

HYBUD: An Energy-Efficient Architecture for Hybrid Parallel Disk Systems

Mais Nijim[†]

Computer Science, School of Computing
The University of Southern Mississippi
Hattiesburg, MS 39406
mais.nijim@usm.edu

Adam Manzanares, Xiaojun Ruan, and Xiao Qin*

Department of Computer Science and Software
Engineering, Auburn University
Auburn, AL 36849
{acm0008, xzr0001, xqin}@auburn.edu

Abstract: In the past decade parallel disk systems have been highly scalable and able to alleviate the problem of disk I/O bottleneck, thereby being widely used to support a wide range of data-intensive applications. Optimizing energy consumption in parallel disk systems has strong impacts on the cost of backup power-generation and cooling equipment, because a significant fraction of the operation cost of data centres is incurred by energy consumption and cooling. Although flash memory is very energy-efficient compared to disk drives, flash memory is too expensive to use as a major component in large-scale storage systems. In other words, it is not a cost-effective way to make use of large flash memory to build energy-efficient storage systems. To address this problem, in this paper we proposed a hybrid disk architecture or HYBUD that integrates a non-volatile flash memory with buffer disks to build cost-effective and energy-efficient parallel disk systems. While the most popular data sets are cached in flash memory, the second most popular data sets can be stored and retrieved from buffer disks. HYBUD is energy efficient because flash memory coupled with buffer disks can serve a majority of incoming disk requests, thereby keeping a large number of other data disks in the low-power state for longer period times. Furthermore, HYBUD is cost-effective by the virtue of inexpensive buffer disks assisting flash memory to cache a huge amount of popular data. Experimental results demonstratively show that compared with two existing non-hybrid architectures, HYBUD provides significant energy savings for parallel disk systems in a very cost effective way.

Keywords: hybrid parallel disk systems; flash drives; buffer disks; energy conservation; I/O performance

I. INTRODUCTION

In the last decade, parallel disk systems have been widely used to support data-intensive applications, including but not limited to, video surveillance [1], remote-sensing database systems [2], and digital libraries [5]. The performance of data-intensive applications deeply relies on the performance of underlying disk systems because of the rapidly widening gap between CPU and disk I/O bandwidth [7]. Parallel disk systems play an important role in achieving I/O high-performance for data-intensive applications, because parallelism and scalability of parallel disk systems can

efficiently alleviate the disk I/O bottleneck problem.

Reducing energy consumption of large-scale computing platforms has become an increasingly hot research topic in the realm of high-performance computing. Green computing has recently been targeted by government agencies; efficiency requirements have been outlined by the U.S. Environmental Protection Agency (a.k.a., EPA) [10]. Large-scale parallel disk systems inevitably lead to a huge amount of energy due to scaling issues in data centers, which unsurprisingly consume power anywhere between 75 W/ft² to 200 W/ft². Such a high energy consumption rate is likely to continuously increase up to 200-300 W/ft² in the near future [15]. Large-scale computing systems not only have a large economical impact on governments, companies, and research institutes, but also produce environmental impacts. Data from EPA indicates that generating 1 kWh of electricity in the United States results in an average of 1.55 pounds (lb) of carbon dioxide (CO₂) emissions. With large-scale clusters requiring up to 40TWh of energy per year at a cost of over \$4B, it is reasonable to conclude that energy-efficient clusters can have huge economical and environmental impacts [3].

One way to save power is to spin down disks when they are sitting idle. However, spinning down disks is effective only if the disks can remain in standby for long time periods. Conventional wisdom of reducing disk traffic is to employ a buffer cache. The fundamental idea behind this research is motivated by buffer caches that allow data blocks to be accessed at memory speed if the blocks are residing in the caches. Read and write requests to data stored in the buffer caches do not need to issue disk operations. By eliminating unnecessary disk accesses, the buffer cache can play a major role in saving power by minimizing the number of disk spin ups and downs. In this study, we intend to seamlessly integrate parallel disk systems with flash drives, in which popular data blocks can be prefetched and cached to reduce disk spin up and down times.

