Creating a Permanent Web Publishing and Access System

David S. H. Rosenthal, Sun Microsystems Laboratories
Vicky Reich, Stanford University Libraries
May 12, 2000

What it is LOCKSS (Lots Of Copies Keep Stuff Safe) is a system prototype to preserve access to scientific journals published on the Web. LOCKSS models how libraries provide access to paper content by allowing individual libraries to safeguard their communities access to Web content. The system ensures that hyperlinks continue to resolve and appropriate content is delivered, even when in the Internet the links don't work and content is no longer available. Libraries running LOCKSS cooperate to detect and repair preservation failures.

LOCKSS models current collection development, circulation, and interlibrary loan systems. In library terms, archives are handled differently from general circulating collections. LOCKSS deals with circulation, not with archiving. Its focus is on providing access rather than on preservation at all costs.

LOCKSS is designed to leverage the characteristics of the material it is safeguarding. The peer review process makes the material essentially immutable. It is delivered via HTTP using static URLs organized in a fairly rational way.

LOCKSS will be distributed free, it is designed to run on very cheap hardware and to require almost no technical administration.

The problem

Web publishing is an effective medium for scientific, technical and medical (STM) communication (datasets, dynamic lists of citing papers, e-mail notification of citing papers, hyperlinks, searching). Increasingly the Web editions are the 'version of record' and the paper is merely a subset of the peer reviewed scientific discourse.

Librarians can provide their readers with access to material published on paper, even when it is centuries old. For good reasons, librarians are skeptical about their ability to provide long-term access to electronic materials. Librarians need an inexpensive, robust tool with which they can provide their readers long-term access to purchased materials.

The Solution

The solution is in three parts:

The bits of content must be preserved.
Access to the bits of content must be preserved.
The ability to understand the bits of content must be preserved.

There is no single solution to this problem. A single solution of itself would be perceived as vulnerable. By proposing one solution we are not arguing that other solutions should not be developed and deployed. Diversity is essential to successful preservation.

Technical Details

LOCKSS is a majority-voting fault-tolerant system that has far more replicas than would be required to survive normal anticipated failures. The system uses off-the shelf open source software to manage a web cache at each library for each journal the library wishes to safeguard, and to pre-load the cache with the pages of the journal as they are published. Thus pages will be preserved even if they aren't read.

Through these caches each library takes physical custody of selected Web journals it purchases. Unlike normal caches, pages in these caches are never flushed; the caches grow indefinitely as the journals continue to publish. Over time, the disks holding an individual cache will fill up or fail.

The system detects and recovers from failures using a newly designed inter-cache protocol called LCAP [Library Cache Auditing Protocol]. LCAP allows caches to:

1. Challenge each other to prove that they have a good copy of part or all of a journal they are safeguarding,

2. Estimate a lower bound on the number of copies of the journal in question.

3. Request volunteers to make additional copies of a journal with too few.

If a cache detects that its copy of a page is missing or corrupt it can ask one of the other caches to provide a replacement copy. LOCKSS will not supply copies except to caches that have proved in the past that they had a copy, thereby respecting publisher's access control mechanisms.

Medium

Creating a Permanent Web Publishing and Access System

David S. H. Rosenthal, Sun Microsystems Laboratories
Vicky Reich, Stanford University Libraries
May 12, 2000

LOCKSS (Lots Of Copies Keep Stuff Safe) is a prototype of a system to preserve access to scientific journals published on the Web.

The Problem

In most respects the Web is a far more effective medium for scientific, technical and medical (STM) communication than paper. Techniques such as datasets in spreadsheets behind graphs, dynamic lists of citing papers, e-mail notification of citing papers and so on are becoming commonplace. These, added to the basic hyperlinks and easy searching, make Web journals easier to access and more useful when than the paper. Web versions frequently appear earlier and contain much more information. Many journals now publish some papers only on the Web.

