Architecture of a P2P Distributed Adaptive Directory

Gennaro Cordasco
cordasco@dia.unisa.it

Vittorio Scarano
vitsca@dia.unisa.it

Cristiano Vitolo
cvitolo@n3xt.it

Dipartimento di Informatica e Applicazioni “R.M.Capocelli”
Università di Salerno,
84081 Baronissi (SA) – Italy

Categories and Subject Descriptors

C.2.4 [**Distributed Systems**]: Distributed Applications; H.3.4 [**Systems and Software**]: Distributed Systems; H.5.4 [**Hypertext/Hypermedia**]: Navigation

General Terms

Design

Keywords

Adaptivity, Bookmark sharing, Peer to Peer

1. INTRODUCTION

Bookmarks are, nowadays, an important aid to navigation since they represent an easy way to reduce the cognitive load of managing and typing URLs. All the browsers have always provided, since the very beginning of the WWW, friendly ways of managing bookmarks. In this paper we deal with the problem of enriching this supportive framework for bookmarks (as provided by the browsers) by adding collaboration and (group) adaptation with a P2P system.

In this paper, we describe a system that offers a distributed, cooperative and adaptive environment for bookmark sharing. DAD (Distributed Adaptive Directory) offers an adaptive environment since it provides suggestions about the navigation based on *(a)* the bookmarks, *(b)* the feedback implicitly provided by users and *(c)* the structure of the Web. DAD is fully scalable because of its peer-to-peer architecture and provides, also, an infrastructure to build easily P2P overlay networks.

Our system is structured on three separated layers (see Fig. 1): on the bottom we have an overlay network named CHILD (CHord with Identification and Lonely Discovery). Because of scalability and fault tolerance, our architecture is a pure Peer to Peer system: we realized a Distributed Hash Table (DHT) to store information about bookmarks. The DHT is based on the well-known Chord [5] protocol, which we extended with a boot mechanism through caching (i.e., no server is required), a mechanism that allows registration and authentication on the system and a tool that allows to save and reload the DHT data after the system shuts itself down or is started again.

We would like to emphasize that the infrastructure provided by CHILD can be used as a base to develop a DHT for any pure P2P system. In fact, CHILD is implemented by a Java package that allows to build and manage any generic DHT, acting as middleware platform for P2P systems. The programmer can, therefore, implement the business logic of his application on top of our layer. CHILD will be soon freely available under the GNU public license at the project home page (http://isis.dia.unisa.it/projects/DAD).

WWW2004, May 17–22, 2004, New York, New York, USA.
ACM 1-58113-912-8/04/0005.

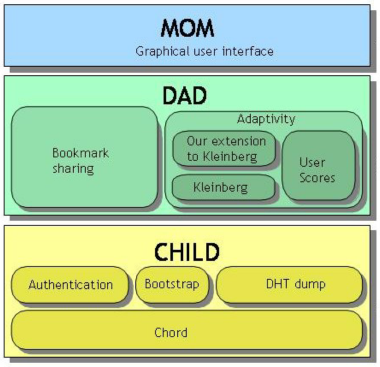

Figure 1: The diagram of the architecture of our system.

The middle layer, named DAD, exploits the bottom level to obtain information about bookmarks and users, providing a collaborative and adaptive system to manage bookmarks. Shared bookmarks are placed in an ontology that is common to all the peers and the adaptivity provided by the system is based on a modified application of the Kleinberg algorithm to evaluate hub and authorities on a web structure. The middle layer allows users to rank the bookmarks from other users as well as bookmarks suggested by the system.

Finally the top layer, named MOM (Multipurpose Ontology Manager), realizes the graphical user interface that represents a typical DAD-peer and allow to use all the functionalities of our system. It provides a smooth interaction with the top most used browser (MS Internet Explorer) but can be used with any other browser.

2. TECHNICAL DETAILS

The P2P infrastructure. The architecture of our system is not based on client-server architecture. In fact, our system should offer a distributed framework where the load of managing and storing the bookmarks is equally distributed among peers because such solution is inherently scalable with the large number of users that are potentially interested in bookmark sharing. Moreover, partial failures are supported by this kind of architectures: if a peer abruptly leaves the system, only the portion of information located there is lost but the rest of the system behaves correctly.

Bookmarks that are shared are physically stored in the peer that is considered its *owner*, i.e., the peer that independently (without other users' or system's suggestions) placed it into the system.

A distributed hash table is used to store information on how to locate bookmarks that are in a category of the ontology. This layer is based on Chord protocol [5], adding a boot phase, registration and authentication, and save/reload capabilities.

The Distributed Adaptive Directory. As pictorially described by the architecture of the system in Fig. 1, the functionalities of the middle layer are divided in two parts: the component that is in charge of sharing the bookmarks and the component that offers the adaptation by suggesting bookmarks and collecting feedback and scores for bookmarks and users.

Several different kind of representation are used in literature for sharing bookmarks. One of the main differences of the available bookmark sharing systems is whether the system asks users to build their own classification (such as Opencola (http://www.opencola.com)) or archives the bookmarks in a fixed ontology (such as Widesource (http://www.widesource.com)).

Our architecture uses an intermediate approach: while we recognize the importance of providing a fixed infrastructure for the shared bookmarks we also let the users organize their own bookmarks as they prefer.

Bookmarks organization is typically a crucial key to efficiently place landmarks on the information space. Since a high percentage of users does not organize at all the bookmarks (or organize them poorly) [1] we believe that leaving the users create their own ontology for sharing bookmarks can be confusing. The compromise proposed here allows users to create their own organization (even none, if that is the choice) for local bookmarks but share (or importing) them with a fixed ontology.

