Robust Tools and Services for Long-Term Preservation of Digital Information

Joseph JaJa and Sangchul Song

Abstract

An unprecedented amount of digital information, appearing on a daily basis, needs to be archived and preserved over long time periods. Such information covers major facets of human activities such as business exchanges and electronic commerce, cultural and social interactions, e-government and legal proceedings, scientific studies and data collections, and even personal data such as digital photos and videos. It has been widely recognized that digital preservation is in general a very challenging process that requires innovations in institutional and business models, technology infrastructure, and social and legal frameworks.

In this paper, we will report on some of the core archiving and preservation tools and services that we developed under a general technology framework called ADAPT—Approach to Digital Archiving and Preservation Technology. The ADAPT model is based on a layered, digital object architecture that includes a set of modular tools and services built using open standards and Web technologies. These tools are designed so that they can easily accommodate new standards and policies while gracefully adapting to the underlying technologies as they evolve. In particular, we will briefly describe our tools to (1) proactively audit and ensure data integrity over the lifetime of an archived digital object, (2) enable compact storage and fast access to large scale Web archives, and (3) manage ingestion workflows under a wide variety of environments. Most of these tools are currently being used to support the digital preservation environment of NDIIPP institution contents through the Chronopolis project.

LIBRARY TRENDS, Vol. 57, No. 3, Winter 2009 ("The Library of Congress National Digital Information Infrastructure and Preservation Program," edited by Patricia Cruse and Beth Sandore), pp. 580–594

Introduction

A large portion of the scientific, business, cultural, and government digital information being created today needs to be maintained and preserved for future use of periods ranging from a few years to decades and sometimes centuries. Since the mid-nineties, the issue of long-term preservation of digital information has received considerable attention by major archiving communities, library organizations, government agencies, scientific communities, and individual researchers. These studies have identified major challenges regarding institutional and business models, technology infrastructure, and social and legal frameworks, which need to be addressed to achieve long-term reliable archiving of and access to digital information. Selected references that cover some of these findings are (Hedstrom, 2002; Hedstrom et al., 2003; Thibodeau, 2002). Focusing on the technology component, we note that a significant number of initiatives have been set up to develop technology prototypes to tackle some aspects of this problem. These initiatives include the Internet Archive (Kahle, 1997), the National Library of Australia's PANDORA project (n.d.), LOCKSS (Maniatis et al., 2005), the TPAP—Transcontinental Persistent Archive Prototype (Moore et al., 2003), the Universal Virtual Computer (Lorie, 2002), the Electronic Records Archives program at the National Archives (National Archives and Records Administration, n.d.), and the Library of Congress National Digital Information Infrastructure and Preservation Program (NDIIPP) (The National Digital Information Infrastructure and Preservation Program, the Library of Congress).

The traditional archiving and preservation approach has been a distributed activity in which each organization maintains and preserves its holdings with relatively little sharing. Such an approach is based on well-understood and proven processes for archiving and preserving physical holdings, which have been refined over the years. On the other hand, digital preservation is a very recent activity that is faced with a major technology challenge due in part to the large amount of important digital information generated on a daily basis, the fast pace of technology evolution, and the relative fragility of digital information and computing infrastructure. As a result, it appears that systematic methodologies are needed to address the following key requirements:

- Encapsulation of information regarding content, structure, context, provenance, and access within each digital object to enable the long-term maintenance and lifecycle management of the digital object.
- Efficient management of technology evolution, both hardware and software, and the appropriate handling of technology obsolescence (for example, format obsolescence).
- Efficient risk management and disaster recovery mechanisms either from technology degradation and failure, or natural disasters such as

fires, floods, and hurricanes, or human-induced operational errors, or security failures and breaches.
- Efficient proactive mechanisms to ensure the authenticity and integrity of content, context, and structure of archived information throughout the preservation period.
- Ability for information discovery and content access and presentation, with an automatic enforcement of authorization and IP rights, throughout the life cycle of each object.
- Scalability in terms of ingestion rate, capacity, and processing power to manage and preserve large scale heterogeneous collections of complex objects, and the speed at which users can discover and retrieve information.
- Ability to accommodate possible changes over time in organizational structures and stewardships, relocation, and repurposing.

The reports (Hedstrom, 2002; Hedstrom et al., 2003), while relatively old, give a good summary of the main technology challenges facing long-term digital preservation and archiving.

