LOCKSS: A Distributed Digital Archiving System

Progress Report for the Mellon Electronic Journal Archiving Program

Stanford University Libraries
8 October 2002

Table of Contents

Introduction

With funds from the Andrew W. Mellon Foundation, Stanford University validated the LOCKSS software and protocol through a rigorous beta test conducted from January 2001 to August 2002. The software is licensed as open source and is available for download from http://sourceforge.net/projects/lockss/. The success of the beta test can, in large part, be measured by community support and enthusiasm; over fifty libraries and over forty publishers are currently participating in the program. In addition, for the immediate future, all three of the project's funding agencies — the National Science Foundation, Sun Microsystems, and the Mellon Foundation — continue to support ongoing work.

Key Accomplishments

The LOCKSS model, which is based on analysis of cultural continuity epitomized by "Lots of Copies Keeps Stuff Safe," creates low-cost, persistent digital "caches" of e-journal content at institutions that 1) subscribe to that content and 2) actively choose to preserve it. Accuracy and completeness of LOCKSS caches are assured through a peer-to-peer polling system, operated through the Library Cache Auditing Protocol (LCAP), LOCKSS' communication protocol, which is both robust and secure. The creation of such caches, given the requirement that the caching library already have the right through subscription to obtain that content, has met with a high degree of publisher and library engagement and commitment.

1. **Technology:** Through its technical development and beta testing (1999-2002), the LOCKSS project has demonstrated that its model and protocol are technically viable.

    - The beta system has been deployed successfully to more than fifty sites around the world.

    - Fifty sites have run with little operator intervention for nearly a year. The average site has spent about an hour a month dealing with its cache. Almost all of this time has been on cache-level problems rather than journal-level problems, e.g.: because a cache was unplugged from the network or from power, the cache needed an IP address change, or the packet filters between the cache and the Internet were changed.

    - The test sites have been happy with the support provided by the LOCKSS team. The team has spent one to two person-days per week supporting the beta sites.

    - The beta test implementation successfully collected and preserved e-journal content, both from static mirrors and from dynamic "clones" of the online editions of PNAS (*Proceedings of the National Academy of Sciences*), BMJ (*British Medical Journal*), and *Science*.

- The system detected and repaired both deliberate and accidental damage.
- The system survived hardware failures, network outages, and attacks by "bad guys."
- Experience with the current system and analogies with such environments as Google's caching architecture (a collection of several thousand PCs that caches the entire accessible Web), give us confidence that the system can scale to many terabytes of journal content.
- Experience with the current system and analogies with the Gnutella peer-to-peer file sharing system give us confidence that the system can scale to many thousands of LOCKSS caches without generating excessive network traffic. The LOCKSS protocol should avoid the scaling problems of the Gnutella protocol.[1]
- The system preserved content in a wide range of formats and delivered it unchanged to readers who directly accessed the beta test caches.

2. **Libraries and Publishers:** Support from the library and publishing communities has been gratifying. Involved in the alpha test were six libraries and one publisher. For the beta test, more than fifty libraries worldwide are running LOCKSS caches and several dozen more are waiting to come online. More than forty publishers are listed on the Web site as LOCKSS program supporters. Many of these publishers are "supporting the project in principle" while some have committed to grant needed permissions for full deployment of the system once production software is released.

3. **Intellectual Property:** Ropes and Gray, the IP law firm retained by Stanford University, consulted with us on the legal implications and requirements of the LOCKSS system. Ropes and Gray recommended that publishers be required to provide two kinds of permission: for libraries, written permission to cache and archive content; for the LOCKSS caches, "machine readable permission" to collect and preserve content. This second "machine" permission fulfills Digital Millennium Copyright Act (DMCA) obligations.[2]

   For LOCKSS to work as designed, publishers will need to grant libraries blanket permission to use the LOCKSS software. We recommend adoption of the following (or similar) language in subscription agreements:

   > *Publisher acknowledges that Licensee participates in the LOCKSS system for archiving digitized publications. Licensee may perpetually use the LOCKSS system to archive and restore the Licensed Materials, so long as Licensee's use is otherwise consistent with this Agreement. Licensee may also provide its digital copies of Licensed Materials to other LOCKSS systems in support of the overall preservation and restoration purposes of LOCKSS, so long as any other LOCKSS system demonstrates it has the rights to the Licensed Materials necessary to access and copy them.*

Currently LOCKSS caches collect content as it is published so permission must be granted at the point of subscription. Our intent is to avoid the need for negotiations with each library for each title. This language grants permission for libraries to:

- Hold copies of subscribed (or otherwise authorized) materials
- Use cached material consistent with original subscription terms
- Provide access to the local community
- Provide copies for audit and repair to other caches only if they have had a copy in the past

Publishers must give the LOCKSS crawler permission to slowly crawl, collect, and cache content. We ask that this permission be granted through a Web page that lists at minimum the top level URLs for whatever is the "archival unit" of a title. We call this Web page the "publisher manifest." The LOCKSS system will work more efficiently as the "publisher manifest" becomes more detailed (for example, article level file description for each issue/volume). We are urging publishers to provide in this manifest the front matter information not usually included in the electronic journal, such as the editorial board, author instructions, etc. This is not an intellectual property concern, but will help ensure collection and preservation of complete content.

