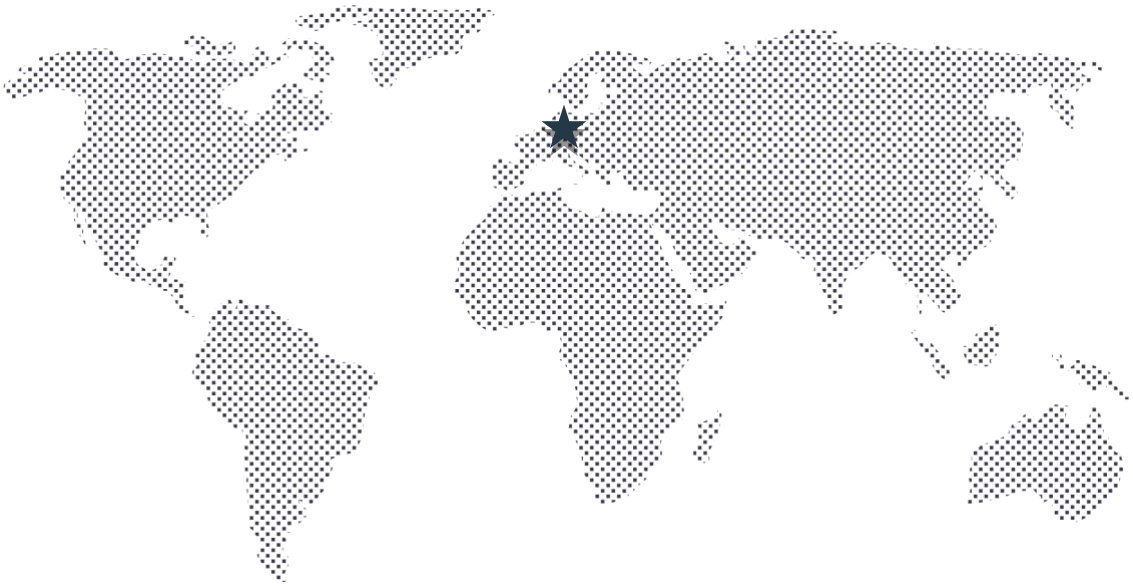
Proceedings *of the*



2022

Web Archiving & Digital Libraries Workshop

June 24, 2022

Virtual Event

Martin Klein

Mat Kelly

Zhiwu Xie

Edward A. Fox

Editors

WADL 2022 Homepage

Web Archiving and Digital Libraries -- a Virtual Workshop in conjunction with JCDL 2022

Date: **June 24, 2022**

We welcome broad attendance; please contact the co-chairs for any questions you may have.

Please see the approved WADL 2022 workshop description from the JCDL proceedings as well as the workshop page hosted by the conference.

Please also refer to past WADL homepages: 2020, 2019, 2018, 2017, and 2016. Past workshop proceedings can be found from: WADL 2017-19, Pre 2016.
Prior workshops have led in part to a special issue of the International Journal on Digital Libraries.

Registration

Since WADL is hosted by JCDL, at least one author per paper must register at least for the workshop at: the JCDL registration page.
JCDL 2022 is a hybrid event:
you can participate on-site for the conference or remotely or just for the workshop (see Satellite Events).

Schedule (using EDT)

```
== Opening Session (Moderator Martin Klein) ==
9 Welcome, Introductions, Tech Ironing (everyone speaks!)

== Talks 1 (Moderator Martin Klein)  ==
9:30am Invited Talk 1: Karolina Holub (see below for details)
10:10 Discussion

== Talks 2 (Moderator Martin Klein)  ==
10:30 Where are the Datasets? A case study on the German Academic Web
Archive
Yousef Younes, Sebastian Tiesler, Robert Jäschke and Brigitte Mathiak

10:45 Comparison of Access Patterns of Robots and Humans in Web
Archives
Himarsha Jayanetti, Kritika Garg, Sawood Alam, Michael Nelson and
Michele Weigle

10:50 Wayback Machine Video Archiving Insights
Sawood Alam, Bill O'Connor and Mark Graham

10:55 Optimizing Archival Replay by Eliminating Unnecessary Traffic to
Web Archives
Kritika Garg, Himarsha Jayanetti, Sawood Alam, Michele Weigle and
Michael Nelson

11:00 Discussion

== Talks 3 (Moderator Zhiwu Xie) ==
11:20 Emulation-based long-term Access to Complex Web-sites
Marcel Tschöpe, Rafael Gieschke and Klaus Rechert

11:35 Web Archiving as Entertainment
Travis Reid, Michael Nelson and Michele Weigle
```

11:40 First steps in Identifying Academic Migration using Memento and Quasi-Canonicalization
Mat Kelly, Deanna Zarrillo, Christopher Jackson and Erjia Yan

11:45 Discussion

12:05 (Lunch) Break

== Talks 4 (Moderator Mat Kelly) ==
13:00 Invited Talk 2: Carrie Pirmann and Erica Peaslee (see below for details)
13:40 Discussion

== Talks 5 (Moderator Mat Kelly) ==
14:00 CDX Summary for Web Archival Collection Insights
Sawood Alam and Mark Graham

14:15 Russia-Ukraine News on the Dark Web
Grant Atkins, Aaron Buehne, Abby Mabe, Zak Zebrowski and Justin Brunelle

14:20 Archiving Source Code in Scholarly Content: One in Five Articles References GitHub
Emily Escamilla, Talya Cooper, Vicky Rampin, Martin Klein, Michele Weigle and Michael Nelson

14:25 Discussion

== Talks 6 (Moderator Ed Fox) ==
14:45 15m Arch-It
Helge Holzmann, Nick Ruest, Jefferson Bailey, Alex Dempsey, Samantha Fritz, Ian Milligan and Kody Willis

15:00 WACZ
Ed Summers, Ilya Kreymer and Cade Diehm

15:05 Moving the End of Term Web Archive to the Cloud to Encourage Research Use and Reuse
Mark Phillips and Sawood Alam

15:20 Discussion

Closing Session (Moderator Ed Fox)
15:40 Closing Discussion (publication and other collaboration opportunities, next event planning)
16:30 end

Invited Talks

1. Karolina Holub, Library Adviser, Croatian Digital Library Development Centre, Croatian Institute for Librarianship
Title: A history of web archiving at the National and University Library in Zagreb
Abstract: The National and University Library in Zagreb (NSK), as a memory institution responsible for collecting all types of resources, early recognized the significance of collecting and preserving web resources as part of its core activities. In 2004, the NSK developed, in collaboration with the University of Zagreb University Computing Centre (Srce), the Croatian Web Archive (HAW). The NSK is using three different approaches and tools to archive the Croatian web. At the beginning, only selective archiving of web resources was conducted. In order to build a more comprehensive national collection, crawls of the whole national domain (.hr), thematic, and event crawls followed a few years later. This talk will present the chronology of working processes and diverse ways the NSK attempts to preserve Croatian web as a contemporary part of the cultural and scientific heritage.

Bio: Karolina Holub is a coordinator of the Croatian Digital Library Development Centre at the Croatian Institute for Librarianship in the National and University Library in Zagreb. Her field of work includes developing, implementing and maintaining digital library systems (Croatian Web Archive, Digital Collections of the National and University Library in Zagreb, Croatian electronic theses and dissertations repositories etc.) as well as taking care of metadata harmonization and interoperability with other systems for all types of resources. She is involved in managing and participating in the development of the Library's digitization projects and thematic portals, and is involved in several national and international projects.

2. Carrie Pirmann (Bucknell University) and Erica Peaslee (Centurion Solutions LLC)
Title: Building a Community of Web Archivers: The Race to Save Ukrainian Cultural Heritage Online
Abstract: In response to Russia's invasion of Ukraine on 24 February 2022, over 1300 cultural heritage professionals—librarians, archivists, researchers, programmers came together to archive the web presence of Ukraine's cultural heritage. In the proceeding 4 months, SUCHO (Saving Ukrainian Cultural Heritage Online) has digitally preserved over 40 TB of websites, databases, and other digitized cultural property to hold in trust for Ukrainian colleagues while they are working to preserve their heritage on the ground. This talk will cover the basics of coming together in a distributed grassroots response, the evolution to collaborating with heritage responders and using open-source information to guide efforts, implementing a workflow across 14+ timezones, and utilizing the Webrecorder suite of tools developed by Ilya Kreymer. We hope that the processes and lessons learned from this path-breaking project can be used to assist with responses to similar archiving emergencies and help institutions preemptively establish similar methods for future use.
Bio: Carrie Pirmann is the Social Sciences Librarian at Bucknell University (USA), working at the intersections of information literacy instruction, research support, and digital scholarship in the social sciences. She holds a master's degree in library science from the University of Illinois, and has put her years of experience as a librarian to use for SUCHO by conducting extensive research to locate cultural heritage sites online that need to be archived, and working the Situation Monitoring team to keep abreast of situations in critical areas of Ukraine.
Bio: Erica Peaslee is the Administrative Operations Coordinator at Centurion Solutions LLC, a Disaster and Emergency Management consultancy in Texas (USA) where she also provides subject matter expertise regarding cultural heritage. Using her background in museum collections and her graduate education in Museum Studies (Harvard), she is particularly interested in centering cultural property in emergency planning and resilience, and promoting communication between the two communities. Erica currently serves as Situation Monitoring Coordinator for SUCHO, leading the observation and coordination of using real-time information from Ukraine to direct efforts to the most at-risk areas. In addition, she also works with other professionals at the intersection of cultural heritage, crime, and emergency response to coordinate and facilitate working towards similar goals.

Submissions:

- Paper length: 3-5 pages for a 15 minute presentation
- Paper length: 1 page for 5 minute lightning talk

- **Due date: May 1, 2022 AoE - CLOSED**
- Notifications: mid-May

- Submit to: Easychair submission system
- Please use the ACM Proceedings template

Description:

Due to the current state of the world, WADL 2022 will be held entirely online.

Please note though that JCDL 2022 is currently planned as a hybrid event and we encourage all WADL attendees to also register and attend JCDL.

WADL 2022 will continue the WADL tradition to provide a forum and collaboration platform for international leaders from academia, industry, and government to discuss challenges, and share insights, in designing and implementing concepts, tools, and standards in the realm of web archiving. Together, we will explore the integration of web archiving and digital libraries, over the complete digital resource life cycle: creation/authoring, uploading, publishing on the web, crawling/collecting, compressing, formatting, storing, preserving, analyzing, indexing, supporting access, etc.

WADL 2022 will cover all topics of interest and specifically invite contributions from practitioners. Topics include but are not limited to:

- Event archiving and collection building
- National and international perspectives on web archiving
- Social media archiving
- Community building
- Ethics in web archiving
- Archival metadata, description, classification
- Archival standards, protocols, systems, tools
- Crawling of dynamic, online art, and mobile content
- Discovery of archived resources
- Diversity in web archives
- Extraction and analysis of archival records
- Interoperability of web archiving systems

Objectives:

- Continue to build the diverse community of people integrating web archiving with digital libraries
- Help attendees learn about useful methods, systems, tools, and software in this area
- Help chart future research and practice in this area, to enable more and higher quality web archiving
- Promote synergistic efforts including collaborative projects and proposals
- Produce an archival publication that will help advance technology and practice

Workshop Co-chairs:

- Chair: Martin Klein, Scientist, Los Alamos National Laboratory Research Library, mklein@lanl.gov
- Co-chair: Mat Kelly, Assistant Professor, Drexel University, College of Computing and Informatics, mkelly@drexel.edu
- Co-chair: Zhiwu Xie, Professor & Chief Strategy Officer, Virginia Tech Libraries, zhiwuxie@vt.edu
- Co-chair: Edward A. Fox, Professor and Director Digital Library Research Laboratory, Virginia Tech, fox@vt.edu, http://fox.cs.vt.edu

Program Committee:

- Brunelle, Justin F., The MITRE Corporation, jbrunelle@mitre.org
- Duncan, Sumitra, Frick Art Reference Library, duncan@frick.org
- Finnell, Joshua, Colgate University, jfinnell@colgate.edu
- Goethals, Andrea, National Library of New Zealand, Andrea.Goethals@dia.govt.nz
- Jones, Shawn, Los Alamos National Laboratory, smjones@lanl.gov
- Ko, Lauren, UNT Libraries, lauren.ko@unt.edu
- Lyon, Meghan, Library of Congress, mlyon@loc.gov
- McCown, Frank, Harding University, fmccown@harding.edu
- Nelson, Michael, Old Dominion University, mln@cs.odu.edu
- Nwala, Alexander, Indiana University, alexandernwala@gmail.com
- Rampin, Vicky, New York University, vicky.rampin@nyu.edu
- Risse, Thomas, University Frankfurt, University Library J. C. Senckenber, t.risse@ub.uni-frankfurt.de

- Taylor, Nicholas, Los Alamos National Laboratory, ntay@lanl.gov
- Weber, Matthew, Rutgers University, matthew.weber@rutgers.edu
- Weigle, Michele, Old Dominion University, mweigle@cs.odu.edu
- Wrubel, Laura, Stanford University, lwrubel@stanford.edu

Where are the Datasets? A case study on the German Academic Web Archive.

Yousef Younes
Yousef.Younes@gesis.org
GESIS – Leibniz institut for Social Sciences
Cologne, Germany

Sebastian Tiesler
sebastian.tiesler@hu-berlin.de
Humboldt-Universität zu Berlin
Berlin, Germany

Robert Jäschke
robert.jaeschke@hu-berlin.de
Humboldt-Universität zu Berlin
Berlin, Germany

Brigitte Mathiak
brigitte.mathiak@gesis.org
GESIS – Leibniz institut for Social Sciences
Cologne, Germany

ABSTRACT

The German Academic Web (GAW) is a longitudinal archive of websites from German academic institutions, mainly universities. It can support answering research questions about academia in Germany. Recent discussions about reproducible research have brought the availability and sharing of research data into focus. Collecting, linking, and providing metadata about research data is thus an important task for infrastructure facilities. In this work, we examine how existing datasets are linked and referenced on German academic web pages using the GAW archive. For that, we use the social sciences and economics datasets registered at da|ra as our case study. The results show that academic web pages as presented in GAW are not a good foundation to answer dataset-related questions. But from the few results found, it was obvious that da|ra datasets are usually mentioned using their DOIs and not their URLs.

KEYWORDS

research data, data findability, web archiving, German Academic Web

ACM Reference Format:
Yousef Younes, Sebastian Tiesler, Robert Jäschke, and Brigitte Mathiak. 2018. Where are the Datasets? A case study on the German Academic Web Archive.. In *Proceedings of Web Archiving and Digital Libraries Workshop (WADL).* ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Re-using research data has become an important driver of scientific innovation. Aggregating, replicating, or applying different methods to existing data leads to new insights and increases the quality of the underlying research results, while lowering costs [6]. As research data becomes more important, so does archiving information on research data. From an observation study, we know that relevant information on research data is not limited to the data itself, but also on a variety of web sites, such as project websites [8].

The web is also an important information source for researchers looking for data. According to a survey conducted among 1,637 researchers from all disciplines [4] 59 % use web search engines to find data often. Other surveys, for example, among social scientists [3] confirm that web search is a very important part of their data discovery process. As such, archiving the research data itself is not enough, its traces in the web need to be archived as well to get a full picture.

This leads to the following research questions:

(1) How can a web archive be used to find references to research datasets?
(2) Which identifiers for datasets can be found?
(3) How does the volume of referenced datasets change over time?

As these questions in their totality cannot be answered easily, due to scaling effects, we are instead focusing on a specific subset: datasets from social sciences and economics as registered through the registration agency da|ra[1]. The web archive The German Academic Web[2] is employed to answer these questions.

This paper is structured as follows: In Section 2 we discuss related work and in Section 3 we explain the methods we have used for our experiments. The results are presented in Section 4, followed by a discussion in Section 5.

2 RELATED WORK

Research data is traditionally stored in databases or data repositories and only recently opening up to web infrastructure, for instance, through the application of the FAIR principles [14]. Schema.org added *Dataset* as an entity type in 2013, which can be used to provide metadata on research data through markup. This is used, for example, by Brickley et al. to build catalogues [2]. But they observed that metadata markup, although it is rather simple, it needs proper curation, as not every *Dataset* entity is describing a dataset [1]. However, not all data repositories adhere to this recommendation yet [10]. Instead, metadata is represented on plain web pages. Thompson et al. did a longitudinal analysis of Common Crawl data [13] to find out about the use of persistent identifiers. They suggest

[1] https://www.da-ra.de/
[2] https://german-academic-web.de/

WADL, June 20, 2022, Virtual

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

to use DOIs over URIs to identify scholarly publications on the web. Like metadata, persistent identifiers come with their own set of problems. For DOIs it is critical to maintain the mapping between DOI and resource location over time and to deliver a consistent response to DOI queries [7].

3 METHOD

In this work, we track mentions of dataset identifiers in an academic web crawl. Particularly, we search for datasets from social sciences and economics registered at the da|ra registration agency in the German Academic Web (GAW). The search process is performed using two dataset identifiers against multiple GAW snapshots. First, we describe the GAW data and how it is collected (Section 3.1). Then we introduce the da|ra system (Section 3.2). Finally, the experimental setup is explained (Section 3.3).

3.1 German Academic Web

The *German Academic Web* (GAW) [11] is a collection of snapshots of German academic institutions' web sites. It is a domain-specific longitudinal web archive and was created to preserve the websites of German academic institutions. By the time of this writing, GAW contains nineteen snapshots obtained by crawling on a biannual basis since 2013 in addition to one snapshot from 2012. Each of these snapshots occupies about 6-8 TB of storage and involves around 100 million breadth-first crawled web pages (text, PDF, and images) stored as WARC files which in turn contain several WARC records. Every crawl is performed using a recent version of the Heritrix[3] web crawler initialised with a seed list of 150 domains associated with all German academic institutions who have the right to award doctorates. The characteristics of the crawling process change over time, for example, a new domain could be added to the seed list if a new university is created or one URL could be retired if it was found to be out of scope (e.g., an e-learning system or a file repository) [11]. These changes are ignored in this paper, because one of the goals is to find which web pages are most likely to contain dataset mentions.

The experiments are conducted on a collection of crawls that consist of the mid-year crawls from 2016 to 2021. Earlier crawls were omitted as da|ra was not fully online prior to 2016. From these crawls, only web pages whose content is text and which were available at the time of crawling are selected. This is achieved by choosing WARC records with MIME type text/html and HTTP status code 200.

3.2 da|ra

The availability of research data is a precondition to make the research results reproducible. To help achieve this availability for social sciences and economics data, GESIS[4] (Leibniz Institute for the Social Sciences) and ZBW[5] (Leibniz Information Centre for Economics) launched da|ra, in 2014, as a registration agency [9]. Beside its registration and archiving services, da|ra offers the metadata of its registered datasets for harvesting through the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH)[6]. It also provides a DOI resolver service.

In this work we are interested in quantifying identifier mentions of the da|ra registered datasets in the GAW archive. We use identifiers available in da|ra for its registered datasets as part of their metadata. da|ra offers multiple identifiers in their metadata. We can differentiate between two types of identifiers: *mandatory* and *optional*. The mandatory identifiers are available for every dataset at da|ra; while the optional ones are only available for some of them. The mandatory identifiers are:

DOI: Digital object identifiers are unique, permanent and case-insensitive strings of alphanumeric characters that are used to identify digital resources (books, research data, etc.) [12]. There are 26,298 unique DOIs registered at da|ra (e.g. 10.17886/RKI-History-0011) – one for each resource. Among these DOIs, there are 17,723 DOIs that are associated with datasets. The other DOIs are for different types of resources such as text, image, service, software, etc. and are not considered for our analysis. DOIs can be resolved to their associated URLs using a DOI resolver, for example, https://doi.org/.

