Architectural Choices in LOCKSS Networks

David S. H. Rosenthal
Stanford University Libraries

Abstract

The LOCKSS digital preservation technology collects, preserves and disseminates content in peer-to-peer networks. Many such networks are in use. The Global LOCKSS Network (GLN) is an open network with many nodes in which libraries preserve academic journals and books that they purchase. The CLOCKSS network is a closed network, managed by a non-profit consortium of publishers and libraries to form a dark archive of e-journal content. There are also many Private LOCKSS Networks (PLNs) preserving various genres of content.

Each of these networks is configured to meet the specific requirements of its community and the content it preserves. This paper describes these architectural choices and discusses a development that could enable other configurations.

1 Introduction

The LOCKSS (Lots Of Copies Keep Stuff Safe) digital preservation technology collects, preserves and disseminates content in peer-to-peer networks. Many such networks are in use. The Global LOCKSS Network (GLN) is an open network with many nodes in which libraries preserve academic journals and books that they purchase. The CLOCKSS network is a closed network, currently of 12 nodes, managed by a non-profit consortium of publishers and libraries to form a dark archive of e-journal content. There are also many Private LOCKSS Networks (PLNs) preserving various genres of content.

Each of these networks is configured to meet the specific requirements of its community and the content it preserves. We describe these architectural choices and discuss a development under investigation in a number of countries that could enable alternative configurations that might be more suitable in some circumstances.

2 Existing Networks

To illustrate the differences between the various networks we describe three examples, the GLN and the CLOCKSS network, both preserving e-journals and e-books, and Data-PASS, a PLN preserving social science data. Social science data has to be preserved in conformity with one set of laws, governing for example the storage and dissemination of personal information. E-journals and e-books have to be preserved in conformity with a different legal framework, copyright law. This is one major reason for the differences between the configurations of these networks.

2.1 The Global LOCKSS Network

The LOCKSS technology was invented in 1997 as a response to librarians' concerns about long-term access to the journal content to which they subscribe [10]. The transition of this content from paper to the Web, while greatly improving the reader's experience, forced libraries to switch from owning a copy of the content to leasing access to the publisher's copy. Future readers' access would be at the mercy of the publisher's whim.

The idea behind the LOCKSS technology was to replicate in the Web world the way that the system worked in the paper world. Each library would run a LOCKSS box, the Web analog of its stacks. The library's box would obtain a copy of the subscribed content from the publisher and keep it for as long as the library wanted. If, for any reason, the library's readers were unable to access the publisher's copy they would instead be provided access to the library's copy.

In principle, the engineering to implement this model was simple. The Web pipeline from the publisher to the reader's browser already contained a number of Web caches with temporary copies of Web content. All that was needed was to build a Web cache with two additional properties:

- It would be *pre-loaded*; content would arrive in the cache not because a human had requested their browser to display it, but because a specialized Web crawler designed to exhaustively visit every page of each subscribed journal had requested it. This was necessary to ensure that the library acquired a copy of all the content to which it subscribed, much of which the library's readers would never actually read. In the paper world libraries had a process called "claiming the serials" in which staff ensured that the library had received every issue of every journal they purchased.

- It would be *persistent*; content that arrived in the cache would stay there instead of being displaced by more recent content. This was necessary to ensure the library continued to own a copy of the content to which it subscribed; it meant that the cache would grow steadily in size as new content was published. In the paper world journal content gradually accumulated along shelves in the stacks.

Even in 1998, Web crawlers and persistent storage were off-the-shelf techniques. The innovative part of the LOCKSS technology was to combine them in a way that minimized the cost to a library of deploying the technology. It did so in two ways: by reducing the up-front cost of obtaining permission from the publisher for the system to make and keep copies of their copyright content, and by reducing the capital and operational costs of subsequently keeping those copies safe from harm.