Flash memory is a form of non-volatile storage device that has gained popularity in the past few years. Data is stored in semiconductor memory that is approximately as fast as DRAM

[†] Corresponding Author. *The work reported in this paper was supported by the US National Science Foundation under Grants CCF-0845257 (CAREER), CNS-0757778 (CSR), CCF-742187 (CPA), CNS-0831502 (CyberTrust), OCI-0753305 (CI-EAM), DUE-0837341 (CCLI), and DUE-0830831(SFS), and an Intel gift (Number 2005-04-070), Auburn University startup grant.

with the added advantage of no needing any refreshing to maintain the data. Besides, flash memory has the non-volatility feature, holding data even when power is turned off. Although flash memory is as fast as memory with respect to reads, write performance of flash drives is relatively low. Other drawbacks of flash drives include high cost and limited number of write cycles [11]. As such, using flash drives to build energy-efficient parallel storage systems is not only highly expensive but also has negative impacts on data reliability. An ideal solution to this problem is to incorporate flash drives into traditional parallel storage systems equipped with hard drives.

Hard disks are slow mechanical storage devices. Compare flash memory, because hard drives are inexpensive while offering large data capacities (e.g., one terabytes hard disks are available). Although disk capacities are expected to increase by a factor of 16 by 2013, disk bandwidth and performance are not expected to keep up at the same pace. As a result, the gap between capacity and disk I/O performance will continue to grow.

To improve both performance and energy efficiency of parallel I/O systems, we aim to develop a cost-effective hybrid disk architecture, where flash drives are incorporated into parallel disk systems. One of challenging issues raised in designing hybrid parallel disks with flash drives is to achieve high energy-efficiency in such novel disk systems. Several techniques proposed to conserve energy in disk systems include dynamic power management schemes [4][14], power-aware cache management strategies [23], software-directed power management techniques [20], redundancy techniques [16], and multi-speed settings [6][8][13]. However, the research on energy-efficient hybrid parallel disk systems with flash drives is still in its infancy. Therefore, we make an effort to reduce energy dissipation in hybrid parallel disk systems while maintaining high I/O performance.

In a hybrid parallel disk system, flash drives are deployed to aggregate smaller idle periods into idle times that are larger than disk break even time, which is defined as the length of an idle time required for a disk to energy efficiently transition from the active state to the standby state. To build cost-effective parallel disks, only a small number of flash drives can be employed due to the high cost of flash drives. In addition to flash drives, a portion of disks (hereinafter referred to as buffer disks) in a parallel disk system can be used to cache popular data sets (see [12][18] for details on buffer-disk-based parallel disk systems). Buffer disks along with flash drives can help in achieving high energy efficiency and performance for hybrid parallel disk systems, where a large number of data disks are placed in standby as long as possible by the virtue of an increasing number of large idle time periods.

The rest of the paper is organized as follows. Section II summarizes related work. Section III presents the hybrid disk architecture or HYBUD. In section IV we describe an energy consumption model to facilitate the development of energy-efficient parallel disk systems. Section V describes disk request processing mechanisms in HYBUD. In Section VI we

analytically study the energy efficiency of HYBUD. Section VII presents experimental results. Finally, Section VIII concludes the paper with discussions of the main features of HYBUD.

II. RELATED WORK

Parallel Disk Systems: Disk I/O has become a performance bottleneck for data-intensive applications due to the widening gap between processor speeds and disk bandwidth [14]. To help alleviate the problem of disk I/O bottleneck, a large body of work has been done on parallel disk systems. For example, Kallahalla and Varman designed an on-line buffer management and scheduling algorithm to improve performance of parallel disks [15]. Scheuermann *et al.* addressed the problem of making use of striping and load balancing to tune performance of parallel disk systems. Rajasekaran and Jin developed a practical model for parallel disk systems [16]. Our research is different from the previous studies in that we focused on energy savings for parallel disk systems. Additionally, our strategy is orthogonal to the existing techniques in the sense that our scheme can be readily integrated into existing parallel disk systems to substantially improve energy efficiency and performance of the systems.