In this paper, we present an overview of a number of our tools that were designed to address several of the requirements listed above and that are currently in use by the Chronopolis preservation environment. The Chronopolis project, a National Digital Information Infrastructure and Preservation Program (NDIIPP) supported effort, offers a distributed data grid architecture with storage located at the University of Maryland, San Diego Supercomputer Center (SDSC), and the National Center for Atmospheric Research (NCAR). The main goal of Chronopolis is to provide long-term archiving and preservation services on contents coming from NDIIPP partners. Initial contents have been provided by the California Digital Library (CDL) and the Inter-University Consortium for Political and Social Science (ICPSR).

The ADAPT Approach

Long-term preservation of digital information is a process that must begin *before* the data is ingested into an archival system and must remain continuously active throughout the life cycle management of the digital objects. In fact, an understanding of exactly what is being preserved and how to precisely incorporate such information is a critical step that must be completed before any ingestion can begin. While the traditional archiving processes of appraisal, accessioning, arrangement, description, preservation, access, and repurposing are well understood for archiving and preserving physical holdings, they are quite lacking in addressing digital preservation.

Our technology approach is based on a number of premises. The first premise is to capture properties of content, structure, context, presenta-

tion, and preservation within a digital object architecture, and enable the infrastructure to manage and preserve these objects. The digital object must contain the essential features that encapsulate what is being preserved, and should include behavioral information about its life cycle management and preservation. An early work to advocate a digital object architecture appears in (Kahn and Wilensky, 1995), which led to the development of the Handle system (Corporation for National Research Initiatives, n.d.) for assigning persistent global identifiers.

The second premise of our approach is to separate the archive's management of the digital objects into three levels of abstraction, resulting in a well-defined three-layered architecture. The data layer is responsible for managing the bits representing the digital object across storage systems (evolving through both time and space), while the second layer deals with the semantics of the data and relationships between objects rather than storage and bits. The third layer deals with services related to monitoring, preservation, and management policies.

Finally, we borrow considerably from the Open Archival Information System (OAIS) reference framework *Reference Model for an Open Archival Information System (OAIS)* (Consultative Committee for Space Data Systems, 2002), including overall terminology. Briefly, this model consists of producers, an archive, and consumers, where the producers prepare and transfer data to the archive, which is responsible for managing the digital information for long-term preservation and for providing an interface to the consumers for accessing the information as needed. For each stage, OAIS provides a detailed model of the information, called respectively the Submission Information Package (SIP), the Archival Information Package (AIP), and the Dissemination Information Package (DIP).

The overall ADAPT model can be represented as shown in figure 1. Our efforts are aimed toward the development of tools and services in support of the components represented by the shaded boxes.

Thus far, our team has developed tools and services to handle the ingestion workflow and some aspects of the preservation, search, and access services. These tools are independent of the architecture of the data or the metadata layer, and will work with either centralized or distributed infrastructure. Our only assumptions regarding these two layers are that (1) each digital object has a unique persistent name; and (2) the data layer maintains more than a single copy of each digital copy, one of which is designated as the master copy. Otherwise our tools are completely platform-independent and will easily interoperate with any archive using the appropriate APIs.

In the remainder of this section, we briefly outline our ingestion workflow environment, called the Producer–Archive Workflow Network (PAWN), which was developed under the Transcontinental Persistent Ar-

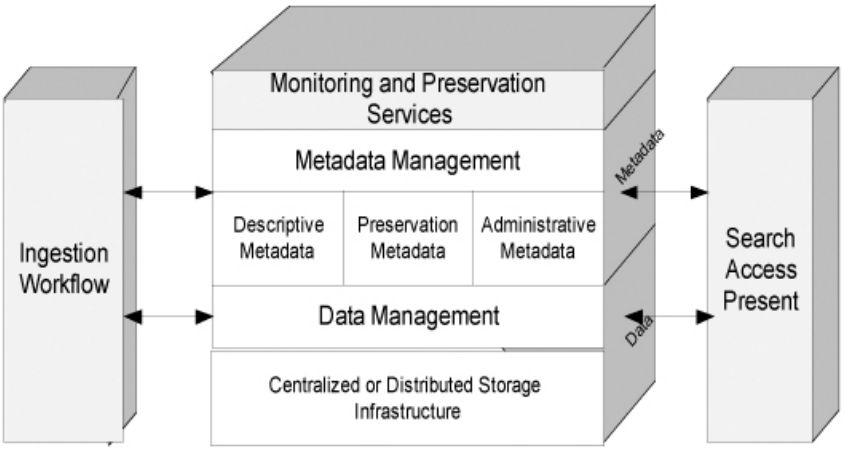

Figure 1. Overall ADAPT Model

chive Prototype (TPAP) funded by the National Archives, while the rest of this paper will be devoted to the work done under our NDIIPP project.