URL: Every da|ra dataset has a URL associated with it. There are 25,312 unique URLs at da|ra. Among them 17,084 are dataset-associated URLs. While different versions of the same dataset have different DOIs, they have the same URL. This is why the number of URLs is lower than the number of DOIs, despite the fact that they are both mandatory.

Titles: For every dataset there is at least one title. For some of them there is more than one. These titles are mostly in English and German but alternative titles in other languages such Chinese, Arabic, etc. exist but they are very rare. Titles do not have to be unique and often contain additional information, such as year of collection, acronyms, which makes many of them rather unwieldy and hard to search for (e.g., "German General Social Survey (ALLBUScompact) - Cumulation 1980-2018").

The optional identifiers are:

URN: Uniform Resource Name is a unique and permanent identifier that uses the URN schema. Unlike DOI, URNs are only resolvable through the assigning institutions web page which make them useful for locally closed systems. da|ra has 345 items with such an identifier (e.g., urn:nbn:de:0168-ssoar-383499).

GESIS-specific identifiers: These identifiers come from the GESIS Archive. There are 6,428 datasets with such identifiers (e.g., ZA0790).

For the sake of this analysis, we focus on the da|ra resources with type *Dataset*. Each of these datasets has a couple of identifiers. Figure 1 shows a distribution of da|ra dataset identifiers. As we can see, every dataset has one URL and one DOI but one or more titles. About a third of the datasets are from GESIS and thus have its identifier. Based on that, we choose to use URL and DOI identifiers, since they are available and unique for every dataset.

[3] https://github.com/internetarchive/heritrix3/wiki
[4] https://www.gesis.org/
[5] https://www.zbw.eu/
[6] https://www.da-ra.de/oaip/

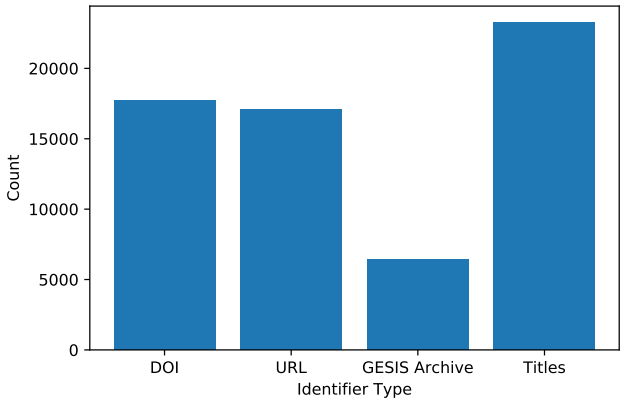


Figure 1: Distribution of the different types of dataset identifiers available in da|ra.

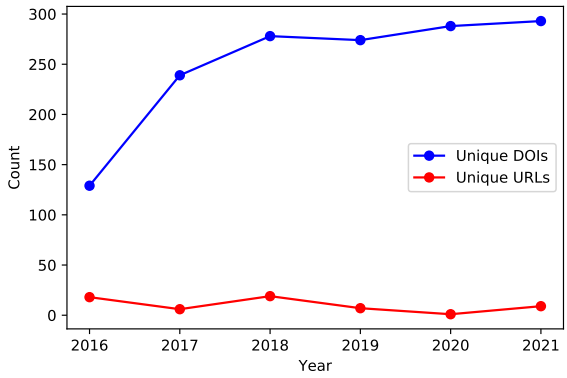


Figure 2: Distribution of the different types of identifiers in GAW over time.

3.3 Experimental Setup

Da|ra provides us with two lists of URL and DOI identifiers for its registered datasets. Our goal is to find web pages in GAW that mention da|ra datasets using the two chosen identifiers. After a preliminary analysis we performed some pre-processing to solve some of the problems we had identified.

First, we found that the research data repository hosted on madata.bib.uni-mannheim.de was archived in the GAW crawls. Since the repository contains a subset of da|ra datasets, we would trivially find all URLs from those da|ra datasets in that subset of GAW. As the repository should have been excluded by the crawl scope of GAW anyway, we exclude dataset URLs to that host from the list of da|ra dataset identifiers.

Then, similar to the results in [1], we also found that some da|ra dataset identifiers are not proper. It should be implicit that the URL identifier of a dataset points to a web resource containing the dataset but that is not always the case. For example, the DOI 10.5684/soep.v36-RV.RTBN2018 uses the landing page http://www.fdz-rv.de/ as resource for the dataset and a set of 48 different DOIs with prefix 10.25654 point to the same landing page https://www.hamburg.de/bsb/ifbq. Having such URLs and DOIs as dataset identifiers would erroneously increase the number of matching pages and thus we also excluded those URLs and DOIs from the list of da|ra dataset identifiers.

Finally, the URLs, DOIs, and crawled web pages have to be converted to lower case to prevent case sensitivity issues. The URLs also need to be normalised by removing common prefixes (https://www., http://www., https://, http//) and suffixes (.html, .htm). Then a simple string search is applied using ArchiveSpark [5] over the full text of the six selected web crawls for the two chosen identifiers. After that the results are analysed on different dimensions to quantify the unique identifiers, hosts, and pages to draw conclusion based on that.

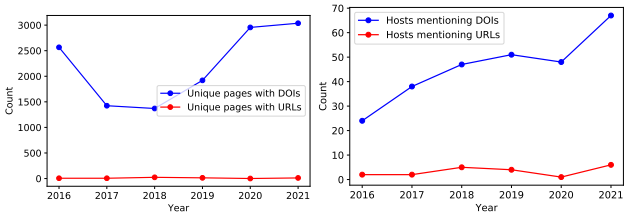
4 RESULTS

In Figure 2 we show the number of unique dataset URLs and DOIs found in six GAW crawls. Over the years, we observe only a small number of unique URLs and that number fluctuates in the range [1, 19], while the number of unique DOIs increases from 129 in 2016 to 293 in 2021. Although both numbers are small, it seems more common for GAW web pages to use DOIs when referring to datasets.

We also looked at the unique pages (Figure 3(a)) and unique hosts (Figure 3(b)). By 'hosts' we mean the first part of the URL up until the first slash excluding the http and www parts. Again, the number of unique hosts and pages that mention DOIs is higher compared to the ones that mention URLs. Both figures show an upward trend for DOI usage. Dips in the graphs could be the result of changes in the crawl scope or web pages retiring or moving their service. An example for a web page retiring its service in 2016 is the host dszbo-portal.uni-bielefeld.de. It is the "Datenservicezentrum Betriebs- und Organisationsdaten" ("Data Service Center for Business and Organizational Data").[7] In 2017 the new host fdzbo-portal.uni-bielefeld.de for "Forschungsdatenzentrum Betriebs- und Organisationsdaten" ("Research Data Center for Business and Organizational Data (RDC-BO)") comes into existence until it becomes part of DIW Berlin in 2019[8] and leaves the scope of the crawl (see also Table 1). Generally speaking, the number of hosts increases over the years which means that datasets are getting more common because they are being mentioned by an increasing number of different web sites.

Since the number of results for the URL identifier is (close to) zero, we analyse the found DOIs in more depth. Additionally, as the figures plot unique results and to show a different dimension of the results, Table 1 shows the hosts with more than 50 matching pages. Additionally, we added the number of unique DOIs mentioned in the pages per host. With this information, we can further evaluate if a host contains interesting information regarding our research question. Furthermore, comparing the number of results in this table to the number of unique pages in Figure 3(a) gives an indication of the amount of duplicate pages for each year.

[7] https://web.archive.org/web/20160321191513/https://dszbo-portal.uni-bielefeld.de/
[8] https://www.diw.de/de/diw_01.c.670982.de/



(a) unique pages (b) unique hosts

Figure 3: Distribution of the number of unique pages and hosts containing dataset identifiers over time.

Table 1: Top hosts for DOI mentions over the years. The colum *pages* shows the number of matching pages that come from the associated host. The column *uDOIs* shows the unique number of DOIs the host refers to.

year	host	pages	uDOIs
2016	dszbo-portal.uni-bielefeld.de	2500	59
	iqb.hu-berlin.de	180	23
	uni-bielefeld.de	113	35
2017	fdzbo-portal.uni-bielefeld.de	1297	62
	fb03.uni-frankfurt.de	348	77
	goethe-university-frankfurt.de	174	77
	iqb.hu-berlin.de	139	38
	uni-bielefeld.de	114	36
2018	fdzbo-portal.uni-bielefeld.de	1125	63
	fb03.uni-frankfurt.de	324	85
	iqb.hu-berlin.de	163	45
	goethe-university-frankfurt.de	162	85
	uni-bielefeld.de	123	36
2019	iqb.hu-berlin.de	1667	52
	fb03.uni-frankfurt.de	439	89
	mzes.uni-mannheim.de	197	48
	goethe-university-frankfurt.de	171	88
2020	iqb.hu-berlin.de	2755	56
	fb03.uni-frankfurt.de	204	93
	mzes.uni-mannheim.de	185	48
	goethe-university-frankfurt.de	101	94
2021	iqb.hu-berlin.de	2770	18
	fb03.uni-frankfurt.de	222	102
	mzes.uni-mannheim.de	200	48
	goethe-university-frankfurt.de	110	103
	uni-bielefeld.de	64	25

From the table, we see that the host name iqb.hu-berlin.de, which is associated with the Institute for Educational Quality Improvement in Germany, is among the top hosts in all involved crawls. This institute is involved in empirical educational research in Germany and also hosts a research data repository, so it references many of the social sciences datasets.

5 DISCUSSION

In this work we have searched for the social sciences and economical datasets registered at da|ra in GAW using two different identifiers. We were somewhat surprised to find only so little on datasets on German academic web pages, given that we know that people use web resources to find information on datasets extensively [8]. Nevertheless, there are lessons to be learned.

The takeaway points from this work can be summarised as follows. First, using DOIs to search for datasets produces better results than using URLs as identifiers. Second, there needs to be a curation mechanism for the da|ra metadata to validate whether, for example, URLs provided refer to those datasets and thus make the metadata more reliable. Third, since we were able to find only a few dataset mentions in GAW, we can say that either GAW is not including such pages or, which is more likely, that it is just not common to cite datasets on web pages. Future work should focus on finding a way to find or track datasets over the years and categorise them according to their importance. This suggests introducing a specialised crawl and the results obtained here could be used for that task.

ACKNOWLEDGEMENTS

Part of this research was funded by the DFG project *Unknown Data – Mining and consolidating research dataset metadata on the Web* (grant number 460676019).

REFERENCES

[1] Tarfah Alrashed, Dimitris Paparas, Omar Benjelloun, Ying Sheng, and Natasha Noy. 2021. Dataset or Not? A Study on the Veracity of Semantic Markup for Dataset Pages. In *The Semantic Web – ISWC 2021*, Andreas Hotho, Eva Blomqvist, Stefan Dietze, Achille Fokoue, Ying Ding, Payam Barnaghi, Armin Haller, Mauro Dragoni, and Harith Alani (Eds.). Springer International Publishing, Cham, 338–356.

[2] Dan Brickley, Matthew Burgess, and Natasha Noy. 2019. Google Dataset Search: Building a search engine for datasets in an open Web ecosystem. In *The World Wide Web Conference*. 1365–1375. https://doi.org/10.1145/3308558.3313685

[3] Tanja Friedrich. 2020. *Looking for data*. Ph. D. Dissertation. Humboldt-Universität zu Berlin, Philosophische Fakultät. https://doi.org/10.18452/22173

[4] K Gregory, P Groth, A Scharnhorst, and S Wyatt. 2020. Lost or Found? Discovering Data Needed for Research. *Harvard Data Science Review* 2, 2.2 (2020).

[5] Helge Holzmann, Vinay Goel, and Avishek Anand. 2016. ArchiveSpark. In *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries*. ACM. https://doi.org/10.1145/2910896.2910902

[6] Jonathan M Jeschke, Sophie Lokatis, Isabelle Bartram, and Klement Tockner. 2019. Knowledge in the dark: scientific challenges and ways forward. , 423–441 pages.

[7] Martin Klein and Lyudmila Balakireva. 2021. An extended analysis of the persistence of persistent identifiers of the scholarly web. *International Journal on Digital Libraries* (10 2021), 1–13. https://doi.org/10.1007/s00799-021-00315-w

[8] Thomas Krämer, Andrea Papenmeier, Zeljko Carevic, Dagmar Kern, and Brigitte Mathiak. 2021. Data-Seeking Behaviour in the Social Sciences. *International Journal on Digital Libraries* 22, 2 (2021), 175–195.

[9] Thomas Krämer, Claus-Peter Klas, and Brigitte Hausstein. 2018. A data discovery index for the social sciences. *Scientific Data* 5 (April 2018), 180064. https://doi.org/10.1038/sdata.2018.64

[10] Fidan Limani, Yousef Younes, Valentina Hiseni, Janete Saldanha Bach, Peter Mutschke, and Brigitte Mathiak. 2021. KonsortSWD Task Area 5 Measure 2 Report Scope: Milestones 1, 2, and 3. https://doi.org/10.5281/zenodo.5901207 Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) as part of NFDI - 442494171.

[11] Michael Paris and Robert Jäschke. 2020. How to Assess the Exhaustiveness of Longitudinal Web Archives: A Case Study of the German Academic Web. In *Proceedings of the 31st ACM Conference on Hypertext and Social Media (HT '20)*. ACM, New York, NY, USA. https://doi.org/10.1145/3372923.3404836

[12] Norman Paskin. 2010. Digital object identifier (DOI®) system. *Encyclopedia of library and information sciences* 3 (2010), 1586–1592.

[13] Henry S. Thompson and Jian Tong. 2018. Can Common Crawl reliably track persistent identifier (PID) use over time? https://doi.org/10.48550/ARXIV.1802.01424

[14] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific data* 3 (2016). https://doi.org/10.1038/sdata.2016.18

Comparison of Access Patterns of Robots and Humans in Web Archives

Himarsha R. Jayanetti
Old Dominion University
Norfolk, Virginia, USA
hjaya002@odu.edu

Kritika Garg
Old Dominion University
Norfolk, Virginia, USA
kgarg001@odu.edu

Sawood Alam
Internet Archive
San Francisco, California, USA
sawood@archive.org

Michael L. Nelson
Old Dominion University
Norfolk, Virginia, USA
mln@cs.odu.edu

Michele C. Weigle
Old Dominion University
Norfolk, Virginia, USA
mweigle@cs.odu.edu

ABSTRACT

In 2013, AlNoamany et al. of our research group (WS-DL, ODU) studied the access patterns of humans and robots in the Internet Archive (IA) using the Wayback Machine's anonymized server access logs from 2012. We extend this work by comparing these previous results to an analysis of server access logs in 2019 from the IA's Wayback Machine and the Portuguese Web Archive (Arquivo.pt). This comparison is based on robot vs. human requests, session data (session length, session duration, and inter-request time), user access patterns, and temporal analysis. We used a variety of heuristics to classify sessions as a robot or human, including browsing speed, loading images, requesting robots.txt, and User-Agent strings. AlNoamany et al. determined that in the 2012 IA access logs, humans were outnumbered by robots by 10:1 in terms of sessions, 5:4 in terms of raw HTTP accesses, and 4:1 in terms of megabytes transferred. The four web archive user access patterns established in 2013 are single-page access, access to the same page at multiple archive times, access to distinct web archive pages at about the same archive time, and access to a list of archived pages for a certain URL (TimeMaps). We are investigating whether similar user access patterns still persist, whether any new patterns have emerged, and how user access patterns have evolved over time.

KEYWORDS

Web Archiving, User Access Patterns, Web Server Logs, Web Usage Mining, Web Robot Detection

ACM Reference Format:
Himarsha R. Jayanetti, Kritika Garg, Sawood Alam, Michael L. Nelson, and Michele C. Weigle. 2022. Comparison of Access Patterns of Robots and Humans in Web Archives. In *Proceedings of Web Archiving and Digital Libraries (WADL '22).* ACM, New York, NY, USA, 1 page. https://doi.org/XXXXXXX.XXXXXXX

WADL '22, June 20, 2022, Virtual

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

Wayback Machine Video Archiving Insights

Sawood Alam
sawood@archive.org
Web and Data Scientist
Wayback Machine
Internet Archive
San Francisco, California, USA

Bill O'Connor
boconnor@archive.org
Crawl Operator
Wayback Machine
Internet Archive
San Francisco, California, USA

Mark Graham
mark@archive.org
Director
Wayback Machine
Internet Archive
San Francisco, California, USA

ABSTRACT

Wayback Machine has specialized video archiving pipelines, but it has been lacking visibility into what was being archived. We built a Video Archiving Insights dashboard, powered by rich metadata we collect, to address this opacity. We learned some lessons and used them as feedback loop to enhance our collections and operations.

1 INTRODUCTION

At the Internet Archive we archive various online media types to serve via Wayback Machine. Video archiving from some sources requires specialized workflows because they are often served as streams on non-deterministic URIs instead of static files. For example, a set of typical YouTube URIs would be curated based on certain criteria to identify potential videos that should be archived or excluded. Candidate video page URIs for archiving are placed in a queue to be consumed by a separate process. We maintain a persistent database of videos we have already archived, which is used both for status tracking as well as a seen-check system to avoid duplicate downloads of large media files that usually do not change. We then use *youtube-dl*[1] (or one of its forks) to download videos and their metadata. We archive the container HTML page, associated video metadata, any transcriptions, thumbnails, and at least one of the many video files with different resolutions and formats. These pieces are stored in separate WARC records (some with *response* type and others as *metadata*).

Like any large archival collection, our video archiving also turned into a large pile of ever-growing data. We felt the need of visibility into what we are collecting with the goal of knowing how can we do a better job of archiving more quality videos and utilizing our finite resources more strategically. Hence, we built a Video Archiving Insights dashboard, which is the focus of this document.

2 IMPLEMENTATION

We create a daily summary of metadata of videos that we have archived. We developed a Video Archiving Insights dashboard using the *Streamlit* library[2] in which we load daily metadata summary to identify any issues or biases. We inspect this insights dashboard regularly for quality assurance and to enhance our curation criteria to better utilize our resources. For example, we learned that less than 20% videos take up more than 80% of the total daily duration, which led us to realize that we might be archiving day-long streams, as a result we added some exclusion rules in our archiving process.

We first select a day using a calendar date picker, which loads necessary data to render insights for the selected day as shown in Figure 1(a). The dashboard reports highlights of the day, such as the number of videos, their total duration, number of unique channels, and the duration of the longest video for the day. It also reports the delta of these statistics with respect to the day before. We then show a word-cloud of the top-k (adjustable using a slider) tags, which gives a a sense of what was trending that day. For example, on February 24, 2022, we see Russia, News, Ukraine, Putin, Afghanistan, Pakistan, COVID-19, etc. in the top tags of videos archived that day.

The dashboard then renders a word-cloud using sumgrams[3] based on the titles of videos as shown in Figure 1(b). We have sliders to select both the number of top sumgrams to render as well as the minimum length of sumgrams. Then it shows some language-specific statistics, which may show some changes in trends when a specific country is in news. It is worth noting that the language information is not available in YouTube's metadata API, so we use language detectors to fill the gap and enhance our metadata. We first try to detect the language of the title, but if it results in English or a failure then we attempt to detect the language of the title and the description combined.

Next, it renders cumulative sorted duration and a sample of top-k longest videos as shown in Figure 1(c). This helps us identify less important, but long videos that we could have excluded in favor of many short or medium length videos to optimize resource utilization.

It then reports per-category statistics to learn about how much of our resources are consumed by different video categories as shown in Figure 1(d). For example, we may want to prioritize videos from the education and news & politics categories over gaming.

