ARTICLE

A 21st Century Technical Infrastructure for Digital Preservation

Nathan Tallman

ABSTRACT

Digital preservation systems and practices are rooted in research and development efforts from the late 1990s and early 2000s when the cultural heritage sector started to tackle these challenges in isolation. Since then, the commercial sector has sought to solve similar challenges, using different technical strategies such as software defined storage and function-as-a-service. While commercial sector solutions are not necessarily created with long-term preservation in mind, they are well aligned with the digital preservation use case. The cultural heritage sector can benefit from adapting these modern approaches to increase sustainability and leverage technological advancements widely in use across Fortune 500 companies.

INTRODUCTION

Most digital preservation systems and practices are rooted in research and development efforts from the late 1990s and early 2000s when the cultural heritage sector started to tackle these challenges in isolation. Since then, the commercial sector has sought to solve similar challenges, using different technical strategies. While commercial sector solutions are not necessarily created with long-term preservation in mind, they are well aligned with the digital preservation use case because of similar features. The cultural heritage sector can benefit from adapting these modern approaches to increase sustainability and leverage technological advancements widely in use across Fortune 500 companies.

In order to understand the benefits, this article will examine the principles of sustainability and how they apply to digital preservation. Typical preservation activities that use technology will be described, followed by how these activities occur in a 20th-century technical infrastructure model. After a discussion on advancements in the IT industry since the conceptualization of the 20th-century model, a theoretical 21st-century model is presented that attempts to show how the cultural heritage sector can employ industry advancements and the beneficial impact on sustainability.

Galleries, libraries, archives, and museums cannot afford to ignore the sustainability of managing and preserving digital content and neither can distributed digital preservation or commercial service providers.[1] Budgets lag behind economic inflation while the cost of and amount of materials to purchase rises, coupled with the need to hire more employees to do this work. If digital preservation programs are going to scale up to enterprise levels and operate in perpetuity, it is imperative to update technical approaches, adopt industry advancements, and embrace cloud technology.

Nathan Tallman (ntt7@psu.edu) is Digital Preservation Librarian, Pennsylvania State University. © 2021.

https://doi.org/10.6017/ital.v40i4.13355

SUSTAINABILITY

For digital preservation programs to succeed, they must be sustainable per the Triple Bottom Line or they risk subverting their mission. The Triple Bottom Line definition of sustainability identifies three pillars: people (labor), planet (environmental), and profit (economic).[2] While there are typically few people with digital preservation in their job title within an organization, it's a collaborative domain with roles and responsibilities distributed throughout organizations, reflecting the digital object lifecycle. It's important that the underlying technical infrastructure can easily be supported and is not so complicated that it is hard to recruit systems administration staff. Digital preservation consumes many technical resources and data centers have a substantial environmental impact. As Ben Goldman points out in "It's Not Easy Being Green(E)," data centers consume an immense amount of power and require extravagant cooling systems that use precious fresh water resources.[3] Because there is no point in preserving digital content if there will be no future generation of users, responsible digital preservation programs will seek to reduce carbon outputs and the number of rare-earth elements in our technical infrastructure.[4] While cultural heritage organizations rarely seek to make a profit, economic sustainability is vital to organizational health and costs for digital preservation must be controlled. Modern technological infrastructures discussed here will help to increase sustainability by using widespread technologies and strategies for which support can be easily obtained, by reducing energy consumption, by minimizing reliance on hardware using rare-earth elements, and by leveraging advances in infrastructure components such as storage to perform digital preservation activities.

BASIC DIGITAL PRESERVATION ACTIVITIES

This paper will examine technical preservation activities and the author acknowledges that basic digital preservation activities are likely to include risk management and other non-technical concepts. While there is no formal, agreed-upon definition of what constitutes a set of basic digital preservation activities, bit-level digital preservation is a common baseline. Bit-level digital preservation seeks to preserve the digital object as it was received, ensuring that you can get out an exact copy of what you put in, no matter how long ago the ingest occurred; however, with no guarantees as to the renderability of said digital object. Two basic digital preservation activities are key to this strategy: fixity and replication.

Fixity

Fixity checking, or the "practice of algorithmically reviewing digital content to ensure that it has not changed over time," is a foundational digital preservation strategy for verifying integrity that aligns with Rosenthal et al.'s "Audit" strategy.[5] Fixity is how preservationists demonstrate mathematically that the content has not changed since it was received. Not all fixity is the same, however; fixity can be broken up into three types: transactional fixity, authentication fixity, and fixity-at-rest.[6]

Transactional Fixity

Transactional fixity is checked after some sort of digital preservation event[7], such as ingest or replication. Depending on the event, it's desirable to use a non-cryptographic algorithm, such as CRC32 or MD5, when files move within a trusted system. When it's only necessary to prove that a file hasn't immediately changed, such as copying between filesystems, cryptographic algorithms are unnecessarily complex and are too expensive, in terms of compute consumption.

Authentication Fixity

Authentication fixity proves that a file hasn't changed over a long period of time, particularly since ingest. Although one could use a chain of transactional fixity checks to cumulatively prove there has been no change, it's often desirable to conduct one fixity check that can be independently verified. Unbroken cryptographic algorithms, such as one from the SHA-2 and SHA-3 families, are well suited to this use case and worth the complexity and compute expense, particularly since this type of fixity check doesn't have to be run as often.

Fixity-at-Rest

Fixity-at-rest is when fixity is monitored while content is stored on disk. While some organizations may choose to only conduct fixity checks when files move or migrate, this strategy can miss bit loss due to media degradation, software or human error, or malfeasance that is only discovered when the file is retrieved.[8] A common approach for monitoring fixity-at-rest is to systematically conduct fixity checks on all or a sample of files at regular intervals. These types of fixity checks may or may not use cryptographic algorithms, depending on their availability.[9]

Replication

Replication is another cornerstone of achieving bit-level digital preservation. The National Digital Stewardship Alliance's 2019 Levels of Digital Preservation, a popular community standard, recommends maintaining at least two copies in separate locations, while noting three copies in geographic locations with different disaster threats is stronger.[10] All of these copies must be compared to ensure fixity is maintained. An important concept to consider when thinking about replication is the independence of each copy. According to Schaefer et al.'s *User Guide for the Preservation Storage Criteria*, "The copies should exist independently of one another in order to mitigate the risk of having one event or incident which can destroy enough copies to cause loss of data."[11] In other words, a replica should not depend on another replica, but instead depend on the original file.

ADVANCED DIGITAL PRESERVATION ACTIVITIES

When considering more robust digital preservation strategies beyond bit-level preservation, additional activities must be considered to ensure that the information contained within digital files can be understood. Implementation of these activities may vary by digital object, depending on the digital preservation goal and appraised value of the content. This paper only describes a handful of the many advanced digital preservation activities as illustrative examples; the ideas in this paper could be applied to most advanced activities.

