Seamlessly and Coherently Locating Interesting Mirrors on the Web

Neilze Dorta Pierre Sens
Christian Khoury

Université Paris 6 - Lip6/CNRS
email : {Neilze.Dorta, Pierre.Sens, Christian.Khoury}@lip6.fr

INRIA - Project SOR
email : Neilze.Dorta@inria.fr

Abstract

Nowadays, the World Wide Web is used mostly as a common medium for information sharing. Therefore, locating an object on this large scale dynamic medium tends to be more and more difficult. Content Distribution Networks, e.g. Akamai, and global naming services, e.g. Globe, do more or less than what is required by most users. In this paper, we are interested in discovering, advertising, and transparently locating *interesting* mirrors of interest to a group of users. Our solution, *ARÃ*, is user-centric; it uses cooperation among organizations to discover, publicize and locate coherently new mirrors that are of interest to them. Access transparency is achieved through a naming service that manages the different aliases for the same replica. Consistency guarantees are given to each user that no document delivered would be older than the one viewed before. The system scales geographically due to the epidemic and asynchronous nature of the cooperation protocol. We propose a methodology for creating homogeneous groups with common interests, using collected Web traces, then give a glimpse of the potential benefits made by using *ARÃ*. It opens a path towards making mirroring ubiquitous, hence fostering a better use of the Internet and its resources. A prototype has been implemented in Java and will be used, in the future, in real-world tests for more accurate and realistic results.

1 Introduction

Today, the World Wide Web (WWW) is the largest information sharing medium. Unfortunately, its popularity highlights problems in access latency for users, overloaded servers, congestionned networks, and network partitions. A widely acknowledged solution to this problem is persistent replication, also known as mirroring; it reduces access latency, network traffic, servers' load, and improves the system's tolerance to network partitions. However, information replication entails a location system with two important properties :

- locates the "best" replica for the requested document with respect to each user,
- locates the most recent version of the document.

Today, locating is done almost exclusively as described below :

1. users find a lists of mirrors , e.g, by surfing on some web page or by any other means. This scheme is not systematically used by users, if at all. On the other hand, locating a mirror this way is time-consuming, thus eliminating fully or partially the benefits of mirroring.
2. Content Distribution Networks (CDN), e.g. Akamai [1], offer their services to big companies to make access to their web pages faster. This scheme is

> useful for organizations that can afford paying such services, but most of all, those who wish to share public information. It leaves unsolved the question of sharing internal documents or sharing documents amongst a very restrained group of organizations.

In this paper, we present *ARÃ*, a *user-centric* location service with consistency guarantees. By user-centric, we mean that it only locates mirrors that are of interest to its users. It is based on cooperation among groups with related interests, sharing location information about mirrors using an epidemic and asynchronous replication protocol. As for consistency guarantees, it enforces *incremental consistency* to ensure that no user would get an older version of a document it has already viewed.

The remainder of this paper is organized as follows. The next section gives a summary of related work. Section 3 sketches the architecture of *ARÃ*, its inter-group cooperation protocol for discovering and publicizing mirrors. Section 4 presents a methodology for identifying groups with common interests, then describes the results of simulations done to substantiate the need for *ARÃ*. Section 5 presents our conclusions and directions for future research.

2 Related Work

[2] examine three possible schemes for "seamlessly locating replicas". They propose using the HTTP [3] *REDIRECT* header to point a request at the "best" replica; the main disadvantage of this method is that it is up to the server to decide which replica is better without any clear knowledge of where the client is and what would really be best for him. They also propose using the DNS scheme of determining the best DNS server, and then put the name resolution of the "closest" replica in this DNS server; the main disadvantage of this scheme is the time to determine the best DNS server. The third method is based on shared IP addressing and "Closest Exit Routing". They propose to give all the replicas the same IP address and let the internal network redirect the request to the closest; the main disadvantage is that all replicas should be on the same autonomous network.

[4] proposes *LDS*, a system similar to DNS in its architecture; it manages not only name to IP mapping at layer 3, but also at the application layer (layer 7 in the OSI model), namely URL to IP. The purpose is to map all the objects on the Internet, which seems unrealistic with an architecture such as DNS originally designed for almost immutable mappings.

Content Distribution Networks, such as Akamai [1], propose a global solution for locating the "best" replica. We will discuss the case of Akamai, knowing that other CDNs proceed in almost the same way. The solution is based on a proprietary and paying network of servers. The first stage is to "akamize" the Web site or documents to be served which entails changing the links inside local Web pages to point to the akamai network holding replicas of theses links' contents. Upon receiving a client's request, the origin server sends the modified Web document to the client which then requests the rest of the document from the Akamai network. The main technical pitfall of this solution is that it depends on the first HTTP request being served correctly and rapidly, otherwise all the benefits of mirroring are lost. Its main usage problem is that it is a globally shared network of mirrors, thus not having the possibility of taking into account mirroring for internal use. Finally, its economical pitfall is that it is not obvious how small organizations/associations could ever afford the services of such a system.

