Bit Preservation: A Solved Problem?

David S. H. Rosenthal
Stanford University Libraries, CA

Abstract

For years, discussions of digital preservation have routinely featured comments such as "bit preservation is a solved problem; the real issues are . . . ". Indeed, current digital storage technologies are not just astoundingly cheap and capacious, they are astonishingly reliable. Unfortunately, these attributes drive a kind of "Parkinson's Law" of storage, in which demands continually push beyond the capabilities of systems implementable at an affordable price.

This paper is in four parts:

- *Claims*, reviewing a typical claim of storage system reliability, showing that it provides no useful information for bit preservation purposes.
- *Theory*, proposing "bit half-life" as an initial measure of bit preservation performance, expressing bit preservation requirements in terms of it, and showing that the requirements being placed on bit preservation systems are so onerous that the experiments required to prove that a solution exists are not feasible.
- *Practice*, reviewing recent research into how well actual storage systems preserve bits, showing that they fail to meet the requirements by many orders of magnitude.
- *Policy*, suggesting ways of dealing with this unfortunate situation.

Introduction

For years, discussions of digital preservation have routinely featured comments such as "bit preservation is a solved problem; the real issues are ...".[1] Indeed, current digital storage technologies are not merely astoundingly cheap and capacious, they are astonishingly reliable. Unfortunately, these attributes drive a kind of "Parkinson's Law" (Parkinson 1957) of storage, in which demands continually push beyond the capabilities of systems implementable at an affordable price.

This paper is in four parts. The first part examines a typical claim made by a storage system vendor for the reliability of their product. It concludes that these numbers provide no useful information for bit preservation purposes.

The second, theoretical, part asks what characterizes a solution to the bit preservation problem adequate to the large numbers of bits to be stored and the long durations for which these bits are to be preserved. It proposes "bit half-life" as a metric for bit preservation, discusses the requirements being placed upon preservation systems in terms of this metric, and investigates the feasibility of benchmarking systems to see if they meet these requirements. It concludes that the requirements are so onerous that it is not feasible to measure whether systems meet them.

The third, practical, part reviews recent investigations into the performance of large-scale storage systems and their components. These studies uniformly report that storage reliability actually delivered to applications such as digital preservation systems is much less than that claimed by the manufacturers of systems and components. Tracking these failures to their root causes shows that every single hardware and software component contributes to some extent to the failures the systems experience. It concludes that current storage technologies fall well short of current requirements for bit preservation.

Given that the actual performance of storage systems is much worse than required, and that even if it improves we still won't be sure that a system will meet its requirements, the fourth part asks what is to be done. As with paper, content in digital archives will inevitably suffer loss and damage. The question is how to invest the limited funds available for preservation to the best effect in terms of improved data survival. There are many ways in which spending more money can reduce (but never completely eliminate) the probability of loss and damage. What is needed to allow informed investment decisions? How can we encourage the development of cost-effective techniques for long-term bit preservation?

Clarification

It is incumbent on those attacking ideas such as the "solvedness" of bit preservation to focus on the strongest version of the idea[2]. If proponents really believed that bit preservation was solved, they wouldn't bother with backups. Of course,

[1] The prevalence of this meme is aptly illustrated by the letter from the programme committee accepting this paper. It cites the title as "Bit Preservation - A Problem Solved".

[2] "we should always try to clarify and to strengthen our opponent's position as much as possible before criticising him" (Popper 1959)

they do. What they really mean by bit preservation being solved is that the set of techniques in common use make it so unlikely that bits will be lost that there is no need for concern at the prospect.

The techniques in which they place such faith are backups and checksums. Their real belief is that if they make a few backup copies of their content, and include in them checksums which they occasionally verify, their content will be safe. The goal of this paper is to show that, while backups and checksums may be adequate for relatively short periods and small amounts of preserved data, the scale and duration of current preservation tasks render them inadequate.

The state of our knowledge about preserving bits can be summarized as:

- *The more copies the safer.* As the size of the data increases, the per-copy cost increases, reducing the number of backup copies that can be afforded.
- *The more independent the copies the safer.* As the size of the data increases, there are fewer storage options available. Thus the number of copies in the same storage technology increases, decreasing the average level of independence.
- *The more frequently the copies are audited the safer.* As the size of the data increases, the time and cost needed for each audit increases, reducing their frequency.

