RELIABLE AUDIOVISUAL ARCHIVING USING UNRELIABLE STORAGE TECHNOLOGY AND SERVICES

M. Addis, R. Lowe, L. Middleton, N. Salvo

IT Innovation Centre, UK

ABSTRACT

The drive for online access to archive content within ‘tapeless’ workflows means that mass-storage technology is an inevitable part of modern archive solutions, either in-house or provided as services by third-parties.

But are these solutions safe? Can they assure the data integrity needed for long-term preservation of Petabyte volumes of data? The answer is no. Field studies reveal data corruption can take place silently without detection or correction, including in 'enterprise class' systems explicitly designed to prevent data loss. The reality is that data loss is inevitable to some degree or another from hardware failures, software bugs, and human errors.

This paper presents ongoing work in the UK AVATAR-m project and in the recently started EC PrestoPrime project on a framework for storing large audiovisual files on heterogeneous and distributed storage infrastructures that allows various strategies for content replication, integrity monitoring and repair to be developed and tested.

INTRODUCTION

A large audiovisual archive, e.g. in a national broadcaster, can contain several million items of film, video and audio material. Traditionally this material has been stored as discrete items on shelves, but this is changing. Efforts to digitise ageing and fragile analogue holdings along with the explosion of new born-digital content means that audiovisual archives are now using mass storage technology and storage services to hold their file-based assets.

There are over 100M hours of audiovisual material in Europe's archives. In the professional audiovisual archive world, broadcast archives estimate that they will grow at 5M new hours every year. Increases in bit rates for this new material, especially resulting from the transition from SD to HD and the introduction of digital cinema, mean a data volume that could double in as little as 18 months.

At the same time, digital audiovisual archives are becoming 'embedded' as services within wider networked infrastructures and content-centric processes. In particular, archiving is no longer ‘at the end of the chain’ - a place where content goes to die. Instead, archiving and preservation activities happen throughout the content lifecycle, with content going in and out of a logical archive as it moves through production, post-production, distribution and reuse.

A natural response to this new way of life is to store these large volumes of digital content in an online (network accessible) way using mass-storage technologies such as disk - servers and tape-robots, along with conventional IT solutions for safety, e.g. backup. However, the long-term safety of content using these technologies is far from assured.

In the AVATAR-m [1] and PrestoPrime [2] projects we are developing a framework that allows new approaches to preservation-grade safety to be explored and tested when using commodity IT storage solutions for long-term preservation of audiovisual material. Feature of this framework include: policy-based replication of content across multiple, distributed and heterogeneous storage locations (e.g. how many copies, in how many places and in what formats); automated integrity checking and repair (e.g. how often to check for corruption and what action to take); and media aware decomposition of large AV assets into smaller 'chunks', each of which can have different preservation policies applied to them (e.g. different strategies might be used for the audio, video and metadata components of an MXF object).

STATE OF THE ART

There is a widespread assumption that maintaining bit-level integrity of data files using mass storage technology is a solved problem, e.g. using RAID disk and offsite tape backup. However, the reality is that for large data volumes (e.g. the Petabyte level or above) data corruption or loss can be caused by failures in hardware, bugs in software, and human errors. Field studies of large disk-based systems, e.g. by CERN [3] and [4], reveal data corruption taking place silently without detection or correction, including by 'enterprise class' systems that are explicitly designed to prevent data loss. This is a significant threat to the long-term archiving of audiovisual assets unless proactive and preventative measures are taken. Using storage manufacturer metrics of MTTF (Mean Time To Failure) or MTTDL (Mean Time To Data Loss) are not helpful as a measure of the ability to preserve data long-term [5]. They deal with the case of complete and catastrophic loss of all the data in a system. They neglect that data loss can actually take place incrementally and in a way where the corruption of just a few bits of information can render large parts of a video file unusable due to the way the content is encoded. Furthermore, manufacturer data on MTTF are based on their own models or tests and often don't match observations in the wild by Google [6], NetApp and others which typically reveal much higher failure rates.

The implication for digital archiving when using mass storage technology is the need for a continuous activity of data integrity checking and repair, which in turn requires copies of content to exist in multiple storage systems in multiple different locations.

