Selecting efficient and reliable preservation strategies: modeling long-term information integrity using large-scale hierarchical discrete event simulation

(Research Paper)

Micah Altman
Massachusetts Institute of Technology

Richard Landau
Massachusetts Institute of Technology

Abstract

This article addresses the problem of formulating efficient and reliable operational preservation policies that ensure bit-level information integrity over long periods, and in the presence of a diverse range of real-world technical, legal, organizational, and economic threats. We develop a systematic, quantitative prediction framework that combines formal modeling, discrete-event-based simulation, hierarchical modeling, and then use empirically calibrated sensitivity analysis to identify effective strategies.

Specifically, the framework formally defines an objective function for preservation that maps a set of preservation policies and a risk profile to a set of preservation costs, and an expected collection loss distribution. In this framework, a curator's objective is to select optimal policies that minimize expected loss subject to budget constraints. To estimate preservation loss under different policy conditions optimal policies, we develop a statistical hierarchical risk model that includes four sources of risk: the storage hardware; the physical environment; the curating institution; and the global environment. We then employ a general discrete event-based simulation framework to evaluate the expected loss and the cost of employing varying preservation strategies under specific parameterization of risks.

The framework offers flexibility for the modeling of a wide range of preservation policies and threats. Since this framework is open source and easily deployed in a cloud computing environment, it can be used to produce analysis based on independent estimates of scenario-specific costs, reliability, and risks.

We present results summarizing hundreds of thousands of simulations using this framework. This exploratory analysis points to a number of robust and broadly applicable preservation strategies,

Submitted 08 07 2019

Correspondence should be addressed to Micah Altman, 6104 Building NE36, MIT, Cambridge, MA, 02139
Email: <micah.altman@gmal.com>

provides novel insights into specific preservation tactics, and provides evidence that challenges received wisdom.

Significance

Deploying cost-effective, reliable bit-level long-term preservation at scale remains an unsolved problem. [Rosenthal 2010; Altman *et* al. 2015] Memory organizations have identified a number of high-level 'best practices', such as fixity checking and geographically distributed replication, but there is little specific guidance or empirically-based information on selecting specific preservation strategies that fit a curating institution's risk-tolerance, threat profile, and budget. Thus, while cloud storage vendors such as Amazon tout 99.999999999% durability; these claims typically lack substantial explanation or even clear definitions [see e.g. Mellor 2018] Further, professional memory organizations vary significantly in the practices they use, and how they use them -- even in the number of copies held. [Gallinger *et* al. 2017]

Strategies for preserving digital information are generally based on the observation that neither digital media, nor formats, nor institutions are reliably durable. While a number of 'good practices' are recognized for digital preservation, many of these practices are heuristic, and most are based on experience with particular technologies and threats. Stewards of digital information are faced with a large set of choices in developing a preservation strategy. These choices include document size and data format; file encryption and compression; storage media durability and reliability; collection replication, distribution, verification, and repair. [see e.g. Gallinger *et* al. 2017] These choices have the potential to change dramatically the cost of a preservation strategy, and how (and where) that strategy is vulnerable to a wide range of threats. Moreover, changes in these factors interact in complex ways, making it difficult to discover optimal/efficient strategies.

This article addresses the problem of formulating simple, efficient, and reliable operational preservation policies that ensure bit-level information integrity over long periods, and in the presence of a diverse range of real-world technical, legal, organizational, and economic threats. We develop a systematic, quantitative prediction framework that combines formal modeling, discrete-event-based simulation, hierarchical modeling, And then use empirically calibrated sensitivity analysis to identify effective strategies.

Methodology

The ultimate goal of information preservation is to communicate across time. Our concrete objective, broadly speaking, is to maintain a collection of documents, so that its contents can be read at a designated future time. Communication will be deemed a success if at some designated future time the integrity of the documents has been maintained. (In the full paper we extend this to the case where additional context about formats, encryption, etc. must be preserved so that the document can be meaningfully rendered.)

More formally, we model the curator's task as the selection of preservation practices and parameters (e.g., auditing frequency) based on feasible practices and available systems, such that preservation costs are minimized, such that the expected loss of content does not exceed the curator's target given a specified risk profile. (In the full paper, we also model a dual problem of minimizing loss given a fixed budget.)

Curators might wish to be aware of the technologies and concerns in the areas we cannot control. In defining the curatorial strategy, we focus on those elements that curators are likely to readily

control [Gallinger, *et* al. 2017] : the number and distribution of copies, how to audit and repair these, and whether to apply file transformations such as compression or encryption.

Prior work by [Baker *et* al. 2006], which we extend, uses single-level simulation to examine a core tradeoff between replication and auditing. Others have used related approaches -- such as Markov Chain Simulation [Lebrecht *et* al. 2011; Li *et* al. 2012] to estimate data failure at the hardware storage layer under simplified threat models in relation to the choice of storage approaches (e.g. RAID configuration). Work such as [Pinheiro et al. 2007] summarizes empirical rates of storage failures -- we use this and subsequent work to calibrate the model we present.

Where both cost and loss functions are simple and behaved, it may be possible to find the optimal solution through closed-form mathematical analysis or simple Monte-Carlo simulation. However, in more realistic conditions, risk of loss is a complex function of multiple threats at multiple levels [see, e.g. Rosenthal, et al. 2005] -- including low-level media failures, mid-level events such as manufacturing defects that affect clusters of media, and high-level events such as government action that can simultaneously affect multiple replicas of entire collections. Thus we use computationally-intensive discrete event simulation to estimate the losses under different proposed strategies.

Our underlying storage model is also hierarchical. A client (library) has a collection of documents in digital form. These documents are recorded on sectors of a storage medium; errors in the sectors within it cause a document to be lost -- as illustrated in **Figure 1**. As also illustrated in Figure 1, large documents are larger "targets" than small documents and are therefore more likely to be hit by random errors.

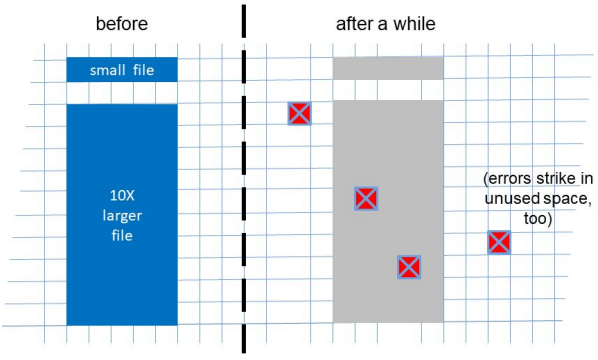


Figure 1: Documents are stored on sectors of disk (or other) storage. Large files occupy more sectors than small files. When errors occur randomly in sectors, large files present a larger "target area" and are more likely to be hit by an error. In this study, we treat a document with any sector errors as corrupt.

