Document Distribution to Internet Digital Libraries

L. Navarro*, V. J. Sosa**, O. Ardaiz*

Universitat Politécnica de Catalunya (UPC)*
Centro de Investigación en Computación (CIC-IPN)*
{leandro, vjsosa, oardaiz}@ac.upc.es
http://www.canet.upc.es/mweb

Abstract

The success of the Web has produced an unmanageable volume of traffic, eating up the available amount of network capacity. Hierarchies of proxy-cache servers may help to reduce the number of times any document is downloaded by saving a copy for reuse in future requests, in exchange for many cache validation requests. Replication, a complementary strategy to caching, is presented, based on Web content distribution. When a document is published, it is distributed to library servers (storage servers subscribed to certain topics) instead of waiting for requests. It is a network for document distribution that minimises the number of times every document circulates through the network, and moves that distribution to hours of low traffic, reducing the peak load traffic on the network.

A Web proxy server within an organisation will serve requests by returning objects from the cache (someone just downloaded a non-expired copy), from the library (we are subscribed to that topic), or finally from the original source. For example, when a department library subscribes to a publication in electronic format, the document distribution service will provide them with a copy of the new issues shortly after (or before) publication, without any penalty for the first reader, and avoiding expensive traffic at the times of read requests.

This is reflected in the Object Distribution System (ODS), a distribution system inspired by the ultra-large scale distribution models used in everyday life (e.g. food, books, newspapers). In addition traditional mechanisms of bringing information to readers, such as caching and mirroring, this system enables the publication, classification and subscription to collections of documents. There is also provision for classification authorities to offer classification schemes for labeling documents.

The scenario, and the additional advantages over proxy-cache in some situation for some documents, are presented. The status of the implementation of the model is discussed at the end.

Key Words:
Proxy, digital library, cache, web

1 Introduction: The Status of the Network

The Internet is carrying out the work of transporting Web documents increasingly slowly and in unpredictable manner. This effect is the result of a complex combination of phenomena, among which are the explosive growth of the user population, the frequent and repetitive traffic of documents, and the proliferation of many contents of diverse value for readers.

The quality of service on Internet varies greatly: most of the time, the aggregated load generated by the growing population is very variable, and chaotic (self-similar, fractal), no matter what the capacity of the network (Leland, 94) (Crosvella, 95).

This is aggravated by the load and failures in servers due to a large number of simultaneous requests and network partitions, specially on remote documents. When a massive number of users have a common interest at a certain moment, this causes a storm of requests to the server and an overload to their network vicinity. This can be observed in the case of resources that thousand of users want to access simultaneously, and which
are available via Internet before any other media: results from
competitions, elections, images from an important event, etc.

As a result of the aforesaid effects, many resources become
inaccessible for a large number of users, who feel frustrated
in their expectations of the Internet.

In addition, the bad news is that Web traffic grows more
than other services: requests point anywhere in the network,
instead of email, news, dns where there are two regions: clients
talk to a local proxy server, and proxies talk among themselves.
This separation facilitates a sustainable growth: Imagine what
would happen if we had to go visit the author every time we
were interested in reading a book !.

Another problem is the large volume and diverse quality of
information on the Internet. Generally speaking, there are no
guarantees about the quality and reliability of such information.
Web pages are not usually classified in terms of any criteria.
Classification schemes are collections of labels or topics
produced by a classification authority, and they are used to
associate meta-information to objects. They are used by
authors, readers, librarians and the distribution network for
describing the documents.

The goal is to provide local access to relevant and pre-
selected information; to obtain the best service from the
ordered use of global interconnections where bandwidth is
scarce, quality is unstable and network partitions occur too
often; and to provide a global and cooperative mechanism for
content classification and qualification (metadata).

We focus on a model centered on communities or
organisations who are producers and consumers of
information: they may produce, classify, label, offer and
publish information, and also look for and consume
information produced by other distant communities. These
interactions occur with a local (region, organisation) service
agent or library, while object distribution is done
asynchronously, reliably and cooperatively among agents
located anywhere on Internet. This model is appropriate
because intra-community networking is usually adequate,
whereas external networking is usually poor and more
expensive.

