How much preservation do I get if I do absolutely nothing?

Using the Web Infrastructure for Digital Preservation

Martin Klein, Frank McCown, Joan A. Smith, Michael L. Nelson
{mklein, fmccown, jsmit, mln}@cs.odu.edu

31st January 2007

Abstract

To date, most of the focus regarding digital preservation has been on removing copies of the resources to be preserved from the "living web" and placing them in an archive for controlled curation. Once inside an archive, the resources are subject to careful processes of refreshing (making additional copies to new media) and migrating (conversion to new formats and applications). For small numbers of resources of known value, this is a practical and worthwhile approach to digital preservation. However, due to the infrastructure costs (storage, networks, machines) and more importantly the human management costs, this approach is unsuitable for web scale preservation. The result is that difficult decisions need to be made as to what is saved and what is not saved. We provide an overview of two of our ongoing research projects that focus using the "web infrastructure" to provide preservation capabilities for web pages. The common characteristic of the projects is they creatively employ the web infrastructure to provide shallow but broad preservation capability for all web pages. Both approaches are not intended to replace conventional archiving approaches, but rather they focus on providing at least some form of archival capability for the mass of web pages that may prove to have value in the future.

Introduction

The prevailing model for digital preservation is that archives should be similar to a "fortress": a large, protective infrastructure built to defend a relatively small collection of data from attack by external forces. Digital preservation services, such as refreshing, migration, and emulation, are provided from within the fortress. Digital preservation projects tend to focus on providing these in-depth services on limited collections of content because of the associated curatorial expenses. We refer to such projects as *in vitro* preservation because of the extensive, controlled environments necessary for their success.

Such projects are a luxury, suitable only for limited collections of known importance and requiring significant institutional commitment for sustainability.

There are, however, other possible models for digital preservation. We describes various examples of *in vivo* preservation: preservation that occurs naturally in the "living web". It is not guaranteed by an in-depth institutional commitment to a particular archive, but achieved by the often involuntary, low-fidelity, distributed efforts of millions of individual users, web administrators and commercial services. This "web infrastructure" includes search engine companies (Google, Yahoo, MSN), non-profit companies (Internet Archive, European Archive) and large-scale academic projects (CiteSeer, NSDL). Web infrastructure refreshes and migrates web content in bulk as side-effects of their user-services, and these results can be mined as a useful, but *passive* preservation service. The results for any given object might not be good, but the aggregate performance for a very large collection can be acceptable.

The WI-based preservation models we will review can be described by the level of effort required by the web administrator:

- *lazy*: reconstructing entire web sites by crawling the caches of the web infrastructure.

- *just-in-time*: trapping http 404 error responses and forwarding them to a server that uses lexical signatures to find the same or similar pages elsewhere on the web.[3, 4]

- *shared infrastructure*: web resources are replicated over the existing network protocol applications: posted as messages to special newsgroups, or attached to outgoing emails.

- *web server enhanced*: an Apache module that provides OAI-PMH access to "preservation-ready" complex object representations of web resources.[5]

In the remaining sections, we focus on the two preservation models "lazy preservation" and "shared infrastructure preservation". We will describe their concepts and implementation in detail and evaluate their performance with respect to their preservation capabilities.

Lazy Preservation

Websites may be lost for a number of reasons: hard drive crashes, file system failures, viruses, hacking, etc. A lost website may be restored if care was taken to create a backup beforehand, but sometimes webmasters are negligent in backing up their websites, and in cases such as fire, flooding, or death of the website owner, backups are frequently unavailable. In these cases, webmasters and third parties may turn to the Internet Archive (IA) "Wayback Machine" for help and although IA is often helpful, it is strictly a best-effort approach that performs sporadic, incomplete and slow crawls of the Web (IA is at least 6 months out-of-date [2]).

Another source of missing web content is in the caches of search engines (SEs) like Google, MSN and Yahoo that scour the Web looking for content to index. Unfortunately, the SEs do not preserve canonical copies of all the web resources they cache, and it is assumed that the SEs do not keep web pages long after they have been removed from a web server.