Simply having many copies of the content in many places is not the answer, not least due to the prohibitive costs. Standard Definition digital video has an uncompressed data rate of about 270 MBit/s and even when stored with compression, e.g. 50MBit/s DV, this means that multiple Petabytes of storage are required for a typical broadcast archive. HD requires five times as much space. In digital cinema, 4K requires up to 30 times the data rate of SD and for 3D cinema with twin data streams at up to 144 fps the volumes are truly vast. This presents a real problem. The cost of maintaining this content is uncertain where estimates range from 'half the price of analogue' [7] to nearly 'twelve times higher' [8]. Current estimates are that it costs $1M for 1PB of storage using online disks, with the cost of tape (in robots) being approximately half of this [9].

Using compression, lossless or lossy, introduces another dimension. Use of compression can save on storage space, and in turn allow more copies to be held for the same cost. However, compression can make the files much more sensitive to data corruption. For example, Heydegger [10] has developed a 'robustness indicator' on the sensitivity of image formats to bit level corruption and investigated how compression affects robustness. This work is notable as it includes JPEG2000, which is emerging as a strong candidate for preservation in the AV community [11] including digital cinema [12]. Tests by Heydegger showed that corrupting only 0.01% of the bytes in a compressed JPEG2000 file, including lossless compression, would result in at least 50% of the original information encoded in the file being affected. In some cases, corrupting just a single byte in a JPEG2000 image would cause highly visible artefacts throughout the whole of that image. The sensitivity of compressed files to bit level corruption is both startling and worrying.

Superficially, the results of Heydegger combined with the 'bit rot' headline findings of NetApp or CERN would imply that maintaining integrity of very large files is near impossible – if bit corruption levels of 10^{-9} exist in the wild (CERN study) and the data files to be stored are 10^{13} bits in size (approx 1TB, which is an hour of uncompressed HD), then it would seem inevitable that these files will become corrupted quite rapidly when stored on disk. However, this neglects the distribution of the corruption. Studies show that corruption is typically block level rather than bit level and tends to be spatially correlated, e.g. successive blocks are more likely to be corrupted than blocks at random [13]. Therefore, corruptions are essentially concentrated rather than evenly spread. Whilst this explains why corruption of large files is not endemic, it does mean further studies are needed on how the files that are hit by integrity loss are actually affected.

Work also exists on encoding AV in a way that makes it more robust to corruption, with JPEG2000 wireless (JPWL) being an example [14]. Redundancy and error checking are built in to improve robustness to errors introduced during transmission over wireless channels. However, whilst this approach, and source/channel coding more generally [15] is used for robust transmission through space, i.e. from one geographical location to another, it has yet to be applied to long-term transmission through time where the channel is the storage solution and noise is introduced by that channel, e.g. silent corruptions.

The use of erasure coding [16] techniques in wide area storage networks has recently become popular, e.g. as used by Permabit [17], especially given the current hype surrounding cloud storage models. This approach accepts that there will be failures in the individual storage nodes and if there are many such nodes then erasure code can be more efficient than brute force replication in achieving high availability [18]. However, there is a reticence in the archive community to use this approach due to the transformation of the data and the inability to recover any of the content if the 'index' that describes the transformation and location of the fragments is lost or corrupted.

It should be clear that the choices to be made are complex and include how many copies to use, how to encode them, where to store them, how frequently to check them, how to repair them efficiently, and how to do all this in a cost effective way. There is no single answer, rather there is a spectrum of options and these will change over time requiring continual reassessment of the strategy chosen and adaption to prevailing conditions. This means monitoring not only integrity but also how corruption is taking place so the way that the content is stored can be adjusted accordingly. The solution for a particular archive depends on how much they are prepared to spend to reduce the risk of loss [19]. The framework we present here as developed in AVATAR-m [20,21] allows these choices to be investigated and implemented.

APPROACH

Our approach is founded on three main areas:

1. Archives want storage solutions that make it easy to combine and use different types of storage (e.g. disk, tape) in different locations (e.g. onsite or remote) where access is through different protocols (e.g. NFS, SMB, ftp, http, scp etc.). This enables the automation of the 'multiple copies in multiple places using multiple technologies' strategy for preservation [22] as well as supporting access for different types of users in different locations with different requirements.