Producer–Archive Workflow Network (PAWN)

The ingestion process of digital objects into a long-term archive constitutes a critical phase during which the object's content, metadata, context, and provenance have to be assembled correctly. This task becomes complicated when there are many independent producers involved, each with a possibly different arrangement with the archive. PAWN is a novel software environment that provides a flexible and scalable platform for creating, packaging, and securely processing digital information into a remote archive while allowing flexible interactions between the producers and the archive. PAWN is policy-driven with built-in core functions and policies that can be customized to address arbitrary ingestion requirements. The environment is platform-independent and is designed to operate across multiple administrative domains using strong security mechanisms. In fact, PAWN provides a common infrastructure for both producers and the archive to manage and monitor the overall ingestion process. PAWN has been tested and evaluated by groups at the National Archives and the Library of Congress. More details about PAWN can be found in (Smorul, McGann and JaJa, 2007).

Ensuring Data Integrity through ACE

A critical component of the Chronopolis project is to ensure the authenticity and integrity of the NDIIPP collections managed by the Chronopolis environment. In this section, we introduce our ACE (Auditing Control

Environment) tool that accomplishes this goal. ACE is based on a rigorous cryptographic approach, which is, at the same time, scalable and cost effective. The first release of ACE is publically available through the ADAPT site (ACE Software, 2008).

Background

Digital information is in general quite fragile because of the many ways errors can be introduced. These include hardware and media degradation, hardware and software malfunction, operational errors, security breaches, and malicious alterations. Permanent loss of data can occur due to major hardware and software upgrades, and the possibility of natural hazards and disasters such as fires, floods, and hurricanes. Note that most of the archive's holdings may be accessed infrequently, and hence several cycles of technology evolution may occur in between accesses to most digital objects thereby causing corrupted files to go undetected until it is too late. Two additional factors complicate this problem further. First, an object will typically be subjected to a number of transformations during its lifetime, including those migrative transformations due to format obsolescence. These transformations may alter the object in unintended ways. Second, all current integrity checking mechanisms are based on some type of cryptographic techniques, most of which are likely to become less immune to potential attacks over time, and hence they will need to be replaced with stronger techniques when this occurs. Therefore an approach to ensure the integrity of a long-term archive has to also be able to address these two factors.

Current Methodologies

The simplest technique for implementing integrity checks is to use some form of replication such as mirroring. The integrity verification can then be made by comparing the replicas against each other. This method can easily detect a change in the stored data only if the modification is not carried out in all the replicas and no errors are introduced during data movement. This is not a practical solution due to the fact that it introduces a substantial computational overhead and it is not scalable. A more effective technique used in RAID storage (Patterson, Gibson, and Katz, 1988) is based on coding techniques, ranging from parity checking to more advanced types of erasure codes. These techniques involve expanding the data using some types of algebraic operations in such a way that some errors can be detected and corrected without the need of any replication. Coding techniques alone are clearly not acceptable for long-term archives because they can only detect certain types of errors but not all. Another widely used method is based on cryptographic hashing (also called *checksum*) techniques. In this approach, a checksum of the bitstream is computed and is stored persistently either with the data or separately. In general a cryptographic hash algorithm takes an input of arbitrary length

and converts it into a single fixed-size value known as a *digest* or *hash value*. A critical property of cryptographic hash algorithms is that they are based on one-way functions, that is, given the hash value of a bitstream A, it is computationally infeasible to find a different bitstream B that has the same hash value. Assuming that the hash values can be maintained correctly, data integrity can be verified by comparing the stored hash value with a newly computed hash from the data. Although no known hash function has been proven to be truly one way, the most common hash functions in use are MD5, SHA-1, SHA-256, and RIPEMD-160, all of which seem to work well in practice, in spite of the recent attacks that illustrated how to break MD5 and SHA-1 (Wang, Yin, and Yu, 2005; Wang and Yu, 2005). In addition to the one-way assumption, a key assumption of this technique is that the hash values can be stored securely with absolutely no changes introduced to these values over time. But this means we are back where we started in the sense that we now have to solve more or less the same problem—ensuring the integrity of the hash values whose number is the same as the number of objects—a slightly simpler problem but still of similar difficulty and scale as the initial problem.