We define *lazy preservation* as the collective digital preservation performed by web archives and search engines on behalf of the Web at large. It exists as a preservation service on top of distributed, incomplete, and potentially unreliable web repositories. Lazy preservation requires no individual effort or cost for Web publishers, but it also provides no quality of service guarantees. In the remainder of this section, we explore the effectiveness of

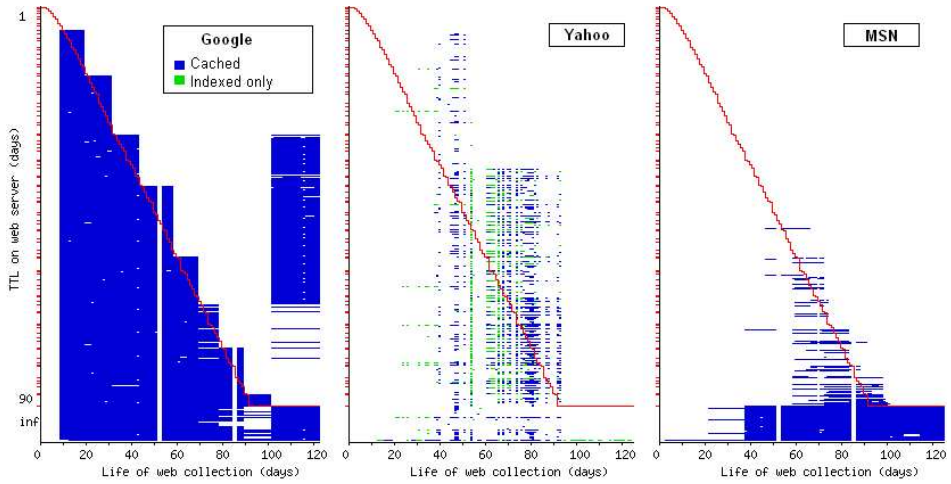


Figure 1: Caching of HTML resources from a decaying web site

lazy preservation by downloading 24 websites of various sizes and subject matter and reconstructing them using a *web-repository crawler* named Warrick[1] which recovers missing resources from four web repositories (IA, Google, MSN and Yahoo). We compare the downloaded versions of the sites with the reconstructed versions to measure how successful we were at reconstructing the websites.

The Caching Experiement

To get an idea of the SE cache longevity we measured how long resources would remain in search engine caches after the resource has been deleted [6]. We established a number of web sites with HTML files, PDFs, and images.

Figure 1 shows the cached HTML resources for one of the web sites. The red line indicates the decay of the web collection. As resources were requested that no longer resided on the web server (above the red line), the web server responded with a 404 (not found) code.

Google was by far the most active of the crawlers and cached more re-

[1]Warrick is named after a fictional forensic scientist with a penchant for gambling.

sources than the other two SEs. Google was quick to purge resources from the their cache when a crawl revealed the resources were no longer available on the web server. Yahoo performed sporadic caching of resources. Resources tended to fluctuate in and out of the Yahoo cache and index. Yahoo also did not provide complete access to all the URLs that Inktomi crawled. Although Inktomi crawled nearly every available HTML resource on day 10, only half of those resources ever became available in the Yahoo cache. There is also a lag time of about 30 days between Inktomi crawling a resource and the resource appearing in the Yahoo cache. MSN was very slow to crawl the resources in the update bins. After day 40 they began to crawl some of the resources and make a small number of them available in their cache. Like Google, MSN was quick to remove 404 resources from their cache. For the interested reader, details to that experiment can be found in [9] and in [7].

Reconstructing Websites

Warrick, our web-repository crawler, is able to reconstruct a website when given a base URL pointing to where the site used to exist. The web repositories are crawled by issuing queries in the form of URLs to access their stored holdings. For example, Google's cached version of `http://foo.edu/page1.html` can be accessed like so: `http://search.google.com/search?q=cache:http://foo.edu/page1.html`. If Google has not cached the page, an error page will be generated. Otherwise the cached page can be stripped of any Google-added HTML, and the page can be parsed for links to other resources from the foo.edu domain (and other domains if necessary). Most repositories require two or more queries to obtain a resource. For each URL, the file extension (if present) is examined to determine if the URL is an image (.png, .gif, .jpg, etc.) or other resource type. All three SEs use a different method for retrieving images than for other resource types. IA has the same interface regardless of the type. We would have better accuracy at determining if a given URL referenced an image or not if we knew the URL's resource

MIME type, but this information is not available to us.