Thus techniques that might be adequate at a small scale will break down as the scale increases.

Claims

How would we know if bit preservation were a solved problem? I suggest that proponents of this claim must feel confident that they could at a minimum preserve a petabyte of data undamaged for a century. Petabyte-scale data collections with long-term value, such as the Sloan Digital Sky Survey (SDSS 2008) and the Protein Data Bank (WWPDB 2008) already exist, so this is asking them to surmount a rather low bar. How confident should proponents feel in their ability to keep a petabyte for a century? I suggest that they should have at least a 50% chance of success. Again, this is a rather low bar.

Proponents might bolster their case that these bars can easily be surmounted by pointing to claims such as: "ST5800 has a MTTDL (Mean Time To Data Loss) of 2.4×10^6 years."[3] (Sun Microsystems 2008), or: "a Pergamum system capable of storing 10^{16} bytes of user data [will have] an MTTDL of 1.25×10^7 hours, or about 1,400 years." (Storer et al. 2008). These, and similar claims by other vendors, at first glance make it appear that bit preservation is indeed solved. Off-the-shelf solutions are ready to hand with performance so good that backups and checksums are quite superfluous. But do these claims stand up to examination?

Before using Sun's claim for its ST5800 as an example, I should stipulate that the ST5800 is an excellent product. It represents the state of the art in storage technology, and Sun's marketing claims represent the state of the art in storage marketing. Neverthless, Sun does not guarantee that data in the ST5800 will last 2.4×10^6 years. Sun's terms and conditions explicitly disclaim any liability whatsoever for loss of, or damage to, the data the ST5800 stores (Sun Microsystems 2006) whenever it occurs.

All that the claim says is that if you watched a large number of ST5800 systems for a long time, recorded the time at which each of them first suffered a data loss, and then averaged these times, the result would be 2.4×10^6 years. Suppose Sun watched 10 ST5800s and noticed that three of them lost data during the first year, four of them lost data after 2.4×10^6 years, and the remaining three lost data after 4.8×10^6 years, they would be correct that the MTTDL was 2.4×10^6 years. But we would not consider that a system with a 30% chance of data loss in the first year had solved the bit preservation problem. A single MTTDL number isn't a useful characterization of a solution.

Consider the slightly more scientific claim made at the recent launch of the SC5800 by the marketing department of Sirius Cybernetics[4]: "SC5800 has a MTTDL of $(2.4 \pm 0.4) \times 10^6$ years". Sirius thus claims that about $2/3$ of the failures occurred between 2.0×10^6 and 2.8×10^6 years after the start of the experiment. They didn't start watching 10 SC5800s 2.8 million years ago. So how would they know?

Perhaps, instead of watching say 10 systems for 2.4×10^6, years they watched more systems for a shorter time. Sirius says they will sell 2×10^4 SC5800s per year at $\$5 \times 10^4$ each (a billion-a-year business), and they expect the product to be in the market for 10 years. The SC5800 has a service life of 10 years. So if Sirius watched their entire production of SC5800s ($\$10^{10}$ worth of storage systems) over their entire service life the experiment would end 20 years from now after accumulating about 2×10^6 system-years of data. If their claim is correct they would have about a 17% chance of seeing a single data loss event.

In other words, Sirius Cybernetics claims that the probability that *no SC5800 will ever lose any data* is over 80%. Or, since each SC5800 stores 5×10^{13} bytes, that there is an 80% probability that 10^{19} bytes of data will survive 10 years undamaged.

If one could believe the Sirius Cybernetics claim, the petabyte would look pretty safe for a century. But the claim clearly isn't based on an experiment that won't provide results until 2028 and even when it does will not validate the number in question. In fact, numbers like these are not the result of experiment at all. No feasible experiment could validate them. They are *projections*, based on models of how components of the system such as disks and software behave.

The state of the art in this kind of modeling is exemplified by the Pergamum project at UC Santa Cruz (Storer et al. 2008). Their model includes disk failures at rates derived from (Schroeder and Gibson 2007; Pinheiro, Weber, and Barroso 2007) and sector failures at rates derived from

[3] Numbers are expressed in powers-of-ten notation to help readers focus on the scale of the problems and the extraordinary level of reliability required.

