High Energy Physics Libraries Webzine

Issue 7 / April 2003

Diffused Knowledge Immortalizes Itself [(1)]

The LOCKSS Program [(2)]

Victoria A. Reich [(*)]
October 31, 2002

Abstract

The LOCKSS model, 'Lots of Copies Keeps Stuff Safe,' creates low-cost, persistent digital 'caches' of authoritative versions of http-delivered content. The LOCKSS software enables institutions to locally collect, store, preserve and archive authorized content, thus safeguarding their community's access to that content. The current version of LOCKSS software is restricted to electronic journals. Accuracy and completeness of LOCKSS caches are assured through a peer-to-peer polling and reputation system (operated through LCAP, LOCKSS' communication protocol), which is both robust and secure. LOCKSS replicas cooperate to detect and repair preservation failures. LOCKSS is designed to run on inexpensive hardware and to require almost no technical administration. The software has been under development since 1999 and is distributed as open source.

Problem

Web-published materials are increasingly the authoritative versions. There are no affordable, widely available techniques for preserving this 'written record'. The web is an effective publishing medium offering enrichment of journal content, such as data sets, dynamic lists of citing papers, e-mail notification of citing papers, hyperlinks, searching. As web editions increasingly become the 'version of record', paper versions of the same titles become merely a subset of peer reviewed scholarly discourse.

One Approach

The LOCKSS Program's mission is to build tools and provide support to research libraries so they can easily and affordably create, preserve, and archive local electronic collections. We who have created this Program believe it is in society's best interest for libraries to own rather than lease electronic information and thus to retain their traditional custodial role for scholarly information. The LOCKSS Program will enable publishers, without risk to their business model or to their publishing platforms, simply to distribute electronic journals to libraries while relinquishing responsibility to provide perpetual access.

Communities of Interest

The LOCKSS model appeals to both librarians and publishers, and, it is hoped, to readers, because it conforms to the needs, desires and characteristic behaviors of each community, as described below.

Appeal to Library Community

Librarians are chartered to preserve access to the scholarly record for future generations. Their communities clearly want e-journals. However, there is a problem: content that libraries have owned in paper is now rented in the electronic version. A unilateral change of policy by the publisher, or failure to renew a subscription can remove their electronic access to past material with no recourse. To date there has been no mechanism to implement the traditional purchase-and-own library model for electronic materials. Librarians have lost the option to build and maintain local collections.

The LOCKSS model restores the notion of building local collections for electronic journals. The system allows librarians to take custody of and preserve access to the e-journals to which they subscribe, restoring the purchase model with which librarians are familiar. Using their own computers and network connections, librarians obtain, preserve and provide access to a purchased copy of an e-journal. This is analogous to librarians using their own buildings, shelves and staff, to obtain, preserve and provide access to paper journals.

Material stored in a local LOCKSS cache remains available to that local community's reader even when the publisher 'goes away' (due to merger, bankruptcy, subscription cancellation, network traffic, etc.). The content is never 'dark'; it is always available to the local community. Installing and populating LOCKSS caches are actions librarians can take for themselves to serve their local communities. LOCKSS must be affordable even to libraries with a limited budget, and thus the LOCKSS Program has emphasized utilizing affordable hardware and 'appliance-like' software.

The benefits of restoring the practice of library material ownership and the choice of long-term access outweigh the costs of maintaining the system and related equipment. We believe the costs of collection management will also prove to be affordable.

Appeal to Publisher Community

Publishers understand archiving is an important social need; few have the resources to take this responsibility for their own content. LOCKSS returns the responsibility for long-term preservation and the corresponding costs to the librarians.

While librarians increasingly understand that paper is no longer the version of record, in the absence of a sustainable digital archiving solution, they are retaining subscriptions to the paper version. Library acquisition funds are insufficient to purchase both formats. As a condition of moving to electronic versions, librarians are beginning to require that publishers guarantee long-term access to content sold. These guarantees are problematic at best. Only the largest publishers have sufficient resource to implement (or negotiate with third parties to provide) archives for content they publish. The smaller publishers have no easy way to meet this requirement. Publishers are motivated to endorse and support the LOCKSS system because of this impasse, and because the cost to them of doing so is low. (A list of supporting publishers is available, http://lockss.stanford.edu/projectstatus.htm.)

Most publishers fear their journal content may be illegally replicated or leaked; they want their access control methods enforced. When their sites are available, they want to retain access to reader usage data and record the reader's interactions with their site. The LOCKSS software meets these requirements. Uses of content in the caches are bound by the same legal agreements as the original subscription. Caches provide content only to the original authorized and authenticated subscriber base. The LOCKSS system supplies content to other caches only to repair damage in content they held previously. LOCKSS does not introduce any new leakage paths. The reader is supplied preferentially from the publisher, with the cache only as a fallback; the publisher sees the same interactions they would have seen without LOCKSS caches.

Appeal to the Reading Community

The LOCKSS system preserves a reader's access to content published on the web. Readers expect, at minimum: when they click on a link to it, or type in a URL, the relevant page will be delivered with minimal delay and no further interaction; when they enter terms into a search engine that should match the relevant page, it will be among the returned matches.