IA is the first web repository queried by Warrick because it keeps a canonical version of all web resources. When querying for an image URL, if IA does not have the image then Google and Yahoo are queried one at a time until one of them returns an image. Google and Yahoo do not publicize the cached date of their images, so it is not possible to pick the most recently cached image. If a non-image resource is being retrieved, again IA is queried first. If IA has the resource and the resource does not have a MIME type of 'text/html', then the SEs are not queried since they only store canonical versions of HTML resources. If the resource does have a 'text/html' MIME type (or IA did not have a copy), then all three SEs are queried, the cache dates of the resources are compared (if available), and the most recent resource is chosen. Warrick will search HTML resources for URLs to other resources and add them to the crawl frontier (a queue). Resources are recovered in breadth-first order, and reconstruction continues until the frontier is empty. All recovered resources are stored on the local filesystem, and a log is kept of recovered and missing resources.

Evaluation

We have constructed a web-repository crawler named Warrick which can automatically reconstruct lost website by recovering resources from four web repositories: Internet Archive, Google, MSN and Yahoo [7, 8, 9].

To evaluate the effectiveness of website reconstruction from the WI, we conducted an experiment using Warrick to reconstruct 24 live websites of various structure and size. We reconstructed live websites in order to precisely measure the percentage of resources that were and were not recovered and to compare the degree of change, if any, of the recovered resources from their live counterparts. On average we were able to recover 68% of the website resources. For a quarter of the 24 sites, we were able to recover more than 90% of the resources.

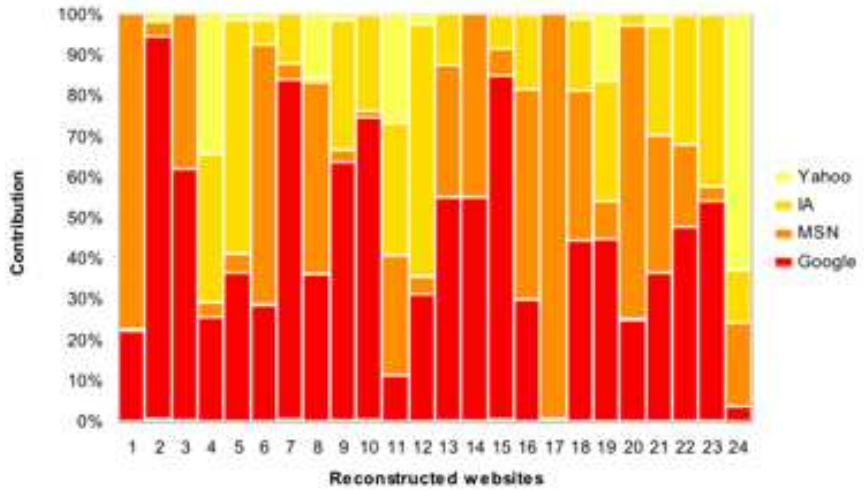


Figure 2: Web repositories contributing to each website reconstruction

The majority of the resources in the 24 websites that were reconstructed were originally composed of HTML and images and we were much more successful at recovering HTML resources than any other MIME type. We also found that when we reconstructed the 24 websites using each web repository by itself, none of them performed as well individually as they did when used together.

As illustrated in Figure 2, some repositories were more helpful than others depending on which website was being reconstructed. For example, all four repositories contributed nearly the same percentage of resources when reconstructing site 11, but MSN was the sole contributor for site 17. Although Google was the largest overall contributor to the website reconstructions (providing 44% of the resources), the breadth of the Web is too large for any single repository to provide the only layer of lazy preservation. For the interested reader we refer to [8] and [9] for a detailed description of the experiment and an evaluation of the results.

