Live Archiving Proxy
Project Closure

June 19, 2013

Ina - Dépôt Légal du Web

Table of content

1) Project summary

Initial goal

The Live Archiving Proxy project is a collaboration between **Ina** and **Netarkivet.dk** to build an HTTP proxy that is able to capture the traffic that flows trough it, and delegate the handling of the captured data to a writer using a simple network protocol. The goal is to be able to write the captured traffic into any kind of archive format using any computer language.

Schedule

The project was selected in response to the IIPC 2012 Call For Project. It was scheduled to start on September 2012 and end in March 2013. The project start was delayed until January 2013 but the scheduled project duration (6 month) was met.

Planned deliverables

The initial project deliverables as described in the project proposal are:

- A standalone package that contains the Live Archiving Proxy software (via GitHub);
- The source code of the Live Archiving Proxy (via GitHub);
- A generic Writer Plugin library in Java (via Maven);
- The source code of the generic Writer Plugin library (via Maven);
- A documentation for the Live Archiving Proxy;
- A WARC Writer Plugin library in Java;
- The source code of the WARC Writer Plugin library;
- A documentation for the WARC Writer Plugin;
- A compiled report of use-cases and feedback from designated partners (current report).

Looking ahead

All goals from the initial project have been successfully met. Requests have been made that some additional features should be included in the LAP (e.g. HTTPS) and to the WARC writer (e.g. write to different WARC files at the same time). Ina and Netarkivet.dk are already working on the next versions of the LAP and the WARC Writer.
The initial aim of the project was to make it easier to crawl “modern” Websites. These Website do not fit into the old one-crawler-fits-them-all model. To address this, a tool to enable easier integration of multiple crawlers in the archiving workflow was needed . The LAP and the WARC Writer enable this. The next step is to build new crawlers, some specialized for media content (Youtube), some specialized for social networks (Facebook, Twitter), some able to crawl through paywall.
Building and deploying these crawlers, and using them to generate standard and reliable WARC files should be easier thanks to this project.

2) Deliverables description

LAP

The code of the Live Archiving Proxy has been released on GitHub:
https://github.com/INA-DLWeb/LiveArchivingProxy
The latest binary for the Live Archiving Proxy can be downloaded here:
https://github.com/INA-DLWeb/LiveArchivingProxy/raw/master/lap.tar.gz

Generic writer

The generic writer plugin has been released on Maven:

- Repository: https://oss.sonatype.org/content/repositories/snapshots/
- Group ID: fr.ina.dlweb
- Artifact ID: lap-writer-generic
- Version: 2.0-SNAPSHOT

The following artifacts are available:

- source code
- compiled JAR
- compiled standalone JAR (including dependencies).

WARC writer

The WARC writer plugin has been released on Maven:

- Repository: https://oss.sonatype.org/content/repositories/snapshots/
- Group ID: fr.ina.dlweb
- Artifact ID: lap-writer-warc
- Version: 1.0-SNAPSHOT

The following artifacts are available:

- source code
- compiled JAR
- compiled standalone JAR (including dependencies).

User guide

A small user guide has been released along with the LAP, on GitHub:
https://github.com/INA-DLWeb/LiveArchivingProxy/raw/master/LAP-UserGuide.pdf

3) British Library feedback

The Live Archiving Proxy is a potentially extremely useful solution for the British Library, as we are expecting to have to archive a wide range of important sites that are not immediately amenable to conventional crawling methods. In many cases, this is because of the complex or dynamic nature of the sites, but the most pressing issues are around harvesting content from behind paywalls.

Ideally, paywall harvesting would be carried out during our usual crawling procedures. However, the variation in login and session mechanisms, and the overheads involved in quality assurance of the results, all mean that we are unlikely to have a mature solution in the near term. For high-profile content that cannot wait, a live archiving proxy provides a way of ensuring the most critical items can be captured safely and soon, by getting a human operator to negotiate the variable complexities and using the LAP to record the outcome.