More elaborate techniques have been suggested for handling the *integrity of bitstreams* in a digital archive. We mention here two such techniques. The first technique, used by LOCKSS (Maniatis et al., 2005), is based on a combination of replication and hashing. In this approach each digital object is already available over a number of caches through the peer-to-peer architecture provided by LOCKSS. Integrity checking is performed by computing the hash of each copy locally, and sending all the hashes to an auditor. A majority vote enables the auditor to discover faulty copies, if any. In general, there are a number of limitations with this approach, the most important of which is the assumption that there are many replicas available for each object. While this assumption may be reasonable for archiving electronic journals, it is clearly not a reasonable assumption for a general archive. Also, the process is expensive and requires a significant communication overhead. Note that LOCKSS nodes can arbitrarily initiate auditing requests thereby tying up distributed resources in an unpredictable way.

The second elaborate approach consists of using digital signatures based on public key cryptography, which depends on the existence of a public key infrastructure. The British Library (Kelly, 2006) uses a time stamping authority (TSA) that attaches a time designation to the object (or its hash) and signs it using the private key of the TSA. The verification process depends completely on the trustworthiness of a single entity, namely the TSA, which is not a reasonable assumption for long-term archives. Also, should the private key be compromised, the whole archive becomes at risk. There are other limitations with this solution including the fact that each public key has a fixed time span and one needs to track the history of these public keys.

Finally, we note that each object in a long-term archive may undergo several transformations during its lifetime (due for example to format obsolescence). To ensure data authenticity, we require a verifiable and an auditable record of every version of a digital object, which links the current version to the original version of the object. None of the schemes mentioned above can address this particular requirement satisfactorily.

The Basic ACE (Auditing Control Environment) Approach

The starting point of the ACE approach (Song and JaJa, 2007) is *cryptographic hashing*. To address the problem of maintaining and ensuring the correctness of a very large number of hash values, we compress the hash values generated during a time period into a single hash value that depends in a cryptographic sense on all the values and their temporal ordering. This can be done through a technique called *linked hash*ing (Haber and Stornetta, 1991; Merkle, 1980). The corresponding single hash value will be indexed by a corresponding time stamp, and its size will be fixed, independent of the number of hash values processed. The same process can be iterated to reduce the number of values whose integrity have to be absolutely maintained to less than a few hundred a year, regardless of the number of objects processed! We note that similar ideas were independently explored in (Haber and Kamat, 2006).

More specifically, our approach is based on the following two ideas. The first is to use a third-party server, independent of the archive, to aggregate the hashes of the set of objects processed by the archive during a certain time frame based on a linked hashing scheme, so as to create a very compact signature or fingerprint for the set. With extremely high probability, any change to any of the objects or to their relative temporal ordering will cause the corresponding signature to change. The signatures will be retained by the independent third party to be used to audit the archive's hashes as necessary. We also have mechanisms to audit the third party either by the archive or by an independent auditor. The second key idea of our approach is to use auditors that continuously monitor the integrity of each object according to a policy set locally by the archive. Any error that occurs between two consecutive auditing rounds will be detected with extremely high probability. We next explain how this approach was implemented through the development of our ACE software.

The computation of the ACE integrity information consists of three tiers. The first tier deals with creating a small size integrity token (IT) for each digital object upon a request from the archive, to be stored either with the object itself or in a registry at the archive as authenticity metadata. Each integrity token is of approximate size 1 KB and captures the integrity of the object as well as its temporal position within the time frame during which it was processed. *Cryptographic summary information* (CSI), a form of linked hashing, which depends on all the objects regis-

tered during a dynamically adjustable time interval is computed, stored, and managed independently of the archive. The number and size of the CSIs are independent of the number of the objects in the archive and depends only on the number of time frames. In general, their corresponding overall size is in the order of hundreds of megabytes a year. These are generated and maintained by the independent third party. The third tier involves the generation of very compact witness values that cryptographically depend on all the CSIs covering a relatively long time period (such as a day or a week). The total size of the witness values is in the order of a few kilobytes a year, and hence they can easily be preserved on CD-ROMs (with frequent refreshing into newer media as necessary), as well as published on the Internet on widely accessible sites. As long as the witness values are kept correct, any change to a digital object will be detected with extremely high probability through the auditing process to be briefly described later.

ACE Prototype

The ACE prototype includes two major components: ACE Integrity Management System (ACE-IMS) and ACE Audit Manager (ACE-AM). The ACE-IMS, set up as an entity separate from the archive, is a server that issues integrity tokens upon the requests of an archive, preserves the CSI values, and computes and publishes the witness values. The ACE-AM is a bridging component between the archive and the ACE-IMS, which is local to the archive. In a distributed setting, the audit managers work asynchronously independent of each other, and hence copies of the same object will be audited independently of each other.

