Using Storage Class Memory for Archives with DAWN, a Durable Array of Wimpy Nodes

Technical Report UCSC-SSRC-11-07
October 2011

Ian F. Adams
iadams@soe.ucsc.edu

Ethan L. Miller
elm@soe.ucsc.edu

David S.H. Rosenthal
dshr@standford.edu

Baskin School of Engineering
University of California, Santa Cruz
Santa Cruz, CA 95064
http://www.ssrc.ucsc.edu/

Using Storage Class Memory for Archives with DAWN, a Durable Array of Wimpy Nodes

Ian F. Adams
University of California, Santa Cruz

Ethan L. Miller
University of California, Santa Cruz

David S. H. Rosenthal
Stanford University

Abstract

The long life and low usage of archival data make cost considerations paramount. Today, most archival storage architectures depend on magnetic or optical media such as tape and disk because they have a low initial cost per byte. The high initial cost of storage class memories (SCMs) has been seen as prohibitive for archival use.

Nevertheless, SCMs have many advantages for archival use, including physical robustness and low power usage. In this work, we argue that a suitable architecture can exploit these advantages to make SCM competitive with magnetic media for archival use. Inspired by the FAWN and Pergamum systems, we outline the design of DAWN, a Durable Array of Wimpy Nodes. DAWN will make use of low-power system-on-chip technology paired with SCM to provide a simple, reliable, self-managing archival storage system with a low total cost of operation.

1 Introduction

Long-term data storage is challenging from both engineering and business perspectives. In engineering terms, traditional well-understood enterprise storage techniques are inappropriate [6] because they tend to have a finite life and high requirements for availability and performance. In contrast, archival systems must be highly reliable, but may have relaxed availability and performance requirements due to low read rates and a focus on long-term reliability over short-term availability. They must also be massively scalable from both very small to very large sizes, all while maintaining a very low total cost of operations. The difficulty of justifying periodic funding for storing data that *might* be useful in the future makes the latter point particularly important—it is often difficult to convince a funding source that occasionally-used data will continue to be important. Endowment approaches to funding have their own set of issues to contend with as well. Managing a one-time endowment over long periods can be difficult in the face of unpredictable short term fluctuations in media, maintenance, infrastructure and long term running costs, let alone fluctuating investment interest rates.

The current use of magnetic (and, to a lesser extent, optical) media for long-term storage is inefficient at best, and dangerous at worst from both technical and cost standpoints. The complex mechanical components of both tape and hard disk [10] render them vulnerable to a wide range of failure modes. Disks require extensive, and expensive, power and cooling infrastructure. Tapes consume zero power while off-line, but require expensive drives, silos and robotics to operate at scale, and suffer from extremely poor random access performance.

Storage class memories (SCMs) have been suggested as an alternative medium for archival storage [21]. At first glance, their much higher capital cost—flash memory is currently about 15 to 20 times more per byte than consumer hard disk—seems prohibitive. However, they have many appealing traits, such as significantly lower running cost and physical robustness, that can compensate for this. To take advantage of these traits we propose DAWN, a Durable Array of Wimpy Nodes. Using techniques borrowed from FAWN [3] and Pergamum [28], we outline the design of a highly reliable, self-managing archive using SCM. Given the physical robustness and long usable lifetimes of SCMs under archival workloads, our analysis suggests that the DAWN approach can already be competitive with disk for long-term archiving.

2 Background

Here we outline the unique characteristics of archival storage, discuss why current tape and disk approaches are less appropriate for long-term data storage, and highlight the advantages of SCMs for archival roles.

2.1 Archival Storage Requirements

Archival workloads differ significantly from that of traditional enterprise storage. First and foremost, most data is rarely, if ever, deleted or overwritten, but *does* need to be protected and accessible indefinitely. Second, the vast majority of traffic typically comes from maintenance processes, such as integrity checking and replica management rather than end-users [1]. Third, end-user data reads are difficult to predict [1, 6], with only modest popularity hotspots. Fourth, because archival data is typically data that *may* be useful in the future, extra emphasis is given on minimizing the cost per byte of storage over time.

Obtaining funding for archival storage is particularly difficult: organizations must consider how to obtain sufficient funding to run a system indefinitely. This includes not only paying for the media itself, but also the long-term management and maintenance of the system and associated infrastructure, which has been found to comprise nearly two-thirds of the total cost of long-term data storage [18].

