Lazy Preservation: Reconstructing Websites by Crawling the Crawlers

Frank McCown, Joan A. Smith, and Michael L. Nelson
Old Dominion University
Computer Science Department
Norfolk, Virginia, USA 23529
{fmccown, jsmit, mln}@cs.odu.edu

Johan Bollen
Los Alamos National Laboratory
Digital Library Research & Prototyping Team
Los Alamos, New Mexico, USA 87545
jbollen@lanl.gov

ABSTRACT

Backup of websites is often not considered until after a catastrophic event has occurred to either the website or its webmaster. We introduce "lazy preservation" – digital preservation performed as a result of the normal operation of web crawlers and caches. Lazy preservation is especially suitable for third parties; for example, a teacher reconstructing a missing website used in previous classes. We evaluate the effectiveness of lazy preservation by reconstructing 24 websites of varying sizes and composition using Warrick, a web-repository crawler. Because of varying levels of completeness in any one repository, our reconstructions sampled from four different web repositories: Google (44%), MSN (30%), Internet Archive (19%) and Yahoo (7%). We also measured the time required for web resources to be discovered and cached (10-103 days) as well as how long they remained in cache after deletion (7-61 days).

Categories and Subject Descriptors: H.3.5 [Information Storage and Retrieval] Online Information Services: Web-based services; H.3.7 [Information Storage and Retrieval] Digital Libraries: Collection

General Terms: Measurement, Experimentation, Design

Keywords: digital preservation, search engine, cached resources, recovery

1. INTRODUCTION

> *"My old web hosting company lost my site in its entirety (duh!) when a hard drive died on them. Needless to say that I was peeved, but I do notice that it is available to browse on the wayback machine... Does anyone have any ideas if I can download my full site?"* - A request for help at archive.org [25]

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. According to a representative from IA, they have performed over 200 website recoveries in the past year for various individuals. 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 [16]).

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. We explore the effectiveness of 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.

We also measure the time it takes for SEs to crawl and cache web pages that we have created on .com and .edu websites. In June 2005, we created four synthetic web collections consisting of HTML, PDF and images. For 90 days we systematically removed web pages and measured how long they remained cached by the SEs.

2. BACKGROUND AND RELATED WORK

The ephemeral nature of the Web has been widely acknowledged. To combat the disappearance of web resources, Brewster Kahle's Internet Archive has been archiving the

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

Table 1: Web repository-supported data types

Type	G	Y	M	IA
HTML	C	C	C	C
Plain text	M	M	M	C
GIF, PNG, JPG	M	M	∼R	C
JavaScript	M		M	C
MS Excel	M	∼S	M	C
MS PowerPoint	M	M	M	C
MS Word	M	M	M	C
PDF	M	M	M	C
PostScript	M	∼S		C

C = Canonical version is stored
M = Modified version is stored (image thumbnails or HTML conversions)
∼R = Stored but not retrievable with direct URL
∼S = Indexed but stored version is not accessible

Web since 1996 [4]. National libraries are also actively engaged in archiving culturally important websites [8]. Systems like LOCKSS [24] have been developed to ensure libraries have long-term access to publishers' web content, and commercial systems like Spurl.net and HanzoWeb.com have been developed to allow users to archive selected web resources that they deem important.

Other researchers have developed tools for archiving individual websites and web pages. InfoMonitor [7] archives a website's file system and stores the archive remotely. TTApache [9] is used to archive requested pages from a particular web server, and iPROXY [23] is used as a proxy server to archive requested pages from a variety of web servers. In many cases these services can be of some value for recovering a lost website, but they are largely useless when backups are inaccessible or destroyed or when a third party wants to reconstruct a website. They also require the webmaster to perform some amount of work in setting up, configuring and monitoring the systems.

In regards to commercial search engines, the literature has mostly focused on measuring the amount of content they have indexed (e.g., [15, 18]), relevance of responses to users' queries (e.g., [5, 14]), and ranking of pages (e.g., [28]). Lewandowski et al. [17] studied how frequently Google, MSN and Yahoo updated their cached versions of web pages, but we are unaware of any research that attempts to measure how quickly new resources are added to and removed from commercial SE caches, or research that explores the use of SE caches for reconstructing websites.

3. WEB CRAWLING AND CACHING

3.1 Web Repositories

There are many SEs and web archives that index and store Web content. For them to be useful for website reconstruction, they must at a minimum provide a way to map a given URL to a stored resource. To limit the implementation complexity, we have focused on what we consider to be the four most popular web repositories that meet our minimum criteria. Recent measurements show that Google, MSN and Yahoo index significantly different portions of the Web and have an intersection of less than 45% [15]. Adding additional web repositories like ask.com, gigablast.com, incywincy.com and any other web repository that allows direct URL retrieval would likely increase our ability to reconstruct websites.