In this way, the LAP would play a critical role in the development of improved automatic crawling methods, under a continual process of improvement. For any new site that requires specific effort to ensure quality and/or negotiate paywalls, we can the LAP to provide a ‘ground truth’ archive of a particular set of resources by recording the manual browsing process. This process could initially be used to capture all important material, while in parallel, the ‘ground-truth archives’ can be analysed in order to understand how to extend the automated crawling processes to cover each case, and when that is done, to validate the automated methods. Once the automated process has run in parallel for a while, the manual work can shift over to QA of the automated output.

So far, we have only run limited, small-scale tests using the LAP. It performed well, and the resulting WARCs were validated using the Hanzo and JWAT tools, and could be played back successfully using our Wayback Player tool.

The main limitation we found was the lack of SSL support. The main concern I have is that, for our use case, we are expected to limit the ways in which we store the usernames and passwords we use to negotiate paywalls. These credentials may well end up embedded in the WARC files, and we would have to audit the system to ascertain if this could be avoided or stripped out.

4) Internet Memory Foundation feedback

Live Archiving HTTP Proxy evaluation

Internet Memory Foundation feedback

1° Deployment and installation

IMF experienced on its internal infrastructure the usage of the Live Archiving Proxy developed by INA. The goal was to participate to the evaluation of the tool, by defining specific use cases and performing some illustrative tests.

The deployment of the tool was achieved without any difficulties, since it can be used as an off-the-shelf application (a single executable linux 64bit binary file called "lap") with default configuration. The same for the two dedicated writers that we used: the default Print writer and the WARC writer developed by the Danish National Internet Archive. The only detail that needs to be checked is that the listening ports for the proxy and the writers are open (by default, ports 4338 and 4365). Another useful functionality is represented by the dashboard Web application that allows to monitor if the Web traffic is correctly detected by the proxy.

2° Use cases

In the purpose of improving the capture of Web content and the overall quality of the Web archive, IMF identified two use cases suited for the usage of the Live Archiving Proxy.

The first use case is relatively simple and straight forward, tackling the harvest of complex Flash applications. More precisely, to enhance the harvest capabilities of Heritrix there where the Flash objects are completely opaque in terms of embedded content. The typical example is represented by the Flash animations or games that load on-the-fly additional resources (text, images, documents, etc.) progressively as the user browses through the application and triggers different actions. For each action, the Flash object directly accesses the Web server and requests for new content. This conversation is completely missed out by the crawler, since it only fetches the initial state of the Flash object. In order to improve the completeness of the harvest, IMF performs a quality assurance process after the crawl. The missing resources are manually identified and patched into the archive, but the global workflow is rather time consuming and involves additional operations.

For this particular cases we successfully experienced the Live Archiving Proxy to detect and fetch the missed images contained by the Flash application.
We identified an illustrative example on the Museum of London website:
http://www.museumoflondon.org.uk/si/iwb_view.asp?title=KS2%20Tudor%20quiz%20level%201&theme=default&colourTheme=default&presentationMode=quiz&dataSource=KS2_Tudor_quiz_level_1.xml

It represents an interactive quiz where new images are loaded each time a new level is reached. Browsing through out the pages of the quiz and having an archiving proxy running on the test machine we could obtain the list of the images URLs (using the print writer), as well as the WARC file generated for this content (using the WARC writer), ready to be added to the archive.

The evaluation of the Live Archiving Proxy involved also a comparison with a crawl performed with Heritrix on the same URL. The analysis of the crawl logs proved that none of the images in the Flash application were harvested by Heritrix.

This example clearly shows that using the Live Archiving Proxy the quality of the harvest can be thus improved and it can substitute additional operations to fetch missing Web content.