Metadata Extraction

Digital files often contain various types of embedded metadata that can be used to help describe both its intellectual content and technical characteristics. This metadata can be extracted and used to populate basic descriptive metadata fields, such as title or author. Extracted technical metadata is useful for broader preservation planning, but also for validating technical characteristics in derivative files. For example, if generating an access file for digitized motion picture film, it's necessary to know the color encoding, aspect ratio, and frame rate. If these details are ignored, the access derivative may appear significantly different than the original file and give a false impression to users.

File Format Conversions

File format conversions help to ensure the renderability of digital content. There are two types of file format conversions to consider: normalization and migration. Normalization generally refers to proactively converting file formats upon ingest to retain informational content, e.g., converting a WordPerfect document to plain text or PDF when only the informational content is desired. Migration may occur at any time: upon ingest, upon access, or any time while an object is in storage. Migration occurs when file formats are converted to a newer version of the same format, e.g., Microsoft Access 2003 (MDB) to Microsoft Access 2016 (ACCDB) or to a more stable and open format that retains features, e.g., Microsoft Access 2016 (ACCDB) to SQLite.

Versioning

Versioning, or the retention of past states of a digital object with the ability to restore previous states, is complex to implement and not always necessary. An organization might choose to apply versioning to subsets of digital content, such as within an institutional repository, but not for born-analog, or digitized material. Additionally, an organization may choose to version metadata only, ignoring changes to the bitstream, such as for born-analog digital objects.

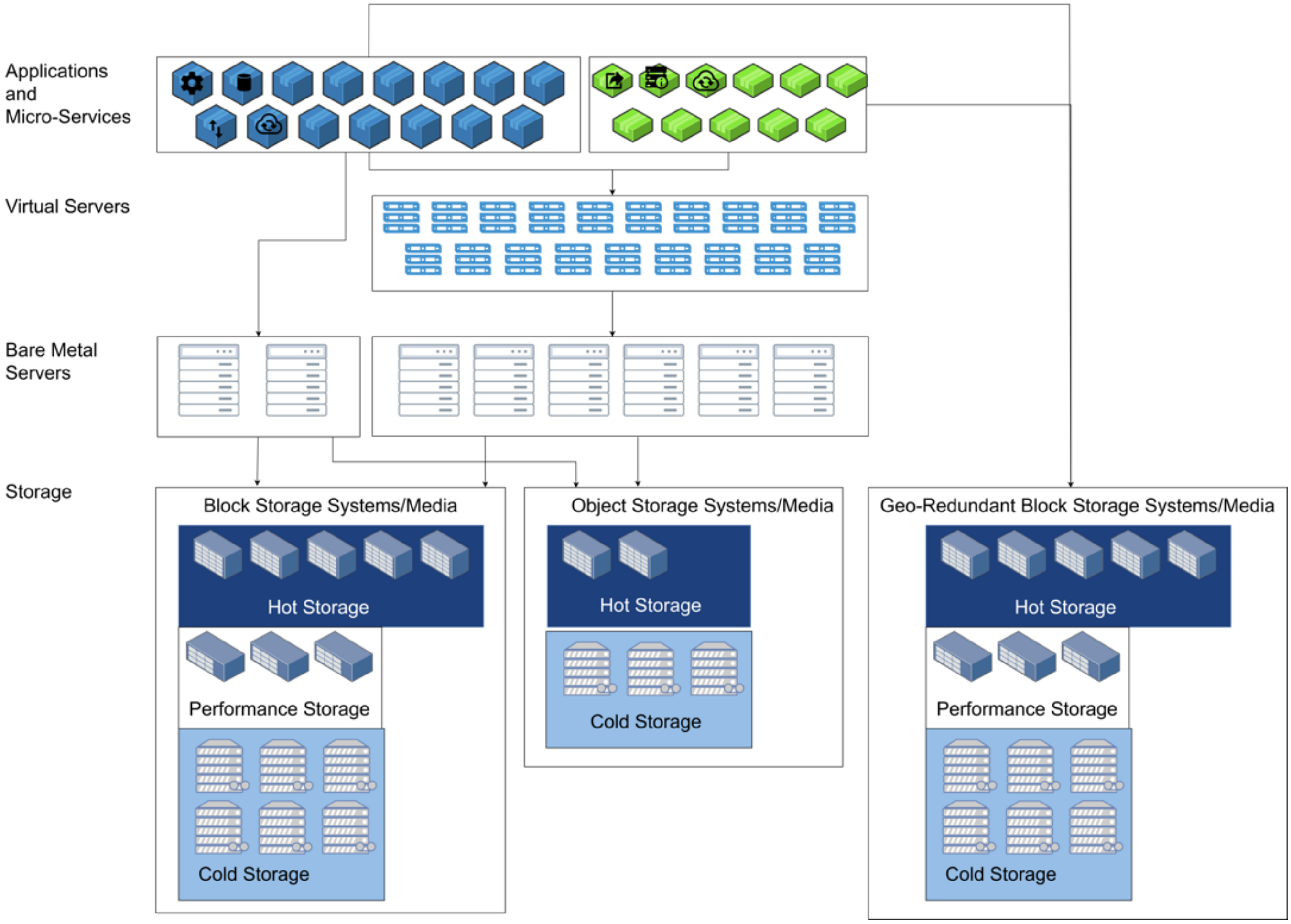

Figure 1. The infrastructure architecture for a typical 20th-century stack.

THE 20TH CENTURY TECHNICAL INFRASTRUCTURE

The technical infrastructure that enables digital preservation can come in many forms. While technology has advanced over the past thirty years, the cultural heritage sector, particularly where digital preservation is concerned, has been slow to adapt. Below are descriptions of three common components of a typical server stack (technical infrastructure), though the author acknowledges that some organizations have already moved past this model. Figure 1 is a diagram of the typical 20th-century stack.

Storage

Storage, at the core of digital preservation, has benefitted from rapid technological advancement since computers first started storing information on punch cards and magnetic media. Twentieth-century servers often use three main types of storage: file, block, and object.

File Storage

File storage is what most people are familiar with. A filesystem interfaces with the underlying storage technology (block or object) and physical media (hard disk drives, solid state drives, tape-based media, or optical media) to present users with a hierarchy of directories and subdirectories to store data. This data can easily be accessed by users or applications using file paths, while the filesystem negotiates the actual bit-locations on the physical media.

The choice of filesystem can impact data integrity (fixity), although choice may be limited by operating system. In the 20th century, journaling filesystems offered the most data protection as the filesystem keeps track of all changes; in the event of a disk failure, it's possible to recover more data if a journaling filesystem is used.

Block Storage

Block storage uses blocks of memory on physical media (disk, tape, etc.) that are managed through a filesystem to present volumes of storage to the server. All interactions between server and storage are handled by the filesystem via file paths, though the data is stored on scattered blocks on the media. Block storage directly attached to a server is often the most performant option, the data does not travel outside the server. Network attached storage, in which an external file system is mounted to the server as if it were locally attached block storage, requires data to travel through cables and networks before it gets to the server, which decreases performance.

