FP7-ICT-2007-3-231161

Deliverable D2.1.2

Tools for modelling and simulating migration-based preservation

Matthew Addis, Mariusz Jacyno (IT Innovation Centre)

22-December-2010

Document administrative table

Document Identifier	PP_WP2_D2.1.2_PreservationModelling Tools_R0 Release 0
Filename	PP_WP2_D2.1.2_PreservationModellingTools_R0_v1.00.pdf
Workpackage and Task(s)	WP2 Models and environments for long-term audiovisual content preservation WP2T1– Models for Audiovisual Preservation
Authors (company)	Matthew Addis, Mariusz Jacyno (IT Innovation Centre)
Contributors (company)	Andrew Baker (IT Innovation Centre)
Internal Reviewers (company)	Richard Wright (BBC) Alberto Messina (RAI) Jeff Ubois (B&G)
Date	22/12/2010
Status	Release
Type	Deliverable
Deliverable Nature	Prototype
Dissemination Level	Public
Planned Deliv. Date	31 December 2010
Actual Deliv. Date	22 December 2010

Abstract

This report describes two tools for modelling and simulating the costs and risks of using IT storage systems for the long-term archiving of file-based AV assets. The tools include a model of storage costs, the ingest and access of files, the possibility of data corruption and loss from a range of mechanisms, and the impact of having limited resources with which to fulfil access requests and preservation actions. Applications include archive planning, development of a technology strategy, cost estimation for business planning, operational decision support, staff training and generally promoting awareness of the issues and challenges archives face in digital preservation.

DOCUMENT HISTORY

Version	Date	Reason of change	Status	Distribution
0.1		First Draft	Outline	Confidential
0.4	12/12/2010	Draft for internal review	Draft	Confidential
0.6	21/ 12/2010	Update responding to internal reviews by: Richard Wright (BBC) Alberto Messina (RAI) Jeff Ubois (B&G)	Final Draft	Confidential
0.7	21/12/2010	Correction of minor typos and formatting issues following proof read.	Release	Confidential
1.00	22/12/2010	Final editing for publication	Release	Public

Table of contents

Scope

The European Commission supported PrestoPRIME project (http://www.prestoprime.eu) is researching and developing practical solutions for the long-term preservation of digital media objects, programmes and collections, and finding ways to increase access by integrating the media archives with European on-line digital libraries in a digital preservation framework. This result will be a range of tools and services, delivered through a networked Competence Centre.

This report describes two tools for modelling and simulating the costs and risks of using IT storage systems for the long-term archiving of file-based AV assets. One tool is for long-term planning, e.g. selecting a storage strategy and the other tool is for more detailed investigation of the likely behaviour of a given storage strategy.

The tools include a model of storage costs, active use of the archive in terms of ingest and access to files, the possibility of data corruption and loss from a range of mechanisms within the archive, and the impact of having limited resources with which to meet user demands and preservation actions.

Applications of the tools include planning, strategy development, cost estimation, operational decision support, staff training and promoting awareness of the issues and challenges archives face in digital preservation.

These tools are the first in a series of tools that IT Innovation plans to release for use by the AV community through the PrestoCentre. Future tools will cover a much wider range of preservation and access processes, e.g. digitisation and quality control as part of transfer chains, or the impact of file-format selection and migration strategies.

Executive summary

The European Commission supported PrestoPRIME project (http://www.prestoprime.eu) is researching and developing practical solutions for the long-term preservation of digital media objects, programmes and collections, and finding ways to increase access by integrating the media archives with European on-line digital libraries in a digital preservation framework. This result will be a range of tools and services, delivered through a networked Competence Centre.

Part of the work of PrestoPRIME is to investigate digital preservation strategies, provide guidelines on what technologies to use, and support archives so they can take informed steps into the largely uncharted territory of file-based digital preservation of audio-visual material. This work includes reports and tools that help archives decide on which technologies to use, e.g. storage systems and file formats, the risks associated with using them for long-term preservation, and what the Total Cost of Ownership is likely to be.

Work by PrestoPRIME has already produced a range of reports that provide guidelines on the strategies, costs, risks, state of the art, and future directions for digital preservation of AV content. These include D2.1.1 Preservation Strategies, ID3.2.1 risks from use of mass storage technology, ID3.2.1 service oriented models for audiovisual storage, and D7.1.3 audiovisual digital preservation status report. D6.3.1 Business models and calculation mechanisms and D7.1.4 audiovisual digital preservation status report 2, both due at the same time as this report, provide an update to some of the earlier work of this project.

This report describes the results of work by IT Innovation that moves beyond these reports and guidelines into providing practical software tools for making quantitative predictions on the costs and risks of loss of using IT storage solutions for digital preservation. Storage solutions in this context can include manual operation of an archive's storage using 'items on shelves' as well as the use of automated hardware/software systems, e.g. data tape robots or hard disk servers. As an input to the tools, the user describes how much content they have, how it will be stored, how often it will be accessed, how much new content will be added, what resources are involved and what these cost. The tools then create projections of cost and loss over time, and allow the user to interact during the simulation, e.g. to explore the impact of changing operational polices or to examine contention for resources or to discover the main cost and loss factors.

The objectives of the tools are not to provide exhaustively detailed models or highly accurate costs. Instead the aim is to provide simple results that are meaningful and useful in specific contexts, e.g. when selecting storage strategies or supporting day-to-day decision making. This is part of a balance to be struck between accuracy, ease of use, flexibility to cover a wide range of scenarios, and likelihood of users being able to provide the necessary inputs needed for the tools to work.

These tools are the first in a series of tools that IT Innovation plans to release for use by the AV community through the PrestoCentre. Future releases of the tool (first quarter 2011) will cover a much wider range of preservation and access processes, e.g. digitisation and quality control within transfer chains, and the impact of file-format selection and migration strategies.

1 Introduction

The mass digitisation of analogue archive holdings plus the transition to tapeless production for new content means that AV archives now face the prospect of file-based archiving using IT storage solutions.

But what is the long-term Total Cost of Ownership (TCO) of these systems, which file formats should be used, what storage technologies make sense, what are the risks involved, what is the additional cost of managing these risks, and what new management approaches can be applied?

Previous PrestoPRIME deliverables provided some qualitative answers to these questions and suggestions on how cost and loss calculations could be done This report describes software tools that allow quantitative simulation and modelling of preservation using IT systems.

The cornerstone of our approach is to recognise that modelling and simulating preservation activities or processes, e.g. storage of files or migration between file formats, necessarily has to consider the compromises that are inevitable in practice (limited time, budget, capacity). The real value of simulation and modelling tools is to allow these compromises and trade-offs to be investigated. There is no one size fits all solution and each archive will want to explore the options and choose the one that best fits their particular needs.

The report has the following structure:

Section 2 explores the concept of 'cost of compromise', e.g. cost of risk of loss when selecting archive storage technology, or the cost of throughput and quality when designing digital transfer and quality control chains.

Section 3provides an overview of the tools that have been built in terms of their objectives and anticipated application areas.

Sections 4and **5**give a walkthrough of the tools including a series of screenshots to show how they might be used in practice.

Section 6 describes how to calculate the input values needed by the tools, e.g. costs of storage and the probability of file loss.

Section 7 describes the simplifications and assumptions made as design decisions when the tools were developed in order to achieve a balance between simplicity, accuracy, flexibility, ease of use and implementation in the time available.

Section 8 provides details of the tools including their implementation.

Section 9 summarises the results of the work and outlines the next steps.

The tools described in this report will be available in March 2011 from the Presto Centre website complete with user guide and installers.

2 The ‘cost of compromise’

Preservation uses resources, has a cost, and is not without risk. Those responsible for designing and operating preservation systems and processes naturally have a strong interest in the cost (annual and long-term), for example as part of making business cases for investment into archive infrastructure (people, equipment, space etc.)

Much work has already been done on cost modelling (see PrestoPRIME D2.1.1 and D6.3.1 for some examples) but little of this work investigates the trade-offs that exist between cost and other important factors such as safety, quality and throughput in preservation systems.

For example, when storing files, it is possible to design a highly reliable preservation system for storing uncompressed master quality content – but this comes at a high cost. Likewise, when digitising analogue material, it is possible to design a high throughput transfer and quality assurance chain using human operators for extensive quality control, but this requires significant staffing and typically has a very high labour cost. In reality, neither is affordable and like most preservation processes there is an element of compromise in order to get costs down to an acceptable level.

The question is what trade-offs to make - what cost savings can be made and what has to be compromised as a result?

The objective of the tools described in this report is to allow quantitative investigation of trade-offs. We anticipate uses of the tools to include:

- **Storage.** What is the cost of risk of loss when using IT storage systems? Lots of Copies Keeps Stuff Safe (LOCKSS) but lots of copies costs lots of money. How do cost and safety relate?

- **Digitisation**. What is the cost of quality and throughput in digitisation and transfer chains? What are the costs/benefits of automated software tools for quality management used alongside or instead of human operators?

- **Access.** What are the costs and benefits of using community annotation in addition to or instead of in-house professional cataloguing?

The first version of the tools focus on the ‘cost of risk of loss’ in storage. Future versions of the tools will focus on the other areas and are expected to be available in March 2011. These set of tools together will cover a wide range of migration scenarios as originally planned for this area of work in the project (e.g. migration of file formats, migration in storage systems, migration from discrete items of media to automated mass storage systems).

2.1 Cost of risk of loss

PrestoPRIME investigated the 'cost of risk of loss' of digital preservation in previous reports. The conclusion is that long-term preservation of audiovisual content is essentially a risk management problem – as described extensively in PrestoPRIME ID3.2.1[1].

It is possible to reduce or mitigate the risks involved in preservation, but at a cost. The issue is establishing an acceptable balance between increased cost and lowered risk of content loss. This is not simple and the outcome will vary over time requiring constant review. For example, the use video compression means less storage space, which in turn means more copies can be held for the same total cost with a consequent increase in safety. However, each copy is more sensitive to data corruption and compressed formats typically become obsolete faster than uncompressed formats and hence require file format migration on a more regular basis. This adds new costs and risks. The total cost of storage is falling rapidly, so the point at which it becomes more cost effective to store uncompressed is a moveable feast. The issue now becomes one of considering not only risks but the long-term trends for the cost of reducing these risks – for example trends for storage, the longevity of file-formats, and the safety of data in IT systems. This approach is shown in Figure 1.

[1] https://prestoprimews.ina.fr/public/deliverables/PP_WP3_ID3.2.1_ThreatsMassStorage_R0_v1.00.pdf

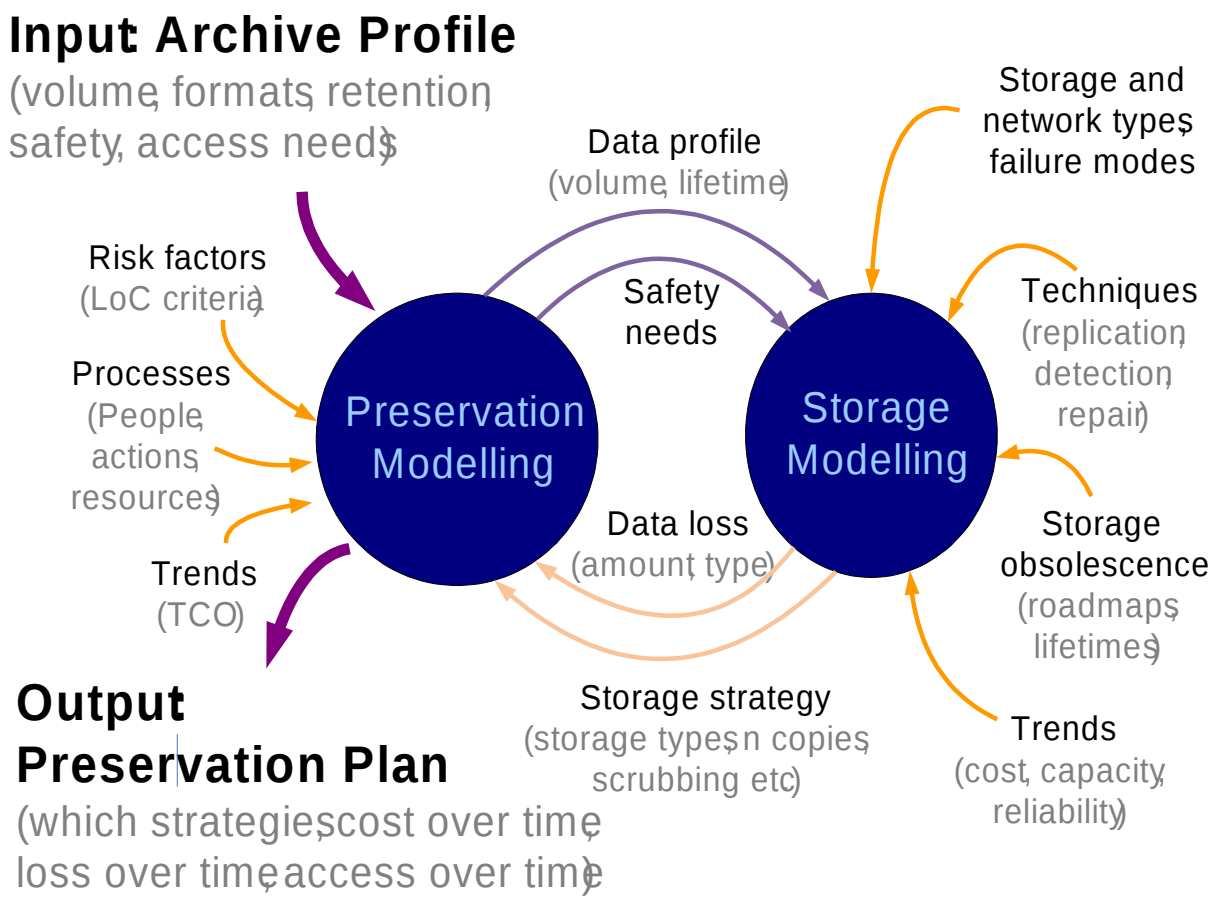

Figure 1 Cost of Risk of Loss assessment and planning

The objective is to convert an archive's needs (how much content it has, how long to keep it, how safe it needs to be, and who needs to be able to access it and how easily) into a preservation plan (what to do, when to do it, what the consequences will be – including accessibility or potential loss of content). Combining preservation modelling (e.g. file level preservation approaches) with storage modelling (bit level preservation approaches) allows the interplay between these two to be considered (e.g. choice of file format impacts on storage required, and formats need evaluating for their sensitivity to data corruption in storage).

Calculations can then be done on when to make transitions, e.g. from data tape to hard disk, from compressed to uncompressed file formats, or even from in-house to remote archive hosting. This depends on budgets, content volumes, retention schedules, frequency and type of access, content value, maintaining in-house skills, and the IT technology used.

2.1.1 File Preservation Modelling

Figure 2 shows a model for file–format migration as an example of preservation modelling. More details can be found in PrestoPRIME deliverable D2.1.1.

The model shows an approach for video file format migration that addresses the question of what format to use and when to migrate between formats. Many long-term sustainability issues are associated with AV file formats, especially modern compressed video formats

used in production and post-production. The issues include tool support, adoption, vendor lock-in, transparency, patents and open standardisation[2].

File format migration is a good example of where preservation decisions at the file level cannot be divorced from the question of how to store those files and hence the need to include a storage model in the decision making process. For example, the cost of online and fully managed storage is currently still prohibitive for high bit rate audiovisual material (e.g. one hour of uncompressed HD or 2k film is approx 1TB with a TCO of $1000 per year when stored on spinning disk when power, cooling, space, maintenance, upgrade are all included in addition to hardware and media[3]), but this is falling fast. Whilst compressed formats (and their associated risks) are often considered the only viable option today, the use of uncompressed formats rapidly becomes viable, e.g. within 5 years. The aim is a lifecycle where compression exists only once at the start, if at all, (i.e. no lossy transcoding) and migration to uncompressed happens as soon as costs permit.

[2] For example, See PrestoPRIME D2.1.1 Preservation Strategies (https://prestoprimews.ina.fr/public/deliverables/PP_WP2_D2.1.1_preservationstrategies_R0_v1.00.pdf) and the Library of Congress Sustainability Factors (http://www.digitalpreservation.gov/formats/sustain/sustain.shtml)

[3] For example, see costs of storage in Section 6.2and PrestoPRIME D2.1.1

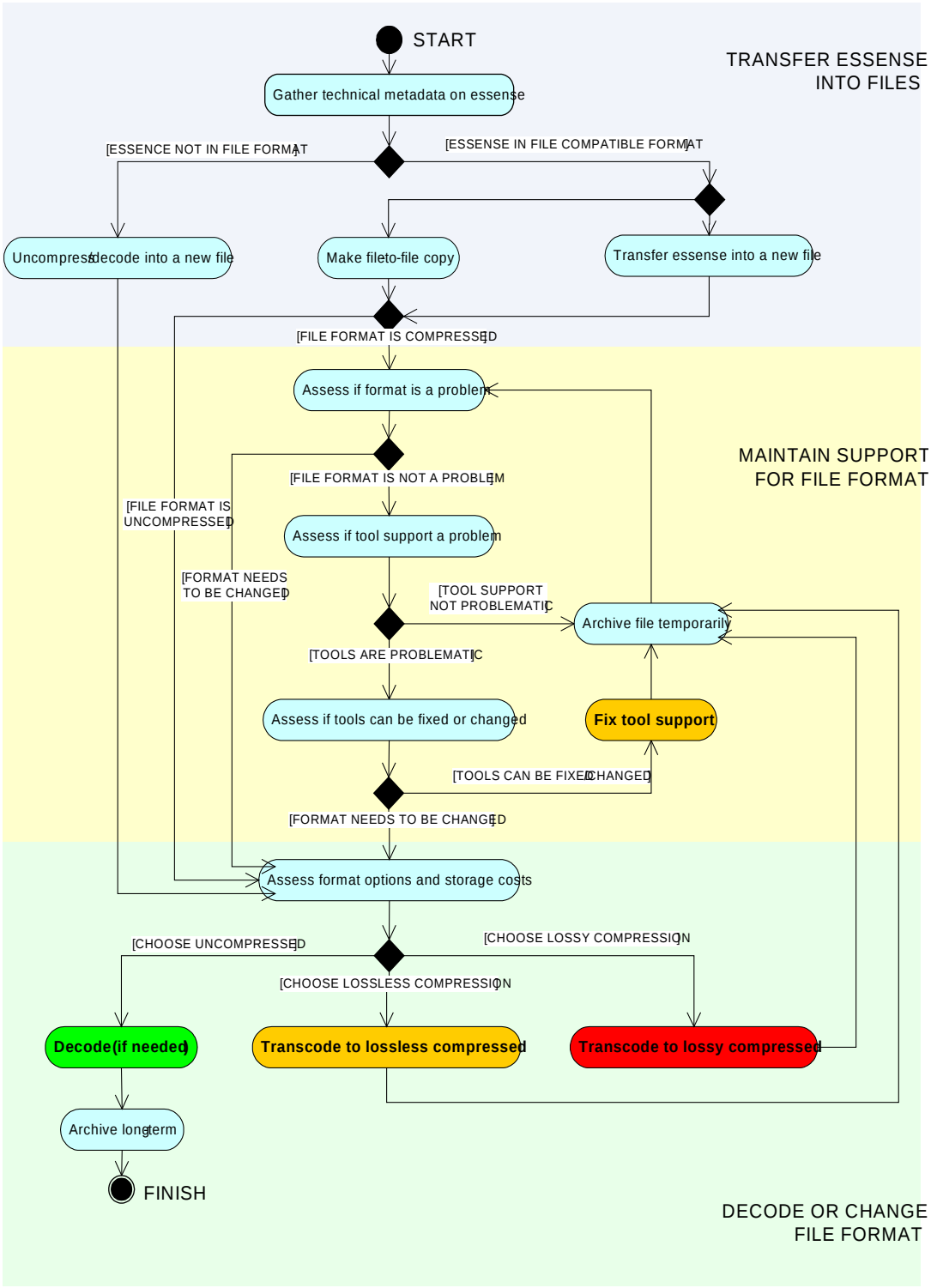

Figure 2 File format migration model

2.1.2 Storage Modelling

There are many approaches to long term preservation of digital audiovisual content. Each one has associated costs and risks as well as delivering differing degrees of content accessibility. No single technique provides a complete solution. Many archives face the challenge of how to compare, assess and combine the options in a consistent way.

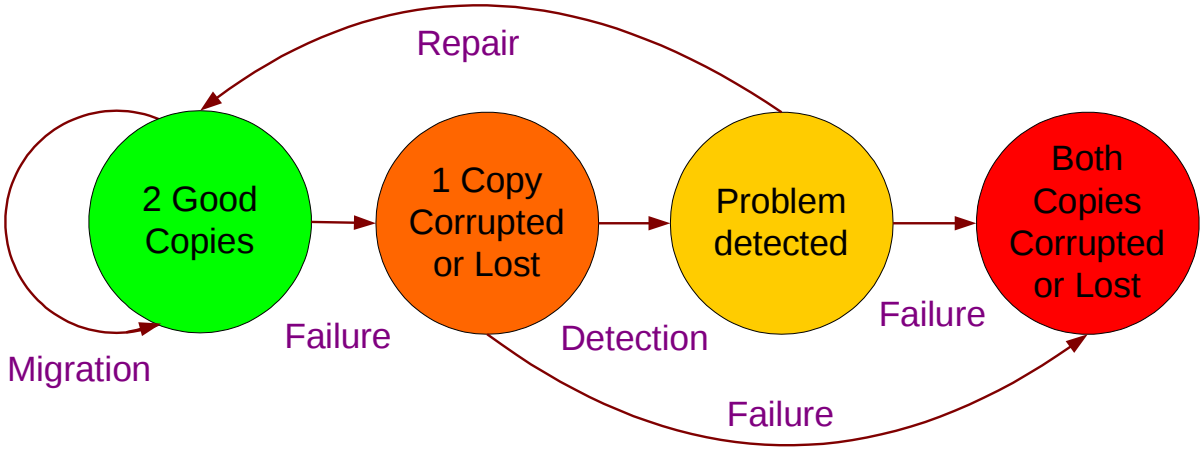

Figure 3 Model of preserving data integrity

Figure 3 presents a simple model for analysing preservation strategies for data safety which would be used as part of storage modelling (see Figure 1).