The solutions sketched have a global approach to mirroring, not centered on users' interests; as users have different interests, it is mostly important for a location service to locate *only* those mirrors interesting to its users. This is the approach we follow in the design of *ARÃ*.

3 Design and Architecture of *ARÃ*

Due to lack of space, we will briefly introduce the design and architecture of *ARÃ*. It is a *user-centric* transparent location system as it *only* locates mirrors that are of interest to its users. It is based on an intra/inter-group cooperation model for discovering and advertising mirrors on the Web. The Cooperation model has been chosen as it is used, in general, for structuring large scale systems; its main advantages are decentralization and extensibility.

The smallest unit in *ARÃ* is an *organization*. It represents, in general, a physical entity, such as a company, a research institute, a university, a research lab, or even a mixture of the latter meeting the above definition.

A *group* (see figure 1) is defined as a set of organizations having common interests and very well connected. Discovering and advertising mirrors is done at this level, as each participating organization shares its local location metadata with the groups to which it belongs; an organization may cooperate with other organizations in more than one group, as shown in figure 1.

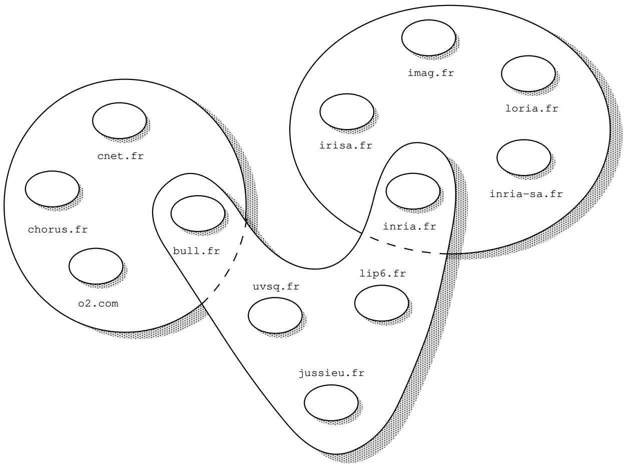

Figure 1: Groups

ARÃ is deployed as a proxy in each organization; we assume that all nodes inside the organization access the Web via this proxy. Hereafter, we explicit the interaction between users and an ARÃ proxy :

- the user makes a URL request via a Web browser,
- the *ARÃ* proxy intercepts this URL request,
- *ARÃ* queries its mirroring database (MDB) for this URL,
- *ARÃ* replies with a list of available mirrors[1]

3.1 Discovering and Advertising Mirrors

The process begins by a discovery of an interesting new mirror, by an organization[2]. The latter updates its local mirroring database(MDB) and sends the location information to its peers using an epidemic protocol : it consists in sending messages using *asynchronous* multicast communication to all the groups it belongs to. Upon reception of a message, an organization updates its database and forwards it to its groups, and so on. Problems arise from such a protocol : messages could either go into cycles as shown in Fig. 2, or they could be forwarded endlessly as long as there are groups interconnected.

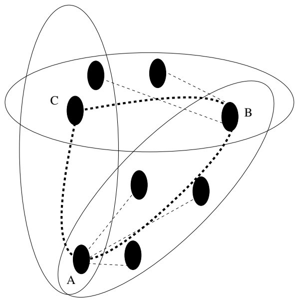

Figure 2: Cycles between A and C

We solve these problems using :

- a *contamination degree* associated to each group (CDG); its value determines the distance in terms of groups from which to accept messages,
- a counter (CDM) associated to each message giving the number of groups it has contaminated; it is hence incremented each time it crosses a group boundary

Consequently, a group accepts messages having a CDM lesser than its predefined CDG. The latter reflects some kind of reference locality between groups as it decides whether location metadata some groups away is still interesting to use or not.

In Fig. 3, we depict the evolution of a message and how it is received by other groups. *O*rg 2 sends a message to its group (Group 1) and CDM is set to 1. *O*rg 4 receives the message and forwards it to *G*roup 2; this is possible as the contamination degree of Group 2 is equal to 3. *Org 7*, which belongs to Group 2 and Group 3, receives the message with a CDM equal to 2, and "contaminates" Group 3; the latter has a contamination degree set to 2 and can

[1] the list might be an empty one
[2] this could be a mirror created by this site

therefore accept the message. *Org 10* receives the message from *Org 7* with a CDM of 3. *Org 10* decides not to "contaminate" Group 4 as the latter's contamination degree is set to 2, thus not satisfying the epidemic condition.