Object Storage

Object storage, which still uses tape and disk media, is an abstraction on top of a filesystem. Instead of using a filesystem to interact directly with storage media, the storage media is managed by software. The software pools storage media and interactions happen through an API, with files being organized into "buckets" instead of using a filesystem with paths. Object storage is web-native and the basis for commercial cloud storage. Software-defined storage, which is discussed in more detail later in this article, also allows users to create block storage volumes that can be directly mounted to virtual servers as part of a filesystem or to create network shares that present the underlying storage to users via a filesystem.[12]

Both block and object storage can be used for high-performance storage, hot storage (online), cold storage (nearline), and offline storage. Generally, tape and slower performing hard disks are used for offline and nearline storage; faster performing hard disks are used for online storage. Solid-state drives (SSDs) using Non-Volatile Memory Express (NVMe) protocols are best suited for high-performance storage.[13]

In the 2019 Storage Infrastructure Survey, by the National Digital Stewardship Alliance, 60% of those aware of their organizational infrastructure reported a reliance on hardware-based filesystems (file and block storage) while about 15% used software-based filesystems (object storage), with 14% reporting a hybrid approach.[14] This indicates that the cultural heritage sector continues to rely more on file and block storage and is not yet fully embracing object storage. The survey did not probe into why this might be.

Servers: Physical and Virtual

Twentieth-century technical infrastructures relied primarily upon physical servers. Physical servers, also called bare metal, dominated the server landscape up through roughly 2005. Virtual servers arrived on the scene after "VMware introduced a new kind of virtualization technology which … [ran] on the x86 system" in 1999.[15] Server virtualization facilitated a fresh wave of innovation by making it easier and more inexpensive to create, manage, and destroy servers as necessary. Dedicating physical servers to one or a limited number of applications requires more resources and expends a higher carbon cost; virtual servers can be highly configured for their precise needs and this configuration can be changed using software, rather than changing parts on a physical server, resulting in less waste.

Cultural heritage organizations have been slow to fully adapt virtual servers. The 2019 NDSA Storage Infrastructure Survey reports that 81% of respondents continue to rely on physical servers with 63% of respondents using virtual servers. Fewer than 10% reported using containers, an even more efficient virtualization technology.[16] Containers are an evolution of virtual servers that act like highly optimized, self-contained servers doing a specific activity.[17]

Applications and Microservices

In the 20th century, applications often required dedicated servers. Business logic was handled by applications or microservices that ran on top of the server and storage, the highest level in the stack. There are advantages to handling the business logic at this high level: it's completely in the control of the developer and can be finely tuned to the needs. Unfortunately, this is also an expensive place to handle all business logic as the application needs to be maintained over time and there's overhead involved in working at this level of the stack. Microservices, in this server model, are generally specific commands that can be invoked as needed. While called microservices because they can be applied individually, they still run in this expensive part of the stack and have the same downsides as applications.

In digital preservation systems using this type of architecture, basic and advanced digital preservation activities occur within this application layer. Fixity can be a costly activity. Garnett, Winter, and Simpson, in their paper "Checksums on Modern Filesystems, or: On the Virtuous Consumption of CPU Cycles," point out that "calculating full checksums in software is not efficient" and "increases wear and tear on the disks themselves, actually accelerating degradation."[18] Fixity, when done this way, is a linear process that requires every file to be read from disk so a checksum can be calculated; when performing fixity over large amounts of content, this is very inefficient and time consuming.

PRESERVATION ACTIVITIES IN THE 20TH-CENTURY STACK

In this model of infrastructure, many cultural heritage institutions are relying on practices created when the field of digital preservation was emerging.

Basic Activities

Basic preservation activities take a generalized approach and mostly occur in the costly application and microservices layer. This follows the general approach of application development from the commercial sector in the 20th century.

Fixity

Although there are differences in frequency, most organizations do not currently make distinctions between transactional fixity, authentication fixity, or fixity-at-rest. Common current practices use the same method (MD5, SHA-256, SHA-512) for all fixity checks.[19] This inefficient approach take place in the application and microservices layer and uses more compute power than necessary, increasing the environmental impact.

Replication

In most 20th-century stacks, replications are handled in the application layer, where it is most costly in terms of computational power and labor to maintain, having a negative impact on sustainability. Some are using 20th-century microservices are well.

Advanced Digital Preservation Activities

Like basic preservation activities, advanced ones chiefly take place in the application and microservices layer if they occur at all.

Metadata Extraction and File Format Conversion

Metadata extraction and file format conversion tends to occur only upon ingest as a one-time event. Archivematica, the popular open-source digital preservation system, uses 20th-century microservices for each and they only occur during the transfer (ingest) process.[20] Other systems often include this in the business logic of the application layer.

Versioning

Version control is a feature that many organizations choose not to implement. The 2019 NDSA Storage Infrastructure surveys shows that fewer than half (40) of respondents (83) used any type of version control.[21] Version control is hard to implement in a custom system, with alternative approaches. Fedora, a digital preservation backend repository, introduced support for versioning in the application layer around 2004.[22]

ADVANCES IN THE COMMERCIAL SECTOR

Since the conceptualization of the 20th-century stack, there have been significant advancements made in the general IT industry. Virtualization technology developed in the 1990s led to the proliferation of cloud computing and infrastructure that transformed the IT industry in the early 2000s, leading to the "long-held dream of computing as a utility" or commodity.[23] Clouds can be public, where anyone is able to provision and use services, or private, where services are only available to a group of authorized users. Public clouds are run in commercial data centers while private clouds are typically built-in privately-owned data centers, though it's possible to use commercial data centers to build private clouds. Hybrid clouds are also possible, typically combing private and public clouds, or combining on-premises infrastructure with a private or public cloud.

In 2009, researchers at UC Berkley identified three strong reasons why cloud computing has been so widely adopted: the illusion of vertical scaling on demand, elimination of upfront cost, and the ability to pay for short-term resources.[24] Surveys from the NDSA and the Beyond the Repository grant project show a steady, but slow adoption of cloud infrastructure by the cultural heritage community.[25] It is unclear whether early adopters have chosen independently or simply followed IT changes in their parent organizations.

Any organization can build a private cloud and take advantage of the benefits described in this article. Using the cloud does not mean that you must contract with commercial cloud providers. Some organizations may choose to build a private cloud if there are concerns over data sovereignty, mistrust in public clouds, or for other reasons. The Ontario Council of University Libraries in Canada has built a private cloud for its members called the Ontario Library Research Cloud using OpenStack, a suite of open-source software for building clouds.[26]

Software-Defined Storage

While virtualization enables cloud *computing*, software-defined storage is the foundation for cloud *storage*. Software-defined storage combines inexpensive hardware with software abstractions to create a flexible, scalable, storage solution that provides data integrity.[27] Software-defined storage can use the same pool of disks to present all three of the common types of storage: file, object, and block.