The two approaches to paying for archival storage are periodic funding and one-time endowments. Annual or monthly funding, while adjustable to actual costs, is also subject to short term income cuts that can be exacerbated by the volatility of storage and infrastructure costs. These risks are compounded by the difficulty of convincing others at regular intervals to continue paying for rarely accessed data. An alternative is the endowment model, or *pay once, store endlessly* discussed by Goldstein [12]. This approach is based on the assumption that, over time, the per-unit of storage costs of procuring and maintaining storage will continue to drop. If this assumption holds, the total cost for maintaining the endowed data forever converges. Though not subject to the volatility of an annual funding model, it still is vulnerable to short-term price variations and forecast failures. For example, if storage costs do not decrease as quickly as predicted, or per-unit maintenance costs do not drop with media costs, the endowment can quickly run short of funds.

These characteristics—high reliability, very low cost of operation, relaxed availability and performance, low read rates—lead to a system with unique requirements at odds with enterprise storage. The system must have very low operating costs, with low power and infrastructure requirements and minimal human administration, since these items are responsible for the majority of the cost of ownership. In addition, the system must be well-protected against operator error, such as accidental data overwrites and deletions. An ideal system is one that autonomously manages replication and integrity checking, consumes little power, and has a long device life. A long device life is particularly important, since it amortizes procurement costs over a longer period of time, lowers the rate at which media must be replaced and data migrated, and reduces the economic, and thus funding, risks of inaccurate cost forecasts and short-term price volatility.

2.2 Why Not Disk or Tape?

Tape and disk have lower per-bit capital cost, but their complex mechanical components render them vulnerable to a wide range of failure modes [10]. Further, the devices themselves are only getting more complex as efforts to increase areal density demand new techniques. For example, disk manufacturers are now looking towards techniques such as heat assisted magnetic recording and shingled writes [2] to increase areal density. Moreover,the power and cooling for hard disks can account for 30% of a data center's power consumption [20], making them one of the primary factors in a data center's running costs.

Tape fares little better. On the one hand, tape is cheap to procure, and is relatively long lived, with manufacturer-stated shelf lives of up to 30 years [8]. However, tape requires significant infrastructure and maintenance, such as periodic re-tensioning, to remain readable, and, though the tapes themselves need no power, the robots and drives in a typical silo are both expensive to obtain and power-hungry in their own right. Moreover, while an individual tape may survive 30 or more years under optimum conditions, the drives themselves may be long gone due to mechanical failures and hardware evolution [14]. In the end, both tape and disk approaches are expensive and complicated to scale and maintain for long term storage.

2.3 Why Use SCM?

Though storage class memories have higher capital cost than magnetic media, they have many characteristics that make them ideal for long-term data storage.

Power and Infrastructure Needs. As data centers become increasingly power hungry [4], with storage accounting for up to 30% of power consumption [20], even modest energy savings are sought after. Most SCMs consume little or no power while idle, and even under heavy load are remarkably power efficient [3, 16]. SCMs generate less heat and can also tolerate higher temperatures, greatly reducing the need for cooling. SCM-based devices, with no moving parts, generate little or no vibration, and can tolerate significant vibration as well. Combined with low cooling requirements, this means they can be packed closely together, reducing the need for data center space. Together, these significantly reduced both capital expenditure and running costs.

Toughness. In a suitable enclosure SCMs, have many fewer failure modes than hard disk or tape. They are much less vulnerable to shock, and repeated power cycles do not cause the mechanical failures common with disks, tape drives and robots. SCMs are also proving to be remarkably resilient to a wide range of temperatures. Desnoyers

	Tape Approaches	Disk Approaches	SCM Approaches
Operating Vibration	Yes	Yes	No
Vibration tolerance	Moderate/High	Low	High
Power Needs per Unit	Low (system dep.)	Moderate	Low
Heat Tolerance	Low/Moderate	Low/Moderate	Moderate/High
Weight/Unit Storage	Low/Moderate	Moderate	Low/Moderate
Random Access	Poor	Good	Excellent
Easy Integrity Check	No	Yes	Yes
Infrastructure Reqs.	High at small scale	Low/Moderate	Low/Moderate
Shock Resistance	Low/Moderate	Low/Moderate	High
Acquisition Cost/Byte	Very Low	Low	High
Est. Device Lifetime	Very Long (30+ yrs)	Moderate (5-10 yrs)	Moderate/Long (10+ yrs)