2.1.1 Obtaining Permission

In the US the Digital Millennium Copyright Act (DMCA) means that libraries and, more to the point, the developers of the software, would be criminally liable if they kept a copy of content obtained from the Web longer than specified by the HTTP Cache-Control header. This threat was the most important factor in the architecture of the LOCKSS software.

There are two ways to avoid the DMCA liability. Web archives such as the Internet Archive use the "safe harbor" provision, under which they are required to delete the copy if the copyright owner sends them a take-down notice[1]. This provision would not provide libraries adequate assurance against disputes with journal publishers.

The alternative is to obtain permission from the publisher. A system in which each library had to negotiate individually with each journal would clearly be too expensive to be viable. The architecture of the GLN in which each library's LOCKSS box obtains its content from the publisher using the library's subscription access avoided this. The LOCKSS team negotiated with the publishers on behalf of all libraries using the system. By amortizing one negotiation across all the libraries wishing to preserve a publisher's content, the system was economically viable even for small and open access publishers. It did not have to charge the publishers fees to have their content preserved.

Very early in the LOCKSS Program's history it was one of six initial e-journal projects funded by the Mellon Foundation. The other five all envisaged a central, third-party archive obtaining a copy from the publisher and supplying it to a library's reader if, for example, the library's subscription had lapsed. Obtaining permission to take the publisher's content and re-publish it for the financial benefit of the third-party archive proved very difficult. Eventually, the difficulty was overcome with the advent of the Portico archive [9].

On the other hand, because the LOCKSS system required each library to use the content they collected only for their own readers, publishers understood that this did not compete with them or materially increase the risk of leakage. This made the negotiation much easier, and less expensive.

Publishers agreeing to their journals being preserved put a statement on the journal's Web site, visible only to readers with a valid subscription, granting permission. Each LOCKSS box, before attempting to collect any content from the journal, checks that the permission statement is visible, and preserves it along with the content to which it relates as evidence that those copies were made with permission.

In this way, the collected content at each library serves as a database showing which libraries had valid access to which content when, in addition to its use in providing the library's readers with access if they cannot obtain it from the publisher. There is no need for an external database of access rights. This was essential, since at that time (and in most cases even now) no such database was available.

2.1.2 Preserving the Copies

Libraries subscribe to journals that serve their readers' interests. Because these interests overlap, library subscriptions overlap. The overlap is greater for important journals, and lower for less important journals. As each library kept its own copy of the content to which it subscribed, the system as a whole had many copies of important journals, and fewer for less important journals. But overall, in comparison to the levels of replication found in other reliable distributed systems, the LOCKSS system had an abundance of replicas of the content it was preserving.

Both because libraries' resources are under stress, and

[1] In practice, anyone representing themselves as the copyright owner can do this; the archive lacks the resources to verify ownership.

because the system envisaged an abundance of copies, it was important to minimize the per-copy cost of the system. As the number of replicas in a system increases, the effect on the reliability of the system of an individual replica's failure decreases. Because the GLN had a large number of replicas, it did not need each replica to be very reliable in order to achieve a high level of overall reliability. A novel sampled voting protocol was developed [8] which performed three functions:

- Boxes were enabled to detect whether their copy matched or did not match the consensus of the other boxes. This serves two purposes:
  - Web crawling is not a completely reliable means of collection. A mutual comparison of each box's independent collection from a particular publisher allows each box to identify missing items and repair these omissions by targeted re-crawling.
  - Data storage is not a completely reliable process. Content in a box that no longer matches the consensus of the other boxes is likely to be damaged and needs to be repaired.
- Boxes were enabled to prove to each other that their copies were the same.
- Boxes were enabled to request a copy from another box to replace a damaged copy. The request would be granted only if the requesting box had in the past proved to the requestee that their copies were the same. This mechanism ensures that each box obtains its content in the first instance from the publisher, and that the repair process does not leak content.

Other than this repair mechanism GLN boxes are permitted only to disseminate content to their library's own readers.