File storage is what most users are familiar with. Software defined file storage creates a network file share from which files can be accessed on local devices via a filesystem.[28] Object storage in this environment is like a web-native file share; files are stored in buckets, which can be further organized by folders. Files are not accessed through a filesystem, but are instead accessed through URIs, which makes object storage very amenable to web applications and avoids some of the pitfalls of relying on filesystems. Block storage is mostly used to mount storage to virtual servers, storage that is directly attached to the server as if it was a physical disk or volume mounted to the server. Block storage is more performant than either file or object storage; as such it's typically used for things like the operating system and application code, but not for storing content. All storage can be managed through APIs, adding to its suitability for automation, software development, and IT operations.[29]

Hardware Diversity

Software defined storage also has features that make a compelling use case for digital preservation. First, software defined storage accommodates hardware diversity. Because software defined storage is an abstraction, it's possible to combine different types of storage media, from different manufacturers and production batches to ensure some technical diversity and avoid risk from catastrophic failure from a hardware monoculture.

Fixity and Integrity

Second, like the use of RAID in traditional filesystems, file integrity can be strengthened through the use of erasure coding.[30] Erasure coding splits files into chunks and spreads them across multiple disks or potentially nodes such that the file can be reconstructed if some of the disks or nodes fail. This can be configured in different ways, depending on the amount of parity desired.[31]

Replication

Third is replication of content. For cloud administrators, replication might be an alternative to erasure coding for ensuring data integrity; for digital preservationists, it's a distinct strategy and basic preservation activity. Operating nodes in a software defined storage network can be in different availability zones; through object storage policies, content can be replicated as many times as necessary to provide mitigation of geographic based threats. It's even possible to replicate to object storage in a different software defined storage network, helping to achieve organizational diversity as well.

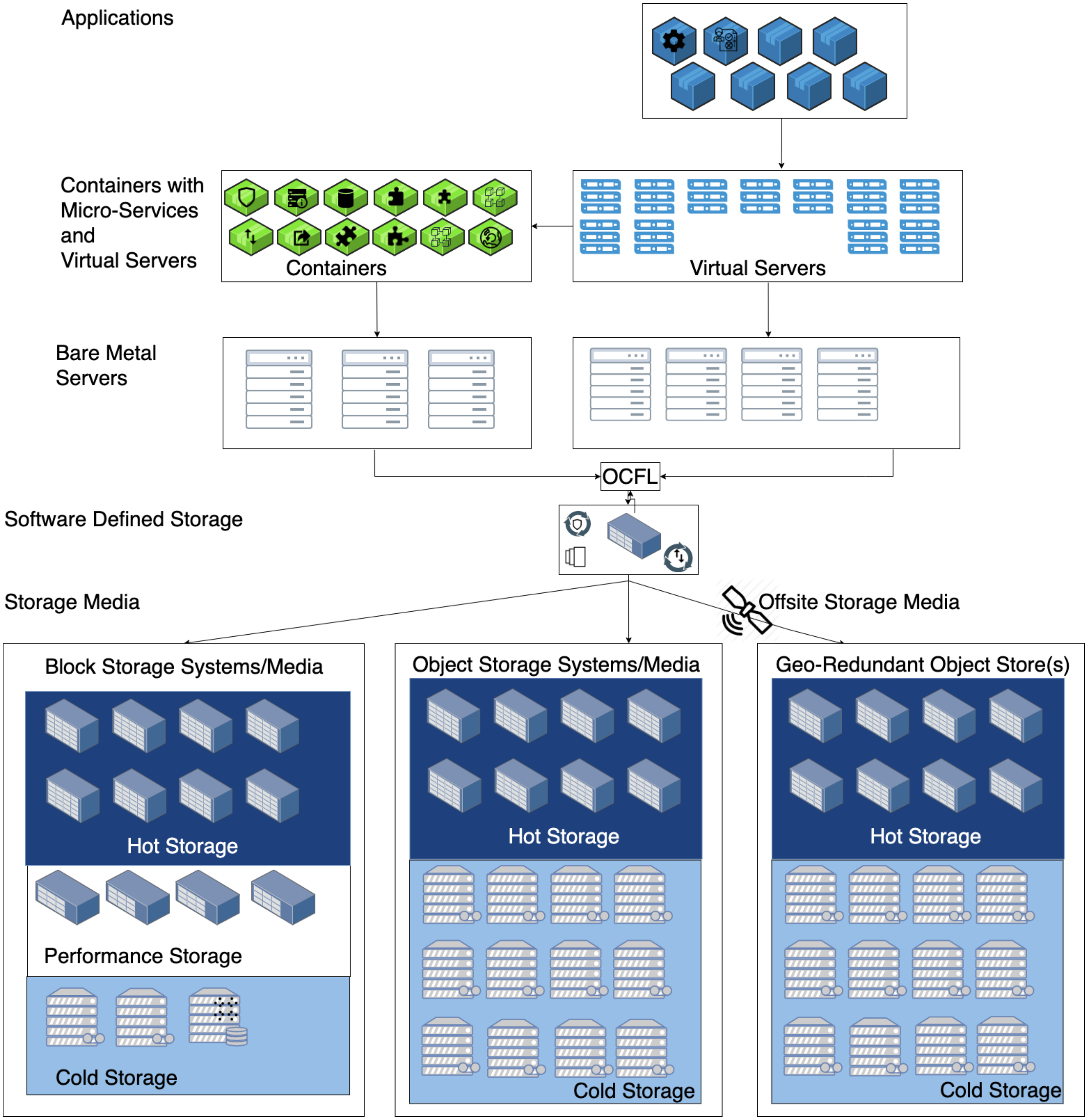

Figure 2. The infrastructure architecture for a theoretical 21st-century stack.

AN UPDATED TECHNICAL INFRASTRUCTURE FOR THE 21ST CENTURY

A theoretical 21st-century stack for digital preservation has many of the same components as its 20th-century antecedent. However, these components are used in different ways, largely due to technological advancements. Leveraging these advancements to handle digital preservation activities at lower levels of the stack reduces the complexity of the business logic in the application layer.

Figure 2 shows an updated architecture diagram for this 21st-century stack, which may be used by an individual organization, consortium, or service provider planning to build a digital preservation system. The storage layer is built on software-defined storage with digital content primarily being stored as objects; these objects are stored using the Oxford Common File Layout (discussed further later). Physical bare metal servers are used to power virtual machines that host applications such as a digital repository. Physical servers also host a container and function as a service to provide a suite of microservices for processing digital content.