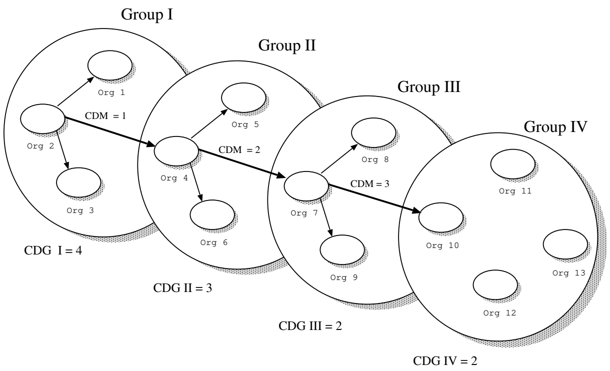

Figure 3: Contamination Example

For efficiency reasons, message propagation is done asynchronously; this is justified by the relatively long lifetime of a mirror. Messages are packed and sent when *ARÃ* is more or less idle, moreover, they are sent grouped and, if possible, piggybacked. One of the main advantages of asynchronous communication is its reduction of network traffic when compared to synchronous communication.

3.2 Transparently and Coherently Locating Mirrors

Documents on the web are identified with a URL; therefore, as the location is part of the name, a replica of a document will usually have a different prefix corresponding to its new location, e.g., `a.b.c/Object_Name` and `d.e.f/Object_Name`. *ARÃ* solves this naming problem using an associative table of location prefixes, linking prefixes of all replicas together - mirrors and server. The naming server, upon receiving a request for a document, uses the prefix table to return a list of the *best* servers available, if present.

Accessing the "wrong" server can be costly in terms of access time and resources[3]. We base our choice on *access time* [5, 6, 7, 8]; *ARÃ* computes users' access time to all mirrors by sending *ping* messages[4] periodically at different hours of the day; a weighted average of all these measurements, past and present, are combined to produce the final mirror/server access time relative to *ARÃ*.

3.2.1 Consistency Sessions

The *minimal* consistency requirement is for users to view documents of *at least* the same version or more recent than what they have seen already; that is what we call *incremental consistency*. With asynchronous replication of mirrors, as it is done on the Web, differing versions of the same document can coexist.

In order to ensure *incremental consistency*, *ARÃ* offers *session guarantees* [9] handled on a user-basis. Inside a session, for each request, the version of each of the possible servers in its MDB should be fresher or equal to that of the mirror that answered the last request with the same location prefix.

We assume that each server/mirror has a version associated to its contents. Any change to the latter result in a change of the version. A query, either as part of the HTTP protocol, or implemented in a CGI-like manner, should return the version of the queried server/mirror. This method is fallible, as its granularity for defining a version is coarse. Nonetheless, it is realistic as the number of documents is large. *ARÃ* keeps in its MDB, this version field for each of the server/mirrors.

A user signals the beginning of a session to the *ARÃ* proxy; the latter then creates a *session identifier* for this user. *ARÃ* creates, for each of its associated location prefixes, a version vector **[VSLA, VRS1, ..., VRSn]** with the following format :

- **VSLA** : **V**ersion of the **S**erver **L**ast **A**ccessed
- **VRSi** : **V**ersion of the **R**eplicated **S**erver **i**

where **n** is the number of servers (mirrors) in the list. Incremental consistency is ensured if the version of the server to access is *at least* equal to or fresher than that of the server last accessed : $\mathbf{VSLA} \leq \mathbf{VSRi}$

[3]Network, CPU, ...

[4]these messages are also used as keep-alive messages

4 Simulation Results

In this section, we briefly present simulation results showing the benefits of using *ARÃ*. We used traces from three french institutions : INRIA[5], ENST[6] and INT[7]. Traces characteristics are shown in Table 1.

Table 1: Characteristics of collected traces

	Requests	Start	End
INRIA	3440847	12/8/1998	3/1/1999
INT	2593281	23/11/1998	15/2/1999
ENST	236436	13/9/1998	8/10/1998
Total	6270564		

We measured locality for each of the collected traces; results (see Fig. 4, 5) show that a large number of requests (more than 60%) and traffic (more than 70%) go to less than 10% of the servers for all three traces. This implies that these corresponding servers, for each trace, are potential candidates for mirroring. We then measured the percentage of common potential mirror candidates between each two sites and the amount of traffic the latter are responsible for. Results show that the three sites have common candidate mirrors (up to 50%) and that the latter are responsible for a large amount of requests (up to 85%). Thus, these three organizations would form a perfect *ARÃ* group. We propose, the scheme we just described, to identify and form groups. The list of common mirrors will be used hereafter in our simulations.

