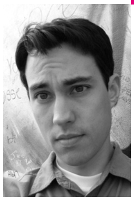
MARK W. STORER, KEVIN M. GREENAN, ETHAN L. MILLER, AND KALADHAR VORUGANTI

Pergamum: energy-efficient archival storage with disk instead of tape

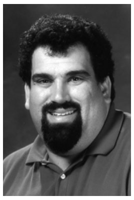


Mark W. Storer is a fourth-year graduate student at the University of California, Santa Cruz. His primary research is archival storage, in particular the security, design, and management of long-term storage. He plans to finish his PhD at the end of calendar year 2008.

mstorer@cs.ucsc.edu

Kevin Greenan is a graduate student at the University of California, Santa Cruz. His interests include system reliability, novel applications of erasure codes in systems, and power-managed systems.

kmgreen@cs.ucsc.edu



Dr. Ethan L. Miller is an associate professor of computer science at the University of California, Santa Cruz, where he is a member of the Storage Systems Research Center (SSRC). His current research projects, which are funded by the NSF, Department of Energy, and industry support for the SSRC, include long-term archival storage systems, scalable metadata and indexing, issues in petabyte-scale storage systems, reliability and security in file systems, and file systems for nonvolatile memory technologies.

elm@cs.ucsc.edu

Kaladhar Voruganti is a Technical Director in the Advanced Technology Group at NetApp. He got his PhD in Computing Science from the University of Alberta in Edmonton, Canada. Kaladhar likes to build systems and also write papers.

kaladhar@netapp.com

IS THE CART LEADING THE HORSE WHEN it comes to long-term digital storage? Few would disagree that there has been a tremendous shift toward writing our personal histories as digital data. The convenience of digital photos has lured people away from film, just as email has supplanted letters. Unfortunately, we have yet to demonstrate that we can reliably preserve digital data for more than a few years. Will future generations be able to browse the sentimental and historical artifacts we leave them the way we might flip through our grandparents' photo albums? Clearly, the long-term preservation of data requires a storage system that can evolve over time, while remaining cheap enough to allow the retention of anything that might be important. To fill this need, we have developed Pergamum, a distributed network of energy-efficient, hard drive–based storage appliances. Each of our devices, which we call a Pergamum tome, offers the low-latency access times of disks while being cheaper to buy and operate than either disk- or tape-based systems.

The Archival Storage Problem

The state of archival storage, in the professional sector, is largely the same as in the private sector, although businesses are slowly starting to recognize the importance of archival storage. Further, they are starting to recognize it as a class of storage distinct from mere backups. This is partially due to legislation that has mandated requirements for the preservation, retrieval, and auditing of digital data. However, outside of legal requirements, data-mining techniques have proven to be a boon in shaping business strategy and have demonstrated the enormous value contained in archival data.

Paradoxically, the increasing value of archival data is driving the need for inexpensive, evolvable storage. The goal of cost-efficient, long-term storage is to enable the potentially indefinite retention of all data that might one day prove useful. With current systems, it is simply too expensive to store everything indefinitely (if any long-term persistence guarantees are available at all). Imagine parents implementing a seven-year retention policy on sentimental data in order to reclaim storage space! With home movies and photography all being done digitally, is this what we are forcing them into? Archival storage therefore needs to be cheap to obtain (static costs), cheap to operate (operational costs), and easy to expand (evolvable). In particular, one of the biggest culprits in high storage costs is energy consumption. Some reports find that commonly used power supplies operate at only 65%–75% efficiency, representing one of the primary culprits of excess heat production and contributing to cooling demands that account for up to 60% of data-center energy usage [7].

Unfortunately, despite its increasing prominence, archival data is still often confused with backup data. The access pattern of archival data is dominated by writes, and data, once written, is rarely changed. Reads are also rare, but although a slight latency penalty is acceptable if it results in cost savings, archival storage must still be fairly accessible. Archival data can be thought of as cold data: You may not need it right away, but it is still useful; its value increases the easier it is to read, query, browse, and search over. In contrast, backup data is a safety net that you only resort to if something else has failed. Moreover, most backup data only needs to live long enough to be superseded by a newer write. Thus, whereas both backup and archival storage are concerned with data safety, the goal of the latter is to maintain both the persistence and the usability of data. This aspect of archival storage can be seen quite clearly in digital libraries. For example, the Digital Library Federation, a consortium of libraries, was formed not only to advance the preservation of digital collections but also to expand their accessibility [4].