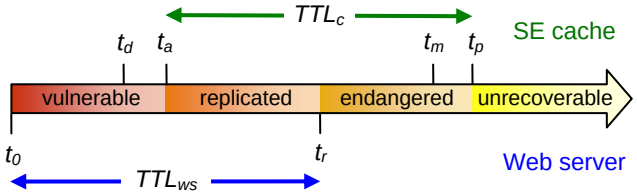

Figure 1: Timeline of SE resource acquisition and release

Although SEs often publish index size estimates, it is difficult to estimate the number of resources in each SE cache. An HTML web page may consist of numerous web resources (e.g., images, applets, etc.) that may not be counted in the estimates, and not all indexed resources are stored in the SE caches. Google, MSN and Yahoo will not cache an HTML page if it contains a NOARCHIVE meta-tag, and the http Cache-control directives 'no-cache' and 'no-store' may also prevent caching of resources [1].

Only IA stores web resources indefinitely. The SEs have proprietary cache replacement and removal policies which can only be inferred from observed behavior. All four web repositories perform sporadic and incomplete crawls of websites making their aggregate performance important for website reconstruction.

Table 1 shows the most popular types of resources held by the four web repositories. This table is based on our observations when reconstructing websites with a variety of content. IA keeps a canonical version of all web resources, but SEs only keep canonical versions of HTML pages. When adding PDF, PostScript and Microsoft Office (Word, Excel, PowerPoint) resources to their cache, the SEs create HTML versions of the resources which are stripped of all images. SEs also keep only a thumbnail version of the images they cache due to copyright law. MSN uses Picsearch for their image crawling; unfortunately, Picsearch and MSN do not support direct URL queries for accessing these images, so they cannot be used for recovering website images.

3.2 Lifetime of a Web Resource

Figure 1 illustrates the life span of a web resource from when it is first made available on a web server to when when it is finally purged from a SE cache. A web resource's time-to-live on the web server (TTL_{ws}) is defined as the number of days from when the resource is first made accessible on the server (t_0) to when it is removed (t_r).

A new resource is *vulnerable* until it is discovered by a SE (t_d) and made available in the SE cache (t_a). The resource is *replicated* when it is accessible on the web server and in cache. Once the resource is removed from the web server (t_r), it becomes *endangered* since it is only accessible in cache. When a subsequent crawl reveals the resource is no longer available on the web server (t_m), it will then be purged from cache (t_p) and be *unrecoverable*. The period between t_a and t_p define a resource's time-to-live in the SE cache (TTL_c). A resource is *recoverable* if it is currently cached (i.e., is replicated or endangered). A recoverable resource can only be recovered during the TTL_c period with a probability of P_r, the observed number of days that a resource is retrievable from the cache divided by TTL_c.

It should be noted that the TTL_{ws} and TTL_c values of a resource may not necessarily overlap. A SE that is trying to maximize the freshness of its index will try to minimize the difference between TTL_{ws} and TTL_c. A SE that is slow in updating its index, perhaps because it obtains crawling data from a third party, may experience late caching where $t_r < t_a$.

For a website to be lazily preserved, we would like its resources to be cached soon after their appearance on a website (have minimal vulnerability). SEs may also share this goal if they want to index newly discovered content as quickly as possible. Inducing a SE to crawl a website at a specific time is not currently possible. Webmasters may employ various techniques to ensure their websites are crawler-friendly [13, 27] and well connected to the Web. They may even submit their website URLs to SEs or use proprietary mechanisms like Google's Sitemap Protocol [12], but no technique will guarantee immediate indexing and caching of a website.

We would also like resources to remain cached long after they have been deleted from the web server (remain endangered) so they can be recovered for many days after their disappearance. SEs on the other hand may want to minimize the endangered period in order to purge missing content from their index. Just as we have no control as to when a SE crawler will visit, we also have no control over cache eviction policies.

3.3 Web Collection Design

In order to obtain measurements for TTL_c and other values in Figure 1, we created four synthetic web collections and placed them on websites for which we could obtain crawling data. We deployed the collections in June 2005 at four different locations: 1) `www.owenbrau.com`, 2) `www.cs.odu.edu/~fmccown/lazy/` 3) `www.cs.odu.edu/~jsmit/`, and 4) `www.cs.odu.edu/~mln/lazyp/`. The .com website was new and had never been indexed by Google, Yahoo or MSN. The 3 .edu websites had existed for over a year and had been previously crawled by all three SEs. In order for the web collections to be found by the SEs, we placed links to the root of each web collection from the .edu websites, and we submitted owenbrau's base URL to Google, MSN and Yahoo 1 month prior to the experiment. For 90 days we systematically removed resources from each collection. We examined the server web logs to determine when resources were crawled, and we queried Google, MSN and Yahoo daily to determine when the resources were cached.