[4] Purveyors of chatty doors, existential elevators and paranoid androids to the nobility and gentry of this galaxy (Adams 1978).

disk vendor specifications. Their system attempts to conserve power by spinning the disks down whenever possible; they make an allowance for the effect of doing so on disk lifetime but it isn't clear upon what they base this. They report that the simulations were difficult:

> "This lack of data is due to the extremely high reliability of these configurations - the simulator modeled many failures, but so few caused data loss that the simulation ran very slowly. This behavior is precisely what we want from an archival storage system: it can gracefully handle many failure events without losing data. Even though we captured fewer data points for the triple inter-parity configuration, we believe the reported MTTDL is a reasonable approximation."

Although the Pergamum team's effort to obtain "a reasonable approximation" to the MTTDL of their system is praiseworthy, there are a number of reasons to believe that it overestimates the reliability of the system in practice:

- The model draws its failures from exponential distributions. They thus assume that both disk and sector failures are uncorrelated, although all measurements of actual failures (Bairavasundaram et al. 2008; Talagala 1999) report significant correlations. Correlated failures greatly increase the probability of data loss (Baker et al. 2006; Elerath and Pecht 2007).
- Other than a small reduction in disk lifetime from each power-on event, they assume that failure rates observed in always-on disk usage translate to their mostly-off environment. A study (Williams et al. 2008) published after their paper reports a quantitative accelerated life test of data retention in almost-always-off disks. It shows that the 3.5¨ disks anticipated by the Pergamum team have data life dramatically worse in this usage mode than 2.5¨ disks using the same underlying technology.
- They assume that disk and sector failures are the only failures contributing to the system failures, although a study (Krioukov et al. 2008) shows that other hardware components contribute significantly.
- They assume that their software is bug-free, despite several studies of file and storage implementations (Jiang et al. 2008; Engler 2007; Prabhakaran et al. 2005) that uniformly report finding bugs capable of causing data loss in all systems studied.
- They also ignore all other threats to stored data (Rosenthal et al. 2005) as possible causes of data loss. Among these are operator error, insider abuse and external attack. Each of these has been the subject of anecdotal reports of actual loss of preserved data.

What can models like this tell us? Their results depend on both:

- the details of the simulation of the system being studied which, one hopes, accurately reflect its behavior, and
- the data used to drive the simulation which, one hopes, accurately reflect the behavior of the system's components.

Under certain conditions, it is reasonable to use these models to compare different storage system technologies. The most important condition is that the models of the two systems use the same data. A claim that modeling showed system A to be more reliable than system B when the data used to model system A had much lower failure rates for components such as disk drives would not be credible.

These models may well be the best tools available to evaluate different techniques for preventing data loss, but they aren't adequate to determine whether bit preservation is a solved problem. We need to know the *maximum* rate at which data will be lost. The models assume things, such as uncorrelated errors and bug-free software, that all experimental studies show are false. The models exclude most of the threats to which stored data is subject. And in those cases where similar claims, such as those for disk reliability (Schroeder and Gibson 2007; Pinheiro, Weber, and Barroso 2007), have been tested they have been shown to be optimistic. It is not reasonable to assume that these factors are negligible, nor that they affect all systems equally; the models thus provide an estimate of the *minimum* data loss rate to be expected.

Even if we believed the models, the MTTDL number doesn't tell us how much data was lost in the average data loss event. Is petabyte system A with a MTTDL of 10^6 years better than a similar size system B with a MTTDL of 10^3 years? If the average data loss event in system A loses the entire petabyte, where the average data loss event in system B loses a kilobyte, it would be easy to argue that system B was 10^9 times better.

It is clear that we need a better way to define and measure bit preservation performance. Mean time to data loss is not a useful characterization of how well a system stores bits through time.

Theory

In order to claim that "bit preservation is a solved problem" we would need three things we currently don't have:

- A specific requirement as to how well bits need to be preserved.
- A technique for measuring whether actual systems achieve the required level of bit preservation.
- Measurements of an actual system using the technique that confirm it meets or exceeds the requirement.

