Self-organising Mobile Agent Based Proxies for Web Digital Libraries

R.Ponnusamy,
Research scholar,
Dept. of Computer Science & Engg,
Anna University,
Chennai - 600 025, India.
&
Sr.Lecturer,
Dept.of Computer Science & Engg.
Anjalai Ammal Mahalingam Engg.College,
Kovilvenni-614 403, India.

Dr.T.V.Gopal,
Asst.Prof.,
Dept. of Computer Science & Engg,
Anna University,
Chennai - 600 025, India.

Abstract

This paper presents a new application proxy design approach for handling huge set of dynamic requests from web digital libraries. In web digital libraries the requests from the various regions are unpredictable. The configuration of proxy is usually based on the static analysis of the number of requests from various users over a period of time. It needs a manual installation and configuration of proxies in order to provide various services of the above type. The designers of the system usually deploy the cache proxy based on the policy and by considering as well various characteristics of good service. But, normally the requests are unpredictable, that is one can't assure which objects will be requested by users at any time. In order to solve this problem a new self-organising mobile agent based application proxy approach has been proposed. In this approach a mobile agent is able to take a survey of user access pattern in a particular region. Based on the user access frequency, it automatically deploys the proxy. Its life is extended based on the number of requests over the specific period of time. The agent is doing the content semantics and user access pattern learning. The proxy cache replacement is based on these attributes. It is observed that increasing the level of human intelligence has contributed to the performance level.

Key words: Web digital Library, Mobile Agent, Proxy, Self-organising, Installation.

1. Introduction

Digital Libraries are modern social virtual institutions for information collection, preservation and dissemination of knowledge. The design and development of digital library requires addressing many issues [1-6] and must be able to perform intelligent human-oriented tasks in order to make the approach highly sophisticated and effective one. For implementation, the content storage management is the one among the important issues, which decides fast and economical retrieval of information in the federated or web digital libraries [3,6]. Caching proxies [7-9] become a critical component to handle web traffic and reduce both network traffic and client latency. These caching proxies effectively migrate copies of popular contents/objects from servers to proxy servers. Web client users see shorter delays when requesting the contents. Network managers see less traffic which reduces the server request rates. Normally, these caching proxies adopt various polices [10], such as LFU (Least Frequently Used). LFU with Dynamic aging, GDSF (Greedy Dual Size Frequency), LRU and etc. Perhaps these policies yield 30-50% hit rate.

On account of the application growth, the main problem posed by web enabled proxies needs dynamic scalability. The unparalleled growth of web digital library resources as well as the user accesses drive one to develop hierarchical proxy-cache scheme [11], which guarantees to yield 50% hit rate and reduces server load dramatically. Harvest cache has shown these features experimentally. Also, it is observed that the replacement of small size contents has done extremely well in all the methods.

Application proxies have been developed to do many application oriented tasks[12]. All these proxies are designed for general purposes(i.e., they are working well at the transport and network layer level). These proxies are considering neither the user access pattern nor consider as well the content semantics. They are installed statically and have no dynamic policy and so one can't expect high degree of performance over a period of time. In considering all these issues, a self-organizing mobile agent based proxy has been developed. It is an application proxy specifically designed for web digital libraries. It considers the content semantics and user access patterns. Also, it is able to clone itself in the appropriate place based on the frequency of access. In addition to this, the mobile agent dynamically is able to play multiple roles [13] according to the needed requirements.

In this paper Section 2 discusses the user access pattern and content semantic learning. Section 3 explains the hierarchical proxy organization. Section 4 explains complete self-organizing mobile agent proxy design and functionality. Section 5 gives implementation details and Section 6 provides the conclusion of the paper.

2. User Access Pattern and Content Semantic Learning.

Different user communities use web digital libraries. Also, there are libraries designed for specific purposes. For example IEEE/ACM Digital libraries are designed for research/academic purpose. The user access frequency is very high in the case of these types of institutions for the aforesaid libraries. The user access pattern and content semantics can be learnt easily and the content caching can be done according to this approach rather than the simple policies. This approach will increase the performance of the caching process. All the content semantics are gathered from the user while uploading the contents into the server. As in the Deway decimal [14] classification, here a hierarchical enumeration is done for the restricted set of topics and its details are maintained in the hierarchical table. Every content object is classified as per this hierarchical order.