A copy of the collection of documents is stored on a server maintained by a separate institution. If the client maintains multiple copies of the collection of documents, then several copies are stored on separate, independent servers. Customers retrieve documents from the server(s) to read them. An error may occur that corrupts a portion of a document or makes that copy inaccessible. In this case, we consider the copy to be lost. Other copies may still persist. If all copies of a document are lost, then the document itself is permanently lost.

Specifically, we are concerned with four hierarchical levels of failures that affect document collections.

- **Disk Sector Failures.** Small scale errors in disk recording can result in partial or total failure of a document copy. For the purposes of this study, we have assumed that a single failure in the body of a document's data causes the document to be considered lost.
- **Environmental Glitches.** The rate at which errors arrive in disk sectors is not always constant. Transient environmental conditions, "glitches," such as failures of HVAC, electrical noise, and such can raise the rates of sector errors.
- **Server Failures.** Data servers are not immortal. Companies may go out of business, be bought by or merge with other companies, change their mission, and so forth. Our research covers a range of expected lifetimes of server companies.
- **Major Shocks.** Economic and regional environmental conditions can cause multiple servers to fail in short periods of time. Economic downturns can stress many companies in an industry. Regional environmental disasters such as earthquakes, floods, wars, and such can result in failure, or inaccessibility, of multiple servers. And government censorship can make servers or groups of servers inaccessible and lost to a replication set of servers. Such correlated failures are a serious threat to a collection and require more intense monitoring by the client library to ensure the health of the collection. .

In Table 1 we summarize these failure types, their distribution, and their visibility. We also provide examples of real world failures that one can represent using these event types.

	Characteristics			Exemplar threats
Layer	Role	Visibility	Distribution	(Low-High Severity)
Hardware (Sector)	Causes sector error / single document loss	Silent - detected on audit	Poisson event	Media error Failed controller
Local environment (Glitch)	Increases rate of storage error	Invisible. Detected through indirect effect on sectors.	Poisson event of some duration	HVAC faults Flood
Institution (Server Failure)	Causes loss of a single copy of a collection	Silent - detected on audit	Exponential life	Corporate merger Curator Error Cyber attack
Macro Environment (Major Shock)	Causes loss of multiple copies of collection / increases the rate of server failure	Varies. May be invisible, silent, or visible.	Poisson event / duration	Recession Regional war Government Suppression

Table 1: A Hierarchical Typology of Preservation Threats. Small errors corrupt storage sectors of individual documents. Greater threats cause entire server(s) to fail, losing all copies of documents on the server(s).

Note that these distributions imply some assumptions about operations:

- The model assumes that some form of local error-correcting storage, such as RAID is used, and that the benefits of server-local error correction, such as the use of RAID storage be incorporated in the logical sector error rate. Thus the sector error rate should reflect the ex-post rate of sector failures, after any RAID repairs. This rate effect of different RAID configurations, pattern of single disk failure and subsequent repair, etc. are not directly modeled.
- The model assumes that a storage unit is replaced, and content automatically migrated, after its normal service life. If not, sector failure risk would typically accelerate after the 3-5 year service lifetime of the initial hardware. We can still model failure to migrate media modeled through introduction of environmental glitches that increase sector error failure.
- The model assumes that the failure of institutions is conditionally independent, absent shocks. In other words, when two servers share an internal dependency -- such as reliance on the same third-party storage layer, a shock should also be added to the model to reflect that dependency.

A Simple Cost Model

Many storage vendors may be available to a client, each with charge schedules. For the most part, vendors will charge for storage and bandwidth.

- A charge per month per byte stored (usually per gigabyte or petabyte). The cost of storage may vary by "quality" of storage, based on its typical error rate or perhaps on speed of retrieval access. Storage is charged per copy; multiple copies cost more.
- A charge per month per byte sent in or out ("ingress" and "egress" charges). Bytes sent do not distinguish between user access for normal retrieval and administrative access for auditing. The cost may vary by speed or reserved bandwidth (Mbps).
- Charge schedules for storage and bandwidth may include quantity discounts.

For the purposes of this study, a client will store a collection on a set of servers of the same "quality" level. Documents with differing quality requirements are considered separate collections and are stored and managed separately.

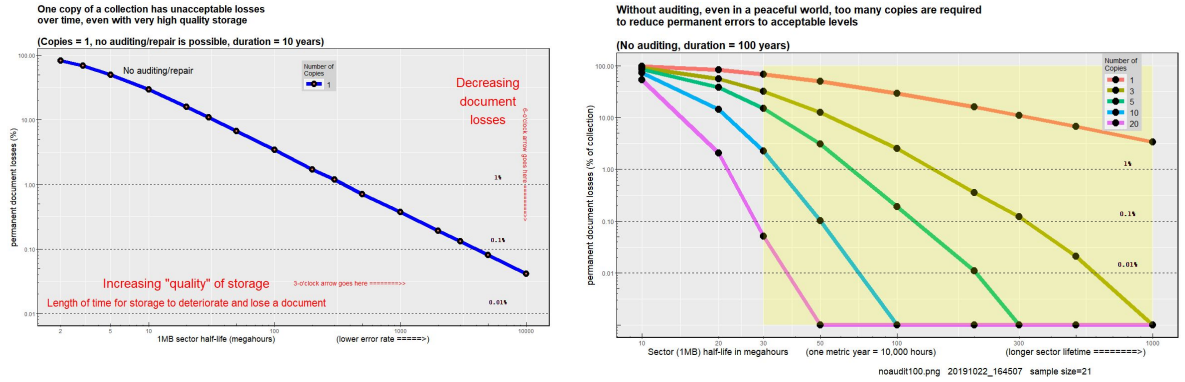
Results

In this section we show how simulation of the multi-level failure model can be used to design a robust preservation strategy: First, we start by modeling sector-level errors that cover a very wide range of error rates, more than three orders of magnitude. We find that maintaining multiple copies of a document collection, along with a regimen of regular auditing, can preserve the collection over a range of error rates wider than is likely to be encountered with real commercial disk drives. Second, we introduce small and large glitches, which increase the base error rate from two to ten times. We find that such temporary excursions are indistinguishable from minor or even major differences in the base error rate, and that the prior strategy remains robust. Third, we introduce whole server failure. We find that, by incrementally increasing the number of copies and auditing frequency, a client library can protect the collection against individual server failures over a wide range of server lifetimes. Finally, we introduce major shocks that increase the rate of server failure and/or simultaneously eliminate up to three servers. We find that sufficiently increasing frequency

of auditing and repair can protect a collection against even the shocks induced by major recessions and minor wars.