Figure 1: A qualitative comparison of tape, disk and SCM media. The darker a cell is, the worse that type of media is for that particular characteristic (row).

found that read and write endurance in flash was not impacted by temperatures as high as 80°C [9], and phase change memory has shown good data retention at high temperatures in laboratories [31]. Media for long term archiving should have minimal demands on the infrastructure and survive benign neglect, so these characteristics are valuable. Furthermore, solid state devices may have a lower likelihood of total device failure due to their lack of mechanical components, but have a correspondingly higher bit-error rate than disk based approaches [17]. This means that the volume of data lost at any one time will be lower, but extra care should be taken to compensate for the higher bit error rates. We address this in our architecture proposal in Section 3.

Data Retention Times. While some models of tape and disk can potentially retain data for upwards of 30 years [8, 33], there are often caveats: limits on the number of power-ups, or periodic physical maintenance and protection from dust. While current estimated data retention times for SCMs are shorter, they are still more than adequate for most archival scenarios. For example, there are models of flash with estimated data retention times of 10 to 40 years [24, 30] provided there are not large numbers of write cycles—a safe assumption in archival storage. With a periodic data refresh cycle, the safe data storage lifetime can be extended well beyond even that, provided the device itself does not physically break. Memristors are early in their development, but are also estimated to have data retention lifespans of many years [15]. Phase change memory is estimated to have a data retention time of at least 10 years, even under high temperatures [31].

Fast Random Access. Access performance is not considered critical for archival data, but has advantages nevertheless. Both tape and spun-down disks incur significant latency and mechanical wear to spin up. Since no particular datum is likely to be more popular than any other, caches can not avoid these latencies. Additionally, fast access and high bandwidth reduce rebuild and replication times in the event of a full device failure, minimizing the window of vulnerability for data loss. Finally, fast access is critical to the perception of an effective digital archive, especially in internet-based systems.

3 Flash Based Archive

This section outlines our vision hardware and software for a self-managing SCM based archive and provides cost-benefit analyses of our proposed system.

3.1 Hardware Vision

Our hardware vision, which we call DAWN (**D**urable **A**rray of **W**impy **N**odes), draws heavily from FAWN [3] and Pergamum [28]. Each node is a self-contained unit with a single power-over-Ethernet connector. It contains SCM, a system-on-chip with RAM, an Ethernet interface and a low-power processor, illustrated in Figure 2. We focus here on the use of flash as our SCM as it is the most mature and best-characterized SCM technology.

By using low-power, fully self-contained field-replaceable units, we simplify management and infrastructure needs. There are no complicated servers to maintain. If a unit fails, it may be discarded and replaced in its entirety. The external interfaces of these nodes are Ethernet and standard network protocols, the most stable interfaces available. Because the nodes combine both storage and a low-power processor, they can manage themselves, performing local integrity checks, and coordinate with others in the network to maintain the desired level of data replication, and to repair any loss or damage that is detected.

Each unit consumes very little power, even when fully

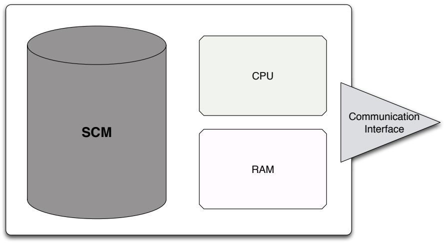

Figure 2: Overview of the DAWN hardware vision. A realized system would likely use an Ethernet interface, but wireless or other yet unrealized technologies are not ruled out.

active. Consider a unit using a present day ARM-based processor with embedded RAM and attached flash. Under the worst-case power consumption, using high-end components which would *not* be used in a realized DAWN system, a fully powered unit under load would still consume no more than 7 watts: 4 W for the computing board and RAM [29] and another 3 W for the flash [22]. In sleep mode, this unit would consume significantly less than a watt. Given these very low power requirements, power-over-Ethernet or even USB could be considered for a power supply.