We organized each web collection into a series of 30 *update bins* (directories) which contained a number of HTML pages referencing the same three inline images (GIF, JPG and PNG) and a number of PDF files. An index.html file (with a single inline image) in the root of the web collection pointed to each of the bins. An index.html file in each bin pointed to the HTML pages and PDF files so a web crawler could easily find all the resources. All these files were static and did not change throughout the 90 day period except the index.html files in each bin which were modified when links to deleted web pages were removed. In all, there were 381 HTML files, 350 PDF files, and 223 images in each web collection. More detail about the organization of the web collections and what the pages and images looked like can be found in [20, 26].

The PDF and HTML pages were made to look like typical web pages with around 120 words per page. The text for each page was randomly generated from a standard English dictionary. By using random words we avoided creating duplicate pages that a SE may reject [6]. Unfortunately, using random words may cause pages to be flagged as spam [10].

Each HTML and PDF page contained a unique identifier (UID) at the top of each page (e.g., 'mlnODULPT2 dgrp18 pg18-2-pdf' that included 4 identifiers: the web collection (e.g., 'mlnODULPT2' means the 'mln' collection), bin number (e.g., 'dgrp18' means bin 18), page number and resource type (e.g., 'pg18-2-pdf' means page number 2 from bin 18 and PDF resource). The UID contains spaces to allow for more efficient querying of the SE caches.

The TTL_{ws} for each resource in the web collection is a function of its bin number b and page number p:

$$TTL_{ws} = b(\lfloor 90/b \rfloor - p + 1) \quad (1)$$

3.4 Daily SE Queries

In designing our daily SE queries, care was taken to perform a limited number of daily queries to not overburden the SEs. We could have queried the SEs using the URL for each resource, but this might have led to our resources being cached prematurely; it is possible that if a SE is queried for a URL it did not index that it would add the URL to a list of URLs to be crawled at a later date. This is how IA's advanced search interface handles missing URLs from users' queries.

To determine which HTML and PDF resources had been cached, we queried using subsets of the resources' UIDs and looked for cached URLs in the results pages. For example, to find PDF resources from the mln collection, we queried each SE to return the top 100 PDF results from the site www.cs.odu.edu that contain the exact phrase 'mlnODULPT2 dgrp18'.[2] It is necessary to divulge the site in the query or multiple results from the site will not be returned. Although this tells the SE on which site the resource is located, it does not divulge the URL of the resource. To query for cached images, we queried for the globally unique filename given to each image.

3.5 Crawling and Caching Observations

Although the web server logs registered visits from a variety of crawlers, we report only on crawls from Google, Inktomi (Yahoo) and MSN.[3] Alexa Internet (who provides crawls to IA) only accessed our collection once (induced through our use of the Alexa toolbar). A separate IA robot accessed less than 1% of the collections, likely due to several submissions we made to their Wayback Machine's advanced search interface early in the experiment. Further analysis of the log data can seen in a companion paper [26].

We report only detailed measurements on HTML resources (PDF resources were similar). Images were crawled and cached far less frequently; Google and Picsearch (the MSN Images provider) were the only ones to crawl a significant number of images. The 3 .edu collections had 29% of their images crawled, and owenbrau had 14% of its images crawled. Only 4 unique images appeared in Google Images, all from

[2]MSN only allows limiting the results page to 50.

[3]Due to a technical mishap beyond our control, we were unable to obtain crawling data for days 41-55 for owebrau and parts of days 66-75 and 97 for the .edu web collections. We were also prevented from making cache queries on days 53, 54, 86 and 87.

Table 2: Caching of HTML resources from 4 web collections (350 HTML resources in each collection)

Web collection	% URLs crawled G	M	Y	% URLs cached G	M	Y	t_{ca} G	M	Y	TTL_c / P_r G	M	Y	Endangered G	M	Y
fmccown	91	41	56	91	16	36	13	65	47	90 / 0.78	20 / 0.87	35 / 0.57	51	9	24
jsmit	92	31	92	92	14	65	12	66	47	86 / 0.82	20 / 0.91	36 / 0.55	47	7	25
mln	94	33	84	94	14	49	10	65	54	87 / 0.83	21 / 0.90	24 / 0.46	47	8	19
owenbrau	18	0	0	20	0	0	103	N/A	N/A	40 / 0.98	N/A	N/A	61	N/A	N/A
Ave	74	26	58	74	11	37	35	66	50	76 / 0.86	20 / 0.89	32 / 0.53	51	8	23