2.1.3 Risks and Resources

One result of this architecture is that the more important the journal the more preservation resources are devoted to it. This sounds sensible until you realize that the more important the journal the lower the risk that a library will cancel its subscription or that the publisher will cease to publish it.

More than 8 years of production use of the GLN demonstrates another aspect of this problem. Most important journals are published by large publishers, as one of many journals sharing a single publishing platform. Thus one publisher negotiation and one technical development effort results in the preservation of a large number of journals. But because these large publishers have a strong business model, their journals are at low risk of loss.

The less important journals are typically published individually from an idiosyncratic Web server. Thus one negotiation and one technical effort results in one journal being preserved. The cost per journal is thus much higher. Not merely are fewer resources devoted to the journals at higher risk, those resources are less effective in preserving them.

It is important to note that although this problem was first revealed in the LOCKSS context, it is shared by all e-journal preservation efforts. Any negotiation is per-publisher, and any technical effort to preserve a journal is at least per-publishing-platform, so fewer journals per publisher or platform is more cost per journal.

2.2 CLOCKSS

In 2005 a small group of librarians and publishers came together at the American Library Association conference to discuss a community governed archive. The librarians came from a culture of cooperation; although the publishers were competitors they had experience of cooperating for the good of the whole, for example in CrossRef [4]. They suggested that an effort, jointly funded and managed by both libraries and publishers, to use the LOCKSS technology to build a dark archive of journal content would be useful. This became the CLOCKSS archive [2].

The goal of the CLOCKSS archive is to maintain a complete archive of community publisher's content. The archive is dark in the sense that content is never released while it is available from any publisher. Community publishers agree that, if the content is no longer available for an extended period, the CLOCKSS board can vote to "trigger" it. Triggered content is extracted from the archive and re-published on the Web under a Creative Commons license [3], and thus made freely available to all. Examples of content triggered in this way can be found at `http://www.clockss.org/clockss/Triggered_Content`.

These different goals led to a different architecture for the CLOCKSS archive from the GLN:

- Because the nodes in the network had to be trusted by the publisher and libraries, the number of boxes was limited to 15, of which 12 have been authorized. For political and robustness reasons, these boxes were placed in different countries around the world.
- Because there were a limited number of CLOCKSS boxes each had to contain the complete content of all participating publishers, and thus needed to be much larger than the typical GLN box, which had to

contain only the content to which that library subscribed.

- Because there is no need to collect and preserve the information as to which institution had access to which content when, there was no need for each box to obtain its content directly from the publisher. Instead, content is collected by a small network of "ingest" boxes which hold a temporary copy while they agree on what is to be collected. The result is then collected by the CLOCKSS boxes around the world for final preservation. This reduces the load on the publisher's web servers and improves operational efficiency.

- Because all boxes have the same content, the restriction that a box provide repair content only to another box that had proved in the past to have the same content is not needed. Instead, all communication between boxes is protected by SSL certificate checks at both ends. Since each box has cryptographic proof that the box to which it is communicating is a valid member of the CLOCKSS network, it is safe for it to supply that box with any of its content.

- Because the CLOCKSS archive is dark, readers do not access content from the CLOCKSS boxes. The only form of dissemination from CLOCKSS is, upon a trigger event, the extraction of content for eventual re-publishing elsewhere.

2.3 Data-PASS

The Data-PASS is a collaboration among presently six institutions that runs a PLN [5] to archive, catalog, and preserve data used for social science research. The PLN is closed. Some members run multiple boxes and others do not run a box, so it is partly hosted. The first PLN, the MetaArchive, is similar in these respects [11].

Unlike the CLOCKSS PLN, each Data-PASS box holds a subset of the total network content, as specified by a replication policy. Access to the content (other than that needed to implement the replication policy) is provided only to the depositing institution, except in the eventuality that the institution fails in its responsibility to make the content available to the research community.

The Data-PASS network has developed a monitoring and audit tool that measures the PLN's conformance with the replication and preservation policies in place, called *SafeArchive* [1]. This tool is being applied to other PLNs.

