A Decade of Preservation: System Migrations in Chronopolis

Sibyl Schaefer
University of California, San Diego
9500 Gilman Drive
La Jolla, CA 92092
+1-858-246-0741
sschaefer@ucsd.edu

Mike Smorul
SESYNC
1 Park Place, Suite 300
Annapolis, MD 21401
+1-410-919-4809
msmorul@sesync.org

David Minor
University of California, San Diego
9500 Gilman Drive
La Jolla, CA 92092
+1-858-534-5104
dminor@ucsd.edu

Mike Ritter
UMIACS
8223 Paint Branch Drive, Room 3332
College Park, MD 20742
+1-301-405-7092
shake@umiacs.umd.edu

ABSTRACT
This paper provides a historical look at the technical migrations of the Chronopolis digital preservation system over the last ten years. During that time span the service has undergone several software system migrations, moving from middleware-based systems to a suite of individual, finely scoped components which employ widely used and standardized technologies. These transitions have enabled the system to become not only less dependent on interpretation by middleware, but also easier to transfer to new storage components. Additionally, the need for specialized software knowledge is alleviated; any Linux systems administrator should be able to install, configure, and run the software services with minimal guidance. The benefits of moving to a microservices approach have been instrumental in ensuring the longevity of the system through staff and organizational changes.

Keywords
Software Migrations; Microservices; Middleware; iRODS; SRB

1. INTRODUCTION
The Chronopolis system provides long-term, distributed, highly redundant preservation storage. Chronopolis was instituted in 2007 and initially funded by the Library of Congress's National Digital Information Infrastructure and Preservation Program.
A variety of issues related to software and system maintenance and relevant staffing prompted two major software migrations resulting in the service moving from a very centralized, middleware-based system to a set of microservices, defined as "independent but interoperable components that can be freely composed in strategic combinations towards useful ends."[1]

2. CHRONOPOLIS HISTORY
The original Chronopolis partners included the San Diego Supercomputer Center (SDSC) in California, the National Center for Atmospheric Research (NCAR) in Colorado, and the University of Maryland Institute for Advanced Computer Studies (UMIACS) in Maryland. Chronopolis was designed to preserve hundreds of terabytes of digital materials, using the high-speed networks of the partner institutions to distribute copies to each node. In addition to geographical variation, the Chronopolis partner nodes all operate different technology stacks, thus reducing risks associated with specific hardware or software component failures.

Chronopolis has always been administered by people employed within the participating data centers. SDSC provided the original management team, including financials and budgeting, grant management, and HR-related functions. The Center also housed the core system administration staff, who focused on storage systems, software management, and network configurations. NCAR and UMIACS allocated less staff who individually took on broader portfolios. So, for example, a single staff member at NCAR or UMIACS could be responsible for systems, software, networking and code development. These kinds of staffing arrangements grew out of the grant-funded nature of Chronopolis and were appropriate for the network's early development. In subsequent years there have been ongoing efforts to redistribute duties so that some staff positions were fully dedicated to Chronopolis and that these positions were full-time and permanent.

Chronopolis is a founding node in the Digital Preservation Network and also offers preservation storage through the DuraCloud service. Chronopolis was certified as a Trusted Digital Repository by the Center for Research Libraries in 2012 and plans to undergo ISO 16363 certification. Original partner SDSC is no longer an active member of the collaborative, and the University of California San Diego Library has assumed full management authority.

Chronopolis was designed to impose minimal requirements on the data provider; any type or size of digital materials is accepted. Data within Chronopolis are considered "dark." Once ingested, access to the data is restricted to system administrators at each node. These administrators can disseminate a copy of the data stored on their node back to the depositor upon request.

Chronopolis constantly monitors content, especially changes, through the Audit Control Environment (ACE). ACE is a standalone product designed to provide a platform-independent, third party audit of a digital archive. Developed by the ADAPT (A Digital Approach to Preservation Technology) team at UMIACS, research on the ACE software was initially funded by a grant from the National Science Foundation and Library of Congress. Additional development has increased the utility of the program in auditing collections and improved its reporting and logging features.

ACE consists of two components: the Audit Manager and the Integrity Management Service (IMS). The Audit Manager is software that checks local files to ensure they have not been altered. Each Chronopolis node runs the Audit Manager on collections an average of every 45 days. The Integrity Management Service issues tokens used by the Audit Manager to verify that its local store of file digests has not been tampered with. The ADAPT project runs a publically available IMS at ims.umiacs.umd.edu and any group may freely use to register and verify tokens. The Audit Manager software has been released under an open source license and may be downloaded from the ADAPT project website[2].