2. Not all AV assets are equal in terms of their needs for safety, accessibility, or longevity. This includes the components of an asset, e.g. metadata, audio and video within an MXF file, or even the constituents of one of these components, e.g. the I,P or B frames within an MPEG stream. A mechanism is needed to allow different rules to be applied at the level of collections, individual assets, and their constituents.

3. Mass storage technology can't be relied upon and no storage technology or storage service provider should ever be assumed to maintain data integrity no matter what their MTTDL or Service Level Agreement might state.

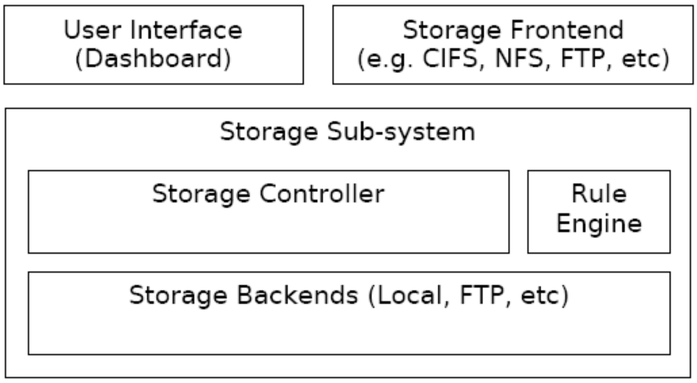


Figure 1 AVATAR architecture

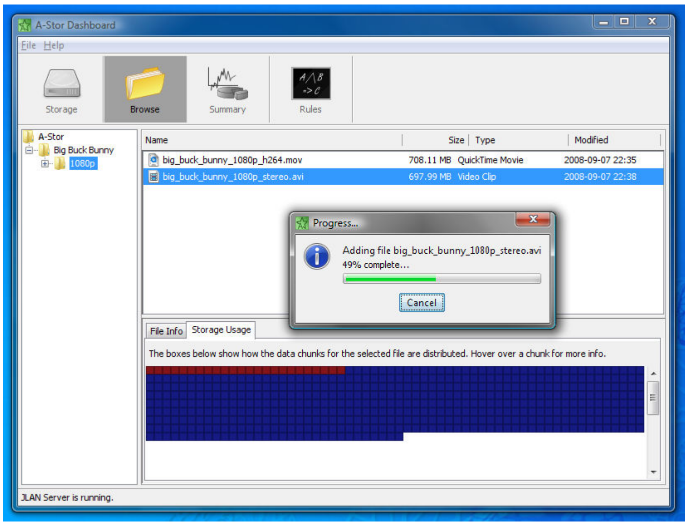
To address point (1) our system allows multiple storage types and locations to be combined and presented to the user of the archive completely transparently using a single access mechanism, e.g. as one file system. Adapters are used for each storage type that the system interfaces to in particular where the interface is not a file system (Figure 1). The emphasis is on seamlessly combining a wide range of networked storage, such as spinning disk or tape, as well as online remote storage provided as a service over protocols such as ftp or http. These disparate storage types and locations aggregated together into a single storage solution. The system maintains a record of the properties of the storage locations, e.g. capacity, bandwidth, availability etc. so they can be ranked or selected appropriately depending on rules on how content should be stored or moved.

To address point (2), files ingest into the system are chopped up into chunks and these chunks are then stored as separate files. A database maintains a mapping from original file to the constituent chunk files. The framework allows plug-ins for 'chunking' so that 'media aware' strategies can be implemented, e.g. to decompose MXF assets. Currently, chunking is done using fixed (configurable) chunk sizes. The ability to support media aware chunking contrasts with conventional distributed storage techniques (e.g. filesystems such as ZFS). Our framework allows strategies to be supported where the boundaries, copies and locations of each chunk can all be optimised according to the modes of corruption and relative significance of the parts of an AV file. The chunks of a given file can be distributed across multiple storage locations (LAN or WAN) and are dynamically reassembled and delivered back to the user of the file 'on demand' and completely transparently (the user simply sees a file on a filesystem and is unaware of how and where the chunks of the file are stored). Not only does this virtualise archive storage from a user perspective, but it has the added benefits of supporting partial restore of files (e.g. using a subset of a video sequence in an editing application) and also allows files to be accessed during the process of migration. For example, a new storage location can be added to the system, a rule set to cause migration of a certain group of files to this new location, but the users of the files can still access them even in the middle of a transfer between locations.