3 Choices

3.1 Open vs. Closed

The initial goal was for the GLN to be an open network which any library could join simply by running a LOCKSS box. The software has always been open source and there was no barrier to them doing so. This led the design of the LOCKSS protocol to incorporate a wide range of defenses against abuse [8, 7].

The "Red Hat" model of free, open source software with paid support has proved effective in many cases, and was chosen for the LOCKSS program. In order to motivate support payment, the GLN chose to make participation in the network a support event, and thus to in effect close the network. A library could collect new content while they continued to make support payments. If they stopped payment, they continued to have the content their box had already collected.

Other networks have followed suit, and are in their various ways closed. Open networks are much harder to manage, to defend, and to make economically sustainable.

3.2 Dark vs. Light

A truly dark preservation network is pointless, in that content would never be used; only eventual use can justify the resources devoted to preservation. Thus in practice all LOCKSS networks are some shade of gray. The CLOCKSS network and some of the PLNs are a rather dark gray, with content being made available to readers only at specified events. In CLOCKSS' case it is the unavailability of the content for an extended period. In the Data-PASS it is typically a request from the institution that deposited the content.

The GLN, on the other hand, is a rather light gray. Content in a library's LOCKSS box is always accessible to the library's readers, although under normal circumstances it will be delivered directly from the publisher rather than from the box. Normally it is only if a request for content from the publisher fails that the content is delivered from the box.

The choice of the appropriate shade of gray is specific to the individual network and the content it preserves. For example, the Data-PASS network preserves social science datasets. Since they contain personally identifiable information, special authorization processes are required for access.

3.3 Local vs. Hosted

A *local* LOCKSS network is one in which each participating institution runs its own LOCKSS box. A *hosted*

network is one in which the boxes are run by a third party.

The GLN is a local network, because each participating institution has to demonstrate its right to the content it preserves by obtaining it from the original publisher. The preserved copy encodes this right.

There are no fully hosted PLNs, but several PLNs including Data-PASS are partly hosted, with some participating institutions running one or more boxes on behalf of other institutions that do not. The CLOCKSS network can be viewed as partly hosted in this way.

3.4 Homogeneous vs. Heterogeneous Collections

In the CLOCKSS network each box contains exactly the same collection of content. It is a homogeneous network. In the GLN each box contains a different collection of content, depending on the particular set of journals to which the host library subscribes. It is a heterogeneous network, as is the Data-PASS PLN.

A homogeneous network is more effective at using computational resources. Boxes in a heterogeneous network must expend polling effort to locate the boxes containing other copies of the content to be compared. In a homogeneous network any other box will contain a copy. On the other hand, a homogeneous network is less effective at using storage resources. A copy in every box may not be necessary to achieve an adequate level of overall reliability.

3.5 Untrusted or Trusted Repairers

An essential property of the GLN is that it does not leak content from a LOCKSS box that is authorized to have it to a box that is not. Thus the requirement that a box deliver a repair only to another box that has in the past proved that it had the same content. This is the *untrusted* repairer model.

In the CLOCKSS network and in some other PLNs all boxes have the same content by design, thus there is no need for a box to maintain knowledge of which boxes have which content. Repairs may be delivered to any other box that is part of the network; the *trusted* repairer model. This trust depends on a strong mechanism for identifying boxes as part of the network. This is provided by an SSL keystore on each box, maintained out-of-band, with certificates for all boxes in the network. All connections from one box to another are verified by a certificate check at each end before any data is transferred.

The advantage of the trusted repairer model is that content ingested into the network achieves adequate robustness more quickly. There is no delay while the polling process establishes at each box holding it a set of proofs from other boxes that they hold the same content.