Storage

In this new stack, storage is primarily managed through software defined storage with data flowing over networks. There are currently two primary open-source options for running a software-defined storage service: Gluster and Ceph. Both can be installed and run on-premises, in a private or public data center, or even contracted through infrastructure as a service (IaaS). In his presentation at the 2018 Designing Storage Architectures for Digital Collections meeting, hosted by the Library of Congress, Glenn Heinle recommended Ceph over Gluster where data integrity is the highest priority; however, others argue that Gluster is better for long-term storage.[32]This is likely because Ceph is better able to recover from hardware failures.[33]

File Storage

Reliance on file storage has become minimal in this theoretical stack, with data primarily stored as objects. However, file storage may still be used; when it is, it benefits from using a modern filesystem. Several modern filesystems have emerged since 2000, most notably ZFS and OpenZFS[34] with their innovative copy-on-write transactional model and methods for managing free space.[35] Both ZFS and OpenZFS can also be configured to use RAID-Z, which maintains block-level fixity by calculating checksums for each block of data and verifying the checksum when accessed. This can be combined with simple software to touch every block on a regular basis to ensure fixity-at-rest. Although this is a different approach than file-level fixity checks, it accomplishes the same thing in a much more efficient method: preservation metadata could be recorded for each block that contains part of the file.[36] ZFS has also inspired similar modern filesystems such as BTRFS, Apple File System (APFS), ReFS, and Resier.[37]

However, even if this theoretical stack isn't relying on file storage to persist data, software-defined storage is an abstraction that sits atop servers and disks (or tape) that do use filesystems.[38] Ironically, ZFS is not the best option for the underlying disks as its data integrity features come with more overhead and data integrity can be achieved through different means with software-defined storage.[39]

Block Storage

Block storage comes in two forms in this future stack. Many virtual servers will leverage the block storage offerings of the software defined storage service, attaching virtual disk blocks to virtual servers. However, the physical servers that support virtualization will still have some physically attached storage using SSDs (through NVMe) to support high performance storage needs. This physically attached block storage is more performant than virtually attached block storage since the system has direct access to the disks and does not have to work through a virtually abstracted filesystem.

Object Storage

Object storage has become the primary method of storing data in this theoretical stack. The flexibility of object storage, with its web-native APIs and authentication, gives it an advantage as systems become less centralized and more integrations are needed. The natural scalability of object storage and the variety of private, public, and commercial offerings greatly simplifies geographic and organizational redundancy when replicating data.

With software-defined storage, it's also possible to offer hot (live) and cold (nearline, offline) options, giving flexibility for how data is stored to better optimize the storage for various needs. Hot storage may use either hard disk or solid-state drives while cold storage would rely on tape or optical media. Presently, options for running software defined storage on tape and optical media are mostly proprietary.[40] While this would be a concern if these systems held the only copy, if the data is replicated to systems using other technology and media, this risk can be managed. While optical media has long been criticized for use as a preservation media, when well-managed, the risk may be overstated.[41]

Oxford Common File Layout

The Oxford Common File Layout (OCFL) is a "shared approach to filesystem layouts for institutional and preservation repositories."[42] OCFL is a specification for organizing digital objects in a way that supports preservation while being computationally efficient. It has several advantages for use in digital preservation, such as the ability to rebuild a repository with only the files, it's both human and machine readable, supports native error detection, allows objects to be efficiently versioned, and is designed to work with a variety of storage infrastructures.[43] Although some implementation details are still being worked out, OCFL can be used with object storage.[44] OCFL is in production use and client libraries are available for JavaScript, Java, Ruby, and Rust.[45] In this future stack, all storage operations are handled by an OCFL client, which then interacts with the underlying software defined storage network as shown in figure 2.

Servers

Physical servers are used chiefly to support virtualization in this future stack. However, this stack moves beyond virtual servers and supports containers and serverless computing. Virtual servers are chiefly used to support applications and databases while containers are perfectly suited for microservices running preservation activities.

Serverless, or function-as-a-service, is the next evolution in virtualization. While a container may be idling all the time, spinning into action when a microservice is called, serverless functions are run on-demand only. They can cost less when using commercial services as AWS Lambda or AWS Fargate where the customer is billed for usage only.[46] Though serverless functions can make use of containers, function-as-a-service platforms have emerged, such as Apache OpenWhisk and OpenFaas that don't always require containers.

PRESERVATION ACTIVITIES IN THE 21ST-CENTURY STACK

This 21st-century stack performs the same preservation activities as its predecessor. However, it generally does this at lower levels of the stack, in the infrastructure layers as opposed to the application and microservice layers. This change reduces the computational load on the stack and simplifies the business logic.

Basic Activities

Fixity and replication are achieved leveraging a combination of microservices and software-defined storage. By optimizing the approach to fixity for each use case, instead of using the same computationally intensive method for all fixity, organizations can use less compute power. While fixity and replication still involve the microservice layer, it is a more targeted approach.

Transactional Fixity

Transactional fixity is maintained through a function-as-a-service-based microservice. Each time a file is moved, the microservice is triggered, which calculates a MD5 checksum and compares it to a stored value that was created upon ingest. If there is a mismatch between the MD5 values, a second microservice is called that fetches a valid file replica. While CRC32 might be preferred (because it's slightly less CPU-intensive), Box has shown that CRC32 values can differ depending on how they are calculated.[47] A stored CRC32 can only be reliably used to confirm fixity if the new calculation uses the exact same method because CRC32 not a true specification—such as MD5—and implementations may differ. CRC32 is recommended only when it's possible to calculate in the same manner, such as the same microservice. As this introduces technical complexity, some organizations may prefer to rely solely on MD5 for transactional fixity.

Authentication Fixity

Authentication fixity is maintained in much the same way as in the 20th-century model, except using a cryptographically secure checksum algorithm, such as SHA-512 (part of the SHA-2 family). However, distinguishing between transactional vs. authentication fixity allows more precise use of algorithms, only requiring more computationally intensive cryptography when it's truly needed. Authentication fixity may require the use of a container-based microservice, versus a function-as-a-service, due to the increased computational need.

Fixity-at-Rest

Fixity-at-rest, the most common type of fixity, is managed by the software-defined storage service and reported in preservation metadata. How this is achieved might look different, depending on which software-defined storage service is used. The Ceph community has developed a new technology called BlueStor which serves as a custom storage backend that directly interacts with disks, essentially replacing the need to use an underlying filesystem.[48] BlueStor calculates checksums for every file and verifies them when read. Because this is all internal and managed by the same system, CRC32 is the default algorithm, but multiple algorithms are supported. Ceph will "scrub" data every week.

Scrubbing is the process of reading the file simply to verify the checksum, even if no user has accessed the file. Because of the way Ceph performs erasure coding, if a checksum is invalid, the file can be repaired. What remains to be done is writing a script that will read Ceph's internal metadata and record preservation events within the object's preservation metadata for the fixity verification and any reparative actions. Ryu and Park propose a "Markov failure and repair model" to optimize the frequency of data scrubbing and number of replicas such that the least amount of power is consumed and that scrubbing occurs at off-peak times.[49] It appears that this optimization causes less media degradation from the process of regularly reading the file, though empirical studies are needed to confirm that there is overall less degradation than conducting fixit checks through an application.