Warrick has been made freely available on the Web, and it has been endorsed by the Internet Archive. It has been used to reconstruct websites

lost due to fire, hacking, hard drive crashes, death of website owners, and discontinued charitable web hosting [8]. Although Warrick is not able to always recover all missing resources, users have been thankful to retrieve even a portion of what could have been permanently lost for all time.

Shared Infrastructure Preservation

In this section we focus on repository replication using *shared, existing* infrastructure. Our goal is not to "hijack" other sites' storage, but to take advantage of protocols which have persisted through many generations and which are likely to be supported well into the future. The premise is that if archiving can be accomplished within a widely-used, already deployed infrastructure whose operational burden is shared among many partners, the resulting system will have only an incremental cost and be tolerant of dynamic participation. With this in mind, we examine the feasibility of repository replication using Usenet news (NNTP, [10]) and email (SMTP, [11]).

There are reasons to believe that both email and Usenet could function as persistent, if diffuse, archives. NNTP provides well-understood methods for content distribution and duplicate deletion (deduping) while supporting a distributed and dynamic membership. The long-term persistence of news messages is evident in "Google Groups," a Usenet archive with posts dating from May 1981 to the present ([12]). Even though blogs and bulletin boards have supplanted Usenet in recent years, many communities still actively use moderated news groups for discussion and awareness. Although email is not usually publicly archivable, it is ubiquitous and frequent. For example, our departmental SMTP email server averaged over 16,000 daily outbound emails to more than 4000 unique recipient servers during a 30-day test period. Unlike Usenet, email is point-to-point communication but, given enough time, attaching repository contents to outbound emails may prove to be an effective way to disseminate contents to previously unknown locations. The open

source products for news ("INN") and email ("sendmail" and "postfix") are widely installed, so including a preservation function would not impose a significant additional administrative burden. In summary, although SMTP and NNTP are not the "right" tools for digital preservation, their ubiquity requires them be studied in a preservation context. For example, who has not at some time emailed themselves a file so as to "not lose it"?

Archiving Policies Using NNTP and SMTP

Figure 3 illustrates the policies of the news method for repository replication. A "baseline," refers to making a complete snapshot of a repository. A "cyclic baseline" is the process of repeating the snapshot over and over again, which may result in the receiver storing more than one copy of the repository. Of course, most repositories are not static. Repeating baselines will capture new additions and updates with each new baseline. The process could also "sleep" between baselines, sending only changed content. In short, the changing nature of the repository can be accounted for when defining its replication policies. A baseline, whether it is cyclic or one-time-only, should finish before the end of the news server message life, or a complete snapshot will not be achieved. The time to complete a baseline using news is obviously constrained by the size of the repository and the speed of the network. Picking the best posting strategy is not straightforward because we do not know the archiving policy of all the recipients news sites. For sites that have small buffers, we would like to use either cyclic baseline or cyclic baseline with updates to make sure the remote news server has as much of the repository as possible. But for news sites with no deletion (e.g. Google Groups), the policy of single baseline with updates is ideal.

One major difference in using email as the archiving target instead of news is that it is passive, not active: the email process relies on *existing* traffic between the archiving site and one or more target destination sites. It passively waits for existing email traffic and then "hitches a ride" to the