As a result of this confusion, digital archives are often relegated to storage systems designed for backup data. Oftentimes, these systems utilize removable media that decouple the media from the access hardware. Although many of these systems seem cost-effective because the media are cheap, when you amortize the high costs of readers, silos, and robots over the number of media, you often discover that these systems are far from a bargain. In addition to hidden costs, decoupled media introduce other problems as well. They generally offer poor access times, and they introduce the need to either preserve complex chains of hardware or institute expensive and time-consuming migration strategies. Tape, the most common media in these decoupled systems, is further hindered by a sequential access pattern. The end result is that it can take on the order of minutes to handle random requests. This conspires against many archival storage operations—such as auditing, searching, consistency checking, and inter-media reliability operations—that rely on relatively fast random-access performance.

In contrast to the decoupled media and reader of tape, hard drive–based storage is an attractive alternative. By coupling the heads and the media, hard drives offer better performance and obviate the need for robotics, reducing physical movement and system complexity. Recent trends showing drive prices dropping relative to tape [8] reinforce the idea that disk-based systems may be feasible for archival storage. Even more promising, recent work on MAIDs (Massive Arrays of Idle Disks) has demonstrated that considerable energy-based cost savings can be realized, while still maintaining high levels of performance, by keeping hard drives spun down [3, 12].

Although inexpensive hard drives can help control static costs, they cannot, by themselves, fully address all the needs of archival storage. Long-lived data has a potentially indefinite lifetime, and that requires a system that can scale across time as well as capacity. Luckily, a number of recent innovations have opened the door to a new model of evolvable storage. High-performance, low-power CPUs and inexpensive, high-speed networks make it possible to produce a self-contained, network-attached storage device [6] with reasonable performance and low power utilization. The Ethernet backplane helps simplify the long-term maintenance, as interfaces and protocols are standardized and have changed much more slowly than storage-specific interfaces. The long-term benefit of a network of intelligent storage devices is that the system can be largely agnostic to how the actual devices are implemented. For example, in fifty years, the devices might utilize holographic storage, but their admission to the group will still only be predicated on the ability to speak a given protocol and their ability to perform a set of well-defined tasks.

The system we have developed using this model, called Pergamum, consists of a distributed network of independent, intelligent storage appliances. Other distributed systems exist, but they either compromise a fully distributed design for easier management [5] or do not achieve the level of power savings needed in archival storage [13, 9]. Although each device in our system is fully self-sufficient and manages its own consistency checking and disk scrubbing, the devices cooperate in inter-device redundancy schemes so that, even if a unit fails, data can be rebuilt.

The Design of Pergamum

The Pergamum tomes that make up our system are simple, intelligent storage devices. As Figure 1 shows, each unit is composed of four hardware components: a commodity hard drive for persistent, large-capacity storage; on-board flash memory for persistent, low-latency metadata storage; a low-power CPU; and a network port. The result is a reliable, low-power storage device that can be used as a building block for more advanced systems.

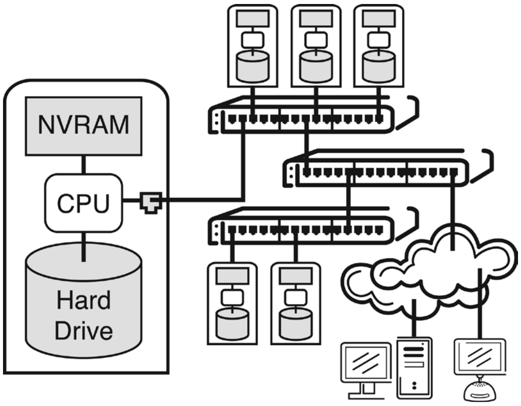


FIGURE 1: HIGH-LEVEL SYSTEM DESIGN OF PERGAMUM. INDIVIDUAL PERGAMUM TOMES ARE CONNECTED BY A COMMODITY NETWORK BUILT FROM OFF-THE-SHELF SWITCHES.

