

Vrije Universiteit Amsterdam

Universiteit van Amsterdam



Master Thesis

The CERN Digital Memory Platform

Building a CERN scale OAIS compliant Archival Service

Author: Jorik van Kemenade (2607628)

Supervisor: dr. Clemens Grelck

2nd reader: dr. Ana Lucia Varbanescu

*A thesis submitted in fulfillment of the requirements for
the joint UvA-VU Master of Science degree in Computer Science*



The CERN Digital Memory Platform

Building a CERN scale OAIS compliant Archival Service

Jorik van Kemenade

Abstract

CERN produces a large variety of research data. This data plays an important role in CERN's heritage and is often unique. As a public institute, it is CERN's responsibility to preserve current and future research data. To fulfil this responsibility, CERN wants to build an "Archive as a Service" that enables researchers to conveniently preserve their valuable research.

In this thesis we investigate a possible strategy for building a CERN wide archiving service using an existing preservation tool, Archivematica. Building an archival service at CERN scale has at least three challenges. 1) The amount of data: CERN currently stores more than 300PB of data. 2) Preservation of versioned data: research is often a series of small, but important changes. This history needs to be preserved without duplicating very large datasets. 3) The variety of systems and workflows: with more than 17,500 researchers the preservation platform needs to integrate with many different workflows and content delivery systems.

The main objective of this research is to evaluate if Archivematica can be used as the main component of a digital archiving service at CERN. We discuss how we created a distributed deployment of Archivematica and increased our video processing capacity from 2.5 terabytes per month to approximately 15 terabytes per month. We present a strategy for preserving versioned research data without creating duplicate artefacts. Finally, we evaluate three methods for integrating Archivematica with digital repositories and other digital workflows.

Contents

1	Introduction	7
2	Digital preservation	11
2.1	Digital preservation concepts	11
2.2	Open Archival Information System (OAIS)	14
2.3	Digital preservation systems	16
3	CERN Digital Memory Platform	21
3.1	Digital Preservation at CERN	21
3.2	Requirements	22
3.3	Building an OAIS compliant archive service	24
4	Vertical scaling	29
4.1	Archivematica Performance	29
4.2	Storage Scaling	31
5	Horizontal scaling	35
5.1	Distributing Archivematica	35
5.2	Task management	37
5.3	Distributed image processing	39
5.4	Distributed video processing	40
6	Versioning and deduplication	45
6.1	The AIC versioning strategy	45
6.2	Case study: Using versioned AICs for Zenodo	47
7	Automated job management	51
7.1	Automation tools	51
7.2	Archivematica API client	53
7.3	Enduro	55
8	Discussion and conclusion	57

Chapter 1

Introduction

For centuries scientists have relied upon two paradigms for understanding nature, theory and experimentation. During the final quarter of last century a third paradigm emerged, computer simulation. Computer simulation allows scientists to explore domains that are generally inaccessible to theory or experimentation. With the ever growing production of data by experiments and simulations a fourth paradigm emerged, data-intensive science [1].

Data-intensive science is vital to many scientific endeavours, but demands specialised skills and analysis tools: databases, workflow management, visualisation, computing, and many more. In almost every laboratory “born digital” data is accumulated in files, spreadsheets, databases, notebooks, websites, blogs and wikis. Astronomy and particle physics experiments generate petabytes of data. Currently, CERN stores almost 300 petabytes of research data. With every upgrade of the Large Hadron Collider (LHC), or the associated experiments, the amount of acquired data grows even faster. By the early 2020’s, the experiments are expected to generate 100 petabytes a year. By the end of the decade this has grown to 400 petabytes a year. As a result of this the total data volume is expected to grow to 1.3 exabytes by 2025 and 4.3 exabytes by 2030 [2].

Before building the LHC, CERN was performing experiments using the Large Electron-Positron Collider (LEP). Between 1989 and 2000, the four LEP experiments produced about 100 terabytes of data. In 2000, the LEP and the associated experiments were disassembled to make space for the LHC. As a result of this the experiments cannot be repeated, making their data unique. To make sure that this valuable data is not lost, the LEP experiments saved all their data and software to tape. Unfortunately, due to unexpectedly high tape-wear, two tapes with data were lost.

Regrettably hardware failure is not the only threat to this data. Parts of the reconstructed data is inaccessible because of deprecated software. In addition to this, a lot of specific knowledge about the experiments and data is lost because user-specific documentation, analysis code, and plotting macros never made it into the experiment’s repositories [3]. So even though the long term storage of files and associated software was well organised, the LEP data is still at risk.

But even when carefully mitigating hardware and software failures, data is simply lost because the value of the data was not recognised at the time. A notable example are the very first web pages of the World Wide Web. This first website, CERN’s homepage, and later versions were deleted during updates. In 2013, CERN started a project to rebuild and to preserve the first web page and other artefacts that were associated with the birth of the web. During this project volunteers rebuilt the first ever website¹, but also saved or recreated the first web browsers, web servers, documentation and even original server names and IP-addresses [4].

This are some examples of lost data, threatened data, and data that is saved by chance. For each example there are countless others, both inside CERN and at other institutes. Fortunately, there is a growing acknowledgement in the scientific community that digital preservation deserves attention. Sharing of research data and artefacts is not enough, it is essential to capture the structured information of the research data analysis workflows and processes to ensure the usability

¹ This page can be found on the original url: <http://info.cern.ch/>

and longevity of results [5]. To move from a model of preservation by chance to preservation by mission, CERN started the CERN Digital Memory Project [6]. The goal of the Digital Memory project is to preserve CERN's institutional heritage through three initiatives.

The first initiative is a digitisation project. This project aims to preserve CERN's analogue multimedia carriers and paper archives through digitisation. The multimedia archive consists of hundreds of thousands of photos, negatives, and video and audio tapes. The multimedia carriers are often fragile and damaged. The digitisation is performed by specialised partners, and the resulting digital files will be preserved by CERN.

The second initiative is Memory Net. The goal of Memory Net is to make digital preservation an integral part of CERN's culture and processes. Preservation is usually an afterthought: it is easy to postpone and does not provide immediate added-value. By introducing simple processes, leadership commitment, and long-term budgets, Memory Net changes the preservation of CERN's institutional heritage from an ad-hoc necessity to an integral part of the data management strategy.

The third initiative is creating the CERN Digital Memory Platform, a service for preserving digitised and born-digital content. The main goal of the CERN Digital Memory Platform is to serve as a true digital archive, rather than as a conventional backup facility. The idea is that all researchers at CERN can connect their systems to the archiving service and use it to effortlessly preserve their valuable research.

Building a digital archive at the scale of CERN is not without challenges. The obvious challenge is the size of the archive. Currently, CERN is storing 300 petabytes of data. This is significantly larger than the median archive size of 25 terabytes [7]. The largest archive in this study is 5.5 petabytes and the total size of all archives combined is 66.8 petabytes. Assuming that CERN can archive material at a rate of the largest archive per year, processing only a quarter of the current backlog takes 14 years.

Fortunately, CERN already takes great care in preserving raw experimental data. This means that the archiving effort only has to focus on preserving the surrounding research: software, multimedia, documentation, and other digital artefacts. One of the characteristics of research is that it is often the result of many incremental improvements over longer periods of time. Preserving every version of a research project, including large data sets, results in a lot of duplication. Consequently, we need to preserve all versions of a research project without duplicating large datasets.

The third, and last, challenge is integrating the CERN Digital Memory Platform into existing workflows. With more than 17,500 researchers from over 600 institutes working on many different experiments there is a large variety in workflows and information systems. The CERN Digital Memory Platform will only be used if it allows users to conveniently deposit new material into the archive. This requires that the archiving service is scalable in the number of connected systems, and in the variety of material that can be preserved.

In this thesis we investigate a possible approach for creating the CERN Digital Memory Platform. More specifically we want to investigate if it is possible to build the platform using currently existing solutions. The first step is investigating a selection of existing and past preservation initiatives, preservation standards, tools and systems. For each component we determine if they meet the requirements for the CERN Digital Memory Platform. This analysis forms the basis for selecting the standards and systems used for creating the CERN Digital Memory Platform.

Based on this analysis we selected Archivematica for building the CERN Digital Memory Platform. Before committing to use Archivematica for the project, it is important to verify that Archivematica can be used to address each of the three challenges. The first challenge is the size of the preservation backlog. To evaluate if Archivematica has the required capacity for processing the preservation backlog, we evaluate the performance of a default Archivematica deployment. During this evaluation we benchmark the performance of Archivematica for simple preservation tasks.

During the initial investigation we identified two bottlenecks. The first bottleneck is the size of the local storage. When processing multiple transfers simultaneously, Archivematica runs out of local storage. The storage requirements of Archivematica are too demanding for the virtual machines offered in the CERN cloud. To solve this problem we investigate various large scale external storage solutions. For each option, we benchmark the raw performance and the impact on the preservation throughput.

The second bottleneck is processing power. A single Archivematica server cannot deliver the required preservation throughput for processing the massive preservation backlog. This means that we need to investigate how Archivematica can scale beyond a single server. We present a strategy for deploying a distributed Archivematica cluster. To evaluate the performance of a distributed Archivematica cluster we benchmark the archiving throughput for both photo and video preservation. For each workload we compare the performance of the distributed Archivematica cluster to the performance of a regular Archivematica deployment and evaluate the scalability.

The second challenge is supporting the preservation of versioned data. One problem with archiving every version of a digital object is duplication. Duplicate data has triple cost: the processing of duplicate data, the storage space of the data, and the migration costs of the data. By default Archivematica does not support deduplication or versioning of preserved data. We propose to solve this by using a strategy that we decided to call “AIC versioning”. AIC versioning uses a week archiving strategy to create a preservation system agnostic strategy for preserving highly versioned data. To assess the effectiveness of AIC versioning for preserving scientific data, we present a case-study using sample data from Zenodo, a digital repository for research data. In this case-study we compare the expected archive size with and without AIC versioning for a sample of Zenodo data.

The third, and final, challenge is integrating the CERN Digital Memory Platform with existing workflows. We investigate three options for managing and automating the transfer of content into Archivematica: the automation-tools, the Archivematica API, and Enduro. For each option we discuss the design philosophy and goals. After this we discuss how each of the alternatives can be used to handle the workload for many different services using multiple Archivematica pipelines.

Finally we evaluate if the combination of a distributed Archivematica deployment, the AIC versioning strategy, and one of the workflow management solutions can be used as the central building block of the CERN Digital Memory Platform. We want to know if this combination solves the challenges and meets the requirements set for the CERN Digital Memory Platform. We also want to know what problems are not addressed by the proposed solution. Ultimately we want to understand if this is a viable strategy, or if an entirely different approach might be advised.

To summarise, the specific contributions of this thesis are:

- A literature study describing the evolution of the digital preservation field.
- A method for creating a scalable distributed Archivematica cluster.
- A strategy for handling the preservation and deduplication of versioned data.
- A comparison of existing Archivematica workload management systems.

The rest of this thesis has the following structure. Chapter 2 introduces digital preservation concepts, the OAIS reference model, and existing digital preservation standards, tools, and systems. Chapter 3 discusses some of CERN’s earlier preservation efforts and the requirements and high-level architecture of the CERN Digital Memory Platform. Chapter 4 evaluates the base-line performance of Archivematica and the performance of different storage platforms. Chapter 5 introduces the distributed Archivematica deployment, discusses the required changes for efficiently using this extra capacity, and evaluates the image and video processing capacity of Archivematica. Chapter 6 introduces the AIC versioning strategy and evaluates the influence of AIC versioning on the required storage capacity in a case-study. Chapter 7 discusses several options for managing the workload on one or multiple Archivematica pipelines and discusses possible solutions for integrating Archivematica in the existing workflows. Finally, Chapter 8 evaluates the entire study.

Chapter 2

Digital preservation

Putting a book on a shelf is not the same as preserving or archiving a book. Similarly, digital preservation is not the same as ordinary data storage. Digital preservation requires a more elaborate process than just saving a file to a hard disk and creating a backup. Digital preservation, just like traditional preservation, can be described as a series of actions taken to ensure that a digital object remains accessible and retains its value.

Within the digital preservation community, the Open Archival Information System (OAIS) reference model is the accepted standard for describing a digital preservation system. The reference model clearly defines the roles, responsibilities, and functional units within an OAIS. The OAIS reference model only defines actions, functionality, interfaces, and responsibilities. The model does not supply an actual system architecture or implementation.

To create a better understanding of the digital preservation field and the existing literature, we start with discussing some of the important digital preservation concepts and challenges. Next, we discuss the goals of the OAIS model, provide an overview of the most important concepts and terminology and discuss some common problems of the OAIS reference model. Finally, we provide an overview of earlier work in OAIS compliant archives and discusses some of the past and present digital preservation initiatives and projects.

2.1 Digital preservation concepts

There is not a single definition for digital preservation. Digital preservation is rather seen as a continuous process of retaining the value of a collection of digital objects [8]. Digital preservation protects the value of digital products, regardless of whether the original source is a tangible artefact or data that was born and lives digitally [9]. This immediately raises the question of what is the value of a digital collection and when is this value retained? The answer to these questions is: it depends. Digital preservation is not one thing: it is a collection of many practices, policies and structures [10]. The practices help to protect individual items against degradation. The policies ensure the longevity of the archive in general. All practices, policies and structures combined is what we call a digital preservation system: a system where the information it contains remains accessible over a long period of time. This period of time being much longer than the lifetime of formats, storage media, hardware and software components [11].

Digital preservation is a game of probabilities. All activities are undertaken to reduce the likelihood that an artefact is lost or gets corrupted. There is a whole range of available measures that can be taken to ensure the preservation of digital material. Figure 2.1 shows some measures in the style of Maslow’s hierarchy of needs [12]. Each of these measures have a different impact, both in robustness and required commitment. The measures can be divided into tow categories: bit-level preservation and object metadata collection.

A vital part of preserving digital information is to make sure that the actual bitstreams of the objects are preserved. Keeping archived information safe is not very different from keeping “regular” information safe. Redundancy, back-ups and distribution are all tactics to make sure

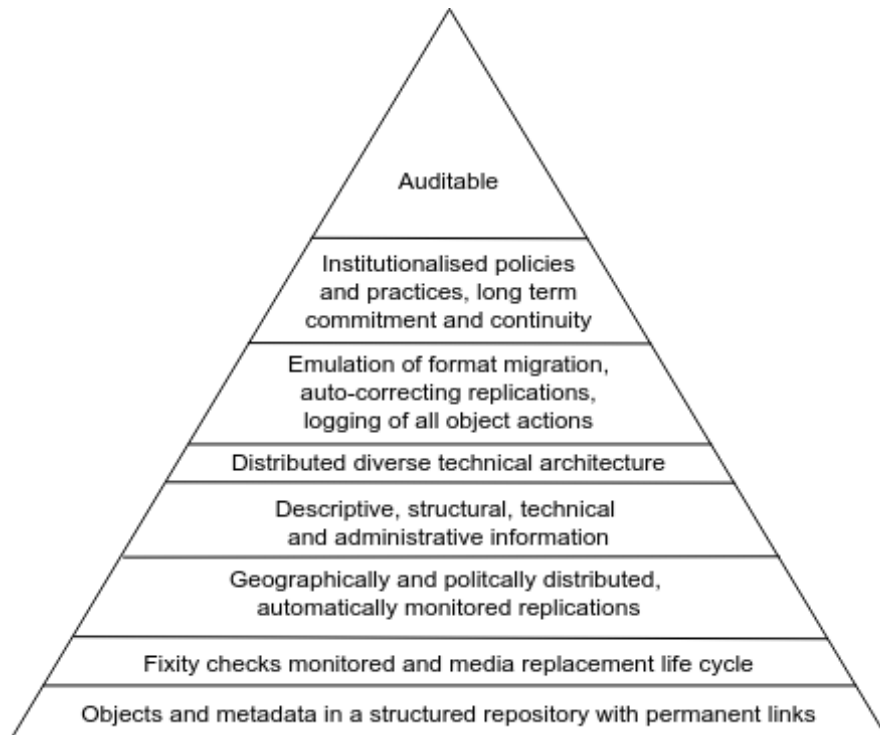


Figure 2.1: Wilson's hierarchy of preservation needs [12]. Each additional layer improves the preservation system at the expense of more commitment of the organisation. Depending on the layer, this commitment is primarily technical or organisational.

that the bitstream is safe. One vital difference between bit-preservation and ordinary data storage is that an archive needs to prove that the stored information is unchanged. This is done using fixity checks. During a fixity check, the system verifies that a digital object has not been changed between two events or between points in times. Technologies such as checksums, message digests and digital signatures are used to verify a digital object's fixity [13]. By performing regular fixity checks the archive can prove the authenticity of the preserved digital material.

Another part of maintaining the integrity of the archive is to monitor file formats. Like any other digital technology, file formats come and go. This means that a file format that is popular today, might be obsolete in the future. If a digital object is preserved using today's popular standard, it might be impossible to open in the future. There are two mechanisms that can prevent a file from turning into a random stream of bits: normalisation and migration. Normalisation is the process of converting all files that need to be preserved to a limited set of file formats. These file formats are selected because they are safe. This means that they are (often) non-proprietary, well documented, well supported and broadly used within the digital preservation community.

Migration is the transfer of digital materials from one hardware or software configuration to another. The purpose of migration is to preserve the integrity of digital objects, allowing clients to retrieve, display, and otherwise use them in the face of constantly changing technology [14]. An example of migration is to convert all files in a certain obsolete file format to a different file format. A common strategy for preserving the accessibility of files in a digital archive is to combine normalisation and migration. Normalisation ensures that only a limited set of file formats need to be monitored for obsolescence. Migration is used to ensure access in the inevitable case that a file format is threatened with obsolescence.

The second category of measures in digital preservation is metadata collection and management. It has been widely assumed that for (digital) information to remain understandable over time there is a need to preserve information on the technological and intellectual context of a digital artefact [11, 15, 16]. This information is preserved in the form of metadata. The Library of Congress defines three types of metadata [17]:

Descriptive metadata Metadata used for resource discovery, e.g. title, author, or institute.

Structural metadata Metadata used describing objects, e.g. number of volumes or pages.

Administrative metadata Metadata used for managing a collection, e.g. migration history.

Metadata plays an important role in ensuring and maintaining the usability and the authenticity of an archive. For example, when an archive uses a migration strategy metadata is used to record the migration history. This migration history is used for proving the authenticity of the objects. Each time a record is changed, e.g. through migration, the preservation action is recorded and a new persistent identifier is created. These identifiers can be used by users to verify that they are viewing a certain version of a record. This metadata is also helpful for future users of the content, it provides details needed for understanding the original environment in which the object was created and used.

To make sure that metadata is semantically well defined, transferable, and can be indexed it is structured using metadata standards. Different metadata elements can often be represented in several of the existing metadata schemas. When implementing a digital preservation system, it is helpful to consider that the purpose of each of the competing metadata schemas is different. Usually, a combination of different schemas is the best solution. Common combinations are; METS and PREMIS with MODS, as used by the British Library [18]; or METS and PREMIS with Dublin Core, as used by Archivematica [19].

METS is an XML document format for encoding complex objects within libraries. A METS file is created using a standardised schema that contains separate sections for: descriptive metadata, administrative metadata, inventory of content files for the object including linking information, and a behavioural metadata section [20]. PREMIS is a data dictionary that has definitions for preservation metadata. The PREMIS Data Dictionary defines “preservation metadata” as the information a repository uses to support the digital preservation process. Specifically, the metadata supporting the functions of maintaining viability, renderability, understandability, authenticity, and identity in a preservation context. Particular attention is paid to the documentation of digital provenance of the history of an object. [21]. Dublin Core [22] and MODS [23] are both standards for descriptive metadata.

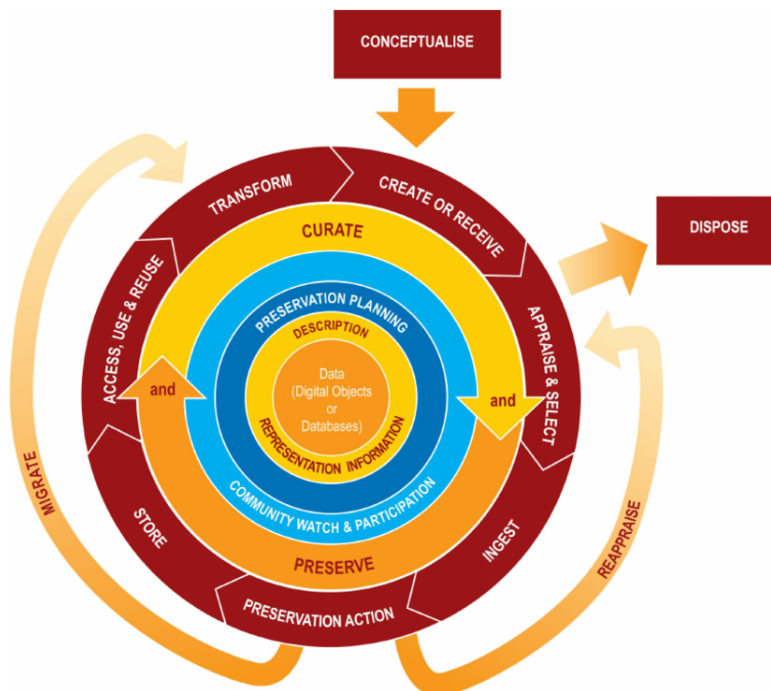


Figure 2.2: The DCC Curation Lifecycle Model [24]. High-level overview of the lifecycle stages required for successful digital preservation. The centre of the model contains the fundamental building blocks of a digital preservation system, the outer layers display the curation and preservation activities.