Librarians have some measure of confidence in their ability to provide their readers with access to material published on paper, even if it is centuries old. Preservation is a by-product of the need to scatter copies around to provide access. A subscription to a paper journal provides the library with an archival copy of the content. Librarians are skeptical about their ability to provide long term access to materials published on the Web. Subscribing to a Web journal rents access to the publisher's copy. The publisher may promise "perpetual access", but there is no business model to support the promise. This poses a problem for librarians, who wish to provide both current and future readers with access to published literature.

Librarians want and need to provide their readers with long-term access to selected content. The problem is in three parts:

The bits themselves must be preserved. All digital storage media have a limited lifetime; the bits need to migrate from one medium to another over time. In practice it is difficult to fund a bulk copying effort when a medium starts decaying, so all but the most valuable bits are lost at each transition. Access to the bits must be preserved. Unless links to pages continue to resolve, the material will effectively be lost because it's likely few will have the knowledge or resources to retrieve it. The ability to translate the bits, once accessed, into human-readable form must be preserved.

There can be no single solution to this problem. A single solution of itself would be perceived as vulnerable. By proposing one solution we are not arguing that other solutions should not be developed and deployed. Diversity is essential to successful preservation.

Requirements

As mentioned, librarians 'scatter copies' to provide access; preservation has been a by-product. Librarians' techniques for preserving access to published paper materials have been honed since 415AD, when much of the world's literature was lost in the Library of Alexandria fire. These techniques include: acquire lots of copies; scatter them around the world so it is easy to find some of them but hard to find all of them; lend or copy your copies when other librarians need them.

In this context, we make a distinction between archives (not LOCKSS) and general circulating collections (LOCKSS). Archives explicitly preserve material that is unique and/or impossible to replicate widely. Access is restricted to protect the artifacts and to ensure preservation. Circulating library collections provide access, explicit risks are taken with each artifact to achieve this goal. Many replicas/copies are loaned to readers on a promise that they will eventually be returned.

Libraries' circulating collections are a model fault-tolerant distributed system. As a whole, the system is highly replicated and far more reliable than any individual component. There is no single point of failure, no central control to be subverted. There is a low degree of policy coherence between the replicas, and thus low systemic risk. The desired behavior of the system as a whole emerges as the participants take actions in their own local interests and cooperate in ad-hoc, informal ways. If librarians are to have confidence in an electronic system, it will help if the system works in a familiar way.

The fundamental requirement for LOCKSS is to model these techniques as closely as possible for web published materials. If libraries can take physical custody of purchased journals, in a form that preserves reader access, they can assume the responsibility for their future. If a library takes custody of a copy of a Web journal, the copy can behave as a Web cache and provide access whether or not it is available from the original publisher. If many libraries do so, the caches can communicate with each other to increase the reliability and availability of the service, as inter-library loan increases the reliability and availability of access to information on paper.

Another perspective on LOCKSS is that it provides librarians with journal access insurance. Reasons journals might become inaccessible include: Subscription cancellations, changes in publisher access policies (a not-for-profit journal is acquired by a for-profit publisher); publisher ceases publication, incompetent management of a publisher's web service.

Librarians balance the cost of preserving access to old material against the cost of acquiring new material. They tend to favor acquiring new material. To be effective subscription insurance must cost much less than the subscription itself.

The biggest journal on the High Wire Press web site (http://highwire.stanford.edu) generates about 6GB/year. A cheap PC to hold 5 years' worth might cost $600 today, which is about 10% of the subscription for the 5 years. If the running costs of the system can be kept low enough, it should now be practical for many libraries to maintain their own copies. The prospects for this insurance improve as equipment prices fall and subscription prices rise.

Open Source development is crucial: One goal of the project is to inspire confidence. It is hard to have well-founded confidence in a system whose operations are kept secret. The system's economics mandate free distribution of the software; there's barely a budget for the hardware. The longevity of the system will require many generations of programmers to refine it as problems are encountered.

Technology

The design goal for LOCKSS is to provide librarians with a cheap and easy way of running Web caches which:

Collect content into the cache as new issues of the journals are published,
Serve content to readers from either the publisher or from the cache,
Preserve the contents of the cache for posterity by never flushing it.

