Functions of a Web Warehouse

Kai Cheng, Yahiko Kambayashi, Seok Tae Lee
Department of Social Informatics
Graduate School of Informatics
Kyoto University, Kyoto 606-8501 Japan
{chengk,yahiko,lst}@kuis.kyoto-u.ac.jp

Mukesh Mohania
Department of Computer Science
Western Michigan University
Kalamazoo MI 49008, U.S.A.
mohania@cs.wmich.edu

Abstract

This paper proposes a web warehouse based approach to facilitating efficiency improvement, information sharing and service personalization for the World Wide Web. We will overview various functions of a web warehouse by considering the following applications: (1) a web warehouse as shared information repository, (2) a web warehouse as large-scale intelligent cache. We conclude that web warehouses will play an important role in digital libraries and other online information services.

1 Introduction

In a typical web environment, information is directly sent from a server to a browser for display or interaction. Such a system, however, is faced with increasingly severe scaling problems. First, the per-request based data transfers, as shown in Figure 1, will quickly overload the popular servers and impose a large amount of redundant traffic on the network. Secondly, as a web server can serve countless clients, it is unrealistic to keep track of all clients' behavior and provide personalized services. Finally, the web space is so large that it becomes increasingly difficult to locate relevant and high-quality information in the web.

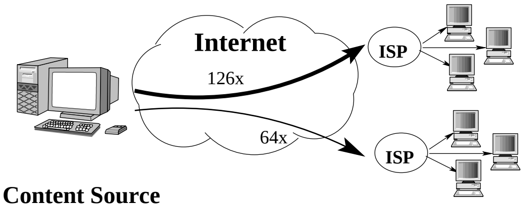

Figure 1. Web environment for direct content delivery

An effective way to improve the system is to provide an intermediate information repository between servers and clients for sharing among a community of users. Figure 2 illustrates such an indirect delivery scheme, where once pages are downloaded into the local site, they can be shared and reused by many clients. This information repository is not just a collection of web contents. In fact, an intermediate repository can keep track of the information the user has viewed, to make it easier to find information again. Or it may enhance the information the user sees by adding annotations and personalization beyond what the server was designed to do [7].

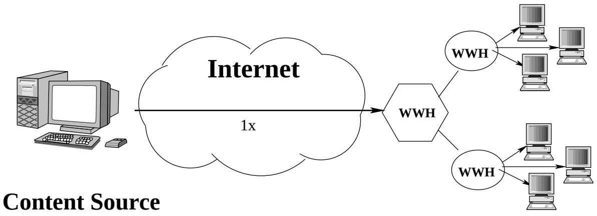

Figure 2. Web environment for indirect content delivery

We are developing a full-featured intermediate information system based on the concept of *web warehouse*. A web warehouse is a repository of localized web information for a given user community. For example, a localized body of web information on an obstinate disease for those who suffer from or care of the disease. A web warehouse is created, updated and terminated in response to the formation, evolution and dissolution of the corresponding user community. In this paper, we will describe the functions of a web warehouse by considering the following applications.

1. Web warehouses as a shared information repository. A web warehouse act as an information server that supports information gathering and provides value added services, such as transcoding, personalization.

2. Web warehouses as a large-scale intelligent proxy cache. A web warehouse is used as a proxy cache that can make best use of various metadata to do cache replacement, such as indices of data, user/community profiles.

The remainder of the paper is organized as follows. Section 2 overviews the architectural consideration and the basic functions of a web warehouse. In Section 3, we will discuss the functions of a web warehouse as a shared information repository. Section 4 describes web warehouse-based intelligent cache. A content-sensitive caching algorithm is proposed. Section 5 shows related work. Section 6 concludes the paper.

2 Overview of a web warehouse

In this section, we present the basic architecture and functions of a web warehouse. A web warehouse uses the web as its input and community-oriented information services as its output.

2.1 Web warehouse for user community

Building a web warehouse, we aim to maximize the sharing of information, knowledge and experience among users. Such sharing is based on *user community*, since a user community is a body of people having common interests, knowledge and experience, or living in the same place under the same laws, regulations and similar cultural background. In other words, a user community is formed by those who are most likely to share the body of information particularly useful to them. As shown in Figure 3, for example, people interested in Home Page design form a user community that own a web warehouse “HP Design”.