Constructing a Preservation Strategy that is Robust to Base Storage Quality

If a collection exists as only a single copy, then it is very likely that some of its documents will be lost within a decade, even if the storage medium is highly reliable. **Figure 2** shows the likelihood that a single document will be lost, over a decade, depending on the sector lifetime of the storage medium. Further, even if the rate of error accumulation is much lower than illustrated above, many documents will be lost over longer periods of time: For example, over 50 years, about 20 parts per million (ppm) to 200ppm of the collection will be lost, depending on document sizes, even if media reliability were 10x the maximum shown.



(2A) **(2B)**

Figure 2. (2A) A single copy of a document collection suffers considerable losses over a short period of time, even with a very high quality storage medium. (2B) Over long periods, even very large numbers of unaudited copies will suffer permanent losses. The shaded area covers what we consider to be the plausible region of disk quality.

A single copy of a document is vulnerable, regardless of the quality of its storage medium. Multiple copies of collections are less vulnerable but still deteriorate over long periods, as sometimes random errors will coincide to cause a permanent loss where all copies of a document have been corrupted. Simulations demonstrate that, without regular auditing and repair, no *reasonable* number of copies will prevent significant document losses over a long period See **Figure 3**, which shows expected losses with various numbers of multiple unaudited copies.

In addition, we simulated environmental disturbances such as variations in temperature, humidity, dust, etc., due to HVAC or electrical problems that act to accelerate sector failure for a period of time. We find that the effect of such “glitches” has the same effect as directly increasing the base error rate. Moreover, this similarity implies that preservation strategies (such as those we

recommend below) that are robust across a range of sector failure rates are also robust to moderate to severe glitches that increase the sector error rate by factors of three to ten times.

Estimating a range of deterioration rates of sectors

At what rate do disk sectors deteriorate and lose document information stored on them? Very little direct information is available from manufacturers or from large disk consumers such as cloud storage facilities regarding either theoretical or empirically measured error rates at the sector, file or collection levels. Most published claims regarding storage reliability are either so ill-defined as to be unmeaasurable (see e.g., Mellor 2018 on AWS and Azure claims of "sixteen nines" of reliability), entirely theoretical (see. e.g. Rosenthal 2010, on the calculation of mean-time-to-data-loss by storage manufacturers), or measure the failure of entire drives during a service lifetime. Based on these latter, evidence-based estimates, hard-drives fail at the rate of roughly 1.25-4% annually (see Backblaze 2018) -- when deployed professionally.[1]

If no redundancy (e.g. RAID, erasure codes) is deployed at the filesystem level, the observed disk failure rate would be an upper-limit on sector life as well -- if the disk cannot be read, no sector within it could be read. Based on the lowest observed rate of 1.2% annually, and typical drive sizes of 1-2TB, the implied maximum sector half-life of a non-RAIDED system would be 250,000KH. However, this number is implausibly *low* in an environment in which some file-system redundancy is used -- at this level we would observe frequent losses of large files. For realism, we assume that RAID or other filesystem is deployed effectively to the extent that the system can be run in production without obvious failures.

Closed-form calculations based on the assumption of Poisson arrivals of sector errors can be used to approximate sector reliability based on experience with small storage systems.[2] How high a sector failure rate would the average computer user tolerate on, say, a system disk? Under a reasonable set of assumptions about document size and sector size, we find that half-lives for megabyte-size disk regions less than about 100 megahours (100E06 hours) would result in a noticeable proportion of files being lost in the first few years of operation, for example, more than two percent of documents in just the first two or three years of disk use. Such losses would render an operating system disk unusable, and would certainly be too high for a data (document) archive. At the other end of the reliability spectrum, sector half-lives more than approximately 1,000 megahours would characterize disks that were nearly perfect and very long-lasting. In the absence of sound empirical data on this topic, we consider the range of sector half-life from 100 megahours to 1,000 megahours to be a plausible range for commercial disk drives that corresponds to real-life experience. However, due to the possibility of environmental glitches (failures of air conditioning, noisy electrical power, etc.) that increase error rates, we would extend the realistic range down to 20 or 30 megahours, as shown in the shaded region of Figure 2B.

[1] Electronics generally experience accelerated failure rates during their initial burn-in period, and after their service lifetime (see e.g. Xin, et al 2003). And most hard drives are designed for a service life-time of 3-5 years.

[2] To simplify calculations and to make our results more accessible, we have simplified some of the numbers and units in the simulations and graphs. Error rates for disk sectors and for servers are stated in half-lives in units of kilo-hours and mega-hours. Also, simulation event time periods are stated in "metric years" of 10,000 hours. We feel that this change in the length of a year makes the results slightly more conservative, since the simulation's year and ten-year periods are somewhat longer than calendar years. Also, to save compute cycles in simulations, the collection size is set to 10,000 documents; results are easily extrapolated to larger collections.