Both the British Library and Archivematica use METS as the basis for creating structured archival objects. The METS file contains all the different elements of the object and their relationships. The descriptive metadata is added to the METS file using a standard for descriptive metadata, in this case MODS or Dublin Core. All the other metadata like file formats, preservation actions, and rights data is added using PREMIS objects. Extending METS with PREMIS and other popular metadata standards is accepted practice within the digital archiving community. Other digital preservation systems use similar solutions, or slight variations, to structure their metadata.

This combination of selection, enhancement, ingestion, and transformation are essential stages for the preservation of data. Figure 2.2 shows how all of these stages fit together in the DCC (Digital Curation Centre) Curation Lifecycle Model [24]. The model can be used to plan activities within an organisation or consortium to ensure that all necessary stages are undertaken. While the model provides a high-level view, it should be used in conjunction with relevant reference models, frameworks, and standards to help plan activities at more granular levels.

2.2 Open Archival Information System (OAIS)

In 2005, the Consultative Committee for Space Data Systems (CCSDS), a collaboration of governmental and quasi-governmental space agencies, published the first version of a reference model for an Open Archival Information System (OAIS). The CCSDS recognised that a tremendous growth in computational power as well as in networking bandwidth and connectivity, resulted in an explosion in the number of organisations making digital information available. Along with the many advantages in the spread of digital technology in every field, this brings certain disadvantages. The rapid obsolescence of digital technologies creates considerable technical dangers. The CCSDS feels that it would be unwise to solely consider the problem from a technical standpoint. There are organisational, legal, industrial, scientific, and cultural issues to be considered as well. To ignore the problems raised by preserving digital information would inevitably lead to the loss of this information. The model establishes minimum requirements for an OAIS, along with a set of archival concepts and a common framework from which to view archival challenges. This framework can be used by organisations to understand the issues and take the proper steps to ensure long-term information preservation. The framework also provides a basis for more standardisation and, therefore, a larger market that vendors can support in meeting archival requirements.

The reference model defines an OAIS as: *“An archive, consisting of an organisation, which may be part of a larger organisation, of people and systems that has accepted the responsibility to preserve information and make it available for a designated community.”* The information in an OAIS is meant for long-term preservation, even if the OAIS itself is not permanent. Long-term is defined as being long enough to be concerned with changing technologies, and may even be indefinite. Open in OAIS refers to the standard being open, not to open access to the archive and its information. The reference model provides a full description of all roles, responsibilities and entities within an OAIS.

This section provides a quick introduction to the OAIS concepts and discusses some of the related literature required for understanding this research. It is not meant as a complete introduction to the OAIS reference model. Figure 2.3 shows the functional entities and interfaces in an OAIS. Outside of the OAIS there are producers, consumers and management. A producer can be a person or system that offers information that needs to be preserved. A consumer is a person or system that uses the OAIS to acquire information. Management is the role played by those who set the overall OAIS policy.

All transactions with the OAIS by producers and consumers, but also within some functional units of the OAIS, are done by discrete transmissions. Every transmission is performed by means of moving an Information Package. Each Information Package is a container that contains both Content Information and Preservation Description Information (PDI). The Content Information is the original target of preservation. This is a combination of the original objects and the information needed to understand the context. The PDI is the information that is specifically used for preservation of the Content Information. There are five different categories of PDI data: references, provenance data, context of the submission, fixity of the content information, and access rights.

Within the OAIS there are three different specialisations of the Information Package: the Submission Information Package (SIP), the Archival Information Package (AIP), and the Dissemination Information Package (DIP). Producers use a SIP to submit information for archival in the OAIS. Typically, the majority of a SIP is Content Information, i.e. the actual submitted material, and some PDI like the identifiers of the submitted material. Within the OAIS one or more SIPs are converted into one or more AIPs. The AIP contains a complete set of PDI for the submitted Content Information. Upon request of a consumer the OAIS provides all or a part of an AIP in the form of a DIP for using the archived information.

For performing all preservation related tasks, the OAIS has six functional entities. Since this is just a reference model it is important to note that actual OAIS-compliant implementations may have a different division of responsibilities. They may decide to combine or split some entities and functionality of the OAIS, or the OAIS may be distributed across different applications. The functional entities, as per Figure 2.3, are:

Ingest Provides the services and functions to accept SIPs and prepares the contents for storage and management within the archive. The two most important functions are the extraction of descriptive metadata from the SIP and converting a SIP into an AIP.

Archival Storage Provides the services and functions for the storage, maintenance, and retrieval of AIPs. Important functions are managing storage hierarchy, refreshing media, and error checking.

Data Management Provides the services and functions for populating, maintaining, and accessing descriptive information and administrative data. The most important function is to manage and query the database of the archive.

Administration Provides services and functions for overall operation. Important functions include the auditing of archived material, functions to monitor the archive, and establishing and maintaining of archive standards and policies.

Preservation Planning Provides services and functions for monitoring the archive. The main function is to ensure accessibility of the information in the archive.

Access Provides the services and functions that support consumers in requesting and receiving information. The most important function is to create DIPs upon consumer requests.

Even though the OAIS reference model has been regarded as the standard for building a digital archive, it has received criticism. In 2006 the CCSDS conducted a 5-year review of the OAIS reference model [26]. This review covers most of the shortcomings that are also identified in independent literature, but so far the CCSDS has not been able to successfully mitigate these.

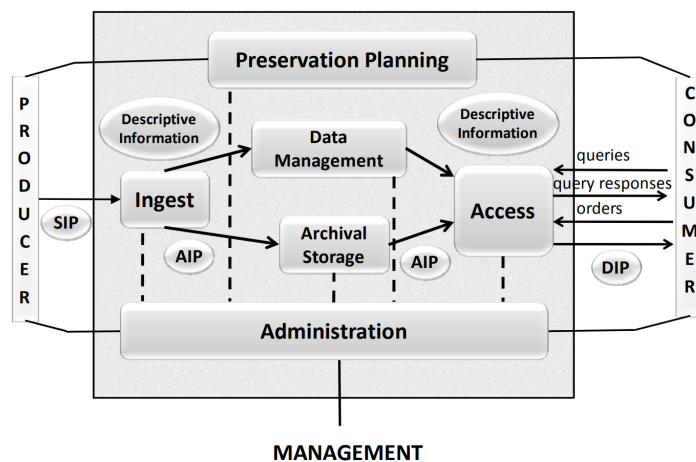


Figure 2.3: Functional entities in an OAIS [25]. The diagram shows the three users of the OAIS and how they interact with the system. The lines connecting entities (both dashed and solid) identify bi-directional communication.

One of the points in the CCSDS' review is the definition of the designated community. The user base of an OAIS is often broader than just the designated community. For example, at CERN the designated community for physics data would be the scientists at the experiments. But the data is also of interest for non-affiliated researchers and students across the globe. A second problem are the responsibilities of the designated community. The reference model forces digital preservation repositories to be selective in the material they archive. For institutions with ethical or legal mandates to serve broad populations, like national libraries, there is a fundamental mismatch between the mission of the institutes to preserve all cultural heritage and the model [27].

During the review, the CCSDS investigators found that the OAIS model has a clashing terminology mapping between PREMIS and other relevant standards that needs to be reviewed. Nicholson and Dobrevá even suggest a complete cross-mapping between the reference model and other preservation standards [28]. The main reason for this is that because of the conceptual nature of OAIS there are many ways of implementing the standard. For example, during the review of OAIS as a model for archiving digital repositories, Allinson concluded that the OAIS reference model simply demands that a repository accepts the responsibility to provide sufficient information [29]. The model does not ask that repositories structure their systems and metadata in any particular way. As a response to this shortcoming Lavoie et al. developed an OAIS compatible metadata standard [30] and Kearney et al. propose the development of special standards and interfaces for different designated communities [31].

The OAIS standard does not specify a design or an implementation. However, the CCSDS reviewers found that the model is sometimes too prescriptive and might constrain implementation. They conclude that: *“there needs to be some re-iteration that it is not necessary to implement everything.”* and that *“the OAIS seems to imply an ‘insular’ stand-alone archive”*. In his seminal article, Rethinking Digital Preservation, Wilson arrives at a similar conclusion and calls for a revision of the OAIS model. According to Wilson the revised OAIS reference should include explicit language that clearly reflects an understanding that a multi-system architecture is acceptable and that a dark archive model can be compliant [12].

According to Wilson several challenges arise when the reference model is taken too literally. It is easy to conclude that an OAIS is a single system. If this was true for the OAIS reference model, it would violate a foundational principle of digital preservation: avoid single points of failure. To avoid misinterpretations like this, a digital preservation framework would be needed. This framework could provide an interpretation of the OAIS standard and can provide fundamental building blocks for building an OAIS [12, 28]. In their 5-year review the CCSDS recognises this problem. They argue that the standard should provide supplementary documentation for full understanding. Examples include detailed checklists of the steps required for an implementation and best practice guides. Extending the standard with a stricter implementation should prevent a proliferation of supplementary standards, frameworks, and implementations – providing much needed clarity for both system designers and users.

2.3 Digital preservation systems

The late nineties saw a rapid increase in the creation and adoption of digital content. Archivists and librarians warned that we were in the midst of a digital dark age [32, 33, 34]. This initiated a debate on how to preserve our digital heritage. In 1995, the CCSDS started a digital preservation working group and began the development of a reference model for an OAIS. At the same time Stanford created LOCKS: Lots of Copies Keep Stuff Safe. The main idea behind LOCKS was that the key to keeping a file safe is to have lots of copies. LOCKS uses a peer-to-peer network for sharing copies of digital material. Libraries keep the digital materials safe in exchange for access. LOCKS was initially designed for preserving e-journals, but is used for preserving web content around the world [35].

A LOCKS network is built using LOCKSS boxes, a LOCKSS box is a server running the LOCKS daemon. Each box crawls targeted web pages and creates a preservation copy. Other LOCKSS boxes subscribe to this content and download copies. The copies are checked for defects by comparing hashes via a consensus protocol. LOCKS is an effective system for the cost effective bit preservation of web pages. However, LOCKS is very limited in the types of materials it can preserve and has no active preservation policies.

In 2003, the Florida Center For Library Automation (FCLA) began development on DAITSS, the Dark Archive In The Sunshine State [36]. In contrast to LOCKS, DAITSS uses an active preservation policy. The archive is made of two parts: DAITSS and the FCLA Digital Archive (FDA). DAITSS is a toolkit that supports the digital archiving workflow. It provides functions for ingest, data management, archival storage, and access. The FDA is the storage back end: a tape-based dark archive with a selective forward migration preservation policy [37]. Building a dark-archive saves costs on both storage and the development and maintenance of access systems.

The FDA offers two levels of preservation. For any file format the FDA ensures bit-level preservation. For a selection of popular file formats the FDA offers full digital preservation. This is ensured by creating a preservation plan for each of the supported file formats. These preservation plans describe the migration strategy for the file format and ensure long time access to the content. The FCLA has been using DAITSS in high volume production since late 2006. From late 2006 to June 2011, the FDA has held 87 TB of data consisting of 290,000 packages containing 39.1 million files with an average ingestion rate of 4-5 TB per month. In 2010, DAITSS was released to the public, but as of 2020 the repositories and website are offline. This is a result of FCLA decommissioning DAITSS and the FDA in December 2018.

Another preservation effort is SPAR, the Scalable Preservation and Archiving Repository [38]. SPAR was developed by the Bibliothèque Nationale de France and taken into production in 2006. The archive is designed to preserve a digital collection of 1.5 PB. The central concept in SPAR is a preservation track. A track is a collection of objects that require the same set of preservation policies. Each track consists of multiple channels. A channel is a collection of objects require similar treatment. Every track has a Service Level Agreement, a machine actionable document that describes the process for preserving transmissions for that track.

SPAR only guarantees bit-level preservation. The added benefit of SPAR is in the metadata management, it uses a global graph that contains all metadata of all ingested objects. This graph is modelled using a Resource Description Framework (RDF). Each ingested object has an XML-based METS file, this file is deconstructed in triplets that are added to the RDF. The resulting graph can be queried, for example: which packages have invalid HTML tables; or which packages are flagged as having a table of contents, but do not have a table of contents entry in the METS? The main problem with the metadata graph is scalability. During testing the researchers found that the RDF could handle approximately 2 billion triples, but only a single channel of the collection already contains 1 billion triples.

Around this time, more and more archives started looking into digital preservation. Out of necessity, the Burritt Library of the Central Connecticut State University, started investigating a digital preservation system. “*We realised the need for a digital preservation system after disc space on our network share ran out due to an abundance of uncompressed TIFF files*” [39]. The main goal of the preservation project was to store 5 TB of data in a reliable and cost-effective way. Burritt compared the costs of using an off-the-shelf digital preservation, with running their own Windows Home Server with backups to Amazons S3 file service. They found that running their own custom service was about 3 times less expensive as using the off-the-shelf preservation solution. Storing 5 TB a year using their solution costs roughly 10,000 dollars a year instead of 30,000 dollars. The final solution is quite simple: a Windows Home Server with a MySQL database and some custom scripts.

The 2000’s showed an increased interest in solving the problems of digital preservation. Many institutes started to develop tools and systems to help preserve our digital heritage. A problem with this approach was that the individual initiatives were not coordinated and people often reinvented the wheel. In 2006, a consortium of national libraries and archives, leading research universities and technology companies, co-funded by the European Union under the Sixth Framework Programme, started a project called Planets: Preservation and Long-term Access through Networked Services. The goal of the project was to create a sustainable framework that enables long-term preservation of digital objects [40]. Planets most important deliveries are: an interoperability framework for combining preservation tools and an environment for testing [41], migration tools and emulators [42], and a method for evaluating and creating a preservation strategy [43].

In 2010, after evaluating the digital preservation market, the Planets project published a white paper [44]. The authors concluded that the market was still in its infancy, but that the engagement

of both the public and private sector was growing rapidly. Many organisations did not have a comprehensive digital preservation plan, or none at all. Budgets for digital preservation were often short-term or on a project basis. Furthermore, institutes said that standards were vital but that there were too many. Ruusalepp and Dobрева [45] came to similar conclusions after reviewing 191 digital preservation tools. A vast majority of these tools were a result of short-term research projects, often published as an open-source project without any support and incomplete documentation. However, together with an increased interest in cloud-computing and Software As A Service (SaaS), they saw a shift towards a more service-oriented model for digital preservation.

Over the last couple of years the digital preservation community has been moving towards a more holistic approach on digital preservation. One of the common criticisms on the OAIS reference model is that it is too conceptual. Practitioners have been asking for a reference architecture for preservation services. In 2013, the European commission started the 4 year E-ARK project. The goal of this project was to provide a reference implementation that integrated non-interoperable tools into a replicable and scalable common seamless workflow [46]. The project mainly focused on the transfer, dissemination and exchange of digital objects. For each of these stages of the preservation cycle, the project analysed and described use-cases and created standards, tools and recommended practices. Preservation planning and long-term preservation processes were outside the scope of the E-ARK project.

The E-ARK project developed a custom SIP, AIP and DIP specification and tools to create, convert and inspect these packages. The E-ARK project also delivered three reference implementations for integrated digital preservation solutions: RODA [47], EARK-Web [48], and ESSARch [49]. During the evaluation of the E-ARK project, participants said that the project has made a significant impact on their institutions [50]. Highlights of the project include: major savings in costs, benefits of using the EARK-Web tool, and robust common standards that can be used across Europe. The participants feel that to maintain these benefits the project needs long-term sustainability. This is achieved by publishing the E-ARK results as part of the CEF eArchiving building block. The aim of eArchiving is to provide the core specifications, software, training and knowledge to help data creators, software developers and digital archives tackle the challenge of short, medium and long-term data management [51].

In early 2010 there was another initiative to create a fully integrated, OAIS-compliant, open source archiving solution: Archivemata. Archivemata was originally developed to reduce the cost and technical complexity of deploying a comprehensive, interoperable digital curation solution that is compliant with standards and best practices [19]. Later, Archivemata was extended to support scalability, customisation, digital repository interfaces and included a format policy implementation [52]. Over the years Archivemata has extended their functionality and user-base considerably.

In 2015, the Council of Prairie and Pacific University Libraries (COPPUL) created a cloud-based preservation service based on Archivemata [53]. Users can choose between three different levels of service. All levels include hosting and training, the main difference is in the available preservation options and the size of the virtual machine used for hosting the service. The results of the pilot were mixed. Most of the participating institutes did not have a comprehensive digital preservation policy. The lack of a framework for preservation policies required the institutes to allocate more staff to the project than expected, but this was not necessarily bad. The project did allow the participants to experiment with digital preservation, without having to invest a lot upfront. To this date, COPPUL still offers the Archivemata service indicating adoption by the participating institutes.

Five collaborating university libraries in Massachusetts started a similar project. The libraries felt that digital preservation was not well understood by single institutes and that they lacked the resources to do it individually. In 2011, they formed a task force to collaboratively investigate digital preservation. By 2014, they had decided to run a pilot using Archivemata [54]. During the pilot period of 6 months, each institute used a shared Archivemata instance to focus on their own research goals, sharing their findings as they went along. The pilot did not result in a concrete preservation system: it provided the institutes an insight into how “ready” they were for digital preservation. A similar pilot in Texas resulted in the founding of the Texas Archivemata Users Group (A-Tex), a group of Texas universities that are either evaluating Archivemata or already using it. In 2018, 4 members were using Archivemata with archives ranging in size between 1 and 12 terabytes [55].

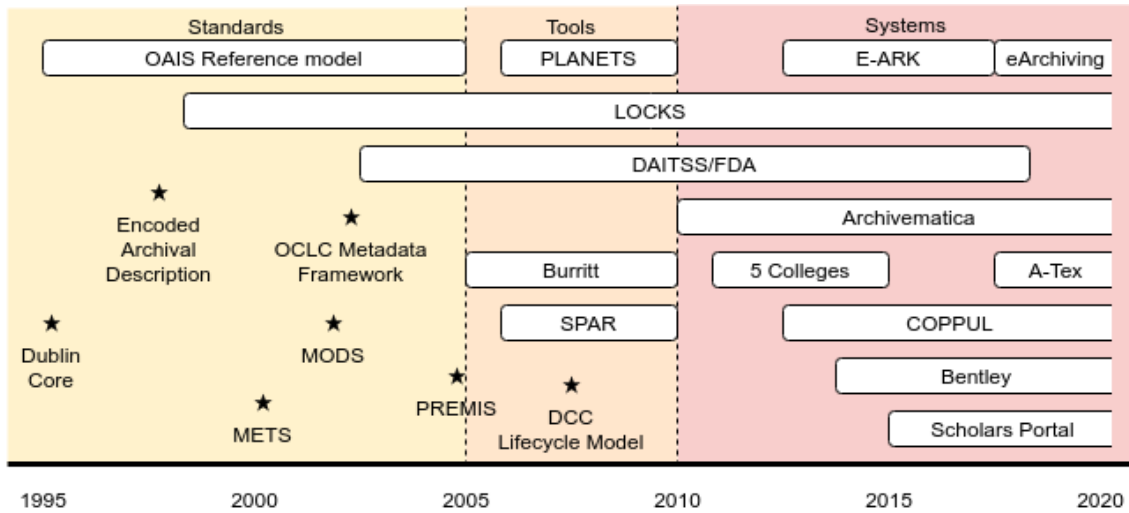


Figure 2.4: Timeline of digital preservation standards, projects, tools and systems. The ★ indicates the publication of a standard. Every bar corresponds with a longer running project that is discussed in this study. The dotted lines indicate a shift of focus in the research activities.

In 2014, the Bentley Historical Society and the University of Michigan received to create a fully integrated digital preservation workflow [56]. They selected Archivemata to be the core of the workflow. During the pilot they used Archivemata to automatically deposit 209 transfers. The archived content had a total size of 3.6 terabytes and contained 5.2 million files. The average transfer size was 7.2 gigabytes, and 6.7% of the transfers made up 99% of the total archive. Their Archivemata instance was a single virtual machine with 4 cores and 16 GB of RAM. The project was very successful, and the Bentley Historical Society is using Archivemata to the present day.

Between 2015 and 2018 Artefactual, the maintainers of Archivemata, and Scholars Portal, the information technology service provider for the 21 members of the Ontario Council of University Libraries, collaborated on integrating Dataverse and Archivemata [57]. Scholars Portal offers research data management services via Dataverse, and digital preservation services via Archivemata to their members. The Dataverse-Archivemata integration project was undertaken as a research initiative to explore how research data preservation aims might functionally be achieved using Dataverse and Archivemata together. In 2019 the integration was finished and a pilot phase started. During the pilot phase user feedback is gathered, this feedback is used to improve the integration and to contribute to the ongoing discussion surrounding best practices for preserving research data.