In this section we suggest a metric that would be more useful than MTTDL, and ask whether it is possible to characterize actual systems in terms of this metric.

Defining a Solution

The most abstract model of a bit preservation system is as a black box, into which a string of bits $S(0)$ is placed at time $T(0)$ and from which at subsequent times $T(i)$ a string of bits $S(i)$ can be extracted. The system is successful if $S(i) = S(0)$ for all i.

No real-world system can be perfect and eternal, so real systems will fail. The simplest model of these failures is analogous to the decay of radioactive atoms. Each bit in the string independently is subject to a random process that has a constant small probability per unit time of causing its value to flip. The time after which there is a 50% probability that a bit will have flipped is the "bit half-life".

The requirement of a 50% chance that a petabyte will survive for a century translates into a bit half-life of 8×10^{17} years. The current estimate of the age of the universe U is 1.4×10^{10} years, so this is a bit half-life approximately $6 \times 10^{7}U$.

Measuring a Solution

Because current storage systems are extraordinarily reliable, measuring their bit half life involves observing very large numbers of bits for a very long time. If you wanted to take a year to measure whether a system met the petabyte-for-a-century requirement you might watch a thousand such systems, an exabyte of data. If the system were just good enough, you would see a single bit flip in just five of the systems.

Even if one were able to afford this experiment, doing so would be challenging. Data must be read from the system and compared with its expected value. Even if each bit is checked only once at the end of the year, the comparisons have to be performed with less than 1 chance in 10^{19} of any error.

In practice, estimates of bit half-life would have to be based upon the same models as estimates of MTTDL, and would thus share many of the same difficulties.

Assessment

There is no escape from the problem that the size of the data collections to be preserved and the times for which they must be preserved mean that experimental confirmation that the technology chosen is up to the job is not economically feasible. Even if it was the results would not be available soon enough to be useful. What this argument demonstrates is that, far from bit preservation being a solved problem, it is in a very specific sense an *unsolvable* problem. Even if we believed a system we developed was reliable enough, there are no feasible experiments that could confirm our belief in time to be useful.

Bit half-life is a more informative metric than MTTDL, because it is a measure of the reliability of the *data*, not a measure of the reliability of the *system* storing it. The data's survival is what we care about. It thus captures the fact that the impact of a data loss event depends not just on when it happens, but also on how much data is lost. It is still far from ideal:

- Bits in real storage systems do not fail independently; they exhibit significant correlations in space and time (Bairavasundaram et al. 2008). These correlations make failure more likely than it otherwise would be. This observation doesn't invalidate the simple "radioactive decay" model; it merely makes adequate bit half-life a necessary but not sufficient condition for a system to meet the requirement.
- Like MTTDL, it is a statistical estimate and thus, like MTTDL, it is not useful without an uncertainty interval.
- Because storage systems are so reliable, it is just as difficult to measure bit half-life as it is to measure MTTDL.

Practice

As enterprises such as Google (Chang et al. 2006) and institutions such the Sloan Digital Sky Survey (SDSS 2008) and the Large Hadron Collider (CERN 2008) collect petabytes of data with long-term value that must remain on-line to be useful, and as the annual cost of keeping a petabyte on-line is more than a million dollars (Moore et al. 2007), questions of the economics and reliability of storage systems have become the focus of researchers' attention.

Storage Failures

Papers at the 2007 FAST conference used data from NetApp (Schroeder and Gibson 2007) and Google (Pinheiro, Weber, and Barroso 2007) to study disk replacement rates in large storage farms. They showed that the manufacturer's MTTF numbers were optimistic. Subsequent analysis of the NetApp data (Jiang et al. 2008) showed that all other components contributed to the storage system failures, and:

> 'Interestingly, [the earlier studies] found disks are replaced much more frequently (2–4 times) than vendor-specified [replacement rates]. But as this study indicates, there are other storage subsystem failures besides disk failures that are treated as disk faults and lead to unnecessary disk replacements."

Two studies, one at CERN (Kelemen 2007) and one using data from NetApp (Bairavasundaram et al. 2008), greatly improved on earlier work using data from the Internet Archive (Baker et al. 2006; Schwarz et al. 2006). They studied *silent data corruption* in state-of-the-art storage systems; events in which the content of a file in storage changes with no explanation or recorded errors.