With reference to Figure 3, the bedrock of data safety is to keep multiple copies of content (green circle), e.g. by using different technologies and in different locations, and ideally operated by different people. This guards against major risks, e.g. by enabling disaster recovery, but also guarding unanticipated problems with individual technologies and processes – i.e. it ensures eggs are not 'all in one basket' at any level.

For each copy, there is the need to regularly migrate each component of the technology stack (hardware, operating system, management software, formats etc.). However, there is always the chance that one of the copies is damaged or lost due to some form of failure in the system (orange circle). But only after this problem is detected (yellow circle) can any action can be taken, e.g. to repair or replace the damaged or lost copy by using the remaining good copy. If at any time something happens to the second copy (the only remaining good copy), then there is a risk that both copies are permanently lost or damaged (red) – i.e. content is lost.

This is of course a simple model and does not consider the case that a corrupted copy can be repaired without needing to resort to accessing the other copy, for example by concealing errors rather than repairing them. Likewise if both copies are damaged, there may be cases where it is possible to use fragments of each to reconstruct a new good copy. This would be a transition from the red circle back to the green circle. It is possible to add these new transitions to the diagram if needed as a refinement. These new transitions would also have new costs associated with them, e.g. the use of an operator or tool to do repair instead of a simple file copy of known good file to replace a known corrupted file.

The rate at which transitions happen between the states dictates how long content is at risk of this loss. Every transition has a cost and hence considering the model as a whole allows the total cost and total risk to be assessed and individual approaches compared as shown in Table 1.

This framework approach for evaluating options therefore provides value as a structured way to consider both costs and risk of loss..

Approach	Example	Migration	Failure	Detection	Repair	Access to content	Notes
Very long lived media	Printing digital bits onto polyester film stock	Infrequent if at all, e.g. film lifetime >200 years	Depends on storage conditions, but very unlikely if good practice followed, e.g. if a film is stored in deep freeze.	Inspection or spot tests. Hard to automate, i.e. high labour cost	Reprinting in whole or in part. Very expensive	Relatively difficult. Expensive. Latency is measured in days or more. Needs a film scanner.	Possibly the only option if there is a risk that 'active' preservation can't be sustained. History suggests film has at least some chance of surviving substantial neglect.
Reliable media	E.g. data tape can be very reliable if used in specific archiving contexts (write once, read occasionally). Data tape for backup can be a different matter!	Frequent, e.g. every 6 years or less for LTO tape due to limited backwards compatibility of new drives with old media	Very low bit error rates. Failure rates, typ. 0.1-1% of tapes. Problems are often in drives not tapes.	Only need to check integrity on access or during migration	Replace damaged tapes or drives. Drives are expensive and have limited life.	Latency can be high, e.g. tapes on shelves, but data rates good. Need multiple drives for concurrent access.	Other types of reliable media, e.g. magneto optical disks bring other risks, e.g. lock-in to vendors who can go bust.
Many copies	2 online copies on HDD and 2 backup copies on data tape	Frequent, but depends on technology used for copies	The number of individual failures will go up as number of copies goes up	Reduced need to check copies due to increased redundancy	Can repair less often, e.g. only after certain number of copies are lost	More copies can mean easier access, incl. sharing of load for multiple users	Number of copies typically limited by prohibitive costs for video or film
Resilient AV encoding	Adapted Dirac or JPEG2000 encoding, uncompressed	Format migration for uncompressed is infrequent, e.g. 30 years. Shorter for compressed formats e.g. dirac or JPEG2000	Some data corruption can occur without loss of usability of content, e.g. impact is not visually significant or is correctable.	Need to detect less often due to increased resiliency to corruption.	Repair built in, or 'graceful degradation' means quality is still acceptable and repair not necessary.	Depends on availability of decoders, but not a problem for established formats e.g. JPEG or uncompressed	Virtually all compressed image, audio and video encodings act as huge 'amplifiers' to data corruption.

Resilient data encoding	Almost all storage uses some form of data redundancy and error correction strategy, e.g. HDD, data CD, tape.	Depends on technology, e.g. 3 years for a HDD, 6 years for data tape, maybe 10 years for archival grade CD	All digital storage has some form of errors, be it the media or the servers/systems it is stored within. Failures can be bits, bytes, blocks, media or systems (e.g. RAID array)	Built into the device (e.g. HDD), player (e.g. LTO drive), system (e.g. RAID controller) or high levels (e.g. ZFS filesystyem).	Range of techniques, e.g. use of CRC, parity, block or file level replication and repair	Depends on type or system e.g. HDD on shelves or in an online server.	Built in protection mechanisms have limits and the complexity (software, hardware, firmware) means techniques never perfect. Residual errors will exist.
Concealment	Digital Video Tape, Audio CD, Video DVD	Obsolescence times can be relatively long, e.g. 10-30 years	Failures, e.g. read errors, are detected and repaired or concealed automatically by the player, e.g. DV deck, CD player.			Hard to automate, e.g. jukeboxes for discs or video tapes are expensive. AV equipment needed to access content (as opposed to IT equipment for file-based storage)	Equivalent to concealment in the IT world is digital restoration tools.
Check often, fix quickly	Hard drive storage	Frequent, e.g. every 5 years or less.	Relatively frequent, can be silent and unrecoverable	Proactive checking of file integrity, e.g. using checksums	Replace damaged copies. Can need large data transfers, e.g. TB files to fix only a few bits of corruption	Low latency, high bandwidth. Random access to parts of files, e.g. ‘partial restore’. Easy to support many users.	Latent errors can occur at all levels of the storage stack, including in parts designed to protect data, e.g. RAID

Table 1 Comparison of data storage strategies

3 Overview of the tools

Our work on tools has initially concentrated on storage modelling for two reasons: it is required in order to do many aspects of wider preservation modelling and hence needs to be implemented first, but it can also be used on its own right e.g. when planning or managing existing storage infrastructure.

Tools for higher level preservation modelling, e.g. file format migration, digitisation, cataloguing are currently under development.

Two 'cost of risk of loss of storage' tools have been created:

- **A long-term planning tool**, which designed to support decision making on what storage strategy to use, for example how many copies to make of files in archive, what storage technologies to use to hold them, and what measures to take to maximise the long-term integrity of these files.
- **An interactive simulation tool,** which is designed to support more operational levels of decision making, e.g. how to allocate resources to tasks such as ingest, access and maintenance, and how to react to unforeseen events, e.g. failures in storage systems, peaks in load.

The long-term planning tool is Web based and takes a set of parameters as input (e.g. number of files to be stored, types of storage system to use including their costs and reliability). The user can choose how they want to store their files (e.g. make 2 copies both on data tape, or make two copies both on HDD). The tool then provides a graphical projection of cost, risk and loss over time, e.g. for a 25 year period. The user is then able to compare the different storage options (e.g. HDD v.s. data tape or a combination of the two) and investigate the costs and benefits of active integrity management (e.g. 'scrubbing'). The result is the user having more confidence in which option makes most sense to them. They can then use this knowledge to start investigating a particular solution in more detail, e.g. more accurate costing, or to have an informed conversation with their IT department or with storage vendors.

The interactive simulation tool starts with a particular storage configuration (e.g. 2 copies, one on data tape, one on HDD) and then simulates events that might happen when using this system in practice. As the simulation progresses time ticks away (e.g. 1 second of the simulation might correspond to 1 week in the real world). Events happen during the simulation that include ingest and access to files, data corruption, system failures, and data management activities such as copying files between systems. The simulation includes the ability to model resources that are limited (which could be tape drives in a server or human operators) and how this limited resource impacts on the ability of the system to cope with events, e.g. whether queues build up or files are put at increased risk. The user can interact with the simulation as it progresses, e.g. changing the amount of resources available or changing the policy for data safety (e.g. making more copies or checking them more often). In this way, the user is in effect playing a game that helps them understand how to react to and manage events they might see in practice when operating a real system.

4 Walkthrough of the long-term storage planning tool

The series of screenshots below show a typical usage scenario for the long-term planning tool.

4.1 Defining the 'archive' to be modelled

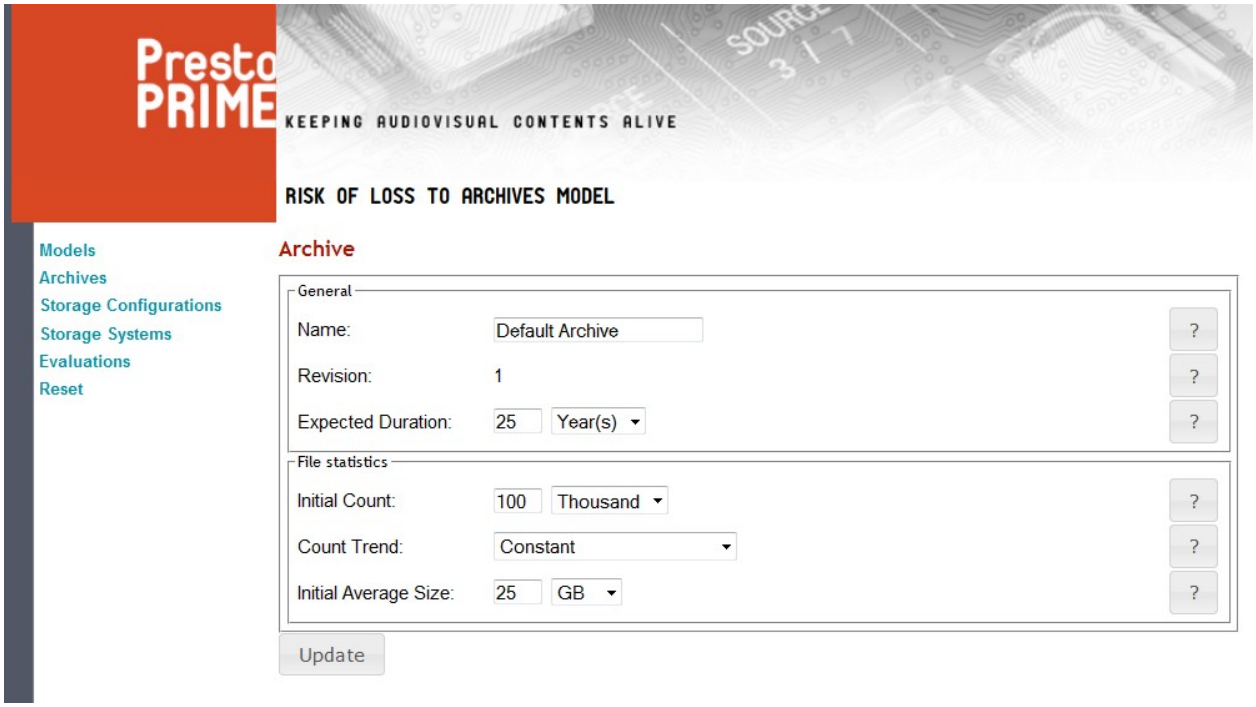

Figure 4. The user defines the 'archive' that they want to model in terms of number of files (100,000), file size (25 GB) and the duration of the cost/loss projection (25 years).

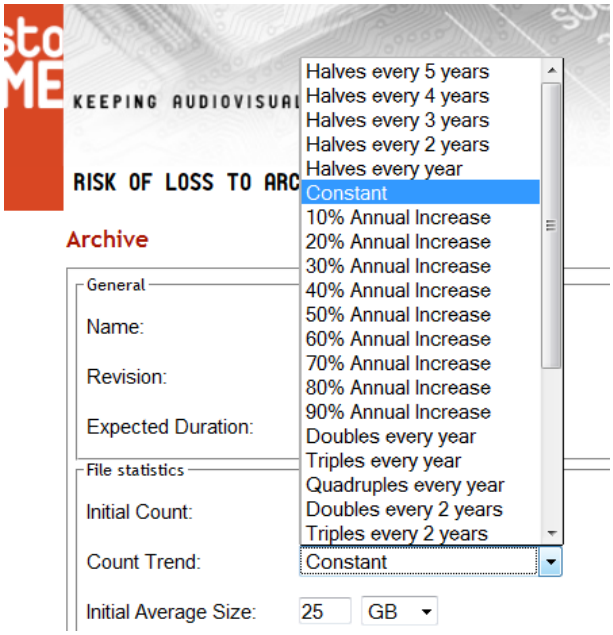

Figure 5 All parameters of the model are configurable by the user and the tool includes the ability to apply trends, e.g. the user might select that number of files increases by 10% each year.

4.2 Storage systems used for storing files in the archive

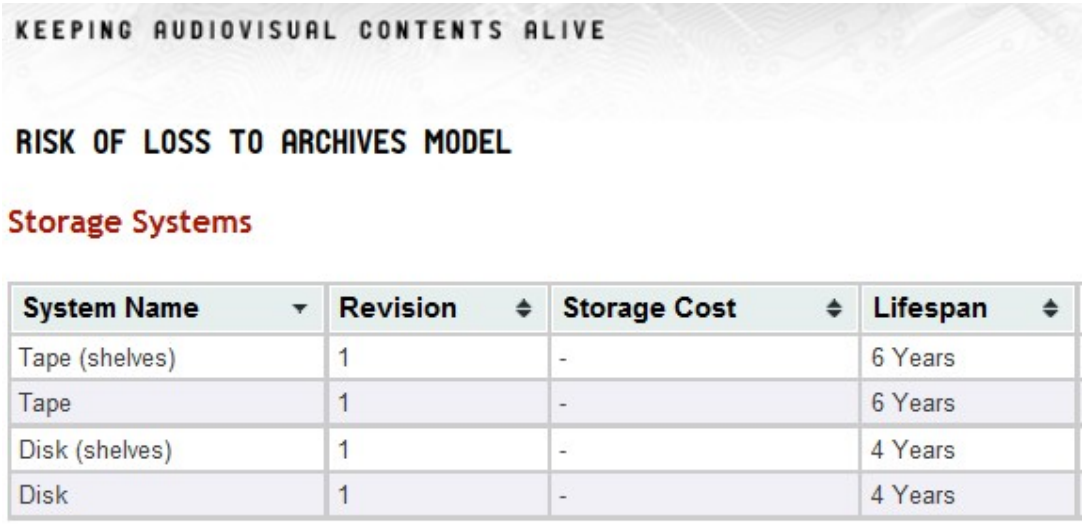

Figure 6 The user can select which storage approach they want to use, e.g. data tape or hard disk drives, or a combination of the two. The tool is pre-loaded with 4 types of storage system: data tapes stored on shelves, data tapes stored in a robot, hard disks on shelves, and hard disks in a server.

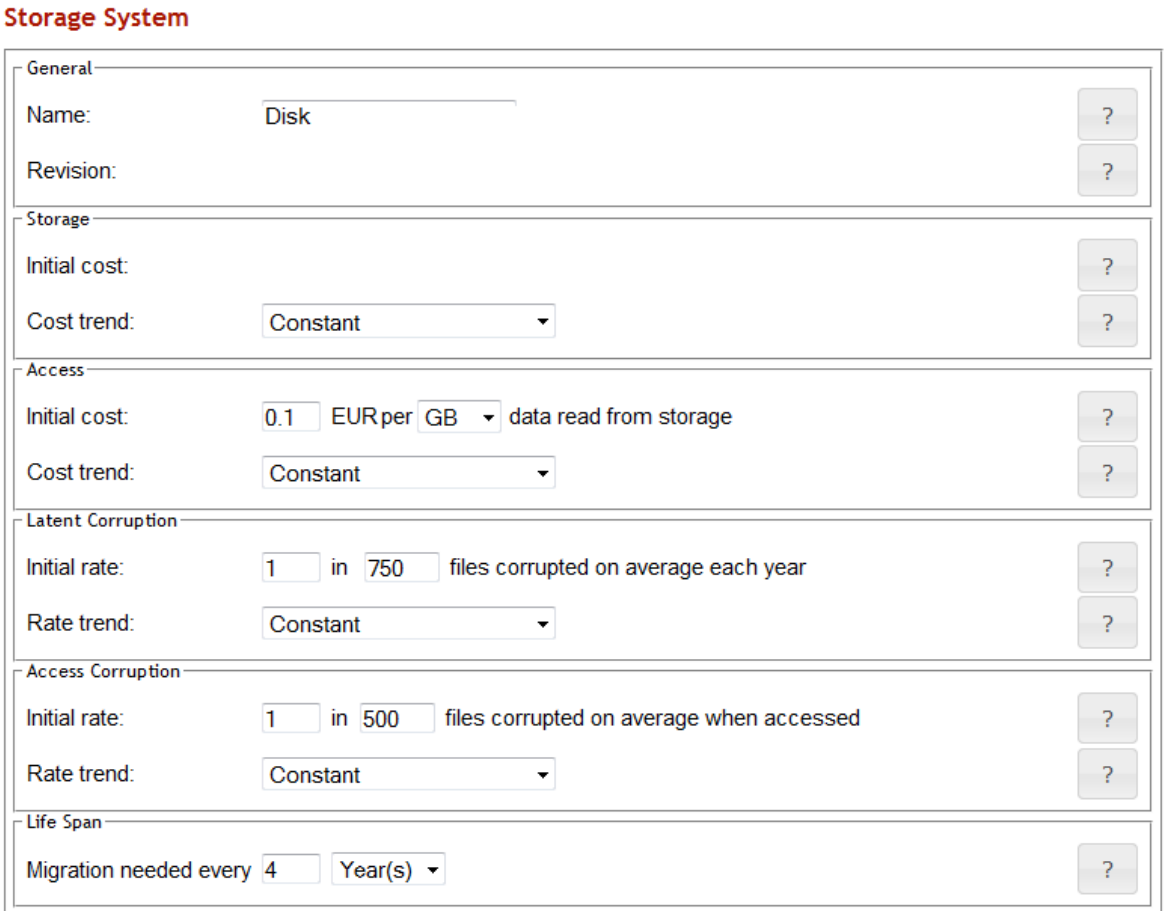

Figure 7 Each storage system has a set of parameters that describes its costs and behaviour. The parameters for storing files on a hard disk server are shown in this example. All parameters can be set by the user with their own values. The numbers shown above are defaults. The parameters describe (a) the costs of using that storage system for both storing and accessing files, (b) the chances of files being lost/corrupted by the storage system when 'at rest' or when 'being accessed', and (c) the lifetime of the storage system after which all files will need to be migrated to a new system. Trends can be set for both costs and corruption rates. A full explanation of these parameters is provided in Section 6

4.3 Storage configuration

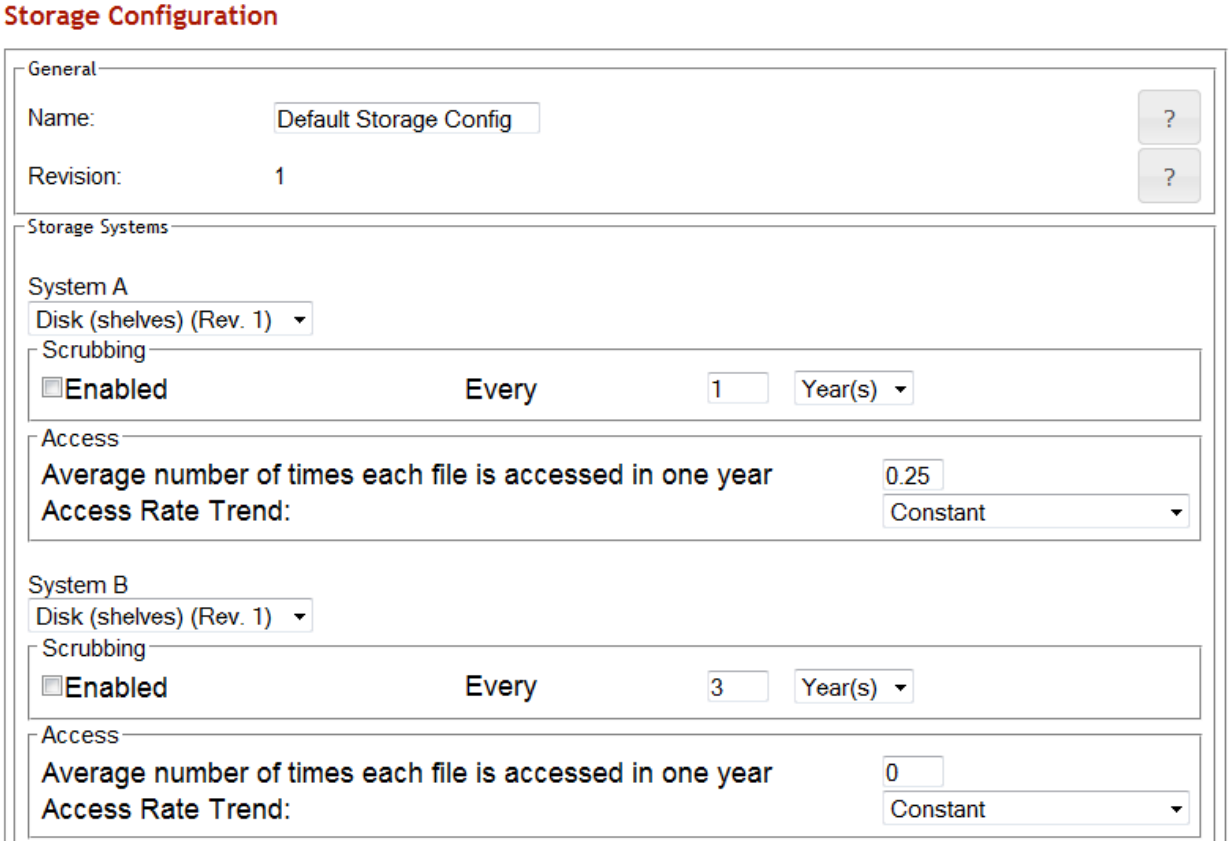

Figure 8 The user can decide which combination of storage systems to use for their archive. The tool supports a 2 copy model, e.g. one copy on hard disk and one copy on data tape. The example above corresponds to both copies of a file being stored on hard disks that are kept on shelves (not a sensible preservation strategy, but good for illustrating the tool – see later screenshots of the resulting cost/loss projection). The user can set which of the storage systems are used to serve access requests to their archive and how often these requests take place (e.g. in the example above, 25% of files are accessed each year and the first storage system is used to support these access requests). The user can also decide whether each of the storage systems is 'scrubbed'. Scrubbing means periodically checking all the files for their integrity (e.g. using checksums and then fixing any files that have problems).