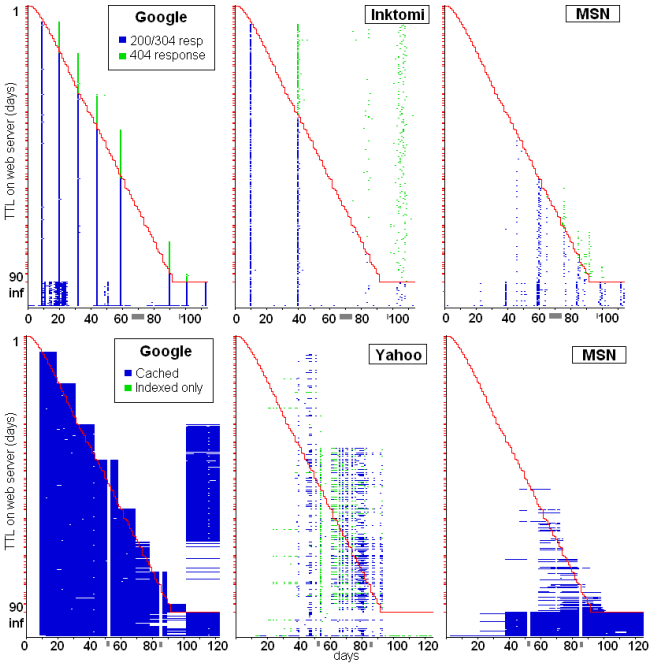

Figure 2: Crawling (top) and caching (bottom) of HTML resources from the mln web collection

the mln collection. Google likely used an image duplication detection algorithm to prevent duplicate images from different URLs from being cached. Only one image (from fmccown) appeared in MSN Images. None of the cached images fell out of cache during our experiment.

Table 2 summarizes the performance of each SE to crawl and cache 350 HTML resources from each of the four web collections. This table does not include index.html resources which had an infinite TTL_{ws}. We believe there was an error in the MSN query script which caused fewer resources to be found in the MSN cache, but the percentage of crawled URLs provides an upper bound on the number of cached resources; this has little to no effect on the other measurements reported.

The three SEs showed equal desire to crawl HTML and PDF resources. Inktomi (Yahoo) crawled 2 times as many resources as MSN, and Google crawled almost 3 times as many resources than MSN. Google was the only SE to crawl and cache any resources from the new owenbrau website.

From a preservation perspective, Google out-performed MSN and Yahoo in nearly every category. Google cached the highest percentage of HTML resources (76%) and took only 12 days on average to cache new resources from the edu web collections. On average, Google cached HTML resources for the longest period of time (76 days), consistently provided access to the cached resources (86%), and were the slowest to remove cached resources that were deleted from the web server (51 days). Although Yahoo cached more HTML resources and kept the resources cached for a longer period than MSN, the probability of accessing a resource on any given day was only 53% compared to 89% for MSN.

Figure 2 provides an interesting look at the crawling and caching behavior of Google, Yahoo and MSN. These graphs illustrate the crawling and caching of HTML resources from the mln collection; the other two edu collections exhibited similar behavior. The resources are sorted by TTL_{ws} with the longest-living resources appearing on the bottom. The index.html files which were never removed from the web collection have an infinite TTL ('inf'). The red diagonal line indicates the decay of the web collection; on any particular day, only resources below the red line were accessible from the web server. On the top row of Figure 2, blue dots indicate resources that were crawled on a particular day. When resources were requested that had been deleted, the web server responded with a 404 (not found) code represented by green dots above the red line. The bottom row of Figure 2 shows the cached HTML resources (blue) resulting from the crawls. Some pages in Yahoo were indexed but not cached (green).

As Figure 2 illustrates, both Google and MSN were quick to make resources available in their cache soon after they were crawled, and they were quick to purge resources from their cache when a crawl revealed the resources were no longer available on the web server. A surprising finding is that many of the HTML resources that were previously purged from Google's cache reappeared on day 102 and remained cached for the remainder of our experiment. The other two edu collections exhibited similar behavior for HTML resources. HTML and PDF resources from owenbrau appeared in the Google cache on day 102 for the first time; these resources had been deleted from the web server 10-20 days before day 102. Manual inspection weeks after the experiment had concluded revealed that the pages remained in Google's cache and fell out months later.

Yahoo was very sporadic in caching resources; there was often a lag time of 30 days between the crawl of a resource and its appearance in cache. Many of the crawled resources never appeared in Yahoo's cache. Although Inktomi crawled nearly every available HTML resource on day 10, only half of those resources ever became available in the Yahoo cache. We have observed through subsequent interaction with Yahoo that links to cached content may appear and disappear when performing the same query just a few seconds apart. This likely accounts for the observed cache inconsistency.

We have observed from our measurements that nearly all new HTML and PDF resources that we placed on known websites were crawled and cached by Google several days af-

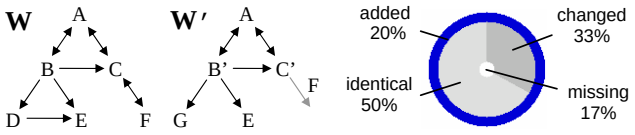

Figure 3: Lost website (left), reconstructed website (center), and reconstruction diagram (right)

ter they were discovered. Resources on a new website were not cached for months. Yahoo and MSN were 4-5 times slower than Google to acquire new resources, and Yahoo incurs a long transfer delay from Inktomi's crawls into their cache. We have also observed that cached resources are often purged from all three caches as soon as a crawl reveals the resources are missing, but in the case of Google, many HTML resources have reappeared weeks after being removed. Images tend to be largely ignored.