The NetApp study looked at the incidence of silent storage corruption in individual disks in RAID arrays. The data was collected over 41 months from NetApp's filers in the field, covering over 1.5×10^{6} drives. They found over 4×10^{5} silent corruption incidents. More than 3×10^{4} of them were not detected until RAID restoration and could thus have caused data loss despite the replication and auditing provided by NetApp's row-diagonal parity RAID (Corbett et al. 2004).

The CERN study used a program that wrote large files into CERN's various data stores, which represent a broad range of state-of-the-art enterprise storage systems (mostly RAID arrays), and checked them over a period of 6 months. A total of about 9.7×10^{16} bytes was written and about 1.92×10^{8} bytes was found to have suffered silent corruption, of which about 2/3 was persistent; re-reading did not return good data. In other words, about 1.2×10^{-9} of the data written to CERN's storage was permanently corrupted within six months. We can place an upper bound on the bit half-life in this sample of current storage systems by assuming that the data was written instantly at the start of the 6 months and checked instantly at the end; the result is 2×10^{8} or about $10^{-2}U$. Thus to reach the petabyte for a century requirement we would need to improve the performance of current enterprise storage systems by a factor of at least 10^9.

Surviving Storage Failures

Despite the manufacturer's claims, current research shows that state-of-the-art storage systems fall so many orders of magnitude below our bit preservation requirements that we cannot expect even dramatic improvements in technology to fill the gap. Maintaining a single replica in a single storage system is not an adequate solution to the bit preservation problem.

Practical digital preservation systems must therefore:

- Maintain more than one copy by *replicating* their data on multiple, ideally different, storage systems.
- Audit or (*scrub*) the replicas to detect damage, and repair it by overwriting the known-bad copy with data from another.

The more replicas and the more frequently they are audited and repaired the longer the bit half-life we can expect. This is, after all, the basis for the backups and checksums technique in common use. In fact, current storage systems already use versions of these techniques, for example in the form of RAID (Patterson, Gibson, and Katz 1988). Despite this the bit half-life they deliver is inadequate. Unfortunately adding the necessary inter-storage-system replication and scrubbing is expensive.

2007 cost figures from the San Diego Supercomputer Center (Moore et al. 2007) show that maintaining a single on-line copy of a petabyte for a year then cost about $\$1.5 \times 10^6$. A single near-line copy on tape cost about $\$5 \times 10^5$ a year[5]. These costs decrease with time, albeit not as fast as raw disk costs. The British Library estimates a 30% per annum decrease. Assuming that this rate continues for at least a decade, if you can afford about 3.3 times the first year's cost to store an extra replica for a decade, you can afford to store it indefinitely. So, adding a second replica of a petabyte on disk would cost about $\$3.5 \times 10^6$ and on tape would cost about $\$1.4 \times 10^6$. Adding cost to a preservation effort to increase reliability in this way is a two-edged sword; doing so necessarily increases the risk that preservation will fail for economic reasons.

Further, without detailed understanding of the rates at which different mechanisms cause loss and damage, it isn't possible to derive from a desired bit half-life the appropriate number of replicas[6] and thus the cost implication of replication. At small scales the response to this uncertainty is to add more replicas, but as the scale increases this rapidly becomes unaffordable.

Replicating among identical systems is much less effective than replicating among diverse systems. Identical systems are subject to common mode failures, for example caused by a software bug in all the systems damaging the same data in each. On the other hand, purchasing and operating a number of identical systems will be considerably cheaper than operating a set of diverse systems.

Each replica is vulnerable to loss and damage. Unless they are regularly audited they contribute little to increasing bit half-life. The bandwidth and processing capacity needed to scrub the data are both costly, and adding these costs increases the risk of failure. Custom hardware (Michail et al. 2005) could compute the SHA-1 (Nat 1995) checksum of a petabyte of data in a month, but doing so requires impressive bandwith - the equivalent of three gigabit Ethernet interfaces running at full speed the entire month. User access to data in preservation systems is typically infrequent; they are therefore rarely architected to provide such high-bandwidth read access. System cost increases rapidly with I/O bandwidth, and the additional accesses to the data (whether on disk or on tape) needed for scrubbing themselves potentially increase the risk of failure.