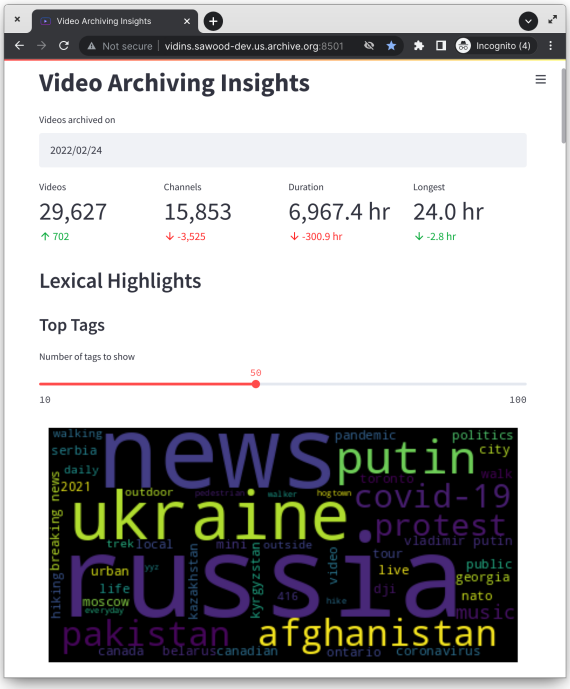
It also reports statistics on channels and some other miscellaneous aspects that we look at to minimize biases. We keep adding more reporting capabilities to the system as the needs emerge.

3 CONCLUSIONS AND FUTURE WORK

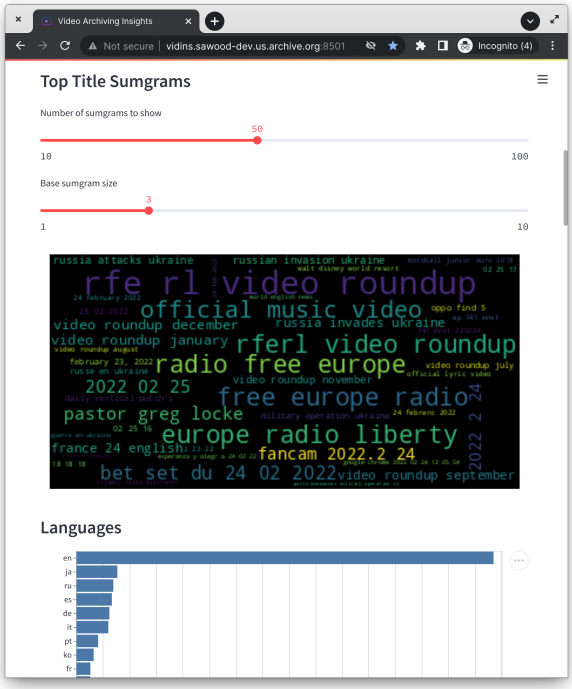
We built a Video Archiving Insights tool to bring more visibility into our video archiving operations and collections. We learned some data-driven lessons from the system and incorporated necessary changes to improve our crawl operations.

It is an evolving system, but we think it has got adequate reporting capabilities already that we can plan to open-source it soon. It requires specific metadata as input for reporting, which may not be collected by other web archives, in which case it will have limited utility for other archivists. However, we still think it will allow people to customize the system according to their needs and data availability.

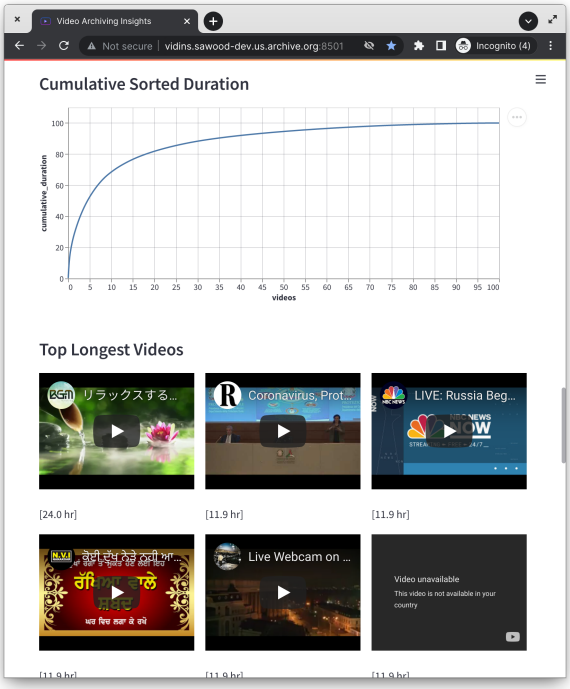
[1] https://github.com/ytdl-org/youtube-dl
[2] https://streamlit.io/
[3] https://github.com/oduwsdl/sumgram



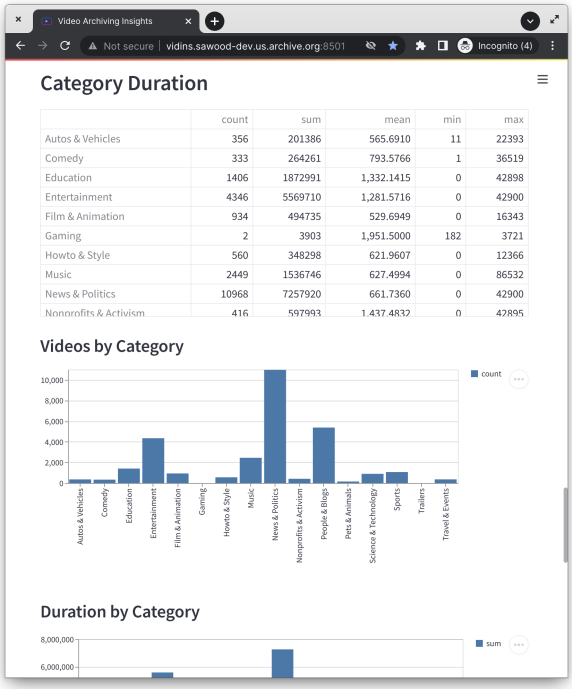
(a) Daily Highlights and Top Tags



(b) Sumgrams and Top Languages



(c) Cumulative Duration and Longest Videos



(d) Per Category Videos and Duration

Figure 1: Video Archiving Insights Portal

Optimizing Archival Replay by Eliminating Unnecessary Traffic to Web Archives

Kritika Garg
Old Dominion University
Norfolk, Virginia, USA
kgarg001@odu.edu

Himarsha R. Jayanetti
Old Dominion University
Norfolk, Virginia, USA
hjaya002@odu.edu

Sawood Alam
Internet Archive
San Francisco, California, USA
sawood@archive.org

Michele C. Weigle
Old Dominion University
Norfolk, Virginia, USA
mweigle@cs.odu.edu

Michael L. Nelson
Old Dominion University
Norfolk, Virginia, USA
mln@cs.odu.edu

ABSTRACT

We discovered that some replayed web pages cause recurrent requests that lead to unnecessary traffic for the web archive. We looked at the network traffic on numerous archived web pages and found, for example, an archived page that made 945 requests per minute on average. These requests are not visible to the user, so if a user leaves such an archived page running in the background, they would be unaware that their browser would continue to generate traffic to the web archive. We found that web pages that require regular updates (for example, radio, sports, etc.) and contain an image carousel, widget, etc., are more likely to make recurrent requests. If the resources requested by the web page are not archived, some web archives may patch the archive by requesting the resources from the live web. If the requested resources are not available on the live web, the resources cannot be archived, and the responses remain HTTP 404. Some archive pages would continue to poll the server as frequently as they did on the live web, while some pages would poll the server even more frequently, if their requests are 404, creating a high amount of unnecessary traffic. On a large scale, web pages like these could potentially cause security issues such as denial of service attacks. Significant computational, network, and storage resources are required for web archives to archive and then successfully replay pages as they were on the live web, and these resources should not be spent on unnecessary HTTP traffic. Our proposed solution is to optimize archival replay using HTTP Cache-Control response headers. We implemented a simplified scenario where we cache HTTP 404 responses to avoid these recurring requests.

CCS CONCEPTS

• **Information systems → Digital libraries and archives**.

KEYWORDS

Web Archiving, Digital Preservation, Archival Replay

ACM Reference Format:
Kritika Garg, Himarsha R. Jayanetti, Sawood Alam, Michele C. Weigle, and Michael L. Nelson. 2022. Optimizing Archival Replay by Eliminating Unnecessary Traffic to Web Archives. In *Proceedings of Web Archiving and Digital Libraries (WADL '22).* ACM, New York, NY, USA, 1 page. https://doi.org/XXXXXXX.XXXXXXX

WADL '22, June 20, 2022, Virtual

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

Emulation-based long-term Access to Complex Web-sites

Marcel Tschöpe
marcel.tschoepe@rz.uni-freiburg.de
University of Freiburg
Germany

Rafael Gieschke
rafael.gieschke@rz.uni-freiburg.de
University of Freiburg
Germany

Klaus Rechert
rechert@hs-kehl.de
University of Applied Science Kehl
Germany

ABSTRACT

LAMP web server setups have been a popular choice for the past 15 - 20 years. Especially, web sites with scientific content (e.g., project web sites, database front-ends) have still active users while their technical back-end reaches end of life. A migration of the content to contemporary systems is difficult, both due to the custom code base and the lack of resources. A crawler-based preservation is usually not able to capture the functionality of the service. This article proposes the use of emulation for preservation and long-term access to these sites. We present a scalable technical solution as well a case study of preserving 234 LAMP-based web sites in a highly automated way.

ACM Reference Format:
Marcel Tschöpe, Rafael Gieschke, and Klaus Rechert. 2022. Emulation-based long-term Access to Complex Web-sites. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

A typical web-site setup of the past 15 - 20 years was a LAMP setup - a Linux-based Apache web server with a MySQL database and PHP back-end for server-side scripting. This setup was in particular popular for scientific project web sites, which have been later replaced with WordPress and similar content management system (CMS) platforms. University's computing centers offered such a managed setup as a service for professorships, scientific projects, and the like. These services entailed a centrally managed LAMP stack, with the ability for tenants to upload and manage their content via FTP or SSH. Tenants were able to implement server-side scripts and utilize the database as needed. This lead to custom web applications, highly version specific implementations (e.g., PHP 5), which had to be migrated by each tenant if a LAMP stack reached end of life and had to be upgraded to an up-to-date version. This model turned out to be problematic, especially when LAMP services were about to be retired. In many cases, there are still outside users of the site and in some cases there is a pledge given to the project's funder to keep project result available for at least 10 years. Migration to new platforms also turned out to be difficult, due to the custom code but also in many cases the original tenants of a web site were not available anymore to perform the code migration (e.g., project team has been dissolved after the project ended or they simply lack the resources). Furthermore, many of these web sites cannot be fully preserved by solely relying on client-side technologies like crawlers. In particular, their dynamic, reactive character will be lost by generating static snapshots (e.g. most of these sites were some kind of database front-end).

Emulation as a preservation tool [7] is capable of keeping software accessible, especially preserving the software's interactive character. We apply the concept of emulation to networked software, in this case web servers to provide long-term access to complex web sites. In most cases of institutional LAMP services, there was a uniform server back-end. The tenant's data is available as a file system snapshot. We have done a case study to investigate and formalize preservation for such complex web sites in order to implement seamless transitions from a production state to a preserved but (on-demand) accessible state. In this case study, we have preserved 234 web sites and during this process we have developed a workflow to automate this process.

We further have investigated and implemented methods for secure and functional long-term access to old, unmaintained, and, thus, insecure machines. We show different ways of end-user access, both using the user's contemporary browser as well as using an emulated historic browser, which can provide an enriched user experience by serving archived pages from public web archives for any potential external links in archived web platforms.

2 EMULATING NETWORKS

Until now, emulation has been mostly associated with instances of singular machines, e.g., running an old Linux server and providing interactive access to this machine. For accessing web sites however, interactive access to the server is not ideal. Instead, access through a web browser (emulated or contemporary) or similar dedicated networked clients is necessary.

In order to support networked scenarios, we need to broaden the scope of emulation by incorporating network components and network services to cover an era beginning at the 2000s when the Internet started to become the predominant publication medium. By emulating network components, we are able to decouple preserved networks both from the host system (for security reasons) and, conceptually, from the technical life-cycle.

As the base layer for emulated networks, we have identified Ethernet, which can serve as a universal common denominator for many different network types.

To implement an emulated network environment, we spawn a new virtual Ethernet network consisting of a central software-based Ethernet switch [2], which allows to connect a number of emulated machines. To exchange network traffic between emulated machines, these need to get "wired", i.e., connected to the

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

central Ethernet switch. The "cables" between software-based Ethernet devices, i.e., an Ethernet network adapter and switch, are implemented as HTTP-based WebSocket connections [3], optionally using TLS [6] for encryption. Using HTTP as transport layer not only allows for (secure) transport of encapsulated Ethernet frames over the public Internet but also ensures a strict separation of virtual network traffic from other network traffic on the host machine, i.e., the host machine does not need to implement custom firewall or network routing rules. To eliminate any danger from archived environments attacking the host system or using the host system's network resources to attack third-party entities on the Internet (or the host system's private network environment), the emulated machines run within a Linux container[4] without network capabilities[1]. The emulated guest's Ethernet frames are passed through a UNIX domain socket to the host system to be forwarded through a WebSocket connection. This setup also shields the archived (unmaintained and insecure) environments from random attacks from the public Internet as they are not visible in the public network.

Due to using HTTP-based connections between emulated machines and emulated network infrastructure, emulated machines can be deployed independently, e.g., on different host machines and can be (re-)connected to a network as needed. This infrastructure also allows to interconnect different, independent network instances, e.g., to orchestrate a larger network of emulated web servers. Lastly, this network setup allows (multiple) ad-hoc client connections, e.g., to start a dedicated emulated client computer to interact with servers within the network (see Section 2.2.3).

2.1 Network Services

A crucial building block for an emulated network are network services, services which provide infrastructure normally present in a network. The most important service is DNS name resolution paired with a DHCP IP-address management service, both crucial for automation and orchestration. Since the network is isolated from the live Internet or intranet, machines with any manually assigned IP addresses are supported, allowing unmodified machines to be added to a network while retaining their configured IP addresses. In practice and especially in multi-machine setups, machines typically use (fully qualified) domain names (FQDNs) to identify and address networked machines or services. For this, the internal DNS/DHCP service maintains IP address and FQDN mappings and manages their assignment on startup. Relations between machine instances and assigned domain names are maintained as network meta-data. Again, as the emulated network is isolated from the live Internet, machines can keep their original FQDN inside the emulated network, while it might already have been deleted or re-used on the live Internet. This is important as FQDNs can often be hard-coded in server-side web applications or even as (absolute) URLs in static web pages.

A further option is to connect the emulated network to the archived Internet, which is implemented through a pywb[1] service. This allows to serve archived web sites from external web archives, e.g., the Internet Archive – using a user-defined archival date – or to serve locally provided WARC files. If the service is activated within the network, it acts a transparent proxy routing all HTTP traffic through the pywb service, without having to reconfigure the individual archived machines.

Additionally, it is necessary to forward traffic from external sources (e.g., machines or services connected via the public Internet or intranet) to an emulated network. External machines which are not part of the emulated network are using the TCP/IP protocol suite directly to communicate with other machines or services. Hence, to support connections from the Internet to an machine or service in an emulated network, a gateway is necessary to translate the network traffic between the external network and the WebSocket-encapsulated emulated network. Depending on where and how the gateway is deployed, different access scenarios can be realized. The first option is to connect an application, e.g., the user's browser with a server inside the emulated network. For this, the user downloads a local instance of the gateway software (*eaas-proxy*, currently available for Windows, macOS, and GNU/Linux)[2] and initiates a session in the EaaS web interface via their web browser. A typical session will open a local (localhost) TCP server socket, connect via WebSocket (over HTTPS) to an emulated network and forward local traffic to a service inside the emulated network environment, e.g., TCP port 80 of a web server. All configuration parameters, e.g., (WebSocket) URL of the emulated network, target machine, destination port, are part of the initiation link passed by the EaaS web interface via the user's web browser. Alternatively to forwarding a single TCP port, eaas-proxy can be configured to act as a SOCKS5 proxy server, allowing local applications with SOCKS5 support to reach any host/port in the emulated network.

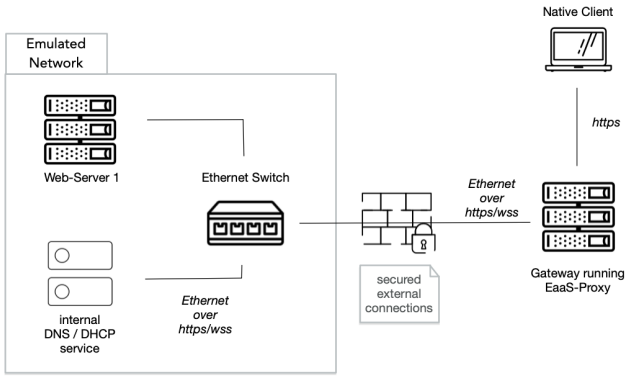


Figure 1: A simple emulated network consisting of a web server instance and an internal DNS/DHCP service, both connected to a central Ethernet switch. Additionally, an eaas-proxy is running as a gateway, securely connecting a native client application (e.g., a contemporary web browser) to the emulated network. The native client does not have to be changed.

Instead of installing the gateway on the user's machine, the gateway can be deployed as a public/shared network service, e.g., acting as a web server substitute accessible on the archived web server's original domain. The eaas-proxy gateway relays traffic

[1] https://github.com/webrecorder/pywb

[2] https://gitlab.com/emulation-as-a-service/eaas-proxy

from the public Internet into the emulated network. Figure 1 shows a setup allowing a contemporary browser to connect to an emulated web server via the eaas-proxy gateway.

Since eaas-proxy is implemented as a JavaScript/WebAssembly-based Node.js application, it could also be run directly in the user's web browser. Using W3C's Service Workers specification, any HTTP request of the browser to the archived web server could here be intercepted, serialized to Ethernet frames on the client, and sent via a WebSocket connection directly to the emulated network environment, eliminating any processing of the unencapsulated network traffic on the (EaaS) server and further increasing security.

2.2 User Access

When preserving web servers, offering simple, ideally seamless access to a wide range of users is the main goal. The main reason to utilize emulation is to preserve the server's interactive character and its full functionality. While the technical challenges of connecting to an emulated network have been discussed in Section 2.1, this section focuses on organizational and administrative questions of providing user access.

2.2.1 Security Considerations. As machines are archived in their original state, several questions regarding security occur. Operating systems and other software within the archived machines have known or yet unknown security issues. In a preservation scenario, updates are usually no longer available since the software reached its end of life. But most likely, there also is no intent to update archived machines in any case as each update is a instance-specific and labor-intensive manual process, which does not scale with a large number of archived machines. Additionally, upgrading software to a more recent, vendor-supported version might change the service's behavior. In many cases, an erroneous behavior might be obvious, e.g., a completely non-functional web site after a major PHP version update, but it could be more subtle, e.g., a broken search function on a sub-page. In any case, detecting changed behavior is very hard to accomplish in a generic way and would be in direct contrast to the stated objective of archiving the web server with its original behavior.