The Need For Auditing

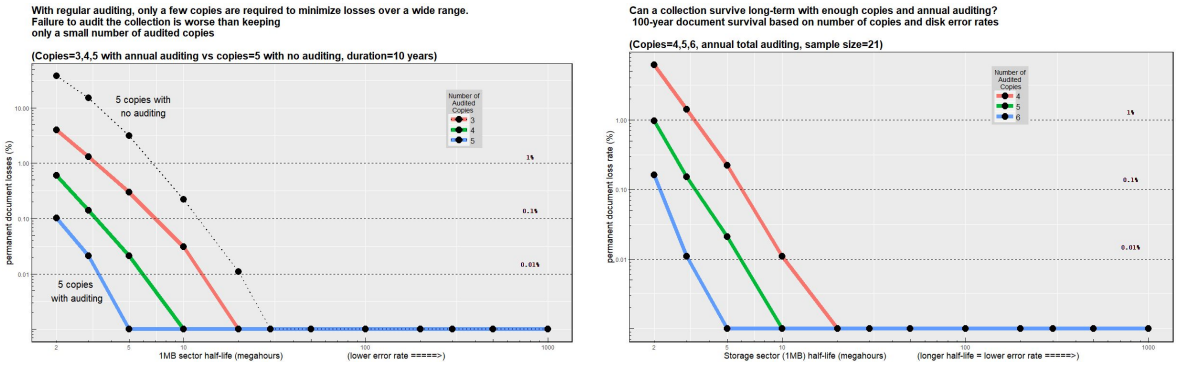
We must adopt active auditing strategies to detect and correct errors in data in order to preserve the corpus over long periods. Such strategies must include three components: detecting errors by fixity information or other means; correcting errors, usually by replacing a damaged copy with a known good copy; and actively locating errors by patrolling through the data to examine all the documents. [CITE]

We need to be wary that failures -- of documents or servers -- are *silent* to the client. A client cannot afford to wait until a document is requested to discover that all copies have been lost. An auditing system must actively search through the data for latent errors in order to locate (and repair) them before these errors pile up and overwhelm the redundancy of storage.

An auditing cycle may be accomplished in a single pass through the documents or broken up into several "segments." It is important to note that total auditing requires that all copies of a document be checked during each auditing cycle. A document may be assigned to any segment within a cycle, but it must be present in some segment of each cycle. The sampling of documents for each segment of the auditing can be systematic or random; but it is important that the auditing actually examine *all* documents. That is, auditing segments must sample documents *without replacement* each cycle. Sampling with replacement permits some documents to be missed in each cycle and reduces the effectiveness of auditing.

Random auditing is often expressed as, for instance, "audit ten percent of the documents every month." The difference between this random strategy and segmented auditing is that the random selection may be chosen *with replacement*. Thus it is likely that some documents will escape auditing during each cycle. This observation is analogous to the experiment of throwing a thousand balls randomly into a thousand urns. Since the balls arrive according to a Poisson distribution, some of the urns will receive no balls, some one, and some more than one, according to distribution. Documents that are audited zero times are not being audited effectively at all and thus are vulnerable to undetected loss.

Figure 3 illustrates the effectiveness of simple annual auditing across a very wide range of sector failure rate over a long period.



(3A) (3B)

Figure 3: Simple annual auditing and repair of a collection greatly reduces document losses. (3A) Note that just three copies with annual auditing is more robust than five unaudited copies. (3B) Even over very long periods of time, e.g., 100 years, simple annual auditing can protect a collection with a modest number of copies. Over the plausible range of sector lifetimes, five copies with auditing suffer minimal or zero losses.

The pattern illustrated by the figure (and confirmed through additional sensitivity analysis) demonstrates that a strategy of deploying five independent replicas and simple annual auditing is sufficient to maintain the integrity of collections for over a century across any practical variation in storage quality conditions. In the full paper, we extend these results to include robustness to higher-level correlated threats, and to analyze more sophisticated and efficient preservation strategy.

Constructing a Preservation Strategy that is Robust to Institutional Risks

Strategies against limited server lifetimes or shocks

Cloud-based storage services provide very reliable storage, largely through the use of error-correcting storage methods, such as RAID, erasure coding, and high-count replication. These error-correcting techniques make the failure rate for disk sectors largely irrelevant. Such storage, properly maintained, is effectively "immortal" and will suffer no significant losses over long periods.

Let us shift our analysis from the reliability of storage media to the reliability of storage services. We will find that similar analysis and similar storage techniques can be used to protect documents and entire collections.

Storage services themselves are not immortal. Services as corporations may fail over time; they may merge and thus lose their independence; they may be subject to physical trauma through natural events or political or economic disturbances. And access to services may be blocked by government action, by cyber attack, or by administrative error. In any case, it is important to note that such failures are generally silent: the client is not actively informed of the failure. The client will notice the failure only when trying to access the stored documents.

There are many ways that a server can fail and render a copy of the collection inaccessible. Table nnn lists a number of conditions that can cause one or more servers to fail. Let's consider just two causes of server failure: first, that servers as corporations have finite lifetimes; and second, that exogenous physical events or government actions may make it impossible for a server to continue to function. We note that a shock that raises the rate of failure of a single server is equivalent to a server with reduced life expectancy.

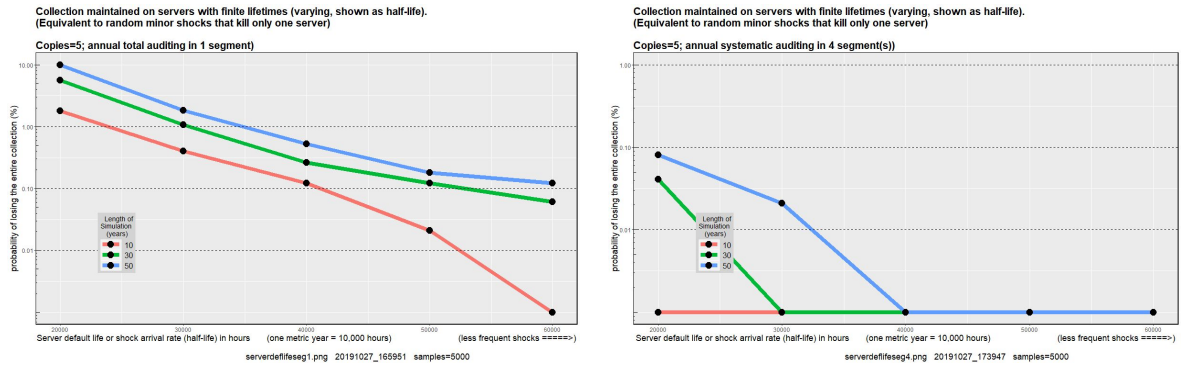
To protect a document collection, a client library must actively test, and if necessary repair, the integrity of document collections stored remotely on such services. Server failures, regardless of the cause, are almost always silent to their clients: earthquakes, floods, wars, mergers, bankruptcies, and government censorship actions do not give notice to the parties affected. A client must examine all servers on a regular basis to verify the presence of the document collection stored there. The

client must actively audit storage services to verify that they are still alive and still have the collections. Due to the assumed high reliability of storage within a server, we assume that a server contains all documents or none. If a service is still alive and contains any documents, then the service is still available as one of the client's replication instances. If a service is found to be inactive, then, to maintain the target number of replications, the client must find a new service and populate that with the collections.