Gluster has a similar scrubbing process for fixity-at-rest in the optional BitRot feature, although it uses SHA-256 by default, instead of CRC32, which requires more computing power.[50]

Replication

Replication in this future stack is mostly handled by the software-defined storage service, but microservices may play a role in achieving independence of copies.[51] Object storage policies allow the automatic copying of data into another region or availability zone that is within the software defined storage network. However, these copies are not replicas, or independent instances, because all copies are in a chain derived from the primary instance; if there is a problem anywhere in the chain, bad data will be copied. In addition to using object storage policies, microservices could be used to independently verify the fixity of downstream copies as well as trigger true replications to independent instantiations, such as an alternative storage service or different storage area within the same software defined storage network. Bill Branan suggested a similar approach in his Cloud Native Preservation presentation at NDSA Digital Preservation 2019.[52]

Advanced Digital Preservation Activities

Advanced digital preservation activities in a 21st-century stack also make use of microservices for metadata extraction and file format conversion. Versioning, however, relies upon features of the Oxford Common File Layout, even though object storage often supports versioning natively.

Metadata Extraction

Function-as-a-service microservices are well suited to metadata extraction, actuated upon ingest or as needed. Since embedded metadata is machine-readable, this activity will not be resource intensive or time consuming. In addition to extracting metadata and storing it as discrete, sidecar files, these microservices can be used to populate specific metadata fields used by the repository, including descriptive, administrative, and technical metadata. This approach is more efficient as it gives flexibility to reuse the functions in multiple workflows as opposed to specific events like ingest.

File Format Conversion

File format conversions use a combination of function-as-a-service and container-based microservices, depending upon the original format. Like metadata extraction, conversion may occur at ingest (normalization) or as needed (migration). Function-as-a-service is well suited for small to medium files, such as converting WordPerfect to OpenDocument Format. Function-as-a-service is also well suited for logical preservation, when only the informational content is necessary to preserve, such as converting a TIF to a TXT file through OCR. Container-based microservices are better suited for converting large media files that may take more memory and time; function-based services often have a time constraint, for example, migrating proprietary encoded digital video to open codecs and container formats.

Versioning

Although object storage typically supports versioning, it is inefficient because each version is an entirely new object. This means that unchanged data is duplicated, taking up more disk space. The Oxford Common File Layout, which negotiates storage between the application and microservices layers and a software defined storage service, supports a forward delta versioning in which each new version only contains the changes. Using the object inventories, it's possible to rebuild any object to any version without duplicating bits.[53] An additional benefit of using OCFL is that it inherently uses checksums, and any changes or corruption are detected when an interaction occurs with the object, creating a layered approach to maintaining fixity-at-rest.

SUSTAINABILITY IN THE 21ST-CENTURY STACK

The differences between our 20th- and 21st-century stacks result in a more sustainable approach to digital preservation, per the triple-bottom-line definition.[54] By adopting commercial sector approaches, cultural heritage organizations can more efficient data centers consumers.

People (Labor)

By shifting activities to lower levels in the stack and letting infrastructure components self-manage, fewer people are needed to develop and maintain the business logic that formerly handled the same action. The application and microservice layers use programming languages and libraries that can become outdated quickly, requiring development work to maintain functionality. While there is still a need for specialized knowledge, fewer people are needed to do the work when these actions take place in more stable parts of the stack.

Planet (Environmental)

Our new stack has a lower environmental impact for a variety of reasons. First, per Kryder's Law (the storage parallel to Moore's Law for computing), the areal density of storage media predictably increases annually, and our new stack uses the latest hard disk and tape technology.[55] This results in needing less media, some of which doesn't need power to run, decreasing the carbon impact. Additionally, our new stack uses a mix of hot and cold storage, making it possible to implement automatic tiering to shift objects to less resource-intensive storage, like tape.[56]

Second, as the stack becomes more serverless, fewer computational resources are needed. Even though container and function-based microservices may incur some overhead in terms of CPU cycles, it is more efficient in terms of system idling to be running these as microservices on one platform rather than doing the same action in the application or VM layer. This further decreases the carbon impact and while also decreasing the dependency on rare-earth elements. Relatedly, by leveraging software-defined storage to maintain fixity-at-rest, the compute load is greatly decreased; the CPU cost to calculating checksums in the storage layer is less than when this is done in the through applications or microservices.

Profit (Economic)

Sustainability improvements for both people and planet may also result in a lower total cost of ownership for a digital preservation system. Cost is a prime motivator when administers and leaders make long term decisions, decreasing annual operating cost related to digital preservation is crucial to the viability of a program.

FUTURE AND RELATED WORK

The 21st-century stack proposed in this paper is not the only way to increase sustainability or the only way in which digital preservation stacks will change. The planet is running out of bandwidth and will need to expand into using 5G and low-earth orbit satellite communications. New, quantum-resistant algorithms will need to be introduced as quantum computing advances.[57]

Blockchain technology introduces many possibilities. Inherently, blockchain maintains fixity. The ARCHANGEL project is exploring practical methods of recording provenance and proving authenticity by using a permissioned blockchain.[58] Blockchain is also the technology behind the InterPlanetary File System (IPFS), a content-addressed distributed storage network, that is in turn used by Filecoin, a marketplace for an IPFS storage. Small Data Industries is building Starling, a Filecoin-based application designed for cultural heritage organizations to securely store digital content.[59] It's important to note that these blockchain-based projects use a Proof-of-Stake model instead of a Proof-of-Work model, which has a significantly lower environmental impact than other blockchain implementations like the cryptocurrency Bitcoin.[60]

While some organizations, like Stanford University, may already leverage software-defined storage, most in the cultural heritage sector are not.[61] The MetaArchive Cooperative, a nonprofit consortium for digital preservation, has begun a noteworthy project to explore using software-defined storage in a distributed digital preservation network. MetaArchive, which currently uses LOCKSS, is one of the few public digital preservation services that mitigates risk through organizational and administrative diversity. Because members host and administer the LOCKSS nodes that contain the replications, each copy is managed by a different set of organizational and administrative policies and often use different technology to do so. Diversifying in this way protects against a single point of failure if only one organization managed the technical infrastructure. How this same diversity is achieved in a software-defined storage-based distributed digital preservation network will be a great contribution to the community.

It would also be useful to study the reasons cultural heritage organizations have been so reluctant to adopt commercial sector technologies. Identifying these hesitations would make it possible to create strategies that would encourage adoption of these approaches. It may simply be that when it comes to digital preservation, familiar and proven technologies provide a level of comfort. Organizations may also be entrenched in custom developed solutions that are hard to move away from.

CONCLUSION

Digital preservation is a long-term commitment. While re-appraisal may take place, it's inevitable that the amount of content stored in digital repositories will only increase over time. It is fiduciarily incumbent upon the cultural heritage community to examine our practices and look for better alternatives. Exceptionalism ignores technological advancements made by the commercial industry, advancements that are very well suited to digital preservation. By adopting commercial industry data practices, such as software-defined storage, while simultaneously implementing innovations from within the cultural heritage community, like the Oxford Common File Layout, it is possible to reduce the ongoing costs, resource consumption, and environmental impact of digital preservation.