For added durability the entire unit could be coated in epoxy, with only the network/power connector exposed. A battery back-up may be added to allow for a limited amount of self checking to be completed, even if a device is physically disconnected from the network and power supply.

3.2 Software Vision

Our system is designed to be self-managing and resilient to attack and operator error. Its primary operations are data ingest, reads, and intra- and inter-device reliability management.

Data Ingest and Reads. To protect against malicious or accidental data loss, all writes to DAWN are version-on-write. Deletes are not allowed; any attempt at an overwrite simply appends a new version of the data. Any device can be read directly through its Ethernet interface. A centralized index could be used for simple, fast data lookup, or the nodes may themselves be queried for data. An additional benefit of per-node data lookups is scalable recovery and resilience. A centralized index is vulnerable to loss as a single point of failure, and can be slow and difficult to recover in large systems. By leveraging the low-power CPUs to do per-node indexing in DAWN nodes, we can either completely remove the centralized index, or use the nodes themselves to aid in reconstruction of the primary index.

Intra-Device Reliability. Using flash as an archival medium introduces certain subtleties. The most commonly discussed is write endurance. Even assuming data is overwritten daily, it would take over 25 years for a conservative write endurance of 10,000 cycles to be exceeded [9]. Of greater concern are the issues of *read disturb* and *data retention*. In flash, repeated reads to a given page may disturb surrounding data, corrupting it. Data retention is the ability of a given cell to retain its charge, and therefore data, over extended periods of time. While storage cells inevitably leak charge over time and must be periodically refreshed, this leakage is fortunately quite low; estimates of data retention in flash—with no reads or writes—range from 10 years to as long as 40 years [24, 30]. If a device is plugged in, it is a simple matter to periodically refresh the data, and if it is disconnected (and thus idle), we need not worry about the impact of reads and writes.

Given these issues, we create high intra-device reliability with two methods. First, we use additional ECCs beyond those provided by the SCM itself. Our approach follows that Kang and Miller proposed, using Reed-Solomon codes to provide extra protection in the event of block failures [13], illustrated in Figure 3. Second, we use a more proactive scrubbing methodology. Scrubbing verifies data by hashing it and comparing the result to a previously-computed digest [23]. We propose to address the issues of data retention and read disturbs by piggy-backing a *data refresh* cycle on top of the periodic scrubbing processes, re-writing the verified data. Though this increases the physical wear on the device, scrubbing is a sufficiently infrequent process that it will not significantly shorten the life of a device. However, there is an inverse relationship between the number of write cycles on a device and its data retention time [25] that must be accounted for, and currently SSD characterization is insufficient for us to accurately estimate its impact, as we discuss further in Section 4. Along another dimension, effective intra-device reliability methods are particularly important because, as the density of flash increases, the data retention and reliability of individual devices correspondingly drops due to increased cell error rates, though internal ECCs also help mitigate this [19]. Note that disks are not free from density-vs-reliability issues either; the fraction of disks impacted by latent sector errors is shown to be increasing with disk capacity [5].

Inter-Device Reliability. To withstand whole device failures, we take a distributed RAID approach similar to that proposed in Pergamum [28]. Leveraging the self-monitoring ability of the nodes, if a device is removed or fails, automatic rebuild can occur to available space elsewhere in the system without any human intervention.

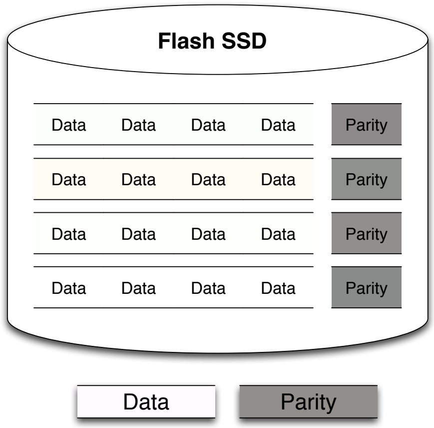

Figure 3: A visualization of a DAWN node's intra-device reliability schemes. We propose adding in additional error-correcting codes to compensate for higher bit-error rates

TB Stored	Per-TB Yearly Cost
1 TB	$1892
50 TB	$1711
500 TB	$1542
1000 TB	$1411
5000 TB date	$1222
7500 TB	$1101

Table 1: The yearly cost per terabyte of stored data using Amazon's S3 storage service under April 2011 prices. The per-TB cost drops as the volume of data stored increases due to Amazon's tiered pricing. This does not include bandwidth costs which can be significant.