Related web warehouses can form a hierarchy. The “Stage Arts”, as in Figure 3, is a web warehouse dedicated to theater performers, directors, designers, and playwrights. It has two sub-warehouses “Playwrights” and “Performers” for playwrights and theater performers respectively. The benefit is that a user community, for instance B, may be only interested in writing plays, in other words, they share most interest in this aspect. A user can belong to multiple communities. Thus, if one particularly likes writing play, but also be interested in general aspects of stage arts, the s/he can join community B and D.

A web warehouse can be dynamically constructed and adjusted in response to the changes of users community. For example, if a web warehouse loses all its users, then the warehouse is no need for existence. In this case, the warehouse will be archived, then become an *inactive* web warehouse. For details, see Section 3.2.

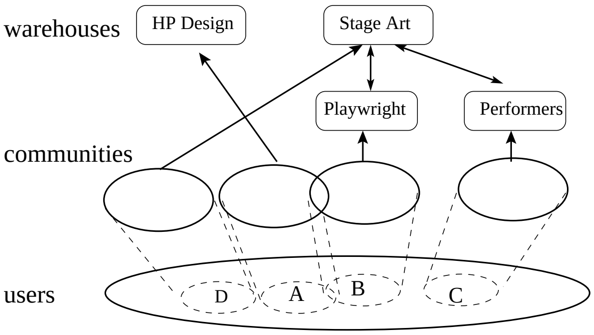

Figure 3. User community based hierarchical web warehouses. A user can belong to several communities and one community can own multiple web warehouses.

2.2 The architecture of a web warehouse

Figure 4 illustrates the basic architecture of a web warehouse, which consists of three main parts: input, storage, and output.

The *input* part is dedicated to resource discovery and content localization. This system provides both automatic and manual modes for resource discovery. Based on user profiles, search engines are used to periodically searching the web. After filtering and clustering, the new contents are then transferred to local storage. Besides this automatic mode, navigation-based resource discovery is also supported.

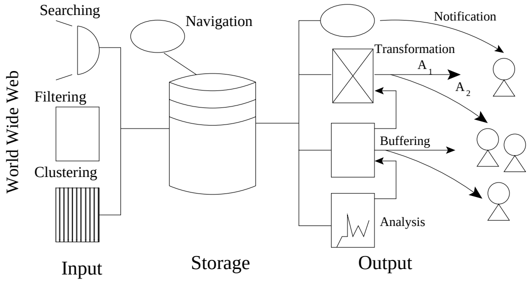

Figure 4. Architecture of a web warehouse consisting of three parts: input, output and storage

The *storage* part is responsible for content management. We distinguish four classes of stored data: prefetched data, cached data, reserved data, and metadata. The *output* part provides basic functions or services for clients. Table 1 lists the basic functions (1) buffering of historic data for sharing and reusing. (2) transformation between different forms, including *transcoding* for different client devices, and *summarizing* for previewing. (3) Notification of information on copyright, new resources. (4) analysis of historic data.

2.2.1 Buffering

Buffering is the most basic function of a web warehouse. By buffering, we mean a process to gather, maintain a collection of information for further use in response to the evolution of user/community profiles. These include prefetched data, reserved data, cached data and metadata. Figure 2.2.1 illustrates the transitions between different classes.

Metadata describe various properties of each data item in a web warehouse to facilitate management and exploitation. The basic model used for metadata is known as "attribute-type-value" model. Metadata are kept for all warehoused data – some may be currently out of storage, for example, the temporarily dropped data.

Metadata standards define sets of attributes which can be used to describe resources, for example, title, author or creator, subject and keywords, description, publisher, other contributors, date, resource, type, format, resource identifier, source, language, relation, coverage, copyrights.

In a web warehouse, we also should consider the following metadata: size, retrieval cost, storage class, downloader, reserver, transcoders. The object *size* is necessary for caching and transformation because a large sized object may be transformed to a suitable format or to a summary before served to the clients. *Retrieval cost* is the average time for downloading the object from its home server, used in cache management. *Storage class* is one of *prefetched* (storage to be determined), *cached* (temporary storage), *reserved* (permanent storage), and *metadata*. *Downloader* is the identifier of the user who downloaded the object. *Reserver* is the identifier of the user who decided to reserve this object. *Transcoders* are the programs used to transform this format to other formats.