Search engines may crawl and cache other websites differently depending on a variety of factors including perceived level of importance (e.g., PageRank) and modification rates. Crawling policies may also be changed over time. This experiment merely provides a glimpse into the current caching behavior of the top three SEs that has not been documented before. Our findings suggest that SEs vary greatly in the level of access they provide to cached resources, and that websites are likely to be reconstructed more successfully if they are reconstructed quickly after being lost. Reconstructions should also be performed several days in a row to ensure maximum access to web repository holdings. In some cases, it may even be beneficial to attempt recovering resources even a month after they have been lost.

4. RECONSTRUCTING WEBSITES

4.1 Reconstruction Measurements

We define a **reconstructed website** to be the collection of recovered resources that share the same URIs as the resources from a lost website or from some previous version of the lost website [19]. The recovered resources may be equivalent to, or very different from, the lost resources. For websites that are composed of static files, recovered resources would be equivalent to the files that were lost. For sites produced dynamically using CGI, PHP, etc., the recovered resources would match the client's view of the resources and would be useful to the webmaster in rebuilding the server-side components. The server-side components are currently not recoverable using lazy preservation (see Section 5).

To quantify the difference between a reconstructed website and a lost website, we classify the recovered resources from the website graphs. A website can be represented as a graph $G = (V, E)$ where each resource r_i (HTML, PDF, image, etc.), identified by a URI, is a node v_i, and there exists a directed edge from v_i to v_j when there is a hyperlink or reference from r_i to r_j. The left side of Figure 3 shows a web graph for some website W if we began to crawl it starting at A. Suppose W was lost and reconstructed forming the website W' represented in the center of Figure 3.

For each resource r_i in W we may examine its corresponding resource r'_i in W' that shares the same URI and categorize r'_i as *identical* (r'_i is byte-for-byte identical to r_i), *changed* (r'_i is not identical to r_i), or *missing* (r'_i could not be found in any web). We would categorize those resources in W' that did not share a URI with any resource in W as *added* (r'_i was not a part of the current website but was recovered due to a reference from r'_j).

Figure 3 shows that resources A, G and E were reconstructed and are identical to their lost versions. An older version of B was found (B') that pointed to G, a resource that does not currently exist in W. Since B' does not reference D, we did not know to recover it (it is possible that G is actually D renamed). An older version of C was found, and although it still references F, F could not be found in any web repository.

A measure of change between the lost website W and the reconstructed website W' can be described using the following **difference vector**:

$$\text{difference}(W, W') = \left(\frac{R_{changed}}{|W|}, \frac{R_{missing}}{|W|}, \frac{R_{added}}{|W'|} \right) \quad (2)$$

For Figure 3, the difference vector is (2/6, 1/6, 1/5) = (0.333, 0.167, 0.2). The best case scenario would be (0,0,0), the complete reconstruction of a website. A completely unrecoverable website would have a difference vector of (0,1,0).

The difference vector for a reconstructed website can be illustrated as a **reconstruction diagram** as shown on the right side of Figure 3. The changed, identical and missing resources form the core of the reconstructed website. The dark gray portion of the core grows as the percentage of changed resource increases. The hole in the center of the core grows as the percentage of missing resources increases. The added resources appear as crust around the core. This representation will be used later in Table 3 when we report on the websites we reconstructed in our experiments.

4.2 Warrick Operation

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.

Table 3: Results of website reconstructions

Website	PR	MIME type groupings (orig/recovered) Total	HTML	Images	Other	Difference vector (Changed, Missing, Added)	Recon diag	Almost identical	New recon diag
1. www.eskimo.com/~scs/	6	719/691 96%	696/669 96%	22/21 95%	1/1 100%	(0.011, 0.039, 0.001)		50%	
2. www.digitalpreservation.gov	8	414/378 91%	346/329 95%	42/25 60%	26/24 92%	(0.097, 0.087, 0.000)		44%	
3. www.harding.edu/hr/	4	73/47 64%	19/19 100%	25/2 8%	29/26 90%	(0.438, 0.356, 0.145)		83%	
4. www.techlocker.com	4	1216/406 33%	687/149 22%	529/257 49%	0/0	(0.267, 0.666, 0.175)		99%	

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. Warrick limits its requests per day to the web repositories based on their published API values (Google, 1000; Yahoo, 5000; MSN, 10,000) or lacking an API, our best guess (IA, 1000). If any repository's limit is exceeded, Warrick will checkpoint and sleep for 24 hours.