The point of writing software that reads and verifies stored data in this way is to detect damage and exploit replication to repair it, thereby increasing bit half-life. How well can we do this? RAID is an example of a software technique of this type applied to disks. In practice, the CERN study (Kelemen 2007) looking at real RAID systems from the outside showed a significant rate of silent data corruption, and the NetApp study (Bairavasundaram et al. 2008) looking at them from the inside showed a significant rate of silent disk errors that would lead to silent data corruption. A study (Krioukov et al. 2008) of the full range of current algorithms used to implement RAID found flaws leading to potential data loss in all of them. Both this study, and another from IBM (Hafner et al. 2008), propose improvements to these algorithms but neither claim that they can eliminate silent corruption, or even accurately predict its incidence:

> "while we attempt to use as realistic probability numbers as possible, the goal is not to provide precise data loss probabilities, but to illustrate the advantage of using a model checker, and discuss potential trade-offs between different protection schemes." (Krioukov et al. 2008)

Thus although replication and scrubbing are capable of decreasing the incidence of data loss in current storage systems, they cannot eliminate it completely. And the replication and scrubbing software itself will contain bugs that can cause data loss. It must be doubtful that we can implement these techniques well enough to increase the bit half-life of systems with an affordable number of replicas by 10^9.

It takes experiments with petabytes of storage to characterize the performance of current systems accurately. Even if we believed we had implemented replication and audit well emough to improve performance by 10^9, we could not afford to do the experiments that would be needed to confirm it.

[5] SDSC reports that the 2008 costs are $\$1.05 \times 10^6$ and $\$4.2 \times 10^5$

[6] The number can be quite large; a study of paper journals (Yano 2008) found between 3 and 31 copies were needed to achieve loss probabilities over a century of between 10^{-3} and 10^{-6} given various plausible loss rates of the individual copies. The lower repairability of paper copies inflates these numbers, while their greater durability deflates them, as against digital copies.

Policy

If bit preservation were a solved problem then it would be reasonable to expect that no bits would be lost. This is not the case; just as in paper archives preserved content in digital archives will be lost or damaged. Setting unreasonable expectations for the performance of our preservation systems, for example by continually making unsupported claims to have solved the bit preservation problem, is simply setting ourselves up to be perceived as failures.

If preserved bits will be lost, the question becomes how to invest the limited funds available to reduce the rate of loss as much as possible. It is a commonplace that if you can measure something you can improve it. The history of technology markets such as CPUs and graphics chips show that competition between vendors based on widely accepted standard benchmarks can drive rapid improvements in component cost-performance. Alas, although raw storage cost is easily measured and is the subject of effective competition to decrease cost per byte (Christensen 1997), long-term storage reliability is very hard to measure and the accepted metric for it is not very informative. Competition to reduce the cost of a given level of bit preservation is therefore much less effective.

It is in the interest of the digital preservation community to improve competition in their market. How could this be done?

- Agreement on a metric for bit preservation performance is an essential first step. It would be extremely valuable if it were possible to define one that was easily measureable, but this seems rather unlikely.
- Given this, it seems likely that numbers for bit preservation performance will continue to be generated by models. Achieving consensus on modeling techniques is important, especially as it appears that traditional techniques are running into difficulties (Storer et al. 2008; Elerath and Pecht 2007).
- These models will need agreed data. Better and more widely available data about the real world performance storage components is thus important. Realistic studies have only begun to be published, and they aren't yet based on shared metrics. The effort by Usenix and Carnegie-Mellon (Usenix 2008) to establish a repository for suitably anonymized data of this kind is to be commended.
- Storage systems are currently designed using completely inadequate models of how components fail. One problem is that these failures are highly correlated, making the models complex and difficult. A shared model of the threats against which bits need to be preserved, models of these threats, and data regarding their incidence is also important.
- Anecdotal evidence suggests that operator error and insider abuse are major causes of data loss in large storage farms; they are difficult to model or characterize. This is in part because sites are very reluctant to admit to data loss incidents. An anonymous incident reporting system modelled on NASA's Aviation Safety Reporting System (NASA 2008) would be very valuable in understanding the mechanisms of, and defending against, these failures.