Prefetched data are information that is automatically retrieved by the warehouse in terms of the user/community profiles. For example, if a user prefers to information on Chinese medicine, the warehouse will automatically download new documents about this topic when finding through its resource discovery mechanism. Prefetched data are managed separately. When a prefetched document is accessed by at least one user, it will move to be *cached data*, if a "Reserve" button is accessed, it will be also preserved

Reserved data are information explicitly identified to be reserved by some user. When user is browsing a prefetched or cached document, a "Reserve" button is provided. After pressing this button, the document will not be flushed. If a reserved document is released, its metadata will still be kept for further use.

Cached data are temporarily stored for further use. In other words, cached data may be replaced if they are determined as no use in the future. Two operations are available when accessing cached data. When pressing a "Release" button, the data will be released while its metadata, for example, URL, title, size, are kept. Alternatively, if "Reserve" button is pressed, the data will be marked "reserved".

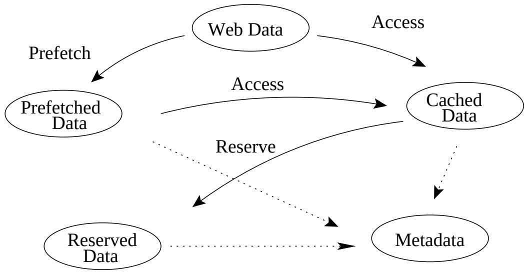

Figure 5. Transition between different data classes. User profiles are based to decide that to be prefetched. The dashed lines indicate to record related information into metadata.

2.2.2 Transformation

Another function of a web warehouse is transformation. For a few reasons, the warehoused data are not suitable for direct delivery to users. First, information format may be not suitable for client devices. Users may access the warehouses from information appliances such as PDAs, cell phones, and set-top boxes. Since these devices do not have the same rendering capabilities as desktop computers, it is necessary for web contents to be adapted, or transcoded, for proper presentation on a variety of client devices. Second, for very large documents, such as high-quality pictures, video files, it is reasonable to deliver to clients a small segment at first time before sending the complete version for the sake of efficiency.

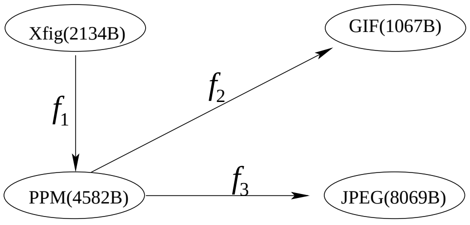

Figure 6. Transcoding between different image formats: the labeled arrows indicates the directions and transcoders, and the numbers in brackets show file sizes.

	Applications	Buffering	Transformation	Notification	Content Analysis
1	Information sharing	o	o	o	
2	Intelligent caching	o			o

Table 1. Basic functions of a web warehouse

Transcoding is the process to transform web contents from one form to another[7]. Transcoding is already commonly used in many applications to change data formats, for instance, to convert a document from one word processor to another. For example, many web pages contain large color images that cannot be viewed on palmtop computers, and XML data on the web often need to be transformed into other forms of XML or possibly into HTML before it can be viewed.

Summarizing is to construct at-a-glance version for previewing from one or more large documents, especially media files, such as video, audio, large sized high-quality picture files. For example, melody is a good summary of a music, a few frames or segments are summary of a video file. Summarizing differs from transcoding in that the former is aimed to reduce unnecessary data transfers before users are sure that the data fit their need, whereas transcoding is

Transformation is useful for content enhancement, communication cost reduction, and adaptability to client environment. Both transcoding and summarizing can be associated with caching technique in either a *demand-based* manner, or a *coverage-based* manner. By demand-based caching, we mean caching the object version resulting from transformation, while coverage-based caching is to cache the origin version on which the transformation is to be applied.

2.2.3 Content analysis

A web warehouse provides a wealth of information in metadata and historic data about the behavior of a user community. Analysis of such data can reveal knowledge that can be used in personalized services. Content analysis provides knowledge for users as well as other functional parts. From the viewpoint of users, it is important to be informed of distribution of contents contained in the current web warehouse. For example, users may want to know what are the most popular contents, what kind of people care of them. From the system perspective, such information can be used in making buffering decision, as will introduced in Section 4.

3 Web warehouse as a shared information repository

A web warehouse provides a platform for user community to share the findings and efforts of each other. As aforementioned, people suffer from a same disease may care most the information about their disease, such as new medicines, treatments, medical institutions and their research. Since such information may scatter all over the world, change frequently, contain truth, falsehood, wisdom, propaganda or sheer nonsense, individuals are difficult to identify, download and manage a complete body of related information.

