduced between the client machines and the proxy server. The Proxy server holds the PPE whose task is to make the rule database by applying the various data mining techniques on the proxy's web log that marks every entry which exists between the client and the web servers. The agents thus effectively trigger the desired rules according to the interests of the users. There is a dedicated agent rendering its services for each user so that at any instant when the user's line of interest changes, the agents change there line of action.

References

[1] Pallis G, Vakali A., "Insight and perspectives for content delivery networks". Commun ACM (CACM) 2006; 49(1):101–6.

[2] Jyoti, A.K. Sharma, Amit Goel, "A Framework for Extracting Relevant Web Pages from WWW using web mining", In Proc of International journal of Computer Society and Network security, Seoul, Korea

[3] Rabinovich M, Spatsheck O., "Web caching and replication." Addison Wesley; 2002

[4] Podlipnig S, Boszormenyi L., "A survey of Web cache replacement strategies". ACM Comput Surveys 2003;35(4):374–98.

[5] Jiang Y, Wu M, Shu W., "Web prefetching: costs, benefits and performance". In: Proceedings of the 7th international workshop on web content caching and distribution (WCW2002). Boulder, Colorado; 2002.

[6] Yang Q, Zhang H., "Integrating Web prefetching and caching using prediction models." World Wide Web 2001;4(4):299–321.

[7] Kroeger TM, Long DDE, Mogul JM, "Exploring the bounds of web latency reduction from caching and prefetching." In Proceedings of the USENIX symposium on Internet technologies and systems. Monterey, California, USA; 1997

[8] Karmouch A. and V. A. Pham (1998), "Mobile Software agents: An Overview", IEEE Communications Magazine, 36(7), 26-37.

Conclusion

The paper discusses the new approach to prefetch the

[9] M. K. Perdikeas et al (1999), "Mobile Agents Standards and Available Platforms", Comp.Net, 31(19), 1999-2016.

[10] J. Cao, G. H. Chan, W. Jia, and T. Dillon (2001), "Check-pointing and Rollback of Wide-Area Distributed Applications using Mobile Agents", Proceedings, International Parallel and Distributed Processing Symposium, San Francisco: IEEE Computer Society Press.

[11] Ahmad, H. F. and Helene Arfaoui, Mori, K (2001), "Autonomous Information Fading by Mobile Agents for Improving User's Access Time and Fault Tolerance", Proceedings 5th International Symposium on Autonomous Decentralized Systems, 279-283.

[12] M. Rabinovich and O. Spatscheck, "Web Caching and Replication". Addison Wesley, 2002.

[13] T. M. Kroeger, D. D. Long, and J. C. Mogul, "Exploring the bounds of web latency reduction from caching and prefetching," in Procc. of the 1st USENIX Symp. on Internet Technologies and Systems, Monterey, USA, 1997.

[14] J. Domenech, J. Sahuquillo, J. A. Gil, and A. Pont, "The impact of the web prefetching architecture on the limits of reducing user's perceived latency," in Procc. of the 2006 IEEE/WIC/ACM Inter. Conf. on Web
Intelligence. IEEE, 2006.

[15] R. Kokku, P. Yalagandula, A. Venkataramani, and M. Dahlin, "NPS: A non-interfering deployable web prefetching system," in Procc. of the USENIX Symp. on Internet Technologies and Systems, Palo Alto, USA, 2003.

[16] W.-G. Teng, C.-Y. Chang, and M.-S. Chen, "Integrating web caching and web prefetching in client-side proxies," IEEE Transactions on Parallel and Distributed Systems, vol. 16, no. 5, pp. 444–455, 2005.

[17] C. Bouras, A. Konidaris, and D. Kostoulas, "Predictive prefetching on the web and its potential impact in the wide area." World Wide Web, vol. 7, no. 2, pp. 143–179, 2004.

[18] Jyoti, A.K. Sharma, Amit Goel, "A Novel Approach to Determine the Rules for Web Page Prediction using Dynamically Chosen K-Order Markov Models", In Proc of IEEE sponsored International Conference on Advances and Emerging trends in Computing Technologies, 2010