Energy Saving for Single Disks: Most of the previous research regarding conserving energy focuses on single disk system (see, for example, [17]) like laptop and mobile devices to extend the battery life. Recently, several techniques proposed to conserve energy in storage systems include dynamic power management schemes [9], power aware cache management strategies [20], power aware perfecting schemes [21], software-directed power management techniques [22], redundancy techniques [22], and multi-speed settings [23]. Our study and the above research are different in that we aim to improve energy efficiency for parallel disk systems with flash drives and hard disks.

I/O Buffer Management: Buffer management has been used to boost performance of parallel disk systems [12][18][24]. Previous studies showed that the data buffers significantly reduce the number of disk accesses in parallel disk systems [25]. More importantly, it is observed from the previous studies that traffic of small writes becomes a performance bottleneck of disk systems, especially when RAM sizes for data buffers are increased rapidly [25]. It is expected that small writes dominate energy dissipation in parallel disk systems that support data intensive applications like remote sensing applications.

Memory Caches: Main memory caches based on volatile memory have long been used to reduce disk traffic in order to improve response time and throughput. More recently, the researchers have explored the idea of using non-volatile memory to reduce write traffic [11]. Marsh *et al.*, developed an architecture that used flash memory as a second level cache to conserve energy as well as to improve performance.

Our study differs from the aforementioned work is that we paid attention to cost-effective parallel disk systems, where

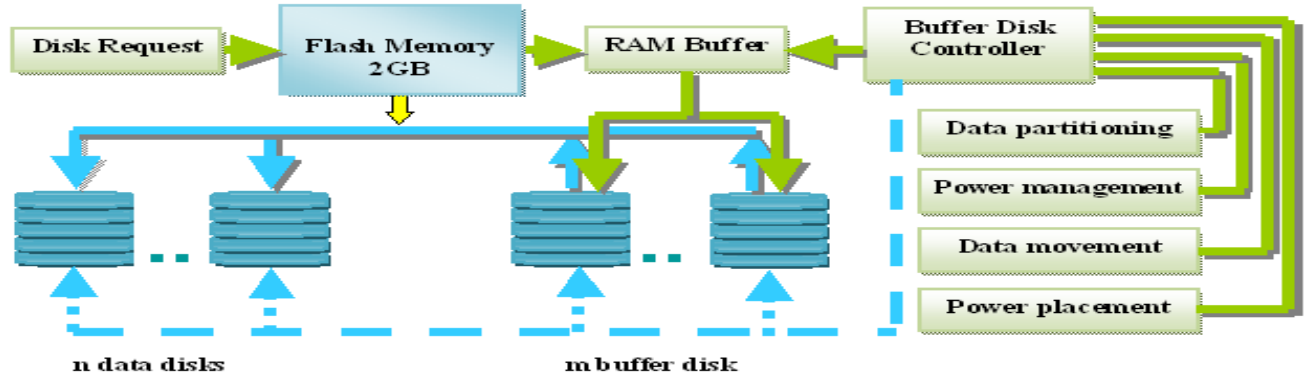


Figure 1. HYBUD: The architecture for cost-effective hybrid parallel disk systems with buffer disks. HYBUD contains a 2-GB flash drive, m buffer disks, n data disks, and an energy-aware flash drive/buffer disk controller.

flash drives and buffer disks are combined to reduce energy dissipation in the disk systems while maintaining high parallel I/O performance.

III. THE HYBUD DISK ARCHITECTURE

Fig. 1 depicts a hybrid disk architecture or HYBUD containing a 2-GB flash drive, m buffer disks, n data disks, and an energy-aware flash drive/buffer disk controller. Note that the values of n and m , which are configuration on the fly, are independent of each other. A RAM buffer with a size ranging from several megabytes to gigabytes is incorporated to further improve I/O performance in HYBUD. The flash drive/buffer disk controller coordinates multiple modules, including power management, data partitioning, disk request processing, and perfecting schemes.