The user interest and personal details are gathered from the user through the user screen interaction. User frequency of access, purpose and other information are also gathered and a user pattern is learnt based on all these attributes. Also, the agent identifies the common user's working in similar area, and in the temporary user requests. For example a post graduate Computer Science student is accessing Distributed System Design book 40 times per week because it is prescribed in his current curriculum syllabi. There are similar other 20 students who are members of this library. It can be cached in the proxy for a long time. In some other situation a Scientist is accessing a journal paper. It is a specific area in which he/she alone is working, and will access it only for a certain period. The content need not be maintained for longer period of time. This type of user pattern identification is done in the system.

3. Hierarchical- Network Proxies

In this work initially the servers are located at the root, and all the contents are served from this root. The network adopts the hierarchical tree architecture. After receiving requests from its child, the mobile agent is doing the frequency analysis over a period of time, that is, if the number of requests increase to a certain limit, then immediately it installs a proxy agent automatically using the agent cloning [15] technique. Likewise if the number of requests decreases, automatically the mobile agent withdraws its proxy in that head end/intermediate node and the user access patterns are moved to its parents. This behavior is explained as self organization [16] that is a functional structure which appears and maintains itself spontaneously. In this present design, the system is designed to accomplish the tasks without many user interactions. This is shown in the following figure. 1.

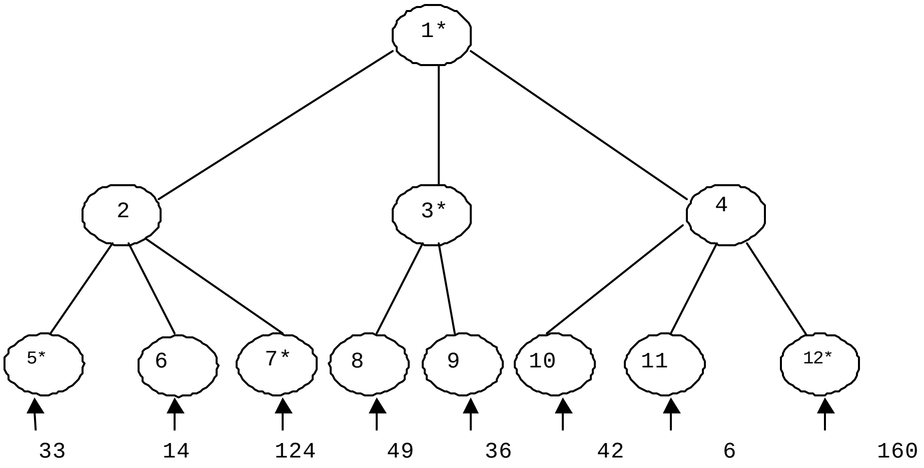

Figure 1 – Hierarchical Network Proxy Location Model

In the above figure '*' indicates the existence of the proxies in those nodes. In the above example if the number of requests exceeds an average of 50, then it will automatically install the proxy in the particular node. This has an approximation that it is able to forward maximum of 50 requests from the particular node. The number in the head-end indicates the number of requests for various objects.

4. Self-organizing Mobile Agent Proxy Design

In this design, an intelligent agent is designed to play various roles. The intelligent agent emits specialized behaviour for handling different tasks. This behavior can be modelled using roles [13]. Since, the agent takes different roles, the object is evolving and cannot be confined to being an instance of a single class. It is perceived as role hierarchy. The current design in which it is performing tasks such as 1) Agent cloning, 2) Cache proxing, 3) Content semantics identification, 4) Users access pattern learning. User pattern learning and content semantics identification have been already explained in the previous section 2. The remaining behaviours are explained here.

1. Agent cloning: It is a process of copying or inheriting an agent and resources for creation of new environment. It not only copies the objects but also starts doing intelligent processing, with the possibility of withdrawing from the system.

2. It does an inherent duplication/replication in the local cache area. This proxy divides the cache area as local cache, global cache and temporary cache. Local cache size is limited to a certain limit and will be computed based on the average number of clients in the system. The very frequently accessed contents are stored in the global area. For the benefit of new users the temporary cache will be allocated. The user caching areas over long period of time will replace these temporary user areas, if it stays in the system for a long period of time.

Normally, the HTTP interface provides the synchronization between the HTTP server and the mobile agent proxy. This server is designed to provide all the resources necessary to clone the mobile agent proxy. All the user requests are forwarded to the agent proxy/server who in turn will provide the necessary contents to the clients.

5. Implementation Details

This system is simulated using the Java Servelets. [17,18]. It is able to pass all the user requests to the IIS servers. Also, two applets are developed to get the user request and the content uploading. These applets are passing the requests to the standard IIS server/agent proxies. A mobile agent residing in the web server will automatically perform the aforesaid roles. A caching algorithm implemented in the system considers all the user access patterns and accordingly it is doing the proxy caching.