3. MIGRATIONS

Over the last decade Chronopolis has undergone several infrastructure migrations. Each migration increases the risk of data corruption; ACE has been used as a central piece of the migration process to maintain data integrity.

Two types of migrations have occurred through the Chronopolis lifespan:

1. Standard storage refreshes and upgrades. Storage and network components are generally refreshed every three to five years within the Chronopolis data centers. When Chronopolis was funded primarily through grants, these updates were often coordinated amongst the nodes. Since then, changes have happened asynchronously so that equipment costs are more distributed. Although refreshes and upgrades are major endeavors, these node-internal changes generally do not impact peer nodes other than the upgraded node being temporarily unavailable.
2. Middleware upgrades and changes. Chronopolis has undergone two major software upgrades. The first generation of Chronopolis used the Storage Resource Broker (SRB) to manage data, which was then superseded by the integrated Rule-Oriented Data System (iRODS). Due to a number of factors, in 2014 the Chronopolis sites began migrating out of iRODS and into a homegrown data management system named ChronCore. For the purposes of this paper, we will only be discussing these system transitions and not the more routine storage migrations.

3.1 First Migration: SRB to iRODS

Chronopolis was initially instantiated using the SRB middleware. One motivator for implementing Chronopolis using the SRB was the unified view it provides of different types of storage. During the initial stages of Chronopolis development, both NCAR and SDSC employed a mix of disk and tape storage and the SRB integrated the management of data across both media. This feature diminished in utility as NCAR and SDSC transitioned to large, centrally maintained, disk-based storage pools that were visible to Chronopolis as a single file system.

These new storage pools were directly controlled by iRODS, which was responsible for creating, writing, and reading files on them. UMIACS did not offer a unified file system and was constrained by the total file system size supported by the UMIACS group, so a custom solution, SWAP (Simple Web-Accessible Preservation), was developed. SWAP efficiently mapped files across multiple locally attached disks and servers in a way that required no centralized metadata catalog. Files from this storage were then registered into iRODS post-replication to provide read-only federated access to peer sites. This ensured that future migrations could be performed using standard Unix utilities.

While not evident at inception, SRB's architecture would pose problems for future migration out of the system. All file metadata (names, permission, etc.) were stored in a central metadata catalog while actual file storage was done by renaming the file identifier to this database. This metadata separation posed problems during migration, because the exporting of collection data was only possible using SRB tools, and not at the file system level. This required all sites to store a duplicate copy of all the data in both the old Chronopolis storage and new iRODS locations to ensure that fixity could be checked at all points of data movement. This migration had to occur at each site. Requiring duplicate copies of all data at each node or re-replicating all data between nodes would be a clear constraint on Chronopolis in the future.

3.2 Second Migration: iRODS to ChronCore

Although the federated file system provided an easy means to view contents across the entire Chronopolis network, the administration of iRODS at each site became more of an issue over time, largely due to the dedicated expertise required to maintain the software. The two data centers employing iRODS, NCAR and SDSC, eventually stopped running production iRODS teams, which impacted Chronopolis operations. Additionally, only a small subset of iRODS features was really being applied; previous experience with the SRB made the Chronopolis team wary of technology lock-in so they decided against implementing the sophisticated rule mechanism and metadata capabilities of iRODS in order to facilitate future migrations out of the system. Rather than expending valuable resources on maintaining iRODS support at two nodes, the team decided to migrate to a third system.

ACE was instrumental in moving off of iRODS. Each collection was updated and audited through the REST API to make sure files and tokens were valid. The audit results reported differences between the registered checksums for files and the post-migration captured checksum on local disk, likely due to a bug in the iRODS ACE driver. These discrepancies were resolved by validating the BagIt[3] manifests for each collection and comparing checksums across partner sites. Upon validation that the files were intact, they were removed from ACE and re-registered with accurate fixity information.

4. CHRONCORE

The main purpose of ChronCore is to package and distribute data securely throughout the system, providing several levels of bit auditing to ensure that nothing is lost in transmission. The distributed architecture of Chronopolis led to the creation of distributed services. As each core service emerged, it was assigned scoped operations depending on its place in the Chronopolis pipeline. ChronCore consists of three such scoped services: intake, ingest, and replication. Currently only the UCSD library node runs the intake and ingest services, which package, record, and stage data for replication. All partner sites run the replication services, which poll the ingest service hourly to determine if new collections have been staged for replication.

4.1 ChronCore Services

4.1.1 Intake

Content is submitted by a depositor through one of the Chronopolis Intake services. If the content is bagged and a manifest is present, the Intake service will verify the manifest and, if valid, register the collection with the Ingest server. If the content has not been previously packaged, the Intake service will bag the content before registering it with the Ingest server.