ENDNOTES

[1] Ben Goldman, "It's Not Easy Being Green(e): Digital Preservation in the Age of Climate Change," in *Archival Values: Essays in Honor of Mark A. Greene*, ed. Mary A. Caldera and Christine Weidman (Chicago: American Library Association, 2018), 274–95, https://scholarsphere.psu.edu/concern/generic_works/bvq27zn11p.

[2] "A Simple Explanation of the Triple Bottom Line," University of Wisconsin Sustainable Management, October 2, 2019, https://perma.cc/2HWF-3MMQ.

[3] Goldman, "It's Not Easy Being Green(e)."

[4] Keith L. Pendergrass et al., "Toward Environmentally Sustainable Digital Preservation," *The American Archivist* 82, no. 1 (2019): 165–206, https://doi.org/10.17723/0360-9081-82.1.165.

[5] Sarah Barsness et al., *2017 Fixity Survey Report*: *An NDSA Report* (OSF, April 24, 2018), https://doi.org/10.17605/OSF.IO/SNJBV; David S. H. Rosenthal et al., "Requirements for Digital Preservation Systems: A Bottom-Up Approach," *D-Lib Magazine* 11, no. 11 (2005), https://perma.cc/X2R7-R5XP.

[6] Matthew Addis, *Which Checksum Algorithm Should I Use?* (DPC Technology Watch Guidance note, Digital Preservation Coalition, December 11, 2020), https://doi.org/10.7207/twgn20-12.

[7] PREMIS Editorial Committee, *PREMIS Data Dictionary for Preservation Metadata*, version 3.0 (Library of Congress, November 2015), https://perma.cc/L79V-GQV7.

[8] Some organizations may continue to use a strategy where fixity is only checked when a file is accessed, if the potential loss fits within a defined acceptable loss. While this strategy may not work for all organizations, recognizing that loss is inevitable and defining a level of acceptable loss is an effective and pragmatic approach to managing risk of bit decay.

[9] Barsness et al., *2017 Fixity Survey Report*.

[10] NDSA Levels of Preservation Revisions Working Group, *"Levels of Digital Preservation, 2019 LOP Matrix*, V2.0 (OSF, October 14, 2019), https://osf.io/2mkwx/.

[11] Sibyl Schaefer et al., "User Guide for the Preservation Storage Criteria," February 25, 2020, https://doi.org/10.17605/OSF.IO/SJC6U.

[12] Mark Carlson et al., "Software Defined Storage," (white paper, Storage Network Industry Association, January 2015), https://perma.cc/AQ4T-9YXQ.

[13] Abutalib Aghayev et al., "File Systems Unfit as Distributed Storage Backends" (Proceedings of the 27th ACM Symposium on Operating Systems Principles—SOSP '19, Huntsville, Ontario, Canada: Association for Computing Machinery, 2019): 353–69, https://doi.org/10.1145/3341301.3359656.

[14] NDSA Storage Infrastructure Survey Working Group, *2019 Storage Infrastructure Survey: Results of the Storage Infrastructure Survey* (OSF, 2020), https://doi.org/10.17605/OSF.IO/UWSG7.

[15] Joseph Migga Kizza, "Virtualization Technology and Security," in *Guide to Computer Network Security*, Computer Communications and Networks (Springer, Cham, 2017), 457–75, https://doi.org/10.1007/978-3-319-55606-2_21.

[16] NDSA Storage Infrastructure Survey Working Group, *2019 Storage Infrastructure Survey*.

[17] Eric Jonas et al., "Cloud Programming Simplified: A Berkeley View on Serverless Computing" (University of California, Berkeley, February 10, 2019), https://perma.cc/YAM2-TZ8W.

[18] Alex Garnett, Mike Winter, and Justin Simpson, "Checksums on Modern Filesystems, or: On the Virtuous Consumption of CPU Cycles," in *IPres 1028 Conference [Proceedings]* (International Conference on Digital Preservation, Boston, Mass., 2018), https://doi.org/10.17605/OSF.IO/Y4Z3E.

[19] Barsness et al., *2017 Fixity Survey Report*.

[20] "Import Metadata," documentation for Archivematica 1.12.1, Artefactual Systems, Inc., accessed May 21, 2021, https://perma.cc/UE3R-BDGZ.; "Ingest," documentation for Archivematica 1.12.1, Artefactual Systems, Inc., accessed May 21, 2021, https://perma.cc/5SN5-GFX3.

[21] NDSA Storage Infrastructure Survey Working Group, *2019 Storage Infrastructure Survey*.

[22] "Fedora Content Versioning," 2005, https://duraspace.org/archive/fedora/files/download/2.0/userdocs/server/features/versioning.html.

[23] Michael Armbrust et al., *Above the Clouds: A Berkeley View of Cloud Computing*, (technical report, EECS Department, University of California, Berkeley, February 10, 2009), https://perma.cc/QJ5W-8S5Y.

[24] Armbrust et al., *Above the Clouds*.

[25] Micah Altman et al., "NDSA Storage Report: Reflections on National Digital Stewardship Alliance Member Approaches to Preservation Storage Technologies," *D-Lib Magazine* 19, no. 5/6 (May 2013), https://doi.org/10.1045/may2013-altman; Michelle Gallinger et al., "Trends in Digital Preservation Capacity and Practice: Results from the 2nd Bi-Annual National Digital Stewardship Alliance Storage Survey," *D-Lib Magazine* 23, no. 7/8 (2017), https://doi.org/10.1045/july2017-gallinger; NDSA Storage Infrastructure Survey Working Group, *2019 Storage Infrastructure Survey*; Evviva Weinraub et al., *Beyond the Repository: Integrating Local Preservation Systems with National Distribution Services* (Northwestern University, 2018), https://doi.org/10.21985/N28M2Z.

[26] Ontario Council of University Libraries, "Ontario Library Research Cloud," accessed April 14, 2021, https://perma.cc/KMP9-FS8K; "Open Source Cloud Computing Infrastructure," OpenStack, accessed April 14, 2021, https://perma.cc/G9GE-92JD.

[27] Nathan Tallman, "Software Defined Storage," (presentation for the NDSA Infrastructure Interest Group, March 16, 2020), https://doi.org/10.26207/3nn2-zv13.

[28] These network shares typically use the SMB (Server Message Block) or CIFS (Common Internet File System) protocols to present file shares through a graphical user interface in operating systems such as Windows or macOS while the NFS (Network File Shares) protocol is more often used to mount storage in Linux.

[29] Carlson et al., "Software Defined Storage."

[30] RAID, or the Redundant Array of Independent Disks, is technology that splits a file into multiple chunks and spreads them across multiple disks in a storage device, adding extra copies of the chunks so that the file can be recovered if an individual drive fails.