The ACE-IMS, operating separately from the archive, provides two important services: the issuing of integrity tokens and CSI verification. The former service generates an integrity token upon a request from the archive. Using the digital object and the integrity token, the archive can at anytime construct the cryptographic summary (CSI) corresponding to the round in which the digital object was registered. In our prototype, we use a separate database at the archive to hold the integrity tokens. The CSI values will be maintained separately and independently by the ACE-IMS.

The ACE Audit Manager (ACE-AM), which is local to the archive, audits each digital object on a regular basis as set by the archive manager or upon request by a user. It also serves as the main interface with the ACE-IMS. In particular, the ACE-AM retrieves the integrity token of the object to be audited, computes the hash of the object, and compares it to the hash value stored in the integrity token. To verify the correctness of the integrity token, the ACE-AM computes the Cryptographic Summary Information using only information contained in the integrity token, and then requests the corresponding CSI from the ACE-IMS. An agreement indicates that no errors have been introduced to the integrity token.

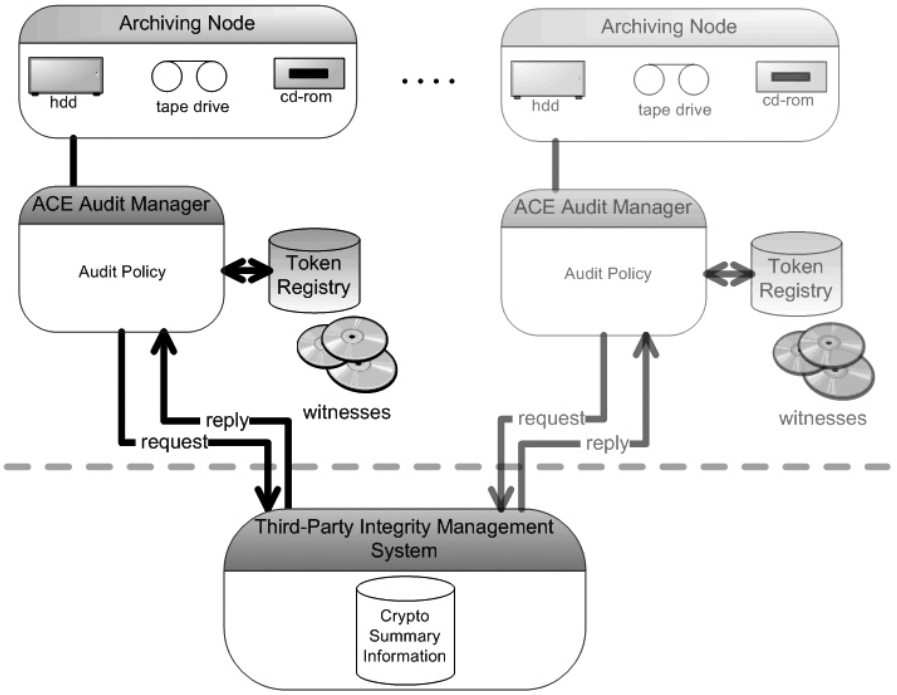

Figure 2. ACE System Architecture

Figure 2 shows the overall ACE architecture assuming a distributed archiving infrastructure. A centralized archiving infrastructure will reduce to a single archiving node. The upper section represents the archive, the middle section contains the ACE-AM that is local to each archiving node, and the lower section represents the ACE-IMS, which is supposed to be outside the archive's domain and can support many different archives.

The ACE software is currently running on the Chronopolis environment to regularly audit hundreds of thousands of files, of widely varying sizes. This software has been used to manage millions of files quite effectively performing at the speed at which data can be transferred to the audit manager. More details can be found in ACE Software (2008).

Web Archiving

A more recent effort by our group has focused on storage and access technologies in support of Web archiving, which has been a major thrust area for NDIIPP.

Background

An unprecedented amount of information encompassing almost every facet of human activity across the world is currently available on the Web and is growing at an extremely fast pace. In many cases, the Web is the only medium where such information is recorded. However the Web is an ephemeral medium whose contents are constantly changing and new information is rapidly replacing old information, resulting in the disappearance of a large number of Web pages every day and a permanent loss of part of our cultural and scientific heritage on a regular basis. The Internet Archive (Kahle, 1997), the world's largest Web archive, has been the leader in developing methodologies and standards for archiving Web contents. Its main goal is to capture significant snapshots over time of the whole Web and to archive the corresponding contents. It currently holds around two petabytes of data and is growing at the rate of twenty terabytes per month (Internet Archive, n.d.). Its overall strategy consists of regular crawls of the Web, followed by downloading the crawled contents into "containers" after each crawl and indexing the archived contents to allow future access based on URLs and crawl times.