The fact that it is possible for digital information to be copied perfectly does not mean that it always will be. While perfection is not within the grasp of real-world engineers, improvement is always possible. However, improvement takes money, and without the research outlined above we are unable to make rational tradeoffs between the cost of preserving content to a given level of reliability and the cost of the losses implied by the given level.

Conclusions

As we have seen, the case that bit preservation is a solved problem rests on the conviction that the conventional techniques of backups and checksums are more than adequate to the scale of the problem. This conviction is odd. Press accounts (e.g. (Brodkin 2008)) of companies, presumably using the conventional techniques, nevertheless losing essential data are common. Awareness that systems frequently encounter scaling problems is also widespread, as is the expectation that the future demands for preserving digital content will be enormous.

But the case for bit preservation not being solved does not rest on this cognitive dissonance. It rests rather on the many orders of magnitude mismatch between the reliability requirements implied by society's expectations of the amount of data to be preserved and the length of time for which it should be preserved, and the observed performance of current storage hardware and software.

Were every bit to come adequately endowed with capital to provide guaranteed funds through time its preservation would not be a major concern, although it would still not be a solved problem. Like almost all engineering problems, bit preservation is fundamentally a question of budgets. Society's ever-increasing demands for vast amounts of data to be kept for the future are not matched by suitably lavish funds. Thus, absent a technological miracle, bit preservation is a problem with which we are doomed to struggle indefinitely.

Acknowledgements

Thanks are due to Michael Bax and the LOCKSS engineering team for critical readings of drafts of this paper, and to the staff of the San Diego Supercomputer Center for the discussions that started me thinking along these lines.

References

Adams, D. 1978. The Hitch-Hiker's Guide to the Galaxy. British Broadcasting Corp.

Bairavasundaram, L.; Goodson, G.; Schroeder, B.; Arpaci-Dusseau, A. C.; and Arpaci-Dusseau, R. H. 2008. An Analysis of Data Corruption in the Storage Stack. In *Proceedings of 6th USENIX Conf. on File and Storage Technologies.*

Baker, M.; Shah, M.; Rosenthal, D. S. H.; Roussopoulos, M.; Maniatis, P.; Giuli, T.; and Bungale, P. 2006. A Fresh

Look at the Reliability of Long-term Digital Storage. In *Proceedings of EuroSys2006*.

Brodkin, J. 2008. Loss of customer data spurs closure of online storage service 'The Linkup'. *Network World*. 11^{th} Aug.

CERN. 2008. Worldwide LHC Computing Grid. `http://lcg.web.cern.ch/LCG/`.

Chang, F.; Dean, J.; Ghemawat, S.; Hsieh, W. C.; Wallach, D. A.; Burrows, M.; Chandra, T.; Fikes, A.; and Grube, R. E. 2006. Bigtable: A Distributed Storage System for Structured Data. In *Proceedings of the 7th Usenix Symp. on Operating System Design and Implementation*, 205–218.

Christensen, C. M. 1997. *The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail*. Harvard Business School Press.

Corbett, P.; English, B.; Goel, A.; Grcanac, T.; Kleiman, S.; Leong, J.; and Sankar, S. 2004. Row-Diagonal Parity for Double Disk Failure Correction. In *3rd Usenix Conference on File and Storage Technologies*.

Elerath, J. G., and Pecht, M. 2007. Enhanced reliability modeling of raid storage systems. In *DSN '07: Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 175–184. Washington, DC, USA: IEEE Computer Society.

Engler, D. 2007. A System's Hackers Crash Course: Techniques that Find Lots of Bugs in Real (Storage) System Code. In *Proceedings of 5th USENIX Conf. on File and Storage Technologies*.

Hafner, J. L.; Deenadhayalan, V.; Belluomini, W.; and Rao, K. 2008. Undetected disk errors in RAID arrays. *IBM J. Research & Development* 52(4/5).

Jiang, W.; Hu, C.; Zhou, Y.; and Kanevsky, A. 2008. Are Disks the Dominant Contributor for Storage Failures? A Comprehensive Study of Storage Subsystem Failure Characteristics. In *Proceedings of 6th USENIX Conf. on File and Storage Technologies*.