Looking at the development of the digital preservation field in Figure 2.4, we can clearly identify three different periods. Initially, the field was focused on understanding the problem and escaping the digital dark age. In this phase the focus was primarily directed at developing standards. After this, the focus gradually moved towards solving the identified problems. In this phase a lot of individual initiatives were started and many preservation tools and projects were developed. The third, and last, phase was less focused on solving individual problems and more on creating systems. In every step the field was gaining collective experience and the maturity of the solutions increased. One theme that is apparent in all phases is that the research is mainly focused on the what, and less on the how. More often than not, only the higher level architecture of systems is described. Performance and scalability are mentioned as important factors, but they are only mentioned and almost never qualified. This makes it hard to identify at what scale the preservation systems are evaluated and if they are suitable for large-scale digital preservation.

Chapter 3

CERN Digital Memory Platform

From the very beginning, CERN was aware of the importance of their research. During the third session of the CERN Council in 1955, Albert Picot, a Swiss politician involved in the founding of CERN, said:

“CERN is not just another laboratory. It is an institution that has been entrusted with a noble mission which it must fulfil not just for tomorrow but for the eternal history of human thought.”

The fundamental research that is performed at CERN, is to be preserved and shared with a large audience. This is one of the reasons CERN has been maintaining an extensive paper archive since the 1970s. However, with the ever growing production of digital content a new archive is needed, a digital archive.

Building a shared digital archive at CERN scale is not without challenges. CERN is a collaboration consisting of more than 17,500 researchers from 600 collaborating institutes. The research at CERN covers many aspects of physics: computing, engineering, material science, and more. A collaboration at this scale requires a diverse set of information systems to create, use, and store vast amounts of digital content. Preserving CERN’s digital heritage means that each of these systems should be able to deposit their material in the digital archive.

To provide the historical context for the CERN Digital Memory, we start with discussing the past digital preservation initiatives at CERN and the need for creating the CERN Digital Memory Platform. Next, we examine the system requirements and discuss the goals and non-goals of the platform. Finally, we introduce the high-level architecture of the CERN Digital Memory platform. We explain why we decided to use Archivematica as the core of the platform, we discuss what kind of functionality is provided by Archivematica, what functionality is not provided, and what are some of the concerns that need to be addressed.

3.1 Digital Preservation at CERN

As early as the late nineties CERN started to investigate digital preservation. In 1997 CERN started the LTEA Working Group [58]. This group was to: *“explore the size and the typology of the electronic documents at CERN [and] their current status, and prepare recommendations for an archiving policy to preserve their historical and economical value in the future.”*

The main recommendations of the working group included: selective archiving of e-mail, archiving of the CERN Web, defining a document management plan, and prevent the loss of information due to format migration or otherwise. The working group decided to postpone the creation of a digital archive. At the time the operational costs were too high, but it was expected that the costs would rapidly decrease in the near future.

In 2009, CERN and other laboratories instituted the Data Preservation in High Energy Physics (DPHEP) collaboration. The main goal of this collaboration was to coordinate the effort of the laboratories to ensure data preservation according to the FAIR principle [59]. The FAIR data

principles state that data should be: Findable, Accessible, Interoperable and Reproducible. This collaboration led to several initiatives to preserve high energy physics data. Examples include: CERN Open Data for publishing datasets [60], CERN Analysis Preservation for preserving physics data and analysis [61], and REANA for creating reusable research data analysis workflows [62].

CERN's most recent project is the Digital Memory Project [6]. This project contains an effort to digitise and index all of CERN's analogue multimedia carriers. The main goal of this effort is to save the material from deterioration and to create open access to a large audience by uploading all digitised material to the CERN Document Server (CDS). CDS is a digital repository used for providing open access to articles, reports and multimedia in High Energy Physics. Together with CERN Open Data, CERN Analysis Preservation, REANA and, Zenodo CDS is one of many efforts of CERN to build digital repositories to facilitate open science.

The original CERN convention already stated that: *"the results of [CERN's] experimental and theoretical work shall be published or otherwise made generally available"* [63]. But in the spirit of Picot's words: sharing the material today is not enough, it needs to be available for the eternal history of human thought. Previous preservation efforts have mainly been focused on identifying valuable digital material and bit prevention. The LTEA and DPHEP projects recommended bit preservation for high energy physics data. In the case of widely used, well documented physics datasets this might be sufficient. But bit preservation only ensures that the actual data stays intact, it does not preserve the meaning of the data.

Each of these preservation efforts have helped to identify a large amount of digital artefacts that need to be preserved for future generations, but with no clear plans to achieve this. One fundamental question which remains unanswered is how to preserve digital artefacts for future generations? The longer CERN waits to answer this question, the longer the preservation backlog gets. This increases the initial costs and effort of creating a digital archive, and more importantly, increases the risk of losing content forever. To solve the problem for all preservation efforts and the numerous information systems within CERN a catch-all digital archiving solution is needed. Building an institutional archive, will allow each information system to preserve relevant content with minimum effort.

3.2 Requirements

It is CERN's public duty to share the results of their experimental and theoretical work, this requires trustworthy digital repositories. A trustworthy repository has *"a mission to provide reliable, long-term access to managed digital resources to its designated community, now and into the future"* [64]. Part of a trustworthy digital repository is an OAIS compliant preservation system. Part of the trust and confidence in these systems is provided by using accepted standards and practices. To provide a similar level of trust and confidence for CERN's digital repositories, the Digital Memory Platform should, wherever possible, be based on accepted standards and practices.

The key to the design of any digital preservation system is that the information it contains must remain accessible over a long period of time. No manufacturer for hardware or software can be reasonably expected to design a system that can offer eternal preservation. This means that any digital preservation platform must anticipate failure and obsolescence of hardware and software. As a result, the Digital Memory Platform as a whole should not have a single point of failure, should support rolling upgrades of hardware and software, and should monitor and verify the viability of the preserved material.

The main focus of the Digital Memory Platform is on reducing the long-term fragility of preserved material. Central to this is an active migration policy. This means that the platform should monitor all file-formats in the archive for obsolescence and apply preservation actions where necessary. The dissemination activities, as described in the OAIS reference model, are outside the immediate scope of the platform. The material is primarily made available to the designated community via the original information systems.

CERN has to preserve the contents of an extensive digitisation project – comprising of photos, videos and audio tapes – as well as born digital content from different information systems such as the CERN Document Server, CERN Open Data, Inspire, and Zenodo [65]. This large variety in information systems and possible types of material, requires the archiving platform to have no

restrictions on accepted file formats. Since each of the connected systems has different requirements for the structure of their data, it is vital that the platform is non-restrictive in the organisation of submitted material. The platform should support versioning and incremental updates of complex AIPs. This is needed for:

- updating the metadata of collections;
- adding, removing or updating single files in large collections;
- partial access to really large datasets;
- and for versioning and deduplication of (research) data.

On top of this, the platform should provide a convenient way for existing and future information systems to submit their content for preservation. This convenience is very important. CERN is a high-energy physics laboratory first, this means that the laboratory and its members will always prioritise research over preservation. In addition to this, users and information system owners are not necessarily familiar with digital preservation concepts. This means that the Digital Memory platform should offer both a simple flexible interface and a high degree of automation.

To allow the integration of the Digital Memory Platform with a variety of workflows, it should be possible for the owners of information systems and archive managers to configure the preservation process. A non-exhaustive list of possible configuration options: encryption, access control lists, interval of fixity checking, and archiving of soft-linked files. It is also important that archive managers can monitor the preservation system. This requires the system to collect and expose important metrics. A non-exhaustive list of possible metrics: number of successful and failed transfers in a given week, the status of a particular transfer, and overall system utilisation. The Digital Memory Platform should also offer scalability in the number of services that can be connected. Currently, there is a handful of information system owners interested in using the platform. In the future, all of CERN's information systems are expected to use the service to preserve their material.

An increase in the number of connected services, together with an expected increase in data production, means that both the storage capacity and the ingestion capacity of the CERN Digital Memory Platform need to be scalable. Currently, for services like the CERN Document Server and Zenodo, the platform is expected to achieve an ingest rate of 1.5 MB/s per connected service [66]. This ingest rate allows for processing approximately 130 gigabytes of content per month, or 47 terabytes per year – almost twice the size of a median archive [7]. This ingestion rate is required to process the backlog of historic data in reasonable time, while at the same time preparing for the expected growth in incoming data.

For example, in 2019 users uploaded 23 terabytes of data to Zenodo. Assuming the same volume for 2020 and an ingest rate of 1.5 MB/s, the Digital Memory Platform can ingest both Zenodo's 2020 content and the backlog of 2019 by the end of 2020. This means that archiving Zenodo's entire backlog and all future data is a multi-year project. Unless, of course, the ingest speed could be increased significantly. These higher ingest speeds are definitely required for other projects. For example, CERN's Digital Memory Project is expected to deliver 1.4 petabytes of digitised content. At a rate of 1.5 MB/s, ingestion of this material takes 30 years. If we manage to increase the ingest rate to 15 MB/s, this backlog is reduced to only 3 years. CERN Open Data poses an even greater challenge. CERN is planning to release about 500 terabytes open-access data per month, archiving only the newly released datasets already requires an ingest rate of almost 200 MB/s.

Finally, the selected solution should fit with CERN's IT provisioning strategy. In 2019, CERN announced the MALT project¹. Within this project, CERN seeks open software solutions and products with simple exit strategies and low switching costs. The project's principles of engagement are: to deliver the same service to every category of CERN users, to avoid vendor lock-in so as to decrease risk and dependency, to keep ownership of CERN's data, and to serve common use-cases. The general strategy for MALT is to use free and open-source software (FOSS) where possible. This means that the CERN Digital Memory Platform should prioritise choose open-source solutions over enterprise or hosted services where possible.

¹ <https://malt.web.cern.ch>

In summary, the CERN Digital Memory Platform should be a centralised service that researchers, archivists and repository owners use to preserve their data in compliance with CERN’s archiving practices. The platform specifically aims to preserve data for future generations, access to content is provided upstream via the numerous information systems and portals CERN has at their disposal. As a result, the only responsibilities of the archiving service are to ingest material that is offered by connected services and to serve as an AIP store. This means that the Digital Memory Platform should:

1. provide reliable OAIS compliant AIP storage facilities;
2. provide simple interfaces for depositing transfers regardless of file format;
3. provide monitoring and status information to information systems;
4. easily accept new information systems;
5. easily increase transfer speeds and storage capacity;
6. support automated ingest of submitted transfers;
7. support the deposit of complex and versioned objects;
8. support the verification of transfer content, compliance and safety;
9. support the configuration of multiple archive workflows.

More specifically, the Digital Memory Platform should not:

1. identify and select relevant digital objects;
2. create metadata for transfers;
3. provide large-scale public access to content;
4. provide an alternative to back-ups.

3.3 Building an OAIS compliant archive service

The core task of the Digital Memory Platform is to create and store Archival Information Packages (AIPs). One of the main requirements for the platform is that it must be based on existing standards and, possibly open, tools. In Chapter 2.3 we provided an overview of existing tools and platforms. In this review we found that over the last couple of years Archivematica has been gaining market-share as the platform of choice for digital preservation projects.

We believe that Archivematica is a good fit for providing the core functionality of the Digital Memory platform. The three main reasons for choosing Archivematica are: Archivematica is OAIS compliant; Archivematica is open-source and uses open standards; and Archivematica has an active community and user-base. The former two reasons are a direct result from the requirements. The latter reason provides us with the confidence that Archivematica will be maintained in the future and can be used as a solid foundation for the platform. The active community can help us by providing best practices, existing use-cases, and possible reuse of existing initiatives to built parts of our platform.

Archivematica can be used via a graphical web interface and an API. The graphical interface offers a high level of control and feedback and is ideal for low volume processing like, for example, manual archiving. The API is more suitable for integrating Archivematica with existing systems. There are successful community examples of integrating Archivematica with digital repository frameworks such as Dataverse [57] and Islandora [67]. Given the requirements for the platform, the focus will be on integrating with Archivematica’s API, automation settings, and possibly community contributions – not using the graphical interface.

Although Archivematica is usually referred to as a whole, it is actually a collection of micro-services. The design philosophy of Archivematica is to bundle open-source preservation tools and scripts into chains of preservation actions. Each microservice is modelled as a task and applied to one of the OAIS information packages: SIP, AIP or DIP. Physically the information packages are a folder structure containing files, checksums, logs, documentation, and XML metadata. The

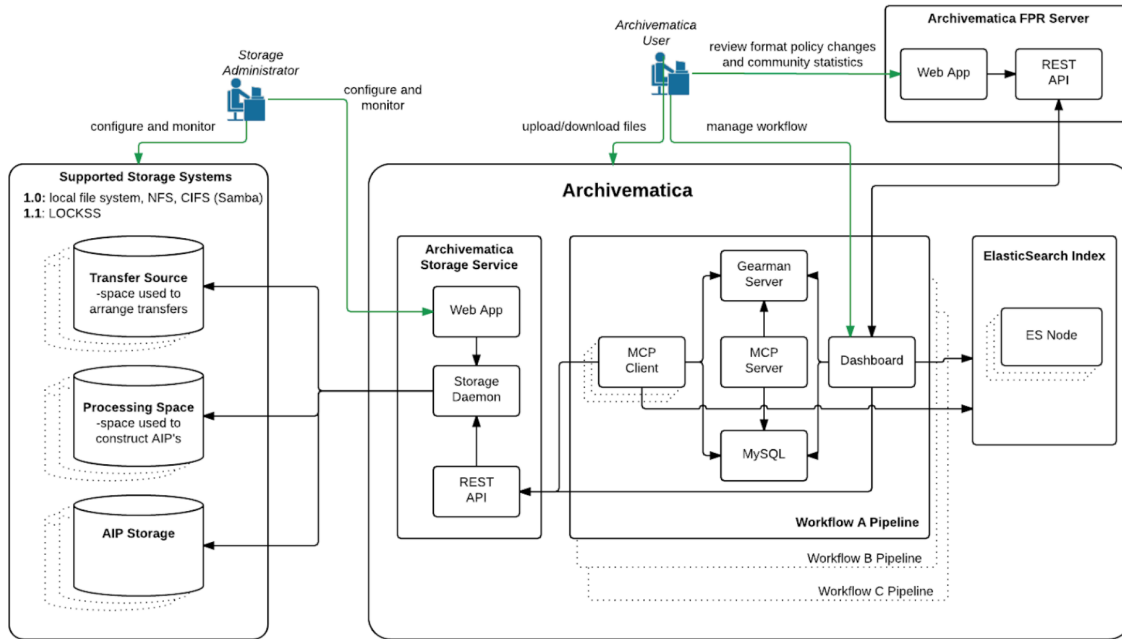


Figure 3.1: Global overview of the Archivematica components. The main components are: the Dashboard, the Storage Service, the MCP server, and the MCP client [68].

information packages are designed in a system-agnostic way, users of the packages do not need to know or understand Archivematica to use the packages.

Creating an AIP in Archivematica is a two step process. First, the user starts a transfer via the web interface or API. A transfer is a folder containing files and metadata. There are not many requirements for creating a transfer, any folder with files will do. Archivematica will apply a first series of preservation actions on the transfer, creating a SIP. This SIP contains all information submitted with the transfer, plus some extra data collected during the transfer stage. Once the SIP is complete, Archivematica moves on to the ingestion phase. This is where the SIP is ingested and converted into an AIP and possibly a DIP.

Archivematica uses a modular design. Figure 3.1 shows the major components in an Archivematica installation. The main components and their responsibilities are:

Dashboard This is the Archivematica (web) interface. The dashboard can either be used via the REST API or via the web app. It can be used to start transfers, check the state of transfers, and change the transfer settings.

Storage Service The Storage Service (SS) is an abstraction layer on top of general storage systems. The SS provides Archivematica with a general storage interface regardless of the used file systems and their location. The Storage Service also offers a service for fixity checking of AIPs.

MCP Server The MCP server is part of the Master Control Programme (MCP). The MCP server is responsible of keeping track of which micro services need to be scheduled for each active transfer. The MCP server uses the job chain to keep track of what needs to be done. Using file hooks and callbacks the MCP server keeps track of the state of each microservice. The MCP server distributes tasks using Gearman, a generic application framework to farm out work to other machines.

MCP Client The MCP client is the workhorse of any Archivematica pipeline. Actually, the MCP client is a collection of scripts and external tools running preservation microservices. An example of this would be the normalisation microservice. The MCP client receives tasks via Gearman and runs as a separate isolated processes on the Archivematica node.

Besides these core services an Archivematica server also has at least two databases: one for the storage service, and one for the dashboard and MCP services. Common installations also include

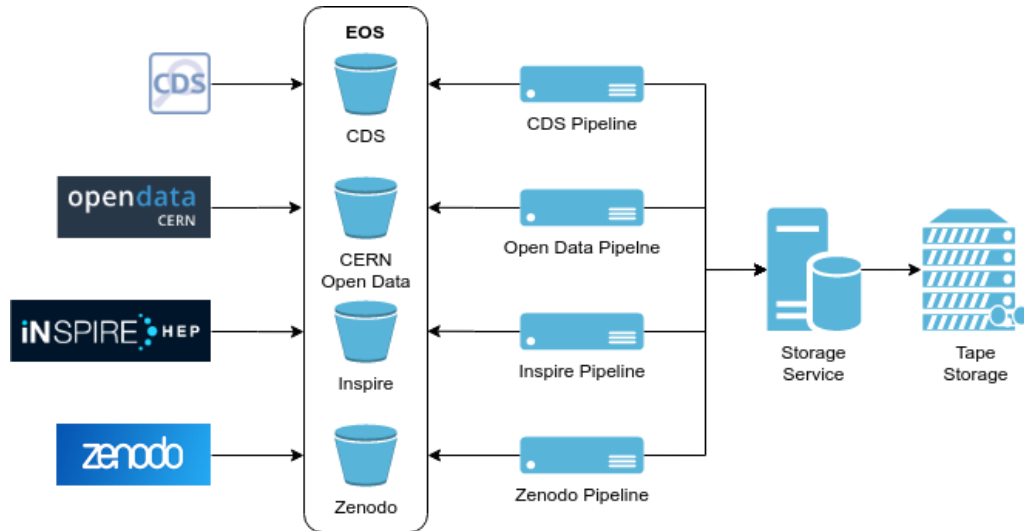


Figure 3.2: High level architecture of the Digital Memory Platform. Each information service has a dedicated pipeline, transfers are started by uploading information to a dedicated bucket. All pipelines share a common Storage Service that manages the dark archive on tape.

Elasticsearch. This is used for building indexes that can be used for querying, among other things, the AIP storage. Finally, for full functionality Archivematica requires several external services for supporting, for example, virus scanning and file-format identification.

There are two common strategies for scaling Archivematica: vertical scaling, increasing throughput by optimising a single system; and horizontal scaling, increasing throughput by adding more systems. Vertical scaling is achieved by increasing the number of MCP Clients on the system. The MCP Client is responsible for execution of the preservation actions, when the MCP Client is duplicated the system can run preservation actions in parallel. Horizontal scaling out is typically achieved by creating multiple pipelines. This means duplicating most of the core services of Archivematica: the database, the MCP server and client, and the dashboard. The Storage Service and Elasticsearch cluster can be shared between multiple pipelines.

The basic architecture of the Digital Memory Platform is similar to DAITSS [36]. The main goal of the platform is the long-term preservation of data – access is only infrequent – making it a perfect candidate for a dark archive design. Figure 3.2 shows a simplified visualisation of the Digital Memory Platform. Every connected information systems has access to a dedicated Archivematica pipeline. Each information system can submit transfers to their pipeline, for example, using a file drop location. The system monitors the file drop locations for new objects. As soon as a new object is deposited into the system, the Archivematica pipeline starts processing the submission. During this processing phase the SIP passes through a series of preservation microservices. The final product of the pipeline are the AIPs, which are uploaded to a shared dark AIP store located in CERN’s storage infrastructure.

There are several advantages to this model. First of all by using CERN’s long term storage infrastructure, the platform has access to a reliable bit-preservation service. CERN’s long term storage is designed for keeping hundreds of petabytes of high-energy physics data safe. The tape storage uses regular fixity checking to ensure data integrity and has tape rotation for ensuring data safety [69].

Using a single pipeline per service provides separation of user data and preservation policies. It also allows scaling of the service in terms of new information systems. Connecting a new information system is a two step process: creating a new drop-off location and deploying a new Archivematica pipeline. To ease the process of provisioning a new pipeline, we have created a semi-automatic deployment process. The installation and configuration of a new Archivematica deployment are automated using Puppet, an open-source software configuration management tool. This reduces the time needed for provisioning and deploying a new Archivematica pipeline from one day to less than 2 hours.

Archivematica already satisfies many of the basic requirements of the preservation platform. It allows for creating AIPs out of the box. Integration with CERN's storage infrastructure is easy: any local mounted filesystem can be used directly and adding support for other filesystem protocols is relatively straightforward. By deploying multiple pipelines Archivematica can offer separation between information services and offers an initial level of scalability. Archivematica supports an active preservation policy through the normalisation of files. Archivematica can parallelise the processing of transfers and has been successfully tested with transfers of 100,000 files and with video files of several terabytes.

Even though Archivematica seems to meet the basic requirements, there are still three challenges. First of all the size of CERN's preservation backlog. The current Archivematica projects are, to the best of our knowledge, used for managing archives in the order of 50 terabytes. With 300 petabytes, CERN's current backlog is significantly larger. This means that we need to know if we can scale Archivematica to reach the desired required of at least 1.5 MB/s, or maybe even more importantly the desired throughput of 200 MB/s.