Leaving aside the major business, social, and legal issues that must be addressed in dealing with Web archiving, our work has focused on the technology challenges facing the archiving of selected Web contents (such as domain- or topic-specific) rather than capturing a snapshot of the overall Web. Web contents present unique challenges well beyond those encountered in archiving and preserving well-defined and static digital objects. In particular, a Web object is usually not well-delineated because of all the links that it contains, which makes it hard to determine its boundaries. Add to that the fact that many Web objects contain highly dynamic contents that change at unpredictable rates, and the fact that a large fraction of the Web contents reside in the deep or hidden Web and hence cannot be extracted through the typical Web crawls. We list here three major categories of technical problems that need to be addressed:

- *Crawling Strategies.* Archiving domain- or topic-specific Web contents requires the development of efficient crawling strategies that can locate the pertinent contents and extract and assemble the appropriate metadata. A difficult problem is to be able to identify and characterize the topic-specific contents that are archive worthy (such as restricting the crawling to a set of "certified" sites) and to constrain the search so as not to crawl many irrelevant sites. Given the dynamic nature of the Web, it is also critical to perform the crawling process either frequently enough so as to capture the contents before they are updated, or dynamically so that any change can be almost immediately detected and captured.
- *Storage and Indexing.* Web objects have widely different sizes, most of which are very small, and more importantly have links embedded in

them referring to other Web objects. A major issue here is how to store Web objects compactly and index them so as to maintain the linked information between the archived objects, which existed at the time the objects were archived. Given that a significant fraction of the Web contents may be the same between two consecutive crawls, we need techniques that will avoid storing duplicate contents while maintaining a complete history of the archived Web objects.

- *Search and Access.* Traditional access mechanisms to archives and libraries rely on a combination of descriptive metadata and contextual information to locate and access information. On the other hand, the extremely successful Web search engines rely on a combination of text search techniques from information retrieval and strategies to rank the large number of Web pages containing the query terms. A challenging research problem is to determine the best strategy to search and access archived Web contents in a cost-effective manner. For large archives, Web search strategies may be computationally too expensive to implement in practice, and moreover new strategies to rank the archived Web pages seem to be required to generate the most relevant contents.

In our recent work reported in (Song and JaJa, 2008a, 2008b), we have developed provably good techniques for some of the problems arising in storage and indexing of Web archives. These techniques have been tested and evaluated on significant-size Web archives. To explain some of our work, we need to briefly introduce some of the currently used methods for organizing and storing Web objects.

We start with the straightforward method of using a local file system to store the Web material by copying each object into the local file system. In this case, within an HTML object, the URI scheme "file" can replace the scheme "http" in the original object. For example, http://www.librarytrends.org/index.html will be rewritten as "file:///archive/2008.07.01/www.librarytrends.org/index.html." The locally stored objects can be republished through a Web server for public access. This method is used by the National Library of Australia's Pandora project (n.d.). However, the most popular method currently in use stores Web objects in containers using a well-defined structure. A Web container holds a set of harvested Web files, each with its own auxiliary metadata, such that the size of a container is typically in the order of hundreds of megabytes. Also an external index is maintained to provide the mapping between hyperlinks inside a container and the locations of the archived objects that the hyperlinks point to. One of the most widely used container formats is the ARC file format (WWW Archive File Format Specification, 1996) that was originally developed by the Internet Archive and adopted by many others. Recently, building on the ARC format, an international collaborative effort developed a standard format called the WARC file format (WARC, Web ARChive file format, n.d.).

In (Song and JaJa, 2008b) we developed a scheme to archive a linked set of Web objects into containers in such a way to enable users to interactively conduct a browsing session over the archived Web contents by navigating through the links in the archived objects as in the case of a typical browsing session over the Internet. In more technical terms, we minimized the number of containers that hold the Web pages to be accessed during a typical browsing session. In essence, the scheme consists of analyzing the Web graph to rank each link in order to determine an estimate of the probability that a user is likely to choose this link, and using this rank information we package Web objects into containers based on graph partitioning techniques. We have tested our scheme on significant Web archives, the results of which have shown that our scheme allows for faster browsing and access compared with conventional container packaging techniques.