4.3 Reconstruction Experiment and Results

To gauge the effectiveness of lazy preservation for website reconstruction, we compared the snap-shot of 24 live websites with their reconstructions. We chose sites that were either personally known to us or randomly sampled from dmoz.org. The websites (some were actually subsites) were predominantly English, covered a range of topics, and were from a number of top-level domains. We chose 8 small (<150 URIs), 8 medium (150-499 URIs) and 8 large (≥500 URIs) websites, and we avoided websites that used robots.txt and Flash exclusively as the main interface.

In August 2005 we downloaded all 24 websites by starting at the base URL and following all links and references that that were in and beneath the starting directory, with no limit to the path depth. For simplicity, we restricted the download to port 80 and did not follow links to other hosts within the same domain name. So if the base URL for the website was `http://www.foo.edu/bar/`, only URLs matching `http://www.foo.edu/bar/*` were downloaded. Warrick uses the same default setting for reconstructing websites.

Immediately after downloading the websites, we reconstructed five different versions for each of the 24 websites: four using each web repository separately, and one using all web repositories together. The different reconstructions helped to show how effective individual web repositories could reconstruct a website versus the aggregate of all four web repositories.

We present 4 of the 24 results of the aggregate reconstructions in Table 3, ordered by percent of recovered URIs. The complete results can be seen in [20]. The 'PR' column is Google's PageRank (0-10 with 10 being the most important) for the root page of each website at the time of the experiments. (MSN and Yahoo do not publicly disclose their 'importance' metric.) For each website, the total number of resources in the website is shown along with the total number of resources that were recovered and the percentage. Resources are also totalled by MIME type. The difference vector for the website accounts for recovered files that were added.

The 'Almost identical' column of Table 3 shows the percentage of text-based resources (e.g., HTML, PDF, PostScript, Word, PowerPoint, Excel) that were *almost identical* to the originals. The last column shows the reconstruction figure for each website if these almost identical resources are moved from the 'Changed' category to 'Identical' category. We considered two text-based resources to be almost identical if they shared at least 75% of their shingles of size 10. Shingling (as proposed by Broder et al. [3]) is a popular method for quantifying similarity of text documents when word-order is important [2, 11, 21]. We did not use any image similarity metrics.

We were able to recover more than 90% of the original resources from a quarter of the 24 websites. For three quarters of the websites we recovered more than half of the resources. On average we were able to recover 68% of the website resources (median=72%). Of those resources recovered, 30% of them on average were not byte-for-byte identical. A majority (72%) of the 'changed' text-based files were almost identical to the originals (having 75% of their shingles in common). 67% of the 24 websites had obtained additional files when reconstructed which accounted for 7% of the total number of files reconstructed per website.

When all website resources are aggregated together and examined, dynamic pages (those that contained a '?' in the URL) were significantly less likely to be recovered than resources that did not have a query string (11% vs. 73%). URLs with a path depth greater than three were also less likely to be recovered (52% vs. 61%). A chi-square analysis confirms the significance of these findings ($p < .001$). We were unable to find any correlation between percentage of recovered resources with PageRank or website size.

The success of recovering resources based on their MIME type is plotted in Figure 4. The percentage of resources