As a consequence, security issues in archived machines have to be accepted and have to be maintained when a machine is reactivated using emulation. For a general risk assessment, different layers have to be considered:

(1) *Isolation of the archived machine and protection of the hosting infrastructure* – In order to protect the hosting infrastructure, the machine needs to be isolated, such that a compromised machine cannot take over the hosting machine or is able to abuse or access any of its resources. Within EaaS, this is realized by running the machine inside an emulator or using hardware virtualization. Since not all emulators are actively audited for security issues, the emulator itself is executed inside a restricted Linux container, in particular without access to the host's resources like other running processes, file system, and network. Emulation and virtualization software but more importantly the host's Linux kernel and container runtime infrastructure are actively maintained software and should be up-to-date, such that the security risk for the host infrastructure running the archived, vulnerable machines is comparable to running any contemporary Internet facing software.
(2) *Malicious user attacks* – Any user accessing the archived web server might be able to exploit known issues of the site's back-end software. The most obvious attack is vandalism, i.e., changing contents of a web page or modifying the web server as a whole. This problem can be mitigated by running an emulated instance in a non-persistent mode by default. In EaaS, all data the emulated machine (and the emulator) is able to access, e.g., disk images, is served via HTTP GET requests and thus read-only by definition. All data written is temporary and destroyed when a session ends unless explicitly saved as a new disk image layer. If the session model (cf. Section 2.2.3) allows multiple users sharing a single session, the vandalism problem can only be mitigated, e.g., by resetting the session in regular intervals.
Furthermore, a malicious user could exploit an emulated machine to launch attacks against other users sharing the same session. If the server is accessed through an emulated *remote* browser, the main risk for an attacked user is a negative user experience, e.g., if the browser crashes, similarly to the vandalism problem. However, social engineering attacks could be possible, too, e.g., asking another user in a shared session to input personal data by making the emulated web server serve a manipulated web page. If the server is accessed through the user's browser, the risk is comparable to the general risk of browsing the Internet. The user should use an up-to-date browser and use common security practice when downloading content from untrusted sources or sharing sensitive (personal) data with the service.
(3) *Exposure of sensitive user data* – Another problem occurs if visitors, in presence of security bugs, can essentially access any part of the web server and web site, including administration interfaces and, in particular, content which was originally not meant to be public for privacy or security reasons, e.g., usernames, email addresses, or birthdays, but also TLS certificates and keys, password hashes, or SSH keys of users of the in-production service. Machines with sensitive content require a rigorous curatorial review and need to be redacted, if necessary, following common practices. Alternatively, access to these machines can be restricted to a dedicated and trusted user-group and through dedicated (emulated) remote clients, which do only have limited capabilities and (hacking-)tools available.

Even though emulated machines, e.g., web servers, are accessible, these are never exposed directly to the Internet and therefore have a reduced attack surface. Traffic can be proxied and, e.g., HTTPS connections are managed and terminated by the proxy, such that security problems in the TCP/IP and TLS stacks are mitigated (e.g., CVE-2014-0160 a.k.a. Heartbleed). Since the emulated machines do not appear in the public IPv4/6 address space but only in emulated network environments, they do not appear in automated scans of the public Internet.

All these measures help to significantly decrease risks of running machines with known security issues and simultaneously allow less restrictive (public) access models.

2.2.2 Maintaining built-in security measures. A different kind of security problem are security measures deployed inside an archived machine with the goal to protect its content or identity. For instance, an archived web server might only provide access to its content via HTTPS (HTTP over TLS). The archived TLS certificates might only be valid for the wrong domain name, already have expired, or lack a valid certificate chain to a root CA certificate trusted by today's systems. The archived server might also only support deprecated and outdated SSL/TLS versions which are no longer supported by today's systems.

Generally, solving the problem is possible by adding a (reverse) proxy in front of the emulated server, which simply accepts any certificate presented by the server. This does not compromise security because, as described in Section 2.1, access to the emulated server running in an emulated network environment is already secured on a lower layer. Accessing a TLS server with an outdated TLS version is more difficult, however, as contemporary software libraries (e.g., OpenSSL) might have dropped support for these versions. Older library versions still supporting the outdated TLS versions cannot simply continue to be used as the reverse proxy is running on the host system and thus has to be up-to-date to not compromise the host system.

2.2.3 Session Setup & Management. The most important consideration for user access is the session setup. Based on the aforementioned technical infrastructure and security considerations, different setups are possible:

(1) *Contemporary browser access* – In this case, the user accesses the web site with their contemporary browser. This is usually the most convenient option since there is typically no notable difference to browsing any other web site. In addition, all URLs (at any depth) can be shared or bookmarked. This approach, however, is usually only available for a (short) *transition period*, as long as there are no significant security concerns and as long as the contemporary browser renders the pages (e.g., there is no need for deprecated plugins like Java, Adobe Flash, etc.).
(2) *Remote browser* – In order to render old web pages with browsers of their time, a lightweight *remote browser*, originally developed and made available by the Old Web Today project [3], is a good compromise. Remote browsers are usually older versions of popular browsers – and due to known security issues not recommended to be used on a desktop computer – running in a concealed and secure environment and offer features a contemporary browser does not support anymore (e.g., Flash support). The variety of the lightweight remote browsers, however, is currently limited to browsers and plugins that have been released for Linux-based operating systems. For native Microsoft Windows browsers (predominately Internet Explorer 6-8) and especially Windows-specific plugins like ActiveX, a fully installed Windows client machine is necessary (see next option).
(3) *Emulated machine client* – While option 1. and 2. are lightweight (with regards to resource consumption) and offer a good user experience, these options are limited, both regarding their availability and coverage of potential use-cases. A long-term available and fully back-end compatible option is a full client installation as shown in Figure 2.

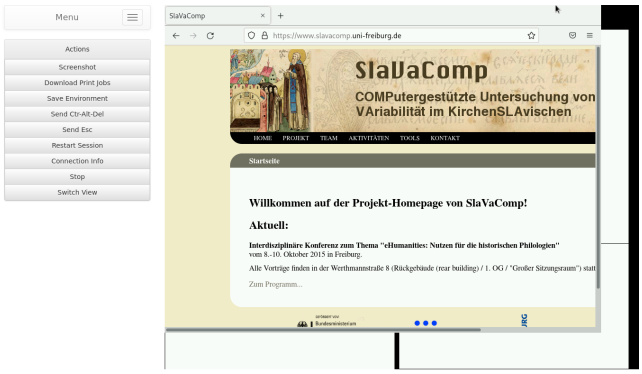


Figure 2: The Firefox browser running in a emulated machine inside the emulated network, accessing the emulated web server using its original domain name.

With remote browser access or emulated machine clients, further options become possible, e.g., integrating web archives (using pywb) and thus creating the illusion of consistently browsing the web of that time. External links from preserved web servers will be relayed either to a pre-configured web-archive (e.g., Internet Archive) with a pre-configured date or, alternatively, WARCs preserved together with the server artifact.

Regardless of which option is chosen, the emulated web sites are accessible through or as a (public) web page, potentially available under the service's old URL or domain.

The second consideration for user access concerns the session management. Currently, three options are possible:

(1) *Static setup* – With this session setup, a network is setup to run permanently. It also offers the best user experience as the site is immediately available and interaction between users is possible. Ideally, this setup is chosen for an initial sunset phase of a web site when there is still significant traffic and the general security risks are considered as low (cf. malicious user attacks and sensitive data exposure).
(2) *On-demand setup with shared sessions* – For web sites with low demand, on-demand access is a cost efficient and more secure option. If a user tries to load the page and no active network for the requested instance is running, the proxy shows the user a waiting message and simultaneously initiates a new emulated network session. The startup time may vary, usually 1-2 minutes, in some cases it can take up to a few minutes. Once a session is running, subsequent users do not experience a waiting time. If there is no active user within a configured grace period, the network is shut down.
(3) *On-demand setup with individual sessions* – For security-critical instances or instances with a restricted user group, for every user a new (private) network instance is created and kept alive as long as it is in use.

[3] https://oldweb.today

3 CASE STUDY PRESERVING WEB SITES

For this use-case, we chose the university's computing centers production web server setup. A total of 10 web servers are in use, currently hosting 297 web sites. The servers run an up-to-date openSUSE system, the Apache HTTP Server is used as web server and is configured so that for each user account an individual configuration is used. One of the servers is still running an older version of openSUSE to remain compatible with PHP 5 for older, unmaintained web sites. The user's configuration and web site data are mounted via NFS from a storage server containing the configuration and content data. Most of the web sites are written in PHP.

For this case study, we have rebuilt the base system as a Docker image by manually installing the relevant Apache packages and dependencies. In order to support both PHP 5 and PHP 7, two independent images have been created. Alternatively, the server's original file system could have been extracted or imaged but this would have resulted in a much larger disk image. The resulting Docker image of the PHP 5 version has a size of 188 MB as gzipped and 437 MB as uncompressed tar archive. The PHP 7 version has a size of 317 MB compressed and 830 MB uncompressed. By using the Docker build, the creation of the base file system is now documented through a Dockerfile, can be rebuilt in a reproducible way (with a preserved version of the openSUSE package repository) and can be further adapted if needed.

For automated extraction and packaging of web site data as well as quality assurance, we have developed a Python application that provides a REST API, such that the preservation task can be operated from a web front-end, e.g., by the respective owners of a web site. The archiving application operates on a list of all configured web hosts, which contains the FQDN of each web host, the tenant's username as well as the local user ID (UID), necessary to recreate the file system of the web host. It produces a tar archive containing the web host's data including the Apache configuration, content web, and the owner's home directories. In a final step, the base file system containing the web server and the individual user data are brought together. This is currently implemented by using Docker (container) techniques due to the effective handling of containers for long-term preservation using emulation [5].

3.1 Results & Future Work

In order to analyze the result, in particular to verify that the preserved version is complete and functional, we have created a sitemap of the live version using a web crawler. We can then verify that each page (i.e., URL from the respective host – external URLs are ignored) available within the live site is also reachable within the archived version. Since the preserved version is deployed in an (emulated) network with its own DNS, the web host is using its original FQDN. The URLs from the previously generated sitemap are then retrieved within the redeployed instance and the HTTP status codes are compared for each entry.

Out of the 297 listed web hosts, 234 web hosts could be extracted, preserved and further evaluated. Some web sites had no content, i.e., their data directory was empty, some web sites only provided a redirect page to external resources. A further problem were web sites with a mandatory user-login, e.g., HTTP Basic Authentication or similar. Although we could successfully archive these web sites, a verification was not possible. We have excluded these sites from our results.

Of the evaluated sites, only ten sites behaved differently within the emulated network. For two of these sites we could verify that the web site's content was changed on the live system during the scans. One site produced four HTTP 500 errors in the archived version. While the live system returned an HTTP 200 status code, the archived system returned the same page with identical content but an HTTP 500 error code. This behavior could indicate missing plugins in the archiving container image. Four more web sites returned an HTTP 403 error within the emulated network because some pages were only accessible from certain IP ranges. One web host returned an HTTP 200 code for only 3 out of 462 pages, the rest were returned with an HTTP 500 error, which indicated missing PHP 7 packages or other software. The last two web sites showed significant differences in the HTTP 200 and HTTP 400 status codes but generated no HTTP 500 error codes. The root cause for this mismatch could not be found yet and requires further investigation.

The technology and the case study presented in this article show that preservation of complex web sites (e.g., based on a typical LAMP stack) can be achieved for long-term access through emulation in a highly automated way. We also laid out a concept for an economical preservation model, i.e., starting the server components only on-demand, as well as with manageable security risks. While the presented results are encouraging, more work needs to be done. Our current verification method focused on general availability of pages for now. For further analysis, web site's content could be verified as well as an in-depth functionality check could be conducted, e.g., by using a screen scrapping tool like Beatiful Soup. Furthermore, the preserved version of a site is using the same database back-end as the original site. In order to obtain a complete archive, the database has to be exported and redeployed in the emulated network such that the archived web site is able to access the archived database. While user credentials and database configuration can be found in the user's data, there are still challenges, both technical and legal, to use this information to archive the database content in a fully automated way. Finally, for a production service offering preserved web sites, the landing page infrastructure needs to be built. A generic service web page for each archived web host (possibly on its original FQDN) needs to be setup. Depending on the chosen service model, it will initiate the start of an emulated network or connect to a running instance. This is currently ongoing work, especially, setting up this infrastructure in an automated way.

REFERENCES

[1] Eric W Biederman and Linux Networx. 2006. Multiple instances of the global Linux namespaces. In *Proceedings of the Linux Symposium*, Vol. 1. Citeseer, 101–112.
[2] Renzo Davoli. 2005. Vde: Virtual distributed ethernet. In *First International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities*. IEEE, 213–220.
[3] Alexey Melnikov and Ian Fette. 2011. The WebSocket Protocol. RFC 6455. https://doi.org/10.17487/RFC6455
[4] Paul B Menage. 2007. Adding generic process containers to the Linux kernel. In *Proceedings of the Linux symposium*, Vol. 2. Citeseer, 45–57.
[5] Klaus Rechert, Oleg Stobbe, Oleg Zharkow, Rafael Gieschke, and Dennis Wehrle. 2020. CiTAR - Preserving Software-based Research. *Int. J. Digit. Curation* 15, 1 (2020), 1–13.
[6] Eric Rescorla. 2000. HTTP Over TLS. RFC 2818. https://doi.org/10.17487/RFC2818
[7] David SH Rosenthal. 2015. Emulation & virtualization as preservation strategies. *Andrew W. Mellon Foundation* (2015).

Web Archiving as Entertainment

Travis Reid
treid003@odu.edu
Old Dominion University
Norfolk, Virginia, USA

Michael L. Nelson
mln@cs.odu.edu
Old Dominion University
Norfolk, Virginia, USA

Michele C. Weigle
mweigle@cs.odu.edu
Old Dominion University
Norfolk, Virginia, USA

ABSTRACT

We want to make web archiving entertaining so that it can be enjoyed like a spectator sport. Currently we are working on applying gaming concepts to the web archiving process and on integrating video games with web archiving. We are creating web archiving live streams and gaming focused live streams that can be uploaded to video game live streaming platforms like Twitch, Facebook Gaming, and YouTube. Livestreaming the crawling and replay of web archives removes some of the mystery and makes it transparent to third parties. The gaming focused live streams will have gameplay that is influenced by the web archiving and replay performance from the web archiving live stream. So far, we applied the gaming concept of speed runs to web archiving and integrated a few video games with an automated web archiving live stream. We recorded a demo that starts with a web archiving speedrun where we gave a set of seed URIs to Brozzler and Browswertrix Crawler to see which crawler would finish archiving the set first. Then we used Selenium to apply the crawler performance results (speed) to character traits in the Gun Mayhem 2 More Mayhem video game. A viewer could then watch the in-game characters battle for top crawler.

KEYWORDS

web archiving, gaming, live streaming

Figure 1: The current process for running our web archiving live stream and gaming live stream

1 INTRODUCTION

We have been working on a proof of concept that involves the integration of gaming concepts, game platforms, live stream platforms, and web archiving. We have created automated web archiving live streams where gaming concepts were applied to the web archiving process to make the live stream more entertaining. We have also created gaming focused live streams where the selections made in the game were influenced by the web archiving performance from the web archiving live stream.

2 INTEGRATING WEB ARCHIVING, GAMING, AND LIVE STREAMING

Figure 1 shows the current process that we are using for the web archiving live streams and the gaming live streams. Our web archiving live stream is a competition between crawlers to see which crawler performs the best when archiving a set of seed URIs. The first step for the web archiving live stream is to use Selenium to setup the browsers that will be used during the live stream. The automated browsers are used to provide context information about each crawler (name) and the current progress for each crawler (current URL being archived and the number of webpages archived so far). The next step is to get a set of seed URIs that will be used for the competition and then let each crawler start archiving the URIs. After the web archiving competition is finished, a game configuration file will be created based on how well the crawlers performed during the web archiving live stream. If a crawler had good performance during the web archiving live stream, then the in-game character associated with the crawler will have better items, perks, and other traits. If a crawler performs poorly, then their in-game character will have the worst character traits.

Our gaming live stream (Figure 1) shows an automated gaming session that is influenced by the results of a web archiving live stream. The first step for the gaming live stream is to use an app automation tool like Selenium (for browser games) or Appium (for locally installed PC games) to select the settings for the in-game characters. After the settings are selected by the app automation tool, the match is started and the viewers of the live stream can watch the match between the crawlers' in-game characters.

3 CONCLUSION

The proof of concept that we have been working on is a new way to integrate gaming, web archiving, and live streaming. We have created automated live streams where the performance of a web crawler was used to influence the selections that were made inside of a video game.

ACKNOWLEDGMENTS

The Game Walkthroughs and Web Archiving project has received a seed grant from the International Internet Preservation Consortium (IIPC). We thank the IIPC for supporting this work.

First steps in Identifying Academic Migration using Memento and Quasi-Canonicalization

Mat Kelly
Drexel University
Philadelphia, PA, USA
mkelly@drexel.edu

Deanna Zarrillo
Drexel University
Philadelphia, PA, USA
dz364@drexel.edu

Christopher Jackson
University of Tennessee
Knoxville, TN, USA
cjacks75@vols.utk.edu

Erjia Yan
Drexel University
Philadelphia, PA, USA
ey86@drexel.edu

ABSTRACT

Academic research faculty change their institutional association over their careers. This association may be documented on the web through the web sites of the faculty member's institutional department. The effects of disassociation are detrimental to historically black colleges and universities (HBCUs), but the degree of "brain drain" is difficult to evaluate without first obtaining evidence of these faculty member's movement over time. This work describes a preliminary effort to utilize web archives to identify faculty that resided at HBCUs in the past in an attempt to further examine the effects of brain drain. We utilize a Memento aggregator along with a manual data selection procedure to first identify the target URI-Ms. We then perform a recursive procedure to supplement TimeMaps with a more comprehensive picture of HBCU department web pages over time. This association and quasi-canonicalization procedure is a first step in identifying the effects of brain draft by utilizing the past Web.

KEYWORDS

web archives, memento, hbcu, canonicalization

1 MOTIVATION

College and universities have historically had a web presence. Their sites are typically structured by college, department, or other sub-units to ensure relevant information about the respective unit is effectively represented. The sub-units' sites often list those employed by the unit, for example, faculty, staff, and research assistants. The faculty employed by the unit changes in time. Thus, identifying how faculty have transitioned between sub-units or among higher education units is possible by consulting historical representations of the sub-units' sites.

The degree to which up-to-date and representative information is present on academic web sites may be relative to a variety of factors including the units' size, staffing, resources, etc. This, too, has changed in time as colleges and universities have recognized the importance of having an online presence.

Our focus is on faculty representation online at historically black colleges and universities (HBCUs). We leveraged the units' live web presence as a basis to identify the degree to which the units' online presence have been preserved. We anticipate being able to supplement the captures (e.g., URI-Ms for Howard University's Pharmacology Department[1]) with previously associated yet currently disassociated archival identifiers (e.g., URI-Ms for the department's URI-R in the past[2]). Our ultimate goal is to more thoroughly identify faculty at these HBCUs and how they have moved between the aforementioned units and sub-units. This work serves as the initial data collection phase toward this goal of identifying "brain drain" from HBCUs.

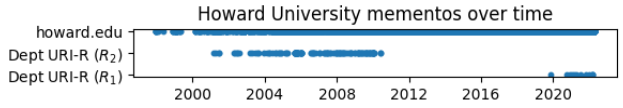


Figure 1: While an HBCU's homepage spans a time range, a department at the HBCU that exists on the live web (R_1) might have existed elsewhere in the past (R_2).

2 METHODOLOGY

Our initial objective was to build a web archive collection for analysis. Our base data set consisted of 35 URI-Rs identifying the homepages of the HBCUs. For archival queries, we utilized a local instance of the open source MemGator Memento aggregator[3] software with the default archival configuration.

We initially requested 35 TimeMaps for these URI-Rs (H) from the aggregator. Next, we manually collected URI-Rs for each sub-unit at each HBCU based on the live web ($\{D_f\}$). We again queried the aggregator for TimeMaps for each $D_i \in D_f$, which varied in quantity among the HBCUs. For example, Howard listed 51 sub-units on the live Web with varying scopes (e.g., colleges, departments, majors). Using the 35 TimeMaps in H, we downloaded the resource response of the oldest memento (M_0). We manually identified the historical departmental URI-Rs in each oldest memento for each $H_h M_0$ where $H_h \in H$. Following this, we identified the differences between the set of departments per each memento ($D_0 \in H_h M_0$) and the list of departments represented on the live web, where D_f need not be equivalent in magnitude to $H_h M_f$ for TimeMaps like R_1.