When fully active, each Pergamum tome consumes less than 13 watts, well within the capabilities of power over Ethernet. Of that, the disk itself is by far the largest energy consumer. This explains why MAID systems have been able to achieve considerable power savings by keeping idle disks spun down. Our design goes a step further and achieves even more cost savings by moving away from the power-hungry, centralized controllers found in most MAID systems. Each Pergamum tome consumes less than a single watt in its spun-down, idle mode! By pairing a 2–3 watt processor with each disk, we can gracefully scale the power consumed to the size of the system's load.

The form factor of each Pergamum tome can be quite compact. As the low-power processing boards are roughly the size of a pack of gum (or smaller), the entire device would not be much larger than the drive itself. With power over Ethernet, each Pergamum tome is essentially a sealed device with a single connection. This, together with the lower air space requirements of idle drives, means that very high storage densities can be achieved. It also opens the door for novel rack configurations. Unlike a tape silo, there is no need to provide room for robots to operate.

In addition to our choice of numerous low-power processors, the theme of scaling the response to the size of the task can also be seen in our reliability model. As Figure 2 shows, Pergamum utilizes two levels of redundancy encoding: intra-disk and inter-disk. Individual segments are protected with redundant blocks on the same disk (those labeled with a P). Redundancy groups are protected by the shaded segments (labeled R), which contain erasure correcting codes for the other segments in the redundancy group. Note that segments used for redundancy still contain intradisk redundant blocks to protect them from latent sector errors. Recent work has highlighted the danger of latent sector errors on disks [2]. These are errors that often go undetected until they are read. In a traditional RAID system, at the first sign of any trouble, all the disks in the redundancy group would be spun up. This one-size-fits-all approach to data recovery works well, but it is very expensive. In our system, for many errors, the Pergamum tome can utilize its own intra-disk parity to recover from such errors without waking up a single other device. By using two levels of redundancy, Pergamum achieves higher reliability compared to traditional RAID setups and much higher power savings for the price of only a slight increase in storage overhead.

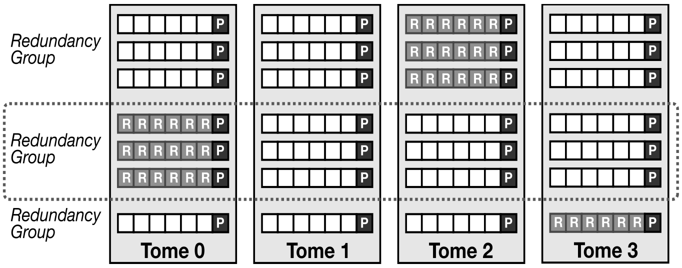


FIGURE 2: THE TWO LEVELS OF REDUNDANCY IN PERGAMUM

Although flash memory has received a fair amount of press lately, it will be some time before it is cheap enough to use as the primary storage medium for archival data. This is not, however, to say that it is not useful in long-term storage. As Figure 1 shows, each Pergamum tome contains a small amount of nonvolatile RAM. This is used as a persistent metadata store. It allows the Pergamum tome to handle many types of requests without spinning up the disk.

One of the most important types of metadata that we store in NVRAM is data signatures. Similar to a traditional hash value, such as SHA-1, data signatures allow us to confirm the correctness of data during a disk scrub or read request; the stored signatures can be compared to a signature calculated over the data being read. The signatures we use, however, are called algebraic signatures and they display a rather useful characteristic [10]. For many codes, the signatures of the data exhibit the same relationship as the data itself. In other words, if data blocks A and B generated parity block P (as in a traditional RAID system), then the signatures of A and B will generate the signature of P. Using these signatures, Pergamum is able to check the consistency not only of the data stored on each Pergamum tome but of the entire inter-disk redundancy group. Moreover, using trees of hash values, we greatly reduce the amount of signature data we need to exchange between nodes in order to confirm the integrity of the inter-disk redundancy groups. We describe our approach in full detail in our FAST ’08 paper on Pergamum [11].