In (Song and JaJa, 2008a) we have considered the problem of storing and indexing Web contents using the crawling strategy but avoiding the storage of any duplicate Web contents examined between consecutive crawls. We have developed a new scheme that stores unique temporal Web contents in the ARC/WARC format, and that provides a quick access to the archived contents for arbitrary temporal queries, that is, for any query that requests an archived page given its URL and/or its crawling date. Our scheme can be shown to be theoretically optimal both in storage utilization and retrieval time. Using two very different data sets from the Stanford WebBase project, one reflecting slow changing Web sites while the other set reflects quickly changing Web sites, we have conducted extensive experiments using our storage and indexing schemes. The experimental results confirm the substantial storage savings achieved by eliminating duplicate contents between consecutive crawls, as well as illustrate the scalable performance of our method in finding the archived contents specified through arbitrary temporal queries.

Conclusion

In this article, we have illustrated some of the modular tools and technologies developed under the ADAPT approach. These tools have addressed needs in ingestion workflow, preservation services, and search and access of archived contents. Our tools are based on Web technologies and open standards and protocols, and make no assumption about the particular architecture of the archive's storage infrastructure. These tools have been tested and evaluated in realistic environments illustrating their flexibility, scalability, and reliability.

In particular, most of our tools are currently in use in the NDIIPP Chronopolis preservation environment, whose main goal is to ingest, manage, and preserve collections from NDIIPP partners. Substantial contents from the California Digital Library (CDL) Web-at-Risk and the Inter-University

Consortium for Political and Social Research (ICPSR) Data-PASS have already been transferred to the Chronopolis environment. The ADAPT tools—ACE, PAWN, and the Replication Monitoring Service—are playing a critical role in the support and maintenance of this environment. We will soon be reporting more details on the performance of these tools and our plans for the next release of the software.

References

ACE Software (Version 1). (2008). Retrieved January 15, 2009, from http://adapt.umiacs.umd.edu/ace

Consultative Committee for Space Data Systems. (2002, January). *Reference Model for an Open Archival Information System (OAIS)*, CCSDS 650.0-B-1, BLUE BOOK, Issue 1 [Equivalent to ISO 14721:2002]. Retrieved January 15, 2009, from http://public.ccsds.org/publications/archive/650x0b1.pdf

Corporation for National Research Initiatives. (n.d.). Handle system: A general-purpose global name service enabling secure name resolution over the Internet, Retrieved June 9, 2008, from http://www.handle.net/

Haber, S., & Kamat, P. (2006, May 23–26). Content integrity service for long-term digital archives. (Paper presented at the Proceedings of Archiving 2006, Ottawa, Canada.)

Haber, S., & Stornetta, W. S. (1991). How to time-stamp a digital document. *Journal of Cryptology, 3*(2), 99-111.

Hedstrom, M. (2002). *It's about time: Research challenges in digital archiving and long-term preservation.* (Final Report of a Workshop on Research Challenges in Digital Archiving and Long-term Preservation, April 12–13, 2002). Washington, DC: National Science Foundation and Library of Congress. Retrieved January 15, 2009, from http://www.digitalpreservation.gov/library/resources/pubs/docs/about_time2003.pdf

Hedstrom, M., Ross, S., Ashley, K., Christensen-Dalsgaard, B., Duff, W., Gladney, H., Huc, C., Kenney, A.R., Moore, R., Neuhold, E. (2003). Invest to save: Report and recommendations of the NSF-DELOS working group on digital archiving and preservation. Retrieved January 15, 2009, from http://eprints.erpanet.org/48/01/Digitalarchiving.pdf

The Internet Archive—Frequently asked question—How large is the Wayback Machine? (n.d.). Retrieved June 10, 2008, from http://www.archive.org/about/faqs.php#9

Kahle, B. (1997, March). Preserving the Internet. *Scientific American, 276*(3), 72–73.

Kahn, R., & Wilensky, R. (1995). A framework for distributed digital object services: National research initiatives. Retrieved January 15, 2009, from http://www.cnri.reston.va.us/k-w.html. Archived at http://www.webcitation.org/5YeGpQ0IP

Kelly, L. (2006, April 25). British Library Secures Integrity of Digital Archive. *Computing*. Retrieved January 15, 2009, from http://www.computing.co.uk/computing/news/2154704/british-li. Archived at http://www.webcitation.org/5eCCT4PwA

Lorie, R. (2002). The UVC: A method for preserving digital documents—Proof of concept KB/IBM Long-Term Preservation Study No. 4. Retrieved January 15, 2009, from http://www.kb.nl/hrd/dd/dd_onderzoek/reports/4-uvc.pdf

Maniatis, P., Roussopoulos, M., Giuli, T. J., Rosenthal, D. S. H., & Baker, M. (2005). The LOCKSS peer-to-peer digital preservation system. *ACM Trans. Comput. Syst., 23*(1), 2–50.