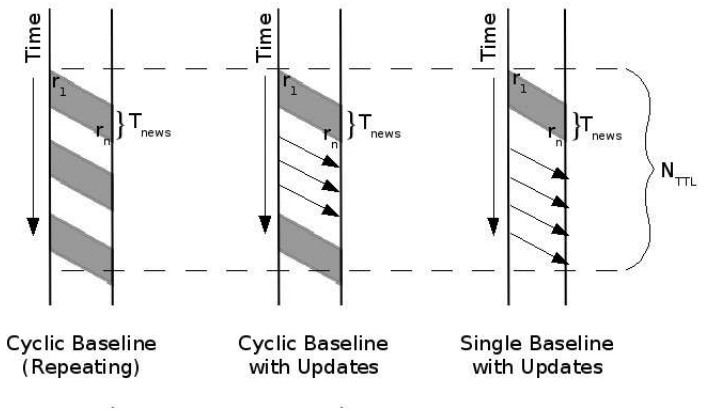
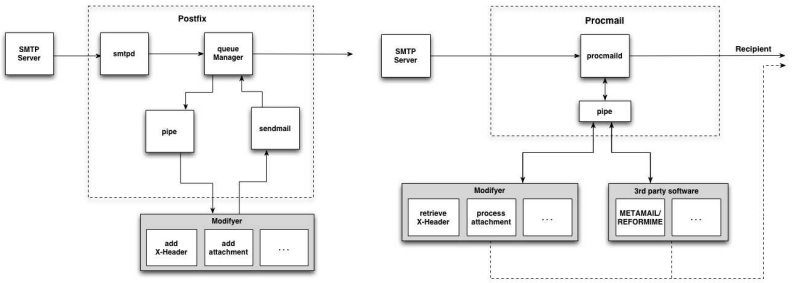


Figure 3: NNTP Timeline for Sender & Receiver Policies

destination host. We are able to attach files automatically with just a small processing delay penalty. Processing options include selecting only every E^{th} email, a factor we call "granularity" [13]; randomly selecting records to process instead of a specific ordering; and/or maintaining replication lists for each destination site. Completing a baseline using email is subject to the same constraints as news - repository size, number of records, etc. - but is particularly sensitive to changes in email volume. For example, holidays are often used for administrative tasks since they are typically "slow" periods, but there is little email generated during holidays so repository replication would be slowed rather than accelerated. However, the large number of unique destination hosts means that email is well adapted to repository discovery through advertising. The techniques used to trap incoming and outgoing messages are shown in Figure 4.

Simulation

We ran a simulation of a repository that starts with 1000 resources, adds 100 new resources each day and updates 20 resources per day. Although it would be reasonable to expect the high rate of change to slow over time as the repository matures, we maintained this high activity level throughout



(a) Outbound Mail (b) Inbound Mail

Figure 4: Archiving Using SMTP

the 2000 days of the simulation.

We found that despite the high activity rate, both the cyclic baseline and the continuous baseline policies manage to keep up with the job of replication for the entire simulation period. The news server retains at least one full copy of the repository for the entire time frame and at its peak it maintains three full copies of the repository.

The performance for the same growing repository using the SMTP method is less promising. If we keep track of which resources we have sent to which sites we found that receiver domains up to rank 10 receive enough emails to maintain a full copy of the repository. The "rank" is the popularity of the receiving domains: the rank 2 domain receives far more email than rank 10 site (the email distribution follows a power law distribution). The results are worse without maintaining that history list. Again, for the interested reader we refer to [14, 13].

The results of the simulation indicate that for active, large repositories, most sites will not have enough email traffic to keep up with the growth of the repository: only the highest few ranks can keep up with the growth of the repository. The SMTP approach is not feasible in nearly all cases. But the Usenet approach is effective in keeping multiple copies of the repository

on remote news servers, some which never expire the news articles.

Conclusion

We have reviewed two out of our four web page preservation models that make use of the actions of millions of users, web administrators, commercial services and research projects. These models are in various levels of maturity and include doing nothing ("lazy preservation"), trying to find suitable replacement pages after they've been lost ("just-in-time preservation"), injecting resources directly into the WI ("shared infrastructure preservation"), and installing an Apache module to facilitate better discovery and description of resources to the WI.

Instead of the "deep infrastructure" for digital preservation as first discussed in the seminal RLG task force report [1], will we get a very broad but relatively shallow infrastructure for preservation? With the exception of the web server enhanced model, none of the other preservation models have more than a trivial description within the Open Archival Information System (OAIS) framework [15]. Commercial search engines bypassed the traditional metadata vs. data constructs; the same thing could happen with preservation.