The 2-GB flash drive performs as a non-volatile data cache to boost I/O performance and improve energy efficiency by absorb disk traffic fluctuations. The flash drive respond to both read and write disk requests. A read miss in the flash drive causes a hit at one of the buffer disks. the block to be fetched from data disks and written into the flash memory. Write requests are served by the flash drive first. If the flash drive is full, write requests are redirected to buffer disks.

A prefetching scheme is designed to bring data into buffer disks or flash drives before its use. Apart from the prefetching scheme, we developed a write strategy to energy-efficiently handle writes using flash drive and buffer disks. The write I/O load imposed on buffer disks is well balanced by equally distributed write request to all the active buffer disks to make the utilization of all the buffer disks identical.

To improve I/O performance of buffer disks handing write requests, we chose to use a log file system that allows data to be written sequentially buffered in buffer disks to minimize disk seek times and rotational delays. We developed a buffer disk manager that is responsible for the following activities. First, the disk manager aims to minimize the number of active buffer disks while maintaining reasonably quick response time for disk requests. Second, the manager must deal with the read and write requests redirected from the flash drive in an energy-efficient way. Third, the manager has to energy-efficiently move data among the flash drive, buffer disks, and data disks.

IV. ENERGY CONSUMPTION MODEL

A. Energy Dissipation in Parallel Disk Systems

To reduce energy consumption for parallel disk system, modern disks use multiple power modes that includes active, idle, and standby mode. The basic power model for the parallel disk system is the summation of all power states multiplied by the time that each power state was active. The states used are start-up, idle, and read/write/seek. Read, write, and seek are put together because they shared the same power consumption.

Let T_i be the time required to enter and exit the inactive state. The power consumption of a disk when entering and exiting the inactive state is P_i . Therefore, energy E_i consumed by the disk when it enters and exits the inactive state is expressed as $P_i \cdot T_i$. Let T_{active} be the time interval when the disk is in the active state. The power consumption rate of the disk when it is in active state is denoted by P_{active} . Thus, the energy consumption of the disk when it is in the active state can be expressed as $E_{active} = P_{active} \cdot T_{active}$. Similarly, let T_{idle} be the time interval when the disk is in idle state. The power consumption rate of the disk when it is in idle state is represented by $E_{idle} = P_{idle} \cdot T_{idle}$. The total energy consumed by the disk system can be calculated as:

$$\begin{aligned} E_{total} &= E_{flash} + E_{tr} + E_{active} + E_{idle} \\ &= E_{flash} + P_{tr} \cdot T_{tr} + P_{active} \cdot T_{active} \\ &\quad + P_{idle} \cdot T_{idle} \end{aligned} \quad (1)$$

where E_{flash} is the energy consumed in flash memory of the disk request where it is computed in section 4.2.

Let T_{ai} and T_{ia} denote the times a disk spends in entering and exiting the inactive state, and let P_{ai} and P_{ia} be the power consumption rates when the disk enters the inactive and active state. N_{ai} and N_{ia} are the number of times the disk enters and exits the inactive state. Thus, the transition time $T_{transition}$ is computed as follows

$$T_{transition} = N_{ai} T_{ai} + N_{ia} T_{ia} \quad (2)$$

The power transmission is computed by

$$P_{transition} = P_{ai} + P_{ia} \quad (3)$$

$$E_{tr} = \frac{T_{ai}}{T_{ai} + T_{ia}} P_{ai} + \frac{T_{ia}}{T_{ai} + T_{ia}} P_{ia} \quad (4)$$

The time interval T_{active} when the disk is in active state is the sum of serving times of disk requests submitted to the parallel disk system.

$$T_{active} = \sum_{i=1}^n T_{service}(i), \quad (5)$$

where n is the total number of requests submitted to the system, and $T_{service}(i)$ is the serving time of the i th disk request and is calculated by

$$T_{service}(i) = T_{seek}(i) + T_{rot}(i) + T_{trans}(i), \quad (6)$$

where T_{seek} is the amount of time spent seeking the desired cylinder, T_{rot} is the rotational delay and T_{trans} is the amount of time spent actually reading from or writing to disk.