In upcoming work, we will programmatically map historical department URI-Rs to contemporary URI-Rs, noting patterns of deviation. Those that have concrete associations can have their URI-Ms for each URI-R variation in the same TimeMap (Figure 1). This association of a department over time is akin to canonicalization and will surface additional nuances in the relative structures encompassed within a TimeMap. We will then execute a recursive procedure to acquire the list of faculty at each department over time using approaches that leverage named entity recognition.

3 ACKNOWLEDGEMENTS

This work is supported by NSF SoS:DCI grant #2122525.

[1]Currently at R_1: https://medicine.howard.edu/graduate-programs/pharmacology
[2]Previously residing at R_2: http://med.howard.edu/pharmacology

[3]https://github.com/oduwsdl/memgator

CDX Summary for Web Archival Collection Insights

Sawood Alam
sawood@archive.org
Web and Data Scientist
Wayback Machine
Internet Archive
San Francisco, California, USA

Mark Graham
mark@archive.org
Director
Wayback Machine
Internet Archive
San Francisco, California, USA

ABSTRACT

Large web archival collections are often opaque about their holdings. We created an open-source tool called, CDX Summary, to generate statistical reports based on URIs, hosts, TLDs, paths, query parameters, status codes, media types, date and time, etc. present in the CDX index of a collection of WARC files. Our tool also surfaces a configurable number of potentially good random memento samples from the collection for visual inspection, quality assurance, representative thumbnails generation, etc. The tool generates both human and machine readable reports with varying levels of details for different use cases. Furthermore, we implemented a Web Component that can render generated JSON summaries in HTML documents. Early exploration of insights on Wayback Machine collections uncovered numerous issues in some of our crawl operations that we improved as a result.

KEYWORDS

Web Archiving, Summarization, Collection Summary, CDX Index, CLI, Internet Archive, Petabox

1 INTRODUCTION

Items[1] in the Internet Archive's Petabox[2] collections of various media types like image, video, audio, book, etc. often have rich metadata, representative thumbnails, meaningful titles, and interactive hero elements. However, web collections, primarily containing an arbitrary number of WARC files [3, 19] and their corresponding CDX files [18], often look opaque with tombstone thumbnails for items, template-based similar looking titles, and limited metadata as shown in Figures 1(a) and 1(b).

Opacity of web archival collections is not an Internet Archive-specific problem [20]. Since the emergence of memento protocol [23, 26, 27], there have been numerous efforts to explore web archival holdings, both for efficient memento routing [1, 4, 9–16, 21, 25] by memento aggregators [2, 8] as well as for collection understanding and scholarly access [17, 24].

To address this issue we created an open-source Command Line Interface (CLI) tool[3] called *CDX Summary* to process sorted CDX files and generate statistical reports [5]. These summary reports give insights on various dimensions of CDX records/captures, such as, total number of mementos, number of unique original resources, distribution of various media types and their HTTP status codes, path and query segment counts, temporal spread, and capture frequencies of top TLDs, hosts, and URIs. We also implemented a uniform sampling algorithm to select a given number of random memento URIs (i.e., URI-Ms) with *200 OK* HTML responses that can be utilized for quality assurance purposes or as a representative sample for the collection of WARC files. Our tool can generate both comprehensive and brief reports in JSON format as well as human readable textual representation. We ran our tool on a selected set of public web collections in Petabox, stored resulting JSON files in their corresponding collections, and made them accessible publicly (with the hope that they might be useful for researchers). Furthermore, we implemented a custom Web Component [7] that can load CDX Summary report JSON files and render them in interactive HTML representations. Finally, we integrated this Web Component into the collection/item views of the main site of the Internet Archive, so that patrons can access rich and interactive information when they visit a web collection/item in Petabox. We also found our tool useful for crawl operators as it helped us identify numerous issues in some of our crawls that would have otherwise gone unnoticed.

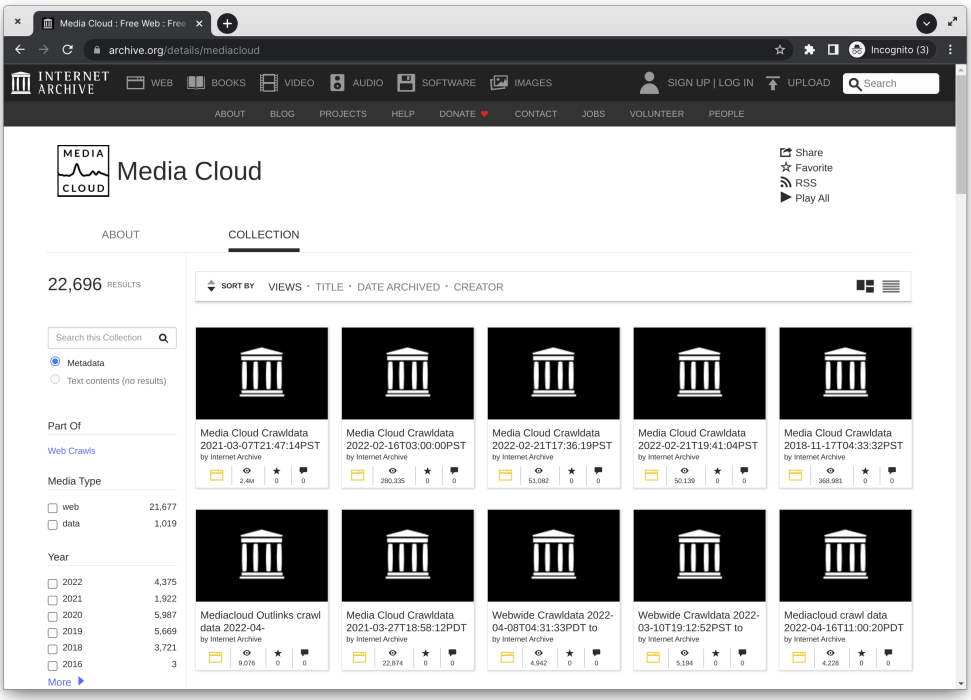
2 IMPLEMENTATION

CDX Summary is a Python package [6] that ships with a CLI tool to process sorted stream of CDX files via *STDIN* or as a file/URI argument and generate a compact or detailed report in human or machine readable formats. It has some Internet Archive-specific features, but can be used against any local or remote CDX files or CDX APIs. Figure 2 shows installation instructions and available CLI options. Below are some feature highlights of the tool:

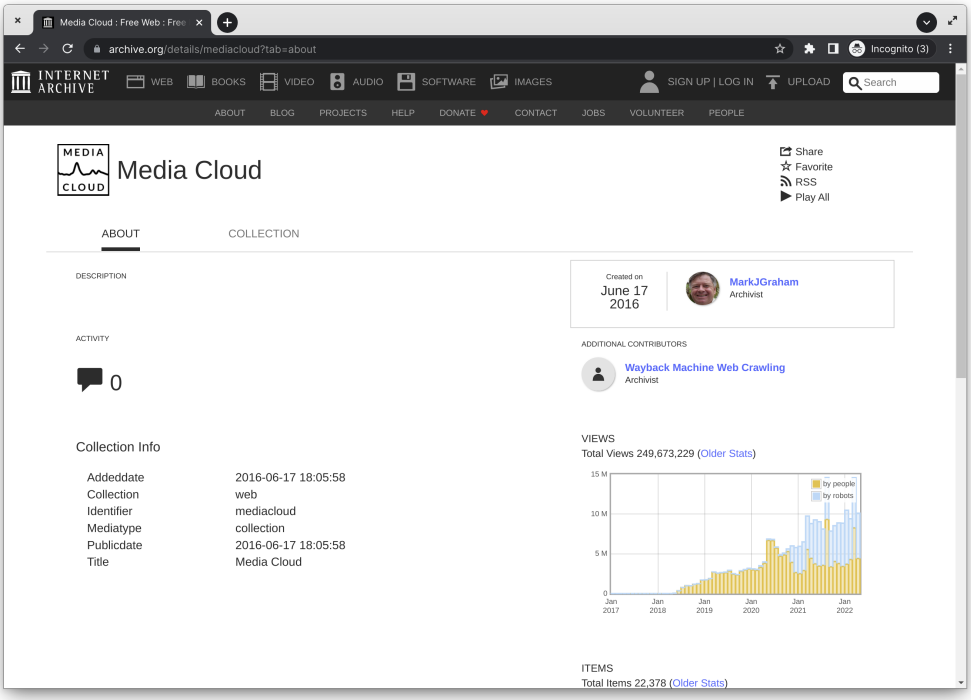
- Summarizes local CDX files or remote ones over HTTP
- Handles `.gz` and `.bz2` compression seamlessly
- Handles CDX data input to *STDIN* from pipe, allowing any filtering or slicing of input
- Supports Internet Archive Petabox web item summarization using item identifier or URI
- Supports Wayback Machine CDX Server API summarization
- Seamless authorization to Internet Archive via the `ia` CLI tool
- Human-friendly summary by default, but supports summarized or detailed JSON reports
- Self-aware, as the input can be a previously generated JSON report in place of CDX data

From Section 2.1 to Section 2.6 we describe various reports generated by the tool. Numbers reported in Table 1 through Table 5 are not evaluations, but illustrations of the CLI output based on the collection-level CDX file generated on February 2, 2022, for the *mediacloud* collection[4].

[1] https://archive.org/services/docs/api/items.html
[2] https://archive.org/web/petabox.php
[3] https://github.com/internetarchive/cdx-summary/
[4] https://archive.org/download/mediacloud



(a) WARC Items Listing (https://archive.org/details/mediacloud)



(b) Collection Metadata (https://archive.org/details/mediacloud?tab=about)

Figure 1: A Petabox Web Collection of Internet Archive

```
$ pip install cdxsummary

$ cdxsummary --help
usage: cdxsummary [-h] [-a [QUERY]] [-i] [-j] [-l] [-o [FILE]] [-r] [-s [N]] [-t [N]] [-v] [input]

Summarize web archive capture index (CDX) files.

positional arguments:
  input                 CDX file path/URL (plain/gz/bz2) or an IA item ID to process (reads from the STDIN, if empty or '-')

optional arguments:
  -h, --help            show this help message and exit
  -a [QUERY], --api [QUERY]
                        CDX API query parameters (default: 'matchType=exact'), treats the last argument as the lookup URL
  -i, --item            Treat the input argument as a Petabox item identifier instead of a file path
  -j, --json            Generate summary in JSON format
  -l, --load            Load JSON report instead of CDX
  -o [FILE], --out [FILE]
                        Write output to the given file (default: STDOUT)
  -r, --report          Generate non-summarized JSON report
  -s [N], --samples [N]
                        Number of sample memento URLs in summary (default: 10)
  -t [N], --tophosts [N]
                        Number of hosts with maximum captures in summary (default: 10)
  -v, --version         Show version number
```

Figure 2: CDX Summary CLI Installation and Help

Table 1: Collection Overview

Total Captures in CDX	3,221,758,919
Consecutive Unique URLs	2,381,208,479
Consecutive Unique Hosts	5,366,007
Total WARC Records Size	172.0 TB
First Memento Date	Jun 09, 2016
Last Memento Date	Feb 02, 2022

2.1 Collection Overview

Table 1 shows a high-level overview of the collection CDX input. It includes reports like the number of mementos/captures, number of unique original URIs and hosts, accumulated WARC record sizes, and the dates of first and last mementos.

2.2 Media Types and Status Codes

Table 2 shows statistics for HTTP status code groups of captures of various media types for the collection. The *Revisit* records do not represents an independent media type, instead, they reflect an unchanged state of representations of resources from some of their prior observations (i.e., the same content digest for the same URI). The *TOTAL* column shows combined counts for each media type irrespective of their HTTP status code and the *TOTAL* row (displayed only if there are more than one media types listed) shows the combined counts of each HTTP status code group irrespective of their media types. When generating the detailed JSON report (i.e., using `-report` CLI flag), original media types and status codes are preserved instead of being grouped.

2.3 Path and Query Segments

Table 3 shows statistics for the number of path segments and the number of query parameters of various URIs in the collection. For example, the cell *P0* and *Q0* shows the number of captures of homepages of various hosts with zero path segments and zero query parameters. The *TOTAL* column shows combined counts for URIs with a specific number of path segments irrespective of their number of query parameters and the *TOTAL* row (displayed only if there are URIs with a varying number of path segments) shows the combined counts for URIs with a specific number of query parameters irrespective of their number of path segments. When generating the detailed JSON report (i.e., using `-report` CLI flag), original path segments and query parameter counts are preserved instead of being rolled up under *Others*.

2.4 Temporal Distribution

Table 4 shows the number of captures of the collection observed in different calendar years and months. The *TOTAL* column shows combined counts for corresponding years and the *TOTAL* row (displayed only if the captures were observed across multiple calendar years) shows the combined number of captures observed in the corresponding calendar months irrespective of their years.

2.5 Top Hosts

Table 5 shows configurable top-k hosts of the collection based on the number of captures of URIs from each host. The *OTHERS* row, if present, is the sum of the longtail of hosts. When generating the detailed JSON report (i.e., using `-report` CLI flag), counts for all the hosts are included instead of only top-k.

2.6 Random Memento Samples

Table 6 shows a list of configurable N random sample of captured URIs linked to their corresponding Wayback Machine playback URIs (this is configurable) from the collection. The sample is chosen only from mementos that were observed with the `text/html` media type and `200 OK` HTTP status code. Any unexpected URIs in the list (e.g., with a `.png`/`.jpg`/`.pdf` file extension) are likely a result of the Soft-404 issue [22] from the origin server. These random samples with uniform distribution across the stream are dynamically selected by a single-pass algorithm as illustrated in Figure 3.

Table 2: MIME Type and Status Code Distribution

MIME	2XX	3XX	4XX	5XX	Other	TOTAL
HTML	965,411,367	276,575,748	125,142,922	10,774,370	254,055	1,378,158,462
Image	595,129,981	1,848,252	1,176,862	110,163	6	598,265,264
CSS	34,916,375	7,815	35,193	658	0	34,960,041
JavaScript	54,437,714	14,160	139,220	1,478	3	54,592,575
JSON	72,073,766	378,508	6,435,865	182,669	0	79,070,808
XML	123,611,319	474,681	8,382,386	265,499	0	132,733,885
Text	9,717,016	4,989,191	5,825,931	89,757	7	20,621,902
PDF	907,503	1,743	297	12	0	909,555
Font	1,348,879	208	5,827	21	0	1,354,935
Audio	1,655,064	7,847	4,701	9	0	1,667,621
Video	2,081,368	14,928	20,936	5	0	2,117,237
Revisit	0	0	0	0	790,098,238	790,098,238
Other	34,327,496	87,943,703	4,302,731	629,212	5,254	127,208,396
TOTAL	1,895,617,848	372,256,784	151,472,871	12,053,853	790,357,563	3,221,758,919

Table 3: Path and Query Segments

Path	Q0	Q1	Q2	Q3	Q4	Other	TOTAL
P0	10,137,110	75,096,506	5,222,175	1,422,227	736,628	2,345,905	94,960,551
P1	322,542,556	43,841,147	27,014,507	29,103,774	7,100,402	41,134,623	470,737,009
P2	370,206,924	55,247,724	28,783,611	31,029,789	9,766,427	26,319,078	521,353,553
P3	367,658,614	61,591,011	17,508,660	15,919,708	5,806,691	13,254,728	481,739,412
P4	310,543,626	140,279,203	80,114,245	7,705,491	2,250,744	6,257,513	547,150,822
Other	881,020,001	157,916,352	26,846,263	18,521,368	6,848,796	14,664,792	1,105,817,572
TOTAL	2,262,108,831	533,971,943	185,489,461	103,702,357	32,509,688	103,976,639	3,221,758,919

Table 4: Year and Month Distribution

Year	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	TOTAL
2016	0	0	0	0	0	710,753	0	0	0	0	0	0	710,753
2018	0	0	0	0	18,409,894	20,138,119	63,997,290	82,864,370	101,464,239	52,569,004	170,693,250	112,690,671	622,826,837
2019	19,598,526	162,402,131	65,102,242	88,210,019	10,937,609	79,065,196	83,674,051	33,172,213	78,516,813	87,399,023	73,646,742	84,030,203	865,754,768
2020	80,033,841	76,467,578	84,829,936	176,568,383	195,626,716	227,839,303	55,837,610	48,706,314	44,075,057	43,451,655	62,242,880	58,438,537	1,154,117,810
2021	56,130,245	38,882,332	55,665,115	45,999,374	45,209,593	43,396,066	46,897,574	45,611,165	20,527,175	45,615,260	42,424,841	50,262,197	536,620,937
2022	40,104,802	1,623,012	0	0	0	0	0	0	0	0	0	0	41,727,814
TOTAL	195,867,414	279,375,053	205,597,293	310,777,776	270,183,812	371,149,437	250,406,525	210,354,062	244,583,284	229,034,942	349,007,713	305,421,608	3,221,758,919

Table 5: Top 10 Out of 5,366,007 Hosts

Host	Captures
wp.me	11,457,705
upload.wikimedia.org	10,667,921
designtaxi.com	10,639,524
indiewire.com	10,341,593
public-api.wordpress.com	9,176,593
youtube.com	7,301,589
facebook.com	6,414,589
elfagr.com	6,122,678
secure.gravatar.com	6,095,165
googletagmanager.com	5,627,624
OTHERS (5,365,997 Hosts)	3,137,913,938

```
SIZE = <desired_sample_size>
sample = List[SIZE]
processed = 0

def toss(item):
    quotient, remainder = divmod(processed, SIZE)
    if random() < 1 / (quotient + 1):
        sample[remainder] = item
    processed++

def main():
    for item in stream:
        if is_valid_candidate(item):
            toss(item)
    return sample
```

Figure 3: Uniform Random Stream Sampler Algorithm

2.7 CDX Summary Web Component

At the Internet Archive we are adopting Web Components[5] for various UI elements. To incorporate collection and item level summaries in the web UI of Petabox for web items we have created and open-sourced the CDX Summary Web Component [7]. To render an interactive HTML element from a CDX Summary JSON file or a Petabox item/collection use the Custom HTML Element as illustrated in Figure 4.

[5]https://www.webcomponents.org/

3 CONCLUSIONS AND FUTURE WORK

In this work we implemented a generic CDX Summarization CLI tool to summarize any web archival collection or CDX API. We

Table 6: Random Sample of 100 OK HTML Mementos

https://web.archive.org/web/20200429135903/https://mcuoneclipse.com/2017/07/09/karwendel/
https://web.archive.org/web/20200529190909/http://www.viralnewslatest.com/2020/01/
https://web.archive.org/web/20210609022051/https://www.towleroad.com/2020/09/canadas-drag-race-queen/
https://web.archive.org/web/20200312232938/https://www.albawabhnews.com/service/nc.aspx?id=3934368
https://web.archive.org/web/20210705095335/https://collingswood.umcommunities.org/events/
https://web.archive.org/web/20220104124729/https://marvamedia.wufoo.com/embed/z9zjza1mpvn62/
https://web.archive.org/web/20211116112557/https://www.urbanet.info/streets-cape-town/
https://web.archive.org/web/20200607204252/https://www.diariopuntual.com/node/40007
https://web.archive.org/web/20200616034412/https://okwave.jp/amp/qa/q388029.html
https://web.archive.org/web/20220103002118/https://www.diarioahora.pe/tag/cc-vulnerables/

```
<script src="https://unpkg.com/@internetarchive/cdxsummary"></script>

<cdx-summary src="CDX_SUMMARY_JSON_URL"></cdx-summary>
<!-- OR -->
<cdx-summary item="PETABOX_ITEM_OR_COLLECTION_ID"></cdx-summary>
```

Figure 4: CDX Summary Web Component

added some Internet Archive-specific features for seamless integration with Petabox. The tool generates both human and machine friendly reports with varying levels of details. For easier HTML rendering of generated summaries we implemented a Web Component and made both the CLI and the Web Component open-source. We implemented a single-pass uniform random stream sample algorithm to efficiently sample mementos for testing, quality assurance, or collection representation.