Subscription to contents is central to this model: we
understand subscription as a contract between a reader (the
subscriber) and the "network" to receive certain contents in
the future. This is opposed to the usual visitor-content relation
on the Web where a visit is an isolated act in the time.
Subscription contracts can be used to predict and organise an
efficient distribution of contents. People usually subscribe to
resource (the URL of a Web page, a set of documents, a soft-
ware package, a multimedia content), or to a group of contents
with the same characteristics (newsgroup, keyword-based
query results).

2 Document Distribution

It is inspired by many distribution models used in everyday
life (e.g. food distribution chains, publications). Consumers
don't go to places where goods are produced (e.g. factories,
author's home). Goods are purchased in the closest retail shop
(a proxy), where most products are in stock waiting for
customers (even though sometimes goods are back-ordered).
Factories (a server) produce at a near optimal pace, supplying
distributors and retailers. This system works because
consumers trust their retail shops: shops provide fresh products
at a reasonable price, probably offer a better deal than one
could get from a factory.

This model is adequate for a very large scale. It does not
exist in the current Internet community, but it may be
introduced over the existing networking infrastructure without
modifying protocols and standards. Its progressive introduction
provides immediate advantages for users.

Distribution differs from caching and mirroring in several
ways (figure 1):

- Distribution is done in units of content (volumes) rather
  than on part/whole sites (mirrors), or parts of a document
  (caching of a page).
- Asynchrony: changes are made by the author (production
  events), rather than on requests (caching), on periodic updates (mirroring). Distribution routes are
  optimised for distribution, which may help to improve
  the use of network bandwidth. Even connections for
  offline readers may be implemented under this model.
- Scalable: fewer HTTP post events related to change of
documents produce long haul ordered traffic, while
  caching is driven by HTTP get events. Since change events
  are much less frequent than get events, this mechanism is
  more scaleable.

![Figure 1: Distribution vs. Caching](image-url)
1 document replicated N times may be kept more consistently than 1 document with M cached fragments. Invalidation or update on changes may be made with ODS, while mirrors converge periodically and many distant caches cannot all be notified or invalidated.

- Traffic may even be more equalised, organised and automated, since users may express their interest in certain documents by subscription, and documents may be labelled under one or several categories. When labelling is done under a quality criteria by an institution, it may help to improve the quality of information perceived by clients.

3 Object Distribution System

The Object Distribution System (ODS) is formed by two independent virtual networks (figure 2): an Object Distribution Network (ODN) and an Object Routing Network (ORN). ODN brings objects close to readers according to their interests, and ORN builds the distribution chains that ODN needs to do its work in a near optimal way.

![Figure 2: Object Distribution System](image)

ODN handles objects that are persistent and replicated in every interested service agent. ODN can handle different collections of objects, either determined by their authors or by some classification authority.

An ODN is composed of a number of cooperating service agents who form several groups, or collections of objects, according to the interests of their users. In every ODN group service, agents cooperate to obtain an efficient replication inside the group, providing a selective replication of objects restricted to interested agents only. In this way we also want to put some order in the chaos caused by having information that is not classified.

ORN builds distribution chains dynamically for each group. To build the chains, the routing agents (members of ORN) take into account the type of membership to a group of each service agent and the underlying network state. Even if systems such as News or GNS distribute objects in a hierarchical manner, they do not build distribution paths dynamically.

The routing mechanisms used in ORN for building distribution chains is completely independent of the class of objects that are being handled by ODN. Both networks are designed to work independently, defining a clear mutual interface so that ORN can provide services to ODN in a transparent way.

4 MWeb

MWeb is a realization of the previous model using news, the NNTP protocol as a transport, and the mhtml format (Palme, 98) to produce document collections. There is no ODN service, and therefore the distribution chains are statically configured at present, and we are evaluating services to take into account measures and variation of bandwidth and delay in the distribution and access to contents.

The most important functions in Mweb are publications of documents (injecting documents in the replication infrastructure when they are created or modified), replication (currently using flooding and the NNTP protocol on newsgroups), and local access (transparent: caching module, non-transparent: local catalogue server, or a mhtml enabled NNTP browser).