A web warehouse supports automated resource discovery by integrating technologies for search engine, filtering, and clustering as well as social recommendation. In this section, we discuss the population, utilization and the related issues.

3.1 Resource discovery

Populating a web warehouse is a process to fill the warehouse with relevant contents in accordance with the community profiles.

3.1.1 User profiles

A web warehouse determines what kinds of information are solicited in terms of metadata of known information and user profiles. Metadata describe the properties of information already known to the warehouse, including currently kept, and once kept but now temporarily deleted, while user profiles reflect users' information needs.

User profiles are information about the users, their identifiers, information needs, and so forth. User's information needs are usually described using one or more *user vectors*. User vector is initialized to a common one and evolute in response to the user's behavior, especially user feedbacks.

3.1.2 Searching and filtering

Resource discovery is a process to find relevant resources nearest to the users' information needs. Resource discovery can be done manually and automatically. Manual discovery is initialized by users who input query formula for searching and sift useful resource by themselves. While automatic discovery is done by warehouse periodically.

Search engines use robots to gather and index the contents of web pages, which have been immensely useful tools for resource discovery. However, the problems with using them for this purpose generally fall into two categories:

1. Difficulty in formulating queries that are discriminatory enough to return a "reasonable" number of hits. Queries that contain common terms often return hundreds of thousands of results. Users must filter the relevant resources from the non-relevant, often resulting in relevant resources being missed. A related problem is that the user is given limited discriminatory information beyond the URL of a search result. The user must explore each result to determine the applicability of the result to the query.

2. Lack of "quality control" of results. Many search engines try to "index the Web"without regard to the quality of the resources in the index. Therefore, even if a user enters a very specific query that returns relevant sites, some or all of the sites may be of little value because they lack authority, valid information on the subject, or currency.

To circumvent these problems, we should turn to *i*nformation filtering

Information filtering was introduced as a key technique to overcome information explosion. The common model in information filtering is to create and maintain user profile. Monitoring users' browsing or providing of key phrases that summarize user's interests can achieve generation of user profile. Comparing users' profiles with contents of documents or items using similarity metric can filter documents or items. Thomas W. Malone et al. [9] classify filtering as the cognitive, social, and economic approaches. discusses the a web warehouse as a community-oriented information repository between servers and clients to

1. *Cognitive filtering or content-based filtering.* Several shortcomings have been pointed out on a pure content-based system [1]. First, Filtering of several contents can be supplied only with shallow analysis. Examples of such domains are movies, music, etc. Even for text documents aesthetic aspects and network factors are neglected. Second, over-specialization is also occurred in information filtering as well as other domains. In other words, users are restricted to seeing similar items.

2. *Social filtering or collaborative filtering.* This approach works by supporting the personal and organizational interrelationships in a community. Collaborative filtering can be divided into two techniques. One is that a neighborhood of people who in the past have showed similar behavior will behave similar action to identify new pieces of information. Another is that the choice of an item often leads to the choice of another item. This approach using pre-computed model could quickly recommend a set of items.

3. Economic filtering. This technique rests on many cost-benefit assessments and pricing mechanisms. A web warehouse should consider for each page whether to keep a local copy or just keep its time and URL for economic concerns,

3.2 Clustering and dynamic formation of warehouse

Warehouse based information sharing is aimed to make sufficient use of localized information while not troubling users too much by pouring irrelevant information onto them. To do this, we adopt clustering and dynamic formation of user community.

Clustering. The basic idea of clustering is that similar documents are grouped together to form cluster. The underlying reason it the so-called *cluster hypothesis*: closely associated documents tend to be relevant to the same requests. Grouping similar documents accelerates the searching. A cluster generate procedure operates on vectors or points of t-dimensional space. Each document is represented as a vector of weighted keywords, constructed by a 'indexing' procedure. Based on the vector representation, each pair of documents are then computed the similarity, those with a similarity less than a threshold will join in a same cluster.

Dynamic formation of warehouse. Several rules are based to dynamically form a user community. First, *dynamic split rule* (Figure 7). If a user community is too large and the corresponding web warehouse is , then (1) divide the community into two in terms of the interested contents by clustering and (2) divide the web warehouse into two or three depending whether there are left contents after clustering results.