The second challenge is handling the highly versioned research data. A very effective way of shrinking the backlog is simply reducing the amount of material that needs to be preserved. This means that we need to find a way to use Archivematica for preserving multiple copies of research data, without duplicating the data.

The third challenge is integrating Archivematica into existing workflows. The Digital Memory platform is expected to transfer terabytes of material automatically. One of the primary methods of interacting with Archivematica is the user focused dashboard. Archivematica offers some settings to automate user decisions and a REST API for starting transfers. But it is important to evaluate if the API exposes enough functionality to use Archivematica as the core of the digital memory platform.

Most likely this list can be extended. However, a positive evaluation for these three challenges is a prerequisite for selecting Archivematica as the core of the CERN Digital Memory platform. This means that need to investigate if we Archivematica can achieve the desired throughput (Chapter 4 and Chapter 5). We need to investigate and evaluate a strategy for archiving highly versioned data (Chapter 6.1). And we need to evaluate different options for automating the digital preservation process and integrating Archivematica into different workflows (Chapter 7).

Chapter 4

Vertical scaling

One of the fundamental problems to running a central archiving service is to ensure that the system can handle CERN scale. All Archivematica projects that we are aware of have archive sizes that vary between 10 and 100 terabytes. In these projects, Archivematica is typically deployed on a single server. To get an understanding of the transfer throughput we can expect from Archivematica, we start with measuring the base-line performance using a vanilla Archivematica deployment on a standard virtual machine. After this, we apply common Archivematica scaling strategies to increase the transfer throughput. The goal of this experiment is to identify potential bottlenecks and to find the maximum achievable transfer throughput for a standard Archivematica deployment.

Next, we investigate the influence of I/O performance on the transfer throughput. For workflow orchestration and certain preservation actions, Archivematica relies heavily on the local disk. During ingest Archivematica's storage space demands can reach up to four times the original transfer size. Because of this the local storage on CERN's virtual machines is not of sufficient size for simultaneously processing multiple large transfers. To solve this problem we need to investigate if the different remote storage systems offered at CERN can be used to increase the transfer capacity. During this investigation we compare the raw performance of different remote storage systems and examine the impact of using these systems on the transfer throughput. Finally, we evaluate Archivematica for a typical future preservation scenario: digitised photos. For this experiment we use photos from the CERN Digital Memory project. We measure the transfer throughput for the sample transfers and compare this to the required transfer throughput for the CERN Digital Memory Platform.

4.1 Archivematica Performance

The first step to scaling Archivematica is establishing the base-line performance. The base-line performance is measured by deploying an unmodified Archivematica 1.9 installation on different virtual machines in the CERN cloud, a private OpenStack cloud with 13,000 servers and 224,000 cores. The base-line performance is measured on two different virtual machine flavours: a standard 4-core 8GB and a larger 8-core 16 GB virtual machine. These flavours correspond to the minimum and recommended production configurations for Archivematica [70].

The base-line performance experiments use a straightforward digital preservation scenario: one of the Archivematica performance acceptance tests. The selected test comprises the ingestion of an album containing 114 images with a total capacity of 1.9 gigabytes. Archivematica is configured to ingest all photos and generate a single AIP. During the ingest phase Archivematica performs basic preservation action such as virus scanning, file-format identification, structural-metadata generation, and metadata ingest. To keep the scenario as simple as possible, the album only contains TIFF images. TIFF is Archivematica's default file-format for preserving images, this means that that no additional normalisation is required.

During the experiments we found that the default virtual machine, 4 cores and 8 GB RAM, has insufficient capacity to reliably ingest the test transfer. Transfers fail on a regular basis because

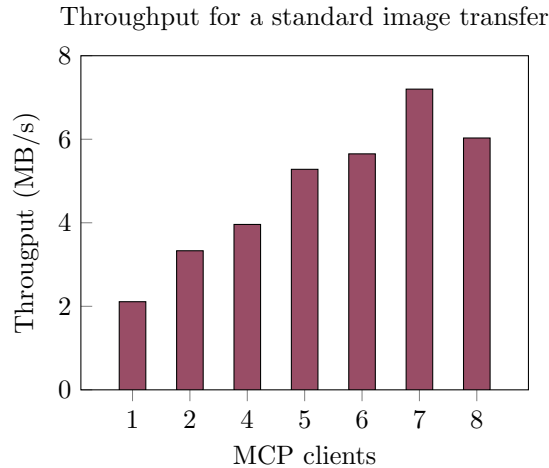


Figure 4.1: Average transfer throughput in Archivematica using an increasing number of MCP Clients.

requests to internal services time out or the virtual machine runs out of memory and essential services are killed. On a larger virtual machine, 8 cores and 16 GB RAM, the test transfers finish without any problems. However, the larger virtual machine turns out to be underutilised by Archivematica. This means that the load on the system should be increased. In Archivematica the load on the system can be increased by deploying multiple MCP clients and possibly combining this with batch-size optimisation [71].

In an Archivematica pipeline, the MCP client applies preservation actions on batches of files. By processing files in batches Archivematica reduces the coordination overhead between the MCP server and MCP clients. For each step in the workflow, the MCP server submits one job per batch to Gearman, Archivematica’s workload distribution framework. The Gearman job server distributes the jobs over the available Gearman workers. Each MCP client presents itself as a Gearman worker to the Gearman server. If a system has multiple MCP clients, Gearman has multiple workers to assign work to. This means that as long as the Gearman server has jobs and workers available, Archivematica can process batches in parallel.

The default batch size in Archivematica is 128 files. This default batch size is selected empirically. For a typical Archivematica installation, 128 files per batch provides a balance between reducing coordination overhead while providing an opportunity for parallelism. The optimal batch size depends on both the system configuration and the workload. For workloads with large computationally heavy files a smaller batch size might be more beneficial. Since the test transfer only contains 114 images it is processed as a single batch. This means that unless additional images are added, or the batch size is reduced, deploying more MCP clients will not result in a higher throughput. Because the Digital Memory Platform needs to be optimised for transfer throughput, and not for transfer latency, we need to increase the total workload. This is done by submitting multiple transfers per experiment. We start with submitting 1 transfer to a pipeline with 1 MCP client, and increase this gradually to 12 transfers to a pipeline with 12 MCP clients.

Figure 4.1 shows the results of this experiment. The results clearly show a significant increase in the average throughput. As expected, the scaling is far from linear. The main reason for this is resource competition between the different Archivematica components. For example, when MCP clients claim resources required by a shared service, such as the storage service, requests to the service might start timing out. The MCP clients also compete with each other. Each MCP client runs a variety of scripts and command line tools. As a result of this, the resources required by an MCP client may vary greatly over time. Depending on the task this can be anything from changing the name of a file, to performing a multi-threaded video conversion. On an 8-core system with 8 MCP clients the number of MCP client threads can, depending on the workload, easily vary between 8 and 64.

One way to limit resource competition is to change the process priorities of Archivematica processes. Reducing the priority of MCP client processes in favour of other user-level processes, such as the

storage service, ensures that one MCP client cannot block the requests of other MCP clients by claiming resources that are required to run the shared services. However, changing process priorities only helps as long as the load on the system is within reasonable bounds. Once the number of MCP clients exceeds the number of available cores, the system becomes highly unstable. The number of timed out requests increases significantly, frequent out-of-memory errors cause critical processes to be killed, and the virtual machine starts to run out of disk space.

On an overloaded system, timeouts and out-of-memory errors are to be expected. Running more than 8 computationally heavy MCP clients on an 8-core system is not particularly sensible. The more concerning problem is virtual machines running out of disk space when only transferring 20 gigabytes of data. At CERN the virtual machines are optimised for computational power. This means that, depending on the selected flavour, a virtual machine has 40 or 80 gigabytes of local disk space. Depending on the workload, Archivematica requires up to 20 gigabytes plus 4 times the transfer size as processing space [70]. On a system with 8 MCP clients, this means that the transfer sizes is limited to only 2.5 gigabytes per transfer. This limit is far less than can be reasonably expected from a large-scale digital archiving system.

4.2 Storage Scaling

Every Archivematica installation has a working directory that is used for preservation actions, e.g. file analysis and normalisation, and for workflow orchestration. An Archivematica pipeline with 80 gigabytes of raw storage, the standard size for virtual machines in the CERN cloud, has a usable transfer capacity of approximately 15 gigabytes. This means that for Archivematica to be usable in a high-volume setting, the working directory needs to be hosted on larger external storage. The CERN cloud has four different platforms for offering large scale storage: S3, CASTOR, EOS and NFS based storage [72].

Currently, Archivematica's S3 support is limited to transfer and AIP storage locations only. This means that an S3 file system cannot be used for hosting the working directory. CASTOR, and its successor the CERN Tape Archive, are primarily designed for long-term storage of data on tape [2]. This makes CASTOR and the CERN Tape Archive perfect candidates for AIP storage, but unsuitable for hosting the working directory. This means that the options that are left are EOS and NFS based storage services.

EOS is an open source distributed disk storage system developed at CERN [73]. EOS is designed for low-latency analysis of experimental data for LHC and non-LHC experiments. The EOS cluster offers 250 petabytes of raw storage in a distributed environment using tens of thousands of (heterogeneous) hard drives. EOS offers CERN major improvements compared to past storage solutions by allowing quick changes in the quality of service of different storage pools. This enables the CERN data centre to quickly meet the changing performance and reliability requirements of the CERN experiments with minimal data movement. Next to EOS, CERN also offers two different NFS options using OpenStack's Cinder and Manilla [74]. Cinder is a block storage service for OpenStack. Manilla is used to integrate a Ceph cluster. This provides users with CephFS, a POSIX compliant, high-throughput, low-latency file system designed for HPC applications [75].

To evaluate the performance of the different storage systems, we compare the systems using a synthetic benchmark and an Archivematica workload. For the synthetic benchmarks we use Fio [76], a storage benchmark tool for creating synthetic workloads. Fio is used for evaluating the sequential read, sequential write, random read, and mixed random read and write performance of the storage systems. To measure the effect of having multiple I/O processes using the same storage location, the Fio benchmarks run 1–8 simultaneous jobs where each job is an I/O process that is reading or writing data. The selected Archivematica workload is the same as in Chapter 4.1: multiple transfers containing 1.9 gigabytes of images each.

Figure 4.2 shows the sequential read and write performance for the different storage platforms. For each platform the average throughput is measured while running 1 to 8 Fio jobs, each job reading or writing 5.4 GB of data. First, it is important to notice that the OpenStack volumes are limited to a throughput of 2 MB/s. This is because the volumes are limited to 500 IO operations per second. Since the Fio benchmarks use a 4 KB blocksize, the maximum achievable throughput is 2 MB/s. For operations that use a larger block size, the throughput of the volume can increase to a

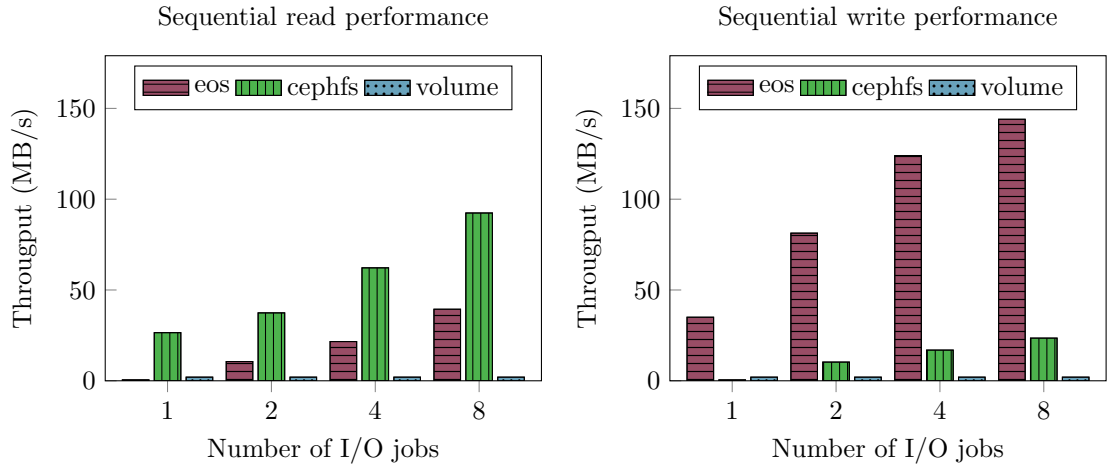


Figure 4.2: Sequential read and write performance for different storage systems. Each job writes or reads 5.4 GB using asynchronous I/O.

maximum of 120 MB/s. The read and write performance for EOS and CephFS is more promising. Both platforms show higher read and write speeds when increasing the number of simultaneous Fio jobs. This means that we do not expect I/O throughput to become an immediate bottleneck when increasing the number of I/O dependent processes on the same virtual machine. The experiment clearly shows the different purpose for which EOS and CephFS are designed: EOS is optimised for writing large amounts of real-time experimental data, while Ceph is optimised for general purpose computing. Since general purpose workloads typically require more reading than writing, it makes sense to optimise a general purpose file system for reading.

Figure 4.3 shows the throughput for the random read and the combined random read and write test. The random read/write experiment uses a 75/25 read to write ratio. The experiment clearly shows that CephFS has a superior random read performance compared to EOS and the volume. The mixed read and write performance for CephFS is significantly better than the mixed read and write performance for EOS and comparable to the volumes. Random read and write performance on EOS turned out to be highly variable during the experiment: the overall throughput was below 100 KB/s. The most likely reason for this is that EOS is not designed as a low-latency high-performance POSIX compliant file system [77, 78].

To get an understanding of the correlation between I/O performance and Archivematica’s throughput, we run the base-line performance experiment from Chapter 4.1 but with the working directory

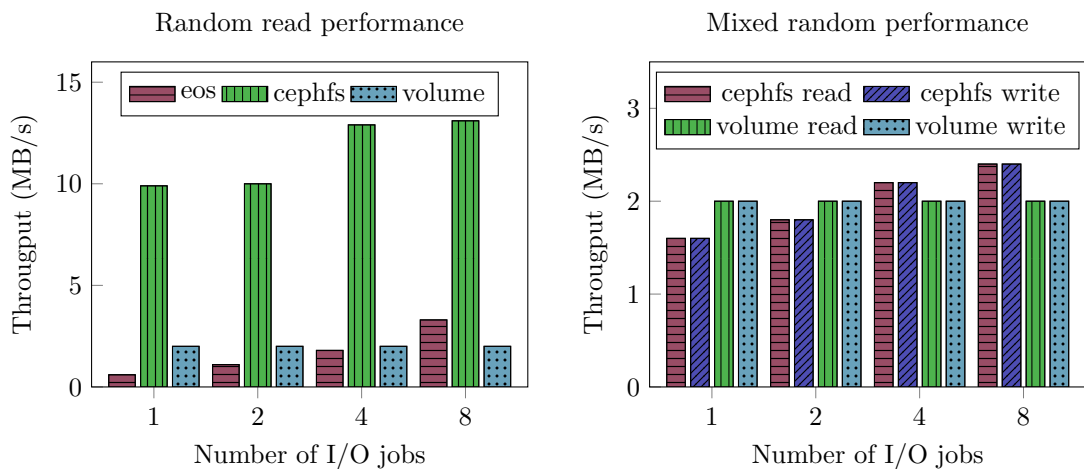


Figure 4.3: Random read and write performance for different storage systems. Each job writes or reads 5.4GB using asynchronous I/O. The mixed random performance benchmark uses a 75/25 read/write ratio.

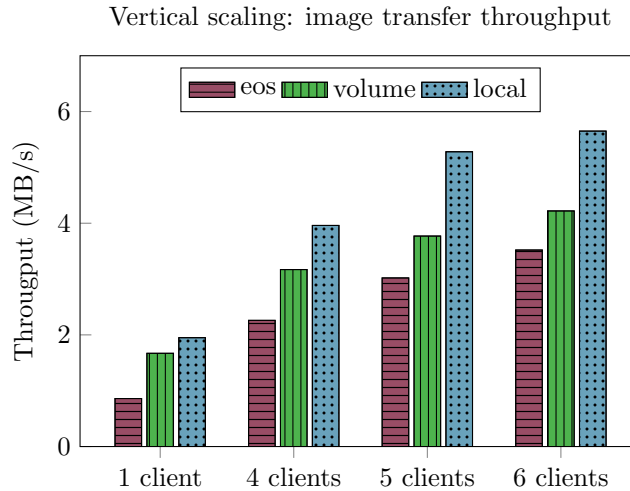


Figure 4.4: Transfer throughput for an increasing number of MCP Clients. The throughput is the average throughput while processing n transfers using n MCP clients. For each group of experiments the working directory is hosted on a different storage system.

on different storage platforms. During this experiment we did not have access to a Ceph cluster, this means that Ceph could not be considered. The results of this experiment are displayed in Figure 4.4. The experiment shows that the average throughput for an Archivemata installation has a correlation with the file system performance of different storage platforms: the throughput is higher on platforms that have a higher overall I/O throughput. Given that Archivemata has many bandwidth limited tasks that involve a lot of content movement, restructuring, and file analysis this is to be expected.

During the experiment we observed that the performance on EOS nodes varies a lot. During the synthetic benchmark we could see that read and write speeds on EOS vary between 0.084–38 MB/s and 0.029–13 MB/s. It appears that this variation in I/O throughput influences the transfer times directly. The volume backed Archivemata transfers show a more stable throughput. However, even though the performance for the volumes is consistent, there still is a large performance penalty compared to local storage. From this experiment we can conclude that Archivemata favours a high performance low latency file system. A local disk is favoured over external storage but, in cases where external storage is required, it is worthwhile to evaluate the performance of different storage systems.

Considering the overall Archivemata performance, we see that the throughput – using local storage – is significantly higher than the minimally required 1.5 MB/s is. However, this is only half the desired throughput of 15 MB/s, let alone the potentially needed 200 MB/s. But the main problem with using local storage is that it significantly limits the maximum size of transfers. This means that if Archivemata is to be used in production, the platform needs to use slower external storage. On these storage platforms the throughput is still higher than 1.5 MB/s, but not even one-third of the desired throughput of 15 MB/s.

One essential digital preservation action, that has not been considered so far, is file normalisation. In preservation systems that use an active migration policy, file normalisation is an essential part of the preservation strategy. File normalisation, e.g. converting a video or audio file, is typically a computationally intensive operation. To get an understanding of the impact of file normalisation on the transfer throughput, we measure Archivemata’s throughput for image transfers with JPEG files. Due to its lossy compression format, JPEG is not an accepted preservation standard. This means that the images need to be normalised to TIFF files, a lossless compression format and accepted preservation standard. The test consists of 175 photo albums with a total size of 11 gigabytes. Every photo album is submitted as a separated transfer to an Archivemata pipeline with 4 MCP clients. The Archivemata pipeline is deployed on a virtual machine with 8-cores and the working directory is hosted on an Openstack Volume. During ingest, Archivemata performs basic preservation actions and normalises the JPEG files to TIFF files.

The additional normalisation reduces the transfer throughput from approximately 4 MB/s without normalisation to 0.17 MB/s with normalisation. This is significantly lower than the required throughput of 1.5 MB/s. One possible solution is to run Archivematica on virtual machines with more local storage, more memory, and more cores. This would allow for deploying more MCP clients and increasing the overall throughput. However, the experiment in Chapter 4.1 already shows that on-node competition can be a significant problem. This means that there is only a limited number of MCP clients that can be deployed reliably on a single virtual machine. An additional limitation is that virtual machines are limited by the physical hardware they run on. It is undesirable to deploy a 32-core virtual machine on a 16-core physical machine. This means that to achieve the desired throughput we need to look into scaling on multiple machines, instead of scaling on more powerful machines.

Chapter 5

Horizontal scaling

To this date, the CERN Document Server and CERN PhotoLab Archives contain over 37,000 photo albums. On top of that, Zenodo contains an additional 500,000 images. Assuming an average of 5 photos per album, this amounts to a preservation backlog of nearly 620,000 images. Given that an Archivematica server can ingest approximately 100 photos in 5 minutes, this results in a preservation backlog of 3 weeks. When normalisation is required, the transfer throughput is significantly lower: approximately 100 photos per hour. This means that the backlog grows from 3 to 37 weeks. Considering that photos are only a small fraction of the entire backlog, a single Archivematica server cannot provide the required throughput for preservation at CERN scale.

In order for Archivematica to possibly achieve the required transfer throughput, we need to investigate how Archivematica can scale beyond a single virtual machine per pipeline. We start this investigation with explaining how Archivematica pipelines can be scaled horizontally by deploying a distributed Archivematica cluster with a pool of MCP clients. Next, we discuss three changes in Archivematica's scheduling and capacity management. These changes are required for efficiently coordinating multiple transfers in a high-volume setting.

Next, we evaluate the impact of the proposed changes on the ingestion throughput. We start with evaluating the ingestion throughput for image processing on a distributed Archivematica cluster. For this evaluation we use a typical future workload containing many images that require normalisation. Finally, we evaluate a similar Archivematica deployment but with a more compute intensive workload that requires the normalisation and preservation of video files. During these experiments we investigate the transfer throughput for video preservation and investigate possible ways of improving the load distribution on an Archivematica cluster.

