Modelling a Hybrid Energy-Efficient Architecture for Parallel Disk Systems

Mais Nijim and Nooh Bany Muhammad
Computer Science, School of Computing
The University of Southern Mississippi
Hattiesburg, MS 39406
mais.nijim@usm.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 due to energy consumption and cooling. Although flash memory is very energy-efficient compared to disk drives, flash memory is expensive. Thus, it is not a cost-effective way to make use of large flash memory to build energy-efficient parallel disk 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 are cached in buffer disks. HYBUD is energy efficient because flash memory and buffer disks can serve a majority of incoming disk requests, thereby keeping data disks in low-power state for longer period times. Furthermore, HYBUD is cost-effective, since a huge amount of popular data can be cached in buffer disks in addition to the flash memory. Experimental results demonstratively show that compared with two existing non-hybrid architectures, HYBUD provides significant energy savings for parallel disk systems.**

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 due to the rapidly widening gap between CPU and disk I/O speeds [7]. Parallel disk systems play an important role in achieving high-performance for data-intensive applications, because the high parallelism and scalability of parallel disk systems can alleviate the disk I/O bottleneck problem.

Reducing energy consumption of computing platforms has become an increasingly hot research field. Green computing has recently been targeted by government agencies; efficiency requirements have been outlined [10]. Large-scale parallel disk systems inevitably lead to a huge amount of energy due to scaling issues. Data centers typically consume anywhere between 75 W/ft2 to 200 W/ft2 and this may increase to 200-300 W/ft2 in the near future [15]. These large-scale computing systems not only have a large economical impact on companies and research institutes, but also produce environmental impacts. Data from the US Environmental Protection Agency 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 easy to conclude that energy-efficient clusters can have huge economical and environmental impacts [3].

One way to save power is to spin down the disk when it is not in use. However, spinning down the disk is only effective if it can remain spun down for some time. The standard approach to eliminate disk traffic is to use a buffer cache. The buffer cache allows files to be accessed at memory speed. Read and write requests to data stored in the buffer cache require no disk traffic in order to be satisfied. By eliminating unnecessary disk access, the buffer cache can play a major role in saving power by minimizing the number of times the disk needs to be spun up.

Flash memory is a form of non-volatile storage that has gained popularity in the past few years. Data is stored in semiconductor memory that is about as fast as DRAM with the added advantage of no needing any refreshing to maintain the data. Besides, flash memory has the non-volatility feature, which keeps data even when the power is turned off, and the speed and compactness of DRAM. Flash memory is as fast as memory when doing read but much slower when doing write. The other limitation includes high cost and limited number of writes cycles [11].

Hard disks are slow mechanical storage devices. However, because they are inexpensive and offer large capacities (one terabytes hard disks are available), they are used as the back-end media for general purpose operating systems. Although disk capacities are expected to increase by a factor of 16 by 2013, disk bandwidth and seek time are not expected to scale as much. As a result, the gap between drive capacity and performance will continue to grow.

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][12]. However, the research on energy-efficient parallel disk system is still in its infancy. Therefore, we developed hybrid hard disk drive that integrates a non-volatile flash memory with a buffer disk

(HYBUD for short) that is used to conserve energy for parallel disk system. Flash memory and buffer disks can be combined to provide cost-effective and energy conservation for parallel disk system.

The rest of the paper is organized as follows. Section 2 summarizes related publications. Section 3 presents the architecture of the hybrid disk. In section 4 we presents an energy consumption model to facilitate the development of energy-efficient parallel disk systems. Section 5 describes our HYBUD strategy. In Section 6 we analytically study the energy efficiency of HYBUD. Section 7 presents our experimental results and provides a discussion of the results. Finally, Section 8 concludes the paper and discussed future research directions.

II. Related Work

Disk I/O has become a performance bottleneck for data-intensive applications due to the widening gap between processor speeds and disk access speeds [13]. 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[14]. 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 [15]. 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.

Most of the previous research regarding conserving energy focuses on single storage system such as 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 [17], power aware perfecting schemes [18], software-directed power management techniques [19], redundancy techniques [19], and multi-speed settings[20]. However, the research on energy-efficient parallel disk systems is still in its infancy. It is imperative to develop new energy conservation techniques that can provide significant energy savings for parallel disk systems while maintaining high performance.

Buffer management has been used to boost performance of parallel disk systems [21]. Previous studies showed that the data buffers significantly reduce the number of disk accesses in parallel disk systems [22]. 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 [22]. It is expected that small writes dominate energy dissipation in parallel disk systems that support data intensive applications like remote sensing applications.

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 work differs from the above work is that we combined buffer disk and flash memory to save energy and to reduce cost since flash memory is currently still more than twice as expensive as disk. For the cost of flash memory, a computer today can be equipped with a much larger buffer disks and small flash memory.