Now the energy saved by our management policy is quantified as,

$$\begin{aligned} E_{save} &= (T_{active} + T_{idle} + T_{tr})P_{active} - E_{total} \\ &= (T_{active} + T_{idle} + T_{tr})P_{active} - \\ &\quad E_{flash} + (T_{active}P_{active} \\ &\quad + T_{idle}P_{idle} + T_{tr}P_{tr}) \\ &= E_{flash} + (P_{active} - P_{idle})T_{idle} \\ &\quad + (P_{active} - P_{tr})T_{tr} \end{aligned} \quad (7)$$

Where E_{flash} is the energy consumed in the flash memory.

The transition power consumption is not considered in this study, for this model it is important to decide the power consumption for each state and the power consumption in flash memory. These values can be obtained based on physical hard disk tests, and published papers. In section 6, we build an analytical model based on queuing theory to calculate the energy consumption for the system.

B. Energy Consumption in Flash Memory

Flash chips have emerged as the storage technology of choice for numerous consumer devices as well as for networked systems. Their low energy consumption makes them an attractive choice for parallel disks systems.

In this study, we use a Toshiba TC58DVG02A1FT00 2GB NAND flash [26]. Table 1 shows the energy cost of the used flash memory.

Based on the measurements mentioned in Table1, the energy cost writing d bytes of data is calculated by

$$W(d) = 24.54 + d.0.0962 \mu J \quad (8)$$

The energy cost of reading d bytes is calculated by

$$R(d) = 4.07 + d.0.105 \mu J \quad (9)$$

TABLE 1 ENERGY COST OF FLASH OPERATIONS

		WRITE	READ
ENERGY COST	FIXED COST	13.2MJ	1.073MJ
	COST PER-BYTE	0.0202MJ	0.0322MJ
LATENCY	FIXED COST	0.0202MJ	0.0322MJ
	COST PER-BYTE	1.530us	1.761us
FLASH ENERGY+CPU ENERGY COST	FIXED COST	24.54MJ	4.07MJ
	COST PER-BYTE	0.0962μJ	0.105μJ

We can notice that the energy cost of write is 13 times larger than read energy cost, whereas the cost per additional byte is almost the same of both write and read.

V. REQUEST PROCESSING IN HYBUD

In this section, we describe in details on a request processing mechanism in the HYBUD architecture (see Section 3 for the architecture). Essentially, the processing mechanism provides solutions of energy-efficiently handling read and writes requests in hybrid parallel disk systems.

A. Read Request

Handling read requests is kind of simple and straightforward. Read requests first arrives to the flash memory. If the data block is resided in the flash, then the data is immediately sent back to the requester. If the requested data is not in the flash, a copy of the data will be written to the flash assuming that this data block is going to be used frequently. The data will be retrieved from the corresponding data disk if it is in active mode, otherwise, the read requests will be clustered together in the flash waiting for the corresponding data disk to be in active mode. When the flash memory is full, dirty blocks will be flushed to the buffer disk. On the other hand, the buffer disk clusters the miss read requests together. By clustering the read requests, the data disks will be able to stay in the sleep mode for longer periods of time.

B. Write Request

Modern parallel disk system usually implements write-back caching. In this case, unlike read, a write request is completed once the data is written to the flash memory. If the corresponding disk is in active state, the data block will be written directly to the data disk. Otherwise, the write request will be kept in flash memory and dirty data are flushed to buffer disk according to a cache replacement policy. In this study we use a least recent used policy (LRU).

Once the dirty data are flushed to the RAM buffer as shown in section 3, the buffer disk controller responsibility is in two fold. First, the controller will check the size of the write requests. The write requests are divided into small write and

large write requests. If the request is large write for example 10MB or more, the request will be sent directly to the corresponding data disk. Otherwise, the controller will send the write request to the RAM buffer that buffers small write requests together and form a log of write requests that will be written to the data disk later. Our focus for this study is on small write requests. Second, the controller will test the state of all buffer disks. If the buffer disk is not busy with writing a previous log, the data will be written to the buffer disk to ensure that a reliable copy resides on one of the buffer disks. Operations which could write the same block data into different buffer disks is forbidden if one legal copy of this block still exists in any buffer disk.

