# Accessing the History of the Web: A Web Way-Back Machine

Joachim Feise

University of California, Irvine
Information and Computer Science
Irvine, CA 92627-3425, USA
jfeise@ics.uci.edu
http://www.ics.uci.edu/~jfeise/

**Abstract.** One of the deficiencies of the World Wide Web is that the Web does not have a memory. Web resources always display one revision only, namely the latest one. In addition, once a Web resource moves from one location (i.e., URL) to another, the resource at the original location ceases to exist.
Since the Web does not provide a mechanism to allow access to the revision history of resources, services like the Internet Archive have sprung into action to collect the history of important Web resources.
The Web way-back machine described here, currently under development by the author, utilizes collections of historical Web resources like the ones provided by the Internet Archive to allow online, read-only access to the revisioned resources.

## 1 Introduction

The World Wide Web is quickly becoming the standard way to present information in cyberspace. However, contents on the Web changes quickly, and there is currently no general way to access the history of Web-based resources. The resource authors may or may not use document management systems that provide a revision history of resources, but even if authors use such systems the resources as published on the Web still lack any history information.

The process of collecting Web resources is well understood and is used on a daily basis by Web search sites, like Google[3] or AltaVista[1], as well as by the Internet Archive[4], an organization dedicated to creating an archive of Web resources over time. All these organizations use Web crawlers[12] to collect resources from the Web.

Web search sites and the Internet Archive differ in their processing and storage of the collected data, though. Web search sites only utilize the last resource revision collected. Access to previous revisions is not possible. The collected resources are processed in various ways to support efficient searches (for details, see, for example, the processing overview at Google[15]).

The Internet Archive stores its collected data in aggregate files with a size of up to 100 Megabyte[5]. This collection was started in October 1996 and has

reached a size of approximately 14 terabytes. The size of this collection and the aggregation of unrelated Web resources in large files facilitates offline access and analysis, but makes online, real-time access impractical. For example, it is unlikely that one organization alone can provide the facilities to allow online access to a collection of this size. In addition, the computational expense to extract resource metadata, like relationship information between resources, from the aggregate files precludes the fast access required by online use. To provide online access to this vast amount of data, a different approach is therefore necessary.

Each Web resource collected by a service like the Internet Archive is associated with a day and time stamp indicating the date the resource was collected. The history of a specific Web resource as stored by the Internet Archive is therefore comprised of the collection of documents from a specific location at discrete dates. Obviously, if the capturing frequency is low, this may not capture the full resource history. On the other hand, a high capturing frequency incurs a high bandwidth utilization, which may not be desirable for sites with bandwidth restrictions. In the case of static resources or resources with a low revision rate, a high capturing frequency would also result in redundant data, which, in the interest of minimizing storage requirements, would require an approach to detect duplicate entries.

This paper describes a Web way-back machine, currently under development by the author, which utilizes a proxy server and any standard Web browser with frame and JavaScript capabilities to allow online access to Web resources collected over time.

## 2 Resource Organization

As mentioned above, the organization of the collected resources is usually optimized for offline storage. An analysis of the storage format used by the Internet Archive, for example, reveals that this format is not suited for efficient online access. The data files aggregate the collected resources without any notion of relationships between the resources.

It is therefore likely that the histories of specific resources are scattered among multiple of these data files. For a proper representation of the changes of resources over time, the resource history needs to be available to the user. A real-time recovery of the history of any given resource from the Internet Archive data files would require an extraordinary amount of processing, which would be prohibitive for any Internet-scale use. In addition, it is highly unlikely that the vast amount of data collected can be stored and presented by one organizational entity alone, nor is it desirable to put the responsibility for maintaining this amount of data into the hands of only one organization.

Distribution of the data storage across organizational boundaries therefore becomes a necessity. Ideally, the distributed nature of the data storage should be transparent to the user. Also desirable is the ability of dynamic reconfigurations of the data storage, for example to provide load balancing or to move parts of the data storage between organizations.