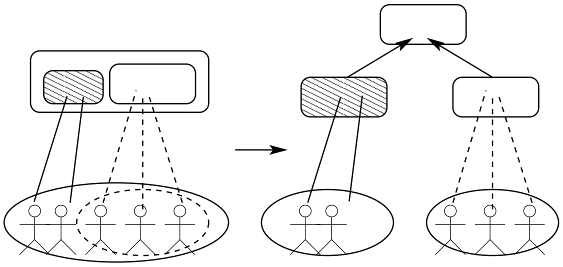

Figure 7. A case for dynamic split of warehouse

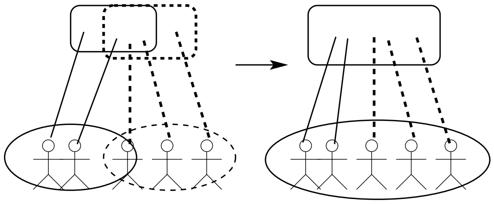

Figure 8. A case for dynamic union of warehouses

Second, *dynamic union rule* (Figure 8). If two web warehouse become similar enough in terms of a threshold, then unite the two web warehouses and the two user communities.

3.3 Localization of web contents

Migration of web contents from origin servers to a web warehouse will result in 'link missing' errors due to the extensive use of relative URL. *Relative URL* is an URL which needs some processing before it is valid. It is a local URL, from which certain information is left out. Often this means some directory names have been left off, or the special sequence ../ is being used. The "relative" comes from the fact that the URL is only valid relative to the URL of the current resource. A relative URL needs the URL of the current resource to be interpreted correctly so that the relative URL is transformed into an absolute URL, which is then fetched as usual.

When migrating an object from its home server to web warehouse, the home directory will change to local disk directory. If the directory structure changed, some links represented as relative URL will be missing. Thus, the URL mapping information should also be kept in metadata that describe the resource of the object.

Recently, *URNs* (Uniform Resource Names)[1] are becoming a generalized form of URLs (Uniform Resource Locators). Instead of naming a resource directly - as URLs do by giving a specific server, port and file name for the resource - URNs point to the resource indirectly through a name server. The name server is able to translate the URN to the "best"(based on some criteria) URL of the resource. The main advantage of URNs is that they are location independent. A single, stable URN can track a resource as it is renamed or moves from server to server. However, due to the scalability and compatibility issues, URNs will not take place URLs in the near future,

3.4 Copyright protection

Warehousing web contents means creating local copies and servicing the copies to a community of users. This may cause problems with copyrighted contents. Most of copyright protection systems work from the assumption that providing accurate copyright information is more important than controlling access, and that most users will respect online copyright if publishers make it easy for them to do so.

In most cases, copyright management of web documents is based on a *notification, not protection* mechanism . For example, *Digital watermarking* uses unique identifiers embedded within digital contents, only proper software can detect and decode it. The mark can contain actual copyright information, such as the author's name and e-mail address, or it can contain an ID number along with a Web URL, fax number or some other means of accessing a database that holds the copyright information.

A web warehouse keep the copyright information as part of metadata for each object. Before, performing transformation operations, such as transcoding, the systems will automatically notify the authors for the usage.

4 Web warehouse as large-scale intelligent cache

Data retrievals from remote web sites are far more expensive than from local sources due to the scarcity of bandwidth and the inherent latency in distance data transport [10]. The rapid growth of the web implies the demand for bandwidth will far out pace the rate of network construction. Even with sufficient bandwidth, the inherent latency is considerable, for example, the latency in transoceanic data transfer is between 200ms and 300ms. Finally and most importantly, web accesses are far from uniform instead hotspots appear suddenly and frequently. Thus, techniques besides network construction are necessary. Caching is a common technique among others to reduce redundant data retrievals, which tries to exploit the overlaps of contents repeatedly requested by different users in different time.

When a web warehouse is populated with contents of common interest to a large number of members in its user community.It is highly desirable to keep the frequently used and highly qualified information for sharing and reusing in the future.However, given a time interval, a specific user or user community will only be interested in and frequently visit a relative small part of it.

Web warehouse provides a large-scale cache of web contents retrieved from different sources in the World Wide Web, which is similar to a data warehouse as buffer of materialized views mediating between information sources and decision support or data mining queries[8, 6].