III. The HYBUD Disk Framework

A parallel disk system is compromised of an array of disks connected by a high -speed network. In this paper, we proposed a hybrid disk energy efficient framework (see Fig. 1), which consists of five major components: a 2GB flash memory, *m* buffer disks, *n* data disks, and an energy-aware buffer disk controller. Hereinafter, we will call this framework HYBUD framework for short.

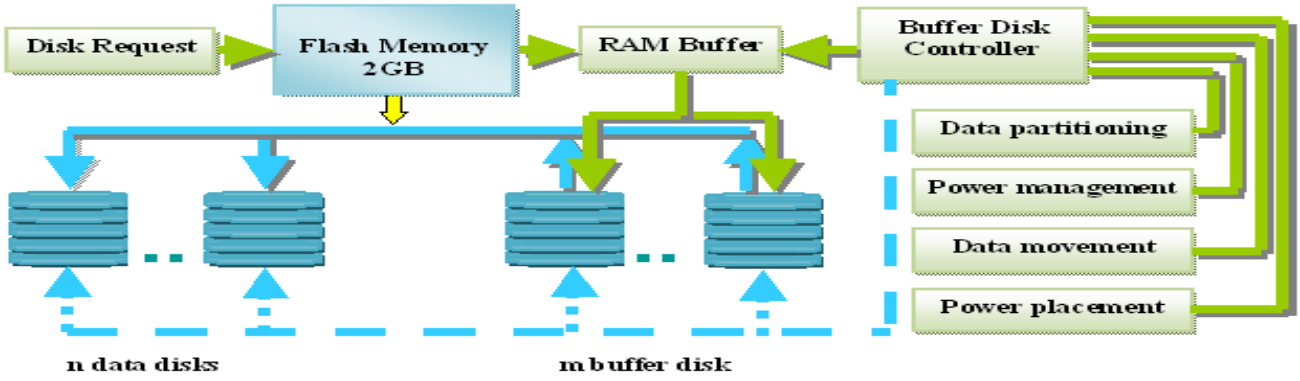

Fig. 1. The Hybrid Architecture for Parallel I/O Systems with Buffer Disks.

We use 2GB flash memory as a non-volatile cache. The flash memory is intended to absorb disk traffic. Blocks are inserted into the flash memory by both read and write requests. A read request whose block is not in the cache will cause the

block to be fetched from data disks and written into the flash memory. Write requests modify blocks in the flash and the modified blocks that no longer fit in the flash are flushed to the RAM buffer.

The RAM buffer with a size ranging from several megabytes to gigabytes is residing in main memory. The buffer disk controller coordinates power management, data partitioning, disk request processing, and perfecting schemes.

To improve the performance of the disk system, we use log disks as buffer disks to allow the data to be written sequentially to improve the performance of disk systems. Note that the value of *n* and *m* are independent of each other where *m* is the number of buffer disks is smaller that *n* which is the number of data disks.

The buffer disk controller is responsible for the following activities. First, it aims to minimize the number of active buffer disks while maintaining reasonably quick response time for disk requests. Second, the controller must deal with the read and write requests in an energy-efficient way. Third, controller has to energy-efficiently move data between buffer and data disk.

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 $P_i . T_i$. 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 $E_{active} = P_{active} . T_{active}$. 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 $E_{idle} = P_{idle} . T_{idle}$. The total energy consumed by the disk system can be calculated as:

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

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

$$T_{transition} = N_{ai} T_{ai} + N_{ia} T_{ia} \tag{2}$$

The power transmission is computed by

$$P_{transition} = P_{ai} + P_{ia} \tag{3}$$

$$E_{tr} = \frac{T_{ai}}{T_{ai} + T_{ia}} P_{ai} + \frac{T_{ia}}{T_{ai} + T_{ia}} P_{ia} \tag{4}$$

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.

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

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

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

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,

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

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 [23]. 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

$$W(d) = 24.54 + d.0.0962\mu J \qquad \textbf{(8)}$$

THE ENERGY COST OF READING *D* BYTES IS CALCULATED BY

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

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.

Table 1 Energy Cost of Flash Operations

		WRITE	READ
ENERGY COST	FIXED COST	13.2*MJ*	1.073*MJ*
	COST PER-BYTE	0.0202*MJ*	0.0322*MJ*
LATENCY	FIXED COST	0.0202*MJ*	0.0322*MJ*
	COST PER-BYTE	**1**.530us	1.761us
FLASH ENERGY+CPU ENERGY COST	FIXED COST	24.54*MJ*	4.07*MJ*
	COST PER-BYTE	0.0962μJ	0.105μJ

V. THE HYBUD ALGORITHM

In this section, we will talk in details about the HYBUD algorithm, which runs on the framework described in section 3. Essentially, our algorithm provides solutions for read and writes requests in parallel disk systems and gives a relatively judicious decision in each scenario.