To address point (3) content stored in the system is replicated across multiple locations (how many and which ones are defined as rules). Integrity of content is checked both periodically, i.e. proactively, and when content is accessed or ingest into the system. If corruption is detected then repair takes place by replicating a known good copy from another location. Both CRC and MD5 checksums are used. The frequency of checking and when to repair are configurable. Importantly it is the chunks that are replicated rather than the original files. This significantly reduces the repair overhead as only the specific chunks that are corrupted need to be repaired by copying known good chunks over a network between storage locations.

All three areas are supported by a rule engine that allows various policies to be encoded and automatically enacted. This includes rules applied on ingest, e.g. where to initially place content or how many replicas to create, as well as rules applied periodically that cause content to be checked or moved, e.g. if content is not being accessed then move it to another tier or location to reduce the storage cost (Figure 2).

The performance of storage locations is also monitored and storage locations can be ranked according to performance. Rules can refer to this ranking rather than to explicit locations, e.g. so content of a particular type is always on the fastest tier whatever that might be, or is always kept on the storage showing highest levels of availability or integrity.



Name	Size	Type	Modified
big_buck_bunny_1080p_h264.mov	708.11 MB	QuickTime Movie	2008-09-07 22:35
big_buck_bunny_1080p_stereo.avi	697.99 MB	Video Clip	2008-09-07 22:38

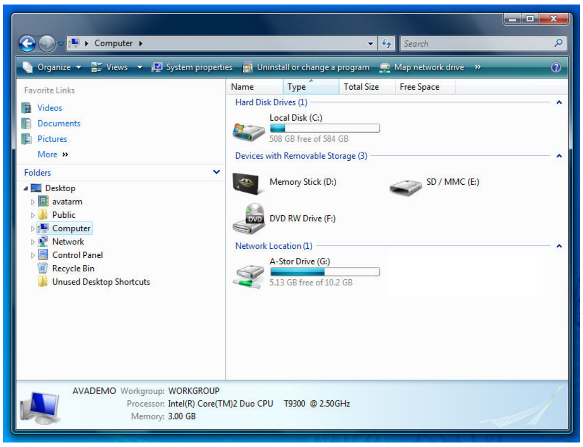
Adding file big_buck_bunny_1080p_stereo.avi
49% complete...

The boxes below show how the data chunks for the selected file are distributed. Hover over a chunk for more info.

JLAN Server is running.

Figure 2 AVATAR data chunking and distribution across locations (blue squares are chunks on one location, red squares are chunks on another location). In this example chunks are assigned to the fastest storage at the time of ingest. Storage location performance monitoring causes the assignment to change during the ingest process.

The details of where a particular file is stored, or how it is accessed and retrieved, are completely hidden from the user. The user is simply provided with access files as if they were on network-attached storage, which may be mapped or mounted to appear as part of the local file system (Figure 3). Whilst transparency is maintained for the user, for the operator of the archive we provide a 'dashboard' (Figure 4) that allows them to see the storage locations, their status and their utilisation. This allows the archive operator to monitor the archive in real time and observe the execution of the rules they define.



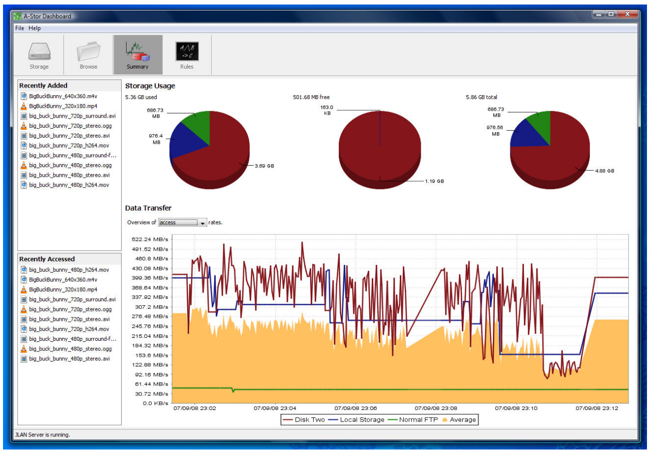
Local Disk (C:)
508 GB free of 584 GB

Memory Stick (D:)
SD / MMC (E:)
DVD RW Drive (F:)