Current design for the proxy process listens on a particular port. When the server receives HTTP request, it parses the URL and finds file availability. If it is available then it will supply it from the local cache otherwise it will be forward the request to the server. In order to elicit the effective functionality, an additional system is designed to accept the RMI messages. This smart proxy supports the agent cloning also. The simulation technique provides the comparative performance of the hit rate between Squid and Mobile agent proxy.

6. Conclusion

In this paper, we have designed and developed a self-organizing mobile agent based application proxy for web digital libraries. It is well known that the web digital libraries are developed only for specific purposes and so it restricts the user community as well as the collection. It helps the system to learn user pattern easily. This work establishes that the application domain learning has much impact in proxy caches and comparing the hit ratios of squid proxy with our agent proxy improves the performance of the later. Though the performance of our design is poor in initial stages later on it improves gradually. In our design considers only a few attributes for user pattern learning. Future research may focus on corporation of more number of attributes.

7. References

1. Daniel Andreson, Tao Yano, Omer Egecioglu, Oscar H.Ibarra, and Terence R.Smith,Scalability Issues for High Performance Digital Libraries on the World Wide Web, Department of Computer Science, University of California, Santa Barbara, CA 93106.

2. Dieter W.Fellner, TU Braunschweig, Reserch Issue for Digital Libraries, http://graphics.tu-bs.de/DLResearch/Issues/df-DlresIssues010313.pdf

3. J.Frew,et.al.,The Alexandria Digital Library Architecture, Int Jour.Digital Libraries (2000)2:259-258, Springer-Verlag 2000.

4. Cezary Mazurek, Maciej Stroinski, Sebastian Szuber, Digital Library for Multimedia Content Management,

5. Rama Chellappa, Digital Image and Video Libraries, www.wtec.org/loyola/diglibs/ob 01.htm

6. J.Leon Jhao et.al., "Data Management for Multi-User Access to Digital Video Libraries, Tech.Report, School of Business Management, The Hong Kong University of Sc.& Tech, Hong Kong, 1999.

7. M.Abrams, C.R.Standridge, G.Abulla,S.Willams and E.A.Fox,"Caching Proxies: Limitations and Potentials," in Proc.4th Int.WWW Conf., Boston, MA, 1995, pp 119-133.

8. J.Almedia and P.Cao, "Measuring proxy performance with the Wisconsin proxy Benchmark," Computer Nerworks ISDN system. Vol. 30, pp2179-2192.

9. Elangelos P.Markatos, Dionisios N.Pnevmatikatos, Michail D.Flouris, Manolis G.H.Katevents, "Web-Conscious Storage Mangement for Web Proxies," IEEE/ACM Trans. on Networking, Vol.10,No.6.December 2002.

10. John dilley, Mortin Arilitt,"Improving proxy cache performance-Analysing three cache placement policies," IEEE Internet Computing, December,2001,pp44-50.

11. A. Chankhunthod, P.B.Danzig,C.Neerdaels, M.F.Schwartz and K.J.Worell,"A Hierarchical Internet Object Cache," in Proc. Usenix Technical Conference., San Diego, CA, 1996,pp 153-164.

12. Charles Brooks, Murray S.Mazer, Scott Meeks, Jim Miller,"Application specific proxy severs as HTTP stream transducer." www.w3.org/conferences/WWW4/Papers/56

13. Georg Gettob, Michael Schrefl et.al., "Extending Object oriented Systems with Roles," ACM Trans.on Info.Systems, Vol.4, No.3,pp 268-296, July 1996.

14. Shabahat Husain, "Library Classification: Facets and Analyses," Tata Mc Graw Hill-Publishing Company Ltd.,New Delhi.

15. F.Maturana, W.Shen, D.H/Norrie, "MetaMorph: An adaptive agent based architecture for intelligent manufacturing." Int.Journal of Prod.Research, 1999, Vol.37, NO.10, 2159-2173.

16. Francis Heylinghen and Carlos Gershenson, "The meaning of Self-organization in Computing," IEEE Intelligent Systems, pp 72-75, July/August 2003, www.computer.org/intellig/intelligent.

17. Jeff Nelson, "Programming Mobile Objects with Java," John Willey & Sons, Inc. 1999.

18. James Goodwill, "Developing Java Servelets: The authoritative solution," Techmedia 1999.