To summarise at this stage, the cost/loss projection has been defined as:

- 100,000 files of 25GB each stored for 25 years
- 2 copies are made of each file. Each copy is stored on a hard drive kept on a shelf.
- On average, 25% of files will be accessed each year, with access being satisfied using the first set of hard drives (the other being safety copies).
- The cost of storage is low (media on shelves), but the access costs are high (because a person has to retrieve the drive and load up the files)
- The probability of loss is high (reflecting typical annual failure rates of bare drives and accidental damage by operators when the drive when handling it for access).
- No trends are used for simplicity, but would be used in more realistic examples

4.4 Model results

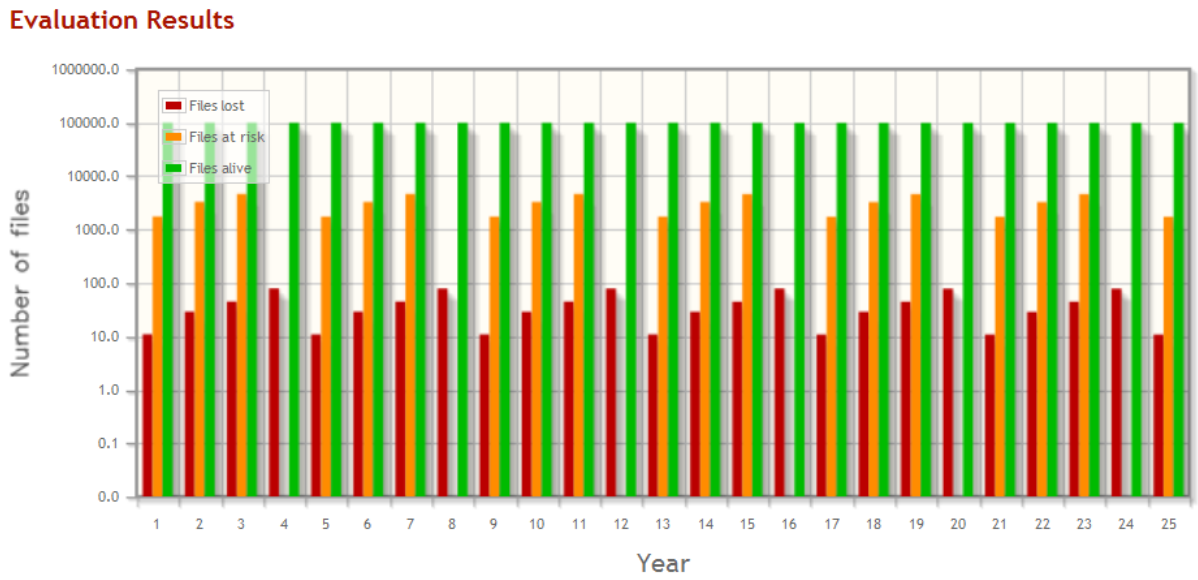

Figure 9 The user is first presented with an overview of number of files that are preserved/lost over time. A log scale is used due to highlight the number of files that are lost against the much larger number of files that will typically be successfully preserved. The green bar is the number of files that are preserved (one or both of the copies is intact). The orange bar is the number of files that are at risk in a given year (one of the copies is lost, but one of the copies is still OK – which means that if this one remaining copy is lost then the whole file will be lost). The red bar is the number of files that are lost in each year (both copies lost). Out of the 100,000 files being stored, between 10 and 100 are lost each year, with several thousand constantly at risk. The rate of loss increases year on year (because more hard drives fail) until a migration takes place (every 4 years), at which point all drives and files are checked and problems fixed where possible.

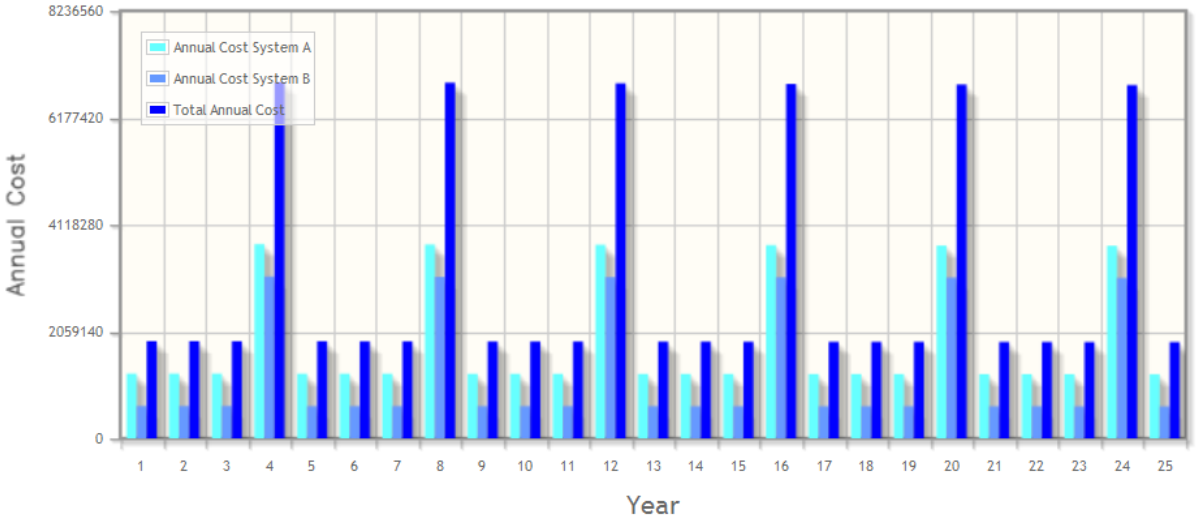

Figure 10 The user can see the cost of storage year on year. The two lighter blue bars represent the annual costs of the two storage systems (note that the first set of hard drives is more expensive than the second since this set is used for serving access requests). The dark blue bar is the total cost of both storage systems combined. The big spikes in cost every 4 years correspond to migration since every file has to be retrieved and copied to a new hard drive, which is a major access cost.

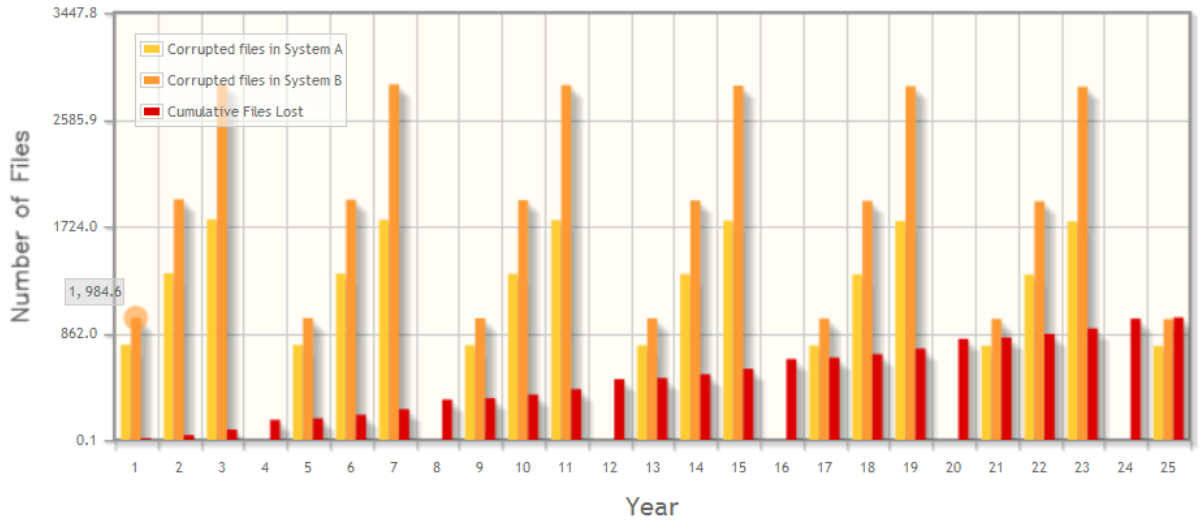

Figure 11 The final graph the user is presented with shows the cumulative number of files lost (dark red bar) and which of the two storage systems is putting the files at most risk (orange bars). In this example, over 1000 files have been lost in the 25 year period. The first of the two sets of hard drives poses less of a risk than the other set since the first set is used for access requests, which affords an opportunity to check the drives and files and pick-up any failures early.

4.5 Exploring options

Having done a projection of cost and loss for files stored on hard drives on shelves, the user is able to revise their choices to see whether a better solution can be achieved, or simply to compare different storage options. Some examples are shown below.

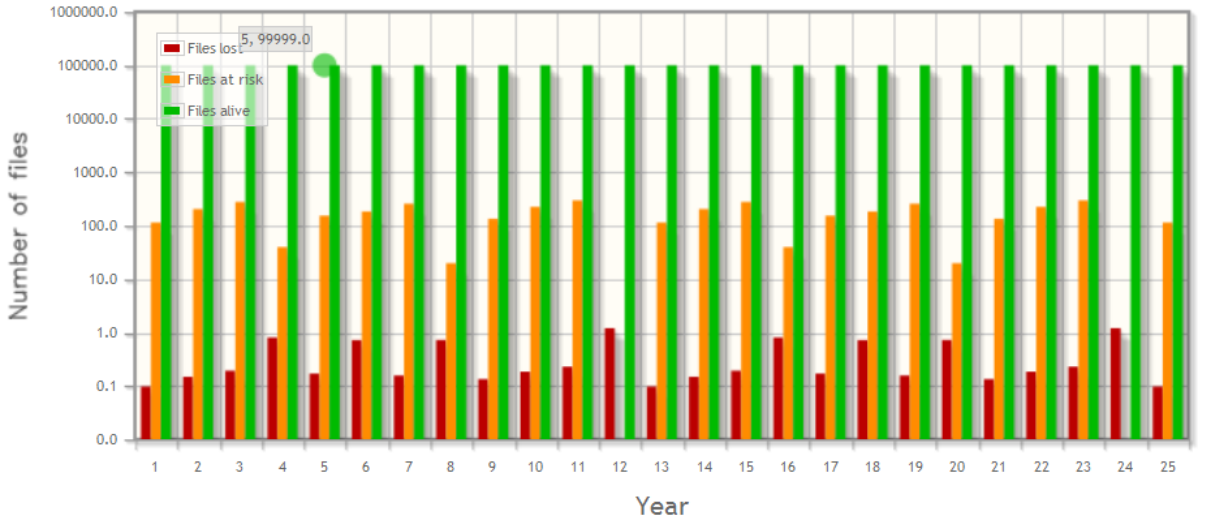

Figure 12 This shows the rate of file loss if the 'hard disks on shelves' approach is replaced with copy1 of the files being held in a disk server (e.g. a RAID array). This makes more sense from both a safety and access perspective. Copy2 has also been changed so it is now held on data tapes put on shelves (deep archive safety copy) instead of on HDD on shelves. It can be seen that the rate of file lost has dropped significantly, with less than one file lost a year on average.

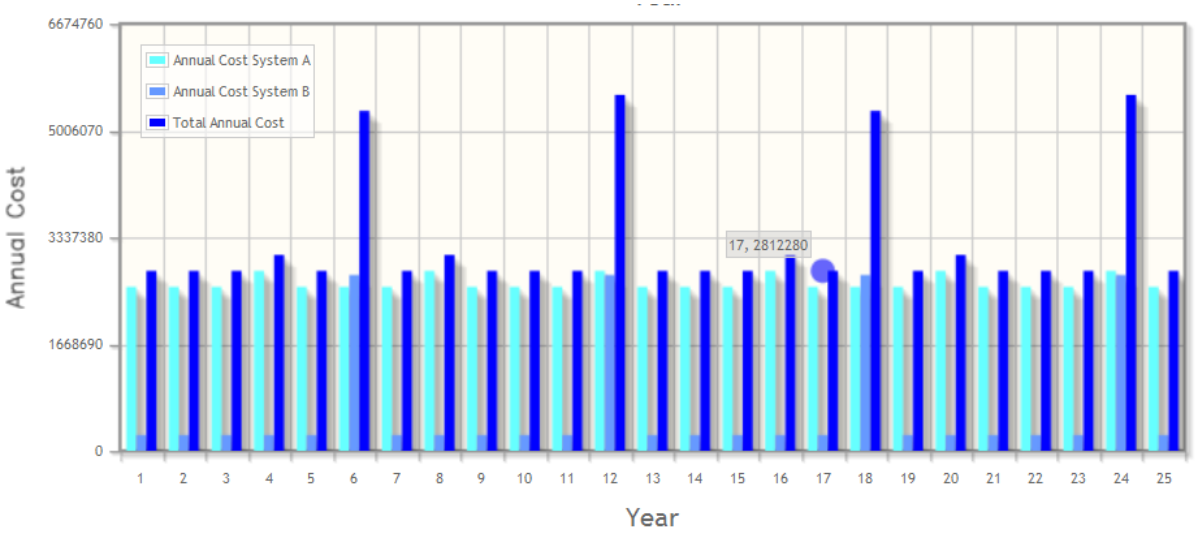

Figure 13 Whilst the rate of loss has been reduced, the cost of storage has increased, especially from the use of a hard drive server compared to hard drives on shelves.

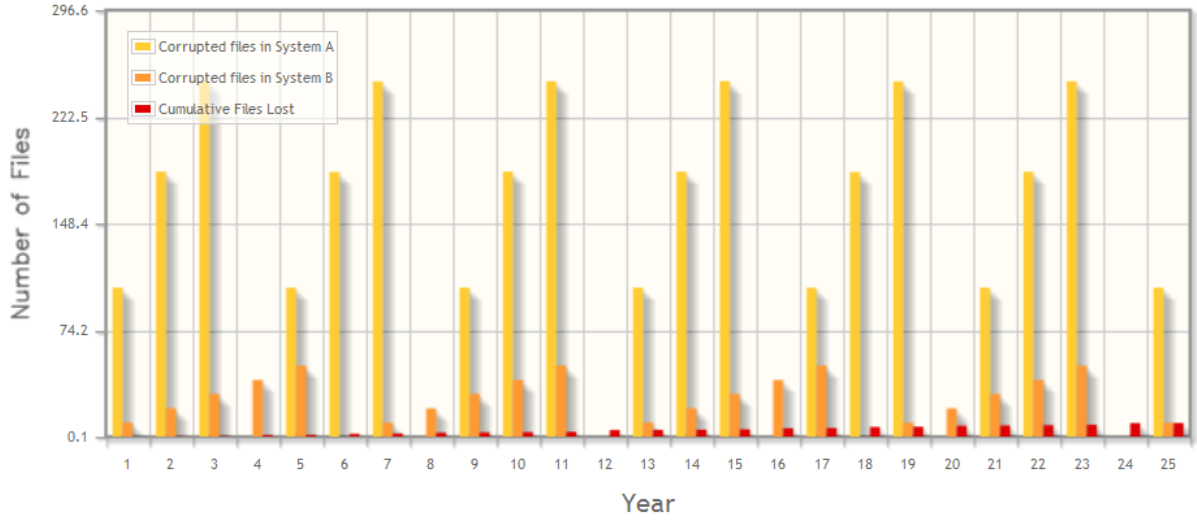

Figure 14 The main contributor to risk of loss is the hard disk server rather than the data tapes on shelves, which indicates an area in which further improvements could be made.

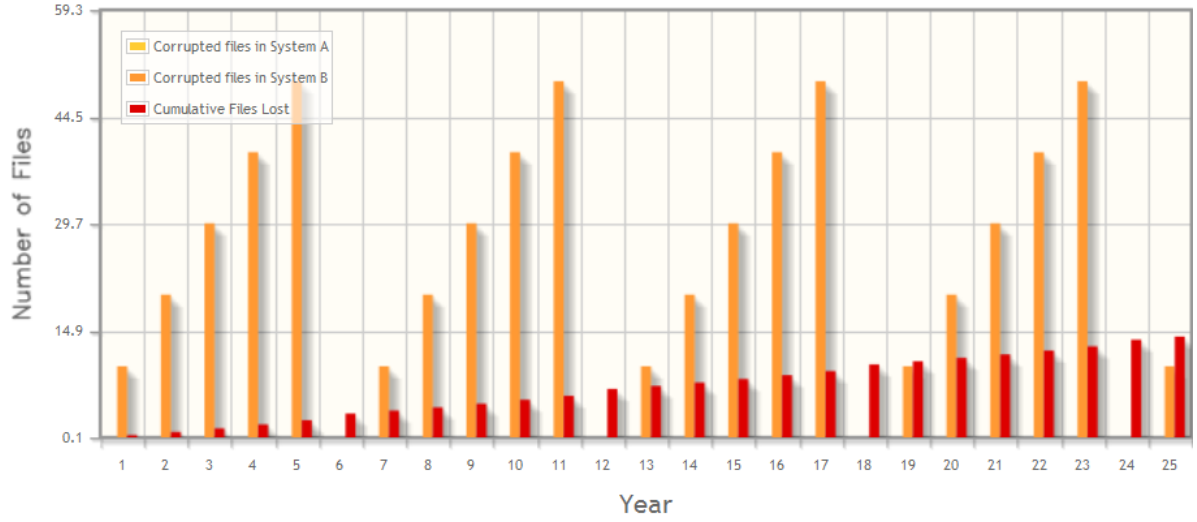

Figure 15 above shows the effect of turning scrubbing on for the disk storage system (all files copies checked each year) – the risk of loss from the disk system has almost been eliminated. Not all file loss has been prevented due to the risk of file loss from manual handling of the tape copy (e.g. when it is used to repair a failed copy in the disk system). Note the difference in scales between this graph and Figure 14.

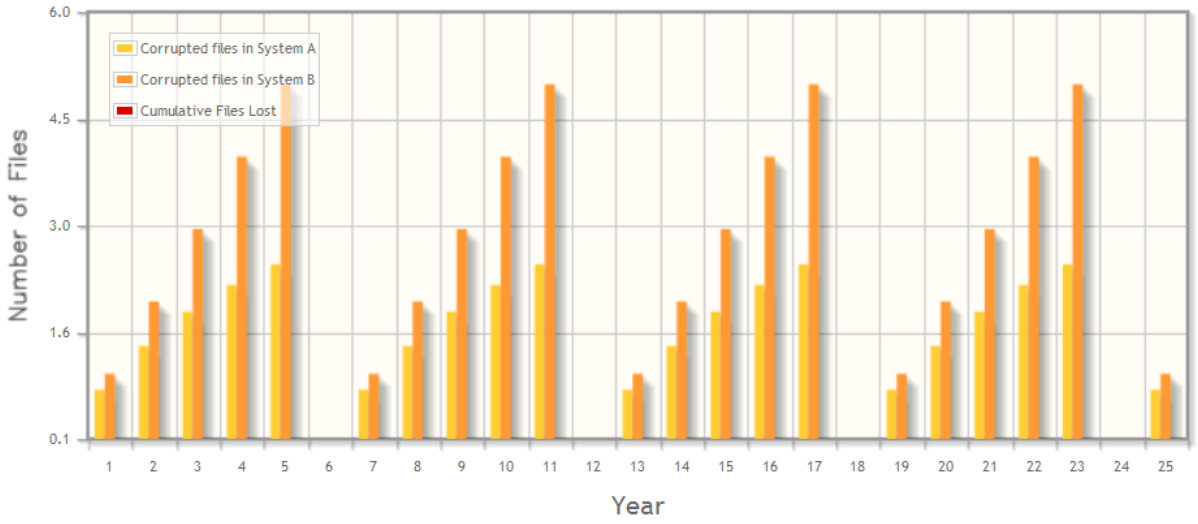

Figure 16 If the HDD server and tapes on shelves approach is replaces with an all tape solution where the tapes are held in robots then file loss can be further reduced as shown in this example. The chances are that no files will be lost at all after the 25 year period.

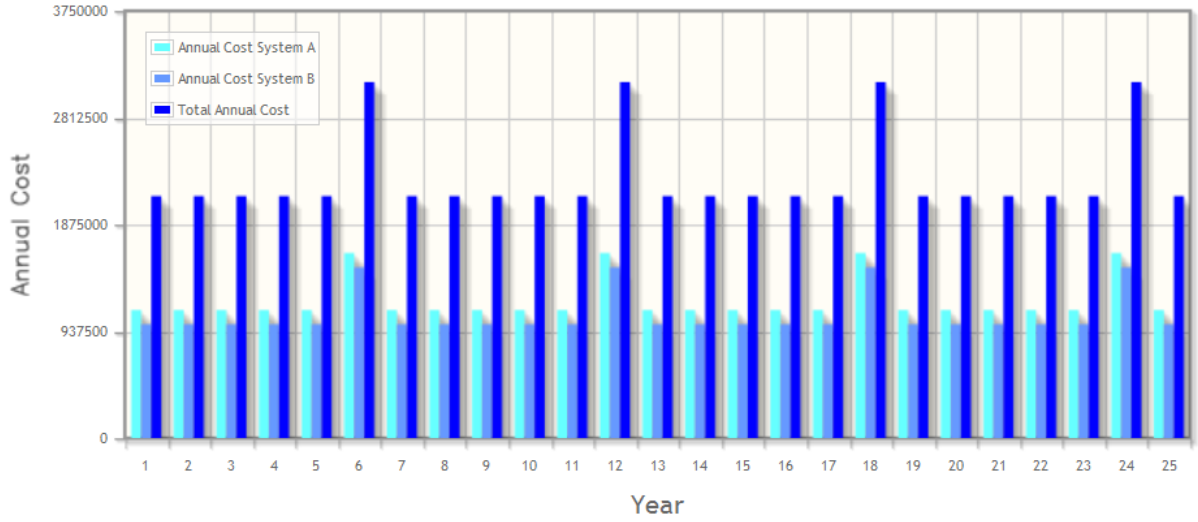

Figure 17 The cost of the two copies on data tape in robots strategy is shown above. This is not substantially different to the hard drives on shelves model yet the level of data safety is dramatically better. This is because the hard drive on shelves approach needs people to retrieve content, perform migrations and to replace failed disks, all of which has a significant cost in the long-term. Data tapes in robots however are much more reliable and access and preservation processes, e.g. migration can be automated which keeps costs down.