4 An Alternative Choice

Suppose some external database were to be available that could be treated as an authoritative source of the information as to which institution had access to which content when. This would trigger a cascade of architectural choices:

- There would be no need for the boxes in a LOCKSS network to discover a library's access rights by probing the publisher and then preserve that information.
- There would be thus no need for each library to run its own box; a smaller network of boxes could collect and preserve the content. A smaller number of somewhat more reliable boxes could have a lower total cost.
- As with the CLOCKSS network, this smaller network could use the trusted repairer model.
- As with the CLOCKSS network, there would be no need for each box in the network to collect content directly from the publisher. It is more efficient both for the publisher and the network to use a CLOCKSS-like set of "ingest" boxes to agree on the collected content before it is transferred to the boxes for permanent preservation.
- If a reader could not get access directly from the publisher their library would request access from a network box, supplying some reader authentication via, for example, Shibboleth [6]. The box would query the external database using the authentication and the identity of the requested content to seek permission to satisfy the request.

Some countries are setting up such "rights databases". Doing so provides two significant opportunities:

- The potential from a model in which each library runs its own LOCKSS box to an outsourced or hosted model in which a small, centrally managed network provides access to content unavailable from the publisher. This could provide greater efficiency and lower cost, but would certainly reduce the barrier to entry and the effort needed at most libraries.
- A rights database could receive information from libraries as to the content they have purchased, and from publishers as to the content they have sold to

libraries. The database could verify these two independent sources against one another, a process that in the paper world libraries used to call "claiming the serials"; ensuring that they actually obtained the content for which they had paid.

Note that with the advent in journals of optional author-pays open access, the process that in the paper world could operate at the level of a journal volume must now operate at the level of individual articles, most likely identified by their DOI.

5 Conclusions

Configuring a network of LOCKSS boxes involves a set of design choices. The existing LOCKSS networks have made some sets of these choices; others have not been implemented. The advent of third-party rights databases, trusted by publishers and libraries in a domain to supply real-time access permission, allows a different configuration of LOCKSS network to provide readers in that domain with access to preserved content if it is not available from the publisher for any reason. This network could be hosted rather than requiring each participating library to run its own LOCKSS box, reducing the barrier to entry and potentially lowering the overall cost.

References

[1] ALTMAN, M. Auditing distributed preservation networks. In *CNI Fall 2012 Membership Meeting* (Dec. 2012).

[2] CLOCKSS. CLOCKSS: A Trusted Community-Governed Archive. `http://www.clockss.org/`.

[3] CREATIVE COMMONS. Web site. `http://creativecommons.org/`.

[4] CROSSREF. crossref.org. `http://www.crossref.org/`.

[5] DATA-PASS. About Data-PASS. `http://www.data-pass.org/`.

[6] ERDOS, M., AND CANTOR, S. Shibboleth Architecture DRAFT v05. `http://shibboleth.internet2.edu/docs/draft-internet2-shibboleth-arch-v05.pdf`, May 2002. XXX update this.

[7] GIULI, T., MANIATIS, P., BAKER, M., ROSENTHAL, D. S. H., AND ROUSSOPOULOS, M. Resisting Attrition Attacks on a Peer-to-Peer System. In *Usenix Annual Technical Conf.* (Anaheim, CA, USA, Apr. 2005).

[8] MANIATIS, P., ROUSSOPOULOS, M., GIULI, T., ROSENTHAL, D. S. H., BAKER, M., AND MULIADI, Y. Preserving Peer Replicas By Rate-Limited Sampled Voting. In *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles* (Bolton Landing, NY, USA, Oct. 2003), pp. 44–59.

[9] ROSENTHAL, D. S. H. A Brief History of E-Journal Preservation. `http://blog.dshr.org/2011/08/brief-history-of-e-journal-preservation.html`, Aug. 2011.

[10] ROSENTHAL, D. S. H., AND REICH, V. Permanent Web Publishing. In *Proceedings of the USENIX Annual Technical Conference, Freenix Track* (San Diego, CA, USA, June 2000), pp. 129–140.

[11] SKINNER, K., AND SCHULTZ, M., Eds. *A Guide to Distributed Digital Preservation*. Educopia Institute, 2010.