LOCKSS focuses on preserving the service of having links resolve to, or searches to find the relevant content. An institution using the LOCKSS system to preserve access to a journal runs a web cache devoted to that journal. Readers use the cache as a proxy in the normal way. The difference between LOCKSS caches and ordinary proxies is that the LOCKSS system crawls the journal publisher's web site and loads itself with newly published content before the first local user seeks it.

Just as other types of caches are invisible to their users, so are LOCKSS caches. They supply pages they preserve even if those pages are no longer available from the original publisher's web site. Preserved content remains accessible at the original publisher's URL. Links and bookmarks, searches through indexing and abstracting databases, etc. resolve either to the publishers site or to the locally-cached content. The techniques used to access content at the publisher also work to find the preserved content.

An institution will be able to include the contents of the cache among the pages indexed by its local search engine, and provide its readers with searching across all the journals to which it subscribes. At present, readers typically have to search individual collections of journals separately.

Technology

Overview

The LOCKSS technology implements a peer-to-peer network of persistent web caches. The caches proactively crawl the web to collect new content as it is published. Unlike normal caches, they are never flushed. The caches cooperate to detect and repair any damage automatically, without human intervention. The cached content is perpetually audited; the archive is never 'dark'. Content can be in any format delivered via HTTP, provided that the content is relatively immutable. HTML pages from e-journals typically contain dynamic content, such as ads. References to these materials are preserved, but the targets of these references are not. This means that the HTML collected by different caches can be different, so the HTML is filtered before being compared between caches. The readers get what the publisher published, but the comparisons are based only on the words the authors wrote.

A community in which a set of like-minded organizations runs a LOCKSS cache targeting the same content can be mutually assured that their local community will each continue to be able to access this content. The content will be accessible from the originally published URLs even after it has ceased to be accessible from the original publisher. The more organizations that run caches, the higher the level of assurance each obtains.

Cooperation and Community

The LOCKSS software is Open Source; the first beta version of the protocol is available on SourceForge, http://www.sourceforge.org/. As with any open-source software system, no one can prevent the emergence of variants. But the LOCKSS software implements a network protocol. Variants that are not interoperable will not be functional; they will not have 'lots of copies' to talk to, and will thus be unlikely to succeed. Variants that are interoperable will be welcome. Diversity of this kind offers Darwinian evolution to better fit a changing environment, and avoids monoculture vulnerabilities. Each version of the software will have different strengths and flaws. There will be economic and social pressure to keep each new version of the software interoperable with the system as a whole. The bigger the base of users, the greater that pressure will grow. As the size of the network grows, there will be a base from which to build open standards, as actual software rather than paper documents.

Libraries and publishers can collaborate on applications and further developments neither could achieve alone. One area of potential collaboration will be to build 'plug-in modules'. Each online publishing platform will require a 'plug-in' module (a downloadable Java program that contains all information about the structure of and data for particular content). Larger publishers may choose to put the 'plug-ins' for their content on the journal web site. Librarians may choose to deposit 'plug-ins' for smaller, more esoteric publishers in a repository. The current project intends to design a plug-in API so that someone with reasonable programming skills (comparable to a computer science graduate student) can tailor a plug-in for any particular journal.

Program History and Status [(3)]

The LOCKSS technology has been undergoing increasingly severe testing since 1999. The alpha test ran through 2000, and an early beta version was successfully deployed to 50 libraries worldwide from 2000 to 2002. (A list of participating libraries is available, http://lockss.stanford.edu/projectstatus.htm.) At these sites it has run with little operator intervention for nearly a year.

The Stanford University LOCKSS Program team is now building production software. The key redesign of the production software is the introduction of a publisher plug-in module. The publisher plug-in module will tailor the processes of collecting, preserving and providing access to a particular e-journal allowing the LOCKSS software to be more flexible and efficient. Testing of the next release of software is scheduled to begin late 2002.

The Stanford University LOCKSS Program team, with library staff from Emory University, Indiana University and the New York Public Library, is addressing a myriad of questions surrounding collection development, collection management, and collection access. One set of issues revolve around new workloads, workflows and processes stemming from the power to build rather than rent e-journals. Another set of issues are the placement of LOCKSS caches into widely varying local network environments and making sure local readers can access the stored content.

LOCKSS software is Open Source and is freely available. No fees are required from any party to use the LOCKSS software to archive content. In theory, the LOCKSS system is decentralized and does not require coordination. In practice, for a sustainable distributed e-journal archival system, some coordinating program infrastructure is indicated both for software development and support and for coordination of collection management initiatives. This coordinated program infrastructure will be provided by a LOCKSS Alliance. The Alliance is planned as a for-fee service organization for both librarians and publishers. Fees will provide tangible coordination services and community driven development for technology and collections. Detailed services and the fee structure are currently under discussion.

Community Participation

Please join us! The LOCKSS Program is soliciting library and publisher participation to fulfill the promise that "Diffused Knowledge Immortalizes Itself" in the digital age . Contact the author for additional information.

References

(1) Sir James Mackintosh 1765-1832

(2) http://lockss.stanford.edu/

(3) Support for LOCKSS: The National Science Foundation [this material is based upon work supported by the National Science Foundation under Grant No. 9907296, however any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation]; The Andrew W. Mellon Foundation; Private donations; Stanford University Libraries.

Author Details

Vicky Reich

Director LOCKSS Program
Stanford University Libraries.
Email: vreich@stanford.edu