The second use case defined by IMF is aiming to the usage of the Live Archiving Proxy together with a JavaScript extractor tool developed by IMF. This tool enables the execution of JavaScript code embedded in the html pages, in order to detect new URLs that can not be identified by simply parsing the html content of the page. It currently works as a Web service that retrieves a list of discovered URLs for a given URL of a page. This URLs list can be afterwards inserted into the frontier of a running instance of Heritrix, so that they are processed and harvested by the crawler.

Using the Live Archiving Proxy together with the JavaScript extractor Web service could enable a direct retrieval of Web content for the discovered URLs, as well as the generation of WARC files using the WARC writer.

We therefore experienced some tests to adapt the JavaScript extractor for the interaction with the proxy. The implementation for recording the queries (for the discovered URLs) is an ongoing work at IMF.

3° Wish list of improvements

Based on these recent experiences, we could also identify several additional functionalities that would be very useful from IMF point of view:

- the usage of an ARC writer, for the legacy systems managing ARC files
- the possibility to define distinct sessions with specific parameters: e.g. the name of the WARC/log files to be set according to a given collection

5) Netarkivet.dk feedback

Live Archiving Proxy - Netarkivet.dk test feedback

Contents

1 Introduction

As testing partners we have tested the **LAP** according to how it can/will be used in our institution.

Since we use **NetarchiveSuite**/**Heritrix1** we have not specifically tested **LAP** with material which we are already able to harvest, although this is not completely avoidable.
Instead we have focused on websites and material which for some reason is causing us problems.

2 Material identified as problematic by Netarkivet.dk

Generally websites involving **Javascript**/**AJAX** and user interaction is problematic for **Heritrix**.
This is most problematic when trying for harvest comments on

- YouTube
- Twitter
- Facebook
- Myspace

2.1 Youtube

With this tool is it possible to harvest and show old and new youtube videos and all comments using **IE 10** (with **shockwave** plugin disabled).
Playback is possible in **Wayback** if you know the correct Target-Uri's!

2.2 Twitter / Facebook

Unable to test since **LAP** does not currently support **https**.

2.3 Myspace

To the best of my recollection, the comments are harvested as **AJAX JSON** or **HTML** data depending on which browser is used.
However this is not replayable in **Wayback**.

3 Material of interest for inhouse researchers

We asked some of our researchers if they could supply us with some example sites which are currently not being harvested satisfactorily. They are currently being partially harvested by our **NAS**/**H1** workflow.

3.1 Procedure

LAP was tested using a small selection of different browsers to ascertain the impact on the harvested data, if any.

3.2 Usecases

Remember: You can't use a browser over **VPN**, it just hangs. (Seems to be caused by ads)
I have used following browser types: **Firefox 18,19.***, **IE 8,10**.

3.2.1 http://jyllands-posten.dk/kultur/article3980279.ece

It was not possible for me to harvest this site in any way, because some part of the site uses a foreign german server. It just hangs on that server - perhaps because of blocked firewalls.

Nicholas succeeded in harvesting the 2 video's by restarting the WARC writer one time and with a linux **Firefox v. 19** on the server with a **realplayer** plugin. He had to extract the the streams from the WARC file and use a **VLC** player to verify that they were valid **flv** and **mp4** files. With the knowledge of the Target-Uri it was possible replay the streams in **Wayback** using **VLC**-player plugin.

The quality of the streams were rather low.

(Note: These problems were maybe caused by **VPN** and similar problems)

3.2.2 http://www.youtube.com/watch?v=G1XRg11kgxM

Do not use **shockwave flash** plugins in your browser, it splits the video stream up into chunks with different Target-Uri's and you can not replay the video in **Wayback**. It was only possible to harvest every sort of youtube videos from 2008 to now with **IE 10** with disabled **shockwave** plugin. If you use an older **realplayer** plugin in **Firefox 19.*** you can only harvest the old videos.