Kelemen, P. 2007. Silent Corruptions. In *8th Annual Workshop on Linux Clusters for Super Computing*.

Krioukov, A.; Bairavasundaram, L. N.; Goodson, G. R.; Srinivasan, K.; Thelen, R.; Arpaci-Dusseau, A. C.; and Arpaci-Dusseau, R. H. 2008. Parity Lost and Parity Regained. In *Proceedings of 6th USENIX Conf. on File and Storage Technologies*.

Michail, H. E.; Kakarountas, A. P.; Theodoridis, G.; and Goutis, C. E. 2005. A low-power and high-throughput implementation of the SHA-1 hash function. In *Proceedings of the 9th WSEAS International Conference on Computers*.

Moore, R. L.; D'Aoust, J.; McDonald, R. H.; and Minor, D. 2007. Disk and Tape Storage Cost Models. In *Archiving 2007*.

NASA. 2008. Aviation Safety Reporting System. `http://asrs.arc.nasa.gov/`.

National Institute of Standards and Technology (NIST), Washington, D.C., USA. 1995. *Federal Information Processing Standard Publication 180-1: Secure Hash Standard (SHA-1)*.

Parkinson, C. N. 1957. *Parkinson's Law*. Buccaneer Books.

Patterson, D. A.; Gibson, G.; and Katz, R. H. 1988. A Case for Redundant Arrays of Inexpensive Disks (RAID). In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 109–116.

Pinheiro, E.; Weber, W.-D.; and Barroso, L. A. 2007. Failure Trends in a Large Disk Drive Population. In *Proceedings of 5th USENIX Conf. on File and Storage Technologies*.

Popper, K. 1959. *Logic of Scientific Discovery*. Hutchinson. Footnote *5, Chapter X.

Prabhakaran, V.; Agrawal, N.; Bairavasundaram, L.; Gunawi, H.; Arpaci-Dusseau, A. C.; and Arpaci-Dusseau, R. H. 2005. IRON File Systems. In *Proceedings of the 20th Symposium on Operating Systems Principles*.

Rosenthal, D. S. H.; Robertson, T. S.; Lipkis, T.; Reich, V.; and Morabito, S. 2005. Requirements for digital preservation systems: A bottom-up approach. *D-Lib Magazine* 11(11).

Schroeder, B., and Gibson, G. 2007. Disk failures in the real world: What Does an MTTF of 1,000,000 Hours Mean to You? In *Proceedings of 5th USENIX Conf. on File and Storage Technologies*.

Schwarz, T.; Baker, M.; Bassi, S.; Baumgart, B.; Flagg, W.; van Imngen, C.; Joste, K.; Manasse, M.; and Shah, M. 2006. Disk Failure Investigations at the Internet Archive. In *Work-in-Progress Session, NASA/IEEE Conf. on Mass Storage Systems and Technologies*.

SDSS. 2008. The Sloan Digital Sky Survey. `http://www.sdss.org/`.

Storer, M. W.; Greenan, K. M.; Miller, E. L.; and Voruganti, K. 2008. Pergamum: Replacing Tape with Energy Efficient, Reliable, Disk-Based Archival Storage. In *Proceedings of 6th USENIX Conf. on File and Storage Technologies*.

Sun Microsystems. 2006. Sales Terms and Conditions, Section 11.2. `http://store.sun.com/CMTemplate/docs/legal_terms/TnC.jsp#11`.

Sun Microsystems. 2008. ST5800 presentation. Sun PASIG Meeting.

Talagala, N. 1999. *Characterizing Large Storage Systems: Error Behavior and Performance Benchmarks*. Ph.D. Dissertation, CS Div., Univ. of California at Berkeley, Berkeley, CA, USA.

Usenix. 2008. The computer failure data repository (CFDR). `http://cfdr.usenix.org/`.

Williams, P.; Rosenthal, D. S. H.; Roussopoulos, M.; and Georgis, S. 2008. Predicting the Archival Life of Removable Hard Disk Drives. In *Archiving 2008*.

WWPDB. 2008. Worldwide Protein Data Bank. `http://www.wwpdb.org/`.

Yano, C. 2008. How Many Journal Copies? A Preliminary Report. Presentation to ALA.