Results and Observations

One of the challenges raised by the use of low-power processors is the need for tight engineering and optimized code. Today, even laptop processors have the horsepower to make non-native, high-level scripting languages perform well, albeit at a huge power cost. As this project has spent its life in a research lab and not in an engineering department, almost all of the software running on the Pergamum tome has been implemented in Python, and it can only be considered proof-of-concept–level code. The results are nonetheless promising. For example, whereas data encoding on a Pergamum tome took almost ten times as long as on a laptop processor, the laptop processor consumed more than ten times the power. However, early system profiling indicates that native implementations and careful optimizations can reap great improvements in performance.

Although we are still early in development, our experiments suggest that our model is a viable approach to archival storage. Our two-level approach to reliability provides excellent protection against latent sector errors as well as full drive failure, and it does so while incurring only minimal storage overhead. Regarding performance, our initial optimizations suggest that we have yet to tap the full potential of our low-power processors, not to mention the possible level of parallelism inherent in our design. Finally, our cost analysis suggests that we can be very price-competitive with tape, while offering functionality that tape systems simply cannot provide. All of these results and a full explanation of our experimental methods are available in our FAST ’08 paper on Pergamum [11].

Where Do We Go from Here?

Pergamum demonstrates some of the features needed in an archival storage system, but work remains to turn it into a fully effective, evolving, long-term storage system. In addition to the engineering tasks associated with optimizing the Pergamum implementation for low-power CPUs, there are a number of important research areas to examine.

Storage management in Pergamum, and in archival storage in general, is an open area with a number of interesting problems. Management strategies play a large part in cost efficiency; many believe that management costs eclipse hardware costs [1]. The goal of our management research is to maintain the decentralized design of Pergamum, while making the addition and removal of drives as automated as possible. In the model we envision, at a frequency of no more than once a month a minimally trained administrator would be tasked with adding new devices to the network and removing failed devices. Once added, the devices would automatically find the existing nodes and either join their redundancy groups or join new redundancy groups created by the system.

In our current implementation, users interact with Pergamum by submitting requests to specific Pergamum tomes using a connection-oriented protocol. In future versions, the use of a simple, standardized `put` and `get` style protocol, such as that provided by HTTP, could allow storage to be more evolvable and permit the use of standard tools for storing and retrieving information. Further, techniques such as distributed searching that take into account data movement and migration could greatly simplify how users interact with the system.

Part of a storage administrator's role has traditionally been to decide how much storage overhead to accept in order to increase storage redundancy. Moving forward, this role will grow slightly more complicated as administrators increasingly consider energy costs as well. In order to make an informed decision, the interplay of redundancy, storage overhead, and power consumption must be better understood. Part of our work in this area is to develop data-protection strategies that are best suited to the unique demands and usage model of archival storage.

Although there is still a lot of work to do to turn Pergamum into a fully functioning, evolvable, archival storage system, our storage model is promising. In its current state, Pergamum uses low-power, network-attached disk appliances to provide reliable, cost-effective archival storage. Two levels of redundancy encoding, within disks and across disks, provide both reliability and cost savings, as data recovery techniques can be appropriately scaled to the size of the data-loss events. Finally, Pergamum achieves its cost-efficiency goals by controlling both static and operational costs. We keep fixed costs low through the use of standardized network interfaces and commodity hardware, allowing each Pergamum tome to be viewed as an essentially "disposable" appliance; a system operator can simply throw away faulty nodes. Operational costs are controlled by utilizing ultra-low-power CPUs, power-managed disks, and a myriad of new techniques such as local NVRAM for caching metadata and redundancy information to avoid disk spin-ups, intra-disk redundancy, and trees of algebraic signatures for distributed consistency checking.

