Exploiting Flash for Energy Efficient Disk Arrays

Shimin Chen[⋆] Panos K. Chrysanthis[†] Alexandros Labrinidis[†]
[⋆]*Intel Labs* [†]*University of Pittsburgh*

A growing concern, energy consumption in data centers has been the focus of numerous white papers, research studies, and news reports [1, 11, 4, 3]. According to a report to U.S. congress [11], the total energy consumption by servers and data centers in U.S. was about 61 billion kWh in 2006, and is projected to nearly double by 2011 [11]. Among the components in data centers, it has been shown that storage experienced the fastest annual growth (20% between 2000 and 2006) in energy consumption [11]. A key goal in energy efficient system design is to achieve energy proportionality [2], i.e., energy consumption being proportional to the system utilization. However, hard disk drives (HDDs), the dominant technology for data storage today, contain moving components, making it difficult to achieve this goal. For example, an enterprise class Seagate Cheetah 15K.4 HDD consumes about 15W under load and 12W when idle for spinning the disk platters [9]. While a disk can be spun down to standby mode for saving energy, it takes on the order of 10 seconds to spin up a disk, potentially incurring significant slowdowns in application response times.

Previous Approach: Exploit Redundancy and NVRAM. One promising solution is to exploit the inherent redundancy in storage systems for conserving energy [5, 12, 7]. Today, most storage systems employ redundancy (e.g., RAID) to achieve high reliability, high availability, and high performance requirements for many important applications. For example, the TPC-E benchmark, which models transaction processing in a financial brokerage house, requires redundancy for the data and logs [10]. As seen by published TPC-E reports on the TPC web site, this requirement is typically achieved by the use of RAID 10 disk arrays.

For saving energy, the idea is to keep only a single copy of the data active and spin down disks containing redundant copies of the data under low load. We call the disks containing the active copy of data, the *active* disks, and the disks that are spun down, the *standby* disks. In order to guarantee the same level of reliability for write operations under low load (e.g., writing to two non-volatile devices), previous studies [5, 12, 7] propose to use NVRAM (i.e., battery-backed RAM) as non-volatile write buffers. When the system is under low utilization, reads are sent to the active disks, while writes are sent to both the active disks and to the NVRAM buffers. When the system sees high load or when the NVRAM buffers are full, the standby disks are spun up and the buffered writes are applied to bring the disks up to date.

Limitations of Previous Approach. There are two main limitations of the previous approach. First, battery-backed RAM is expensive. Its size is often limited to a few hundred MB for a RAID array. Typically, a server-class disk can support about 100MB/s read/write bandwidth. Suppose that under low load, a disk sees 1MB/s write traffic. Then, a 500MB NVRAM buffer will be filled up for the write traffic of a single disk in less than 9 minutes. When the buffer is filled, the standby disks must be spun up to apply the buffered writes. However, a disk supports only a limited number of spin-up/down operations because they introduce wear to the motor and the heads in a disk. In particular, server and desktop disks are often rated at 50,000 spin-up/down cycles (a.k.a. start-stop cycles) [8]. Given a five-year lifetime, this puts a limitation of an average 1.1 spin-up/down per hour. Therefore, the above example with a *single* standby disk will significantly shorten the disk lifetime by about 6 times! Note that in real-world disk arrays, an NVRAM buffer in a RAID controller often handles tens of disks, and thus the situation could be an order of magnitude worse.

Second, the energy savings are bounded by the RAID schemes, leaving a big gap to reach the ideal goal of energy proportionality. For example, when the system is under 1% load, mirror-based RAID schemes (e.g., RAID 10) still keeps 50% of the disks active, and parity-based RAID schemes (e.g., RAID 5) have even smaller savings.

Our Proposal 1: Exploit Flash as Write Buffer. There are several desirable properties of flash: (i) it is non-volatile; (ii) flash is much cheaper with much larger capacity; and (iii) flash is energy efficient and supports energy proportionality well. Moreover, flash-based cache products with hundreds of GB capacity are already available for storage systems [6]. One can utilize the same flash for saving energy. This nicely shares the resource: *the flash-based cache improves I/O performance under high system load and saves energy under low load*. We would like to achieve the following design goals: (i) low disk spin-up/down counts; (ii) good performance of RAID under low utilization, under high utilization, and during state transitions; (iii) flash-friendly data structures and operations; and (iv) low flash capacity requirement. We propose a QMD (Quasi Mirrored Disks) design to achieve these design goals. Preliminary experiments using real-world server traces show that QMD can save 11%–31% energy, and reduce the number of spin-up/downs by 80%.