In the future, we would like to report various statistics based on unique URIs, not just the number of mementos. We would also like to identify and highlight some takeaway points based on heuristics (such as reporting just a few URIs being archived too many times or unusual HTML to page requisites ratio).

ACKNOWLEDGMENTS

We thank various Internet Archive staff members for their help. Brewster Kahle for testing the CLI, Kenji Nagahashi for feedback on the web UI text, Brenton Cheng for main site integration, Jason Buckner for Web Component help, and Jim Shelton for UX feedback.

REFERENCES

[1] Sawood Alam. 2014. Archive Profiler: Scripts to Generate Profiles of Various Web Archives. https://github.com/oduwsdl/archive_profiler.
[2] Sawood Alam. 2015. MemGator: A Memento Aggregator CLI and Server in Go. https://github.com/oduwsdl/MemGator.
[3] Sawood Alam. 2018. Web ARChive (WARC) File Format. https://www.slideshare.net/ibnesayeed/web-archive-warc-file-format.
[4] Sawood Alam. 2019. MementoMap: A Tool to Summarize Web Archive Holdings. https://github.com/oduwsdl/MementoMap.
[5] Sawood Alam. 2021. CDX Summary. https://github.com/internetarchive/cdx-summary.
[6] Sawood Alam. 2022. CDX Summary in PyPI. https://pypi.org/project/cdxsummary/.
[7] Sawood Alam. 2022. CDX Summary Web Component. https://www.npmjs.com/package/@internetarchive/cdxsummary.
[8] Sawood Alam and Michael L. Nelson. 2016. MemGator - A Portable Concurrent Memento Aggregator: Cross-Platform CLI and Server Binaries in Go. In *Proceedings of the 16th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '16)*. 243–244. https://doi.org/10.1145/2910896.2925452
[9] Sawood Alam, Michael L. Nelson, Herbert Van de Sompel, Lyudmila L. Balakireva, Harihar Shankar, and David S. H. Rosenthal. 2015. Web Archive Profiling Through CDX Summarization. In *Proceedings of the 19th International Conference on Theory and Practice of Digital Libraries (TPDL '15)*. 3–14. https://doi.org/10.1007/978-3-319-24592-8_1
[10] Sawood Alam, Michael L. Nelson, Herbert Van de Sompel, Lyudmila L. Balakireva, Harihar Shankar, and David S. H. Rosenthal. 2016. Web Archive Profiling Through CDX Summarization. *International Journal on Digital Libraries* 17, 3 (2016), 223–238. https://doi.org/10.1007/s00799-016-0184-4
[11] Sawood Alam, Michael L. Nelson, Herbert Van de Sompel, and David S. H. Rosenthal. 2016. Web Archive Profiling Through Fulltext Search. In *Proceedings of the 20th International Conference on Theory and Practice of Digital Libraries (TPDL '16)*. 121–132. https://doi.org/10.1007/978-3-319-43997-6_10
[12] Sawood Alam, Michele C. Weigle, and Michael L. Nelson. 2021. Profiling Web Archival Voids for Memento Routing. In *Proceedings of the 21st ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '21)*. 150–159. https://doi.org/10.1109/JCDL52503.2021.00027
[13] Sawood Alam, Michele C. Weigle, Michael L. Nelson, Fernando Melo, Daniel Bicho, and Daniel Gomes. 2019. MementoMap Framework for Flexible and Adaptive Web Archive Profiling. In *Proceedings of the 19th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '19)*. 172–181. https://doi.org/10.1109/JCDL.2019.00033
[14] Ahmed AlSum, Michele C. Weigle, Michael L. Nelson, and Herbert Van de Sompel. 2013. Profiling Web Archive Coverage for Top-Level Domain and Content Language. In *Proceedings of the 17th International Conference on Theory and Practice of Digital Libraries (TPDL '13)*. 60–71. https://doi.org/10.1007/978-3-642-40501-3_7
[15] Ahmed AlSum, Michele C. Weigle, Michael L. Nelson, and Herbert Van de Sompel. 2014. Profiling Web Archive Coverage for Top-Level Domain and Content Language. *International Journal on Digital Libraries* 14, 3-4 (2014), 149–166. https://doi.org/10.1007/s00799-014-0118-y
[16] Nicolas Bornand, Lyudmila Balakireva, and Herbert Van de Sompel. 2016. Routing Memento Requests Using Binary Classifiers. In *Proceedings of the 16th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '16)*. 63–72. https://doi.org/10.1145/2910896.2910899
[17] Helge Holzmann, Vinay Goel, and Avishek Anand. 2016. ArchiveSpark: Efficient Web Archive Access, Extraction and Derivation. In *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries (JCDL '16)*. 83–92. https://doi.org/10.1145/2910896.2910902
[18] Internet Archive. 2003. CDX File Format. http://archive.org/web/researcher/cdx_file_format.php.
[19] ISO 28500:2017. 2017. WARC File Format. https://iso.org/standard/68004.html.
[20] Andy Jackson. 2014. Messy Web Archive Collections. https://twitter.com/anjacks0n/status/466690812269846528.
[21] Martin Klein, Lyudmila Balakireva, and Harihar Shankar. 2019. Evaluating Memento Service Optimizations. In *Proceedings of the 19th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '19)*. 182–185. https://doi.org/10.1109/JCDL.2019.00034
[22] Luis Meneses, Richard Furuta, and Frank Shipman. 2012. Identifying 'Soft 404' Error Pages: Analyzing the Lexical Signatures of Documents in Distributed Collections. In *Proceedings of the 2nd International Conference on Theory and Practice of Digital Libraries (TPDL '12, Vol. 7489)*. 197–208. https://doi.org/10.1007/978-3-642-33290-6_22
[23] Michael L. Nelson and Herbert Van de Sompel. 2018. Adding the Dimension of Time to HTTP. In *The SAGE Handbook of Web History*.
[24] Nick Ruest, Jimmy Lin, Ian Milligan, and Samantha Fritz. 2020. The Archives Unleashed Project: Technology, Process, and Community to Improve Scholarly Access to Web Archives. In *Proceedings of the 20th ACM/IEEE Joint Conference on Digital Libraries (JCDL '20)*. 157–166. https://doi.org/10.1145/3383583.3398513
[25] Robert Sanderson, Herbert Van de Sompel, and Michael L. Nelson. 2012. IIPC Memento Aggregator Experiment. http://www.netpreserve.org/sites/default/files/resources/Sanderson.pdf.
[26] Herbert Van de Sompel, Michael L. Nelson, and Robert Sanderson. 2013. HTTP Framework for Time-Based Access to Resource States – Memento. RFC 7089.
[27] Herbert Van de Sompel, Michael L. Nelson, Robert Sanderson, Lyudmila L. Balakireva, Scott Ainsworth, and Harihar Shankar. 2009. *Memento: Time Travel for the Web*. Technical Report arXiv:0911.1112. https://arxiv.org/abs/0911.1112

Russia-Ukraine News on the Dark Web

Grant C. Atkins, Aaron T. Buehne, Abby Mabe, Zak Zebrowski, Justin F. Brunelle
{gatkins;abuehne;amabe;zaz;jbrunelle}@mitre.org
The MITRE Corporation
USA

ABSTRACT

This lightning talk will highlight the importance of archiving the dark web. Some news media sites (e.g., BBC News, New York Times) have a dark web mirror of their surface web sites. However, the representations of the dark web mirror will lag and sometimes present different information from the surface web counterparts. We will discuss the conflict in Ukraine and the differences in coverage on news media between the surface and dark webs as motivation to establish a dark web archive to preserve media coverage of this event for future historians.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

web archiving, dark web, web crawling

ACM Reference Format:
Grant C. Atkins, Aaron T. Buehne, Abby Mabe, Zak Zebrowski, Justin F. Brunelle. 2022. Russia-Ukraine News on the Dark Web. In *Proceedings of Web Archiving and Digital Libraries Workshop 2022 (WADL '22).* ACM, New York, NY, USA, 1 page. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The dark web is accessed through privacy-preserving tools such as the Tor browser. The dark web's privacy protection offers anonymity to users accessing both illicit and non-illicit content. Despite the prevalence of illicit material, the dark web provides users – such as those wishing to access media – an ability to safely consume non-illicit news media [2]. Several news sites have both surface web and dark web mirrors (e.g., Buzzfeed News, BBC News, New York Times). In this lightning talk, we demonstrate that while the content on the surface and dark web are largely the same, the presentation, emphasis, and features of the representations sometimes differ. This means that dark web users are sometimes presented information that differs between the surface and dark webs.

The recent conflict in Ukraine has emphasized the social importance of the dark web. Several news and media sites have increase their advertising of the dark web mirrors of their surface web sites, such as BBC News advertising their dark web mirror in both Ukrainian and Russian. Twitter has also launched a dark web mirror during the conflict. As such, future historians analyzing the consumption and delivery of news media surrounding the crisis in Ukraine must mention the availability of dark web resources that advertise anonymity-protecting news media consumption. To date, only Archive.is has a dark web service for archiving dark web pages. However, this is an on-demand page-at-a-time archiving service and is not designed for site- and dark-web-scale archiving.

In this lightning talk, we discuss a longitudinal sampling of news media on both the surface and dark web mirrors of news pages. Our goal is to motivate increased attention on the importance of archiving the dark web [1].

2 DATASET

Since November 14, 2021, our team has been performing daily crawls of 44 non-illicit surface web pages and their dark web mirrors. The resulting surface and dark web archive includes several news pages that were covering the crisis in Ukraine. As of March 28, the resulting archive contains 830992 mementos totaling 114GB of storage across the 134 days of observations.

These observations contain several differences, including how articles are presented on the homepages of the news sites. This shows that users will see different content when accessing the sites through the surface and dark webs. The delay in crawled the surface web and dark web mirrors ranges from 0 seconds to 5 minutes which accounts for some of the differences but other differences occur due to how the content is provided. We provide examples of both in our lightning talk.

3 DISCUSSION TOPICS

From our dataset described in Section 2, we will use our lightning talk to provide examples of how archived dark web pages and their surface web counterparts differ, such as article placement, titles, and images used for news. We will highlight the differences between the pages to motivate the establishment of a dark web archival ecosystem by the web archiving community.

ACKNOWLEDGMENTS

This work was funded by MITRE's IR&D program; we thank the numerous collaborators that assisted with the award and execution of this research project.

REFERENCES

[1] Justin F. Brunelle, Ryan Farley, Grant Atkins, Trevor Bostic, Marites Hendrix, and Zak Zebrowski. 2021. *Introducing Dark Web Archival Framework.* Technical Report arXiv:2107.04070.
[2] Eric Jardine. 2018. Tor, what is it good for? Political repression and the use of online anonymity-granting technologies. *New Media & Society* 20, 2 (2018), 435–452.

WADL '22, June 20, 2022, Virtual

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

Archiving Source Code in Scholarly Content: One in Five Articles References GitHub

Emily Escamilla
evogt001@odu.edu
Old Dominion University
Norfolk, Virginia, USA

Talya Cooper
tc3602@nyu.edu
New York University
New York, New York, USA

Vicky Rampin
vs77@nyu.edu
New York University
New York, New York, USA

Martin Klein
mklein@lanl.gov
Los Alamos National Laboratory
Los Alamos, New Mexico, USA

Michele C. Weigle
mweigle@cs.odu.edu
Old Dominion University
Norfolk, Virginia, USA

Michael L. Nelson
mln@cs.odu.edu
Old Dominion University
Norfolk, Virginia, USA

ABSTRACT

The definition of scholarly content has expanded to include the data and source code that contribute to a publication. Major archiving efforts to preserve scholarly content in PDF form (LOCKSS, CLOCKSS, and Portico) are well underway, but no analogous effort has yet emerged to preserve the data and code referenced in those PDFs, particularly the scholarly code hosted online on Git Hosting Platforms (GHPs). Similarly, the Software Heritage Foundation is working to archive public source code, but there is value in archiving the contemporary look and feel of the GHPs, including issue threads, pull requests, and wikis while maintaining their original URIs. For academic projects where reproducibility matters, this ephemera adds important context. To understand and quantify the scope of this problem, we analyzed the use of GHP URIs in the arXiv corpus from April 2007 to December 2021 including the number and frequency of GHP URIs. In total, there were 217,106 URIs to GitHub, SourceForge, Bitbucket, and GitLab repositories across the 1.56 million publications in the arXiv corpus. Additionally, the frequency of URLs to GHPs is increasing. We found that GitHub, GitLab, SourceForge, and Bitbucket were collectively linked to 206 times in 2008 and 74,227 times in 2021. As shown in Figure 1, in 2021, one out of five publications included a URI to GitHub which is a significant increase from 2007 where less than 1% of publications contained a link to a GHP. However, the complexity of GHPs like GitHub is not amenable to conventional Web archiving techniques. Therefore, the growing use of GHPs in scholarly publications points to an urgent and growing need for dedicated efforts to archive the holdings of GHPs to preserve research code and its scholarly ephemera.

CCS CONCEPTS

• **Information systems → Digital libraries and archives**; **Open source software**.

KEYWORDS

Web Archiving; Digital Preservation; Memento; Open Source Software

ACM Reference Format:
Emily Escamilla, Talya Cooper, Vicky Rampin, Martin Klein, Michele C. Weigle, and Michael L. Nelson. 2022. Archiving Source Code in Scholarly Content: One in Five Articles References GitHub. In *Proceedings of Web Archiving and Digital Libraries (WADL '22).* ACM, New York, NY, USA, 1 page. https://doi.org/XXXXXXX.XXXXXXX

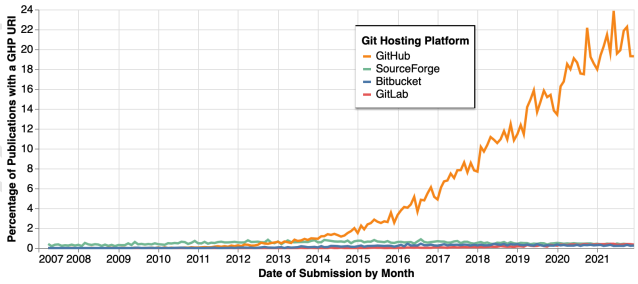


Figure 1: The percentage of publications with a URI to a repository platform over time

WADL '22, June 20, 2022, Virtual

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

Arch-It

Helge Holzmann[1] , Nick Ruest[2] , Jefferson Bailey[1] , Alex Dempsey[1] , Samantha Fritz[3] , Ian Milligan,[3] , and Kody Willis[1]

[1] Internet Archive
[2] Digital Scholarship Infrastructure Department, York University
[3] Department of History, University of Waterloo

ABSTRACT

Over the past quarter-century, web archive collection has emerged as a user-friendly process thanks to cloud-hosted solutions such as the Internet Archive's Archive-It subscription service. Despite advancements in collecting web archive content, no equivalent has been found by way of a user-friendly cloud-hosted analysis system. Web archive processing and research require significant hardware resources and cumbersome tools that interdisciplinary researchers find difficult to work with. In this paper, we present ARCH (Archives Research Compute Hub)[1], an interactive interface, closely connected with Archive-It, engineered to provide analytical actions, specifically generating datasets and in-browser visualizations. It efficiently streamlines research workflows while eliminating the burden of computing requirements. Building off past work by both the Internet Archive (Archive-It Research Services) and the Archives Unleashed Project (the Archives Unleashed Cloud), this merged platform achieves a scalable processing pipeline for web archive research.

1 INTRODUCTION

While collecting web archive content has matured into a user-friendly process, thanks in no small part to cloud-hosted solutions such as the Internet Archive's Archive-It service, this ease-of-use has not been matched on the analysis side. We accordingly need a user-friendly system that can enable the creation of research datasets from web archives so that researchers can work with material at scale.

In this paper we present an overview of the Archives Research Compute Hub (ARCH) as opposed to our more robust examination of the platform [2]. As this paper is intended as a demonstration at WADL, we draw heavily on that paper. ARCH is a production system tightly integrated with the Internet Archive infrastructure and services. It grew out of the Archives Unleashed Cloud: a proof-of-concept platform that demonstrated the ability of a web browser-based system to power backend Apache Spark-driven jobs on web archival datasets [6]. Powered by the Archives Unleashed Toolkit and the Internet Archive's Sparkling data processing library[2], the ARCH platform will become a complementary component of the Internet Archive's Archive-It system.

The Archives Unleashed project aims to address this problem [7] by being for web archive analysis as Archive-It is for web archive capture: powerful, scalable, and above all, accessible and intuitive for users. The Archives Unleashed Cloud (2017-2020) provided user access to the features of the Archives Unleashed Toolkit in a cloud-hosted environment [6]. The Cloud worked with Archive-It collections, using APIs to transfer data from the Internet Archive to Compute Canada cloud-hosted infrastructure. Yet the initial approach of having a separate analysis service presented shortcomings. When a user wished to carry out analysis, data had to be transferred. More importantly, connections between Archive-It and the Cloud required a complicated interplay of APIs, bulk data transfers, and other workflows, leaving a separate analysis service vulnerable to network disruptions or changing standards. These factors combined to make it an interesting proof-of-concept but one that presented considerable sustainability challenges. Our goal, then, was to integrate Archives Unleashed tools with the Internet Archive's Archive-It service.

2 RELATED WORK AND PROJECT CONTEXT

Established in 2017, the Archives Unleashed project recognizes the collective need among researchers, librarians and archivists for analytical tools, community infrastructure, and accessible web archival interfaces. To this end, the project aspires to make petabytes of historical internet content accessible to scholars and others interested in researching the recent past. Between 2017 and 2020, the project focused on developing the "Archives Unleashed Cloud," a web-based interface for working with web archives at scale using the Archives Unleashed Toolkit and Apache Spark [6]. This work built on the project's long-standing interests in building exploratory search interfaces for web archive collections [3]. Similar noteworthy work includes the SolrWayback project from The Royal Danish Library. Combining Apache Solr with OpenWayback or pywb, SolrWayback provides search and discovery of web archive collections, as well as replay, and a number of analysis and visualization features [5].

In 2020, the project's first phase was completed. The next phase involved exploring integration and collaboration with the Internet Archive [7]. We were influenced by the global adoption of the Internet Archive's Archive-It subscription service and the stability of the Apache Spark platform [1].