[1] URN Syntax, IETF Request For Comments (RFC) 2141`http://www.ietf.org/rfc/rfc2141.txt`

4.1 Warehouse enhanced web caching

The major task of a cache manager is to maintain as many popular contents as possible while excluding less popular ones to make space. Two kinds of information are used as indicators to the popularity of an object: (1) *recency*, the time since last reference(s) and (2) *frequency*, the number of references, the more recently or frequently an object has been referenced, the more possible it will be referenced again in the future [3, 11]. Both of them however is only significant to objects with at least two accesses, since both of recency and frequency are based on cumulative information about object usage. There are a large fraction of objects, nearly 60% in cache only being accessed for the first time [11]. For such objects, we call "new objects", there is no way to determine whether they are popular or not, and if they are treated uniformly, a lot of real popular objects would be evicted too early before they begin to get popular.

Warehouse enhanced web caching is capable of inducing the popular topics from current object collection. Because users' access behaviors depend heavily on the contents of web objects: objects with popular contents are much likely to get more accesses than those with less popular contents, the new objects could be evaluated in terms of their similarity to the popular waiting for the subsequent accesses. In this section, we introduce LRU-SP+, a content-sensitive extension to our previous work LRU-SP [5].

4.2 LRU-SP+: a content-sensitive caching algorithm

4.2.1 LRU-SP: popularity-aware SLRU

To deal with the above problem, we improve SLRU by introducing a new factor, the frequency of reference to differentiate the popularity of documents. The idea is, if one hit saves time and retrieval cost once, more hits should reasonably save more times. Thus, the benefit/cost function of document i with $nref_i$ times of references should be:

$$nref_i \cdot \frac{1}{\Delta T_{it}} \frac{1}{S_i}$$

Therefore, to choose an document with least benefit, we should re-index all documents in cache in order of nondecreasing values of $(S_i \cdot \Delta T_{it})/nref_i$ instead of $S_i \cdot \Delta T_{it}$.

$$\frac{S_1 \cdot T\Delta_{1t}}{nref_1} \leq \frac{S_2 \cdot T\Delta_{2t}}{nref_2} \leq \cdots \leq \frac{S_k \cdot T\Delta_{kt}}{nref_k}$$

4.2.2 LRU-SP+: a content-sensitive extension to LRU-SP

The limitation of LRU-SP is that the $nref_i$ adjustment is only useful to the objects with two or more references, while the majority of objects in cache have only one access. Here we distinguish two kinds of popularity: *inherent popularity* and *observed popularity*. In LRU-SP, only observed popularity is considered, so that there is no way to differentiate the newly accessed objects.

The inherent popularity is based on the content of a document, we denote the inherent popularity of document i as ρ_i . For example, a document concerned with a popular topic will be considered as popular. This kind of popularity can be decided even the document is accessed at the first time. ρ_i is computed by,

$$\rho_i = 1 + \frac{\sqrt{\sum_{t \in \Omega}(TF_{t,i} \cdot DF_t)^2}}{N} \qquad (1)$$

where $TF_{t,i}$ is term frequency of t in document i, DF_t is document frequency of term t, N is the size of document collection.

ρ_i defined by formula (1) is calculated on a daily basis, together with the daily indexing operation. The observed popularity is actual reference frequency observed by the cache manager, that is $nref_i$. Here, we simply use the reference count to measure the value. Set Ω of popular topics is calculated periodically, on a daily basis. That is to say, while the indices are updated daily, the ρ_i is also recomputed to keep consistence with the cache contents .

We use a content-sensitive benefit/cost model:

$$\rho_i \cdot \frac{e^{\alpha(nref_i - 1)}}{S_i \cdot \Delta T_{i,t}} \qquad (2)$$

where, α is a parameter belonging to $[0, 1)$, for instance, $\alpha = 0.5$ in our simulation. If $\alpha = 0$ and ρ_i is not considered, it becomes the SLRU case. In terms of (2), even when an object is referenced for the first time, i.e. $nref_i = 1$, it is still possible to differentiate popularity by ρ_i, the content-based inherent popularity.

5 Related work

There are several projects similar to ours, dealing with scalability of the Internet and the World Wide Web. The LSAM Proxy Cache project [12] uses multicast push of related web pages, based on automatically-selected interest groups, to load caches at natural network aggregation points. The proxy is designed to reduce server and network load, and increase client performance. The multicast based approach, however, can not easily provide personalized services. INTELSAT Internet Delivery System (IDS) [4] is based on a Warehouse-Kiosk paradigm, which provides global access and Internet wormholes via a fleet of INTELSAT satellites, the largest commercial satellite communications system in the world. Web contents such as cacheable HTTP, FTP and streaming objects are fetched or pushed both actively and reactively into a central repository cache via intelligent Web agents. Fresh objects are constantly sent via IP multicast reliably to registered Kiosk caches. Distributed Web caches in the Kiosks offer contents to their local users directly with improved quality of service and less bandwidth cost. This is also limited to web performance without transformation function.