C. The HYBUD Power Management

The ultimate goal of this study is to conserve energy as much as possible without sacrificing the system performance. To reduce energy consumption, modern disks use multiple power modes that include active, idle, standby and shutdown modes. In active mode, the platters are spinning and the head is seeking or the head is actively reading or writing a sector. In idle mode, a disk is spinning at its full speed but no disk activity is taking place. Therefore, staying in idle mode when there is no disk request provides the best possible access time since the disk can immediately service request, but it consumes the most energy. To simplify discussion, we don't differentiate active mode and idle mode since in both modes the disk is operating at its full power. In the standby mode, the disk consumes less energy, but in order to service a disk request, the disk will incur significant energy and time overhead to spin up. When the dirty data is flushed to the buffer disk, the buffer disk controller will be always trying to keep as more data disks in sleeping mode. Once a data disk is waken up, it will be keeping busy for a while because a large trunk of data coming from RAM buffer directly or from buffer disks will be written to it.

In order to fully utilize the gap of energy consumption rate under different mode in the hybrid architecture, the flash memory and the buffer disk controller keeps as more data disks as possible in sleeping mode. In term of flash memory, the data block will be written in the flash. If the corresponding disk is in active mode, *the data block will be written to the disk. Otherwise, the data block will be kept in the flash and the dirty block will be flushed to the buffer disk.* As a result, the data disk will stay in sleeping mode saving more energy same time, the controller will set up a time threshold for the weakened up data disks. If the idle time exceeded the threshold, the data disk will be turned back to sleep mode to save power. Bu using this strategy, we can conserve energy without sacrificing the performance of the parallel disk system.

VI. EXPERIMENTAL RESULTS

Our preliminary results consist of developing a simulator, which meets all projects specifications and implementing all the required functions that are necessary to model our

distributed system. We will compare our HYBUD strategy with two baseline disk architectures without using a hybrid scheme. The first one is called flash-based disk architecture where a parallel storage system only contains flash drives. The second architecture is referred to as BUD [12][18], where only the buffer disks are employed to conserve energy in storage systems.

A. Impacts of miss rate

This experiment is focused on comparing the HYBUD strategy against the two baseline architectures described above. Let us study the impacts of miss ratio on the normalized energy consumption ranging from 0 to 1. To achieve this goal, we gradually increased the miss ratio from 75% to 100%.

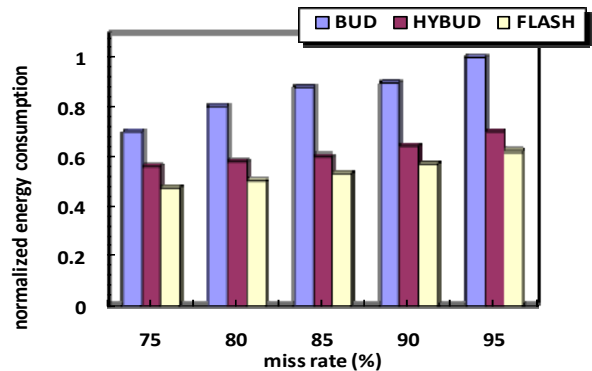


Fig.2 energy consumption versus miss ratio

Fig. 2 plots empirical results when there are five disks in a parallel I/O system and the average size of disk requests is 300 MB. As the miss rate is increased, the energy consumption of the three strategies also increased. The Flash strategy consumes less energy than the other two alternatives strategy. Different from the hard disk, the flash drive is made of solid-state chips without any mechanical component, such as disk platters, which consumes a huge amount of energy. Moreover, the flash drive does not need power to maintain its data. Thus, the energy consumption of the flash drive is almost negligible compared with the hard disk.