As base ontology we choose a shortened version (i.e., only the first four levels) ontology of the Open Directory Project DMOZ [6] stored by using XML Bookmark Exchange Language (XBEL) [7]. The limitation in size of the original version of the ontology provided by DMOZ is due to the size: since it is distributed with the application, the complete version was too large (almost 300 Mbytes) to ensure easy downloading of our software.

Bookmarks are stored in the DHT that is acting like a distributed filesystem. When a user places a bookmark in the ontology (i.e., makes it shared with the system) his/her peer is in charge of storing the bookmark for the whole system. When a user needs to open a category (folder) in the ontology (an operation called *expansion*) his/her peer needs to know all the peers that are storing bookmarks for the category (called *interest group*).

The DHT (offered by the bottom layer CHILD) stores information about each interest group. Each category of the ontology is associated with an ID, the peer interested in the bookmarks of a category, first, accesses the DHT to get the members of the associated interest group and, successively, asks each member the bookmarks that it contributed to the category.

Joining/leaving an interest group is totally transparent to the user and is managed by the operations of insertion/deletion of bookmarks into the ontology.

Adaptivity. One of the many interesting research lines that are stemmed from the World Wide Web is the field of adaptive hypermedia [2]. The goal is to provide dynamically adaptive navigation/content to users in such a way that their interaction with the WWW is tailored to personal inclinations, tastes, attitudes and predispositions. The system presented here has similar goals but, rather than classify resources and rely on profiles for users and groups, we tie the adaptivity of our system to user bookmarks as well as the underlying structure of the web. The system is also adaptable since the user can explicitly tune how relevant bookmarks are presented.

In order to add adaptivity to our system we use a revised version of Kleinberg algorithm [4] that is used in systems like CLEVER [3]. The algorithm allows to discover authoritative sources of information (authorities and hubs) and is based on the links structure of the Web. The goal of the algorithm is to assign a score to nodes of a graph (i.e., HTML pages) based on the edges among them (i.e., links among HTML pages). Of course, pages with high scores are signalled as significative.

Our extension to Kleinberg algorithm consists in adding users to the set of nodes (i.e., bookmarks) therefore modelling the interactions between users and bookmarks. In this way, we add a new type of nodes (i.e. users) and new links (from a user to a page or from a user to another user) on the graph G. The new kinds of links have the following meaning:

- **from user U to a page P:** when U is the owner of page P, i.e., U placed P as shared bookmark into the category of the ontology.
- **from user U to user V:** when U is showing interest in a bookmark that was inserted in the system by V, i.e., when U copies the bookmark inserted by V locally among his/her private bookmarks.

We define a user score for each category in the ontology. At the beginning, each user has score set to 0 for each category. When user U reveals interest in bookmarks B owned by V, then the score of V (in the category where the B is placed) is increased. The interest of U in B is manifest when U decides to mark B (i.e. copy it locally), in such a way to be able to use B even if V is not on.

User scores are stored in the DHT as fields of the interest group[1]. For each category, the information about the interest group consist of the list of members, each with his/her score in the category.

Now, each bookmark has four different kind of scores:

- **authority and hub weight:** provided by Kleinberg algorithm;
- **owners' scores:** the scores of the user(s) that inserted the bookmark;
- **occurrences:** the number of users that keeps locally the bookmarks stored.

According to user preferences, the system computes a global score in a $[1, 10]$ range where 10 indicates a very interesting bookmark. Then, the bookmarks in the folder are sorted and presented to the user, each with its global score.

3. CONCLUSIONS AND FUTURE WORK

Our system represents a "pure" P2P application (therefore scalable) for managing communities of users that want to share the resources that they found, being supported by an adaptive mechanism and by a highly tunable ranking mechanism.

As future developments of this project, we plan to investigate the necessary modifications to our system in order to support bookmarks placed simultaneously in multiple folders. Several advantages can be obtained by this extensions as, for example, enrich the hierarchical structure of the ontology with cross-links among categories which several bookmarks contemporaneously belong to.

Another possible avenue of research is to provide a mechanism to suggest the category where to place a bookmark, based on an analysis of page content.

4. REFERENCES

[1] D.Abrams, R.Baecker, M.Chignell, "*Information Archiving with Bookmarks: Personal Web space Construction and Organization*". In Proc. of Human Factors in Computing Systems (CHI 98) ACM Press, pp 41-48.

[2] P. Brusilovsky, "*Methods and techniques of adaptive hypermedia*". User Modeling and User-Adapted Interaction, 6 (2-3), 87-129, 1996.

[3] S. Chakrabarti, B. Dom, D.Gibson, J. Keinberg, P. Raghavan and s. Rajagopalan, "*Automatic Resource list Compilation by Analyzing Hyperlink Structure and Associated Text*". Proc. of the 7th Inter. World Wide Web Conference, 1998.

[4] J. Kleinberg, "*Authoritative sources in a hyperlinked environment*". Journal of the ACM, 1999.

[5] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, H. Balakrishnan, "*Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications*", IEEE/ACM Transactions on Networking February 2003.

[6] DMOZ, Open Directory Project, http://www.dmoz.org/

[7] http://pyxml.sourceforge.net/topics/xbel/docs/html/xbel.html

[1]There is a single DHT that, for each ID, stores information on a peer (user), information on a folder (interest group) or both.