Moore, R., Marciano, R., JaJa, J., Wilensky, R., & Deken, J. (2003). NARA persistent archives: NPACI collaboration project. SDSC Technical Report No. TR-2003-2. San Diego Supercomputer Center.

Merkle, R. C. (1980). Protocols for public key cryptosystems. (Paper presented at the IEEE Symposium on Security and Privacy.)

National Archives and Records Administration. (n.d.). The Electronic Records Archive (ERA). Retrieved June 9, 2008, from http://www.archives.gov/era

The National Digital Information Infrastructure and Preservation Program, the Library of Congress. Retrieved June 9, 2008, from http://www.digitalpreservation.gov/. Archived at http://www.webcitation.org/5YSZBCgxW

Pandora—Australia's Web Archive. (n.d.) Retrieved April 22, 2008, from http://pandora.nla.gov.au/. Archived at http://www.webcitation.org/5XHOp9Kso

Patterson, D. A., Gibson, G., & Katz, R. H. (1988). A case for redundant arrays of inexpensive disks (RAID). (Paper presented at the SIGMOD '88, Chicago, Illinois: Proceedings of the 1988 ACM SIGMOD international conference on management of data, New York, NY.)
Smorul, M., McGann, M., & JaJa, J. (2007, May 21–24). PAWN: A policy-driven environment for implementing producer-archive interactions in support of long term digital preservation. (Paper presented at the Archiving Conference 2007, Arlington, Virginia.)
Song, S., & JaJa, J. (2007, May 21–24). ACE: A novel software platform to ensure the integrity of long term archives. (Paper presented at the Archiving Conference 2007, Arlington, Virginia.)
Song, S., & JaJa, J. (2008a). Archiving temporal Web information: Organization of Web contents for fast access and compact storage (UMIACS Technical Report No. UMIACS-TR-2008-08): University of Maryland Institute for Advanced Computer Studies.
Song, S., & JaJa, J. (2008b, September 14–19). Fast browsing of archived Web contents. (Paper presented at the 8th International Web Archiving Workshop, Aarhus, Denmark.)
Thibodeau, K. (2002, April 24–25). Overview of technological approaches to digital preservation and challenges in coming years. (Paper presented at the The State of Digital Preservation: An International Perspective, Washington, DC).
Wang, X., Yin, Y. L., & Yu, H. (2005, August 14–18). Finding collisions in the full SHA-1. (Paper presented at the CRYPTO, Santa Barbara, California).
Wang, X., & Yu, H. (2005, May 22–26). How to break MD5 and other hash functions. (Paper presented at the EUROCRYPT, Aarhus, Denmark).
WARC, Web ARChive file format. (n.d.). Retrieved August 27, 2007, from http://www.digitalpreservation.gov/formats/fdd/fdd000236.shtml. Archived at http://www.webcitation.org/5RPhvw0Wa
WWW Archive File Format Specification. (1996). Retrieved April 22, 2008, from http://pages.alexa.com/company/arcformat.html. Archived at http://www.webcitation.org/5XHOLYGOG

Joseph JaJa currently holds the position of professor of Electrical and Computer Engineering with a joint appointment in the Institute for Advanced Computer Studies at the University of Maryland, College Park. JaJa received his PhD degree in Applied Mathematics from Harvard University and has since published extensively in a number of areas including parallel and distributed computing, combinatorial optimization, algebraic complexity, VLSI architectures, and data-intensive computing. His current research interests are in parallel algorithms, digital preservation, and scientific visualization of large scale data. JaJa has received numerous awards including the IEEE Fellow Award, the 1997 R&D Award for the development of software for tuning parallel programs, the ACM Fellow Award, and was a member of the team that won the 2006 Internet2 Driving Exemplary Applications for the TPAP project. He served on several editorial boards, and is currently serving as a subject area editor for the *Journal of Parallel and Distributed Computing* and as an editor for the *International Journal of Foundations of Computer Science*.

Sangchul Song is a PhD candidate in Electrical and Computer Engineering at the University of Maryland, College Park, MD. Before joining Maryland, he worked as a security software engineer for several years in San Jose, CA. He received his BE and MS degree at Korea University, Seoul, Korea. At Maryland, he has been actively involved in the long-term digital preservation group led by Professor Joseph JaJa.