Whilst one use of the tool is to compare different storage strategies, another is the ability to do sensitivity analysis to variation in different values of input parameters, e.g. best case or worst case for a given storage system. One approach to this would be to provide error bars on the data points in the graphs. For example, rather than providing a single value for the number of files lost in a given year, a range might be displayed within which there is a 90% chance that the number of files actually lost would fall. The probability distributions for error rates in storage systems are not necessarily simple (e.g. due to correlations as described in Section 7). The tool currently assumes a Poisson distribution and converts the error rates provided by the user into a probability that one or more corruptions occurs to each file.

One simple option for sensitivity analysis is to for the user of the tool to run the model several times using inputs that are at the extremes of what they think might occur in practice. The results of these runs of the model can then be compared, e.g. the difference in file loss if a Bit Error Rate was 10^{-14} instead of 10^{-15} for a SATA HDD, or the difference if 1 in 500 data tapes were damaged by an operator when retrieved from a shelf instead of 1 in 5000. This comparison of extremes allows the user to look at how sensitive the model results are to variation in input values without the need to build probability distributions into the tool and the added complexity this brings. This approach is about finding the right balance between simplicity, flexibility and accuracy of the tool and is an area where user feedback will help inform development of future versions.

4.6 Summary

This section has shown how the long-term planning tool can be used to explore the different options for storage and how these impact cost and loss.

The cost model is not intended to be accurate (and this is impossible anyway over 25 years). For example, the model does not include sophisticated models of storage systems, detailed descriptions of the errors that occur in these systems including their distribution functions, or models of archive access that go beyond random access to files. All these things could be added but this would make the model harder to use – these are also things that are better supported by the interactive simulation tool described in the next section/

The approach taken, and the purpose of the tool, is to include just enough detail in order to make top-level choices, e.g. between a storage strategy of disks in servers or data tapes in a robot. The detailed operation and optimisation of the chosen approach can then be analysed using the interactive simulation tool described in the next section.

5 Walkthrough of the interactive storage simulation tool

The interactive simulation tool extends the long-term planning tool by adding several extra features:

- Parameters can be changed interactively by the user during the simulation.
- The simulation includes limited resources, e.g. queues for ingest, access, file maintenance.
- The simulation allows for the occurrence of catastrophic events, for example a whole storage system being lost due to fire, theft, major crash etc.
- The simulation supports 2, 3 or 4 copy models using multiple storage systems.
- The user can choose for the tool to automatically make simple optimisations in order to make most efficient use of resources, e.g. to minimise ingest queue lengths.

5.1 Setting up the simulation

The user sets up the simulation by defining the rate at which files come into the archive, how those files get allocated to storage systems, how many storage systems are used, and how files are replicated between them, and how the storage systems behave, e.g. corruption rates.

The interactive tool consists of three main components: (1) storage systems together with archived files, (2) ingest queue, and (3) user access queue.

In the simulation, individual file copies are stored within storage systems. The user can choose to have 1,2 or 3 storage systems and to make one or more copies of a file on each of these storage systems. As with the long-term planning tool, a storage system might be a hard disk server, a tape robot, or media on shelves operated by a person.

Storage systems follow a set of storage management policies defined by the archive system administrator. Instructed by these policies, each storage system may be configured to perform number of data preservation operations involving:

- Periodic scrubbing of stored files (integrity checks, e.g. using checksums)
- File repair using an alternative file copy (e.g. from another storage system)
- File ingestion (if the storage system is selected to be one to ingest files from the ingestion queue)
- File access (if the storage system is selected to provide access to archived files).

In the simulation, all operations performed by storage system such as file copy, ingestion, scrubbing or access, are all assumed to consume storage system resources. It is assumed that each storage system has only a limited amount of resources for each above listed operation. The fewer resources that are allocated to a specific operation then the longer its execution will take, i.e. the throughput is proportional to resource allocation. For example, during periodic scrubbing operation a large number of scrubbing requests (for

individual files) will be initiated and sent to scrubbing service that will perform the actual file auditing.

The speed at which file integrity check requests are processed by the scrubbing service depends on the number of available scrubbing resources. If the storage system has only one scrubbing resource available, the scrubbing service will only process single request at a time. Because provisioning of each scrubbing operation consumes certain amount of time, the remaining requests are put into the scrubbing service queue from which they are processed in the order of their arrival.

The speed of scrubbing operations can be increased by increasing the number of resources dedicated to scrubbing service. For example, if the number of resources is increased to two, the storage system will be able to process two scrubbing requests in parallel and thus at the same time. Consequently this doubles the speed of scrubbing operations.

The ability of the tool to allow the user to change the resource levels means that the user can explore a range of situations that might occur in the real world – for example, adding extra tape drives to a robot, adding more staff to a workflow, or maybe investigating the impact of staff losses or illness.

Similar resource constraints apply for the file copy, ingest and access operations allowing the user to observe how resource limits and thus time-dependent processing of vital system operations will all affect the overall system performance.

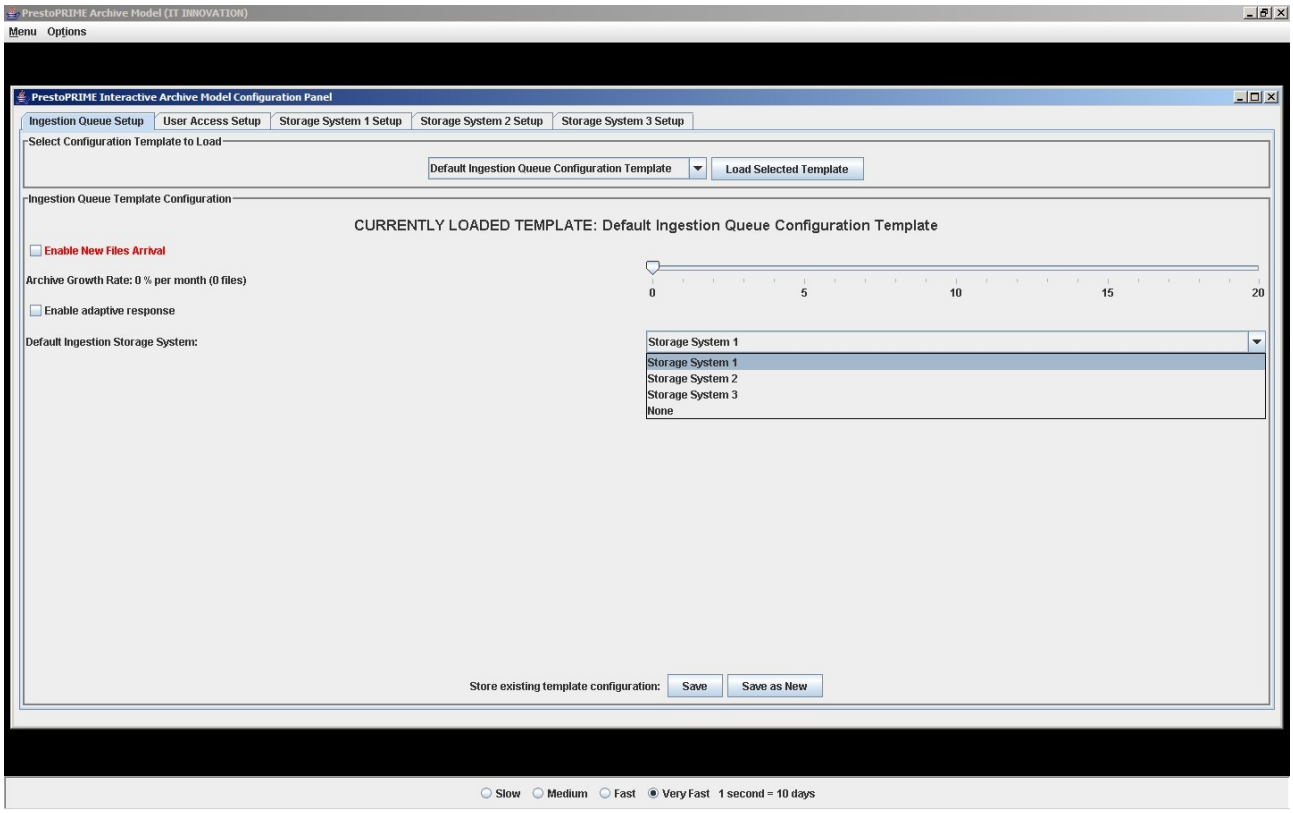

Figure 18 The user can set the rate at which files come into the archive and which storage system they are ingested into.

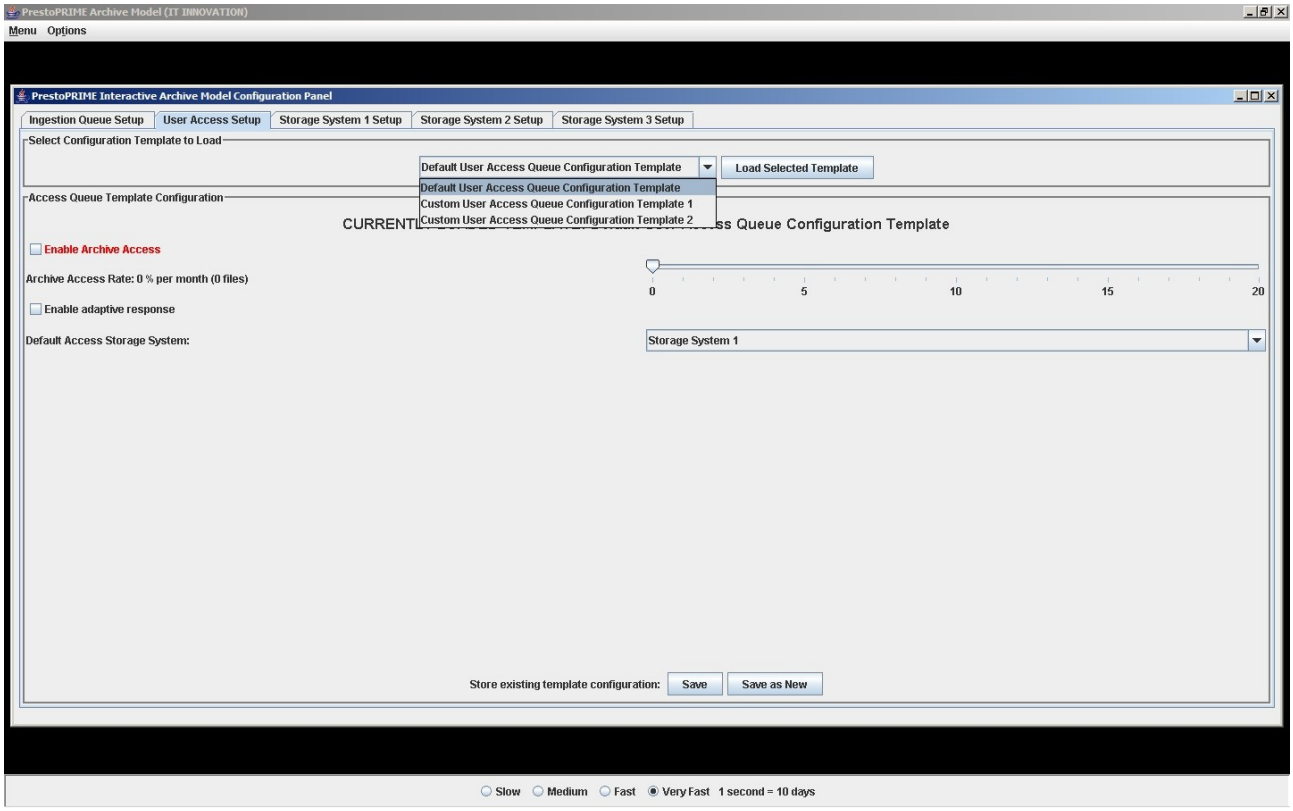

Figure 19 The user can set whether files in the archive are accessed, the rate at which they are accessed, and which storage system is used to serve access requests.

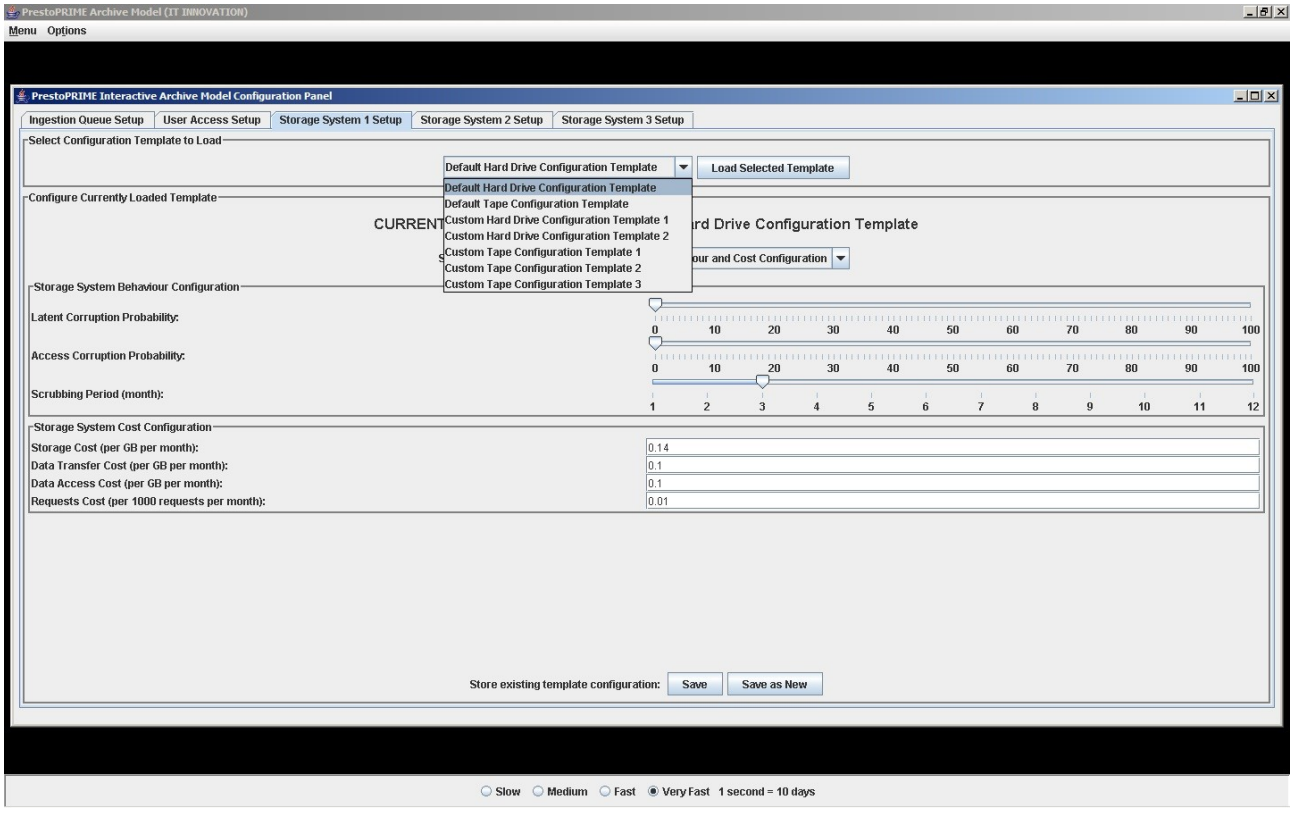

Figure 20 The user can select one of several different types of storage system, each of which has a set of default parameter values (templates), or the user can define their own type of storage system.

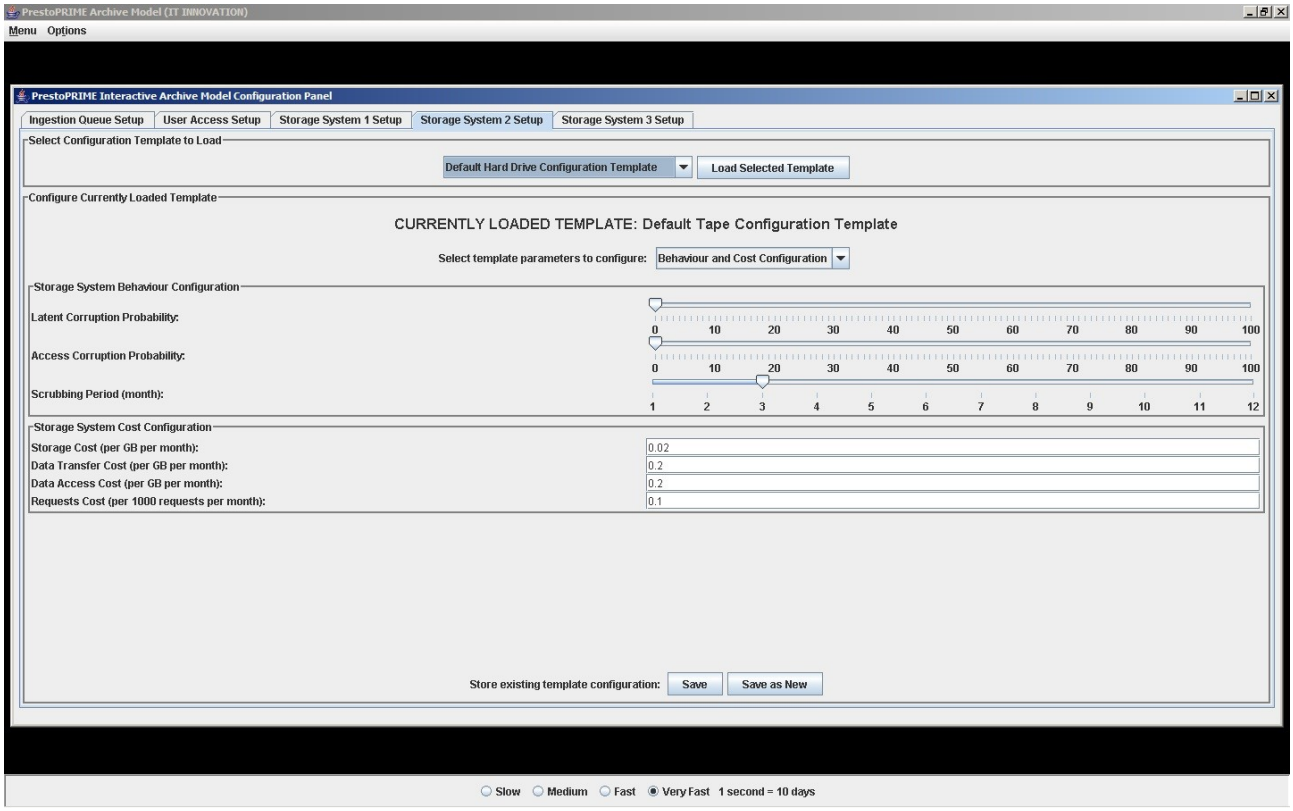

Figure 21 For each storage system, the user can set the probability of files being corrupted and the costs of storing of accessing files in the storage system. The user can set whether the files on the storage system should be scrubbed (integrity checked) and if so with what interval.

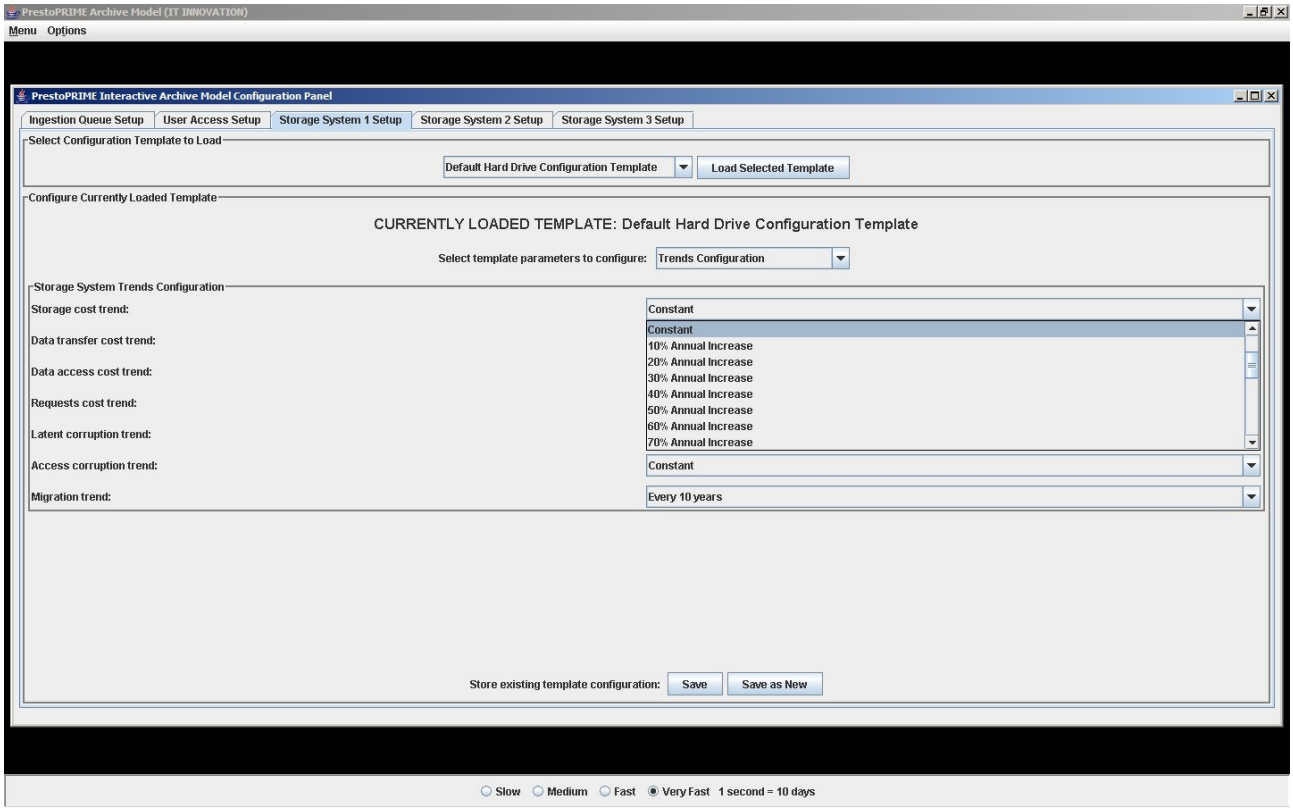

Figure 22 For each of the parameters for a storage system, the user can set trends for how the parameter values will change over time. The user can also at any time in the simulation adjust the values to a setting.