Our initial results indicate that the WI is good at refreshing resources and allowing them to be recovered several months after the original resource was lost. The WI is also providing tentative first steps in migrating formats. The results are somewhat crude and heavy handed, but we believe the functionality will improve as users begin to request this functionality. We have yet to see the WI tackle emulation, but this could change in the future as commercial search engines encroach on the OS and desktop environments. The rise of archival functions in social bookmarking tools are also an indication of the growing general interest in preserving digital resources. We believe it is only a matter of time before the commercial search engines develop a business

model for preservation and begin to shape the discussion about web-scale preservation. It may in fact be possible to "save everything".

Acknowledgements:
Johan Bollen (Los Alamos National Laboratory) contributed to the initial development of the "lazy" and "just-in-time" preservation models. Aravind Elango, Terry Harrison, Ignacio Campo del Garcia (Old Dominion University), Xiaoming Liu and Herbert Van de Sompel (Los Alamos National Laboratory) all contributed to the "web server enhanced" preservation model. Terry Harrison also contributed to the "just-in-time" preservation model.

Bibliography

[1] RLG: Preserving digital information: Report of the task force on archiving of digital information. *http://www.rlg.org/ArchTF/ (1996)*

[2] Internet Archive FAQ: How can I get my site included in the Archive?, 2006. `http://www.archive.org/about/faqs.php`.

[3] Terry L. Harrison: *Opal: In Vivo Based Preservation Framework for Locating Lost* Old Dominion University, 2005 `http://www.cs.odu.edu/~tharriso/thesis/`

[4] Terry L. Harrison and Michael L. Nelson *Just-In-Time Recovery of Missing Web Pages* HYPERTEXT 2006: Proceedings of the seventeenth ACM conference on Hypertext and hypermedia (2006) Odense, Denmark, p.145-156

[5] Michael L. Nelson and Joan A. Smith and Ignacio Garcia del Campo and Herbert Van de Sompel and Xiaoming Liu *Efficient, Automatic Web Resource Harvesting* WIDM '06: Proceedings of the 8th annual ACM international workshop on Web information and data management 2006, Arlington, Virginia, p.43-50

[6] Joan A. Smith and Frank McCown and Michael L. Nelson *Observed web robot behavior on decaying web subsites.* D-Lib Magazine 12(2) (2006). DOI doi:10.1045/february2006-smith

[7] Frank McCown and Joan A. Smith and Michael L. Nelson and Johan Bollen *Reconstructing websites for the lazy webmaster.* Tech. Rep. arXiv cs.IR/0512069 (2005). `http://arxiv.org/abs/cs.IR/0512069`

[8] Frank McCown and Michael L. Nelson *Evaluation of crawler policies for a web-repository crawler.* HYPERTEXT 2006: Proceedings of the seventeenth ACM conference on Hypertext and hypermedia (2006)

[9] Frank McCown and Joan A. Smith and Michael L. Nelson and Johan Bollen *Lazy Preservation: Reconstructing Websites by Crawling the Crawlers* WIDM 2006: Proceedings of the 8th annual ACM international workshop on Web information and data management 2006, Arlington, Virginia, p.67-74

[10] Brian Kantor and Phil Lapsley *Network News Transfer Protocol, Internet RFC-977* February, 1986

[11] Jonathan B. Postel *Simple Mail Transfer Protocol, Internet RFC-821* August, 1982

[12] 20 Year archive on Google groups *Google, Inc.* `http://www.google.com/googlegroups/archive_announce_20.html`

[13] Joan A. Smith and Martin Klein and Michael L. Nelson *Repository replication using NNTP and SMTP* Old Dominion University, 2006, Tech. Rep. arXiv cs.DL/0606008 `http://arxiv.org/abs/cs/0606008`

[14] Joan A. Smith and Martin Klein and Michael L. Nelson *Repository replication using NNTP and SMTP* ECDL 2006: Proceedings of the 10th European Conference on Research and Advanced Technology for Digital Libraries 2006, Alicante, Spain, p.51-62

[15] Consultative Committee for Space Data Systems: *Reference model for an open archival information system (OAIS)* Tech. rep. (2002)