Here is what youtube writes about support regarding **html5**:
http://www.youtube.com/html5

The low quality video was saved with a very long and cryptic Target-Uri with a lot of different parameters. Some of them seem to be about the quality, so it seems to be possible to get a better quality, too. (not tested)

During the test (by clicking on the link more comments) I discovered that all comments were also harvested under different Target-Uri's:

http://www.youtube.com/watch?v=qjNvC_wu0cg
http://www.youtube.com/all_comments?v=qjNvC_wu0cg
http://www.youtube.com/all_comments?v=qjNvC_wu0cg&page=2

Wayback could replay all target-Uri's using **IE 10 flash plugin** or **Firefox 19.*** but the web-page button for the video or the link for more comments did not work because of **Javascript** and **AJAX** communication.

3.2.3 http://www.myspace.com/nephunited

When I tried to get all comments, it was only possible to get comments as "**Javascript**" like txt with a different URL. **Wayback** could show the comments using the different URL, but the webpage button for more comments did not work because of **Javascript** and **AJAX** communication.

3.2.4 dr.dk/musik

It was not possible to get any of these flash based streams.

3.2.5 http://efterklang.net/home/videos/

It was possible to harvest the video's using **Firefox 19.*** with older **realplayer** plugins and **VLC** but not with **IE 10**!

3.2.6 tv2.dk videos

It was only possible to harvest tv2.dk videos using **IE 8** with disabled **shockwave** plugin.

4 Issues

Some important issues:

- Missing support for https: It is not possible (nor does Wayback) to use the tool for harvesting facebook.com, twitter.com or other very common domains which use https (Advanced)
- Crawllog: Missing a "crawllog" or similar information when trying to browse for harvested Url's in Wayback. There is no link between the harvested video Url and the video play button on the webpage. You need access to the Target-Uri's in the warcfile to verify what is actual saved in the warcfile and for Url replay in Wayback. So do not start using deduplication in the warc writer before you know what you are harvesting.
- Client IP: During parallel tests with two different browser versions on to different IP's of the same video site, I was missing identification for which Target-Uris came from which IP in the warcfile.
- Multiuser: Different people should be able to harvest to separate WARC files for different projects using the same LAP / writer. (Advanced)
- Flash apps: It is not possible to harvest other types of flash apps like flash videos on dr.dk
- Browser plugins: Depending of which material is being harvested it is important to be mindfull of the impact different plugins can have on the result.

5 Prefered usage

At KB/SB we would prefer to be able to have one LAP instance running instead of possibly having different instance running at the same time. The reason we could be tempted to run several instance at the same time is because we do not harvest continiously into the same WARC files. So we would like to harvest on a per project basis across the whole organization, into separate WARC files.

Another reason is access to servers which is preferably restricted to technical personel whereas curtators and research personel prefer higher level access.

One way to solve this would be for technical personel to be responsible for servers and keeping LAP and writers running. Curators and researchers should have easier access to define which client IP gets stored into which WARC file and possibly even have a workflow for uploading the file for ingest or simpler making files available for download.

One way to acomplish this could be a simple webinterface for the WARC writer.

6 Feature requests for LAP

- Client and destination IP: I'm guessing the A record is still queried even though a frontier proxy may be used.
- Discarding data: Command Line option to discard data when no writer is attached.
- **https**: At some point the in the near future it would be nice with **https** support, most likely as a future enhancement after this projects conclusion.

These are ofcourse subject to prioritization according to everyones whishes and time allocated.

7 Feature requests for LAP WARC writer

- Crawlog: As long as the **referer http** header is present when supplied by the client it would probably be best to include crawllog functionality in the WARC writer, especially if the following requests are to be implemented.
- Client IP based WARC file writing: Write to different WARC files based on Client IP as a way to separate harvested data by project/user.
- Simple webinterface for WARC writer: Simple webinterface to re-assign WARC file(s) for a Client IP.

These are ofcourse subject to prioritization according to everyones whishes and time allocated.

8 Ending remarks

This is an important tool to lookup some of the many technical problems we have today.
Use it, if you want to know more about harvesting difficult web domains, but remember you need a fair amount of technical skills.