5.1 Distributing Archivematica

The vertical scaling experiments show that the transfer throughput of a single Archivematica server is not sufficient for meeting the throughput requirements of the CERN Digital Memory Platform. A common strategy for increasing the throughput is to run multiple Archivematica pipelines. However, this is not efficient in terms of service administration, accessibility of the archived material, and the usage of computing resources,

When running multiple Archivematica pipelines, each pipeline is completely isolated from the other pipelines. In the case of completely separate archival workflows this can be an advantage. The preservation settings for an image pipeline can be entirely different from the settings for a video pipeline. People interested in the content of an archive with experimental data might not care about the content of an archive with photos from official visits. However, when processing a single collection using multiple pipelines this separation is a problem. Preservation settings need to be kept in sync across multiple pipelines, and there is no single place where all AIPs are administered and stored. Running multiple Archivematica pipelines also results in the unnecessary duplication of Archivematica resources. Simultaneous transfers only require multiple MCP clients, not multiple copies of supporting components like the MCP server. Deploying multiple copies of the supporting components requires resources that should be put towards increasing the throughput.

It is therefore safe to conclude that pipeline duplication is only a desirable strategy in cases where workflows are, or should be, completely separate. If, however, the main goal is to increase the throughput, only the number of available MCP clients needs to be increased. This raises the question: is it possible to create an Archivematica cluster with a configurable pool of worker nodes that provide additional MCP clients?

Even though Archivematica is designed as a whole, it is actually made up of four components: the dashboard, the storage service, the MCP server, and the MCP client. Each of these components is isolated from the other components: the dashboard and storage service communicate via a REST API; the dashboard sends requests to the MCP server via remote procedure calls; and the MCP server distributes tasks to the MCP client via Gearman. Gearman uses a master-worker scheme for task distribution: the MCP server submits tasks to the Gearman job server, the MCP client presents itself as a worker. This means that creating a pool of MCP clients is simply a matter of deploying extra virtual machines with additional MCP clients.

The problem with moving MCP clients to separate worker nodes is that the MCP client is not truly isolated from the MCP server. Archivematica implicitly assumes that every component has access to the same working directory. A common strategy for providing a remote MCP client access to this directory is to mount the shared directory using NFS. The disadvantage of this approach is that the node hosting the NFS connection has to process all of the I/O traffic from all remote MCP clients. This creates a bottleneck for both the remote and local MCP clients. A better solution is to host the shared directory on a distributed filesystem.

In Chapter 4.2 we discussed different remote storage systems for hosting the working directory. From the available options, only Ceph seems to be suitable: the Openstack Volumes can only be mounted on a single virtual machine, and the performance experiments show that hosting the working directory on EOS significantly lowers the transfer throughput. Looking at the I/O performance we expect that Ceph is equally, or better, suited for hosting the shared directory as the Openstack Volumes. Ceph achieves high read and write speeds for both sequential and random workloads. On top of this, the Ceph filesystem has strong consistency guarantees across nodes. This is an essential requirement for allowing Archivematica's local file system assumptions.

During the vertical scaling tests we also discovered that resource competition among the many processes on an Archivematica server causes requests between internal components to time-out. The isolation of Archivematica components, combined with hosting the shared directory on a distributed file system, offers the possibility of moving every component to a separate virtual machine. Moving the components to separate virtual machines removes resource competition between components resulting in a more stable and predictable load on the system. An additional benefit is that every virtual machine has individual monitoring, this makes it easier to discover and diagnose bottlenecks in the different components of the system.

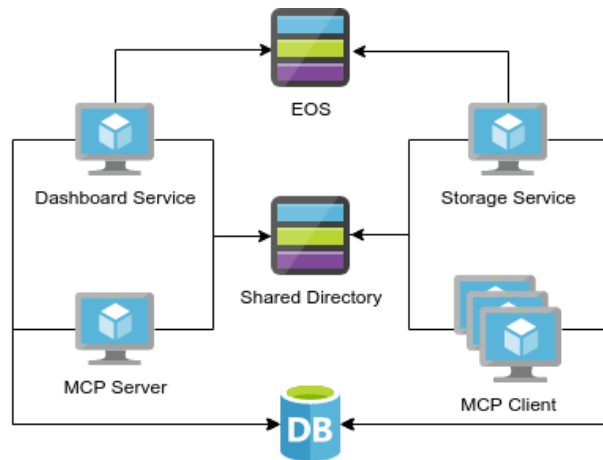


Figure 5.1: Overview of the horizontal scaling strategy. Each Archivematica component is deployed on a separate virtual machine. The transfers and the final AIPs are stored on EOS. The shared working directory is hosted on a distributed file system.

Figure 5.1 shows the configuration of a distributed Archivematica cluster. The dashboard, storage server, and MCP server are deployed on separate virtual machines. The shared database is hosted on CERN's Database On Demand service [79]. Every node in the cluster has access to the same Ceph cluster that hosts the shared directory. EOS is used for hosting transfer locations and to serve as an AIP storage location. The reason for choosing EOS as a transfer source is that every service and user at CERN already has access to EOS. The reasons for choosing EOS as AIP storage location are the raw storage capacity and the integration with the CERN Tape Archive. The transfer capacity in the Archivematica cluster is supplied by a pool of virtual machines running MCP clients. Increasing the capacity of the Archivematica cluster only requires deploying additional virtual machines to this pool.

5.2 Task management

Increasing the transfer throughput requires more than deploying an Archivematica cluster with a pool of MCP clients. An increase in transfer capacity automatically results in more active transfers. Unfortunately, there is a problem with how Archivematica handles accepting and scheduling multiple transfers in the same pipeline: every transfer that is submitted to Archivematica is automatically accepted and scheduled. In addition to this, Gearman uses a round-robin scheduling policy for all active transfers. This means that every submitted transfer not only increases the number of active transfers, but also the processing time of all existing transfers.

Solving this requires three changes in how Archivematica accepts and schedules transfers: support package queuing, add terminal links to the preservation workflow, and support rate limiting. Package queuing is developed by Artefactual, the maintainers of Archivematica, terminal links and rate limiting are developed as part of this work. The package queuing and terminal links are already released as part of Archivematica 1.11 [80], the rate limiting work will be included in a future release.

Before explaining the changes it is important to understand how the MCP server works. When archiving a submission, Archivematica performs a series of preservation actions on the submitted transfer. Each preservation action in Archivematica is performed by one of the preservation microservices. These microservices are arranged in preservation pipelines. The MCP Server schedules preservation actions from these pipelines depending on the type of submission, archiving policies, and decisions made by the archivist. This workflow is modelled as a graph of preservation actions. The workflow graph is not fully connected. A transfer in Archivematica has to go through different stages: the transfer stage, the ingest stage, and possibly the dissemination stage. Each of these stages is a separate part of the graph. Within the stages the graph is also not fully connected. Some vertices are linked through file-system events, rather than having a direct edge in the workflow graph. An example of this is the file format identification microservice: the MCP client watches the file format identification directory for new files, as soon as a preservation action moves a file to this directory the MCP clients triggers microservice.

The workflow graph defines the order of preservation actions, but not the priority. Before Archivematica 1.11, preservation actions were scheduled using a round-robin schedule. In Archivematica 1.11 this was changed by the introduction of package queuing. The main goal of package queuing is to prevent Archivematica from processing all transfers simultaneously. Instead of immediately processing every new transfer, transfers are queued until current transfers are finished. The queuing mechanism is based around packages. A package is an entire stage of the Archivematica workflow, i.e. a transfer, ingest, or dissemination package, and can either be active or inactive. The MCP server only schedules preservation actions for active packages. Once a package is completed it is deactivated and a new package can be activated. The MCP server activates packages using a weighted first-in-first-out policy. This policy favours dissemination packages over ingest packages, and ingest packages over transfer packages. Within each category, the packages are put in a first-in-first-out queue, favouring older submissions over newer submissions. This ensures that Archivematica prioritises finishing a package over starting a new package.

The problem with the package queuing implementation is the way packages are defined. In the original implementation the queuing logic contains a list of workflow vertices that mark the preservation action after which a package need to be deactivated. This creates a tight coupling between the queuing logic and the workflow definition, while at the same time introducing a

separation between the workflow and package definitions. Because of this separation it is very likely that removing or updating an edge in the workflow accidentally causes packages to be either deactivated prematurely or active indefinitely. This can be solved by moving the package definitions directly into the workflow.

In Archivematica the workflow graph is defined using a workflow language that bears many similarities to the Amazon States Language [81]. The workflow language defines a state machine in which each state represents a preservation action. Every state has zero or n transitions to other states. A transition is based on an explicit exit code of the preservation action, or in the case of an undefined exit code, a transition to a default error state. To identify the end of a package we propose to add an extra property to the vertices in the workflow graph. This property indicates that a vertex is terminal and that the package does not require further processing. The scheduler uses this property to deactivate the package and activate a new package. Moving this property to the workflow language, instead of defining the terminal states in the scheduler, simplifies inspection and verification of workflow modifications while at the same time decoupling the scheduling logic from the workflow graph.

One problem that is not solved, by either the package queuing concept or the terminal links, is the possibility to overload Archivematica with new transfers. Currently, every transfer that is submitted to Archivematica is accepted. Accepting a transfer requires a non-trivial amount of computation and disk space. Once a transfer is submitted, the MCP server creates a transfer package and downloads all the submitted content. When this is done, the transfer is moved to the working directory where it is queued until it can be processed further. When multiple transfers are submitted simultaneously, the MCP server may get blocked downloading transfers, and the shared directory is filled with queued packages causing potential storage problems. To make sure that this cannot happen we need to introduce rate limiting.

In Archivematica new transfers are submitted using a REST API call. Submitting a transfer starts an asynchronous process that downloads the submitted material, creates a package, and queues the transfer process for the package. The problem with this approach is that the API call requires a direct response. This means that before accepting a transfer, the system needs to know if there is sufficient capacity to accept a new transfer. We propose to solve this by introducing transfer

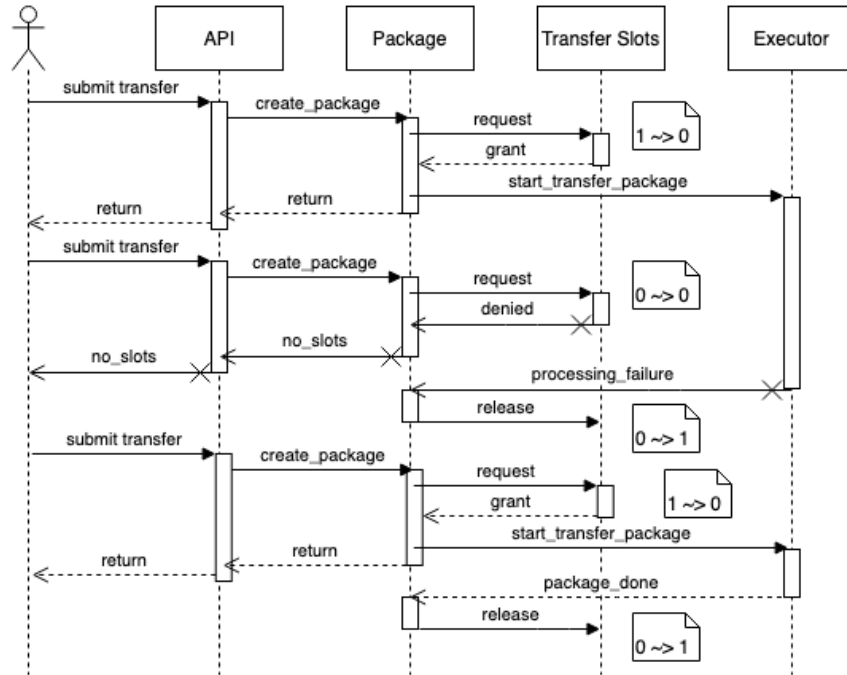


Figure 5.2: Example of a typical workflow for a pipeline with a single transfer slot. The user submits two transfers: the first transfer is accepted, the second transfer is rejected because there are no slots available. Once the transfer slot becomes available, the transfer is resubmitted and accepted.

slots, an example of a transfer slot based workflow can be found in figure 5.2.

In a transfer slot based workflow every Archivematica pipeline is assigned a fixed number of transfer slots. Every active transfer occupies one transfer slot, once all transfer slots are occupied no new transfer can be started. Since there can be many simultaneous requests to the API, each starting an asynchronous process, it is important to prevent races. This is achieved using a counting semaphore. After every API request the handler attempts to acquire the semaphore. If this fails the API returns an error, otherwise the handler starts a new asynchronous transfer process and returns a successful submission status. Once the asynchronous transfer has finished, or in case of an error, the transfer slot is released and available for new transfers.

Combining these three changes enables high volume processing of transfers in Archivematica. The introduction of package queuing and terminal links enables efficient scheduling of concurrent transfers. The introduction of rate-limiting ensures that external systems can submit transfers to the cluster without having to take the current load or any capacity restrictions of the Archivematica pipeline into consideration.

5.3 Distributed image processing

The experiment in Chapter 4.2 shows that image normalisation has a significant impact on the achievable throughput. Image normalisation is a compute intensive task that simply requires more computational power. To assess if a distributed Archivematica setup, as proposed in the previous sections, can be used for large-scale image preservation we need to evaluate the horizontal scaling behaviour.

For this experiment the transfer throughput is measured on a distributed Archivematica cluster with separate virtual machines for the dashboard, the database, the storage service, the MCP server, and the MCP client(s). Every virtual machine has 4-cores and 8 GB of RAM. The cluster is deployed with a release candidate of Archivematica 1.11 that includes the workflow scheduling changes as described in Chapter 5.2. During the experiment the size of the worker pool is varied between 2 and 5 nodes. The number of MCP clients per node is varied between 2 and 4 MCP clients. The test workload comprises multiple photo albums with a total size of 2.5 gigabytes. After normalisation the total transfer size, including the original images, has doubled to approximately 5 gigabytes. Each album in the test data is submitted as a single transfer. The number of active packages is the same as the number of MCP clients in the cluster. The number of transfer slots is half the number of available clients. This ensures that, as long as there are new albums, the MCP server is preparing new transfers for the MCP clients, but that the system won't get overloaded with queued packages.

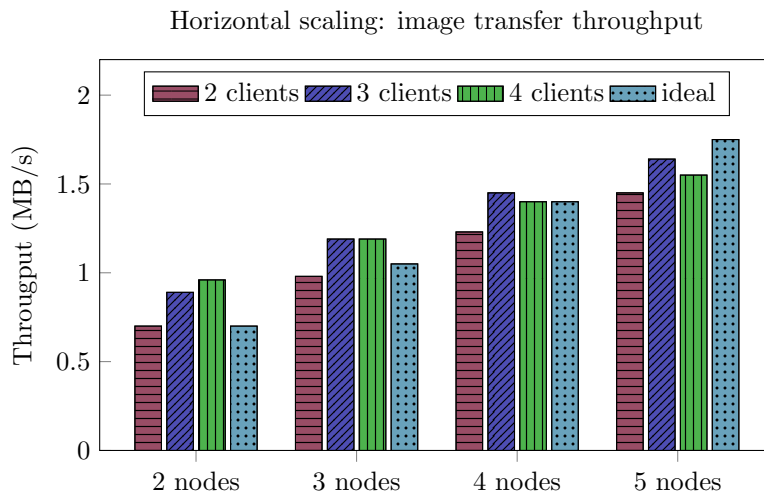


Figure 5.3: Throughput of an image transfer workload. Each image is a JPEG that needs to be normalised. The ideal scaling is the extrapolated throughput of an ingest using 2 MCP client nodes, each running 2 MCP clients.

Figure 5.3 shows the results of the image preservation experiment. The ideal scaling scenario is indicated by the extrapolated throughput of a cluster with 2 client nodes having 2 MCP clients each. The cluster exhibits typical horizontal scaling behaviour: every additional node in the worker pool increases the transfer throughput with approximately 90 percent. Increasing the number of MCP clients per node shows similar scaling behaviour as the vertical scaling experiment from Chapter 4.1.

One difference with with earlier image preservation experiments is the saturation point of the Archivematica server. In earlier experiments this was approximately 1 MCP client per available core, in this experiment the saturation point is slightly earlier. This is most likely caused by the different workload. Some preservation actions are simple single threaded processes, e.g. a script, other preservation actions are more complex multi-threaded processes. Normalisation is an example of a multi-threaded preservation action. This means that workloads that require normalisation already use more, or all, of the available resources. This is evident when looking at load on the MCP client nodes. During the experiment, a system running 1 MCP client per node had a long-term system load of 0.5. This means that, on average, the system used 50 percent of the available processing power. On a system with 2, 3, and 4 MCP clients, the long-term system load increased to 1, 1.5, and 2 respectively. This explains why the performance gain between 2 and 3 MCP client is only minimal. Since Archivematica depends heavily on I/O intensive operations, moderate over-utilisation of a system is beneficial for latency hiding. However, there is a limitation to the amount of performance that can be gained from this.

Another important factor on the throughput is tail latency. In order to achieve full overall system utilisation, every MCP client in the cluster must have transfers available. If we consider a worker pool with 5 nodes, the total number of MCP clients varies between 10 and 20. This means that for a workload of 50 transfers, the number of transfers per client varies between 5 and 2.5. The result of this is that in the case of 5 worker nodes with 4 MCP clients per node, the system is only fully utilised for 70 percent of the experiment. In cases like this, increasing the number of MCP clients does not provide any performance benefits. This will only result in over-utilisation of the client nodes in the first stages of the experiment, and under-utilisation in the final stages.

Overall, this experiment shows that a distributed Archivematica cluster can achieve the required throughput. A cluster with a worker pool of 4 or 5 nodes achieves, depending on the exact configuration, a transfer throughput of 1.4–1.7 MB/s. Using a worker pool for increasing the throughput is also significantly more efficient than deploying multiple pipelines. Compared to the throughput of a regular Archivematica deployment, an Archivematica cluster with 2 MCP client nodes achieves more than 5 times the throughput with only 2.5 times the amount of cores. When adding 2 additional nodes, the throughput is more than 8 times higher with only 3.5 times the amount of cores. We have no reason to believe that increasing the worker pool beyond 5 nodes will cause any problems. During the experiments the only systems that are fully loaded are the MCP clients nodes, the other machines in the cluster have system loads that are below 10 percent. This indicates that the throughput can simply be increased beyond 1.5 MB/s by deploying additional MCP client nodes.

5.4 Distributed video processing

As of today, CERN has almost 15,000 publicly available video, with new footage being created every day. One of the challenges in preserving videos, compared to preserving images, is simply the size. Because of the significantly larger file sizes, Archivematica spends more time on downloading the transfer, normalising the video, and compressing the final AIP. This means that preserving CERN’s public and non-public video material requires an enormous effort. For example, the CERN Digital Memory project comprises more than 5000 digitised video carriers. The lowest resolution copies have a file size of 12 GB per hour. Assuming an average length of 30 minutes, this results in a preservation backlog of 30 terabytes. Processing this backlog at a rate of 1.5 MB/s requires almost 34 weeks of continuous processing. This means that unless Archivematica can achieve a significantly higher transfer throughput, only selected videos can be preserved.

To assess if a distributed Archivematica setup can be used for large-scale video preservation, we need to evaluate the horizontal scaling behaviour. The first step is establishing the base-line throughput for video preservation. This is done by measuring the transfer throughput for a transfer containing

a 629 MB QuickTime (MOV) video file. During the transfer this video is inspected, normalised to a 4 GB MPEG file, and finally compressed. The Archivematica cluster is based on a release candidate of Archivematica 1.11 and consists of four 4-core 8 GB virtual machines. The transfer pool contains 1 node with 1 MCP client.

During the experiment the cluster achieves an average transfer throughput of 0.87 MB/s, approximately 60 percent of the required throughput of 1.5 MB/s. Adding an additional node to the transfer pool, creating a pool with 2 nodes running 1 MCP client each, increases the throughput to 1.62 MB/s. This is already sufficient to reach the minimal required 1.5 MB/s. However, considering that the backlog of the CERN Digital Memory project is already estimated at 34 weeks, it makes sense to evaluate if, and how, the throughput can be increased beyond 1.5 MB/s.

During the transfer, the MCP client node has a long-term system load of 0.5. Because of this higher overall system utilisation, when compared to image preservation, deploying additional MCP clients on the same node only marginally improves the throughput. Deploying an additional MCP client results in an increase in throughput of approximately 10 percent. Even though the throughput has increased by only 10 percent, the overall system utilisation is increased from 0.5 to approximately 0.95 – with short-term peak loads of 2.5. Deploying more than 2 MCP clients on the same node even results in an overall throughput reduction. The overhead from sharing multiple computationally heavy workloads is simply too large.

One of the disadvantages of deploying multiple MCP clients per node is the highly variable system load. This variation is caused by the diversity in MCP client tasks. For example, during the preservation of test video, the MCP client node has two obvious load spikes. These spikes coincide with the multicore friendly normalisation and compression tasks. Normalisation and AIP compression are two examples of computationally heavy and multi-threaded preservation tasks, but there are also many simple single-threaded operations. This means that if two MCP clients, sharing a node, receive a computationally heavy task the node is overloaded. However, when the same two clients receive a lighter operations, the node is underutilised. Because Archivematica distributes tasks using Gearman’s round-robin scheduling policy, both scenarios are equally likely to happen. This can be solved by moving to a load-based distributions strategy. Unfortunately, Gearman does not support alternative scheduling strategies. This means that switching to a load-based scheduling strategy requires replacing the entire job distribution mechanism.