[31] Abhijith Shenoy, "The Pros and Cons of Erasure Coding & Replication vs RAID in Next-Gen Storage Platforms" (Software Developer Conference, Storage Networking Industry Association, 2015), https://perma.cc/YFS5-KXKK.

[32] Glenn Heinle, "Unlocking Ceph" (presentation, Designing Storage Architectures for Digital Collections, Washington, DC: Library of Congress, 2019), https://perma.cc/Z2R9-79ZE; Tamara Scott, "Big Data Storage Wars: Ceph vs Gluster," *TechnologyAdvice* (blog), May 14, 2019, https://perma.cc/2YY2-BBXG.

[33] Giacinto Donvito, Giovanni Marzulli, and Domenico Diacono, "Testing of Several Distributed File-Systems (HDFS, Ceph and GlusterFS) for Supporting the HEP Experiments Analysis," *Journal of Physics: Conference Series* 513, no. 4 (June 2014): 042014, https://doi.org/10.1088/1742-6596/513/4/042014.

[34] Matthew Ahrens, "OpenZFS: A Community of Open Source ZFS Developers," in *AsiaBSDCon 2014* (AsiaBSDCon, Tokyo, Japan: BSD Research, 2014), 27–32, https://perma.cc/XG79-PBU7.

[35] Brian Hickmann and Kynan Shook, "ZFS and RAID-Z: The Über-FS?" (University of Wisconsin–Madison, December 2007), https://perma.cc/W5PD-ENPP.

[36] Garnett, Winter, and Simpson, "Checksums on Modern Filesystems."

[37] Edward Shishkin, "Resier5 (Format Release 5.X.Y)," MARC mailing list archive, 2019, https://perma.cc/DN8Y-V8KQ.

[38] "Fujifilm Launches 'Fujifilm Software-Defined Tape,'" FUJIFILM Europe, May 19, 2020, https://perma.cc/B3GN-PLR9.

[39] Aghayev et al., "File Systems Unfit as Distributed Storage Backends."

[40] IBM Systems, "Tape Goes High Speed," 2016, https://perma.cc/FNV9-RTG9; "Fujifilm Launches 'Fujifilm Software-Defined Tape'"; Desire Athow, "Here's What Sony's Million Gigabyte Storage Cabinet Looks Like," *TechRadar* (blog), 2020, https://perma.cc/VHN4-LAYT.

[41] David Rosenthal, "Optical Media Durability: Update," *DSHR's Blog*, August 20, 2020, https://perma.cc/VKW9-83J3.

[42] Andrew Hankinson et al., "The Oxford Common File Layout: A Common Approach to Digital Preservation," *Publications* 7, no. 2 (June 2019): 39, https://doi.org/10.3390/publications7020039.

[43] Andrew Hankinson et al., "Oxford Common File Layout Specification," July 7, 2020, https://perma.cc/S73Z-3N6K.

[44] Marco La Rosa et al., "Our Thoughts on OCFL over S3 · Issue #522 · OCFL/Spec," GitHub, accessed March 12, 2021, https://perma.cc/PA3G-CB78.

[45] Hannah Frost, "Version 1.0 of the Oxford Common File Layout (OCFL) Released," *Stanford Libraries* (blog), July 23, 2020, 1, https://perma.cc/5J5M-GYQW; Andrew Woods, "Implementations | OCFL/Spec," GitHub, February 10, 2021, https://github.com/OCFL/spec.

[46] While serverless might be the ultimate microservice, requiring the least amount of overhead, costs may still be hard to predict.

[47] Ryan Luecke, "CRC32 Checksums; The Good, the Bad, and the Ugly," *Box Blog*, October 12, 2011, https://perma.cc/MVP7-YVZV.

[48] Aghayev et al., "File Systems Unfit as Distributed Storage Backends."

[49] Junkil Ryu and Chanik Park, "Effects of Data Scrubbing on Reliability in Storage Systems," *IEICE TRANSACTIONS on Information and Systems* E92-D, no. 9 (September 1, 2009): 1639–49, https://doi.org/10.1587/transinf.E92.D.1639.

[50] Raghavendra Talur, "BitRot Detection | Gluster/Glusterfs-Specs," GitHub, August 15, 2015, https://github.com/gluster/glusterfs-specs/blob/fe4c5ecb4688f5fa19351829e5022bcb676cf686/done/GlusterFS%203.7/BitRot.md.

[51] Schaefer et al., "User Guide for the Preservation Storage Criteria."

[52] Bill Branan, "Cloud-Native Preservation" (OSF, October 22, 2019), https://osf.io/kmdyf/.

[53] Andrew Hankinson et al., "Implementation Notes, Oxford Common File Layout Specification," July 7, 2020, https://perma.cc/PVF8-SQFN.

[54] Although out of scope in terms of the stack, the policies and practices implemented by organizations can have a direct impact on digital preservation sustainability. For example, appraisal can be the most powerful tool available to an organization to control the amount of content being preserved. Despite storage vendors proclamations that storage is cheap, digital preservation is not. It is not wise nor necessary to keep every digital file. Organizations will benefit from applying flexible appraisal systems that reduce the amount of content needing preservation, but also establishing different classes of preservation so the most advanced activities are only applied as needed. Additionally, organizations should consider allowing lossy compression to decrease disk usage, where appropriate; compression as an appraisal choice is very similar to choosing to sample a grouping of material rather than preserving the whole. For additional information see Nathan Tallman and Lauren Work, "Approaching

Appraisal: Framing Criteria for Selecting Digital Content for Preservation," in *IPres 1028 Conference [Proceedings]* (International Conference on Digital Preservation, Boston, Mass.: OSF, 2018), https://doi.org/10.17605/OSF.IO/8Y6DC.

[55] David Rosenthal, "Cloud for Preservation," *DSHR's Blog*, 2019, https://perma.cc/ZLS9-R857.

[56] Pendergrass et al., "Toward Environmentally Sustainable Digital Preservation."

[57] Henry Newman, "Industry Trends" (presentation, Designing Storage Architectures for Digital Collections, Washington, DC: Library of Congress, 2019), https://perma.cc/3MGK-N5U3.

[58] T. Bui et al., "ARCHANGEL: Trusted Archives of Digital Public Documents," in *Proceedings ACM Document Engineering 2018* (Association for Computing Machinery, arxiv.org, 2018), https://doi.org/10.1145/3209280.3229120.

[59] Ben Fino-Radin and Michelle Lee, "[Starling]" (presentation, Designing Storage Architectures for Digital Collections, Washington, DC: Library of Congress, 2019), https://perma.cc/7LGU-UEW9.

[60] For additional information on the differences of proof-of-stake vs. proof-of-work models, see Peter Fairley, "Ethereum Plans to Cut Its Absurd Energy Consumption by 99 Percent," *IEEE Spectrum* (blog), January 2, 2019, https://perma.cc/GCH7-T556.

[61] Julian Morley, "Storage Cost Modeling" (presentation, PASIG, Mexico City, Mexico, 2019), https://doi.org/10.6084/m9.figshare.7795829.v1.