This prototype of Mweb allows accessing locally a document store or a library of Web documents distributed in an efficient way to subscribers (frequently readers) of a specific topic or category. When a document is going to be published, an utility is used to produce a mime-mhtml message (multipart with html and graphics together) (Palme, 98). This document is posted to the local news store where it will be distributed (using the NNTP protocol). Afterwards, client will find at their local document store, copies of the latest version and probably some older version.

The prototype is composed of three tools that we have implemented to interact with the news based distribution network:

- Publication (mhtmlnews/mhtml programs)
- Proxy-Library (Apache module modified)
- Library-Catalogue (CGI program)

This separation among requests, publication and distribution, helps to optimise the use of saturated Internet links, minimising the number of copies traversing those links (only when they change). In addition, the author of a document has to send to the network just one copy of it to reach their audience.

**Publication**

The main function of a publisher is the transmission of documents at the moment they are generated. Publisher process takes URL(s) of documents in order to create mime mhtml objects (grouping text and images). This mhtml object is published in a news store (NNTP), or e-mail address (SMTP).
to be distributed. Additional mechanisms are being considered to give publishers control and accountability over their content (mechanism to ensure various aspects of copyright protection: signature, certificate, electronic payment, etc.).

**Proxy-Library**

A transparent local access to the document store: Every reader site may use their local web store to access to documents published and replicated in a given library (newsgroup) in the form of a collection of related MHTML documents. This is done using the proxy interface (a http proxy server connected to the local news store). This mechanism incorporates new functionality to an existing proxy-cache server: it will server documents from the library, from the cache or directly from the network.

The implementation of a transparent proxy to a library has been studied for two popular and publicly available servers: Squid and Apache. Both are available in source code and their internals are documented.

Due to Apaches's modular-design and the existence of mechanisms to extend its code, we chose to extend the proxy-cache module. Extensions were for retrieving mhtml documents from a local nntp server, decoding and delivering them to the user and the cache.

**Library-Catalogue**

It is a non-transparent local access to the document store: Library catalogue is a CGI tool that provides the most recent version of a document, or shows all available versions, and the content is classified in terms of meta-information.

The CGI looks for URL at the local news store. If we ask for http://www.sample.es/index.html using store.org as document store, we have the following choices:

- The most recent version:
  http://store.org/www.sample.es/index.html
- The list of available versions (figure 4):
  http://store.org/www.sample.es/index.html:
- A given version or timestamp (following the proposed extension by (Simonsen, 98)):

![Figure 3: The Mweb System](image)
Furthermore, these URLs become persistent names: It doesn’t matter if the original document disappears or moves, because we will be able to access one or more versions (specific date) of the document from our local library.

We are working to extend this CGI application in order to show the content of the news store in terms of other meta-information in a Yahoo-like catalogue.

The prototype implementation is available at http://www.canet.upc.es/mweb.

Conclusions

Web has become the “killer app” of Internet, because its explosive development and its growing use. Many strategies and techniques to improve the transmission of documents in the Web have been proposed until now, most of them based on taking advantage of read request (caching). Local proxy caching in real world experiments can reduce the access time to documents around 30%, prefetching has the potential to additional improvements. However, prefetching is only effective when the right documents are identified and when future requests are correctly predicted. Otherwise, prefetching may be yet another way to waste bandwidth. Mweb may help in some cases, for some communities to complement caching.

Some communities share a large volume of information (articles, technical reports, notes, etc.), this is frequently consulted by members which can be readers and authors. This information is valuable and is worth to keep it available locally, and classified in a catalogue. In this context we can use our distribution model to benefit from the separation of distribution (publication and document transport) from query/publication to a local digital library (Mweb + news store). Local access can be visible to the user (catalogue interface) or invisible (transparent) through a http proxy to the local document store.

The main advantages of Mweb are: nearly instantaneous access to documents (no penalty to the first reader of a document, a difference with the proxy-cache); the author has to send just one copy of a document; Mweb has the ability to equalise traffic (take advantages of hours of low traffic); it eliminates the validity checks of document (conditional HTTP GET requests), and provides URLs persistency and versioning control.

We also expect contribute with the new working group of IETF in cache and replication (WREC, 98).

References