The design capitalizes on two features of STM Web journals: peer- reviewed articles are immutable once published, and journal web sites have a logical structure. We are not designing a general-purpose Web content preservation system; LOCKSS is not suitable for volatile content. LOCKSS is work in progress; the design will evolve as we gain experience.

Collecting: A librarian instructs an instance of LOCKSS to preserve a volume of a journal by providing the publisher's root URL for the volume and a frequency of publication, say monthly. At that frequency a web-crawler starts from the root URL and fetches all new pages within that sub-tree. The publisher's web server sees this access as coming from an authorized IP address, so it is allowed. Readers need not access the material to populate the cache. This component of LOCKSS uses off-the-shelf technology – the w3mir crawler.

Serving: The prototype uses the Apache web server to export the contents of each cache to the local network's users.

Preserving: The heart of LOCKSS is an inter-cache protocol we call LCAP [Library Cache Auditing Protocol]. LCAP is an IP multicast protocol caches talk to discover which URLs should exist and what their contents should be. It runs continually but very slowly between all the caches. If a cache discovers a missing or damaged URL it can fetch a new copy via HTTP from the original publisher, or from one of the other caches. Care is taken not to subvert the publisher's access control mechanism; content should only be delivered to sites that have rights to it.

The process to detect and repair damage to cached content is:

Each cache "wakes up" at preset time intervals to check how long it's been since the integrity of some content was checked. If it's been too long, the cache starts a poll (similar to an opinion poll) among caches. The poll challenges other caches to prove they have the same content as the caller using secure hashing techniques. Other caches will respond to the challenge by computing appropriate hash values and replying. The caches hearing the replies will tally the poll. If they are on the winning side their cache is intact. If they are on the losing side, their cache contains some damage. The damaged cache calls another poll to zero in on the location of the damage, When a damaged file is located a new copy is fetched from the publisher, if the publisher still exists, or from one of the winners.

To assess a participant's credibility, LOCKSS observes and remembers their behavior over time. Because LCAP is a peer-to-peer multicast protocol, the behavior of each participant is visible to the others and all actions are public. Taking bad actions reduces your credibility.

LOCKSS is strong in some unusual ways:

. --There is no central coordination point that can be attacked.

It doesn't depend on the Domain Name System, or a Public Key Infrastructure.

There’s no dependence on preserving meta-information. Provided enough other participants preserve the journal articles, at some cost in credibility a participant can corrupt or lose any or all of its information.

There are no passwords or encryption keys to be kept secret.

By operating slowly even on human timescales the system makes it easier to detect an attacker and limits the damage he can do before being stopped.

Future Plans

Complete Alpha and Beta tests:
We're running an initial test of the prototype with a handful of libraries and a single journal starting in Summer 2000. We plan to assess this test, incorporate the experience and run a second test at a much more realistic scale later in the year.

Evolve a Production Version: Plans for a production system will be released once alpha and beta testing is complete.

Expand beyond journals: We're also exploring the suitability of LOCKSS for applications other than journals. One obvious example is the government documents that used to be kept on paper in the "depository library" system, but which are now being published on the Web.

Measuring LOCKSS Performance: There are three important performance metrics for LOCKSS once it is deployed in production: What does it cost a library to run it? How often does the system as a whole lose or corrupt journal articles? What is the probability that a reader will encounter a missing or corrupt article?

Other implementations: All monocultures are vulnerable, and if deployeden masse LOCKSS would be a monoculture. A bug in the implementation could wipe out information system-wide. It would be very valuable to have multiple independent implementations of the LCAP protocol. We hope that by keeping the protocol very simple we will encourage other implementations. The source code will be released under a Stanford equivalent of the U.C. Berkeley license

Acknowledgments

Heartfelt thanks to Michael Keller, the Stanford University Librarian for his support and encouragement, to Michael Lesk at the NSF for funding the project with grant IIS-9907296, and to Sun Microsytems Laboratories, which has provided both time and funds. We're also grateful to our long-suffering alpha sites, and to AAAS, who allowed the valuable content of Science Online to act as the experimental subject.