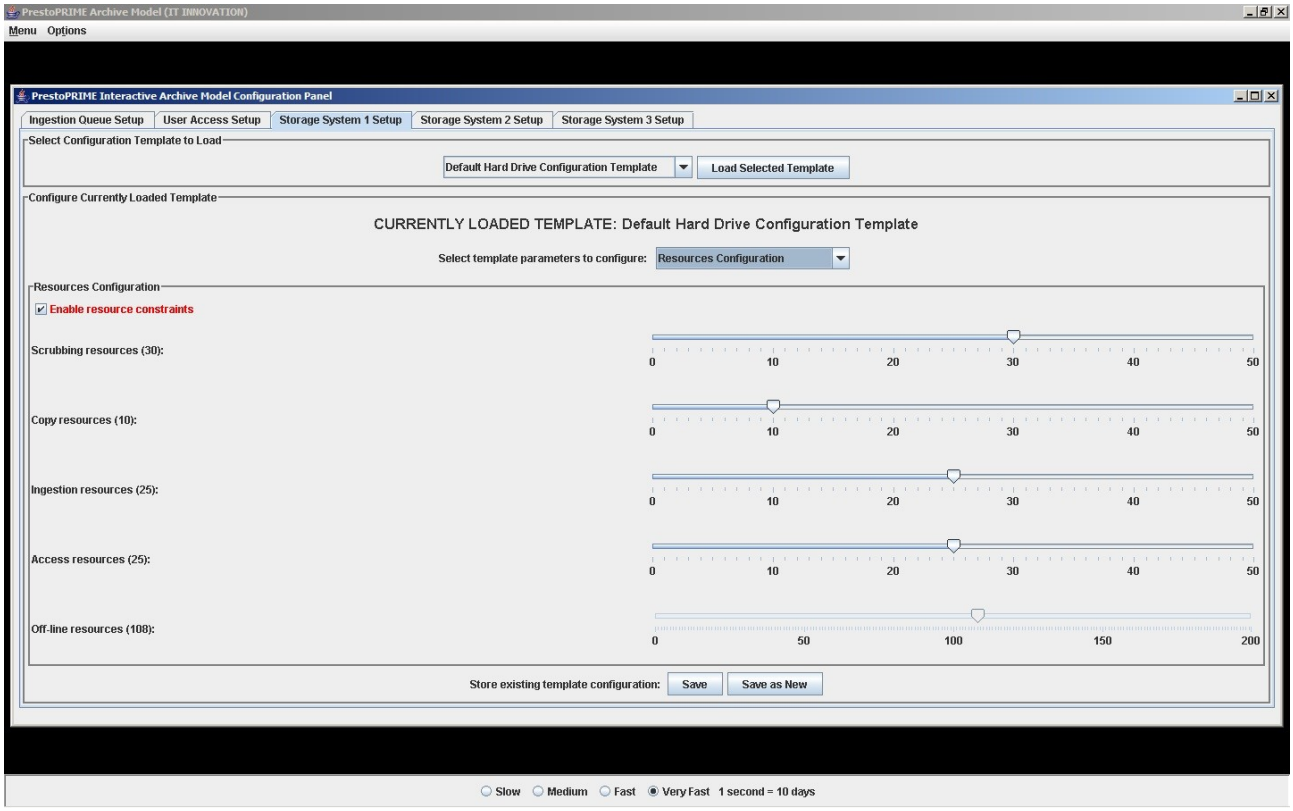

Figure 23 The user can set how much resource is available to perform different actions related to each storage system, e.g. copying files, ingesting new files, supporting access to files for archive users and so on. The amount of resource available for each function can be set independently. Depending on the storage system being simulated, resources might be for example tape drives in a robot (automated archive) or people handling tapes manually (tapes on shelves archive).

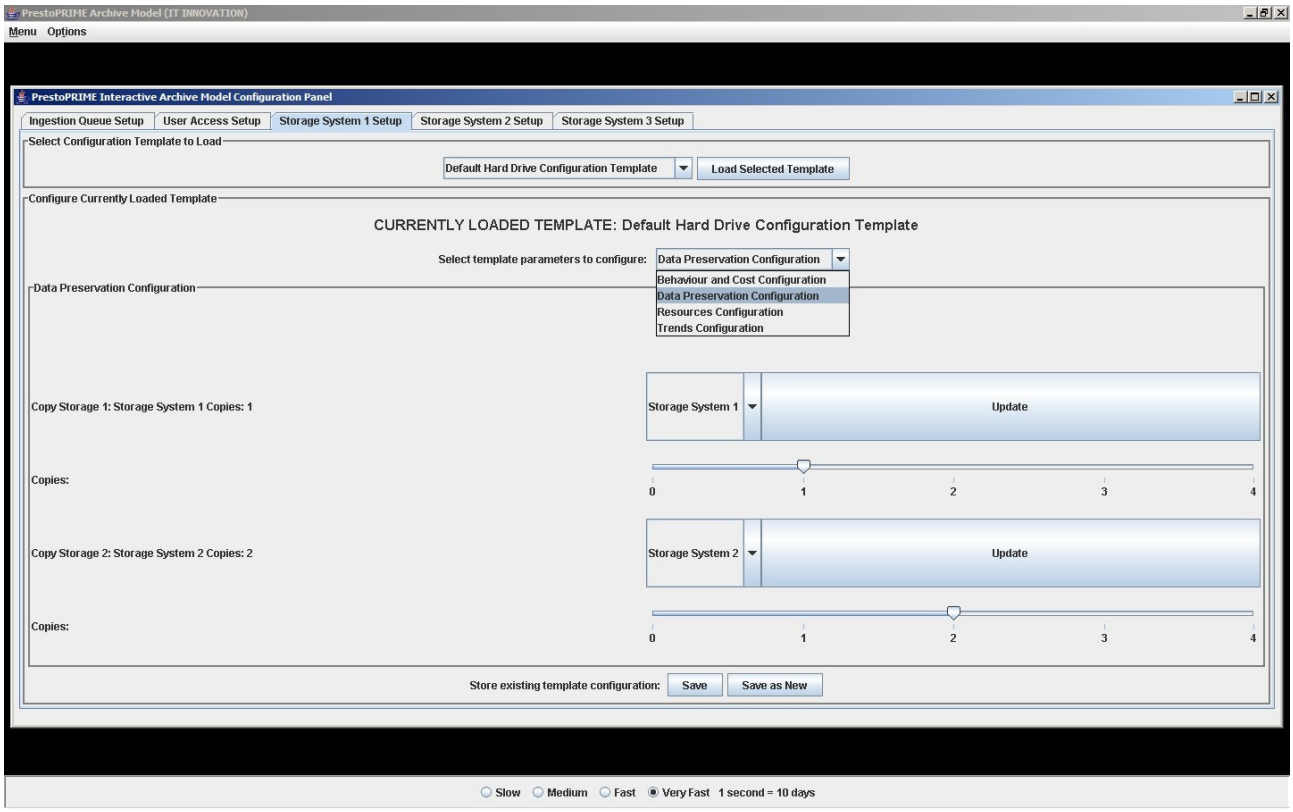

Figure 24 The user can define which storage systems are used to hold copies of the file and how many copies to make on each storage system.

5.2 Running a simulation

The simulation is interactive, meaning that it simulates a continuous operation of the archive system where user is able to adjust at run-time some its critical parameters and observe short and long-term consequences of these actions. The simulation takes into account a number of real archive system properties that play significant role in their operation and management. In particular, resource constraints are included in the model to model operations that take time and consume resources. By providing and adjusting these properties, the simulation offers the user the possibility to play 'preservation game' that takes place in realistic resource-limited conditions and allows to learn how various actions affect archive system operational efficiency.

Unlike the long-term planning tool, the interactive tool simulates actual corruption, ingest and access events. These happen at random according to the probabilities set by the user as input. This means that no two runs of the tool will be exactly the same. For example, for each tick of the clock, files will be chosen at random to be corrupted. It is not the case that a fixed number of files will be corrupted each year. The use of a random corruption model and corruption rate set by the user means that the tool is simulating a Poisson process[4] and corruption follows a Poisson distribution. The same applies to ingest and access events. Although the model uses a simple random model of ingest, access and corruption, the model could be extended if needed to include different distributions (e.g. to model a batch of media failing, or all items in a specific collection such as a TV series being accessed at the same time).

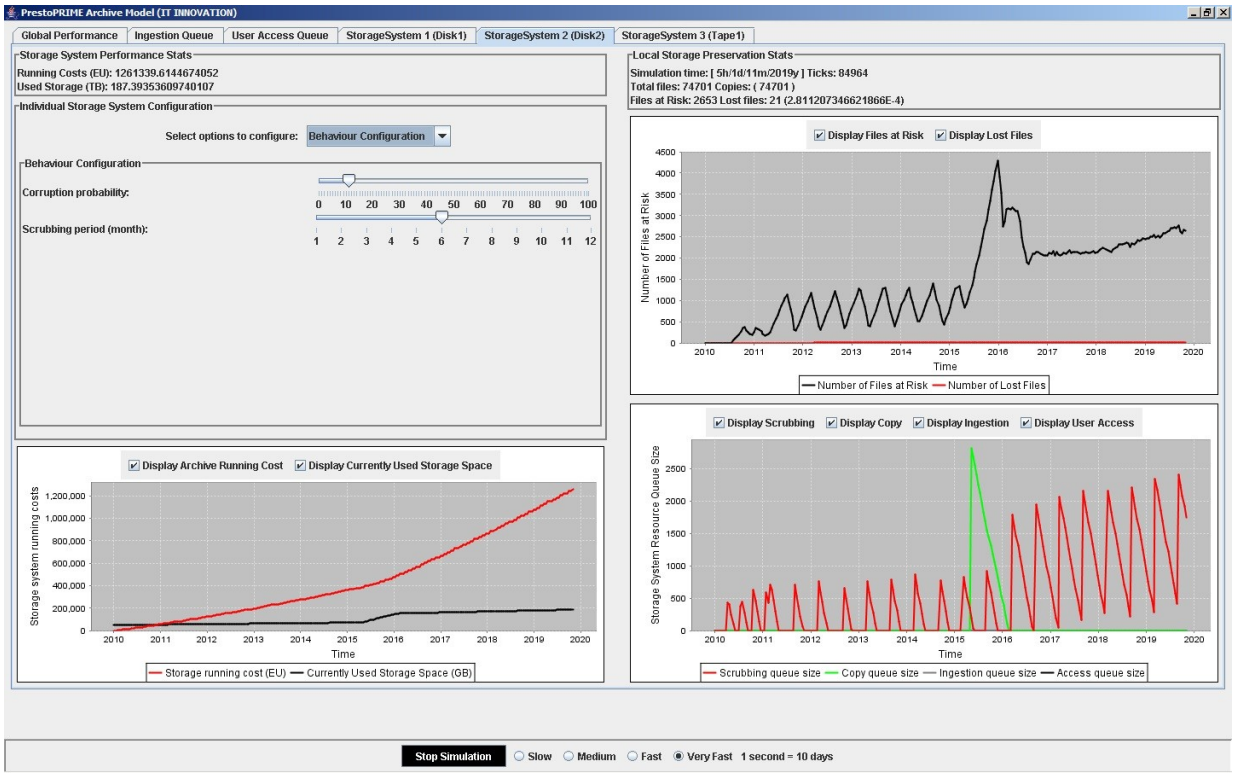

Figure 25 When the simulation is running, the user has access to all the parameters from the set-up stage but now also has a set of graphs that show costs and loss over time. For each storage system the user can see the storage used, the cost of this storage, the number of files at risk (the copy on the storage system has been corrupted, but other copies on other storage systems are OK) and the files lost (all copies of the file have been lost). The user can see the size of queues for scrubbing, copying, ingest and access.

[4] http://en.wikipedia.org/wiki/Poisson_process

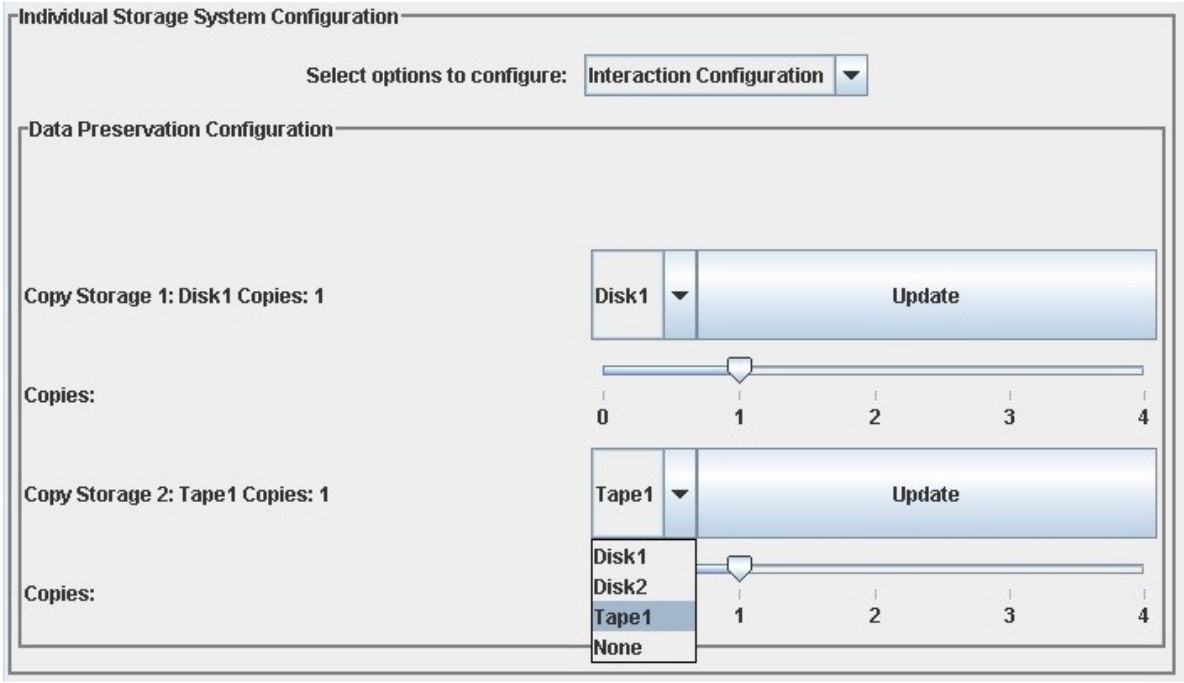

Figure 26 During the execution of the simulation, the user can choose to increase the number of copies made of each file, e.g. because they are seeing too many files being lost and want to increase the level of file safety.

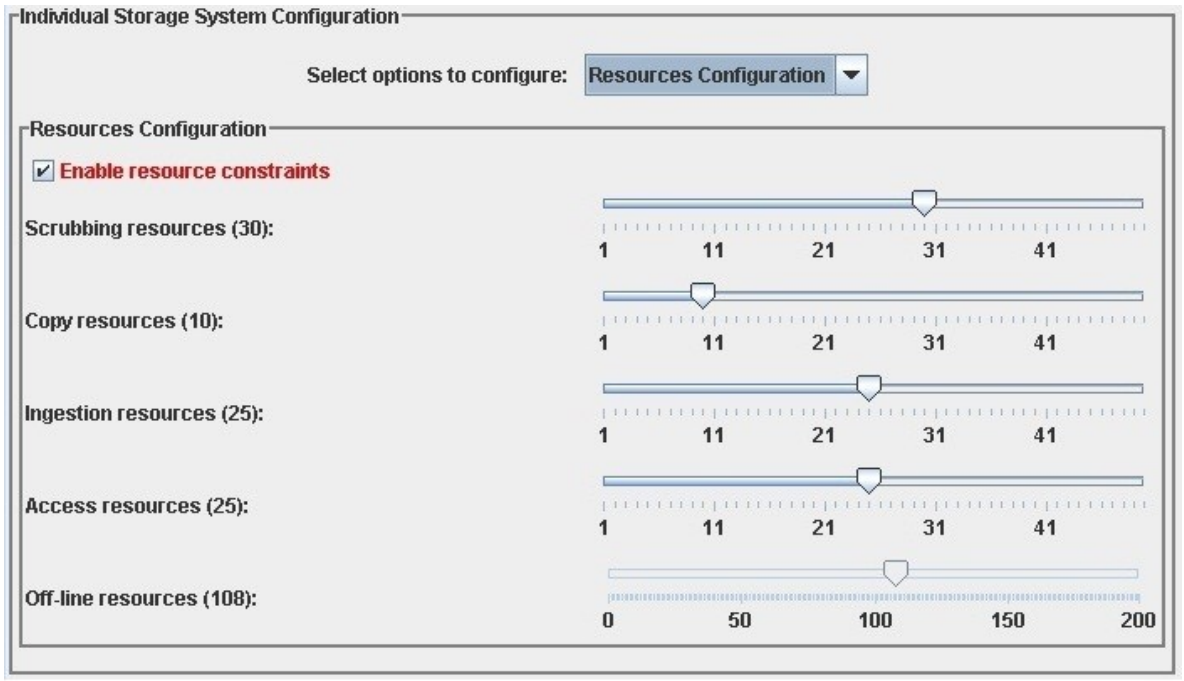

Figure 27 The user can also change the allocation of resources, e.g. because there either insufficient resources for a particular activity, or because too many resources have been allocated and they are underutilised. The objective is to achieve a high level of resource utilisation whilst still meeting ingest, access and data safety needs.

5.2.1 Ingest

The simulation models the arrival of new files to the archive system. The files to be ingested are initially introduced into the queue from which they are picked by the storage system configured to ingest them. Ingest is normally to one storage system, but it is possible to select multiple systems e.g. if the model is for a mirrored model. Once the file

becomes ingested and the necessary number of copies (according to the storage management policy) is made, the file is removed from the queue.

The speed of processing the ingestion queue is dependent on the number of ingestion resources set on the storage system(s) that performs ingestion.

The user is able to observe the number of files queued up for ingestion together with the monthly ingestion throughput. The interactive interface allows to dynamically activate ingestion, set file to ingest arrival rate as well as choose which storage system should be responsible for file ingestion.

Instead of directly setting a specific ingestion storage system, the user is also provided with an option that activates adaptive selection of storage systems for ingestion. When selected, the model adaptively chooses ingestion storage system that has the least resource utilization of all the storage systems. This minimizes the risk of resource overutilization and thus has the potential to speed up the file ingestion process. The result is automatic load balancing of ingest across storage systems

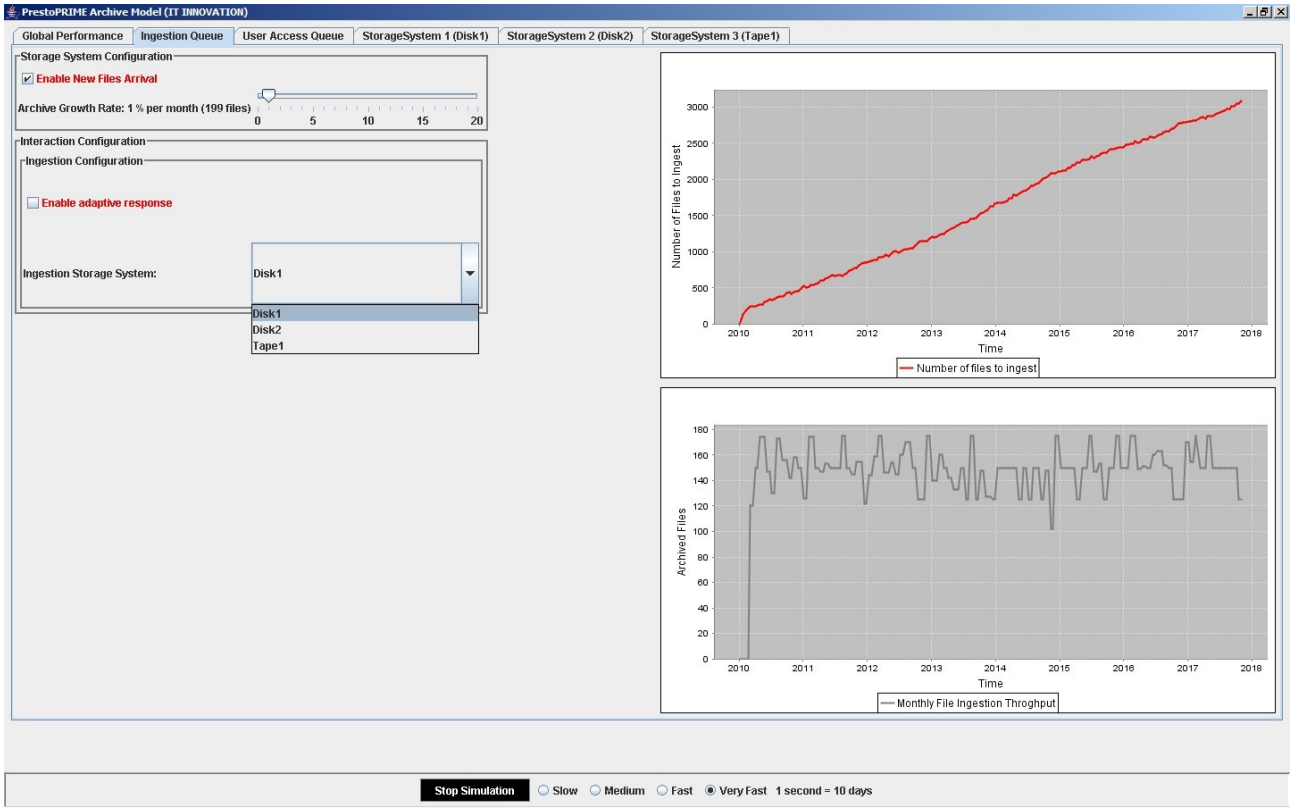

Figure 28 The way the system is responding to ingest workload can be investigated in terms of ingest throughput and queue size.