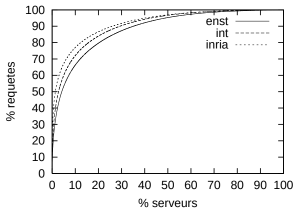

Figure 4: Number of requests per server

[5] Institut National de Recherche en Informatique et Automatique
[6] Ecole Nationale Supérieure de Télécommunications
[7] Institut National de Télécommunications

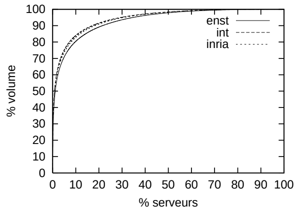

Figure 5: Traffic per server

For our simulations, we place the mirrors in areas closer to users than the real servers, based on access time. We then compare the benefits and costs of locating them based on a list of criteria : hit ratio, global access time, inter-network traffic, memory costs, cooperation costs. One representative simulation (see Table 2) shows that using *ARÃ*, compared to using an ideal system which locates mirrors perfectly, results in almost the same benefits with minor costs.

Table 2: Representative simulation results

	↑ % Hit	↓ % Latency	↓ % Traffic	Mem. Cost	Coop. Cost
ARÃ	99.93	96.63	68.72	837 KB	2042 msg
ideal	100	96.79	68.82	279 KB	0

What is interesting in these results is that *without ARÃ*, all the benefits, or at least a big part are lost as mirrors are not, or partially discovered. This is even true for global location systems, such as Akamai, which cannot locate mirrors defined locally by organizations or not declared into their system. The main point in using *ARÃ*, is that cooperation is the key to discovering mirrors, which could otherwise stay undisclosed, and therefore *all the benefits shown are lost*.

5 Conclusion

Nowadays, the World Wide Web is an overloaded information sharing medium. Data replication is used, first and foremost, to improve users' access latency. However, it introduces a problem of how to locate these replicas and access the closest according to each user's location. Location systems proposed tend to locate *all* the objects, an approach which is, in our view, complicated and unnecessary in this context as : (1) the large number of objects tend to make the location database huge, and (2) users address only a bunch of objects that are of interest to them, the others could be requested at their source. In this paper, we presented *ARÃ*, a *user-centred* system that *only* locates objects of interest to its users. It is based on cooperation among organizations of common interests, using an epidemic asynchronous protocol to discover and publicize mirrors' location. Besides, *ARÃ* tackles a forgotten issue : Consistency. It offers consistency guarantees to every user : has a version of a document been viewed before, the user will always be served a newer or equal version; that is what we call *incremental consistency*. Finally, we examplified a methodology for defining groups and showed, using simulations, the potential benefits of using *ARÃ*; without cooperation, location benefits could be lost as mirrors are hardly discovered. We have already implemented a version of *ARÃ* which we intend to use in real world situations for a more realistic study of its impact on Web accesses.

References

[1] Akamai. http://www.akamai.com/.

[2] Yair Amir, Alec Peterson, and David Shaw. Seamlessly selecting the best copy from internet-wide replicated web servers. In *DISC'98 - 12th International Symposium on DIStributed Computing*, Andros, Greece, Sep 1998.

[3] T. Berners-Lee. Hypertext transfer protocol http/1.0. " ", HTTP working Group Internet draft, October 1995.

[4] Jussi Kangasharju, Keith W. Ross, and James W. Roberts. Locating copies of objects using the domain name system. In *Proceedings of the 4th International Web Caching Workshop*, San Diego, California, April 1999.

[5] Andy Myers, Peter Dinda, and Hui Zhang. Performance characteristics of mirror servers on the internet. *School of Computer Science, Carnegie Mellon University*, June 1998.

[6] Samrat Bhattacharjee, Mostafa H. Ammar, Ellen W. Zegura, Viren Shah, and Zongming Fei. Application-layer anycasting. In *INFOCOM'97*, 1997.

[7] Robert L. Carter and Mark E. Crovella. Dynamic server selection using bandwidth probing in wide-area networks. Technical Report BU-CS-96-007, Boston University, (USA), March 1996.

[8] C. Partridge, T. Mendez, and W. Milliken. Host anycasting service. Request for comments 1546, (USA), November 1993.

[9] Douglas Terry, Alan Demers, Karin Petersen, Mike Spreitzer, Marvin Theimer, and Brent Welch. Session guarantees for weakly consistent replicated data. In *International Conference on Parallel and Distributed Inormation Systems*, Austin, TX, US, September 1994. http://www.parc.xerox.com/csl/projects/bayou/.