This process is clearly analogous to the auditing of individual documents to repair documents corrupted by sector failures, but in this case, it is merely the presence of the server that is to be tested. We find also that the auditing process can be much more efficient than with sectors: to verify that the server is alive, only a few documents need to be retrieved. Either the server is dead and contains no documents, or it is alive and contains all documents.

When a client detects a server failure of this sort, the redundancy of the collection storage is (temporarily) reduced. The client must find another server to hold a copy of the collection and then transmit the entire collection to that server. Only then is the collection fully replicated with the desired number of copies.

The simulation framework demonstrates that server lifetime is an important risk factor. The replication strategy previously developed (shown in Figure 3), although robust to sector failures and glitches, will fail (that is, result in collection losses) if the server failure rate is high. For that simple strategy -- five copies audited annually -- to preserve a collection over long periods of thirty to one hundred years would require servers to have half-lives of at least eight to ten years. To require all servers to have such long life expectancies is optimistic. However, on the bright side, collection loss can be prevented even with relatively short-lived servers by substantially increasing the auditing frequency and adding additional replicas.



(4A) (4B)

Figure 4: The impact of auditing speed on preservation of a collection. If servers have finite (random) lifetimes, due to economic downturns, mergers, etc., failures can result in losing the collection. (The lifetimes used here are short, to illustrate the effects.) (4A) Annual total auditing shows a significant risk of losing the entire collection over a range of short but plausible server lifetimes. (4B) Annual auditing in four quarterly segments -- so that every server is verified four times a year rather than only once -- dramatically improves the survival of a collection. Increasing the speed of auditing, say, in monthly segments would further improve the likelihood of survival.

Strategies against correlated server failures

The strategy above is robust to server failure -- when servers fail independently. However, server failures may be correlated in two ways. We model these correlations formally, through introducing "shock" events of varying types, severity, and frequency.

We use two statistical approaches to model shocks. First, we model shock that *decrease the expected lifespan of some or all servers.* Second, we model shocks that *cause immediate failure of multiple servers simultaneously.* Our simulations included a range of "shock" conditions that contribute to server failures; modeling such threats as recession, targeted censorship, administrative error.

For the most part, our results show that the most important factors are the frequency of shocks and their "span," that is, the number of (correlated) servers affected by a shock. Overall, shocks that raise the rate of single server failure have an impact equivalent to reducing the expected lifespan of all servers. That is, whether the cause of a server's premature failure is exogenous, through economic or political pressure, or endogenous, due to financial instability, the result is the same: a single server fails at some random time with some frequency independent of other servers. This failure reduces the redundancy of the storage of the collection and therefore increases the risk of collection loss.

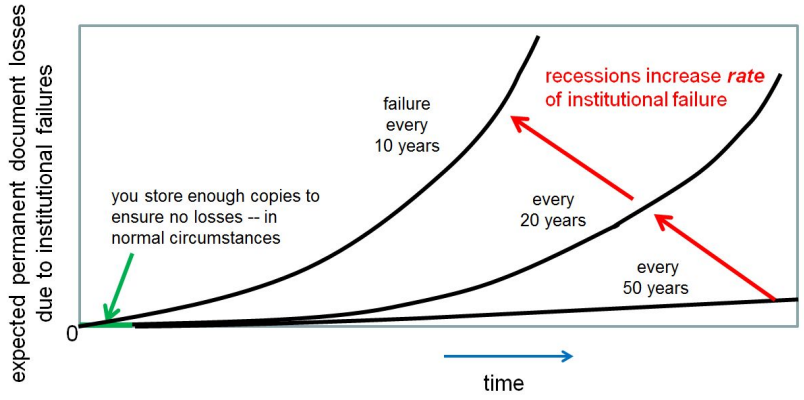
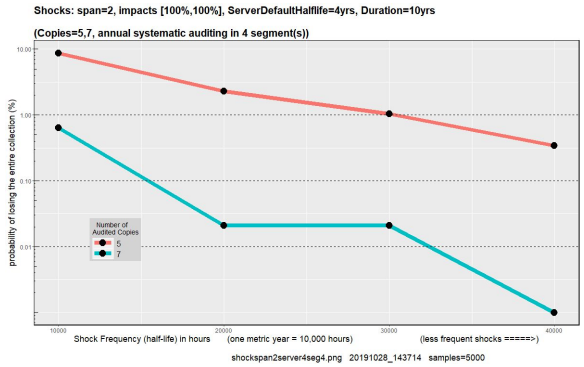


Figure 5: The effect of economic pressures on institutional failures. For any chosen level of redundancy that protects the collection in normal circumstances, economic recessions will increase the rate at which the servers that keep copies fail. Severe recessions will cause institutional failures sooner than mild recessions, but still, over time, some institutions will fail. Without a strategy to detect and repair these failures, the redundancy of collection storage is reduced and the likelihood of collection loss is increased.

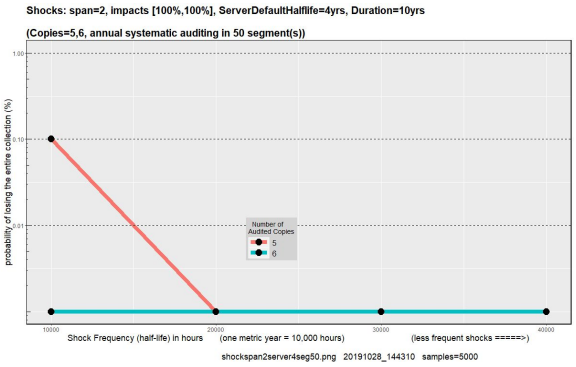
In contrast, shocks that cause multiple simultaneous failures can dramatically increase the likelihood of collection loss. In our simulations, we have found that the major protective factor against correlated failures is the speed of detecting a dead server. The client should test all the servers for responsiveness very frequently, such as quarterly or monthly; in very severe cases, testing should be done even faster, e.g., weekly .

Again, it is important to note here that it is not necessary to test the entire collection, or even a large segment of it, during each audit. Since a server either contains all documents or contains none, the client need only probe a few documents during each audit to determine the health of the server. This dramatically reduces the bandwidth requirements and increases the speed with which a client can test all its servers.