Organization and access of multiple versions of resources is generally the task of versioning and configuration management systems. These systems seem to be a natural fit for the task of providing fast online access to multiple versions of resources collected from the Web. However, configuration management systems are usually limited to an intra-organization scale. Conventional configuration management systems typically do not scale beyond these boundaries. There are numerous attempts to provide support for distributed operation in commercial configuration management systems, but these attempts are usually limited to simple client-server interfaces (for a detailed discussion see A. van der Hoek's Ph.D. dissertation[11].)

### 2.1 Data Storage

We chose to use the Network Unified Configuration Management System (NUCM)[11] for the storage and organization of the revisioned resources, since NUCM was designed to provide the transparent distributed storage capabilities required for the Web way-back machine. NUCM also allows dynamic reconfiguration of the data storage across machine boundaries.

Since configuration management systems like NUCM usually store each resource in a local filesystem, efficient filesystem access becomes important. In fact, the difficulty of accessing 100,000s of files efficiently led the Internet Archive to aggregate the collected resources in large files. Configuration management systems, however, usually use just one file for all of a resource's history, which decreases the burden on the filesystem. In addition, a clever organization of the resources can further alleviate this problem. A natural hierarchical organization of Web resources can be achieved by exploiting the organization of the URLs used to reference these resources. This organization mirrors the domain name structure of URLs.

At the top-level of the hierarchy, resources can be classified as belonging in one of the .com, .edu, .uk, .de, etc. domains. The second level would comprise organization names like uci, microsoft, etc. The hierarchical decomposition continues along the whole domain name, followed by an analogous decomposition of the actual directory path of the URL.

A resource referenced with the URL http://www.ics.uci.edu/~jfeise/index.html would therefore be referenced in the NUCM configuration management system as //nucm_root/edu/uci/ics/www/~jfeise/index.html. If necessary, a finer grained storage system can easily be devised. In fact, since the number of resources in the .com domain is likely to cause the same filesystem efficiency problems as outlined above, a more useful mapping between URLs and the filesystem storage may separate resources by the first two letters of the organization name. This scheme can easily be extended, so that the resource http://www.microsoft.com/index.html may be stored in NUCM under //nucm_root/com/mi/cr/microsoft/www/index.html.

The dynamic reconfigurability of NUCM allows to perform storage schema extensions when necessary. It is therefore possible to start out with the coarse grained mapping, and refine it as desired during the population of the system

or even during the use of the Web way-back machine, if performance problem manifest themselves.

The transparent nature of resource distribution in NUCM allows for branches in resource tree being located on different machines and across organizational boundaries. For example, the branch //nucm_root/com could be located on one machine, while the the branch //nucm_root/com/mi/cr/microsoft may be stored on a different machine located in a different organization. A move of the branch to yet another location would not affect the tree structure at all.

## 3 Resource Access

The use of a Web way-back machine should be as transparent to the user as possible. This implies that the user should be able to view current Web resources and archived versions of the same or other resources from the same browser with a minimum of required browser configuration changes. The easiest way to accomplish this goal is to configure the user's Web browser to use a proxy server. The proxy server then would have the task of distinguishing between requests for normal, current Web resources and requests for historical revisions of Web resources.

### 3.1 Client

Ideally, a modification to the browser would allow users to specify the kind of requests they issue. In the case of requests for historical versions of Web resources, the user also needs to be able to specify the date or revision of interest. This information would then be sent with the request to the proxy server, in the form of an additional line in the HTTP header of the request. Unfortunately, the existing Web browsers do not provide facilities for the addition of such browser extensions. In particular, none of the existing browser extension capabilities, like plugins or ActiveX controls, allow for the modification or addition of HTTP headers.

It is therefore necessary to use other data transfer mechanisms like cookies or scripting approaches. An analysis of the cookie specifications[7,6] shows that cookies are not suitable for the task at hand, since cookies are designed to be exchanged between the browser and the target Web site. Although interception and manipulation of cookies in the proxy server is possible, the requirement of a user-changeable date or revision of interest rules out the use of cookies, which can not be modified in the browser.