An alternative solution for improving load distribution is MCP client specialisation. Every MCP client presents itself to Gearman as a Gearman worker with a set of supported tasks. This means that it is possible for selected MCP clients to, for example, support normalisation, while others choose not to. By removing support for computationally demanding tasks, we can create MCP light clients that only perform shorter, less demanding, tasks. These MCP light clients can be used to reduce the peak loads on worker nodes. Instead of deploying two regular MCP clients on a single MCP client node, we can deploy one regular MCP client and one MCP light client. This means that any of the many lightweight tasks can be executed by any of the MCP clients in the cluster, while the more computationally demanding tasks are executed by a smaller group of clients that do not have to compete for resources. This strategy can even be applied on a per node basis; for example, a video processing pipeline that has dedicated normalisation nodes.

To measure the impact of MCP client specialisation on the transfer throughput, we evaluate the horizontal scaling behaviour for three deployment strategies. The first strategy deploys two regular MCP clients per MCP client node. The second strategy deploys one light and one regular MCP client per MCP client node. The third strategy deploys one MCP client node with multiple light MCP clients, and multiple MCP client nodes with a single regular MCP client. The rest of the experimental setup consists of an Archivematica cluster comprising of 4-core 8 GB virtual machines with a release candidate of Archivematica 1.11. The number of transfer slots is the same as the number of MCP client nodes. The number of active packages is equal to the number of MCP clients. The workload for the experiments consists of multiple transfers, each containing a single video file of approximately 1 GB. The total workload for each experiment is proportional to the number of deployed MCP clients: 2-4 videos per MCP client.

Figure 5.4 shows the results of horizontally scaling the video archiving workload. The ideal scaling scenario is indicated by the extrapolated throughput of the base-line performance experiment. This experiment showed that adding one additional node to the worker pool increases the throughput

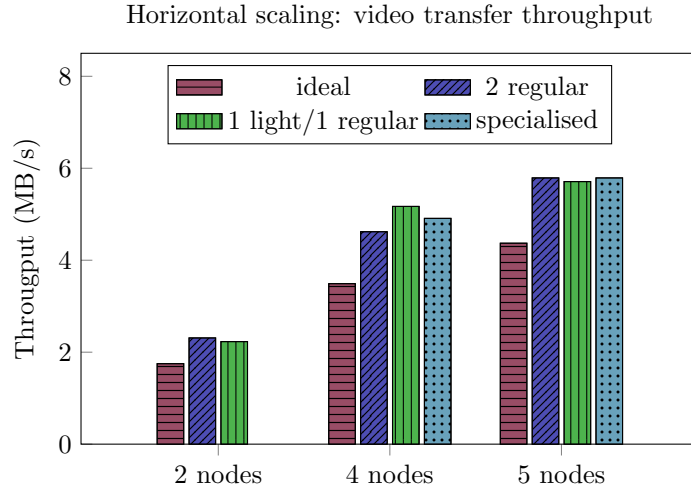


Figure 5.4: Throughput of video transfers. Each transfer is a video file of approximately 1 GB that needs to be normalised. The ideal scaling is the extrapolated throughput of a distributed Archivematica cluster with a single MCP client.

from 0.87 MB/s to 1.62 MB/s, a scaling factor of 0.93. Adding an additional MCP client to each of the 2 MCP nodes, increases the throughput with approximately 30 percent to 2.31 MB/s. Doubling the size of the transfer pool from 2 to 4 nodes, doubles the throughput. Adding a fifth node to the worker pool increases the throughput with an additional 25 percent. At this scale, the throughput scales approximately linear with respect to the number of MCP client nodes.

During the transfer, the client nodes alternate between being severely overloaded and under-utilised. Likewise to the earlier experiments, the load spikes coincide with the normalisation and compression tasks. The alternating load is caused by the approximately equal transfer sizes. Because every transfer has a similar size, every transfer requires the same preservation action at approximately the same time, this causes overloading during the computationally heavy tasks and under-utilisation during the more trivial tasks. The “1 light/1 regular” MCP client deployment strategy is supposed to prevent this from happening. Every client node only has 1 MCP client that can run resource demanding tasks such as video encoding. One possible disadvantage of this strategy is that sometimes all transfers are waiting for the computationally heavy tasks at the same time. If these tasks are not using all available resources on the node, e.g. because of I/O latency or a non-parallel sections in the task, the system is effectively wasting cycles.

The horizontal scaling of this hybrid approach seems to be slightly better than the scaling of the “2 regular” strategy. Increasing the size of the worker pool from 2 to 4 nodes increases the throughput from 2.23 MB/s to 5.17 MB/s, a scaling factor of more than 1. This is most likely caused by the improved load distribution. A comparison of the load distribution between the two strategies can be found in Figure 5.5. This figure shows the 1 minute and 15 minute load averages of one of the nodes in the worker pool. The node running two regular MCP clients shows load spikes of more than 2.5, meaning that there is an average of 2.5 active or waiting threads per core, and a sustained load of 1.5–2. The system with one light and one regular MCP client, shows load spikes of 1.5 and a sustained load of slightly approximately 1.1. This shows that the hybrid deployment strategy ensures that each client node can perform multiple tasks, while limiting resource competition.

When increasing the worker pool from 4 to 5 nodes, the advantage of the “1 light/1 regular” approach seems to fade away. When inspecting the load on each of the components in the Archivematica cluster we find that for this worker pool size there is a significant tail latency. For the majority of the MCP clients in the worker pool there are long periods of almost no activity, while other MCP clients perform small tasks. One possible explanation for this is that the cluster might not have enough MCP clients for efficiently processing the many small tasks after normalisation. Another reason can be that the computationally heavy tasks from the regular MCP client steal resources required by the MCP light clients.

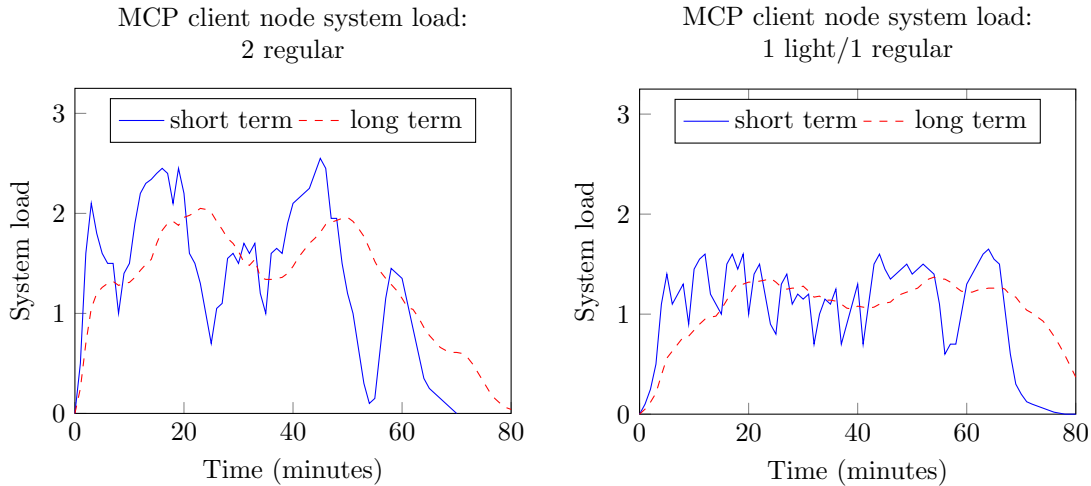


Figure 5.5: The load on two different MCP client nodes during the performance experiments. The main difference between the two nodes is the type of MCP clients that is deployed. A light MCP client is a MCP client with all multi-threaded capabilities turned off. The short term load is the average system load over 1 minute, the long term load is the average system load over 15 minutes. A load average of 1.0 means that there is an average of 1 thread per cpu core in the system.

This is an indication that the cluster may need to increase the priority and processing capacity for the smaller tasks. We propose to do this by separating the regular MCP clients from the MCP light clients. Instead of equipping each MCP client node with 1 light and 1 normal MCP client, each node in the worker pool will either run 1 client dedicated to compute intensive tasks, or multiple lighter clients. For example, in a cluster with a worker pool of 4 nodes we deploy 1 node with 4 MCP light clients, and 3 nodes that have only 1 heavy MCP client.

Figure 5.4 shows that the throughput of this strategy is between the “2 regular” and the “1 light/1 heavy” strategies. When inspecting the load distribution, it appears that the node with the MCP light clients is underutilised. The heavy MCP clients create a bottleneck causing transfers to spent a lot of time waiting to be normalised. Deploying an extra node with a heavy MCP client node solves this problem. Compared to the other strategies, the performance of this strategy is similar to the “2 regular” strategy. When looking at the overall system load in worker pool, it is approximately equal across nodes. The system utilisation is especially constant halfway the experiment, when the cluster is fully loaded. This indicates that this strategy might be more efficient when used for a video archiving pipeline. At the same time, this experiment also shows that there is a risk to using specialisation strategies for load distribution: it is easy to introduce bottlenecks in your system. The optimal system configuration depends on both the system configuration and the workload. Depending on the workload tuning the MCP client capabilities might provide better performance, but at the cost of extensive system testing and tuning.

Considering the overall scaling of the video preservation throughput the results are promising. Increasing the worker pool from 1 to 5 nodes results in an increase of the transfers throughput from 0.87 MB/s to 5.79 MB/s. Increasing the number of cores in the Archivematica cluster 16 to 32 results in more than 6.5 times the throughput. The final transfer throughput is almost 4 times the required throughput of 1.5 MB/s. We have no reason to believe that increasing the worker pool beyond 5 nodes will cause any serious problems. For compute intensive workloads, such as video encoding, the main bottleneck seems to be the available compute power. Nonetheless, it is important to note that every additional MCP client increases the load on the storage service. When storing an AIP, the MCP client uploads the completed AIP to the storage service. Once the AIP is uploaded, the storage service transfers the entire AIP to EOS. If too many MCP clients upload their AIPs at the same time, the processor and network capacity of the storage server become a bottleneck. During the experiments the network traffic for the storage service already reached 450 mbps up and 800 mbps down. For this particular configuration this is most likely not a problem, every MCP client node already has access to EOS. This means that instead of uploading the AIPs to the storage service, every MCP clients upload the AIPs directly to the AIP storage location.

Chapter 6

Versioning and deduplication

One of the main differences between a data warehouse and an archive is context. Part of the historical context of research is how it evolves over time. Having access to the research at different points in time and see how it evolves can give valuable inside in how the research was conducted. But the value of context can also be very simple. Certain data analysis pipelines might only work for a particular version of a dataset or use a specific version of a script or framework. Not having access to a part of the project, can easily result in the entire project being unusable.

Research is often the result of many incremental improvements. As a result, a research project can have many versions. One problem with archiving every version of a digital object is duplication. Consider a project containing a couple of large datasets and some scripts that are used to analyse the data. After publication, some flaws in the research need to be addressed. This results in the restructuring of one data file and some changes in a script. When this project is archived naively, the majority of the datasets is duplicated. Which raises the question: how can we capture the state of an object over time, without unnecessary duplication of the components?

We propose to solve this by disconnecting the object that needs archiving from the artefacts that are stored. In the previous example, the object is the research project, and the artefacts are the datasets and the scripts. Each artefact is archived separately, and every object – i.e. a specific version of the research – is archived with references to the artefacts it comprises. When archiving a new version of the object, only the new or changed artefacts need to be archived. We decided to call this the Archive Information Collection (AIC) versioning strategy.

We start with introducing the AIC versioning strategy and explaining how the strategy fits within the OAIS model. After this, we evaluate the effectiveness of the strategy in a case-study using Zenodo. Zenodo is a platform used by researchers to share their data. In the case-study we simulate the required storage space for archiving approximately 5% of the Zenodo records. Using the results of the case-study, we compare the required storage space for an archive all strategy to the required storage space for the AIC versioning strategy and evaluate the effectiveness for different categories of content: datasets, multimedia, publications and software.

6.1 The AIC versioning strategy

The main goal of the archiving platform is to create AIPs. According to the OAIS model, an AIP is defined to provide a concise way of referring to a set of information that has all the qualities needed for permanent, or indefinite, preservation of a designated information object. The OAIS model describes the responsibilities of the AIP, not the actual implementation. An example of a specific AIP implementation is the Archivematica AIP implementation [82]. In this section we will introduce an AIP implementation that can be used for efficiently saving versioned objects. This AIP implementation is part of the AIC versioning strategy and based on Archivematica's AIP implementation. First, we discuss the requirements of the AIC versioning strategy. After that we introduce the AIP structure and discuss construction and reconstruction of objects.

In the OAIS model, an AIP is defined as to “*provide a concise way of referring to a set of information that has, in principle, all the qualities needed for permanent, or indefinite, Long Term Preservation of a designated Information Object*” [25]. This means that an AIP must be usable and meaningful without any knowledge of the system that is used to create it. For example, it is perfectly possible that Archivematica has ceased to exist by the time the AIPs it created are used. This is why the Archivematica AIP, and any other AIP implementation is system-agnostic.

The AIPs as defined in the AIC versioning strategy should provide any of the CERN services with OAIS compliant AIPs. This means that the material that is deposited can have many different formats, from a simple directory with images to large complex physics datasets. This means that the AIC versioning strategy should be able to represent objects with complex file hierarchies. As a result of the above, the generated AIPs are required to:

1. be system agnostic;
2. support versioning of deposited material;
3. support complex nested file structures.

For the first requirement we decided to use a similar model as Archivematica. The AIPs in the entire archive will be accessible via the regular filesystem and must be based on accepted archiving standards. This means that the only hard requirement for any future access to the archived material is that the filesystem needs to be accessible. In order for the contents of the archive to be usable, the files need to be usable, this is secured by the adherence to accepted archiving standards and using an active migration policy.

The main challenges are the second and third requirement. How to capture the state of a complex object over time without duplication? This is achieved by differentiating between the object that needs to be archived, and the artefacts the object comprises. Instead of creating one AIP that contains all information, there is one AIP for the metadata and structure of the object and one AIP per artefact that contains the actual artefact and the associated metadata. In the case of a research project this would mean that for every version of the project there is one AIP that contains the relevant context of the project, e.g. who worked on the project and when, and a list of all the artefacts that exist within that specific version of the project. Every artefact, e.g. a file with data or a script, is saved in a separate AIP which is referenced by at least one of the object AIPs.

This model is supported within the OAIS reference model by means of AIP specialisation. The OAIS model identifies two subtypes of the AIP: the Archive Information Collection (AIC) and the Archive Information Unit (AIU). The main difference between the AIC and the AIU is semantic. An AIC is defined as an AIP that can contain references to both AICs and AIUs. Every AIC must contain information on how the referenced AICs and AIUs relate to each other and how they can be retrieved. An AIU is defined as an AIP that can only contain artefacts. An example would be an archive containing the most watched movies of 1984 and 1985. Each movies that is referenced in either list will be archived in an AIU. Next to this there will be two AICs: one AIC will contain references to all the movies that are on the list from 1984, the other AIC will reference all the movies that are on the list from 1985.

An example of how this can be used for versioning research data can be found in figure 6.1. For every version of an object, in this case a research project, the system will create an AIC. This AIC contains a reference file and a metadata file that contains all the relevant data which is specific to the project. The reference file contains a mapping from the original path and name of every artefact in the object to a sha256 checksum of the file contents. Every artefact in the object is added to an AIU that contains the artefact itself and a metadata file that contains artefact specific context. The AIU name will be the checksum that is used to reference the AIU in the AIC.

Lets assume that a new version of this project contains a changed version of “file1”. When this project is archived, only the changed file requires a new AIU. Since the contents of the file have changed, the checksum and the resulting AIU name change. The reference file in the new AIC will contain references to the already existing AIUs for the files that haven’t changed, and a reference to the new AIU for the new file. This way the other two, potentially large, files are not duplicated while the historical context of the project is preserved.

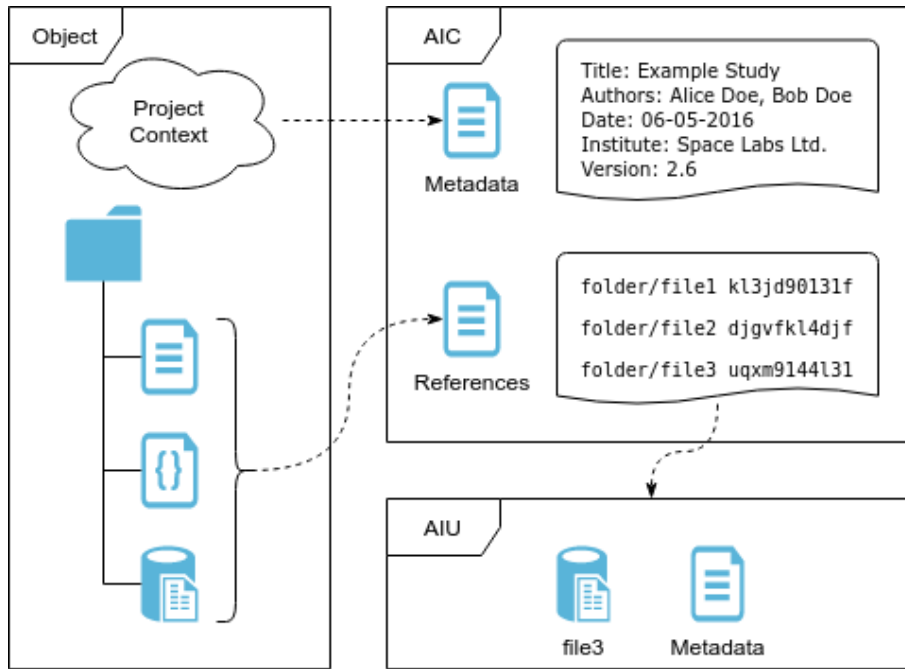


Figure 6.1: Example of how a simple project can be versioned using a combination of an AIC and multiple AIUs. The AIC contains the the structure of the project and all relevant metadata for the project. Each of the AIUs contains an artefact and the metadata that belongs to that specific artefact. The AIC and AIU are linked via the reference file in the AIC. Each line of the reference file contains the original location of the object and a checksum. This checksum is the checksum of the object and is used to uniquely identify the relevant AIU.

One of the main disadvantages of AIC versioning is in the future usage of the objects in an archive. When an entire object is preserved using a single AIP, future users of the content can easily inspect or use the preserved objects. In the case of preservation with AIC versioning, the original object needs to be rebuild before it can be used. This is where the reference file is essential. If a user wants to rebuild the project from the example in Figure 6.1, the first step is to locate the AIC and find the reference file. The next step is to find all the AIPs that are referenced in the reference file. The third, and last, step is to restore the objects structure by moving the contents of the referenced AIPs to their original location.

Using this combination of AICs and AIUs, together with the naming conventions, allows for the creation of system agnostic AIPs. There is no limit on the complexity of structure of the objects that can be archived. The AIUs allow for referencing the same (large) files across different versions of an object, allowing for deduplication of archived data. The disadvantage of AIC versioning is the increased complexity in the creation and future usage of AIPs. This makes AIC versioning a better fit for archiving workflows that rely heavily on automation and infrequent access patterns. If the AIPs are created using manual workflows, frequently accessed by users, or rarely contain duplicate data, then AIC versioning will most likely only add a lot of complexity. However, if the AIPs in the archive are created automatically, are part of a dark-archive, contain many (large) duplicate files, then AIC version will most likely save a lot on storage costs without significant impact on the complexity of the entire preservation system.

6.2 Case study: Using versioned AICs for Zenodo

In 2013 CERN and OpenAIRE launched a new online repository: Zenodo [83]. Zenodo is a platform that allows researchers to share publications and supporting data more easily. To enable researchers to publish their full research Zenodo does not impose any requirements on format or size of the digital artefacts. This can be anything from raw data and software, to images, papers, posters and more. Every record on Zenodo is identified using a digital object identifier (DOI).

This DOI can be used to identify the current version of a digital object, but also a specific past version of a digital object. This allows users to edit and update their records after they have been published, but it also enables citation of specific versions or all versions of a record.

In order to truly preserve a Zenodo record, it is essential preserve the different versions of the record over time. One problem with saving many versions of a similar record is that there might be a lot duplication. Using a versioned AIP strategy, such as described in the previous section, can be a solution to this problem. However, using this strategy adds additional complexity to the Archiving system. To decide if the AIC versioning strategy is useful for the Zenodo data, it is essential to have an idea how AIC versioning impacts data storage requirements. For example, when saving different versions of a PDF there is no file duplication and the AIC strategy will only add additional overhead. However, using the AIC versioning strategy for an archive with many datasets will have a significant impact on required storage space. To evaluate if the AIC versioning strategy is a valid option for preserving Zenodo data, we need to estimate the difference in storage space for archiving records with and without AIC versioning. This is done by comparing at the expected storage requirements of up to 10,000 of the most viewed records for every major record category in Zenodo.

The storage space for the default strategy, one AIP per record, is estimated by indexing the size of all files in each record in the sample. The storage space for the AIC versioning strategy is based on the total size of the files that have been updated or added between two versions. This size is computed by comparing the md5 checksum of the files that belong to each version of a record. The file sizes of the unique checksums in the new version are used as the estimated size of the archive. To compensate for the size of the additional AICs and AIUs an extra 50 kilobytes per file is added to compensate for each AIU, two times the nominal size of our test AICs. This comparison does not take space savings by the compression of the final AIPs or file normalisation, logging files and any other Archivematica generated files into consideration.