Each device coordinates with others at its site by forming *reliability groups* (RGs). Each RG is effectively a distributed RAID group, and can use a desired reliability mode. For example, creating an RG of 4 nodes with 3 nodes storing data and 1 node storing parity allows for recovery from a single device failure per RG. To withstand entire site failures, we can aggregate individual sites into effectively another RG as well, treating portions of entire sites as data or parity storage [26], allowing the repair of a complete site in a fashion similar to an individual device failure. Figure 4 illustrates our inter-device and inter-site reliability schemes.

3.3 Cost-Benefit Analysis

San Diego Supercomputer Center [18] estimates that infrastructure and running costs are 2/3 of total cost of ownership (TCO) and media costs are 1/3, with a typical hardware refresh cycle of 3 to 4 years. A DAWN architecture should achieve similar power savings to FAWN: two orders of magnitude. In addition, its hardware refresh cycle should be much longer, with administrative costs reduced from fewer data migrations and greater device autonomy. With a factor of 3 from effectively eliminating running costs, and a factor of 5 (15 years) from extended hardware refresh, spending a factor of 15 more per byte on the storage medium but achieving comparable monthly total cost is reasonable. Note that per-GB flash costs are generally dropping faster than hard disk costs [11], and longer hardware refresh cycles decrease vulnerability to media and infrastructure price volatility. Even with the added cost of embedding a CPU and RAM onto the device, we still maintain competitive costs. For example, a consumer 1 TB disk-based NAS box costs as little as $100 [7] suggesting that the additional cost to embed CPU, RAM and the network interface is very low. The additional cost will be offset by the reduction of administrative needs and reduced high-level infrastructure costs such as cooling.

Node Lifespan	Usable Per-TB Yearly Cost
1 Year	$4300
2 Years	$2150
3 Years	$1433
4 Years	$1075
5 Years	$860
6 Years	$716

Table 2: The yearly cost per terabyte of stored data using DAWN with 3x replication with a 350 dollar per 250 GB node cost, see discussion of node cost in text body. Note this only includes estimated acquisition costs for DAWN nodes, and not running and infrastructure costs. Regardless, when compared to Table 1's S3 costs, we see DAWN's acquisition costs, once we reach a 4 year lifespan, are significantly cheaper than S3, especially at smaller scales. This suggests that DAWN has the potential to be competitive if a suitable architecture can be realized.

Alternatively, owning flash can be compared to renting storage from Amazon's S3. Amazon replicates data for reliability, but probably not more than 3 times. Amazon's S3 pricing is illustrated in Table 1. We assume $350 for a 250 GB DAWN node—we estimated the per-node cost using retail prices for an SSD and low-power compute board. Thus, if the nodes last more than 3 years, buying and we store 3 replicas, DAWN's acquisition cost is already be cheaper. This does not include, however, the infrastructure costs, but does show that we have a window of opportunity wherein we may be able to maintain lower TCO than S3. As we have been arguing throughout the

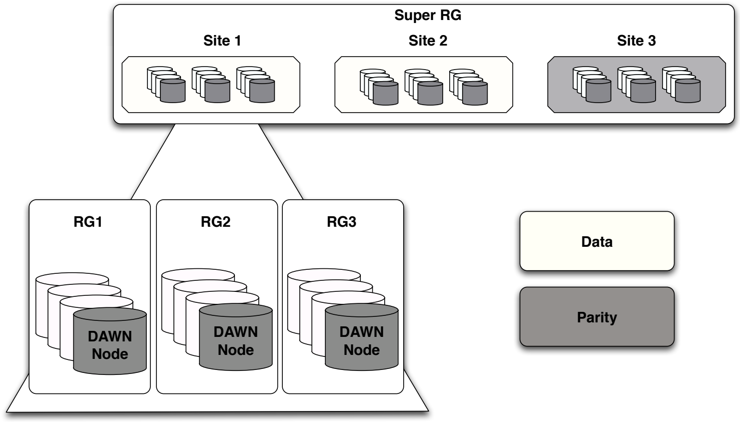

Figure 4: A visualization of our inter-device and inter-site reliability schemes for DAWN. Devices coordinate with each other to form reliability groups for ECC (or replica) management within sites. Entire sites may be similarly aggregated to form a super reliability group to withstand entire site failures.

prior sections, we should be able to significantly reduce running and infrastructure costs through an SCM based approach.