5.2.2 Access

The user can monitor and manage the user access queue. Analogous to the ingestion queue interface, the user is provided with information about the number of files that await access as well as the monthly throughput of accessed files. By interacting with the interface, the user is able to activate user access, set file to access arrival rate as well as choose which storage system should provide access to archived files. It is possible to specify multiple systems to serve access requests, e.g. to model a mirrored system, or to investigate the effects of fail-over if one system is temporarily set as having no resources for access (see 'catastrophic events' Section 5.3

As with ingest, the tool supports an adaptive selection of storage systems responsible for access. In this case, instead of the user specifying a specific storage system for access, the tool will adaptively choose an access storage system that has the least resource utilization of all storage systems. This simulates load balancing across storage systems. The objective is to increase throughput of access requests and hence minimize the risk of overloading archive system resources.

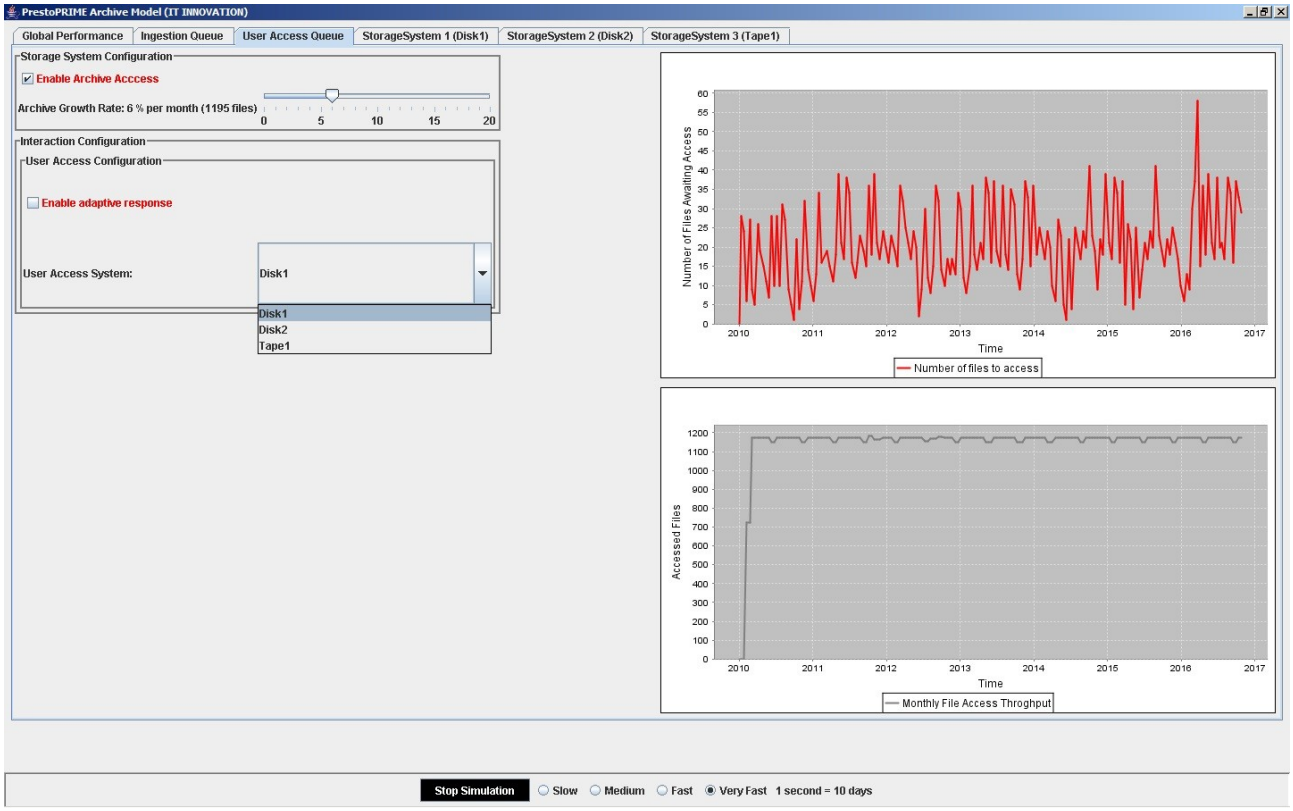

Figure 29 Access queue and throughput of serving file access requests

5.2.3 Global system behaviour and performance

The model can be initialized in three different configurations depending on the number of storage systems used. These include: one storage system configuration where only single storage system exists along with ingestion and user access queues; two storage systems with both queues; and, finally, three storage systems plus the queues.

When the simulation is running, random data corruption events trigger the corruption of the archived files. The frequency of these corruption events is given by the probability for each storage system.

Each storage system counteracts corruption events by performing file integrity checks and repair (involving making a copy of a known good file to replace a known corrupted file). This checking can be scheduled, e.g. once a month, or it can rely on user access to files (opportunistic scrubbing).

Such operations may require interaction with other storage systems in order to fix the corrupted file using a file copy located on the other storage system. Apart from data preservation operations, selected storage systems also respond to file ingestion and user access requests arriving from ingestion and access queues.

Whilst storage systems perform above described actions independently, asynchronously and possibly concurrently with other storage systems, the overall system performance depends also on the reliability and fluidity of the interactions between these systems. The degree to which it is achieved depends strongly on the efficient management of system resources. For example, the successful file ingestion requires not only the file to be ingested and archived on one of the storage systems but may also involve its copy distribution to other storage systems. If these systems have insufficient number of copy resources, they may slow down copy operation and thus risk the newly archived file to become corrupted before its redundant copies are made. Relevant in this situation is monitoring of the system performance and, if necessary, timely readjustment of the system configuration. The tool provides the means to achieve both in an interactive manner that allows the user (considered as the system administrator) to be involved in the management-loop.

Global system monitoring is realized within the model with the help of Global Performance interface. Apart from displaying the global running cost and used storage, the user is provided with run-time updated plots of all storage system performances presenting the number of files at risk and lost files. Monitoring these data, the user may quickly identify anomalies in one of the storage system's performance and attempt to interactively adjust the system configuration to bring the system back to the controlled state. Such a 'preservation game' may also be inverted, allowing the user to change various local storage system configurations (e.g., allocation of resources for different services or scheduled scrubbing period) and observe how such changes affect the global system performance.

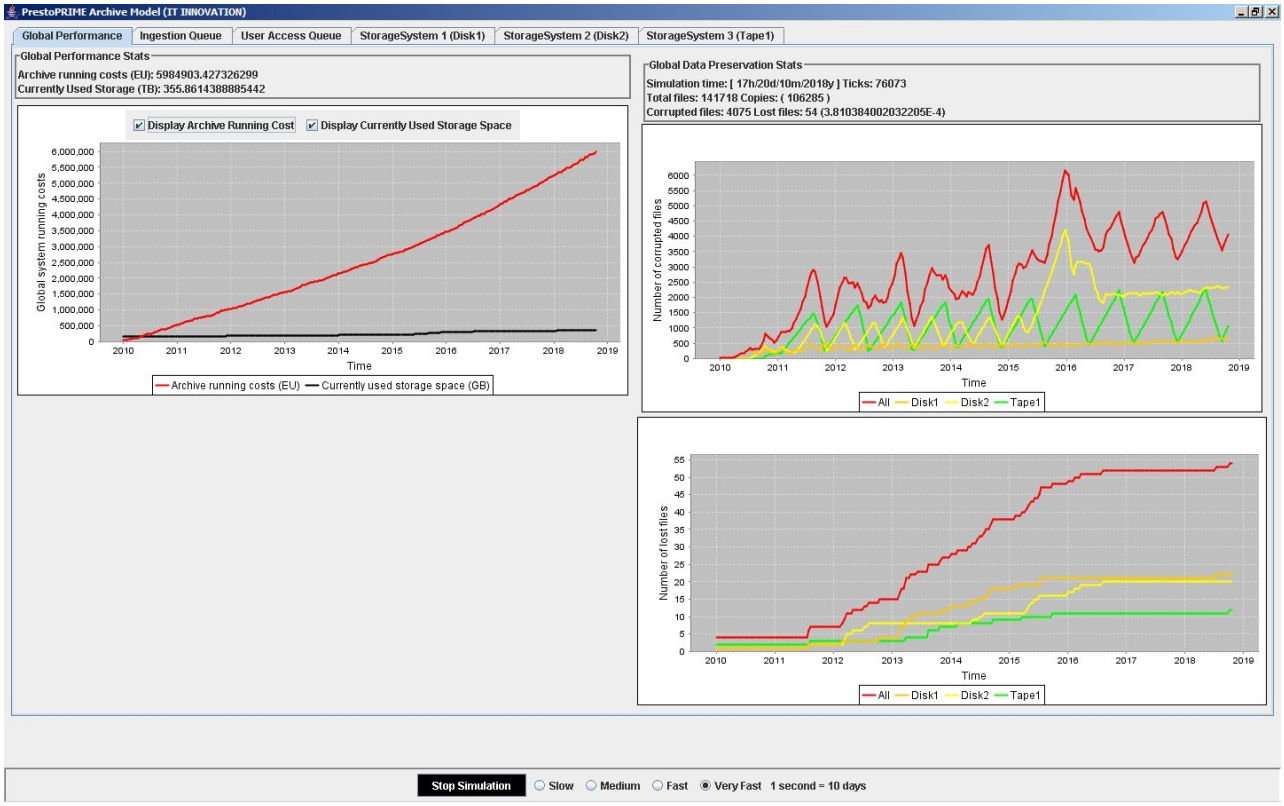

Figure 30 Global Performance view of the archive as a complete system showing overall numbers of files at risk or lost, including the main culprits.

5.3 Catastrophic events

The simulation tool presented so far includes corruption events that happen randomly and affect individual files (e.g. bit rot in a hard disk storage system, damage to a data tape in a tape drive). However, other loss modes exist which threaten much larger numbers of files, e.g. a whole RAID array being lost, or perhaps even fire, flood, theft or other catastrophic types of disaster (a catastrophe can include operator error, e.g. a person instructing the system to perform the wrong operation by mistake such as deleting files, or an operator failing to respond to errors such as not noticing that a copy operation has failed). These events will typically happen very infrequently and hence can't be included in the simulation in the same way as individual file corruptions.

To accommodate the simulation of 'catastrophic events', the tool allows the user to define such events in terms of the number of files lost in a given storage system (it could be all of them), whether this loss is permanent or not (e.g. physical loss of a storage server, or just temporary loss of access to a storage site), or whether the loss is resource (e.g. people used to do file ingest) or files (e.g. because a RAID array fails). The user can then press a button to initiate the catastrophe at the time of their choosing during the simulation and then observe the consequences and evaluate recovery options (e.g. to instigate extra replication in the other parts of the system to ensure a given number of file copies is maintained). Outside of the scope of the model is also the possibility and cost of rescue operations being performed, e.g. using data recovery tools or specialist service providers.

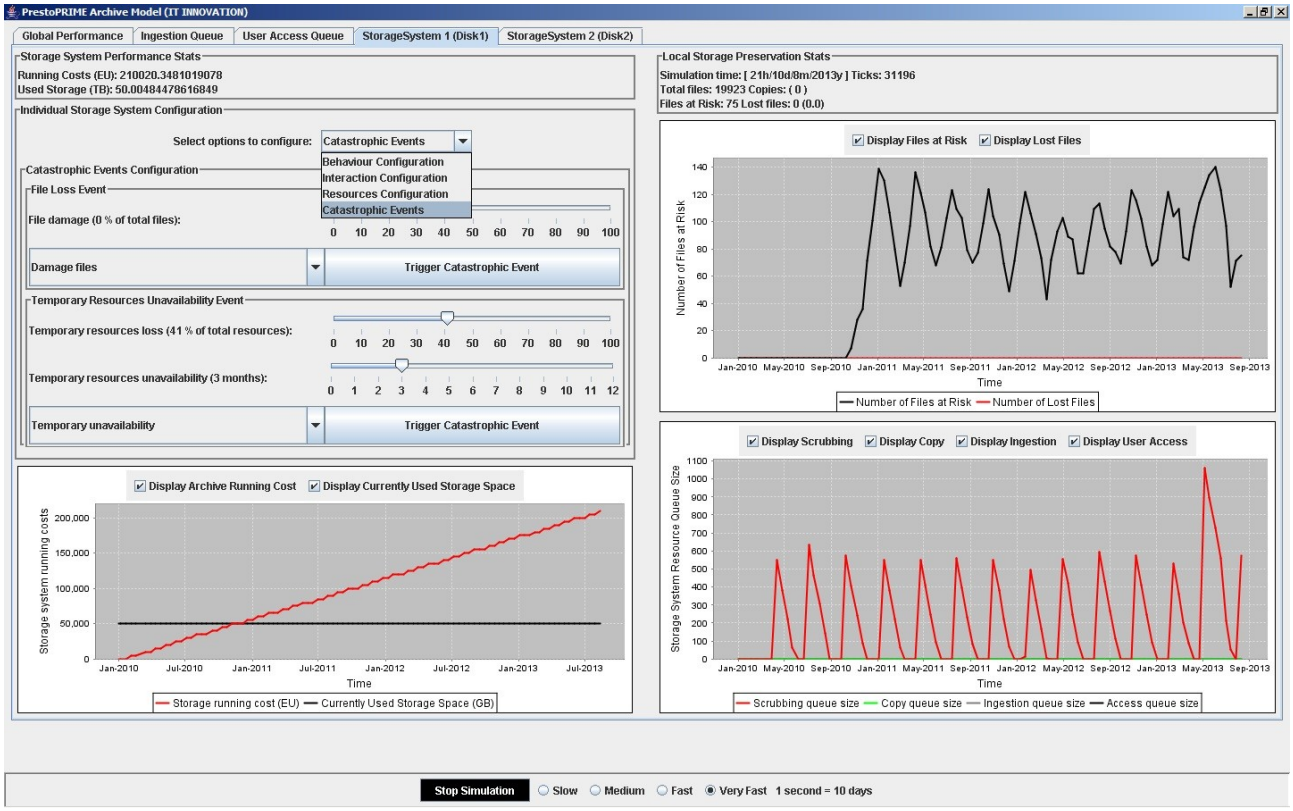

Figure 31 The user can describe catastrophic events and then trigger their occurrence whenever they want during the simulation.

Figure 32 Catastrophic file loss events can result in the permanent corruption or recoverable damage to files in a storage system (e.g. this might correspond to complete loss of a RAID array or only the need to rebuild the array in order to recover the files).

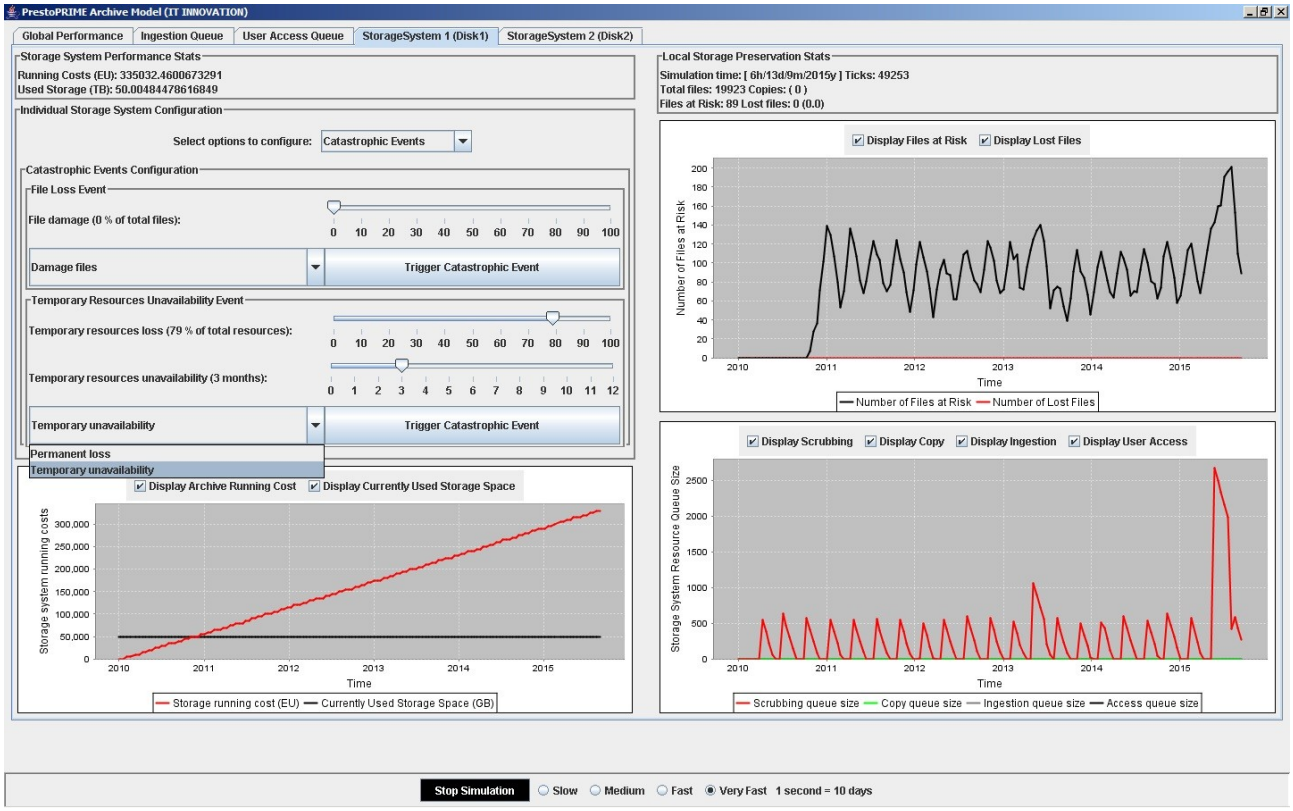

Figure 33 Catastrophic events can be loss of resources (temporary or permanent) as well as loss of files.

6 Parameters: what do they mean, where do they come from

One of the challenges with using either the long-term planning or interactive simulation tools is providing suitable values for the input parameters, e.g. costs of different storage systems or probabilities of data corruption.

The models strike a balance between accuracy, complexity and flexibility. For example, a detailed and accurate cost model can be built for a tape storage system, but this requires an extensive set of parameters and results in a model that this inflexible, i.e. hard to adapt to other storage approaches. On the other hand, a model that is too simple won't be accurate enough to allow meaningful comparisons between storage approaches or to provide even initial input to an investment case. In our tools we try to strike a balance using a small set of parameters and trends that can be used to represent a wide range of storage types and storage-related preservation processes (e.g. migration).

To help users of the tool, we provide a set of default values. This section explains how they have been calculated.

6.1 Corruption in storage systems (risk of loss)

Both tools allow the user to enter the rate at which files are corrupted when they are stored or accessed.

A corruption could be a single bit flip or it could be a complete loss of the file. It could be caused by bit rot in a RAID array or it could be caused by someone dropping a tape or hard drive stored on a shelf.

Whatever the type of corruption, the tool considers that a copy of the file has been 'lost' if it that has been corrupted, in other words the copy is irretrievably damaged and unusable at this point (more on this in Section 7). If all copies of the file are lost then the file as a whole is considered lost at this point.

6.1.1 Example file corruption rates

In some cases corruption rates can depend on how big the files are (e.g. the bigger the file then the more likely that bit rot in a HDD system will flip at least one of the bits). In other cases, corruption rates depend on factors not related to the file, e.g. how likely it is that someone drops a data tape when taking it off a shelf, or how likely it is that a tape drive will damage the tape. Damage to the tape can be considered as loss of all the files on the tape irrespective of whether they are large and small.

Some example corruption rates are below.

25GB file (1 hour of SD video at 50MBit/sec, e.g. D10)

HDD storage (servers with RAID etc.)
- Latent: 1 in 750 files on average per year
- Access: 1 in 500 files on average when file is retrieved

HDD storage (disks on shelves)
- Latent: 1 in 100 files on average per year
- Access: 1 in 500 files on average when file is retrieved

Tape storage (tapes in a robot)
- Latent: 1 in 100,000 files on average per year
- Access: 1 in 10,000 files on average when file is retrieved

Tape storage (tapes on shelves)
- Latent: 1 in 10,000 files on average per year
- Access: 1 in 1,000 files on average when file is retrieved

1 TB file (1 hour of uncompressed HD video, tar of the DPX files from 2k film scan)

HDD storage (servers with RAID etc.)
- Latent: 1 in 20 files on average per year
- Access: 1 in 10 files on average when file is retrieved

HDD storage (disks on shelves)
- Latent: 1 in 100 files on average per year
- Access: 1 in 10 files on average when file is retrieved

Tape storage (tapes in a robot)
- Latent: 1 in 100,000 files on average per year
- Access: 1 in 10,000 files on average when file is retrieved

Tape storage (tapes on shelves)
- Latent: 1 in 10,000 files on average per year
- Access: 1 in 1,000 files on average when file is retrieved

As can be seen, some of the numbers can be scaled, e.g. 1TB files are 40x more likely to be corrupted using HDD storage servers as a result of bit rot, but are not necessarily more likely to have an error when held in a tape robot or as tapes on shelves.

6.1.2 Estimating latent file corruption rates

Latent corruption describes corruption or failures that occur that aren't immediately detected, i.e. you don't find out about them until you actively access or check the data.

Examples include 'bit rot' in HDD systems, but equally this could be used to describe media deterioration or failures in off-line media, e.g. tapes or HDD on shelves.

Hard disks in servers.

When investigating latent corruption in HDD servers, CERN found 1 in every 3 million blocks corrupted due to RAID controller problems when they looked at 8.7TB of user data in 33700 files. The block size was 64KB. They contacted the manufacturer who fixed the firmware and improved the situation by a factor of 100. This can be considered as latent corruption, e.g. as time progresses and the system continues to be used then the number

of errors will increase, but won't be detected until the files are explicitly checked. Even after manufacturer intervention, latent corruption was still present.

- A 25GB file has 4 x 10^5 blocks. 1-in-750 files will have an error on average
- A 1TB file has 1.6x10^7 blocks. 1-in-19 files will have an error on average.

When using RAID systems, if a disk fails and the array will need to be rebuilt. Normally this process takes place without any data loss (which is the whole point of RAID). However, if there is there is an uncorrectable read error from one of the remaining disks during the rebuild then this may get built into the rebuilt data set (parity pollution), although this does not happen in all types of RAID (RAID5 is more susceptible than RAID6). The data corruption rates here depend on both the HDD failure rates and the Bit Error rates for those HDD. Provided that a modern storage system is used with an appropriate RAID level (e.g. RAID6) or a file system that actively manages integrity (e.g. ZFS), the corruption rates are likely to be lower than other types of failure mode as observed by CERN and others (e.g. NetApp), so aren't considered further in this document.