Thus a strategy of having X copies audited every Y period can be robust even in the presence of frequent widespread shocks: such a strategy is likely to withstand both major recessions and minor wars. As **Figures 6 and 7** below illustrate, protecting a collection under conditions of frequent large shocks, economic or political, may require increasing the redundancy of storage to seven or eight copies, and increasing the frequency of auditing of servers from quarterly or monthly to weekly.



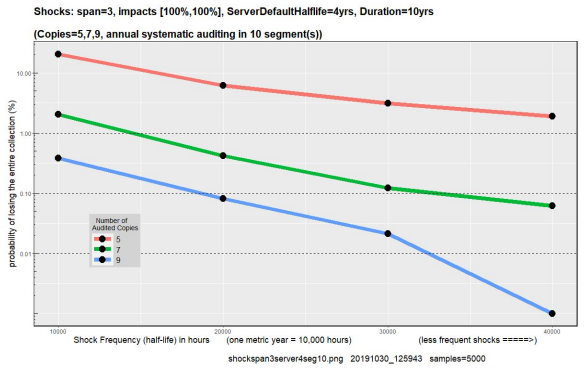
(6A)



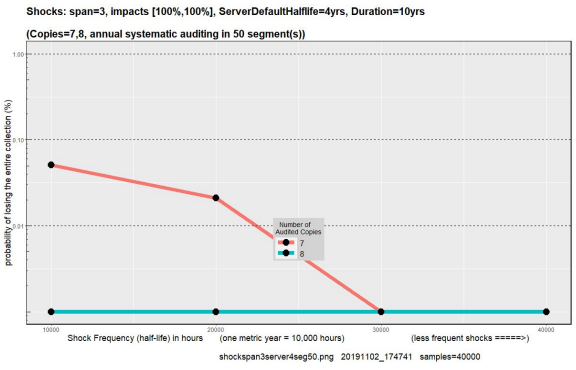
(6B)

Figure 6: The impact of moderately severe correlated failures of servers over ten years. In these cases, shocks that cause two servers to fail immediately occur randomly with half-lives shown on the X axis. (6A) Five copies of a collection with quarterly auditing is not sufficient to protect against a high probability of total loss. Increasing the redundancy to seven copies also does not protect the collection sufficiently. (6B) Accelerating the rate of auditing, e.g., to weekly segments, and possibly increasing the redundancy level slightly, can improve the survival of the collection even under these severe conditions. Recall that auditing to determine the presence of a server requires interrogating only a few documents on each server.

Correlated Shocks Span=3: severe shocks (large-ish war, depression)



(7A)



(7B)

Figure 7: The impact of very severe and frequent correlated failures of servers over ten years. In these cases, shocks that cause three servers to fail immediately occur randomly with half-lives shown on the X axis. Much more frequent auditing of more copies is required to protect a collection. (7A) For shocks where three servers are removed from service, even frequent monthly auditing with many more copies, up to nine, cannot protect the collection from total loss. (7B) Accelerating the auditing to weekly and increasing the redundancy to eight copies may suffice to protect a collection from total loss even under these extremely harsh conditions.

We note that a combined strategy might be a good choice: Audit the collection on an annual cycle in weekly segments; every week choose two percent of the collection, selected *without replacement*, to be audited. Replace any servers found to be not functioning with new servers, and repopulate the new servers with the entire collection. Using this strategy, redundancy of five copies should protect the collection in all but the most dire circumstances. Increasing the redundancy to seven or eight copies should protect the collection even in very severe upheavals.

Corollaries - Applications of the Multi-Level Model to Other Failures

In this section we show how the existing simulation results can be applied to modeling additional threats, including attacks against the auditing system; encryption-key loss and other related correlated failures; and fragility induced by document compression.

Augmenting the Core Replication Strategy with Distributed Auditing

In the model above auditing always returns the correct results, when invoked. In practice, the auditing system itself may fail either due to a fault in the auditing software, or to a malicious attack against the auditing system.

Risks due to unintentional faults in the auditing system may be mitigated by using multiple independent software implementations to perform the proof-of-retrieval. For example, in an auditing system that involves retrieval of the object content, followed by computation of a cryptographic hash on the contents; computation of this hash could be replicated using multiple independent implementations of the hashing algorithm. [CITE] More generally, to prevent loss due to well-resourced attacks against the auditing system itself, a secure auditing system that engages multiple auditors should be used [see Jin et al 2019 for a review]. Further, since in the worst case, a well-resourced adversary could subvert one or more of the auditors, at least $3s+1$ servers will be needed to provide *byzantine fault tolerance* -- where s is the maximum number of subverted nodes.

Thus in a world without any types of server failures (besides malicious attack), deploying seven servers would be sufficient to protect against subversion of two. Distributed preservation systems such as LOCKSS, have each server play the role of both a content replica, and an independent auditor. When using distributed preservation, or in other systems in which auditing servers can fail at random, additional replicas will be needed to ensure that sufficient *uncorrupted* replicas remain. For example, if there are two potentially subverted servers, and external shocks can destroy three servers simultaneously at random, then post-shock, two out of four of the remaining servers could be corrupted. Thus with a shock-span of three, ten replicas are necessary to prevent against orchestrated attacks by two subverted noded on the auditing system that are launched following a failure. Eight copies would be sufficient, if the adversary controls at most one auditing node.

Applying Analysis to Replication of Encryption Keys

We identified loss of encryption keys (or more generally of shared secrets -- where at least n are needed to reconstruct the key) as a threat to content above. In this section we discuss strategies for evaluating the impact of encryption key loss, and mitigating these risks. As it turns out, it is not necessary to add encryption directly to the discrete simulation model described above -- we can evaluate this risk using the existing model.

Encrypting a collection of content creates three additional threats of loss. First, and most important, losing all copies of the encryption keys for the collection effectively results in a loss of all replicas of the collection -- while the bits comprising such collections may continue to exist, they are rendered meaningless. Second, if the knowledge of the encryption algorithm is lost, the collection is likewise destroyed. Third, encryption may make documents more fragile -- a single block loss will destroy the entire document rather than a portion.

The last threat (fragility) has a minor impact (see the next section) and is readily mitigated, if necessary, by adding an additional copy to the replication scheme. The second threat (algorithm loss) can be effectively mitigated by selecting a well-known standard encryption algorithms -- standard algorithms are widely documented, and independently replicated. Thus, we focus on the threats from loss of encryption keys.