Since the launch of the Internet Archive's subscription service in 2006, over 700 institutions from 23 countries have used Archive-It to preserve over two petabytes of data consisting of over 40

[1] https://github.com/internetarchive/arch

[2] https://github.com/internetarchive/Sparkling

WADL'22, June 20, 2022, Cologne, Germany

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

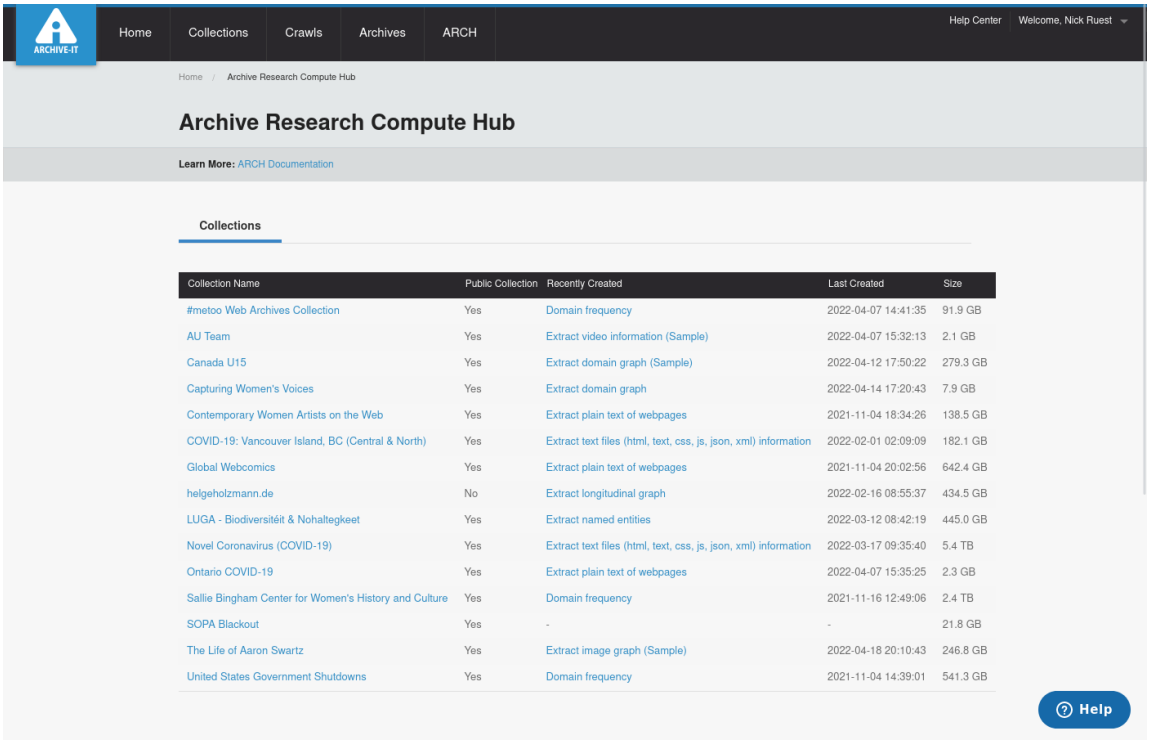


Figure 1: ARCH main collections page.

billion born-digital, web-published records in over 12,000 public collections. It is a successful service. A survey by the National Digital Stewardship Alliance reported that by 2017, 94% of surveyed institutions were using Archive-It to preserve web material – and an additional 4% were using other services provided by the Internet Archive [4]. Archive-It is thus effectively the de-facto platform for web archiving, used by nearly all Association of Research Library members, hundreds of other higher education, memory institutions, public libraries, governments, and non-profit organizations.

Despite this widely-accepted solution for the capture of web material, the problem of analysis remains. By this, we refer to at-scale explorations of data that require more than the replay interface of the Wayback Machine. While web archive data is captured and preserved in the ISO-standard WARC file format, the formation of a scholarly ecosystem around web archive analysis has been slow.[3]

3 ARCHIVES RESEARCH COMPUTE HUB

In this section, we present our interface and its broader context within Archive-It. As of December 2021, ARCH has both feature parity with the earlier Archives Unleashed Cloud, and also additional functionality to generate several additional datasets. As functionality from the earlier Cloud was ported, all features were redesigned and reimplemented. We addressed known issues, fixed existing bugs, and more importantly, implemented an approach that scales to meet our needs.

3.1 Design Considerations

ARCH now runs on an infrastructure that is physically connected to Archive-It servers and computing infrastructure, mitigating the need to copy data before processing. As not all Archive-It data is kept in its dedicated computing cluster, ARCH is connected to the Internet Archive's long-term storage system (the "Petabox") to fetch missing data. In addition, we implemented a smart caching mechanism to avoid re-fetches for consecutive access to the same data. Cognizant of researcher needs beyond Archive-It collections, we also support custom collections which can be located on ARCH's own cluster.

Given the sensitive nature of web archival collections, we have implemented a user and permissions system. There are two authentication providers: Archive-It user accounts and dedicated ARCH users. For Archive-It users, we rely on Archive-It's internal permissions process. We have also implemented a permission control access that allows ARCH and Archive-It users to cross-access additional Archive-It collections (pending permission from the data collector) and ARCH custom collections.

To control jobs and enable the downloading of files via different tools (browser-based downloads for smaller files, command line for larger ones), we provide multiple APIs and authentication methods. While the actual implementation details are beyond the scope of this paper, ARCH is a native Scala application built using Scalatra[4]. The underlying toolkit is based on the Archives Unleashed Toolkit (previously known as Warcbase) as well as the Internet Archive's Sparkling library. Jobs and queues are controlled via APIs,

[3]The best place to learn about available tools is the "Web Archiving Awesome List" maintained by the International Internet Preservation Consortium and researchers across the field. See https://github.com/iipc/awesome-web-archiving.

[4]https://scalatra.org/

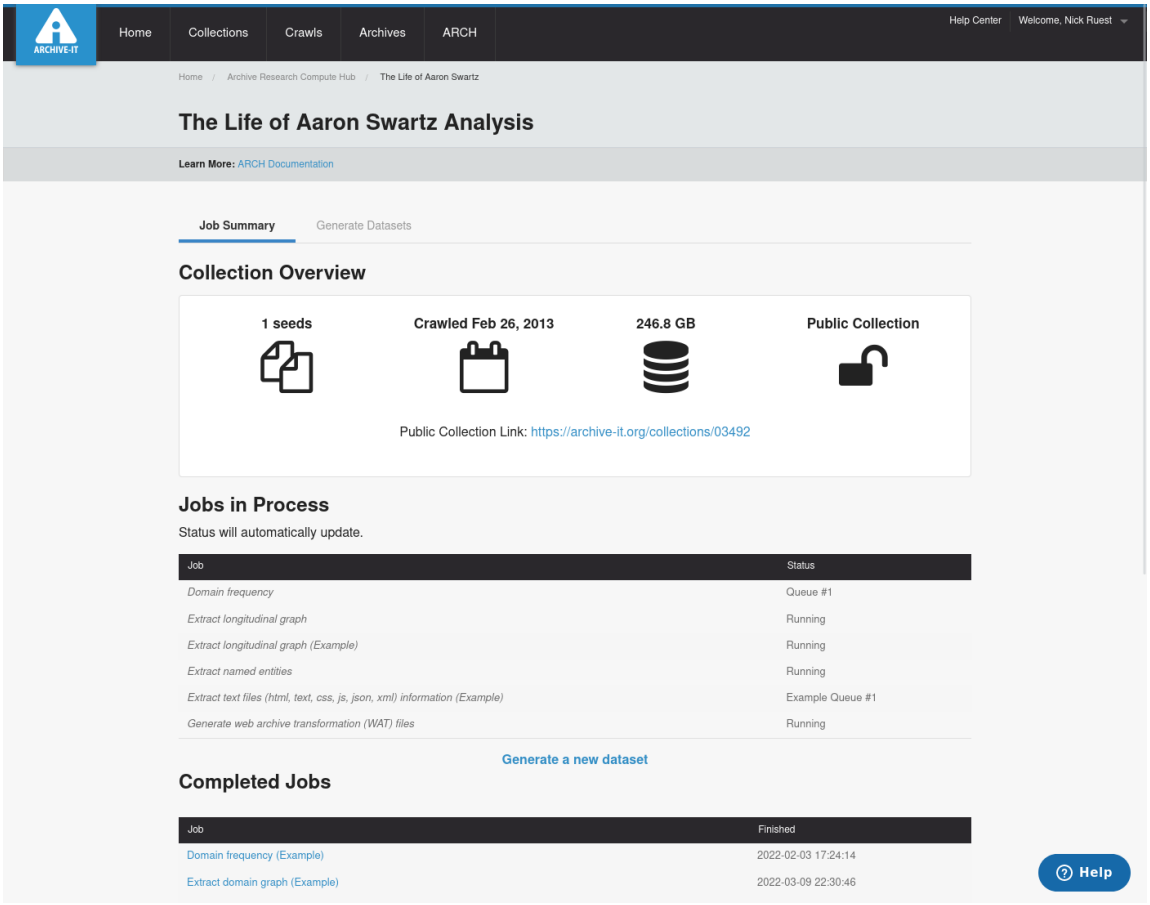


Figure 2: ARCH job summary page.

enabling Spark jobs to be chained with post-processing jobs, as well as separate queues for example/full jobs, Spark operations, and post-processing.

3.2 User Interface

ARCH's interface consists of four levels. These guide users to interact with their collections by generating datasets for analysis and engaging with in-browser features. The goal of ARCH is to provide an efficient, streamlined workflow without burdening users with computing requirements or actions.

The first level is the **main collections** page. All of a user's Archive-It collections are presented in a table (Figure 1), accompanied by information about the most recent analysis conducted and other collection-based metadata. Each collection title provides an access point for conducting analysis.

The second is a **job summary** page, where users can generate, download, and monitor derivative datasets. An overview of the collection identifies basic metadata about the collection, including collection size and whether it is a public or private collection. The second main feature of this space provides tables that summarize "Jobs in Process" - the stage and queue of any current jobs being run - and a "Completed Jobs" table identifying all datasets previously generated, noting an accompanying date/time stamp (Figure 2).

The third level is the **generation of datasets** (Figure 3). As a core feature of ARCH, users can generate sixteen different datasets for scholarly exploration. These datasets are categorized into four main themes of analysis (Table 1). This supports different dataset generation jobs based on a generic interface to start jobs, monitor their status, and explore the ensuing output.

Finally, the last level are the **derivative dataset** pages themselves. For each dataset generated, users can access an overview

Dataset Category	Description
Collection	Offers an overview of a collection by looking at simple statistical counts.
Network	Produces files that provide network graphs for analysis and offer an opportunity to explore the way websites link to each other.
Text	Allows the user to explore text components of a web archive, including extracted "plain text" HTML, CSS, and other web elements.
File formats	Provides files that contain information on certain types of binary files found within a web archive.

Table 1: ARCH Datasets

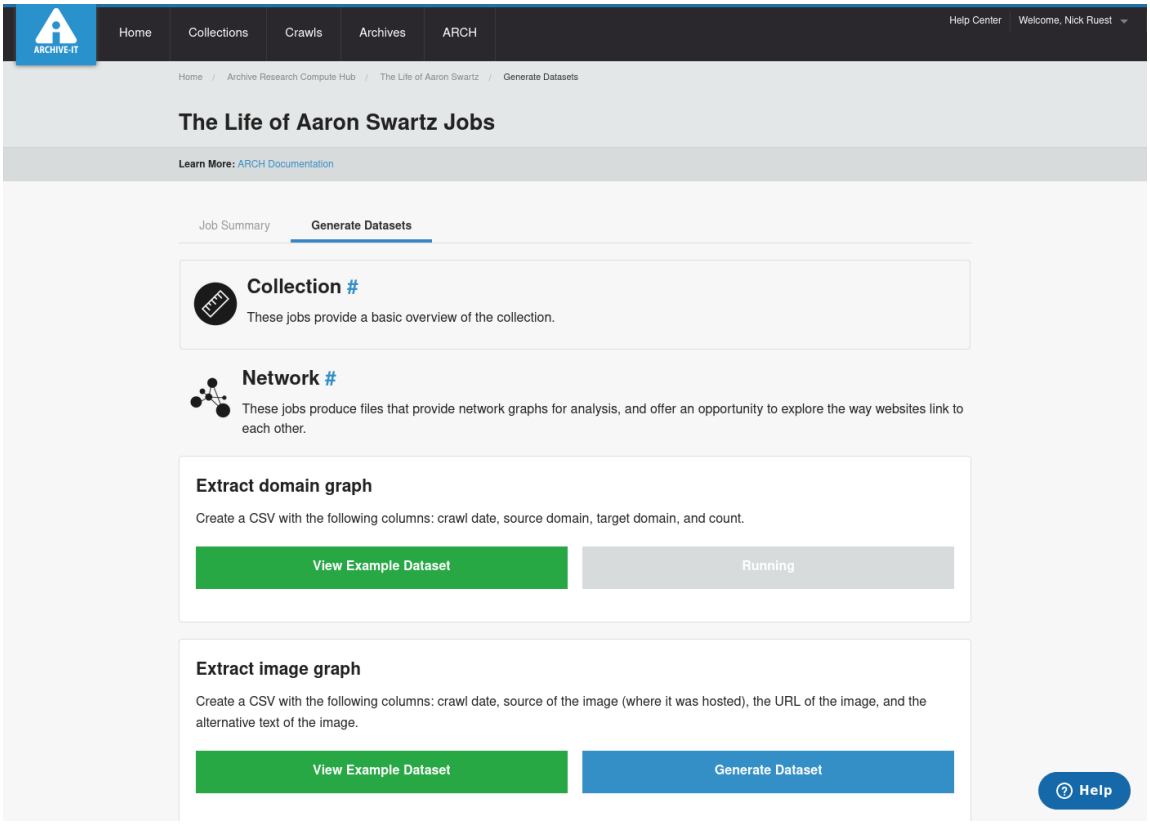


Figure 3: The "generate datasets" page in the ARCH interface.

page of the dataset, which provides metadata (file name, file size, results count, and date completed), download options, a preview of up to 100 lines, and the option to re-run any job. An example of this can be seen in Figure 8. Where possible, in-browser visualization and charts present a summary of the data. For instance, the *extract web graph* dataset page offers an interactive network graph that users can explore using simple functionalities like zooming in and out on modes and clusters and exporting a high-resolution image. These datasets are intended be downloaded and further explored with other analytical tools and methods.

4 CONCLUSION AND EVALUATION

We have presented ARCH, the Archives Research Compute Hub, a novel data processing platform for web archives, closely integrated with Archive-It.

The design process for ARCH involved a variety of interconnected stages, from designing wireframes to building infrastructure to connecting backend processes to the user interface. User experience (UX) evaluations were essential for measuring and understanding the needs of researchers. As such, the team conducted iterative and multi-staged user testing and surveying to assess user needs and experience. By engaging with Archive-It power users and Archives Unleashed Cloud alumni in five closed user testing rounds, our team gathered feedback and initial impressions of ARCH. Testing was primarily conducted through surveys, which collected qualitative and quantitative data to determine user satisfaction and experience.

Findings from the survey were translated into actionable tickets to provide action-based tasks for development cycles. We were able to implement the majority of action items, with some needing further planning and only a few that fell outside of our scope of work.

As a multi-stage UX testing process, each subsequent round of testing served as another opportunity to review and refine impressions of prior development and enhancements — improving our accuracy and capacity to match user needs at each stage. Our final rounds of testing concluded in early 2022. This final process served two purposes. First, we expanded testing to include a larger group (approximately 100 participants) to serve as a stress test. As this was our largest testing group to date, this offered an opportunity to verify ARCH's robustness, capacity, and efficiency while noting any bottlenecks or areas for improvement. Second, we conducted focused interviews with a small group of researchers who have extensively used ARCH since August 2021. These researchers were ideal for understanding the real-life application and use cases of the web archives research journey.

ACKNOWLEDGEMENTS

This research was supported by the Andrew W. Mellon Foundation's Public Knowledge program, the Social Sciences and Humanities Research Council of Canada, as well as Start Smart Labs, the University of Waterloo, and York University.

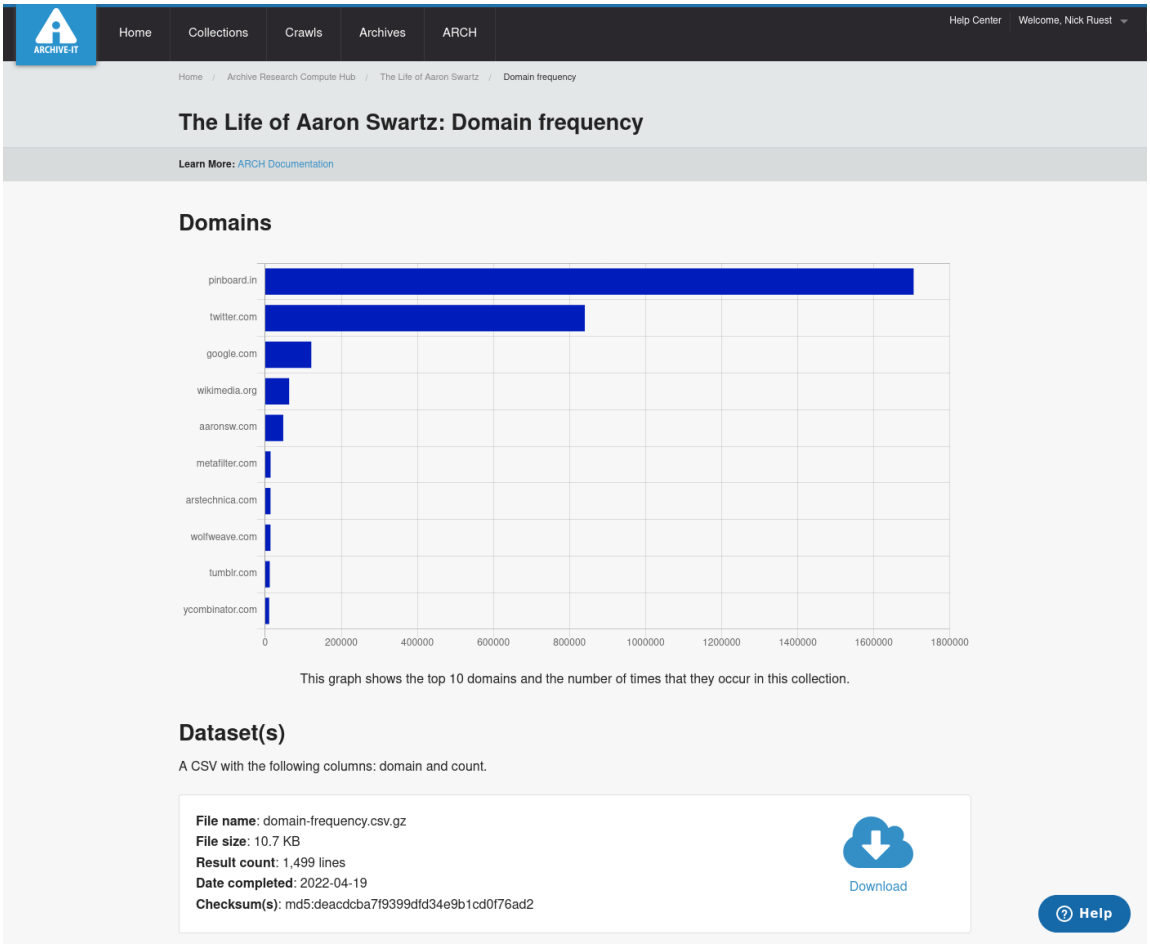


Figure 4: One of ARCH's Dataset Results Pages.

REFERENCES

[1] Helge Holzmann, Vinay Goel, and Avishek Anand. 2016. ArchiveSpark: Efficient Web Archive Access, Extraction and Derivation. In *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries*. ACM, Newark New Jersey USA, 83–92. https://doi.org/10.1145/2910896.2910902

[2] Helge Holzmann, Nick Ruest, Jefferson Bailey, Alex Dempsey, Samantha Fritz, Peggy Lee, and Ian Milligan. 2022. ABCDEF - The 6 key features behind scalable, multi-tenant web archive processing with ARCH: Archive, Big Data, Concurrent, Distributed, Efficient, Flexible. *International Journal of Digital Humanities*. https://doi.org/10.1145/3529372.3530916

[3] Andrew Jackson, Jimmy Lin, Ian Milligan, and Nick Ruest. 2016. Desiderata for Exploratory Search Interfaces to Web Archives in Support of Scholarly Activities. (2016). https://doi.org/10.1145/2910896.2910912 Accepted: 2016-05-09T11:57:37Z.

[4] Katherine Kim, Wayne Graham, Paige Walker, National Digital Stewardship Alliance (NDSA), Carol Kussmann, and Aliya Reich. 2018. 2017 Web Archiving in the United States - A 2017 Survey. (Oct. 2018). https://doi.org/10.17605/OSF.IO/3QH6N Publisher: OSF.