VII. CONCLUSIONS

Parallel disk systems play an important role in achieving high-performance for data-intensive applications, because I/O parallelisms can alleviate the disk I/O bottleneck problem. Growing evidence shows that a substantial amount of energy is consumed by parallel disk systems in data centers and; therefore, we have to design next generation parallel disk systems that are energy-efficient in nature. Although flash drives are highly energy efficient compared to hard disks, flash drive is too expensive to make completely flash-drive-based parallel storage system cost-effective. To address this problem, in this paper we proposed a hybrid disk architecture - HYBUD - that integrates a flash drive with parallel disks to facilitate the development of cost-effective and energy-efficient parallel disk systems. Hard disks in HYBUD are classified into two groups

– buffer disks and data disks. HYBUD attempt to keep only a small number of buffer disks active to serve requests redirected from the flash drive and data disks, where a large number of data disks are allowed to be placed into standby for a long time periods. The most popular data sets are prefetched and cached in the flash drive in HYBUD, whereas the second most popular data sets are prefetched to buffer disks. To improve I/O performance of buffer disks handing write requests, we chose to use a log file system storing data sequentially in buffer disks to minimize disk seek times and rotational delays.

Our approach is conducive to improving energy efficiency in hybrid parallel disk systems because of the following reasons. First, both flash drives and buffer disks in HYBUD can serve a majority of incoming disk requests, thereby placing a large number of data disks in standby for increased time periods. Second, the number of disk spin ups and downs experienced in data disks is substantially decreased. Third, extra energy incurred by power state transitions is reduced.

Importantly, HYBUD makes parallel disks highly cost-effective, because a huge amount of popular data can be cached in buffer disks in addition to flash drive. We conducted experiments to quantitatively compare HYBUD with two existing non-hybrid parallel disk architectures. Our experimental results show that HYBUD provide a cost-effective way of conserving energy consumption in large-scale parallel I/O systems equipped with both buffer disks and data disks.

ACKNOWLEDGMENT

This work was made possible partially thanks to NSF awards CCF-0845257 (CAREER), CNS-0757778 (CSR), CCF-742187 (CPA), CNS-0831502 (CyberTrust), OCI-0753305 (CI-EAM), DUE-0837341 (CCLI), and DUE-0830831(SFS), and an Intel gift (Number 2005-04-070), Auburn University startup grant, as well as The University of Southern Mississippi startup grant.

References

[1] D. Avitzour, “Novel scene calibration procedure for video surveillance systems,” *IEEE Trans. Aerospace and Electronic Systems*, Vol. 40, No. 3, pp. 1105-1110, July 2004.

[2] C. Chang, B. Moon, A. Acharya, C. Shock, A.Sussman, and J. Saltz. “Titan: a High-Performance Remote-Sensing Database,” *Proc. 13th Int’l Conf. Data Eng.*, Apr 1997.

[3] E. Carrera, E. Pinheiro, and R. Bianchini. “Conserving Disk Energy in Network Servers,” *Proc. Int’l Conf. Supercomp.*, pp.86-97, 2003.

[4] F. Dougliis, P. Krishnan, and B. Marsh, “Thwarting the Power-Hunger Disk,” *Proc. WinterUSENIX Conf.*, pp.292-306, 1994.

[5] T. Sumner and M. Marilino, “Digital libraries and educational practice: a case for new models,” *Proc. ACM/IEEE Conf. Digital Libraries*, pp. 170 – 178, June 2004.

[6] S. Gurumurthi, A. Sivasubramaniam, M.Kandemir, and H. Fanke, “DRPM: Dynamic Speed Control for Power Management in Server Class Disks,” *Proc. Int’l Symp. Computer Architecture*, pp. 169-179, June 2003.

[7] X. Qin, “Performance Comparisons of Load Balancing Algorithms for

I/O-Intensive Workloads on Clusters,” *Journal of Network and Computer Applications*, 2007.

[8] D. P. Helmbold, D. D. E. Long, T. L. Sconyers, and B. Sherrod, “Adaptive Disk Spin-Down for Mobile Computers,” *Mobile Networks and Applications*, Vol. 5, No.4, pp.285-297, 2000.