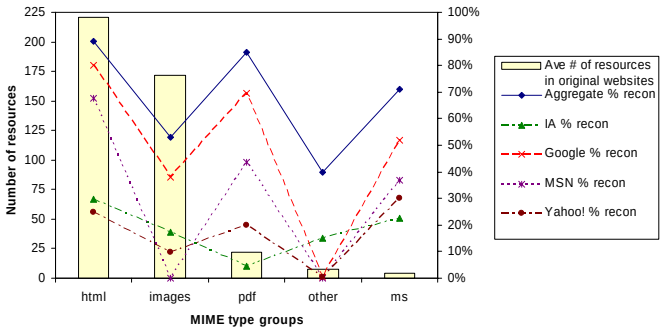

Figure 4: Recovery success by MIME type

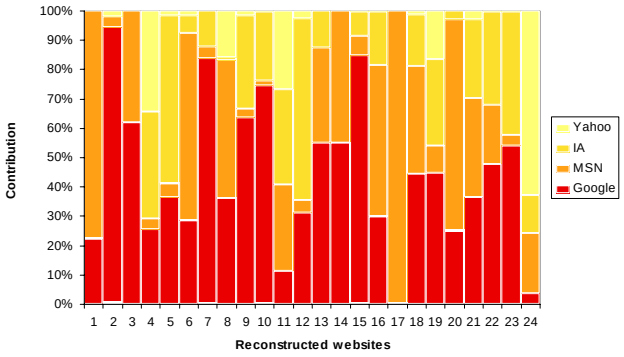

Figure 5: Web repositories contributing to each website reconstruction

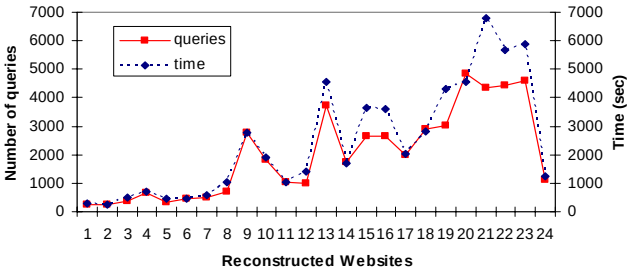

Figure 6: Number of queries performed and time taken to reconstruct websites

that were recovered from the five different website reconstructions we performed (one using all four web repositories, and four using each web repository individually) are shown along with the average number of resources making up the 24 downloaded (or original) websites. A majority (92%) of the resources making up the original websites are HTML and images. We were much more successful at recovering HTML resources than images; we recovered 100% of the HTML resources for 9 of the websites (38%) using all four web repositories. It is likely we recovered fewer images because MSN cannot be used to recover images, and as our caching experiment revealed, images are also much less likely to be cached than other resource types.

Figure 4 also emphasizes the importance of using all four web repositories when reconstructing a website. By just using IA or just using Google, many resources will not be recovered. This is further illustrated by Figure 5 which shows the percentage of each web repository's contribution in the aggregate reconstructions (sites are ordered by number of URIs). Although Google was the largest overall contributor to the website reconstructions (providing 44% of the resources) they provided none of the resources for site 17 and provided less than 30% of the resources for 9 of the reconstructions. MSN contributed on average 30% of the resources; IA was third with 19%, and Yahoo was last with a 7% contribution rate. Yahoo's poor contribution rate is likely due to their spotty cache access as exhibited in our caching experiment (Figure 2) and because last-modified datestamps are frequently older than last-cached datestamps (Warrick chooses resources with the most recent datestamps).

The amount of time and the number of queries required to reconstruct all 24 websites (using all 4 repositories) is shown in Figure 6. Here we see almost a 1:1 ratio of queries to seconds. Although the size of the original websites gets larger along the x-axis, the number of files reconstructed and the number of resources held in each web repository determine how many queries are performed. In none of our reconstructions did we exceed the daily query limit of any of the web repositories.

5. FUTURE WORK

We have made Warrick available on the Web[4], and it has been used to reconstruct several websites have been lost due to fire, hard-drive crashes, death of the website owner, hacking, and discontinued charitable website hosting [19]. Although the reconstructions have not been complete, individuals are very thankful to have recovered any resources at all when faced with total loss.

There are numerous improvements we are making to Warrick including an API for easier inclusion of new web repositories and new methods for discovering more resources within a web repository [19]. We are planning on reconstructing a larger sample from the Web to discover the website characteristics that allow for more effective "lazy recovery". Discovering such characteristics will allow us to create guidelines for webmasters to ensure better lazy preservation of their sites. Our next experiment will take into account rate of change and reconstruction differences over time.

We are also interested in recovering the server-side components (CGI programs, databases, etc.) of a lost website. We are investigating methods to inject server-side components into indexable content using erasure codes (popular with RAID systems [22]) so they can be recovered from web repositories when only a subset of pages can be found.

A web-repository crawler could be used in the future to safeguard websites that are at risk of being lost. When a website is detected as being lost, a reconstruction could be initiated to preserve what is left of the site. Additionally, websites in countries that are targeted by political censorship could be reconstructed at safe locations.

6. CONCLUSIONS

Lazy preservation is a best-effort, wide-coverage digital preservation service that may be used as a last resort when

[4] `http://www.cs.odu.edu/~fmccown/warrick/`

website backups are unavailable. It is not a substitute for digital preservation infrastructure and policy. Web repositories may not crawl orphan pages, protected pages (e.g., robots.txt, password, IP), very large pages, pages deep in a web collection or links influenced by JavaScript, Flash or session IDs. If a web repository will not or cannot crawl and cache a resource, it cannot be recovered.

We have measured the ability of Google, MSN and Yahoo to cache four synthetic web collections over a period of four months. We measured web resources to be vulnerable for as little as 10 days and in the worst case, as long as our 90 day test period. More encouragingly, many HTML resources were recoverable for 8–51 days on average after being deleted from the web server. Google proved to be the most consistent at caching our synthetic web collections.

We have also used our web-repository crawler to reconstruct a variety of actual websites with varying success. HTML resources were the most numerous (52%) type of resource in our collection of 24 websites and were the most successfully recoverable resource type (89% recoverable). Images were the second most numerous (40%) resource type, but they were less successfully recovered (53%). Dynamic pages and resources with path depths greater than three were less likely to be recovered. Google was the most frequent source for the reconstructions (44%), but MSN was a close second (30%), followed by IA (19%) and Yahoo (7%). The probability of reconstruction success was not correlated with Google's PageRank or the size of the website.