4.1.2 Ingest

The Ingest service serves as a central registry for content in Chronopolis. It generates ACE tokens, which provide provenance data about when content was first ingested and what fixity values it arrived with. Once tokenization is complete, the Ingest service will create replication requests which are picked up by each partner site. Replication of both the content and tokens are served through a RESTful API.

4.1.3 Replication

Each partner site runs a Replication service that periodically queries the Ingest service API and performs replications on any requests. The general flow of events for a replication is:

1. Query for new requests.
2. Transfer data (rsync/https/gridftp).
3. Respond with checksum of transferred content.
4. If content is valid, register it with the local Audit Manager and kick off the initial local audit.
5. Close transaction.

If a replication fails, the Ingest server is notified and a new request needs to be generated with the replication server. The cause of failure is first manually reviewed to determine if the cause was intermittent (network issues) or something more serious (bit flips).

4.2 Industry Standard Technologies

As a lightweight system of microservices, ChronCore does not contain the entire breadth of functionality that the previously employed middleware systems offered; time has proven that this advanced functionality is not necessary for Chronopolis operations. Instead of developing new tools or implementing new technologies, project leaders decided to take advantage of older, simpler technologies that have been demonstrated over time to operate at the necessary scales.

- SSH: by providing access to a service account at each site, a federated system can be 'mocked' with strict access controls ensuring no data is tampered with.
- rsync: this is a proven transport mechanism for transferring data to each site. It allows for transfers to be restarted, mitigating the impact of intermittent network problems. Over Chronopolis' lifetime, the community at large has shown that this tool could scale to Chronopolis-sized connections.
- HTTP/REST: REST/JSON has rapidly become an accepted communication protocol, replacing older vendor-specific binary protocols. In addition, its support by numerous languages and toolkits assures vendor or language lock-in will not be an issue.

5. CONCLUSIONS

Throughout the last ten years, Chronopolis has been able to migrate a set of replicated collections through a variety of systems while maintaining bit integrity and provenance. The experience gained from system migrations has led the Chronopolis team to espouse the following tenets.

Use Independent Microservices. Chronopolis has migrated from very centralized middleware systems to a mix of off-the-shelf and custom technologies. By creating a system of specialized services, each can follow its natural technical lifecycle and be replaced as appropriate. ACE was an early example of this model and has persisted as a key part of the Chronopolis infrastructure even as every other component has been switched out.

Always have direct data access. The move from systems where raw file data was hidden below layers of middleware to standard file system storage has removed a reliance on specialized tools or software and enabled the use of more widely used and supported utilities. Conceptually this also follows the independent services lesson mentioned previously, as it is a critical aspect in allowing technologies to be switched out as necessary. In previous implementations, access to files was dependent on metadata services managed by the middleware system. Potential loss of this metadata catalog due to a higher-level service failure created increased risk within the system, requiring additional care to ensure the catalog was highly available and recoverable. Complete loss of these services could render the on-disk data unusable. Information about provenance, preservation actions, and even original filenames could also be lost. These middleware systems also required each Chronopolis partner to maintain in-house expertise to support this custom software. Maintaining the necessary staff expertise at all three sites increased the operational costs of the network.

Choose "boring" technologies that don't require specialized expertise[4]. Chronopolis has changed not only the software it uses over time but also the staff that runs the system. Each node has experienced significant staff turnover over the past ten years; sometimes within as little as a year one or more nodes would undergo changes in management. By migrating from large proprietary systems to common technologies, Chronopolis has greatly increased its resilience to personnel changes at any of its sites. All of the core tools are well supported, have large, active user communities, and are within the skill sets of most system administrators. Should there be a personnel shortage at a site, it would be fairly easy to contract the necessary expertise to keep the node up and running. Using widely adopted tools also lowers the barrier for new nodes to participate in Chronopolis, and was instrumental in the ease with which the management of the San Diego node transferred from SDSC to the UCSD Libraries.

6. REFERENCES

[1] Abrams, S., Cruse, P., Kunze, J., and Minor, D. 2010. Curation Micro-services: A Pipeline Metaphor for Repositories. *Journal of Digital Information*. 12, 2. (April. 2011), https://journals.tdl.org/jodi/index.php/jodi/article/view/1605/1766.

[2] ADAPT: An Approach to Digital Archiving and Preservation Technology. https://adapt.umiacs.umd.edu.

[3] Kunze, J. et al. The BagIt File Packaging Format (VO. 97) Network Working Group Internet-Draft. https://tools.ietf.org/html/draft-kunze-bagit-13

[4] McKinley, Dan. Choose Boring Technology. http://mcfunley.com/choose-boring-technology