[9] F. Dougliis, P.Krishnan, and B. Marsh, “Thwarting the Power-Hunger Disk,” *Proc. Winter USENIX Conf.*, pp.292-306, 1994.

[10] E.Jones, (2006-10-23). *EPA Announces New Computer Efficiency Requirements*.U.S. A.Retrieved on 2007-10-02.

[11] B.Marsh, F.Dougliis, and P. Krishnan, “Flash Memory File Caching for Mobile Computers” *Proc. the 27th Annual Hawaii International Conference on system sciences*, 1994.

[12] A. Manzanares, K. Bellam, and X. Qin, “A Prefetching Scheme for Energy Conservation in Parallel Disk Systems,” *Proc. NSF Next Generation Software Program Workshop*, April 2008.

[13] P. Krishnan, P. Long, J. Vitter, “Adaptive Disk Spindown Via Optimal Rent-to-buy in Probabilistic Environments,” *Proc. Int’l Conf. on Machine Learning*, pp. 322-330, July 1995.

[14] S. Rajasekaran, “Selection algorithms for parallel disk systems,” *Proc. Int’l Conf. High Performance Computing*, pp.343-350, Dec. 1998.

[15] M. Kallahalla and P. J. Varman, “Improving parallel-disk buffer management using randomized writeback,” *Proc. Int’l Conf. Parallel Processing*, pp. 270-277, Aug. 1998.

[16] S. Rajasekaran and X. Jin, “A practical realization of parallel disks Parallel Processing,” *Proc. Int’l Workshop Parallel Processing*, pp. 337-344, Aug. 2000.

[17] A. Roth, A. Manzanares, K. Bellam, M. Nijim, and X. Qin, “Energy Conservation for Real-Time Disk Systems with I/O Burstiness,” *Proc. IEEE Int’l Workshop Next Generation Autonomous Storage and High Performance Computing*, St. Thomas, Virgin Islands, Aug. 2008.

[18] X.-J. Ruan, A. Manzanares, K. Bellam, X. Qin, “DARAW: A New Write Buffer to Improve Parallel I/O Energy-Efficiency,” *Proc. the 24th Annual ACM Symposium on Applied Computing*, March 2009.

[19] D. Kotz and C. Ellis, “Caching and writeback policies in parallel file systems,” *Proc. IEEE Symp. Parallel and Distributed Processing*, pp. 60-67, Dec. 1991

[20] Q. Zhu, F.M David, C.F. Devaaraaj, Z. Li, Y.Zhou, and P. Cao, Reducing Energy Consumption Of Disk Storage Using Power Aware Cache Management,” *Proc. High Performance Computer Framework*, 2004.

[21] S.W. Son and M. Kandemir, “Energy Aware data perfecting for multi-speed disks,” *Proc. ACM International Conference on Computing Frontiers*, Ischia, Italy, May 2006.

[22] S.W. Son, M. Kandemir, and A. Choudhary, “Software-directed disk power management for scientific applications,” *Proc. Int’l Symp. Parallel and Distr. Processing*, April 2005.

[23] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, and H. Fanke, “DRPM: Dynamic Speed Control for Power Management in Server Class Disks,” *Proc. Int’l Symp. of Computer Arch.*, pp. 169-179, June 2003.

[24] J.-H Kim, S.-W. Eom, S.H. Noh, and Y.-H. Won, “Stripping and buffer caching for software RAID file systems in workstation clusters,” *Proc. 19th IEEE Int’l Conf. Distributed Computing Systems*, pp. 544-551, 1999.

[25] Y. Hu and Q. Yang, “DCD-Disk Caching Disk: A New Approach for Boosting I/O Performance,” *Proc. Int’l Symp. Computer Arch.*, 1996.

[26] Toshiba America Electronic Componentd, Inc. (TAEC), www.toshiba.com/taec.Datasheet:TC58DVG02AIFT00, Jan 2003