Historical Note

Our system is named after the Library of Pergamum, one of the most famous libraries of the ancient world. Located in modern-day Turkey, then a part of ancient Greece, the library was built by Eumenes II. Two distinctions make this an apt inspiration for our system. First, the Library of Pergamum is the home and namesake of parchment. At the time, manuscripts were written on papyrus, which was expensive, as it was produced only in Alexandria. Second, the library took great care in the layout of its shelves, as it recognized the importance that airflow played in the long-term persistence of its works.

ACKNOWLEDGMENTS

We would like to thank our colleagues in the Storage Systems Research Center (SSRC), who provided valuable feedback on the ideas in this paper, helping us to refine them.

This research was supported by the Petascale Data Storage Institute under Department of Energy award DE-FC02-06ER25768 and by the industrial sponsors of the SSRC, including Los Alamos National Lab, Livermore National Lab, Sandia National Lab, Agami Systems, Data Domain, Digisense, Hewlett-Packard Laboratories, IBM Research, LSI Logic, Network Appliance, Seagate, Symantec, and Yahoo!.

REFERENCES

[1] E. Anderson, M. Hobbs, K. Keeton, S. Spence, M. Uysal, and A. Veitch, "Hippodrome: Running Circles around Storage Administration," in *Proceedings of the 2002 Conference on File and Storage Technologies (FAST '02)*, Monterey, CA, Jan. 2002.

[2] L.N. Bairavasundaram, G.R. Goodson, S. Pasupathy, and J. Schindler, “An Analysis of Latent Sector Errors in Disk Drives,” in *Proceedings of the 2007 SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, June 2007.

[3] D. Colarelli and D. Grunwald, “Massive Arrays of Idle Disks for Storage Archives,” in *Proceedings of the 2002 ACM/IEEE Conference on Supercomputing (SC ’02)*, Nov. 2002.

[4] Digital Library Federation: http://www.diglib.org (accessed Mar. 2008).

[5] S. Ghemawat, H. Gobioff, and S.-T. Leung, “The Google File System,” in *Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP ’03)*, Bolton Landing, NY, Oct. 2003.

[6] G.A. Gibson and R. Van Meter, “Network Attached Storage Architecture,” *Communications of the ACM*, 43(11):37–45, 2000.

[7] Green Grid Consortium: http://www.thegreengrid.org, Feb. 2007.

[8] W. C. Preston and G. Didio, “Disk at the Price of Tape? An In-depth Examination,” Copan Systems white paper, 2004.

[9] Y. Saito, S. Frølund, A. Veitch, A. Merchant, and S. Spence, “FAB: Building Distributed Enterprise Disk Arrays from Commodity Components,” in *Proceedings of the 11th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pages 48–58, 2004.

[10] T. Schwarz and E.L. Miller, “Store, Forget, and Check: Using Algebraic Signatures to Check Remotely Administered Storage,” in *Proceedings of the 26th International Conference on Distributed Computing Systems (ICDCS ’06)*, Lisbon, Portugal, July 2006.

[11] M.W. Storer, K.M. Greenan, E.L. Miller, and K. Voruganti, “Pergamum: Replacing Tape with Energy Efficient, Reliable, Disk-based Archival Storage,” in *Proceedings of the 6th USENIX Conference on File and Storage Technologies (FAST ’08)*, Feb. 2008.

[12] C. Weddle, M. Oldham, J. Qian, A.-I. A. Wang, P. Reiher, and G. Kuenning, “PARAID: A Gear-shifting Power-aware RAID,” in *Proceedings of the 5th USENIX Conference on File and Storage Technologies (FAST ’07)*, Feb. 2007.

[13] W.W. Wilcke, R.B. Garner, C. Fleiner, R.F. Freitas, R.A. Golding, J.S. Glider, D.R. Kenchammana-Hosekote, J.L. Hafner, K.M. Mohiuddin, K. Rao, R.A. Becker-Szendy, T.M. Wong, O.A. Zaki, M. Hernandez, K.R. Fernandez, H. Huels, H. Lenk, K. Smolin, M. Ries, C. Goettert, T. Picunko, B.J. Rubin, H. Kahn, and T. Loo, “IBM Intelligent Bricks Project—Petabytes and Beyond,” *IBM Journal of Research and Development*, 50(2/3):181–197, 2006.