Loss of encryption key may be modeled by treating the keys as a small, separate collection of documents. As we have shown above -- mitigating risks of loss to a collection requires replication, auditing, and repair. For security reasons, encryption keys should be kept in separate administrative domains than the content they encrypt. We recommend that a separate set of 'servers' be used to replicate the collection of encryption keys.

Because the size of the collection is small (encryption keys are very small relative to the content they protect), risks to the collection of keys will be dominated by 'shocks' that disrupt organizations and affect multiple replicas. For example -- wars, economic recessions, and government actions may lead to organizational failures -- thus we recommend maintaining at least four copies of the encryption key and auditing the integrity of the keys (e.g. a challenge mechanism) at least monthly.

Modeling Risks of File-Format Obsolescence

At first blush, file-format obsolescence appears to present a radically different form of risk that the bit-loss modeled. After all a collection may be lost to format-obsolescence even if all of the replicas are intact -- bit-level auditing does nothing to prevent this.

It is possible to model file-format obsolescence in a way that parallels server-level failure, and thus use server- and shock- analysis to estimate risks of loss, and to develop mitigation strategies. In this model:

- Formats represented as individual collections, containing a test corpus of documents in that format.
- Servers represent independent versions of format readers -- software that is able to read and semantically validate objects of that format. (Readers should be independent within a format, but a single reader may be used for multiple formats.)
- Format obsolescence is represented by server failure. If a reader can no longer be executed it has “failed”. If all readers fail, the collection is unintelligible, and the format is lost.
- Sector- and glitch level errors are ignorable. (Following similar logic as used in the analysis of encryption key loss above.)
- Server failure is discovered through auditing. Auditing a server consists of executing the corresponding reader against the test corpus.

Unlike storage systems, there is less evidence to assess the distribution of failures of software readers -- it is likely that software failures follow a modified bathtub curve. In this case the assumption of an exponential distribution of reader failure would yield overly optimistic predictions for loss. Notwithstanding distributional assumptions, by conditioning on using format readers that are established (past their infant mortality period), modeling with a conservative (lower than expected) expected lifetime, and modeling shocks that cause multiple readers to fail simultaneously, we can generate strategies that are robust to format failure. In particular, based on a conservative lifetime of X-years, and 3-span shocks with a frequency of y years, regular testing with *Z* readers, accompanied by migration of formats, when less than Z readers can be identified, should be sufficient to protect against format failure indefinitely.

Modeling Compression Risks

Documents stored on digital media are fragile; storage errors corrupt the content of a document. How much of a document is corrupted depends largely on the data format of the document. Even small errors in highly compressed or encrypted documents may render part or all of the document unusable.

For documents that might not be fatally corrupted by a single sector error, lossless compression of the document involves a clear trade-off. A smaller document is a smaller target for a randomly occurring error, but a highly compressed document is more fragile. A small error in an audio or video file, or an uncompressed text file, might not be fatal to the document, but a highly compressed text document (or an encrypted document) might be lost.

In these simulations, we have modeled documents as very fragile: one sector error causes the document to be judged as lost. In this model, at least these two considerations should be included in the decision to compress documents.

- Smaller is safer. A smaller document presents a smaller target for random errors. If a document is compressed, say, by 90%, that is, to 10% of its original size, then a random error is only one-tenth as likely to strike that document. When placed on a storage medium of any given quality level, that smaller, compressed document is likely to persist without error ten times longer than the uncompressed version.
- Smaller is less expensive. A stored collection incurs costs for both storage of the document images and the bandwidth used in auditing and repair. Smaller documents consume less space and less bandwidth and therefore cost less to maintain. On a given budget, a compressed collection can be replicated into more copies and audited more frequently. Both the increased copy count and more frequent auditing contribute directly to reducing or eliminating permanent losses in the collection.

The linear increase in document losses based on size is to be expected from straightforward Poisson calculations. In addition, we ran simulations over a wide variety of conditions to verify that this linear relationship holds for multiple copies of collection documents, various auditing strategies, and a very wide range of storage quality (error rates, sector lifetimes).

Compression offers another major advantage: potentially higher redundancy. If compression reduces a document's size by, say, 50%, then a client can store two copies of the document for the same cost in storage. That extra copy provides higher redundancy and thus greater resistance to document loss. On a fixed budget, a client can store additional copies of documents depending on how effective the compression algorithm is. High compression permits more copies to be replicated to offset any increased fragility of a compressed document. Text and image compression are particularly effective in this regard.

Viewed in a slightly different way, compression can increase the "repairability" of a document in the following sense. In our model, document copies are "repaired" by being replaced on their servers from other copies; that is, the "repairability" of a document copy depends on the presence of one or more valid copies stored elsewhere. If compression permits an additional copy or copies of a document to be stored, then there will be more copies from which a "repair" can be effected when one copy fails and needs to be "repaired." For example, if five copies of a collection are to be stored on servers, then a mere 20% reduction in size due to compression would permit one additional copy to be stored and maintained within the same budget. That additional redundancy, six copies instead of five, would make the document more resistant to failure.

Finally, compression permits more aggressive auditing, to protect a collection, without increasing costs for bandwidth and server egress. Smaller, compressed documents can be retrieved more quickly without increasing bandwidth, and consume less bandwidth and less egress charge from the storage vendors. Auditing of the collection can be done more frequently on the same budget, which improves document survival rates.

Compression Sometimes Increases Fragility -- But Not Too Much

A disadvantage of compression is that it may make documents more fragile. We define fragility as the proportion of the document (and document's value) that is eliminated by a single block failure -- we refer to this proportion as the fragility index.

For simplicity in the rest of the article we have assumed that documents were completely fragile (i.e. a fragility index of 1) -- blocks failures cause an entire document to be lost. However, it is trivial to model losses for a collection of documents with a uniform fragility index: These losses are mathematically equivalent to the losses found in a collection of fragile document, that is, has 1/F times the number of documents each of which is S/F in size.

Using this fact we can readily estimate the combined effects of compression. Compression will shrink the document as a whole -- reducing the likelihood of a block failure affecting it; increase the amount of storage available for replication; and multiply the fragility of a document by some factor CF (bounded by 1)

Where is compression a win? First, wherever C<F compression is a win based on the target-reduction effect alone -- even if each of the compressed documents is completely fragile. Moreover, when C>F compression will still be a win in most circumstances -- except where the fragility index is very low, and compression makes the document much more fragile without substantially reducing size, as illustrated in the figure below.