Our Proposal 2: Applications and Storage Systems Collaborate to Further Save Energy. To further save energy under low load, more disks have to be spun down and not all data can be available on the active disks. While storage systems may guess the future access patterns based on history, the penalty of wrong guesses (i.e., the spin-up delay) is high especially for latency sensitive applications. Therefore, we propose two interfaces for application software (e.g., DBMS) and storage systems to collaborate on saving energy. First, software can divide the address range of a RAID volume into hot and cold address ranges. For example, DBMS can create hot and cold table spaces, and place database objects based on access history. DBMS may opt to expose the choices to the end users (e.g., DBMS admin) showing also the estimated energy costs and response times for query workloads. The storage system guarantees that data in the hot address range are always available on active disks, while it may take a spin-up delay to access the cold data. Second, software can query the status of a cold address range, i.e., whether a spin-up will be necessary to access it. Software may intelligently schedule its work based on the answer. For example, if a database query must access both hot and cold data, knowing that the latter is not immediately available, DBMS can choose to process the part of the query involving hot data first, and postpone accessing the cold data to hide the spin-up delay as much as possible.

Author Background. *Shimin Chen* is a research scientist at Intel Labs. He has been working on exploiting new storage and memory technologies, such as flash and Phase Change Memory (PCM), for improving the performance and power efficiency of database and data-intensive systems.

Panos K. Chrysanthis is the director of the Advanced Data Management Technologies Lab at the University of Pittsburgh. He has been working on energy-efficient data management in the context of battery-operated mobile and tiny devices, including sensor networks, since 1993. Recently, he has been studying the trade-offs among QoS, QoD and energy consumption on new hardware.

Alexandros Labrinidis is an associate professor at the Department of Computer Science of the University of Pittsburgh and co-director of the Advanced Data Management Technologies Lab. His research focuses on user-centric data management for network-centric applications, including web-databases, data stream management systems, sensor networks, and scientific data management.

References

[1] L. A. Barroso. The price of performance: An economic case for chip multiprocessing. *ACM Queue*, pages 48–53, Sept 2005.

[2] L. A. Barroso and U. Holzle. The case for energy-proportional computing. *IEEE Computer*, 40:33–37, Dec 2007.

[3] Emerson Network Power. Energy logic: Reducing data center energy consumption by creating savings that cascade across systems. http://emersonnetworkpower.com/en-US/Brands/Liebert/Documents/White Papers/Energy Logic_Reducing Data Center Energy Consumption by Creating Savings that Cascade Across Systems.pdf.

[4] J. G. Koomey. Estimating total power consumption by servers in the U.S. and the world. http://enterprise.amd.com/Downloads/ svrpwrusecompletefinal.pdf.

[5] D. Li and J. Wang. EERAID: energy efficient redundant and inexpensive disk array. In *ACM SIGOPS European Workshop*, 2004.

[6] NetApp Corporation. Flash cache. http://www.netapp.com/us/ products/storage-systems/flash-cache/flash-cache.html.

[7] E. Pinheiro, R. Bianchini, and C. Dubnicki. Exploiting redundancy to conserve energy in storage systems. In *SIGMETRICS*, 2006.

[8] Seagate Technology LLC. Barracuda 7200.12 data sheet.

[9] Seagate Technology LLC. Cheetah 15K.4 SCSI product manual, rev. d edition, May 2005. Publication number: 100220456.

[10] Transaction Processing Performance Council. TPC benchmark E standard specification version 1.12.0.

[11] US Environmental Protection Agency. Report to congress on server and data center energy efficiency: Public law 109-431.

[12] X. Yao and J. Wang. RIMAC: a novel redundancy-based hierarchical cache architecture for energy efficient, high performance storage systems. In *EuroSys*, 2006.