The approach we chose is therefore based on the use of frames and JavaScript in the Web browser. These features are available in the vast majority of browsers available in the marketplace. Text-only browsers like Lynx and very old versions of Netscape's and Microsoft's browsers would be excluded from using the Web way-back machine, though.

The browser displays two frames, a navigation frame and a resource frame. The navigation frame contains several dialog elements to select a date of interest

or a specific revision of the resource. The navigation frame also allows to easily navigate between revisions or between dates. In addition, the navigation frame can easily be extended to provide information like the date when the resource was collected, or the frequency of collection.

The navigation frame also needs to contain an off-switch, to indicate to the proxy server that the user no longer wishes to view historical revisions of Web resources.

### 3.2 Server

Once the user's Web browser is set up according to the discussion above, a proxy server is utilized to receive the user input and provide the user with the appropriate resource.

Like browsers, proxy servers do not support the functionality required for this task natively. Unlike browsers, however, a variety of proxy servers are distributed as Open Source. Even the leading Web server, the Apache server[2], can be used as proxy server. However, the Apache server is not optimized to serve as proxy server. We therefore settled for the Squid proxy server and Web cache[10], which, like Apache, is distributed as Open Source.

The availability of the source code for the proxy server allows us to implement the functionality required to support our Web way-back machine. Especially, we require the proxy server to receive and process any action the user performs in the navigation frame in the Web browser.

As long as the user does not utilize the navigation frame, or the off-switch in the navigation frame is selected, the proxy server performs its standard processing, i.e., for each incoming request, it first checks if the requested Web resource is in the proxy's cache. If it is, the resource is delivered to the requestor. If the proxy determines that it does not hold the Web resource, it in turn requests it from the original Website or from another proxy server. The details of this process are described in RFCs 2186[13] and 2187[14], as well as in the Squid documentation[8,9].

A different processing path is chosen in the proxy server if the user selects a specific date or revision in the navigation frame in the browser.

First, if the user selects a specific date or revision in the navigation frame, the navigation frame in turn encodes this selection in a URL and sends this data as a request to the proxy server. A special URL handler (e.g., a CGI handler) decodes the request and stores the selection data in the proxy server for later use.

Second, when the user now requests a resource, the proxy server determines that a historic revision of the resource is required. At this point, the path of standard processing is left, and the proxy server utilizes the NUCM configuration management system to locate the appropriate revision of the requested resource.

## 4 Limitations

As is customary in configuration management systems, requests with specific dates are mapped to the last resource with a date equal or prior to the requested one. It is therefore possible that subsequent requests with different dates return the same resource. In order to indicate this possibility to the user, the proxy server, in addition to the requested resource, updates the navigation frame with the revision of the resource and the date the resource was collected.

Unfortunately, this mechanism can not be applied to a resource that changed locations during its history. In this case, the revision history of this resource would be treated as two independent revision histories, with the separation between them at the date of the location change. The resource at the original location does not change after this date anymore, and the revision history does not contain any indication that the location of the resource changed.

Resources that are deleted at a certain point in time present another special case. Unless the collection archive explicitly records that a resource does no longer exist, no knowledge about deletions exists or can be inferred from the resource collection.

The Internet Archive, for example, allows Web administrators to opt out of the collection process. If the administrator of a Web site does so, no further resources from that Web site are collected. However, the previously collected resources are still accessible in the collection archive. In this case, the limited revision history therefore does not reflect the real revision history of the resources anymore.

Another factor limiting the accuracy of the projected revision history is the fact that the frequency of the resource collection is finite. It is therefore likely that some resource changes on highly dynamic Web sites will not be recorded.

The collection of resources is usually performed with the help of Web crawlers, which automatically visit Web sites, retrieve the resources, and follow any links contained in them. In order to limit the bandwidth impact of these crawlers, they generally do not collect all the resources of a Web site in one transaction. This can easily lead to link inconsistencies within supposedly identical revisions. An example is a reorganization of a Web site in the minutes between two consecutive visits from the Web crawler.