A-Stor Drive (G:)
5.13 GB free of 10.2 GB

AVADEMO Workgroup: WORKGROUP
Processor: Intel(R) Core(TM)2 Duo CPU T9300 @ 2.50GHz
Memory: 3.00 GB

Figure 3 AVATAR archive exposed as a single file share (G:) for archive users.



Recently Added
- BigBuckBunny_640x360.m4v
- BigBuckBunny_320x180.mp4
- big_buck_bunny_720p_surround.avi
- big_buck_bunny_720p_stereo.ogg
- big_buck_bunny_720p_stereo.avi
- big_buck_bunny_720p_h264.mov
- big_buck_bunny_480p_surround-f...
- big_buck_bunny_480p_stereo.ogg
- big_buck_bunny_480p_stereo.avi
- big_buck_bunny_480p_h264.mov

Recently Accessed
- big_buck_bunny_480p_h264.mov
- BigBuckBunny_640x360.m4v
- BigBuckBunny_320x180.mp4
- big_buck_bunny_720p_surround.avi
- big_buck_bunny_720p_stereo.ogg
- big_buck_bunny_720p_stereo.avi
- big_buck_bunny_720p_h264.mov
- big_buck_bunny_480p_surround-f...
- big_buck_bunny_480p_stereo.ogg
- big_buck_bunny_480p_stereo.avi

5.36 GB used
501.68 MB free
5.86 GB total

Overview of access rates.

JLAN Server is running.

Figure 4. AVATAR archive management dashboard for the archive operators.

RESULTS

Having built a framework that allows experimentation with various strategies for long-term high-integrity storage of audiovisual files, tests were done on the performance of this system using a simple chunking, replication and integrity checking strategy. A range of video files in professional formats, e.g. Apple ProRes422, were used up to 411GB in size (HD video encoded at 880MBit/sec). The hardware environment consisted of 3 storage servers connected by a GigE network switch. Each server had 8GB of memory, quad core Xeon processor, 10 SATA hard drives configured as a software RAID array providing 10TB of storage, and 2 further separate drives for the OS and backup storage. The servers ran 64bit Ubuntu 8.10, XFS over dmraid (software) for the filesystem, and were accessible to each other via either NFS or ftp depending on which transfer protocol is under test. The AVATAR-m software is written in Java.

As shown in Figure 5, the ingest time for a video file was measured for a range of chunk and file sizes. Ingest is simply making a single copy of a file (in chunks) and doesn't include replication or hashing. There is a drop in performance for very large files (411GB) using very small chunk sizes (1MB) as a result of the large number of chunks involved and the overheads of writing such a large number of files to disk as well as maintaining the internal AVATAR database. Other than this, ingest rates of 40MByte/sec or higher are achieved. This is close to native performance of the disk from which the data was being ingest.

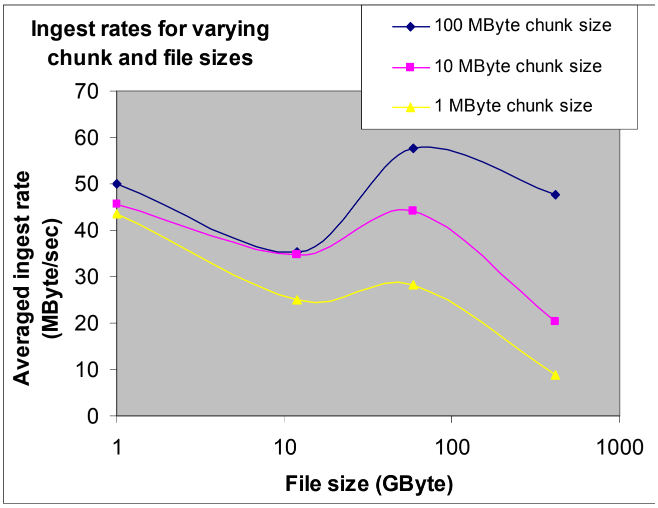


Figure 5 The file ingest rate of AVATAR for a range of file and chunk sizes.