Hard drives on shelves, JBOD servers

The annual failure rate (AFR) for modern SATA HDD is around 0.5-1%. Storing HDD on shelves or using a JBOD server model (Just a Bunch of Disks, i.e. no RAID) will result in approx 1 in 100 files being lost on average per year (given a suitably large number of drives).

Data tapes on shelves

Data tape, e.g. LTO, is a relatively reliable storage technology. Provided the media is not frequently used (by which we mean not hundreds or thousands of cycles, only access a few times a year), and if storage is in a controlled environment (temp, humidity, contaminants), then latent corruption will be very low.

It is more likely that other problems will cause data loss, e.g. errors introduced by manual operation such as data tapes being misshelved or not returned. For example, a reasonable estimate would be say 1 in 10,000 tapes being lost each year. This can be modelled as a latent error rate of 1 in 10^4 files being lost each year.

Data tape in robots

The combination of reliable media and automated handling by a tape robot means the probabilities of latent corruption are very low. Typically data is verified on write (e.g. LTO) and with infrequent access is unlikely to develop problems before the media becomes obsolete and needs to be migrated (e.g. every 6 years skipping generations in the LTO roadmap).

If there are problems, then this will often be related to a 'bad batch' of tapes from a manufacturer, e.g. a production line problem. Given that media is often purchased in bulk it is not uncommon for a set of tapes from the same manufacturing batch to be used in a tape robot. This means that whilst data tape is reliable, if there are problems then several tapes could be affected at the same time. The probability of this happening is very hard to estimate, so we chose 1 in 10^5 files being lost due to tape failures as an initial default value pending better information from the archive or tape vendor communities.

6.1.3 Access corruption.

When data is accessed there is always some risk of damage or loss caused by the system used to hold that data. For example, the risk that a data tape is damaged in a drive, dropped by a manual operator, or that reading the data from HDD has a head crash or torn read or some other form of problem. There is also the chance that the system simply returns the wrong data, e.g. result of misdirected reads or writes in a HDD system or an indexing error in a tape library.

Data tape

PrestoPRIME partners were asked about problems with data tape and tended to say that the drives were the issue not the data tapes themselves. Typical problem rates reported were 0.1%. This is an example of access corruption (the tapes are fine until they are put in drives).

Data tape lifetime is limited by the number of times a cartridge can be loaded/unloaded (typ. 10,000) and the number of data read/write cycles. Tests on LTO tape on how many times data can be read suggest that 100,000 passes[5] are possible from a signal level degradation perspective. Practical tests on the number of times a tape should be used for read+write come up with lower numbers[6] (e.g. for backup applications it might be is a few hundred, but this is based on using the whole tape and not just retrieving a subset of the data on it).

Overall, for an archive application, the probability of a data tape itself causing read errors for a given file is low, e.g. 1 file in 10^5 having problems would appear conservative.

Hard disks on shelves

Manufacturer specifications for unrecoverable Bit Error Rates are between 1 in 10^{14} and 1 in 10^{15} for commodity SATA HDD (For data tape they are 1 in 10^{17} or lower).

A 25GB file (e.g. 1 hour of SD video at 50MBit/sec) has 2×10^{11} bits and hence the chance of an error when reading the file from HDD is between 2×10^{-3} and 2×10^{-4}. I.e. the rate at which files are corrupted is between 1-in-500 and 1-in-5000.

A 1TB file (e.g. 1 hour of 1080p HD video uncompressed) would have a chance of an error when reading the file from HDD of between 0.01 and 0.1. I.e. that rate at which files are corrupted is between 1-in-10 and 1-in-100.

Hard disks in servers

Read errors from HDD systems depend on whether integrity is checked before files are returned (typically not the case in RAID5 etc.) and whether any errors that do occur are uncorrectable or not. Therefore, error rates are dependent on the specifics of the HDD server approach chosen.

As starting point, we could assume that read errors from a well-chosen HDD server are an order of magnitude better than for bare HDD (i.e. the hard disks on shelves model). How-

[5] http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5257121&tag=1
[6] http://www.buzzle.com/articles/hp-lto-3ultrium-data-cartridges-comparative-brand-testing.html

ever, in order to be conservative, we use the same values as for HDD on shelves, i.e. 1-in-500 25GB files will see some form of corruption on access.

6.2 Costs of storage systems

In our tools, storage costs are split into cost of storing files and the cost of accessing files.

Cost of storage is the annual cost of storing a copy of a file for 1 year. The cost of access is the cost of retrieving the file each time it needs to be accessed. Access includes retrieving the file for any reason, which could be user access, but also access in order to do scrubbing (integrity check, e.g. running a checksum), migration (e.g. copying a file between HDD or data tapes), replication (e.g. copying a file from one storage system to another in order to make multiple copies), or repair (e.g. copying a file from one storage system to another to replace a corrupted copy).

Cost modelling is often a contentious issue. Accurate models require huge effort to develop and are typically obsolete within a year or two because of the continual advances made in the storage industry. Simple models can be longer lasting and easier to develop, but may not give the accuracy required for financial planning.

It is not our objective to create complex and detailed cost models for use in our tools. Instead the tool uses a single ‘cost per GB per year’ number for storage and ‘cost per GB’ for access.

The advantage of a simple model that this can be used to describe a wide range of options, e.g. HDD on shelves through to Amazon storage as a service and can be used to cover the Total Cost of Ownership (TCO) of storage (power, cooling, space, maintenance, equipment, media etc.) which is important since the hardware component of storage costs can be as low as 25% of the total cost. The disadvantage of the generalisation we apply is that it is a gross simplification and has little value for detailed cost analysis or budgeting.

6.2.1 Example costs of storage

The costs of storage are often proportional to file size. Therefore our tools support costs per GB as a starting point. The cost of access may also depend on file size, i.e. the data i/o needed (e.g. Amazon S3), but it can also be a fixed cost per file (e.g. someone retrieving a data tape from a shelf).

Example costs

HDD storage (servers with RAID etc.)		
Storage:	1	Euro per GB per year
Access:	0.1	Euro per GB per access
HDD storage (disks on shelves)		
Storage:	0.1	Euro per GB per year
Access:	5	Euro per file access
Tape storage (tapes in a robot)		
Storage	0.5	Euro per GB per year
Access:	0.1	Euro per GB per access

Tape storage (tapes on shelves)

Storage:	0.05	Euro per GB per year
Access:	5	Euro per file access

The cost of storage can also depend on how much content is being stored, i.e. the type of systems needed and the economies of scale that may be possible. The Amazon S3 rate card is a good example[7] where the cost of storage per GB falls by a factor of 3 depending on whether 50TB is being stored or more than 5PB.

6.2.2 Estimating costs of storage

One quick way to get example storage and access costs is to use figures from online storage services, for example Amazon S3 below. Likewise, rates are published by the San Diego Super Computing Centre (SDSC)[8] which currently stand at 390Euro per TB per year for dual-copy tape and 690Euro per TB per year for single copy SATA spinning disk. These are only an indication of storage costs and will need adjustment for archiving instead of online storage. However, because these are costs charged by service providers they do include all cost elements (power, cooling, space, kit etc.) and hence are to some extent representative of the true total cost of storage.

Pricing

Pay only for what you use. There is no minimum fee. Estimate your monthly bill using the AWS Simple Monthly Calculator.
We charge less where our costs are less, and prices are based on the location of your Amazon S3 bucket.

US – Standard | US – N. California | **EU – Ireland** | APAC – Singapore

Storage (Designed for 99.999999999% Durability)		Reduced Redundancy Storage (Designed for 99.99% Durability)		Data Transfer*		Requests	
Tier	**Pricing**	**Tier**	**Pricing**	**Tier**	**Pricing**	**Type**	**Pricing**
First 50 TB / Month of Storage Used	$0.150 per GB	First 50 TB / Month of Storage Used	$0.100 per GB	All Data Transfer In	Free until November 1st, 2010**	PUT, COPY, POST, or LIST	$0.01 per 1,000 Requests
Next 50 TB / Month of Storage Used	$0.140 per GB	Next 50 TB / Month of Storage Used	$0.093 per GB	First 1 GB / month data transfer out	$0.000 per GB	GET and All Other Requests***	$0.01 per 10,000 Requests
Next 400 TB / Month of Storage Used	$0.130 per GB	Next 400 TB / Month of Storage Used	$0.087 per GB	Up to 10 TB / month data transfer out	$0.150 per GB		
Next 500 TB / Month of Storage Used	$0.105 per GB	Next 500 TB / Month of Storage Used	$0.070 per GB	Next 40 TB / month data transfer out	$0.110 per GB		
Next 4000 TB / Month of Storage Used	$0.080 per GB	Next 4000 TB / Month of Storage Used	$0.053 per GB	Next 100 TB / month data transfer out	$0.090 per GB		
Storage Used / Month Over 5000 TB	$0.055 per GB	Storage Used / Month Over 5000 TB	$0.037 per GB	Greater than 150 TB / month data transfer out	$0.080 per GB		

Figure 34 Storage costs for Amazon S3

[7] http://aws.amazon.com/s3/#pricing
[8] http://www.sdsc.edu/services/StorageBackup.html

Another approach is to calculate the in-house cost of storage. As a simple example, suppose the cost of 500TB in a tape robot is 150kEuro capex plus 50kEuro per annum opex (media, data centre, support etc.). Over a 5 year amortisation period, TCO is 80kEuro per annum, which for 500TB is 0.160Euro per GB per year.

An estimate of storage costs could be based on the use of off the shelf storage servers and commercial support (in which case dealing with storage vendors is the fastest route to getting real costs although some information on hardware costs is also available from StorageMojo[9]).

An estimate of storage costs could be based on what is possible from DIY efforts, e.g. BackBlaze Storage Pods[10], or it could come from experience of others in their efforts to find or create low-cost reliable and scalable storage solutions, e.g. the Internet Archive Petabox[11]

Storage and access costs can also be calculated for a 'media on shelves' model instead of a media in servers/robot model. For example, suppose the archive consists of tapes or HDD on shelves. To retrieve a file involves retrieving the media from the shelf, loading it into a tape drive or HDD cradle, finding and extracting the file, then delivering it to the user. Suppose this process takes 10 minutes per access (1 person can service 40 requests per day = 8000 per year). Suppose total staff cost 50kEuro per year (incl. overheads etc.). This makes 1 access approximately 5Euro.

An alternative way to calculating access costs is to work top down, e.g. suppose an archive serves 1million items per year and has an annual running cost of 2million Euro. This could be used to estimate that the average cost of access to an item is 2Euro (of course a more detailed analysis would factor out other activities, e.g. cataloguing, storage etc.).

More detailed analysis of the costs of storage and access are provided in PrestoPRIME deliverable D2.1.1 'preservation strategies' and D6.3.1 'Business models and calculation mechanisms'.

[9] http://www.storagemojo.com
[10] http://blog.backblaze.com/2009/09/01/petabytes-on-a-budget-how-to-build-cheap-cloud-storage/
[11] http://www.archive.org/web/petabox.php

7 Assumptions and simplifications

The tools produced necessarily make several simplifications and assumptions. There is a trade-off between accuracy, complexity and ease-of-use. The tools are designed so the simplifications made could be removed if necessary, although this would require more development effort. The most notable simplifications include:

If a copy of a file is corrupted then copy is considered lost. If all copies of a file are corrupted then the whole file is considered permanently lost.

In some cases, flipping just one bit of a file will render it difficult or impossible to open or use, e.g. if it is zip compressed or encrypted. On the other hand, some files can sustain significant damage and still be usable, e.g. uncompressed video. The impact of data corruption on the usability of a file depends on the type of content it contains and its encoding. This can become complicated when content is also wrapped, e.g. MXF, where, for example, corruption of header information can be catastrophic but corruption of essence might be much less so. There is also the possibility of attempting some form of data recovery or repair, including the use of special recovery tools or service providers, which can often come at significant cost (time, labour, equipment). Concealment is another option, e.g. applying digital restoration tools or simply just dropping a video frame. Some examples of the effects of data corruption on AV content can be found in PrestoPRIME ID3.2.1.

The tools we have developed take the worst case scenario that any type of data corruption in a file will result in the permanent inability to access the contents of that file. If further information is known on what percentage of data corruption events can be recovered from, or simply accepted as tolerable, then the results of the tool can be adjusted accordingly.

The only risks to files are those that arise from the storage system containing them.

There are many risks to AV content when using IT systems for preservation. Many are listed in PrestoPRIME ID3.2.1. In the current tools we focus on a specific subset of these risks that come from storage, e.g. data corruption.

We do not explicitly include factors such as the risk of accidental deletion or the risk of deliberate attacks, e.g. theft. These events can be accommodated to some extent in the interactive simulation tool by treating them as catastrophic events.

The cost of each storage system in an archive can be considered as independent.

The tools have a simple model that the total cost is the sum of the costs of the individual storage systems. In practice, there can be significant additional costs in combining and integrating storage systems, not just the obvious need for hardware for the physical connection, e.g. networking infrastructure, but also software to manage the system. Commercial products for management of HSM, distributed or heterogeneous storage systems can add major extra costs. Likewise, operators of the combined system need to have a wider range of skills and hence will command higher salaries or be harder to find. These costs need to be considered, e.g. whether it is more cost effective to (a) add a data tape tier to a HDD system or (b) simply replicate the HDD system so there are two identical instances.

The costs can be simplified to annual cost of storage + cost of access.

As described elsewhere in this report, the cost model we use is extremely simple. If more detailed and accurate cost modelling is required then this will need to be done outside of the tools and probably in the context of the specific organisation's accounting system. The benefit of a simple model is in its flexibility to describe a wide range of scenarios.

Ingest, access and corruption events happen at random.

We consider ingest, access and corruption to be random and uncorrelated events. For example, we consider corruption to be equally likely for all files and to be independent of previous observed corruptions. However, in the real world, events are often correlated, e.g. bursts of ingest or access requests. If one event happens then others may then follow, e.g. if one file in a given collection is accessed then it may be that the rest will shortly follow. Or if one data tape in a batch from a particular manufacturer has a problem, then the others from the same batch may also exhibit problems too.

All the copies of a file are identical.

Modelling all copies a given file to be identical is the simplest situation to manage, and reflects the typical approach taken by most archives. However, other options might be viable, e.g. using different files formats for different copies so that compression would be used to save space and reduce I/O on expensive HDD servers but uncompressed formats used for infrequently accessed deep archive data tape copies.

All files are equal in value.

In our tools, all files are considered equal in terms of likelihood of access or corruption. They all have the same level of replication. They are all given the same priority for repair or retrieval. In the real world, not all files have equal value and hence different levels of cost/safety might be applied. For example, this might be done considering the relative value of different genres of content in an archive (news, sport, drama, children, regional etc.) and deciding to have a different storage strategy for each. 'Value' is a complex thing and very hard to incorporate directly into a model (e.g. archives don't ascribe a numerical score to files to indicate their value). Therefore, the approach we take is to allow the archive to decide how to 'partition up' their content into collections of files with different cost/safety requirements and then run a model for each.

All files are kept forever.

The model does not include retention scheduling with reclassification/removal of files. However, not all content needs to be kept forever. Archives tend to review on a regular basis (e.g. every 5 years) whether to continue retaining a file or not. It may be that a file is dropped or it may be that the cost/safety balance is revised. The current model simply treats all files as having a retention period of 'forever' and no change to the level of safety required during that time.

Files are just data.

In the tools, all files are considered as blobs of data with no special AV characteristics that need to be taken into account. In practice, files contain AV assets, e.g. a TV programme, with metadata, video, audio, timecodes, subtitles etc. Not all parts should necessarily be treated equally. For example:

- A higher level of safety might be applied to the metadata and audio compared to the video in an MXF file (e.g. video content is often the bulk of a file and hence is the bulk of the cost but some corruption of the video might be tolerable, whereas audio takes up less of the file and no corruption is tolerable. Therefore, it makes sense to have more copies of the audio part than the video part.

- Some parts of the programme material will be more important than others (e.g. consider the 10 o'clock news where the headlines and key stories are more important than some of the other content). Again, these might be treated differently e.g. by making more copies, scrubbing more often or using different storage systems.

- There may be ways to split up video into parts that are more important to protect than others. This might be frequency bands (e.g. in JPEG2000) or it might be separating I and P frames in MPEG2. Likewise, integrity checking and repair tools could monitor the content of a file and not just the file as a whole, for example as being developed by RAI for D10 video where checksums are maintained for each video frame.

The user has the knowledge needed to set all the input values.

Both tools assume that the user has easy access to all the input data needed, e.g. the number and size of files in their archive, the corruption rates in storage, the per-GB access and storage costs etc. For example, the tool would expect the user to input that they have say 30,000 files each of which is 10GB in size. However, the user may be more comfortable with the number of programme hours they have and what format it is in, e.g. 10,000 hours in IMX format with an average length of 20 minutes per programme. Likewise, the user may know how many files they have historically lost in their archive, but not how this translates into specific failure rates. Or the user may know how many staff they have working in an archive and hence might want to simulate staff illness or job loss, but the user may not know how this translates into specific archive activities, e.g. ingest, preservation actions etc.

It is possible to translate between the 'business level' of archive activities and resources and the 'technical level' of file sizes, volumes, rates etc. Explicit support for this isn't included in this tool[12] Depending on how easy tool users find it to provide the input data to the tools, we will investigate 'translation tools' to help the users map from one to the other.

[12] Simple conversion tools were included in PrestoSpace calculators, e.g. converting from feet of shelves to TB of data.

8 Implementation of the models

8.1 Modelling Costs and Data Loss in File Storage Systems

Data loss in files includes data corruption (file has same number of bits, but the values are changed) and data deletion (file has fewer bits, i.e. some have been removed). Conceivably, a file could also increase in length (e.g. inserting or appending extra bits), but this is uncommon. Data loss can be all or part of a file.

The impact of data loss can also be variable (ranging from inability to open the file at all in a given application through to just one pixel being changed in one frame of a video).

In our model we consider that storage systems have the function of accepting files for storage (writes), returning files from storage (reads) and storing the data inside the files using some form of physical media (hard drive platter, data tape, optical disc etc.).

This simple model can be applied to automated hardware/software, e.g. a HDD server, or it can be applied to a more manual process, e.g. data tapes on shelves with archive staff that put new tapes onto shelves and retrieve existing tapes to serve user access requests.

In the process of recording files to the physical media, various operations may be applied to the data in the file when written by the storage system, e.g. encoding it to add redundancy and applying error correction when it is read back. We model this as being done through some form of 'controller', which might be the firmware on a HDD, the RAID controller in a HDD array, integrity management in a ZFS filesystem, or a combination of all of these.

Again the idea of a 'controller' could equally be applied to staff in an archive who manage discrete items of media on shelves, e.g. by making replicas or periodically checking condition.

This process is shown in Figure 35 where the storage system appears as a 'black box' in terms of file reading and writing, but internally it has some form of controller that determines exactly how the physical media is used to store the data.

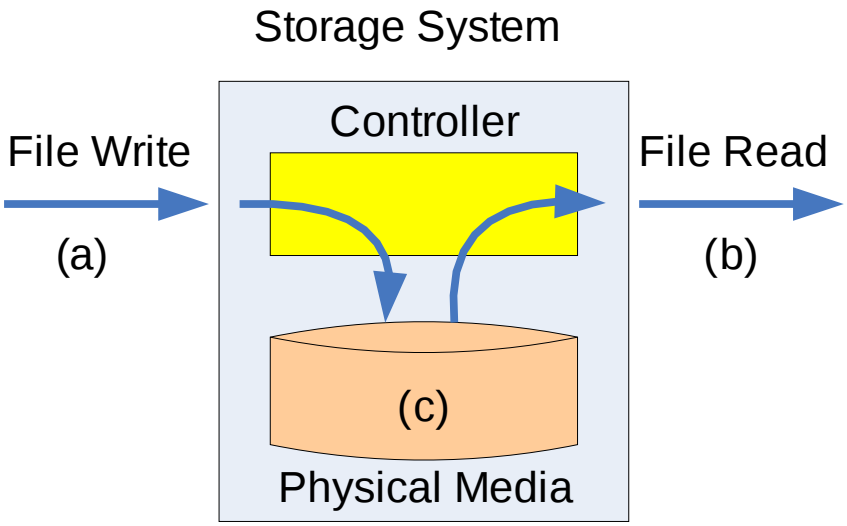

Figure 35 Abstract representation of a storage system

With reference to the diagram, each of the three activities of (a) file write, i.e. ingest, (b) file read, i.e. access and (c) physical storage will have a cost associated with them.

Therefore, our cost model of storage consists of:

- One off ingest cost per file when adding it to a storage system
- Access cost per file incurred each and every time it is retrieved from the storage system
- Storage cost per file when it is inside the storage system with the cost being a function of how long the file has been stored for.

With reference to the diagram, when considering data losses, we recognised that these can happen (a) when data is written, (b) when data is read, and (c) when the data on the physical media is in effect 'doing nothing'.

For example, consider a HDD. When data is written there might be a torn write resulting in some of the data for a file being lost. When data is read there might be a bug in the controller or vibration of read head - either resulting in the wrong data for part of the file being returned. When the data is stored, then misdirected writes in other files, head crashes, magnetic media deterioration, cosmic rays and other effects can all cause the data to be corrupted. Note that some errors are not necessarily permanent, e.g. an error might be introduced into a file due to a misdirected read, but the error might not be present the next time the file is read – therefore, data is not permanently lost in this case.

Some errors can be guarded against, e.g. using checksums and verifying that a file has been written properly by reading it back from the storage system immediately afterwards and checking it is OK. Sometimes this is built into the storage system (e.g. LTO data tape includes verify on write) and other times it can be added using external checks, but this comes at a cost due to the extra work done. In our model we assume that all files undergo some form of 'verify on write' so that errors on ingest are eliminated. This is then reflected into the cost of the storage system.

In the case of human operators being part of the 'storage system', other errors might occur, e.g. accidental damage when handling media, accidental mis-shelving or mis-cataloguing media. As with errors in automated systems, we model these as being either errors that occur as part of day-to-day storage or errors that occur when items in the archive are accessed for users.