## 5 Early Results

As proof of concept, I implemented a first, limited prototype to show the feasibility of the approach and perform first scalability tests. This prototype does not use real Web resources, but instead relies on artificially generated resources, created by a small helper program and stored in a NUCM database. Tests with the prototype showed that first-time access to a versioned resource is reasonably fast, even if the NUCM database storing the resources is accessed over a LAN on a machine different from the proxy server. The access time in this setting is in the order of 2-3 seconds. Co-locating the proxy server and the NUCM database

on the same computer increases the access time roughly by a factor of 3. Subsequent accesses to different revisions of the same resource remained in the under 1 second range, even with the NUCM archive located on a different computer on the LAN.

However, the tests showed that the population of a NUCM archive at a rate of even one second for the local filesystem case will require some thought as the amount of resources to be stored grows. At this rate, the population of a NUCM archive with just 1 million resources would take about half a month. The author is investigating the feasibility of bypassing the NUCM API in order to increase the speed of the population process, thereby avoiding the overhead of maintaining a temporary NUCM workspace on the local machine solely for the purpose of populating the database.

Arguably, the population time could be considered negligible, given that the population task is a one-time process. However, we feel that in the interest of scalability, the population process should be able to handle the current one billion resources collected by the Internet Archive in a reasonable amount of time.

## 6 Future Work

I intend to expand the existing prototype to a fully functional Web way-back machine. In recent months, my collegue Jim Whitehead has established contact with the Internet Archive in the expectation to get access to parts of the Web resources the Archive has collected. Access to this collection will allow the creation of a larger NUCM archive than the one currently in use, and will facilitate the evaluation of the scalability of my approach. I also want to make the Web way-back machine available to a reasonably sized group of users to allow the evaluation of scalability with respect to the number of users. In addition, I expect the feedback from a diverse group of users to be very valuable in improving the design of the user interface presented in the browser.

## 7 Acknowledgements

I am greatly indebted to Jim Whitehead, Yuzo Kanomata and André van der Hoek for numerous discussions helping to shape the ideas presented in this paper. Yuzo also provided very helpful comments on an earlier version of this paper.

I also wish to acknowledge the receipt of a generous financial scholarship from Eagle Creek Systems, Inc. in support of my research.

## References

1. AltaVista, http://www.altavista.com/
2. The Apache Server Project, http://www.apache.org/httpd.html
3. Google, http://www.google.com/
4. The Internet Archive, http://www.archive.org/

5. The Internet Archive: Storage and Preservation of the Collections, http://www.archive.org/collections/storage.html
6. Kristol, D. and Montulli, L. HTTP State Management Mechanism, Lucent Technologies, Netscape, February 1997, http://www.ietf.org/rfc/rfc2109.txt
7. Netscape Cookies, 1997, http://developer.netscape.com/docs/manuals/communicator/jsguide4/cookies.htm
8. Squid Frequently Asked Questions: How does Squid work, http://www.squid-cache.org/Doc/FAQ/FAQ-12.html
9. Squid User Manual, http://www.squid-cache.org/Doc/Users-Guide/
10. Squid Web Proxy Cache, http://www.squid-cache.org/
11. van der Hoek, A. A Reusable, Distributed Repository for Configuration Management Policy Programming, Ph.D. Dissertation, University of Colorado at Boulder, 2000, http://www.ics.uci.edu/~andre/research/papers/DISSERTATION.ps
12. The Web Robots Pages, http://info.webcrawler.com/mak/projects/robots/robots.html
13. Wessels, D. and Claffy, K. Internet Cache Protocol (ICP), version 2, National Laboratory for Applied Network Research, U.C. San Diego, September 1997, http://www.ietf.org/rfc/rfc2186.txt
14. Wessels, D. and Claffy, K. Application of Internet Cache Protocol (ICP), version 2, National Laboratory for Applied Network Research, U.C. San Diego, September 1997, http://www.ietf.org/rfc/rfc2187.txt
15. Why Use Google?, http://www.google.com/why_use.html