Researches on web intermediaries are conducted extensively in IBM Almaden. Web Intermediaries (WBI) [2] is a prototype system aimed to personalize the web through transcoding. It is one of the main functions of our web warehouse system.

6 Concluding remarks

The exponential growth of the World Wide Web results in both the Internet and the web users overloaded. The network suffers from the loss of scalability, while the users become increasingly difficult to find useful information and obtain them efficiently. In this paper, we have proposed an integrated solution to these problems, that is, using web warehouse as a community-oriented intermediate information repository to maximize the sharing of information and experience among a community of users. We have reviewed the main functions of a web warehouse. Web warehousing provides powerful support for data and information sharing among a community of users, which will be particularly significant to digital libraries and other online information services.

References

[1] Marko Balabanovic and Yoav Shoham. Fab: Content-based Collaborative Recommendation. *Communications of the ACM*, 40(3):66 –72, March 1997.

[2] Rob Barrett and Paul P. Maglio. Intermediaries: An Approach to Manipulating Information Streams. *IBM Systems Journal: Pervasive Computing*, 38(4):629–641, 1999.

[3] Pei Cao and Sandy Irani. Cost-Aware WWW Proxy Caching Algorithms. In *Proceedings of the 1997 USENIX Symposium on Internet Technology and Systems*, pages 193–206, December 1997. http://www.cs.wisc.edu/cao/publications.html.

[4] Hua Chen, Marc Abrams, Tommy Johnson, Anup Mathur, Ibraz Anwar, and John Stevenson. Wormhole Caching with HTTP PUSH Method for a Satellite-Based Web Content Multicast and Replication Syste. In *Proceedings of 4th International WWW Caching Workshop*, San Diego, California, March 31 - April 2 1999. http://www.ircache.net/Cache/Workshop99 /Papers/chen-html/.

[5] Kai Cheng and Yahiko Kambayashi. LRU-SP: A Size-Adjusted and Popularity-Aware LRU Replacement Algorithm for Web Caching. In *Proceedings of 24th IEEE Computer Society International Computer Software and Applications Conference (Compsac'00)*, pages 48–53, Taipei, October 2000. IEEE Computer Society Press.

[6] Kai Cheng and Yahiko Kambayashi. Multicache-based Content Management for Web Caching. In *Proceedings of 1st International Web Information Systems Engineering(WISE'00)*, pages 42–49, Hong Kong, June 2000. IEEE Computer Society Press.

[7] Masahiro Hori, Goh Kondoh, Kohichi Ono, Shin ichi Hirose, and Sandeep Singhal. Annotation-based Web Content Transcoding. In *Proceedings of 9th International World Wide Web Conference*, Amsterdam, May 2000. http://www9.org/w9cdrom/index.html.

[8] Matthias Jarke, Manfred A. Jeusfeld, Christoph Quix, and Panos Vassiliadis. Architecture and Quality in Data Warehouses: An Extended Repository Approach. *Information Systems*, 24(3):229–253, 1999.

[9] Thomas W. Malone, Kenneth R. Grant, Franklyn A. Turbak, Stephen A. Brobst, and Michael D. Cohen;. Intelligent Information-Sharing Systems. *Communications of the ACM*, 30(5):390 – 402, May 1987.

[10] Venkata N. Padmanabhan. *Addressing the Challenges of Web Data Transport*. PhD thesis, Computer Science Division, University of California at Berkeley, USA, September 1998. Also published as Technical Report UCB/CSD-98-1016. Available at http://www.research.microsoft.com/ padmanab/phd-thesis.html.

[11] Luigi Rizzo and Lorenzo Vicisano. Replacement Policies for a Proxy Cache. Technical report rn/98/13, University College London, Department of Computer Science, Gower Street, London WC1E 6BT, UK, 1998. http://www.iet.unipi.it/ luigi/ caching.ps.gz.

[12] Joe Touch and Amy S. Hughes. The LSAM Proxy Cache - a Multicast Distributed Virtual Cache. In *Proceedings of 3rd International WWW Caching Workshop*, Manchester, England, June 1998. Also in Computer Networks and ISDN System, V30 N22-23, Nov. 25, 1998, pp. 2245-2252.