The overhead of generating both a CRC and an MD5 hash on each chunk was quantified by measuring the time taken to hash the chunks in a 59GB file for a range of different chunk sizes. The rate at which checksums could be generated is approximately independent of the chunk size, i.e. it is proportional to the data volume and not number of chunks (1, 10, 100MB chunk sizes resulted in hashing rates of 93, 101 and 107 MByte/sec respectively). This is the rate at which hashing can be done when the data is in memory, e.g. whilst it is being ingest or whilst it is being read from disk for delivery back to an archive user. There is an overhead in generating integrity check data, e.g. on ingest, but this is relatively small compared to the time needed to chunk and store the data. The overhead of replicating the chunks on one of the servers to multiple remote storage servers was measured by using the AVATAR-m software to replicate a 59GB file in 10MB chunks between two servers using NFS. The transfer rate achieved of 37MByte/sec is comparable with the native performance of the system as measured using a direct operating system level copy command for the same set of chunks (43MByte/sec).

To assess the performance of a repair operation, chunks were deliberately corrupted on one of the servers and the time was measured to replace these chunks with known good versions from one of the other servers. This is shown in Figure 6 Repair time is a function of the number of chunks corrupted. Initial integrity checking is constant as all chunks need to be checked, varying only if the operation is done locally to the data or a network transfer is required to retrieve the chunks from another server first.

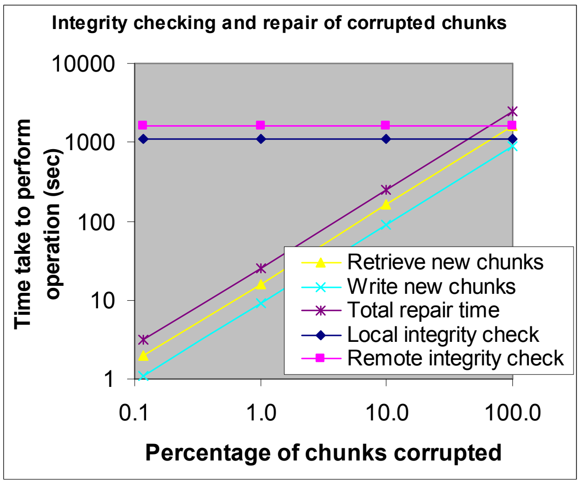


Figure 6 Repair of a 59GB file in 10MB chunks with varying levels of corruption

Combining all the results above indicates that, despite our testbed system only being designed for initial performance testing, if it were given sufficient extra storage capacity then it would be possible to ingest, store and do two complete cycles of integrity checking and repair of half a Petabyte of data per year.

CONCLUSIONS

Using mass-storage technology for audiovisual archives presents many challenges when seeking a cost effective and reliable way to maintaining long-term data integrity whilst still allowing easy access to archive assets. This paper has demonstrated a new framework that allows strategies for to be explored and tested when using commodity IT storage solutions for long-term preservation of audiovisual material. Features include replication of content across multiple, distributed and heterogeneous storage locations according to archive policies, automated and proactive integrity checking and repair, and the ability to deconstruct large AV assets into smaller 'chunks', each of which can have appropriate preservation policies applied to them. The performance of the system has been evaluated using a simple chunking, replication and integrity monitoring strategy which shows that the system has sufficient performance to allow a range more realistic replication and integrity management strategies to be tested, which will be the next phase of our work.