Since the internals of the storage system are not always visible, it doesn't really make sense to think of a file as somehow intact (or not) whilst it is within the storage. You can give a file to the storage system and then you can ask for it back – you only know for sure whether there is any data loss after you have received the file from the system and checked that it is OK.

Neither does it make sense to try to map failure rates of the media inside the storage system to probability of data loss when retrieving files from that storage system. For example, the Annual Failure Rate (AFR) of SATA HDD can be as high as 10%. Yet if these drives are used inside a RAID6 array and a failed HDD is replaced promptly, then there is little chance of data loss (two drives can fail before errors have the chance of creeping in during array rebuild). Therefore, there is typically little correlation between

media error rates and data loss rates. Indeed it can be that other components of the system introduce new and more significant errors, e.g. bugs in RAID firmware or accidental damage by staff (for more details see PrestoPRIME ID3.2.1).

Returning to the issue of what data loss may be seen when reading files from a storage system, data loss within a file can be characterised in terms of whether it is:

- **Correctable/Uncorrectable**
    - Correctable: read errors frequently occur when using storage media (HDD, DVDs etc.), but they are typically correctable because of redundancy added when a file is written to that media (e.g. CDs use multiple levels of error correction[13]). This is an example of a correctable error, i.e. the result after the inbuilt error correction mechanism of the storage device is that the file returned has no error.
    - Uncorrectable: these are permanent errors where despite efforts by the storage device or system, the data returned is not the same as the data originally written. For example, for HDD this is the uncorrectable bit error rate (BER) which represents the proportion of data returned during a read that can't be guaranteed to be the same as the data originally written[14].
- **Silent (latent)/reported (extant)**
    - Silent: errors that are introduced into the file but are not reported by the system used to store the file. In other words, the storage system either believes the data returned in a file is correct, or it fails to report that it has found an error. Note that both are possible, e.g. a bug in RAID firmware might prevent error reporting, or a misdirected write on a HDD could cause a block that belongs to file A to be accidentally replaced by a block from file B, but the checksum for the block would still be valid so the HDD wouldn't think anything was wrong.
    - Reported: errors that are detected by the storage system and reported in a timescales that is to all intents and purposes instant, e.g. reported failed write at the time the file is written or read.
- **Located/Unlocated**
    - Located: the location of the data loss is known, e.g. which block in a file is affected. This can be important, e.g. knowing where a 512 byte block has been corrupted within a 1TB file.
    - Unlocated: the location is not known, e.g. computing the hashcode for a file before/after storage shows that there is an error somewhere in the file, but it doesn't reveal where that error is.

The errors themselves may be **deletion, corruption or addition** of bits, bytes, blocks or whole files. For example, corruption might be random bit flipping or it might be all the corrupted bits being set to zero. The errors will also have some form of **distribution** where

[13] http://home.btconnect.com/geffers/cd.html
[14] http://delivery.acm.org/10.1145/1320000/1317403/p28-elerath.pdf?key1=1317403&key2=1196889721&coll=ACM&dl=ACM&CFID=95778333&CFTOKEN=78327209

corruption could be randomly distributed, repeating or follow some other form of pattern. Corruption might be contiguous in one part of a file or distributed across a file

The corruption of data in a file also translates in various ways to impact on the content (audio, video etc.) in that file (e.g. corrupted pixels, blocking artefacts, dropped frames etc.) – some of which can be concealed (e.g. replacing corrupted blocks in one frame with good blocks from the previous frame – see DV tape as an example).

In our tools we do not attempt to build a full model of data corruption and possible recovery/repair options, but instead consider the simplified case:

- Any form of data loss in a file will make the whole file unusable. Therefore, we can ignore the distribution of data loss within the file, including whether the problem is corruption or deletion, or whether the loss can be located or not within the file. We can also make the assumption that any correctable errors within a storage system are indeed corrected (e.g. HDD failures result in RAID rebuilds where necessary) and this is reflected in the operational cost of that storage system.

- We consider silent uncorrected (permanent or temporary) data loss in files when in storage, and the corruption of files when read from storage system.

Integrity checking techniques can reduce the probability of data loss or detect that a data loss has occurred. For example, external data integrity checking could be used to verify that a write to a storage system had been successful, e.g. by reading the file back and comparing checksums. Integrity checking operations have a cost associated with them, so being able to include/exclude them in the model would allow the cost/benefits to be explored.

An example is shown in Figure 36 below. The x axis starts when the file is read. The y axis shows the probability of some form of data loss in that file when it is read back.

- The red line shows what might happen if no external integrity checks are used, i.e. there is a chance of data loss at initial write, the chance of loss then goes up over time e.g. because other misdirected writes might corrupt the data, and there is the chance of further data loss being introduced at read time.

- The orange line shows the use of verify on write to reduce the initial write error (e.g. repeated write attempts are made until the file verifies).

- The green line then shows the use of verify on read to catch any temporary read errors (with repeated reads made until the data is returned correctly). There are of course still residual errors, e.g. those that accumulate when the data is in storage or are permanent read errors (e.g. tape failures on playback).

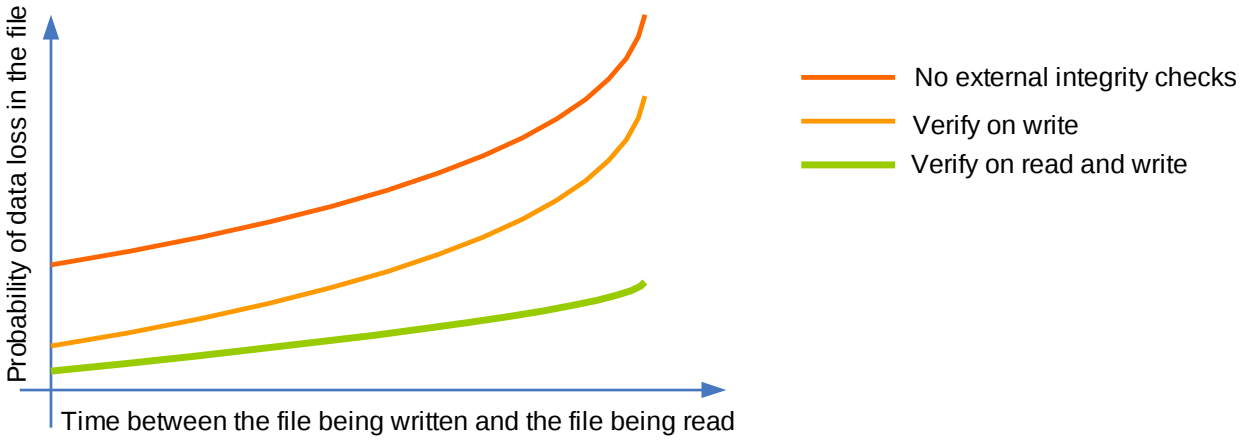

Figure 36 How probability of data loss in a file read from a storage system might depend on use of integrity measures at read or write time

In summary, we model a storage system as having:

- On-going costs associated with storing a file within the storage system
- Cost of accessing the file each time it is retrieved from the storage system
- One-off cost of ingesting a new file into the storage system
- Probability of a file being lost due to the act of accessing that file, which applies every time the file is accessed.
- Probability of a file being lost due to some form of latent corruption when it is inside the system and this probability increases in proportion to the time spent in the storage system.

8.2 Implementation of the long-term planning tool

The long-term planning tool adopts a Markov Model approach.

Markov chain models have been used before for modelling losses in storage systems, e.g. using a Continuous Time Markov Chain (CTMC) approach[15] [16]. CTMC relies on several assumptions including: an exponential distribution of the waiting time in each state; the probability of transitioning between states is solely dependent on the current state and not on previous states (the memoryless property); and that the probability of transitions is not a function of time. In our case the waiting time for state transitions is not typically exponentially distributed (e.g. consider scrubbing or migrations which are both periodic activities). Likewise, storage system failures and hence the rate of data corruption might follow a 'bathtub' curve, i.e. it is a function of time. The probability of repair might also

[15] The Modeling System Reliability For Digital Preservation: Model Modification and Four-Copy Model Study. Yan Han, Chi Pak Chan The University of Arizona Libraries. iPRES 2008. http://www.bl.uk/ipres2008/presentations_day2/44_Han.pdf

[16] Constantopoulos, P., Doerr, M., and Petraki, M. 2005. Reliability modelling for long term digital preservation. http://delos-wp5.ukoln.ac.uk/forums/dig-rep-workshop/constantopoulos-1.pdf

increase the longer an item is known to have been in a corrupted state (i.e. priority is given to repairing failed files that have been known to be that way for some time).

We assume that the 'memoryless' feature of Markov Processes (the Markov property) still holds, e.g. the probability of accessing a file in the future is not dependent on whether it was accessed in the past, and that the probability that a file will be corrupted is not dependent on whether it has been corrupted before. This allows us to continue with a Markov model approach, but instead of a CTMC we deal with the time dependent state transition probabilities through a Discrete Time Markov Chain (DTMC). The transition probabilities are not time homogeneous, i.e. they are not stationary, so, in the model, time will tick away in discrete steps and at each tick we can construct a suitable matrix of transition probabilities that hold at that time and determine the allowable state transitions. For example, if there was a disk scrubbing activity every six months, then the probability of transition from a state where a file was corrupted but undetected to a state where the file was corrupted and this was detected would be 1. A simple simulation approach is to construct the matrix P of transition probabilities at each time t and then multiply all these together to give a matrix describing the probability of ending up in the various states of the system after total time T.

An example state model and set of transitions is shown in Figure 37 below which represents a 2 file copy model where the files are on different storage systems.

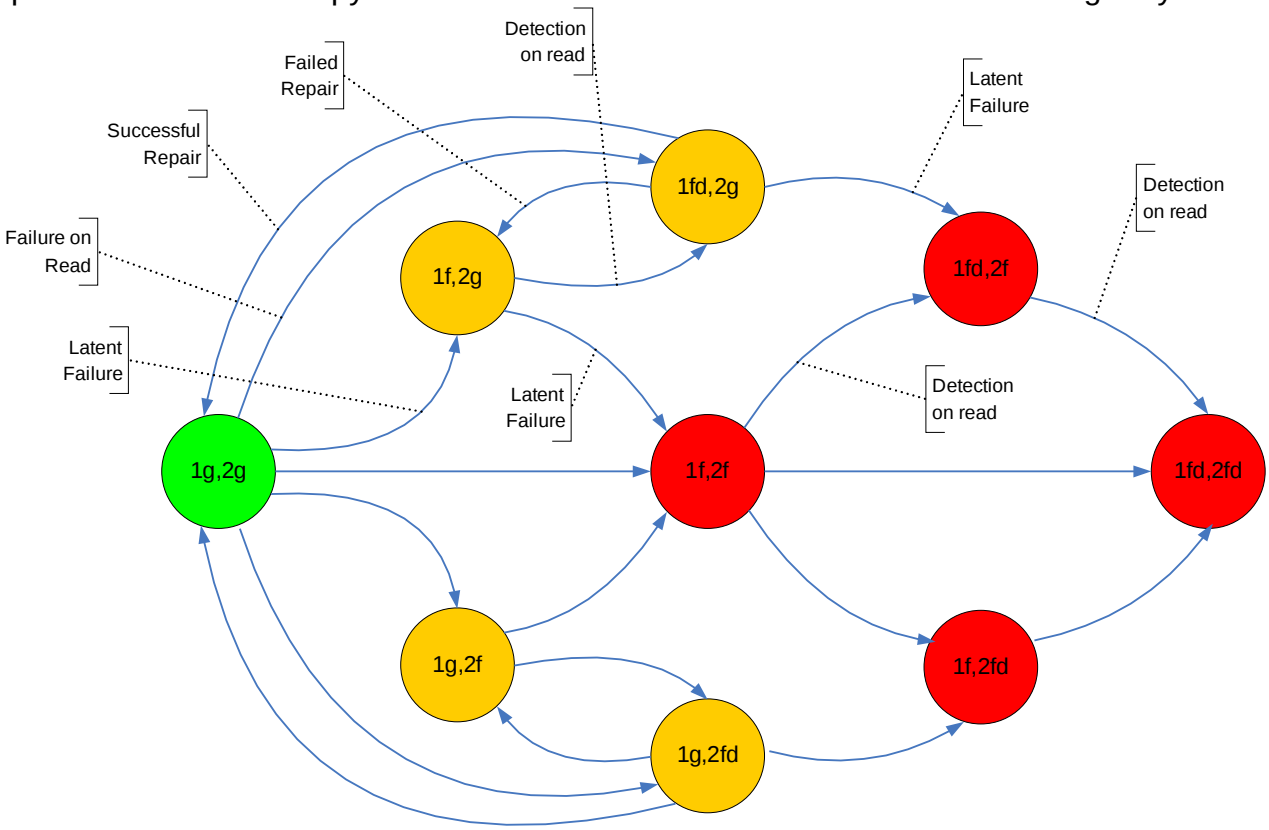

Figure 37 Markov Model showing the states for a 2 copy model.

Notation:
1 = first file copy
2 = second file copy
g = good
f = failed
fd = failed and detected

This diagram includes both extant and silent corruption modes. For example, state transition 1g,2g →1f,2g is silent corruption where file copy2 has become corrupted but this corruption is not yet detected. In contrast, 1g,2g →1fd,2g is extant corruption, e.g. reading a file from a data tape where the drive mangles the tape, which would be immediately detected. The model also includes failures on write, e.g. 1fd, 2g →1f,2g is where repair is undertaken but is actually unsuccessful for example where there is an undetected torn write so the storage system thinks it has written the data properly but in fact hasn't.

The approach that we take is to decompose the transitions into a series of separate matrices, each of which corresponds to a particular activity, e.g. latent corruption, access, scrubbing. The product of these matrices is an approximation of what happens in the real world. So, for example, if L is the latent corruption per month, A is the access-triggered repair each month, S is scrubbing at the end of each year, then the model is the product: L.A.L.A.L.A.L.A.L.A.L.A.L.A.L.A.L.A.L.A.L.A.L.A.S...

In reality, L and A take place concurrently rather than sequentially, however provided that the transition probabilities in these matrices are small, then applying them sequentially is a valid approximation. Using separate matrices simplifies understanding and maintenance.

The model was initially developed in Matlab[17] and then deployed behind a Web based GUI using Octave[18] as an open source and freely available alternative. The use of a completely non-proprietary stack (Apache, python, Octave) allows plenty of flexibility to host the web-tool on a range of web-servers, including the PrestoCentre site, and hence make the tool available to a wide community of users.

8.3 Implementation of the interactive simulation tool

The interactive simulation tool takes a discrete event simulation approach[19].

The simulation contains one or more storage systems, each of which is modelled as providing a set of services (e.g. ingest, access, storage) where each service uses one or more resources (e.g. copying data, checking integrity). Requests to use a service are added to a queue for that service (e.g. queue of files to be ingest).

During the simulation, time ticks away and events are generated (e.g. random corruption of files in a storage system, requests to access a file, new files to be added to the archive). These events can trigger actions, e.g. a copy/repair process might be triggered if a file access event identifies that a copy of a file is corrupted. These actions are added to the relevant service queues (e.g. file access queue for access events, file copy queue used as part of a repair process or scheduled file migration).

A storage system will process items in the queues for its services according to how much resource it has available (e.g. serving access requests sequentially or in parallel). The available capacity of the resources used by each service determines how many items in the queue for that service will be processed for each tick of the clock. If there is insufficient resource then not all items in a queue will be dealt with and the unprocessed items in the queue will be carried over to the next tick of the clock.

[17] http://www.mathworks.com/products/matlab/
[18] http://www.gnu.org/software/octave/
[19] http://en.wikipedia.org/wiki/Discrete_event_simulation

For a simulation of more than one storage system, a series of interactions are defined between storage systems, for example replicating files.

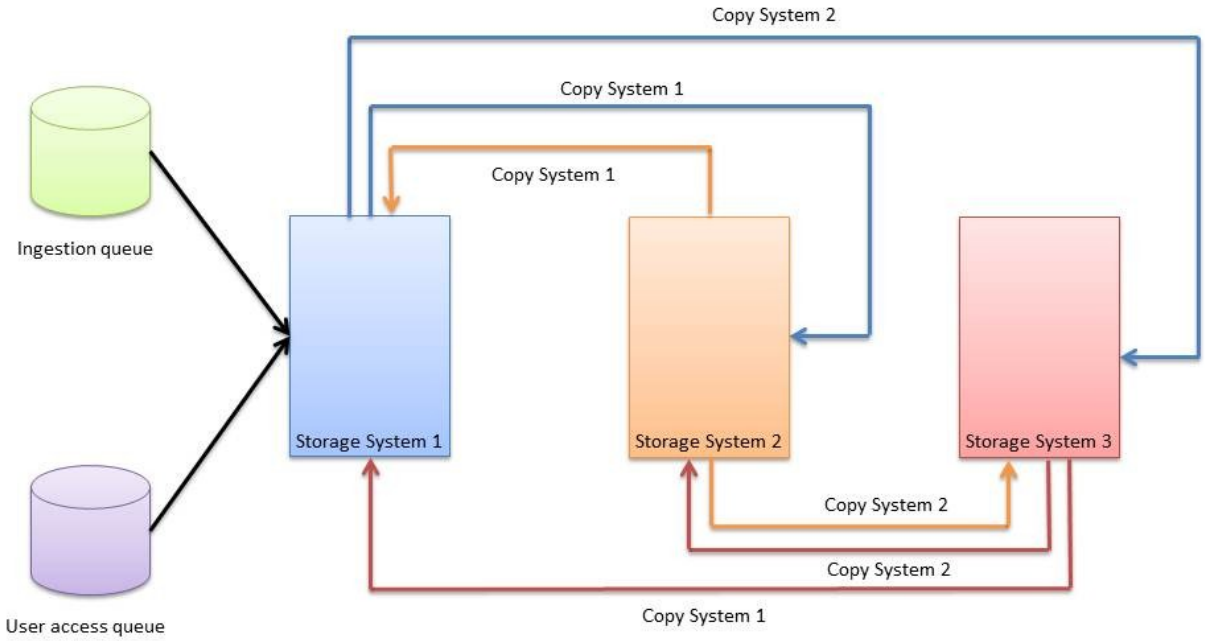

Figure 38 Example storage configuration for the interactive simulation tool showing three storage systems with System 1 used for ingest and access and the other two systems used to store replicated files.

In this way, the services for the storage systems become coupled. For example, if storage system 1 is used for ingest of files and the policy is to replicate these files to storage system 2 and storage system 3, then the rate at which items will be processed on the ingest queue is dependent on the copy resources available to create replicas of the file on the other storage systems. A set of template configurations are provided that correspond to common patterns for real world storage configurations, e.g. mirrored servers, HSM, online + deep archive.

The core of the simulation is a relatively simple one – a set of services with queues and resources, a set of event generators, and a set of template configurations for how storage systems are connected together.

On top of the core simulation is the user interface that allows the user to set parameters, interact with the simulation, and view results. This is where specific UI features are used, e.g. sliders, radio buttons, auto scaling graphs, easy tabbing between storage systems – all of which are designed to make the tool easy to use and tailored to the problem of cost and loss simulation.

The cost model used by the simulation is similar to that of the long-term planning tool in that ingest, access and storage all have a cost. In the interactive simulation tool the costs can be modelled in a finer grained way with costs associated with each resource for each storage system, e.g. resources used for copying files, checking their integrity, providing access. This allows further detail to be easily added, e.g. if there are specific costs associated with user access to files (e.g. checking rights) that might not be needed for other forms of 'system access' e.g. accessing a file so it can be copied across storage systems.

The tool is implemented in Java. The tool will be available in early 2011 from the PrestoCentre as a freely available standalone tool that people will be able to download and run on their desktop. An installer will be used to make the installation as simple as possible, e.g. a 'one click' process.

Existing simulation frameworks were considered (e.g. Simul8[20], iGrafx[21], SimEvents[22], PRISM[23] and others). Whilst some are able to cover the core of the simulation, they all have difficulties when it comes to building custom user interfaces, using non-standard probability distributions or queue disciplines, and allowing user interaction and changes to the settings during simulation. Some of this would make the existing tool hard to develop on one of these platforms and in particular hard to extend to include more complex functionality. There is also the major problem that these frameworks are mostly commercial and expensive to license which would significantly limit the ability to provide the PrestoPRIME tool to the community to use for free.

[20] http://www.simul8.com/
[21] http://www.igrafx.com/products/process/
[22] http://www.mathworks.com/products/simevents/
[23] http://www.prismmodelchecker.org/

9 Conclusion and future developments

This report has described a pair of tools that can be used for investigating the costs and risks of file loss from use of IT storage systems or 'media on shelves' approaches to digital archiving. Applications include long-term planning, storage strategy development, cost estimation, operational 'what if' decision support, and staff training.

These tools are the first in a series of tools that IT Innovation plans to release for use by the AV community through the PrestoCentre. Future tools will cover a much wider range of preservation and access processes, e.g. digitisation and quality control as part of transfer chains, or the impact of file-format selection and migration strategies.

In developing the storage modelling tools a series of simplifications and assumptions were inevitable to ensure that the problem remains tractable, the tool usable, there is a reasonable chance that users will be able to provide the necessary inputs, and yet the outputs will be accurate enough to be useful in a range of contexts. Despite these simplifications, the tool is a considerable advance to the state of the art. The report lays out the design of the model and the simplifications quite carefully. It is important to note that the approach taken does allow more complex scenarios to be modelled, e.g. by aggregating together the results of individual simulations for different parts of an archive.

The tool has been well received by both PrestoPRIME partners during an internal evaluation workshop and by the wider community at the PrestoPRIME public event in November 2010. It is clear that there is a strong demand for this sort of tool and a frequent question was 'when can I use it?' – which will be addressed by making the tool publicly accessible through the PrestoCentre in early 2011.

The approach taken of modelling 'cost of risk of loss' as an example of how archives will always face trade-offs when deciding what preservation strategy and technology to use. This concept of tools allowing the investigation of trade-offs was also well received. This bodes well for the next versions of the tool which will investigate other trade-offs in a quantitative way, e.g. the cost of throughput and quality in transfer chains.