[5] The Royal Danish Library. 2021. SolrWayback. https://github.com/netarchivesuite/solrwayback

[6] Nick Ruest, Samantha Fritz, Ryan Deschamps, Jimmy Lin, and Ian Milligan. 2021. From archive to analysis: accessing web archives at scale through a cloud-based interface. *International Journal of Digital Humanities* 2, 1 (Nov. 2021), 5–24. https://doi.org/10.1007/s42803-020-00029-6

[7] Nick Ruest, Jimmy Lin, Ian Milligan, and Samantha Fritz. 2020. The Archives Unleashed Project: Technology, Process, and Community to Improve Scholarly Access to Web Archives. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020 (JCDL '20)*. Association for Computing Machinery, New York, NY, USA, 157–166. https://doi.org/10.1145/3383583.3398513

WACZ

Portable Web Archives

Ed Summers
Stanford University
ehs@pobox.com

Ilya Kreymer
Webrecorder
ilya@webrecorder.org

Cade Diehm
New Design Congress
cade@newdesigncongress.org

Web Archive Collection Zipped (WACZ) is a new packaging format that provides portability features for decentralized web archives. Similar to the Open Document Format, WACZ is a convention for bundling collection metadata, indexes, WARC data, and fixity information together as a ZIP file in order to make web archives available as discrete files, not unlike other types of digital objects, like videos or PDFs.

Classical web archives are deployed as server-side web applications that users access in their web browser to view past representations of the web. This process is often referred to as “replay” and it is dependent on externally owned, complex software infrastructures which crawl, store, index, and retrieve archived content.

WARC is a standard serialization format for HTTP network traffic that allows web crawlers to store the raw network data in a consistent way. However key information, such as the name and description of the collection, who created it, why they created it, the URL index, as well as useful entry-point pages, are not represented in WARC. While this allows for a separation of concerns between raw crawled data and other web archive data, as a result, most web archives have implemented these features in application-specific ways, limiting data portability.

The lack of portability resulted in web archives being developed as complex, monolithic application infrastructures, tied to bespoke tooling which can become single points of failure, and are expensive to maintain as they grow in size and use. In turn this concentration shapes the types of actors who can create web archives and provide continued access to them [1].

WACZ is built on top of the Frictionless Data Package [2] standard, which provides a convention for JSON encoded descriptive metadata and a manifest for describing the enclosed data files and their fixity values for verification. A WACZ file includes one or more WARC files and their corresponding indexes. WACZ specifies a standard index format (CDXJ), that allows entry metadata to be expressed using an extensible JSON block. The descriptive and structural metadata along with the WARC and CDXJ files compressed as a ZIP file comprise a complete WACZ, which can be transferred between systems and placed on the web as a resource just like other media formats.

```
example.wacz
├── archive
│   └── data.warc.gz
├── datapackage.json
├── datapackage-digest.json
├── indexes
│   └── index.cdx
└── pages
    └── pages.jsonl
    └── extraPages.jsonl
```

Once hosted on a web server, standard HTTP Range requests can be used to efficiently retrieve the components of a WACZ for replay. A property of the ZIP format is that all the contents can be accessed via random access without downloading the entire file. In this way, much like audio and video players, a client-side web archive replay system can easily load one or more WACZ files. ReplayWeb.page provides a reference implementation for this replay process, which is also available as a reusable Web Component. To date, WACZ files as large as 1.3TB have been replayed. However there remain two primary areas of future work, which we are seeking community feedback on.

We anticipate that it will be useful to *aggregate* WACZ files for seamless replay. For example, archives could iteratively publish web crawls over time as a series of WACZ files. Groups of WACZ files could represent thematic collections, or virtually unify WACZ files that are published by multiple collaborating parties at different locations on the web.

Fully portable web archives also raise new use cases and potential threats around issues of trust, integrity and identity. WACZ ensures that all web archive data (including individual WARC and CDXJ records) are hashed and cryptographically signed. However choices around signing mechanics, and key identity entail positive and negative social impacts that must be factored and designed for.

REFERENCES

[1] H. Zinn, “Secrecy, archives, and the public interest,” *Midwest. Arch.*, vol. 2, no. 2, pp. 14–26, 1977.

[2] P. Walsh and R. Pollock, “Data Package,” Frictionless Data, Nov. 2007. [Online]. Available: https://specs.frictionlessdata.io/data-package/

Moving the End of Term Web Archive to the Cloud to Encourage Research Use and Reuse

ANONYMOUS AUTHOR(S)

The End of Term Web (EOT) Archive is a collaborative project with a goal of collecting the United States federal web, loosely defined as .gov and .mil, every four years coinciding with presidential elections and often a transition in the Executive Branch of the government. In 2021 the End of Term team began to process the longitudinal web archive for EOT-2008, EOT-2012, EOT-2016, and EOT-2020 to move into the Amazon S3 storage service as part of the Amazon Open Data Program. This effort adopted tools, structures, and documentation developed by Common Crawl in an effort to maximize potential research access and reuse of existing tools and documentation. This paper presents the process of organizing, staging, processing, and moving these collections into the Amazon cloud.

Additional Key Words and Phrases: web archives, cloud storage, research datasets, web archive datasets

ACM Reference Format:
Anonymous Author(s). 2022. Moving the End of Term Web Archive to the Cloud to Encourage Research Use and Reuse. In . ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

1 BACKGROUND

The End of Term (EOT) Web Archive [1] is a collaborative project with a goal of collecting the United States federal web, loosely defined as .gov and .mil, every four years coinciding with presidential elections and often a transition in the Executive Branch of the government. Starting in 2008 [14], this project documented the federal web before the transition of the Bush administration to the Obama administration, then documented the transition from one Obama term to another in 2012, the transition from Obama to Trump in 2016 [13], and the transition from Trump to Biden in 2020. In total, the EOT has collected nearly 500TB of content in its four iterations. The EOT is an ad-hoc collaboration that comes together every four years to plan, publicize, and execute the crawls related to this effort. Long-term access to the content is often a more challenging component of the process. So far, access has been provided by the Internet Archive through different configurations of their Wayback Machine and currently access is provided by the Global Wayback collection. Additionally, some members of the EOT team have curated and hosted secondary access points to the crawled content in their own infrastructure to provide redundancy in access. This provides a minimum level of access to the harvested resources for general users, but over the years the EOT team has found that there are logistical limitations in place when users want to use the EOT archives to answer computationally focused research questions that would require larger portions of the archive. In the Fall of 2021, the EOT began to explore working with the AWS Open Data Sponsorship Program [3] to host a copy of the four EOT web crawls as a longitudinal dataset.

[1] End of Term Web Archive : https://eotarchive.org/

Web Archiving and Digital Libraries '22,
,

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

2 ORGANIZING THE COLLECTION

Before moving the EOT collections to the cloud, there were a number of decisions to be made about how the data would be organized and made available. The AWS Open Data Sponsorship Program provides high-speed cloud storage for open datasets through the Amazon Simple Storage Service (S3) [1]. The EOT team looked for prior work in this area and decided on the organizational structures in place in the Common Crawl program [9]. Common Crawl broadly crawls the web and provides the data freely to users for research and analysis. In addition to the crawl data, Common Crawl will create derivative formats for each WARC file consisting of a Web Archive Transformation (WAT) [5] which provides content-metadata about the crawled resources such as out links, anchor text, and overall structural information. Another file provided is Web Extracted Text (WET) files that present just the text for formats like HTML and TXT to the user. These WAT and WET files are created for every WARC file in the dataset. An index of all captured content is provided in the CDXJ format [6] and organized using the ZipNum structure. Finally, the CDXJ data is processed and compiled into a columnar data format called Parquet [2] that provides another entry point into the collection that can be used by many common tools and services. By adopting these structures for the EOT collections, the team was able to build on existing workflows and leverage tools used by Common Crawl in their processing pipeline. Another goal was to work with researchers already using Common Crawl data to allow the EOT dataset to fit into their research workflows and tools with little modification. Finally, the ability to reuse and adapt existing documentation about the formats and processes was also a benefit of using existing formats.

3 LAYOUT OF THE CRAWLS

One goal of this project is to provide self contained versions of each crawl. To enable this, each of the four End of Term crawls, 2008, 2012, 2016, 2020 would have their own path structure in the Amazon S3 buckets. The EOT-2008, EOT-2012, EOT-2016, and EOT-2020 crawls can be seen as analogous to the Common Crawl monthly crawl structures in a straightforward way. The next thing the EOT team wanted to maintain was the provenance of which institution was responsible for the crawling of the data. For example, in the EOT-2008 dataset, the crawls were conducted by the California Digital Library

[2] Apache Parquet https://parquet.apache.org/

(CDL) [3], the Internet Archive (IA) [4], the Library of Congress (LOC) [5] and the University of North Texas Libraries (UNT) [6]. The Common Crawl organizational structure includes a concept of segments that divide a crawl into subsets. These segments are holdovers from the distributed Nutch crawler that they use for collecting content. Each segment in the Common Crawl dataset contains under 10,000 WARC files and the EOT team felt that provided for a reasonable limit for others looking to download and work with the content locally. Many of the crawl partners generated over 10,000 WARC files in the EOT project and it was necessary to create several segments per crawling partner. An example of this structure can be seen in the example below.

```
crawl-data/EOT-2008/segments/CDL-000/
crawl-data/EOT-2008/segments/CDL-001/
crawl-data/EOT-2008/segments/CDL-002/
crawl-data/EOT-2008/segments/IA-000/
crawl-data/EOT-2008/segments/IA-001/
```

This structure works for organizing files in a normal POSIX filesystem as well as in an object store like S3 by making use of the common concept of a prefix. Inside each segment, another prefix/folder structure for WARC, WAT, WET, and CDX files was created. This results in the final structure of a segment as you can see below.

```
crawl-data/EOT-2008/segments/CDL-000/cdx/
crawl-data/EOT-2008/segments/CDL-000/warc/
crawl-data/EOT-2008/segments/CDL-000/wat/
crawl-data/EOT-2008/segments/CDL-000/wet/
```

The ZipNum and Parquet indexes are stored in a similar layout but with a path structure separate from the crawl data. The layout is presented below.

```
cc-index/collections/EOT-2008/indexes/
cc-index/collections/EOT-2012/indexes/
cc-index/collections/EOT-2016/indexes/
cc-index/collections/EOT-2020/indexes/
cc-index/table/eot-main/warc/crawl=EOT-2008/
cc-index/table/eot-main/warc/crawl=EOT-2012/
cc-index/table/eot-main/warc/crawl=EOT-2016/
cc-index/table/eot-main/warc/crawl=EOT-2020/
```

4 INVENTORY OF DATA

One of the surprisingly challenging parts of this project was to completely identify the crawl data within different institutions' respective repository infrastructure. Many institutions will include their web archives in both a preservation repository to provide long-term stewardship, replication, and structured access to the files, as well as locating them in a secondary access system where they are indexed and served using a replay system such as Open Wayback [7] or pywb [8]. Because of this, it can be challenging to identify all of the components of a large web crawl within a repository. Additionally, because the EOT crawls were completed by different institutions and then aggregated into single collections, it was challenging to identify contributed WARC files from the locally crawled content. Finally, because the EOT projects generally last from September until March of the following year, there are many individual crawls during that period that all need to be accounted for during the process. In the EOT project, the EOT-2008 and EOT-2012 crawls were fully replicated by three institutions: the Internet Archive, Library of Congress, and the UNT Libraries. For the EOT-2016 and EOT-2020 crawls, the only complete copy of the data is held at the Internet Archive, with the crawling partners maintaining a copy of their own crawled data. Because of this distribution of content, the EOT team was able to split the responsibility of inventorying all content between the Internet Archive for the EOT-2016 and EOT-2020 crawls and UNT for the EOT-2008 and EOT-2012. As an example to demonstrate this effort, the EOT-2008 crawl totalled 16TB of data and was distributed across 110 Archival Information Packages in the UNT Libraries' Coda preservation repository [11]. Similar divisions of a complete EOT crawl across dozens to hundreds of archival packages occurred at both IA and UNT for the other crawls.

Once inventories were complete, the next step was to download the archival packages, verify that everything was complete and valid based on package checksums, and then reorganize the WARC content into the structures mentioned above. The EOT-2008 crawls contained both WARC and ARC files, whereas the remaining crawls only held WARC files. The decision was made early in the project to not rename files but to leave them as they were originally contributed for better provenance and lineage. There were a small number of WARC/ARC files (36) from the CDL dataset in EOT-2008 that had duplicate filenames but different content and those were renamed with "-duplicate-name-" inserted into the filename to allow them to be included. The EOT team decided early in the process not to concatenate, modify, or convert the content files into other formats to preserve integrity at file and record levels. This would have included converting the many 100MB WARC/ARC files into larger 1GB files, format conversion of ARC files to the modern WARC format, updating legacy WARC versions to modern versions of the specification, or including additional metadata records inside existing content files. When additional metadata is desired for these files, it will be generated and stored in a separate file being referred to as a "metadata sidecar file" [12]. This decision allows for the provenance of the files to be maintained for archival purposes, but does require additional attention to be paid to the early crawls when building tools because of mixed use of the ARC/WARC format and early versions of the WARC standard that are present in the EOT-2008 crawls.

5 CREATING THE DERIVATIVES

In order to provide datasets that can be used in a wide variety of applications, there was a need to create derivatives of the ARC/WARC files for different use-cases. The Common Crawl organizational structure mentioned above includes standard derivative formats in the web archiving community of WAT and WET files. For the EOT-2008 and EOT-2012 crawls, data was aggregated and processed at the UNT Libraries before uploading into the AWS S3 service.

[3] California Digital Library: https://cdlib.org/
[4] Internet Archive: https://archive.org
[5] Library of Congress: https://loc.gov
[6] UNT Libraries: https://library.unt.edu/
[7] Open Wayback: https://github.com/iipc/openwayback
[8] Webrecorder pywb: https://github.com/webrecorder/pywb

For the EOT-2016 and EOT-2020 crawls, the EOT teams planed to upload the WARC data to AWS S3 and then create the derivative files afterward. Derivatives were generated using the Common Crawl branch of the ia-hadoop-tools [4] package and specifically the WEATGenerator functionality. No modifications were made to this tool before running it on the collections. The index files were generated by the CDX-indexer from the pywb project. We used the CDXJ format with flags to sort each output file and also include the full relative path to the WARC/ARC file based on the Common Crawl organizational structure. The ZipNum format was generated using scripts in the webarchive-indexing repository [7] again from Common Crawl. Finally, the cc-index-table repository [8] was used to generate the Parquet format from the ZipNum index [10]. All of these tools required no modification to the base code and only small configuration changes to make them work in our various processing environments.

6 UPLOADING THE DATASETS

Once datasets had been locally staged, verified, and derivatives created, the next step in the process was to load the data into the AWS S3 infrastructure. The AWS Open Data Sponsorship Program provides access to a storage bucket in this case called "eotarchive" where the original and derived data of various crawls would be uploaded. The EOT team made use of the AWS Command Line Interface [2] to load data into the service. At UNT this was generally accomplished one segment at a time after all derivatives were generated. At the Internet Archive, WARC files were loaded individually as a sequential process that was later run in parallel. These two approaches were necessary due to local organizational structures and infrastructure constraints for staging large datasets. While the project was only nominally interested in understanding the throughput of the different approaches, it can be noted that at UNT the upload speed of data to AWS was limited by local IO from disk, and at IA was generally limited by network bandwidth. The resulting datasets for EOT-2008 and EOT-2012 took about a month each to stage, create derivatives, and upload at UNT. EOT-2016 and EOT-2020, being much larger in size, are still ongoing from IA and are expected to take upwards of six months to complete the initial upload.

7 DOCUMENTATION AND PUBLIC ACCESS

The final step in the process includes the documentation of the datasets for researchers and also providing guides and examples on how to use these datasets to answer research questions. The two datasets that are completed are available at the End of Term website [9], where the others will be added in the future. Documentation of the formats as well as guides and examples for using these data formats will be based on the previous work of the Common Crawl project. It is expected that during the summer of 2022 this documentation will be generated by the EOT team and the final datasets will be registered with the AWS Open Data Sponsorship Program in their Open Data on AWS catalog. The team also has an interest in working with tools from the Archives Unleashed [10] program to document how these datasets can be used with their tools and services. For a complete listing of dataset sizes see Table 1.

Dataset	WARC #	WARC Size Compressed (TiB)
EOT-2008	125,704	16.85
EOT-2012	78,509	45.57
EOT-2016	TBD	159 + 150 (FTP)
EOT-2020	TBD	300

Table 1. Summary of End of Term Datasets on Amazon S3

8 CLOSING

This effort by the collaborative End of Term Web Archive to stage a copy of the four EOT crawls in the cloud has been helpful in understanding many of the challenges that organizations will face when thinking about staging content for large-scale computational use. The greatest challenge encountered during this project was accounting for the collections that had been stored in various repositories and infrastructures for over a decade. In many situations, those repository structures have changed in ways that are forgotten to the current EOT team and required investigation and the rebuilding of a knowledge-base of previous operations. The decision to base this work on the Common Crawl organizational structure and subsequently leverage existing tools and documentation was a major benefit to this project. If those tools had not been in place and previous examples were not available, the whole process would have been more challenging and required greater allocations of time and resources. The EOT team is excited to make these datasets available more broadly to researchers who are interested in using the End of Term web archives in their research and scholarship.

[9] End of Term Web Archive Datasets: https://eotarchive.org/data/
[10] Archives Unleashed: https://archivesunleashed.org/

REFERENCES

[1] Amazon. 2022. *Amazon S3.* Retrieved May 1, 2022 from https://aws.amazon.com/s3/
[2] Amazon. 2022. *AWS Command Line Interface.* Retrieved May 1, 2022 from https://aws.amazon.com/cli/
[3] Amazon. 2022. *Open Data Sponsorship Program.* Retrieved May 1, 2022 from https://aws.amazon.com/opendata/open-data-sponsorship-program/
[4] Internet Archive. 2020. *ia-hadoop-tools.* Retrieved May 1, 2022 from https://github.com/commoncrawl/ia-hadoop-tools
[5] Jefferson Bailey. 2016. *WAT Overview and Technical Details.* Retrieved May 1, 2022 from https://webarchive.jira.com/wiki/spaces/ARS/pages/90997503/WAT+Overview+and+Technical+Details
[6] International Internet Preservation Consortium. 2016. *OpenWayback CDXJ File Format 1.0.* Retrieved May 1, 2022 from https://iipc.github.io/warc-specifications/specifications/cdx-format/openwayback-cdxj/
[7] Common Crawl. 2019. *WebArchive URL Indexing.* Retrieved May 1, 2022 from https://github.com/commoncrawl/webarchive-indexing
[8] Common Crawl. 2022. *Common Crawl Index Table.* Retrieved May 1, 2022 from https://github.com/commoncrawl/cc-index-table
[9] Common Crawl. n.d.. *Common Crawl.* Retrieved May 1, 2022 from https://commoncrawl.org/
[10] Sebastian Nagel. 2019. Accessing WARC files via SQL. In *2019 International Internet Preservation Coalition General Assembly and Web Archiving Conference* (Zegreb, Croatia). https://digital.library.unt.edu/ark:/67531/metadc1608961/
[11] University of North Texas Libraries. 2022. *Coda.* Retrieved May 1, 2022 from https://github.com/unt-libraries/coda
[12] University of North Texas Libraries. 2022. *warc-metadata-sidecar.* Retrieved May 1, 2022 from https://github.com/unt-libraries/warc-metadata-sidecar
[13] Mark E. Phillips and Kristy K. Phillips. 2017. End of Term 2016 Presidential Web Archive. *Against the Grain* 29, 6 (2017), 27–30. https://doi.org/10.7771/2380-176X.7874

[14] Tracy Seneca, Abbie Grotke, Cathy Nelson Hartman, and Kris Carpenter. 2012. It Takes a Village to Save the Web: The End of Term Web Archive. *Documents to the People* 40, 16 (2012), 16–23.