4. **Economics:** The LOCKSS software is open source and freely available for download from http://www.sourceforge.org/projects/lockss/. 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 advised both for software development and support, and for coordination of collection management initiatives.

   During the period of this grant award, we verified through hour-long interviews with most of the participating organizations that libraries and publishers are interested in working together through a for-fee service organization to pay for tangible LOCKSS technology, collection development services, and management coordination services. The business plan is under development.

Lessons Learned

Beyond validating that the LOCKSS model is viable, the beta test has revealed, as hoped, a number of strengths and weaknesses to the initial technical design. We now need to perfect the technology and to establish the LOCKSS model as an ongoing, operating archival solution based on the knowledge and insight gleaned over the past twenty months.

1. **Technology:** Much additional work is needed to produce an "appliance" based on the LOCKSS technology that can be used by a community of libraries as a sustainable e-journal archiving system. In particular, we determined we need to build a set of content-specific plug-in modules that drive the processes of collecting, preserving, and providing access to specific e-journals. Each "online publishing platform" will require a separate plug-in module (for example, one for HighWire Press titles, one for Blackwell Synergy titles, etc.). This will entail rewriting the existing Java daemon to segregate all potentially journal-specific knowledge behind a set of Java interface definitions (i.e., an API) that can be implemented by downloadable Java classes. This software will be designed to use whatever journal-specific information is available to make more efficient the searches for new content and for damage to preserved content by:

    - Exploiting knowledge of the e-journal's URL structure to target the search for newly published content

    - Exploiting knowledge of the e-journal's URL structure to drive the checking process and target the search for damage

    - Using knowledge of the e-journal's HTML formatting to assist the comparison process by filtering out variable content such as advertisements

    - Mapping between bibliographic information, URL, and file names for content

From the development point of view, the platform or system on which e-journals are mounted is critical. The addition of an e-journal to a library's collection is dependent on the presence of an appropriate plug-in for the technology supporting that e-journal. While it is more efficient to develop plug-ins for widely-used platforms rather than idiosyncratic, one-off or few-title platforms, there is a parallel need to acquire competence and to embrace smaller and/or less sophisticated titles.

2. **Libraries and Publishers:**

For LOCKSS to work in production, publishers must provide written permission to libraries to cache and archive content and "machine-readable permission" to the LOCKSS caches to collect and preserve content. It is our challenge and the challenge of librarians worldwide to obtain these permissions from publishers. The LOCKSS Alliance (under development) is designed to assist with this process.

The LOCKSS software provides libraries a tool for building local digital collections. Local use of the LOCKSS software will impose tasks and responsibilities on the collection development, technical services, and system librarian staffs. At minimum, libraries must develop and implement collection development decisions and then make locally cached content available to their local communities of readers. As libraries choose ever-larger numbers of e-journals for preservation, they will need collection management tools and support for the LOCKSS administrative interface to interoperate with collection management programs. In our current LOCKSS Mellon

grant, Indiana University is leading efforts to assess collection management user needs, to specify data flows between LOCKSS caches and collection management systems, and to build a "proof of concept" prototype.

3. **Economics:** Both libraries and publishers have expressed interest in participating in a new organization supporting the LOCKSS Program. We proceed with "enthusiastic realism" to build a LOCKSS Alliance.

Immediate Plans

With continued support from the Mellon Foundation, Stanford University Libraries is endeavoring to build a production archive system for electronic journals. Mellon Project Officer Don Waters has publicly stated our charge succinctly:

During the next phase of development, the key issues for the LOCKSS system are to separate the underlying technology from its application as an e-journal archiving tool; explore ways of ensuring the completeness and quality of e-journal content on acquisition and of managing the content as bibliographic entities rather than simply as Web-addressed files; expand the coverage of journals; maintain the LOCKSS software; and identify strategies for migrating the e-journal content. To help undertake and finance these tasks, Stanford has identified a variety of partners and is planning the development of a LOCKSS consortium.[3]

With our partners (Emory University, Indiana University, and the New York Public Library), Stanford University Libraries intends to construct a process so the community can drive functional specifications, design a general set of query/response interactions so the functional specifications can be implemented within most library technical environments, and as possible, prototype one potential implementation of collection management software.

Continued Sun Microsystems and National Science Foundation support will allow us to continue core technology development, focusing on the peer-to-peer, fault-tolerant aspects of the system.

Postscript: 2003

The development of the LOCKSS system continues to move forward at a considerable pace. For current status please see the LOCKSS Web Site at http://lockss.stanford.edu or contact vreich "at" stanford "dot" edu (vreich@stanford.edu).

Endnotes

[1] See the Gnutella Protocol Specification v0.4 online at http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf.

[2] See http://www.loc.gov/copyright/legislation/dmca.pdf.

[3] See Donald Waters, "Good Archives Make Good Scholars: Reflections on Recent Steps Toward the Archiving of Digital Information," *The State of Digital Preservation: An International Perspective: Conference Proceedings*, Publication 107 (Washington, DC: Council on Library and Information Resources, July 2002). Online at http://www.clir.org/pubs/reports/pub107/waters.html