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 scarifying 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 scarifying 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 strategies. The first strategy is called flash strategy where only the flash memory is used to serve the requests. The second strategy is BUD strategy where only the buffer disks are used to serve the disk requests.

A. *Impacts of miss rate*

This experiment is focused on comparing the HYBUD strategy against the two other strategies described above. We study the impacts of miss ratio on the normalized energy consumption measured in joule. To achieve this goal, we increased the miss ratio of disk request from 75 to 100.

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.

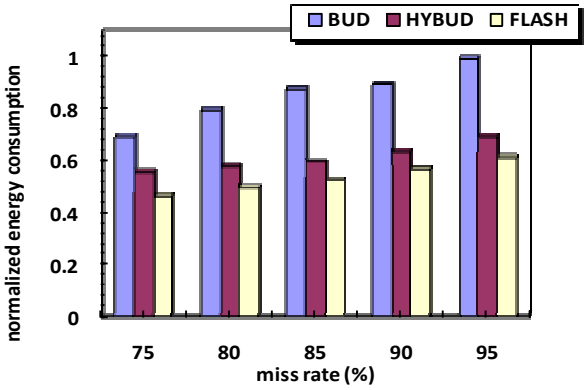

Fig.2 energy consumption versus miss ratio

B. *Impacts of Data size*

In this experiment we compared the three strategies in term of the size of data block. Fig. 3 illustrates the impact of data size over the energy consumption for the three strategies.

As the data size increases, the energy consumption for the three strategies decreases. This can be explained by the fact smaller data sizes decrease the time window in which a disk is able to sleep.

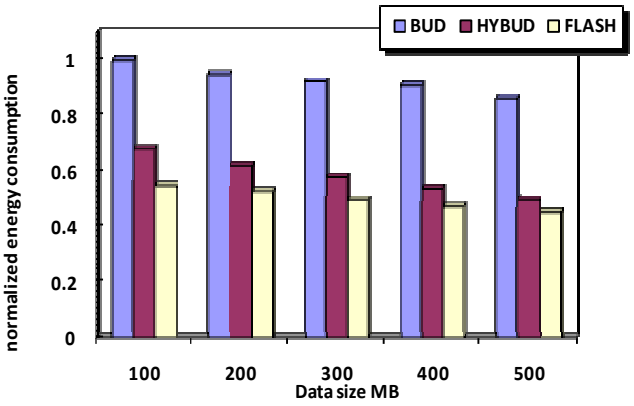

Fig.3 energy consumption versus data size

VII. Conclusion

Parallel disk systems play an important role in achieving high-performance for data-intensive applications, because the high parallelism and scalability of parallel disk systems can alleviate the disk I/O bottleneck problem. However, growing evidence shows that a substantial amount of energy is consumed by parallel disk systems in data centers. Although flash memory is energy-efficient compared to disk drives, flash memory is very expensive. Thus, it is not a cost-effective way to make use of large flash memory to build energy-efficient parallel disk systems. To address this problem, in this paper we proposed a hybrid disk architecture - HYBUD - that integrates a non-volatile flash memory with buffer disks to facilitate the development of cost-effective and energy-efficient parallel disk systems. The most popular data sets are cached in flash memory in HYBUD, whereas the second most popular data sets are cached in buffer disks. HYBUD improves parallel I/O energy efficiency because flash memory and buffer disks can serve a majority of incoming disk requests, thereby placing data disks in low-power state for increased period times. Furthermore, HYBUD makes energy-efficient parallel disks cost-effective, since a huge amount of popular data can be cached in buffer disks as well as the flash memory. We conducted experiments to quantitatively compare the HYBUD architecture with two existing non-hybrid architectures for parallel disk systems. Our experimental results show that HYBUD significantly reduce energy dissipation in cost-effective parallel I/O systems with buffer disks.

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. Douglis, P. Krishnan, and B. Marsh, "Thwarting the Power-Hunger Disk," *Proc. WinterUSENIX Conf.*, pp.292-306, 1994.

[5] T. Sumner and M. Marlino, "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. Douglis, 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.Douglis, and P. Krishnan," Flash Memory File Cashing for Mobile Computers" *Proc. the 27th Annual Hawaii International Conference on system sciences*, 1994.

[12] 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.

[13] S. Rajasekaran, "Selection algorithms for parallel disk systems," *Proc. Int'l Conf. High Performance Computing*, pp.343-350, Dec. 1998.

[14] 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.

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

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

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

[18] 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.

[19] 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.

[20] 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 Architecture*, pp. 169-179, June 2003.

[21] 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.

[22] Y. Hu and Q. Yang,"DCD-Disk Caching Disk: A New Approach for Boosting I/O Performance," *Proc. Int'l Symp. Computer Framework*, 1996.

[23] Toshiba America Electronic Componentd, Inc. (TAEC), www.toshiba.com/taec.Datasheet:TC58DVG02A1FT00, Jan 2003