ACKNOWLEDGEMENTS

AVATAR-m is a UK Technology Strategy Board supported collaborative R&D project (November 2006 to February 2010) involving the BBC, Ovation Data Services, Xyratex and the IT Innovation Centre. Please contact avatar-m@it-innovation.soton.ac.uk or see www.avatar-m.org. PrestoPrime is a European Commission supported 7th Framework Programme ICT project (FP7-231161) that started in January 2009 and finishes in 2012. The project is coordinated by INA (Institut national de l'audiovisuel) in France and other partners include BBC, RAI, ORF, B&G to name but a few. For more information please contact prestoprime@it-innovation.soton.ac.uk or see www.prestoprime.org

REFERENCES

[1] www.avatar-m.org.uk
[2] www.prestoprime.org
[3] Silent Corruptions, KELEMEN Péter. CERN IT. LCSC 2007, Linköping, Sweden.
[4] Are Disks the Dominant Contributor for Storage Failures? A Comprehensive Study of Storage Subsystem Failure Characteristics. Weihang Jiang, Chongfeng Hu, and Yuanyuan Zhou, University of Illinois at Urbana-Champaign; Arkady Kanevsky, Network Appliance, Inc. FAST '08 pp. 111–125 of the Proceedings.
[5] Bit Preservation: A Solved Problem? David H Rosenthal, Stanford University. In proceedings of iPRES 2008: The Fifth International Conference on Preservation of Digital Objects, British Library, London. http://www.bl.uk/ipres2008/index.html
[6] Failure Trends in a Large Disk Drive Population, Eduardo Pinheiro, Wolf-Dietrich Weber, Google. Proceedings of the 5th USENIX Conference on File and Storage Technologies (FAST'07), February 2007
[7] "Archiving Movies in a Digital World". David Cavena et al. SUN Microsystems report, VERSION 2.1, June 8, 2007; 29pp David.Cavena@Sun.COM
http://wikis.sun.com/display/SunMediaSpace/2007/11/05/Archiving+Movies+in+a+Digital+World
[8] "The Digital Dilemma: Strategic Issues in Archiving and Accessing Digital Motion Picture Materials" Academy of Motion Picture Arts & Sciences, 2007; 74pp available from the academy: http://www.oscars.org/contact/council.html
[9] Moore, R. L.; D'Aoust, J.; McDonald, R. H.; and Minor, D. 2007. Disk and Tape Storage CostModels. In Archiving 2007.
[10] Heydegger, V (2008) Analysing the Impact of File Formats on Data Integrity. Proceedings of Archiving 2008, Bern, Switzerland, June 24-27; pp 50-55
[11] Pearson, Glenn and Michael Gill, "An Evaluation of Motion JPEG 2000 for Video Archiving", Proc. Archiving 2005 (April 26- 29, Washington, D.C.), IS & T (www.imaging.org), pp. 237-243
[12] Enhanced Digital Cinema project (EDCINE) http://www.edcine.org/intro/
[13]Lakshmi N. Bairavasundaram, Garth R. Goodson, Bianca Schroeder, Andrea C. Arpaci-Dusseau, Remzi H. Arpaci-Dusseau Proceedings of the 6th USENIX conference on File and Storage Technologies (FAST'08) San Jose, California. February 2008.
[14] http://www.jpeg.org/jpeg2000/j2kpart11.html
[15] http://en.wikipedia.org/wiki/Information_theory
[16] http://en.wikipedia.org/wiki/Erasure_coding
[17] http://www.permabit.com/
[18] Erasure Coding vs. Replication: A Quantitative Comparison. Hakim Weatherspoon and John D. Kubiatowicz. Computer Science Division University of California, Berkeley. Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02).
http://oceanstore.cs.berkeley.edu/publications/papers/pdf/erasure_iptps.pdf
[19] Wright, R; Matthew Addis; Ant Miller (2008) The Significance of Storage in the 'Cost of Risk' of Digital Preservation. Proceedings of iPRES 2008:
http://www.bl.uk/ipres2008/ipres2008-proceedings.pdf
[20] SUSTAINABLE ARCHIVING AND STORAGE MANAGEMENT OF AUDIOVISUAL DIGITAL ASSETS. M. Addis, R. Beales, R. Lowe, L. Middleton, C. Norlund, Z. Zlatev. IT Innovation Centre, UK. In proceedings of IBC2008.
[21] A Service Oriented Approach to Online Digital Audiovisual Archives. Matthew Addis, Richard Lowe, Charlotte Norlund, University of Southampton IT Innovation Centre, UK. In proceedings of the NEM Summit 2008. http://www.nem-summit.eu/
[22] http://wiki.prestospace.org/