Table 6.1 shows the results of simulating both storage strategies for more than 70,000 Zenodo records. The records are divided into 4 different categories. The datasets and software categories directly correspond to a Zenodo category. The multimedia and publications categories are a combination of Zenodo categories. The multimedia category is a combination of records from the images (10,000) and videos (1,635) categories. The publications category is a combination of records from the articles (10,000), books (3,824), conference papers (10,000), lessons (1,470), posters (5,289), and presentations (10,000) categories. In every Zenodo category the AIC versioning strategy saves on the required data storage space. For some categories, like the books category, the amount of saved space is only enough to compensate for the overhead induced by the extra AICs and AIUs. For other categories, like the datasets categories, the saved space is almost 25%.

One of the reasons why the AIC versioning strategy is more effective for datasets than for other categories, is that research datasets are highly versioned. Almost a quarter of the datasets in the sample data is versioned, resulting in almost 6000 additional Zenodo records. But this is not the

Category	Records	Records with versions	Naive archive size (GB)	AIC archive size (GB)	Category saving (%)	Total saving (%)
Datasets	10,000	2,345	69,243	52,150	24.68	21.54
Multimedia	11,635	220	4,732	4,035	14.74	0.88
Publications	40,583	1,437	2,952	2,545	13.81	0.51
Software	10,000	5,515	2,419	2,231	7.79	0.24
Total	72,218	9,517	79,347	60,961	23.17	23.17

Table 6.1: Comparison of the archive size for the most viewed submissions in Zenodo with and without the AIC versioning strategy. The multimedia and publications categories are an aggregation of several Zenodo categories.

only reason. For example, more than half of the software records is versioned – resulting in more than 35,000 additional records – but the AIC versioning strategy only saves a mere 8 percent. At the same time the multimedia category only contains 220 versioned records, an extra 300 records, but saves almost 15 percent.

This has to do with how datasets are generally versioned. Many of the datasets uploaded to Zenodo contain multiple files. Usually, only a couple of these files change across versions. For software this is different. Most of the software is uploaded to Zenodo as a single archive, this means that every new version is a new file. In the cases where the software is not uploaded as a single archive, the space saved is still minimal: writing software is more a process of changing files, rather than adding or replacing files. For AIC versioning to be effective, there have to be duplicate files. For example, in the datasets category the number of files that has to be preserved is reduced with 34 percent when using the AIC versioning strategy, for software this is only 5 percent.

The AIC versioning strategy can also be effective in cases where there are not many versions, or only a few files. From the 10,000 image records in the multimedia category, only 155 have versions. However, these record contain large image collections with only minor changes. For these records, the AIC versioning strategy reduces the number of preserved files with 26 percent. The resulting reduction in storage space is 21 percent. At the same time, only 65 of the 1635 video records are versioned. The AIC versioning strategy reduces the number of preserved files for this category with only 4 percent. But because of the size of some video files, the size reduction for the category is still a respectable 13 percent.

When looking at the overall saving in storage, it is clear that the majority of the savings can be gained from the datasets. The saved storage space for this category is 17 terabytes, while the other categories combined only saves 1.3 terabytes. This shows that it is important to look at the structure of the data that needs to be archived, when considering the preservation strategy. However, even if the relative performance is good, e.g. the AIC versioning strategy reduces the storage needs for publications with almost 15 percent, the total saved storage space, in this example 400 GB, might not be worth it.

The Zenodo case study shows that the versioned AIC strategy can reduce the total archive size with more than 20 percent without compromising the context of the individual records. The versioned AIC strategy introduces some overhead: additional storage for the extra AICs and increased complexity of AIP creation and reconstruction. The overhead for the additional AICs is small, ingesting an AIC in Archivematica is done in less than a minute and the storage costs are negligible, less than 50 KB per AIC. The technical overhead is more significant. Both construction and reconstruction of the AIPs is more complex and requires a significant degree of automation to be usable. For this particular use-case, saving almost 20 TB of storage space for our sample data is worth this overhead, but for other use-cases this might be different.

Chapter 7

Automated job management

Traditionally, Archivematica has always put the archivist at the centre of the digital preservation workflow. Archivematica offers a great amount of control over the preservation process by asking for user input during the preparation, processing, and storing of transfers. However, when using Archivematica for processing large amounts of data a more automated workflow might be preferred. Within Archivematica there exist several options that can be used to automate the digital preservation process. The two most notable examples are the configuration files that are used for automating decisions in the preservation pipeline, and a REST API that can be used for submitting and monitoring transfers.

An important part of the Digital Memory Platform is the convenience for the end-user. This means that the user should have access to a simple interface for depositing transfers. Once a transfer is deposited, the transfer should be ingested automatically. In case of failure, the failure should be resolved by the system – or at least reported to a user or maintainer. Maintainers want a similar level of convenience. Adding new information systems, creating preservation rules, migration of AIPs, and monitoring the preservation system, all of these essential tasks should be easy and convenient.

During the design of the Archivematica service we have identified and tested three different approaches to automated job management. First, we evaluated Archivematica’s automation tools. The automation tools are developed by Artefactual to aid the process of automating, otherwise manual, workflows. Next, we directly interacted with Archivematica’s API to manage the Archivematica jobs via custom scripts. Finally, we discuss Enduro. Enduro aims to provide a tool to orchestrate reliable workflows where multi-pipeline Archivematica deployments are integrated with other systems. Enduro is a system that enables users to run failure-oblivious and durable preservation workflows.

For each of the approaches we start with discussing the design philosophy and goals. After this we reflect on how the approach can be used to automate, orchestrate, integrate, and monitor multiple Archivematica pipelines. Finally, we evaluate each approach to see if it can be used for the CERN Digital Memory Platform.

7.1 Automation tools

There are many ways in which Archivematica users have automated parts of their Archivematica workflow. The common factor in all strategies are Archivematica’s processing configuration files. A processing configuration file is used to configure decision points in a preservation pipeline. Configuring these decision points allows submissions to pass through an entire pipeline without having to wait for user interaction.

No two digital, or physical, objects are the same. These differences have a direct impact on the preservation workflow. There are obvious differences in how images, videos, and research projects need to be preserved – but there are also less obvious differences. For example, external content might need to go through a virus scan, while internal content is assumed to be safe, or

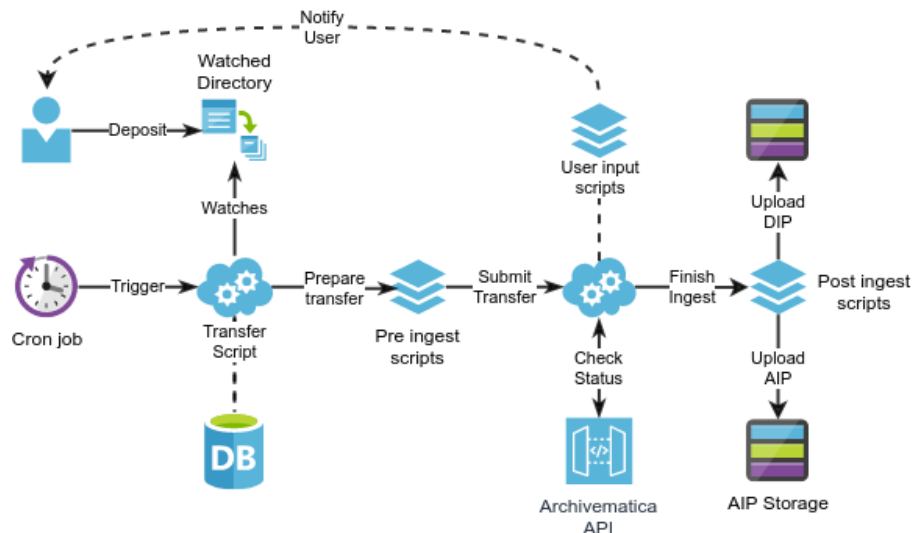


Figure 7.1: High level overview of the Archival workflow using the automation tools. The user deposits a new transfer in the watched directory. This deposit is detected by the transfer script. The transfer is prepared using the pre-ingest scripts. Once the ingest is finished a DIP and AIP are uploaded to the storage service.

material from a professional digitisation service might be normalised, while donated images require normalisation and restructuring. To handle these differences without having to deploy multiple pipelines, Archivematica supports multiple processing configurations per pipeline.

The main problem with processing configurations is that they can only be used for automating the decision points in an Archivematica pipeline. Users still need to submit and monitor the progress of their transfers. A common strategy for fully automating the preservation pipeline is to combine Archivematica’s processing configurations with Archivematica’s automation tools [84].

The automation tools are a collection of scripts. The heart and soul of the automation tools is the transfer script. The transfer script defines a state machine for processing a transfer in an Archivematica pipeline from start to finish. Besides the transfer script, the automation tools comprise three other types of scripts: pre-ingest scripts, user-input scripts and post-ingest scripts. The pre-ingest scripts are used to automate transfer preparation. Examples of pre-ingest scripts are metadata creation and copying processing configurations. The user-input scripts are used when a transfer requires user intervention. An example of a user-input script would be to send an e-mail to a user that indicates that manual intervention is required. The post-ingest scripts are used to automate the finalisation of transfers. An example of a post-ingest script is uploading the final DIPs to an access systems.

Figure 7.1 displays a typical transfer scenario using the automation tools. The transfer script is triggered via a cron job and uses a SQLite database for preserving the program state. First, the transfer script checks if there is no transfer in progress. If this is the case, the script checks a pre-configured file system location for new deposits. When a new deposit is detected, the transfer script triggers the pre-ingest scripts. After the pre-ingest phase is complete, the transfer is submitted to Archivematica. Each time the cron job triggers the transfer script during processing the status of the transfer is checked. When the transfer requires user-input, e.g. because there is no processing rule available, the user-input scripts are triggered. Once the ingest has completed, the transfer script triggers the post-ingest scripts and marks the transfer as complete.

To understand how the automation tools fit within CERN’s preservation workflows, we use the automation tools for processing 1000 small to medium sized transfers. The transfers are a random subset of Zenodo deposits, where each transfer contains a single file. The majority of the transfers is smaller than 1 megabyte, while some are up to a few hundred megabytes in size. Using the automation tools for processing these transfers revealed three major problems for using the automation tools as a basis for the CERN Digital Memory Platform.

The first and foremost problem is that the automation tools only support one active transfer. When a transfer is in progress, no new transfers are started until the active transfer has finished. The only trivial way to process transfers in parallel is to create complete copies, including the database and transfer folders, of the automation tools. By dividing the transfers across multiple watched directories, each separate instance can submit a subset of the transfers to the Archivematica pipeline.

The second problem is the interval at which the automation tools track the progress of transfers. The transfer script is triggered using a cron job. The smallest interval at which a cron job can be triggered is 1 minute. This means that, the an Archivematica pipeline is idle between the moment a transfer is finished and the next time the transfer script is triggered. In a worst case scenario, Archivematica is idle for almost a minute between finishing a transfer and starting the next one. For this particular test case, many small transfers, this results in a lot of idle time for the pipeline.

The last problem concerns the overall stability of the automation tools. During testing the automated processing was frequently interrupted because the transfer script reached an undefined state. Usually this was caused by unexpected user input events or missed transfer updates. Besides this the database is an on-disk SQLite database. When deploying a new virtual machine or migration of all processing information was lost.

Some of these problems are trivial to solve. Moving from a SQLite database to a MySQL database improved both database performance and overall system stability. Adding monitoring and solving most of the undefined state bugs can be done in a couple of days. However, managing and monitoring multiple concurrent transfers is less trivial. When taking all of these problems into consideration, it shows that in terms of scope and maturity the automation tools are not designed for high-throughput processing. The automation tools are designed for automating user-centred workflows, not a workflow management system. This would require, amongst other things, the ability to handle concurrent transfers, recovery from processing failures, and error and workflow reporting.

7.2 Archivematica API client

The architecture behind Archivematica's automation tools is straightforward: a state machine that interacts with Archivematica's API to start and monitor transfers. However, the chosen implementation provides only limited flexibility for scaling beyond a single transfer in a single pipeline. One option to bypass the limitations of the automation tools is to directly use the API that is exposed by Archivematica.

The Rockefeller Foundation used this approach to automate the preservation of 39,132 digitised pages from 189 diaries [85]. The Foundation's solution uses several scripts to automate the otherwise manual preparation of transfers and a separate script for submitting the prepared transfers. The rate at which the script submits transfers to Archivematica can be configured by changing a transfer submission delay. The number of simultaneous transfers can be controlled by increasing or decreasing the transfer submission delay. Using these scripts, the Rockefeller Foundation automatically ingested all diary pages in a period of two months.

Using a fixed submission rate is a simple way of limiting the number of active transfers. The problem with using a fixed submission rate is that the number of active transfers is only indirectly controlled by the submission interval. The number of concurrent transfers is, depending on the chosen submission interval, anywhere between 1 and many. To maintain a constant load on the system, the submission rate must be approximately equal to the transfer rate – a higher submission rate results in overloading the system while a lower submission rate results in under utilisation. This means that the optimal transfer delay is the average processing time of the transfers.

The processing time of a transfer in Archivematica depends on multiple factors. The most important factors are: the type of submitted material, the total size of the transfer, and the required preservation actions. If there is a large variation in only one of these parameters, the transfer rate can fluctuate significantly. This means that submission delays can only be used to control the load for homogeneous workloads. In the case of the Rockefeller foundation the transfer time is approximately similar for each page – thus resulting in a constant transfer rate. For our test

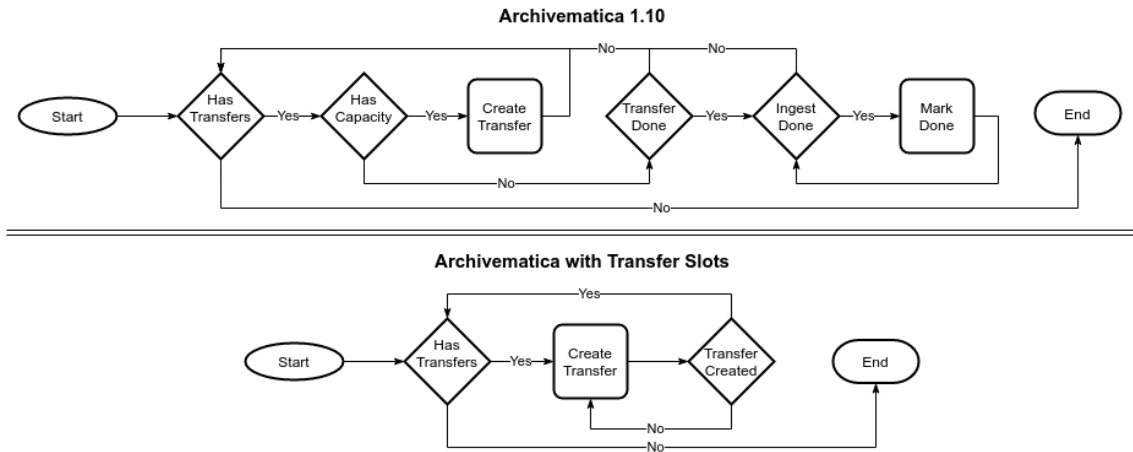


Figure 7.2: Flowchart displaying two control flows for bulk submission of transfers. The upper flowchart shows the control flow for the transfer script with transfer slots. The lower flowchart shows a simplified version of the transfer script. This requires extending Archivematica with transfer slots, as proposed in Chapter 5.2.

workload, 1100 Zenodo transfers, the transfer times vary between 1 and 60 minutes per transfer – resulting in a highly variable transfer rate.

For heterogeneous workloads, like the sample of Zenodo transfers, the number of active transfers needs to be controlled directly. To limit the number of concurrent transfers, we decided to use transfers slots instead of submission delays. A simplified version of the control flow for a transfer script using transfer slots can be found in the upper half of Figure 7.2. The idea is simple: the transfer script can only submit a transfer if there is a transfer slot available. If there are no transfer slots available, the transfer script tries again later. As soon as a transfer has failed or succeeded, the transfer slot is released and the transfer script can submit a new transfer.

During testing we identified two disadvantages with using this poor man’s rate limiting strategy. The first disadvantage is that any undefined state in the transfer script’s state machine results in a lost transfer slot. During our experiments we had several cases where a transfer slot was not released because a release condition was not set or a transfer required user intervention. The second disadvantage is that the number of transfer slots is controlled by the transfer script. This means that even though an individual script might submit transfers at a reasonable rate, multiple scripts can still overload an Archivematica pipeline.

This problem can be solved by moving the rate limiting from the transfer script to the Archivematica pipeline. In Chapter 5.2 we provide a detailed explanation describing how transfer slots can be implemented in Archivematica. When using Archivematica with the proposed transfer slot implementation, the transfer slot logic in the transfer script can be replaced with a simple retry loop. This simplified control flow is visualised in Figure 7.2. An additional benefit of moving the transfer slots to Archivematica, is that Archivematica controls the release of the transfer slots. This solves the problem of losing transfer slots to reaching an undefined state in the transfer script.

Combining a transfer script with transfer slots in Archivematica has proven to be very effective. All performance experiments in Chapter 5 use this approach for submitting the test transfers. The load on Archivematica can easily be increased or decreased by changing the number of available transfer slots. The transfer script keeps submitting new transfers as long as there are transfers available, maintaining a constant load. One disadvantage of using this approach for submitting transfers is that, at least in our implementation, the program state is not persistent. If the transfer script encounters an error during processing, it is not trivial to resume the submission process. In the case of performance experiments it is acceptable – and even required – to reprocess all data after a failure, but in a high volume production environment this must not happen.

7.3 Enduro

There are three problems with using a script based workflow management system such as the transfer scripts discussed in Chapter 7.2 or Archivematica’s automation tools. First of all, the solutions are not fault tolerant. When there is a failure during processing, the program state is lost and can often not be recovered. This results in transfers being either resubmitted or lost. The second problem is the tight coupling between the script, the Archivematica pipeline, and the transfer upload location. Adding a new location or pipeline requires changes in the scripts or deploying an additional instance. On top of this, the workflow management system does not provide a holistic view of the entire preservation system. This is especially a problem when managing multiple simultaneous pipelines or transfers. The last problem is that script based workflows are often very limited in their capabilities and tailor made for specific workflows. Supporting different workflows requires new custom scripts and sharing implementations is hard, if done at all.

In the Fall of 2019, Artefactual announced Enduro [86]. The main goal of Enduro is to “*provide a tool to orchestrate reliable workflows where multi-pipeline Archivematica deployments are integrated with other systems.*” Enduro is designed for workflow orchestration in large production environments. Currently, Enduro is still in the proof of concept phase. During the proof of concept phase, Enduro is designed and tested for a project with the Norwegian Health Archive. The goal of this project is to build a digital preservation system for all historical, current and future medical records from deceased patients treated by the Norwegian Specialised Healthcare-services.

The starting point of every workflow in Enduro is a watcher. Each watcher checks a specific file system location for new submissions. Enduro offers an integration with locally mounted storage and MinIO, a high performance object storage system. Once Enduro detects a new submission, it starts preparing a transfer. During transfer preparation the submission is verified and possibly restructured. Next, the prepared samples are submitted to one of the possibly multiple Archivematica pipelines. Pipeline selection is based on the source of the submission, each watcher corresponds to a specific Archivematica pipeline. Once submitted, every transfer is tracked and success and error messages are logged and displayed in the Enduro dashboard.

Enduro is built using Cadence, an orchestration engine originally developed by Uber [87]. Cadence is designed to create reliable, scalable, distributed workflows. The central abstraction in Cadence are activities. In its simplest form, a Cadence activity is a function or an object method in one of the supported languages. Cadence coordinates the execution of the activities. For all activities, or groups of activities, it is possible to set timeouts and retries. For long running activities it is also possible to monitor the health of the activity using a heartbeat function. Cadence keeps track of all activities using task lists. Task lists allow for flow control, throttling, routing, versioning and prioritisation of activities. This allows for fine grained control of the active workflows, scaling and rolling updates.

To evaluate if Enduro can be used as a workflow orchestration engine for the Digital Memory Platform we create a small evaluation environment. This environment consists of a couple of watched locations on EOS and two Archivematica pipelines. The watched location are filled with sample transfers, either images or Zenodo files. One of the connected pipelines is configured for image processing, while the other pipeline is configured for the more general Zenodo submissions.

The first tests are promising. Connecting new pipelines and watched directories to Enduro is simple. Because the submission mechanism is based on watching directories, integrating Enduro with existing workflows is simple. Once submitted, Enduro reports the transfer states in a clear dashboard, making it easy to identify slow transfers and failures. Connecting watched locations to specific Archivematica pipelines makes pipeline specialisation convenient. Sharing the same pipeline between different watchers enables the separation of CERN workflows. For example, experiments can submit datasets in their own separate and protected environment, while CERN can create one efficient pipeline for preserving the datasets of all experiments.

During the experiments we did encounter two problems. The first problem is the integration between EOS and Enduro. By default, Enduro uses Unix’ inotify API for watching directories. The inotify API provides a mechanism for monitoring filesystem events. Inotify reports only events that a user-space program triggers through the filesystem API [88]. As a result, inotify does not report remote events that occur on network file systems such as EOS. This means that watching an

EOS directory requires continuous polling of the watched directory. However, a caching problem between EOS and the operating system causes the mounted directory to be refreshed randomly. As a result of this, the file system poller reports all existing files in the directory as new files. This triggers new Enduro transfers, resulting in many duplicate transfers.