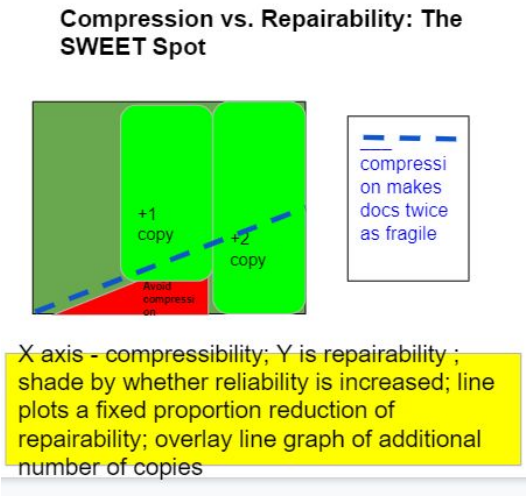


Figure X: Loss for a collection of 10,000 documents of initial size ZZ, a sector failure rate of XX, and Y replications, based on compressability and repairability. Red shaded region shows small region where compression increases loss and is not recommended.

In summary, we consider lossless compression to be benign for a variety of reasons.

- Smaller documents are smaller targets for errors. They are less likely to corrupted than large documents.
- Smaller documents permit higher storage redundancy without increasing costs, thus offering greater protection for the documents in a collection.
- Smaller documents can be audited more frequently to protect the collection.
- Overall, while compression may increase the fragility of an individual document, it can greatly increase the survival of an entire collection without increasing costs. In all but a few extreme cases, the trade-off favors compression.

Discussion

The results above highlight a number of robust and broadly applicable operational preservation policies: for example, these results demonstrate that the commonly used strategy of sample-based auditing is ineffective; and that the risks of compression-related fragility noted by the preservation community, are typically more than offset by reductions in the efficiency of replication and auditing. More generally, we identify simple preservation strategies involving diversification, 5-7 replicates, and auditing partitioned weekly across every year, that are robust both to variations in storage quality and conditions; and robust to correlated organizational threats, including global recessions and regional wars.

This analysis demonstrates that the most critical source of risk to collections is shared vulnerabilities across services that can result in multiple simultaneous failures. Curators who choose services for replication need to be wary of characteristics that results in shared vulnerabilities. These include geographic location of server infrastructure; legal jurisdictions to which the server is subject; and shared technical dependencies across servers. It is particularly important that service providers disclose in a verifiable way the extent to which they rely on other third-party storage vendors to host content stored with them.

Further, these results underscore the need to increase the efficiency of external auditing, since auditing frequency is critical to robustness. Currently, a bottleneck to auditing is the need to transfer the content of a document to be audited from a server to a trusted source. If storage services

provided an API for trustworthy verification of a document directly, costs and time would be substantially reduced. For instance, if a storage service offered the ability to compute, on demand, a combination cryptographic hash and nonce for a selected portion of a document, reliable external auditing could be completed with greatly reduced network bandwidth. (See Lin, et al. 2019 for a survey of cryptographic auditing approaches to cloud replicas.) Stakeholders in both the storage and preservation sectors would benefit from developing standards API's for cryptographic verification of stored content -- which increase both the trustworthiness of cloud services and the use of external auditing.

Moreover, the framework we present offers flexibility for the modeling of a wide range of preservation policies and threats. Since this framework is open source and easily deployed in a cloud computing environment, it can be used to produce analyses based on independent estimates of scenario-specific costs, reliability, and risks. We invite the community to probe our results by calibrating risk profiles based on their own context and to use the system to estimate the costs and losses of their preferred preservation strategies.

Acknowledgments

The authors describe contributions to this article using a standard taxonomy. [Allen, et al. 2014] MA provided the core formulation of the research goals and aims and methodology. RL led the data collection, software design, and data analysis, MA and RL shared the writing of the original manuscript. All authors contributed through commentary, review, editing, and revision.

References

Allen L. , Jo Scott, Amy Brand, Marjorie Hlava & Micah Altman, Publishing: Credit Where Credit Is Due, 508 Nature 312 (2014).

Altman, et al., National Agenda for Digital Stewardship, National Digital Stewardship Alliance (2015)

Baker, M., Shah, M., Rosenthal, D. S., Roussopoulos, M., Maniatis, P., Giuli, T. J., & Bungale, P. (2006, April). A fresh look at the reliability of long-term digital storage. In *ACM SIGOPS Operating Systems Review* (Vol. 40, No. 4, pp. 221-234). ACM.

Mellor, C., Mmm, yes. 11-nines data durability? Mmmm, that sounds good. Except it's virtually meaningless, The Register (2018)

Gallinger, M., Bailey, J., Cariani, K., Owens, T., and Altman, M. (2017).Trends in Digital Preservation Capacity and Practice: Results from the 2nd Bi-annual National Digital Stewardship Alliance Storage Surve

Lebrecht, A. S., Dingle, N. J., & Knottenbelt, W. J. (2011). Analytical and simulation modelling of zoned raid systems. The Computer Journal, 54(5), 691-707.

Li, Y., Miller, E. L., & Long, D. D. (2012, June). Understanding data survivability in archival storage systems. In Proceedings of the 5th Annual International Systems and Storage Conference (p. 16). ACM.

Lin, Y, Li, J, Jia, X, Ren, K. Multiple-replica integrity auditing schemes for cloud data storage. Concurrency Computat Pract Exper. 2019;e5356. https://doi.org/10.1002/cpe.5356

E. Pinheiro, W.D. Weber, L.A. Barroso, Failure trends in a large disk drive population. In Proceedings of the 5th USENIX Conference on File and Storage Technologies (FAST '07) (2007).

D. S. H. Rosenthal, T. Robertson, T. Lipkis, V. Reich, and S. Morabito, "Requirements for Digital Preservation Systems", D-Lib Magazine 11 (11), (2005)

Rosenthal, David SH. "Bit preservation: A solved problem?." International Journal of Digital Curation 5, no. 1 (2010): 134-148.

Rosenthal, 2010, Keeping Bits Safe: How Hard Can It Be? https://queue.acm.org/detail.cfm?id=1866298

Xin, Qin, Thomas JE Schwarz, and Ethan L. Miller. "Disk infant mortality in large storage systems." 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems. IEEE, 2005.