Another important factor often overlooked is the ability of flash and future SCMs to scale down as well as up. Home users and smaller organizations don't have the funds for expensive infrastructure for storage, and their data growth rates are typically sufficiently low that it is not important to try to keep up with Kryder's law [32], which states that areal densities for magnetic storage double roughly annually. Buying fewer, low-maintenance devices for archiving of personal data is more appealing than repeatedly purchasing hardware and migrating data. In this scenario, the "set-it and forget-it" self managing architecture of DAWN would have a large advantage over disk and tape based architectures.

4 Future Work

Our investigation has shown that storage class memories have many characteristics favorable to digital archiving. Despite this, more investigation in several areas is needed in order to fully understand the tradeoffs and feasibility of a storage class memory approach.

Understanding SCM Error and Data Retention. More investigation into the error, failure and data retention characteristics and rates of SCM is needed. Flash, which is currently the best understood of the storage class memories, is only *beginning* to be effectively characterized and modeled. Up and coming SCMs such as phase change memory and memristors are even less understood. If we are to intelligently analyze their cost-benefit tradeoffs, let alone design effective architectures, we *must* have accurate information on their data retention characteristics, failure modes and bit-error rates.

Opportunity Costs and Infrastructure Needs. More investigation into the opportunity costs incurred when using long-lived storage devices is needed. For example, consider the ongoing increase in the bit-density of modern storage media. What currently takes 2 to 4 storage devices to store currently, may only take a single device in 5 years time. We are arguing that a long-lived device has advantages in archival storage, but it may be less economically sound in the face of increasing storage densities. Much of this opportunity cost is based around how valuable a given piece of physical space is. This calculation not only includes the storage density, but the necessary networking and power infrastructure as well.

We need to have a better understanding of the infrastructure costs and needs of both current storage systems and a fully realized DAWN system. While we firmly believe that SCM based approaches will have significantly lower requirements, we must accurately quantify them. For example, this includes potentially *increased* network infrastructure requirements due to our use of many small, low-power nodes rather than heavier weight storage arrays.

Scaling Up. We must also do more investigation into scaling up to very large scales. If we are looking at an automated system of 10 or even 100 thousand DAWN nodes, current distributed management and communication techniques will not scale. As such, we are looking at hierarchical management techniques, like proposed by Storer *at al.* with their Logan management system [27]. In Logan, they subdivide the system into semi-autonomous *management groups* for the purposes of health monitoring, resource location, and replica management.

5 Conclusions

Despite their relatively high cost per bit, storage class memories have many qualities that make them appealing for long-term storage: they are physically robust, low-power, very fast, and capable of withstanding extended periods of benign neglect, since they lack moving parts and their corresponding failure modes. Their very low energy consumption significantly reduces infrastructure needs and running costs. To leverage these qualities, we have described DAWN, an architecture that combines SCM with low-power system-on-chip technology in durable nodes whose only connector is power-over-Ethernet. DAWN nodes manage themselves and cooperate to maintain data replication and integrity for the long term, providing long-term archival storage based on purely electronic media.

References

[1] I. F. Adams *et al.* Analysis of workload behavior in scientific and historical long-term data repositories. Tech. Report UCSC-SSRC-11-01, UC Santa Cruz, Mar. 2011.

[2] A. Amer *et al.* Design issues for a shingled write disk system. In *Proc. of MSST '10*, May 2010.

[3] D. G. Andersen *et al.* FAWN: A fast array of wimpy nodes. In *Proc. of SOSP '09*, Oct. 2009.

[4] N. Anderson. Epa: Power usage in data centers could double by 2011. `http://arstechnica.com/old/content/2007/08/epa-power-usage-in-data-centers-could-double-by-2011.ars`, August 2007.

[5] L. N. Bairavasundaram *et al.* An analysis of latent sector errors in disk drives. In *Proc. of the SIGMETRICS '07*, June 2007.

[6] M. Baker *et al.*. Why traditional systems don't help us save stuff forever. In *Proc. of HotDep '05*, June 2005.