The second, and most pressing, problem is the current state of the Enduro project. In this early stage the workflows are completely tailored to the Norwegian Health Archive project. There is no clear design that ensures that other workflows are supported. This is both a liability and an opportunity. Considering the state of this project, and the currently existing solutions, there are two options: contribute to a promising project, or create custom workflow software. Given the engineering effort required for both – combined with CERN’s historical preference for contributing to Open Source software – contributing to Enduro might be the preferred option. This is, of course, only as long as the goals of the Enduro project align with the goals of the CERN Digital Memory Platform.

Chapter 8

Discussion and conclusion

The main goal of this research was to investigate if we can build a central archiving service using existing preservation tools. We decided to evaluate Archivematica and investigate if can build the CERN Digital Memory Platform using a distributed Archivematica cluster. We showed that it is possible to run and deploy a distributed Archivematica cluster. Without normalisation, and using a local disk instead of a distributed file system, we managed to achieve a transfer throughput of more than 30 MB/s for image preservation. For more computationally and storage heavy workloads that include normalisation we achieved a transfer throughput of 1.64 MB/s for image preservation, and 5.79 MB/s for video preservation. This means that for audio-visual material an Archivematica cluster can achieve the speeds required for production at CERN.

The required ingestion speed for CERN Open Data, 200 MB/s, is not yet achieved. Given that CERN Open Data only requires bit-level preservation, there are no gains in normalising experimental data. The main role for Archivematica in preservation of physics data will be structuring of the archive and metadata. This means that the raw I/O performance is more indicative of the possible performance of an Archivematica cluster. The performance tests for both EOS and the Ceph distributed file-system indicate that an ingest rate of 200 MB/s should be achievable. However, since the physics data is already stored on EOS and on tape, the best solution for Archiving CERN Open Data might be to ingest only the metadata and include references to the bit-preserved physics data.

The performance experiments indicate that it is possible to achieve a higher transfer throughput, but given the limited experimental data so far it is not possible to estimate the maximum achievable transfer throughput for Archivematica. However, we have been able to identify some opportunities for increasing the transfer throughput. From preliminary tests we believe that the focus of Archivematica development should be on two areas: I/O access patterns and the “distributed readiness” of Archivematica.

First of all, the I/O access patterns. We believe that the main bottlenecks in Archivematica are access to I/O and possibly the Storage Service. During our benchmarks the load on both the Dashboard and MCP server was low, while the load on the MCP clients was directly proportional, albeit unpredictable, to the workload. During the storage system evaluation we found that moving from internal to network attached storage severely impacts Archivematica’s performance. One possible solution to this problem is to move away from the shared directory concept in Archivematica. Instead of each MCP client performing arbitrary preservation tasks for different transfers, each MCP client will perform all preservation tasks for a single transfer. This means that an MCP client can operate from their own own local storage, instead of using a shared directory. The main problem with this approach is that it requires a complete overhaul of the Archivematica scheduling strategy. A more realistic direction would be to optimise the I/O access patterns of Archivematica by minimising access to the shared directory and moving I/O intensive parts of the workflow to local storage.

The second focus area is the “distributed readiness” of Archivematica. Even though the modular Architecture and microservice design of Archivematica make it a perfect candidate for a distributed deployment, it is not (yet) a truly distributed application. The obvious problem is the shared

directory concept, but during our performance tests we identified multiple race conditions, database deadlocks, and transaction time-outs. All of these problems were hard to find and easy to solve. However, it shows one fundamental flaw in our strategy: we are using Archivematica in a way it was not designed to operate. The basic architecture is there, the community is willing, but an engineering commitment is required to take the next steps.

A similar engineering commitment is required with regards to workflow management. With Enduro, the Archivematica project has made their first steps into providing an easy integration of Archivematica into existing workflows. However, in its current statue Enduro is still experiment and too focused on the requirements of funding partners. This means that even Enduro is more mature, there will probably be a significant engineering effort required for integration. Mainly because of the large diversity in information systems and material offered at CERN, but also because tailoring the workflow to a specific project might result in significant performance benefits.

One crucial aspect that is not covered in this research is whether the Digital Memory Platform can be used as the corner stone of CERN’s digital preservation strategy. In the same way as installing digital archival software is not enough for building an OAIS, using a digital preservation platform is not necessarily enough for having on OAIS. According to the reference model an OAIS is *“an archive, consisting of an organisation of people and systems, that has accepted the responsibility to preserve information and make it available for a Designated Community”*. The key element in this definition is the “organisation of people and systems that has accepted the responsibility”. There is no technical solution, no matter how sophisticated, that can compensate for an organisation that is not fully committed to long-term preservation.

The Digital Memory Platform will only add value, if it is accompanied with a package of measures to ensure a commitment to digital preservation. Looking at the Wilson’s pyramid of preservation needs from Chapter 2.1 there are three groups of needs: preservation of the bits of the original object, metadata, and policies. The policies are at the top of the pyramid, they are not an essential pre-requisite for a digital archive – but they are essential for a trustworthy archive. In “Requirements for Digital Preservation Systems: A Bottom-up Approach” Rosenthal et al. propose a list of requirements that complements the OAIS reference model [89]. These requirements are heavily focused on policies and disclosure. According to Rosenthal et al. organisations should: have an explicit thread model, have policies for the protection of data and Intellectual Property, disclose their external interfaces such as their SIP/AIP/DIP specifications, disclose access and preservation of source code, disclose the audit policy, and create policies around handling data loss. Each of these requirements have related technical components, but the requirements themselves are non-technical. These and other, very relevant, problems are not covered by this study and require at least the same amount of time and consideration.

Bibliography

- [1] Chester Bell, Tony Hey, and Alexander Szalay. “Beyond the Data Deluge”. In: *Science (New York, N.Y.)* 323 (Apr. 2009), pp. 1297–8. DOI: 10.1126/science.1170411.
- [2] Davis, Michael C. et al. “CERN Tape Archive – from development to production deployment”. In: *EPJ Web Conference* 214 (2019), p. 04015. DOI: 10.1051/epjconf/201921404015.
- [3] André G Holzner et al. “Data Preservation at LEP”. In: arXiv:0912.1803 (Dec. 2009). URL: <https://cds.cern.ch/record/1228028>.
- [4] *Restoring the first web site*. 2013. URL: <https://first-website.web.cern.ch/>.
- [5] Xiaoli Chen et al. “Open is not enough”. In: *Nature Physics* 15.2 (2019), pp. 113–119.
- [6] Jean-Yves Le Meur. *The Digital Memory of CERN*. <https://cds.cern.ch/record/2200146>. Oct. 2018.
- [7] Michelle Gallinger et al. “Trends in Digital Preservation Capacity and Practice: Results from the 2nd Bi-Annual National Digital Stewardship Alliance Storage Survey”. In: *D-Lib Magazine* 23.7/8 (2017).
- [8] Brian Lavoie and Lorcan Dempsey. “Thirteen ways of looking at... digital preservation”. In: *D-Lib magazine* 10.7/8 (2004).
- [9] Paul Conway. “Preservation in the age of Google: Digitization, digital preservation, and dilemmas”. In: *The Library Quarterly* 80.1 (2010), pp. 61–79.
- [10] Jeff Rothenberg. *Avoiding Technological Quicksand: Finding a Viable Technical Foundation for Digital Preservation. A Report to the Council on Library and Information Resources*. ERIC, 1999.
- [11] D-Lib Magazine. “Requirements for Digital Preservation Systems”. In: *D-Lib Magazine* 11.11 (2005), pp. 1082–9873.
- [12] Thomas C Wilson. “Rethinking digital preservation: definitions, models, and requirements”. In: *Digital Library Perspectives* 33.2 (2017), pp. 128–136.
- [13] Audrey Novak. *Fixity checks: checksums, message digests and digital signatures*. 2006.
- [14] Donald Waters and John Garrett. *Preserving Digital Information. Report of the Task Force on Archiving of Digital Information*. ERIC, 1996.
- [15] Michael Day. “Issues and Approaches to Preservation Metadata”. In: *Joint RLG and NPO Preservation Conference Guidelines for Digital Imaging* (1998).
- [16] Michael Day. “Metadata for digital preservation: a review of recent developments”. In: *International Conference on Theory and Practice of Digital Libraries*. Springer. 2001, pp. 161–172.
- [17] Bernard J Hurley. *The Making of America II Testbed Project: a digital library service model*. Digital Library Federation, 1999.
- [18] Angela Dappert and Markus Enders. “Using METS, PREMIS and MODS for archiving eJournals”. In: *D-lib Magazine* 14.9/10 (2008). DOI: 10.1045/september2008-dappert.
- [19] Peter van Garderen. “Archivematica: Using Micro-Services And Open-Source Software To Deliver A Comprehensive Digital Curation Solution.” In: *iPRES*. Citeseer. 2010.
- [20] Jerome P McDonough. “METS: standardized encoding for digital library objects”. In: *International journal on digital libraries* 6.2 (2006), pp. 148–158.
- [21] PREMIS Editorial Committee et al. *Premis data dictionary for preservation metadata*. 2015.
- [22] Stuart Weibel et al. “Dublin core metadata for resource discovery”. In: *Internet Engineering Task Force RFC* 2413.222 (1998), p. 132.
- [23] Rebecca S Guenther. “MODS: the metadata object description schema”. In: *Portal: libraries and the academy* 3.1 (2003), pp. 137–150.
- [24] Sarah Higgins. “The DCC curation lifecycle model”. In: (2008). URL: <https://doi.org/10.2218/ijdc.v3i1.48>.

- [25] Consultative Committee for Space Data Systems. “Reference model for an open archival information system (OAIS). CCSDS 650.0-M-2”. In: *Recommended Practice, Issue 2* (2012).
- [26] Sarah Higgins and Najla Semple. *OAIS Five-year review: Recommendations for update*. 2006.
- [27] Rhiannon S Bettivia. “The power of imaginary users: Designated communities in the OAIS reference model”. In: *Proceedings of the Association for Information Science and Technology* 53.1 (2016), pp. 1–9.
- [28] Dennis Nicholson and Milena Dobрева. “Beyond OAIS: towards a reliable and consistent digital preservation implementation framework”. In: *16th International Conference on Digital Signal Processing*. IEEE. 2009, pp. 1–8.
- [29] Julie Allinson. *OAIS as a reference model for repositories: an evaluation*. 2006.
- [30] OCLC/RLG Preservation Metadata Working Group et al. “A Metadata Framework to Support the Preservation of Digital Objects”. In: *OCLC Online Computer Library Center, Inc., Dublin, USA* (2002).
- [31] Michael W Kearney et al. “Digital Preservation Archives—A New Future Architecture for Long-term Interoperability”. In: *SpaceOps Conference*. 2018, p. 2402.
- [32] Stewart Brand. “Escaping the Digital Dark Age.” In: *Library Journal* 124.2 (1999), pp. 46–48.
- [33] Philip W Graham and Gregory N Hearn. “The digital Dark Ages: a retro-speculative history of possible futures”. In: *First Conference of the Association of Internet Researchers*. 2000.
- [34] Terry Kuny. “The digital dark ages? Challenges in the preservation of electronic information”. In: *International preservation news* 17 (1998), pp. 8–13.
- [35] David SH Rosenthal. “LOCKSS: Lots of copies keep stuff safe”. In: *NIST Digital Preservation Interoperability Framework Workshop*. 2010.
- [36] Priscilla Caplan. “DAITSS, an OAIS-based preservation repository”. In: *Proceedings of the 2010 Roadmap for Digital Preservation Interoperability Framework Workshop*. 2010, pp. 1–4.
- [37] Priscilla Caplan. “Building a digital preservation archive: Tales from the front”. In: *Vine* 34.1 (2004), pp. 38–42.
- [38] Emmanuelle Bermes, Louise Fauduet, and Sébastien Peyrard. “A data first approach to digital preservation: the SPAR project”. In: *Proceedings of the 76th IFLA general conference and council (Gothenburg, Sweden, 2010)*. 2010.
- [39] Edward Iglesias and Wittawat Meesangnil. “Using Amazon S3 in Digital Preservation in a mid sized academic library: A case study of CCSU ERIS digital archive system”. In: *Code4Lib Journal* 12 (2010).
- [40] Adam Farquhar and Helen Hockx-Yu. “Planets: Integrated services for digital preservation”. In: *International Journal of Digital Curation* 2.2 (2008).
- [41] Andrew Lindley, Andrew N Jackson, and Brian Aitken. “A collaborative research environment for digital preservation—the planets testbed”. In: *2010 19th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises*. IEEE. 2010, pp. 197–202.
- [42] Eld Zierau and Caroline van Wijk. “The PLANETS approach to migration tools”. In: *Archiving Conference*. Vol. 2008. 1. Society for Imaging Science and Technology. 2008, pp. 30–35.
- [43] Stephan Strodl et al. “How to choose a digital preservation strategy: Evaluating a preservation planning procedure”. In: *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*. 2007, pp. 29–38.
- [44] Pauline Sinclair, Tessella Bernstein, and Amir Bernstein. *An Emerging Market: Establishing Demand for Digital Preservation Tools and Services*. July 2010. URL: <https://www.planets-project.eu/docs/reports/Planets-VENDOR-White-Paperv4.pdf>.
- [45] Raivo Ruusalepp and Milena Dobрева. *Digital preservation services: state of the art analysis*. 2012.
- [46] István Alföldi et al. *General pilot model and use case definition*. Version 1.4. Feb. 2018. DOI: 10.5281/zenodo.1170009.
- [47] Rui Castro, Luis Faria, and Miguel Ferreira. “Meet RODA, a full-fledged digital repository for long-term preservation”. In: *8th International Conference on Preservation of Digital Objects*. 2011.
- [48] *E-ARK Web*. URL: <https://github.com/E-ARK-Software/earkweb>.
- [49] *ESSArch*. URL: https://github.com/eark-project/ESSArch_EPP.
- [50] Janet Anderson et al. *E-ARK Final Report*. Version 8.0. Feb. 2018. DOI: 10.5281/zenodo.1173157.

-
- [51] CEF Digital. *What is eArchiving*. Apr. 2020. URL: <https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/What+is+eArchiving>.
 - [52] Courtney Mumma and Peter van Garderen. “Realizing the Archivematica vision: delivering a comprehensive and free OAIS implementation”. In: *10th International Conference on Preservation of Digital Objects*. 2013.
 - [53] Bronwen Sprout and Mark Jordan. “Archivematica As a Service: COPPUL’s Shared Digital Preservation Platform/Le service Archivematica: La plateforme partagée de conservation de documents numériques du COPPUL”. In: *Canadian Journal of Information and Library Science* 39.2 (2015), pp. 235–244.
 - [54] Shaun Trujillo et al. “Archivematica outside the box: Piloting a common approach to digital preservation at the Five College Libraries”. In: *Digital Library Perspectives* 33.2 (2017), pp. 117–127.
 - [55] Sean Buckner. *Archivematica Across Texas*. 2018. URL: <http://hdl.handle.net/2249.1/87455>.
 - [56] Max Eckard, Dallas Pillen, and Mike Shallcross. “Bridging Technologies to Efficiently Arrange and Describe Digital Archives: the Bentley Historical Library’s ArchivesSpace-Archivematica-DSpace Workflow Integration Project”. In: *Code4Lib Journal* 35 (2017).
 - [57] Meghan Goodchild and Grant Hurley. “Integrating Dataverse and Archivematica for Research Data Preservation”. In: *16th International Conference on Digital Preservation*. 2019.
 - [58] Bernd Pollermann et al. *Report on Long-Term Electronic Archiving (LTEA)*. Tech. rep. Geneva: CERN, 2000. URL: <https://cds.cern.ch/record/1028139>.
 - [59] DPHEP Collaboration et al. *Status Report of the DPHEP Collaboration: A Global Effort for Sustainable Data Preservation in High Energy Physics*. Update to the previous report with more information from ATLAS, HERA, ZEUS, DELPHI. Feb. 2016. DOI: 10.5281/zenodo.46158. URL: <https://doi.org/10.5281/zenodo.46158>.
 - [60] Jake Cowton et al. “Open data and data analysis preservation services for LHC experiments”. In: *Journal of Physics: Conference Series*. Vol. 664. 3. IOP Publishing. 2015, p. 032030.
 - [61] Xiaoli Chen et al. “CERN analysis preservation: a novel digital library service to enable reusable and reproducible research”. In: *International Conference on Theory and Practice of Digital Libraries*. Springer. 2016, pp. 347–356.
 - [62] Tibor Šimko et al. “REANA: A system for reusable research data analyses”. In: *EPJ Web of Conferences*. Vol. 214. EDP Sciences. 2019, p. 06034.
 - [63] CERN. *Our History*. Apr. 2020. URL: <https://home.cern/about/who-we-are/our-history>.
 - [64] CCSDS. “Audit and certification of trustworthy digital repositories”. In: (2011).
 - [65] Jean-Yves Le Meur and Nicola Tarocco. “The obsolescence of Information and Information Systems: CERN Digital Memory project”. In: *EPJ Web of Conferences*. Vol. 214. EDP Sciences. 2019, p. 09003.
 - [66] João Fernandes et al. *ARCHIVER D2.1- State of the Art, Community Requirements and OMC Results*. Version 1.0. Jan. 2020. DOI: 10.5281/zenodo.3618215.
 - [67] Tim Hutchinson. “Archidora: Integrating Archivematica and Islandora”. In: *Code4Lib Journal* 39 (2018).
 - [68] Ashley Blewer, Kelly Stewart, and Justin Simpson. *Archivematica Camp Geneva – Stream 1: Camp slides*. Oct. 2019.
 - [69] Germán Cancio et al. “Experiences and challenges running CERN’s high capacity tape archive”. In: *Journal of Physics: Conference Series*. Vol. 664. 4. IOP Publishing. 2015, p. 042006.
 - [70] Artefactual Systems Inc. *Installing Archivematica*. Oct. 2019. URL: <https://www.archivematica.org/en/docs/archivematica-1.10/admin-manual/installation-setup/installation/installation/>.
 - [71] Artefactual Systems Inc. *Scaling Archivematica*. Oct. 2019. URL: <https://www.archivematica.org/en/docs/archivematica-1.10/admin-manual/installation-setup/customization/scaling-archivematica>.
 - [72] Hervé Rousseau et al. “Providing large-scale disk storage at CERN”. In: *EPJ Web of Conferences*. Vol. 214. EDP Sciences. 2019, p. 04033.
 - [73] Andreas Joachim Peters, Elvin Alin Sindrilaru, and Geoffrey Adde. “EOS as the present and future solution for data storage at CERN”. In: *Journal of Physics: Conference Series*. Vol. 664. 4. IOP Publishing. 2015, p. 042042.
-

- [74] José Castro León. “Advanced features of the CERN OpenStack Cloud”. In: *EPJ Web of Conferences*. Vol. 214. EDP Sciences. 2019, p. 07026.
- [75] Sage A Weil et al. “Ceph: A scalable, high-performance distributed file system”. In: *Proceedings of the 7th symposium on Operating systems design and implementation*. 2006, pp. 307–320.
- [76] Jens Axboe. *Fio-flexible i/o tester*. Sept. 2019. URL: <https://github.com/axboe/fio>.
- [77] Andreas Joachim Peters. “EOS as a Filesystem”. In: *EOS workshop*. CERN. 2017.
- [78] Andreas Joachim Peters. “To FUSE(x) or not to FUSE(x) ...” In: *EOS workshop*. CERN. 2020.
- [79] Ruben Gaspar Aparicio and Ignacio Coterillo Coz. “Database on Demand: insight how to build your own DBaaS”. In: *Journal of Physics.: Conference Series*. 664.4 (2015), 042021. 8 p. DOI: 10.1088/1742-6596/664/4/042021.
- [80] *Archivematica 1.11 and Storage Service 0.16 release notes*. Apr. 2020. URL: https://wiki.archivematica.org/Archivematica_1.11_and_Storage_Service_0.16_release_notes.
- [81] Amazon.com Inc. *Amazon States Language*. URL: <https://states-language.net/spec.html>.
- [82] Artefactual Systems Inc. *AIP structure*. Nov. 2019. URL: <https://www.archivematica.org/en/docs/archivematica-1.10/user-manual/archival-storage/aip-structure/>.
- [83] Andrew Purcell. *CERN and OpenAIREplus launch European research repository*. May 2013. URL: <https://cds.cern.ch/record/1998025>.
- [84] Ross Spencer. *Automation tools: making things go...* Mar. 2019. URL: <https://www.slideshare.net/Archivematica/automation-tools-making-things-go-march-2019>.
- [85] Bonnie Gordon. *Automating Archivematica Ingests*. Jan. 18, 2019. URL: <https://blog.rockarch.org/automating-archivematica-ingests>.
- [86] Artefactual Systems Inc. *Enduro*. Oct. 2019. URL: <https://enduroproject.netlify.com/>.
- [87] *Cadence*. URL: <https://cadenceworkflow.io/>.
- [88] *inotify - monitoring filesystem events*. URL: <https://www.man7.org/linux/man-pages/man7/inotify.7.html>.
- [89] David SH Rosenthal et al. “Requirements for digital preservation systems: A bottom-up approach”. In: *arXiv preprint cs/0509018* (2005).