7. REFERENCES

[1] H. Berghel. Responsible web caching. *Communications of the ACM*, 45(9):15–20, 2002.

[2] K. Bharat and A. Broder. Mirror, mirror on the web: a study of host pairs with replicated content. In *Proceedings of WWW '99*, pages 1579–1590, 1999.

[3] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. Syntactic clustering of the Web. *Computer Networks & ISDN Systems*, 29(8-13):1157–1166, 1997.

[4] M. Burner. Crawling towards eternity: Building an archive of the world wide web. *Web Techniques Magazine*, 2(5), 1997.

[5] F. Can, R. Nuray, and A. B. Sevdik. Automatic performance evaluation of web search engines. *Info. Processing & Management*, 40(3):495–514, 2004.

[6] J. Cho, N. Shivakumar, and H. Garcia-Molina. Finding replicated web collections. In *Proceedings of SIGMOD '00*, pages 355–366, 2000.

[7] B. F. Cooper and H. Garcia-Molina. Infomonitor: Unobtrusively archiving a World Wide Web server. *International Journal on Digital Libraries*, 5(2):106–119, April 2005.

[8] M. Day. Collecting and preserving the World Wide Web. 2003. `http://library.wellcome.ac.uk/assets/WTL039229.pdf`.

[9] C. E. Dyreson, H. Lin, and Y. Wang. Managing versions of web documents in a transaction-time web server. In *Proceedings of WWW '04*, pages 422–432, 2004.

[10] D. Fetterly, M. Manasse, and M. Najork. Spam, damn spam, and statistics: using statistical analysis to locate spam web pages. In *Proceedings of WebDB '04*, pages 1–6, 2004.

[11] D. Fetterly, M. Manasse, M. Najork, and J. Wiener. A large-scale study of the evolution of web pages. In *Proceedings of WWW '03*, pages 669–678, 2003.

[12] Google Sitemap Protocol, 2005. `http://www.google.com/webmasters/sitemaps/docs/en/protocol.html`.

[13] Google webmaster help center: Webmaster guidelines, 2006. `http://www.google.com/support/webmasters/bin/answer.py?answer=35769`.

[14] M. Gordon and P. Pathak. Finding information on the World Wide Web: the retrieval effectiveness of search engines. *Inf. Process. Manage.*, 35(2):141–180, 1999.

[15] A. Gulli and A. Signorini. The indexable web is more than 11.5 billion pages. In *Proceedings of WWW '05*, pages 902–903, May 2005.

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

[17] D. Lewandowski, H. Wahlig, and G. Meyer-Beautor. The freshness of Web search engine databases. *Journal of Information Science*, 32(2):131–148, Apr 2006.

[18] F. McCown, X. Liu, M. L. Nelson, and M. Zubair. Search engine coverage of the OAI-PMH corpus. *IEEE Internet Computing*, 10(2):66–73, Mar/Apr 2006.

[19] F. McCown and M. L. Nelson. Evaluation of crawling policies for a web-repository crawler. In *Proceedings of HYPERTEXT '06*, pages 145–156, 2006.

[20] F. McCown, J. A. Smith, M. L. Nelson, and J. Bollen. Reconstructing websites for the lazy webmaster. Technical report, Old Dominion University, 2005. `http://arxiv.org/abs/cs.IR/0512069`.

[21] A. Ntoulas, J. Cho, and C. Olston. What's new on the Web? The evolution of the Web from a search engine perspective. In *Proceedings of WWW '04*, pages 1–12, 2004.

[22] J. S. Plank. A tutorial on Reed-Solomon coding for fault-tolerance in RAID-like systems. *Software: Practice and Experience*, 27(9):995–1012, 1997.

[23] H. C. Rao, Y. Chen, and M. Chen. A proxy-based personal web archiving service. *SIGOPS Operating Systems Review*, 35(1):61–72, 2001.

[24] V. Reich and D. S. Rosenthal. LOCKSS: A permanent web publishing and access system. *D-Lib Magazine*, 7(6), 2001.

[25] A. Ross. Internet Archive forums: Web forum posting. Oct 2004. `http://www.archive.org/iathreads/post-view.php?id=23121`.

[26] J. A. Smith, F. McCown, and M. L. Nelson. Observed web robot behavior on decaying web subsites. *D-Lib Magazine*, 12(2), Feb 2006.

[27] M. Weideman and M. Mgidana. Website navigation architectures and their effect on website visibility: a literature survey. In *Proceedings of SAICSIT '04*, pages 292–296, 2004.

[28] J. Zhang and A. Dimitroff. The impact of webpage content characteristics on webpage visibility in search engine results (part I). *Information Processing & Management*, 41(3):665–690, 2005.