[7] Buffalo linkstation live. `http://www.buffalotech.com/products/network-storage/home-and-small-office/linkstation-live-ls-chl/`, March 2011.

[8] M. Canada. LTO-4 Ultrium data cartridge. `http://www.maxellcanada.com/faqs/LTO4/LTO4.htm`, March 2011.

[9] P. Desnoyers. Empirical evaluation of NAND flash memory performance. In *Proc. of HotStorage '09*, Oct. 2009.

[10] J. Elerath. Hard-disk drives: The good, the bad, and the ugly. *Communications of the ACM*, June 2009.

[11] D. Floyer. Flash pricing trends disrupt storage. `http://wikibon.org/wiki/v/Flash_Pricing_Trends_Disrupt_Storage`, May 2010.

[12] S. Goldstein. Storing research data "forever". `http://www.cni.org/tfms/2010b.fall/Abstracts/Presentations/cni_nsf_goldstein.ppt`, December 2010.

[13] Y. Kang and E. L. Miller. Adding aggressive error correction to a high-performance compressing flash file system. In *Proc. of EMSOFT '09*, Oct. 2009.

[14] J. Martin. Four decades later, recovering lunar images. `http://news.cnet.com/2300-11386_3-10004237.html?tag=mncol`, July 2010.

[15] M. McLaren. Memristors and new memory hierarchies. `http://www.cse.scitech.ac.uk/disco/mew21/presentations/HP.pdf`, 2010.

[16] H. Mehling. Phase change memory: The next big thing in data storage? `http://www.enterprisestorageforum.com/technology/article.php/3862741/Phase-Change-Memory-The-Next-Big-Thing-in-Data-Storage.htm`, February 2010.

[17] N. Mielke *et al.* Bit error rate in NAND Flash memories. In *Proc. of IRPS '08*, July 2008.

[18] R. L. Moore *et al.* Disk and Tape Storage Cost Models. In *Archiving 2007*, May 2007.

[19] Y. Pan *et al.* Exploiting memory devicewear-out dynamics to improve nand flash memory system performance. In *Proc. of FAST '11, February 2011*, 2011.

[20] D. Robb. Storage turns power hungry. `http://www.enterprisestorageforum.com/management/features/article.php/3639286/Storage-Turns-Power-Hungry`, October 2006.

[21] D. S. H. Rosenthal. Keeping bits safe: How hard can it be? *Communications of the ACM*, Nov. 2010.

[22] P. Schmid and A. Roos. Benchmark results: Power consumption. `http://www.tomshardware.com/reviews/windows-7-ssd-trim,2705-19.html`, August 2010.

[23] T. J. E. Schwarz *et al.* Disk scrubbing in large archival storage systems. In *Proc. of MASCOTS '04*, Oct. 2004.

[24] S. Skorobogatov. Data remenance in flash memory devices. In *Proc. of 2005 Cryptographic Hardware and Embedded Systems*, 2005.

[25] E. Spanjer. White paper: Flash management - why and how? `www.adtron.com/pdf/Flash%20Management%20(FINAL).pdf`, November 2009.

[26] M. Stonebraker and G. A. Schloss. Distributed RAID—a new multiple copy algorithm. In *Proc. of ICDE '90*, Feb. 1990.

[27] M. W. Storer *et al.* Logan: Automatic management for evolvable, large-scale, archival storage. In Proc.PDSW '08, Nov. 2008.

[28] M. W. Storer *et al.*. Pergamum: Replacing tape with energy efficient, reliable, disk-based archival storage. In *Proc. FAST '08*, Feb. 2008.

[29] Technologic Systemss. TS-7800. `http://www.embeddedarm.com/products/board-detail.php?product=TS-7800`, March 2011.

[30] Kingston Technology. Flash memory guide-kingston. `www.kingston.com/flash_memory_guide/`, February 2011.

[31] D. Vogler. Phase change memory, charge-trapping memories discussions with De Salvo, Leti. `http://tinyurl.com/6yupdhg`, February 2011.

[32] C. Walter. Kryder's law. *Scientific American*, July 2005.

[33] P. Williams *et al.* Predicting the Archival Life of Removable Hard Disk Drives. In *Archiving 2008*, June 2008.