Performance Evaluation of Energy-Efficient Parallel I/O Systems with Write Buffer Disks

Xiaojun Ruan[†], Adam Manzanares[†], Shu Yin[†], Ziliang Zong[‡], and Xiao Qin[†*]
[†] Department of Computer Science and Software Engineering
Auburn University
Auburn AL, 36849-5347
{ xzr0001, acm0008, szy0004, xqin}@auburn.edu
[‡] Department of Mathematics and Computer Science
South Dakota School of Mines and Technology
Rapid City, SD 57701
Ziliang.Zong@sdsmt.edu

***Abstract*-In the past decade, parallel disk systems have been developed to address the problem of I/O performance. A critical challenge with modern parallel I/O systems is that parallel disks consume a significant amount of energy in servers and high-performance computers. To conserve energy consumption in parallel I/O systems, one can immediately spin down disks when disk are idle; however, spinning down disks might not be able to produce energy savings due to penalties of spinning operations. Unlike powering up CPUs, spinning down and up disks need physical movements. Therefore, energy savings provided by spinning down operations must offset energy penalties of the disk spinning operations. To substantially reduce the penalties incurred by disk spinning operations, we developed a novel approach to conserving energy of parallel I/O systems with write buffer disks, which are used to accumulate small writes using a log file system. Data sets buffered in the log file system can be transferred to target data disks in a batch way. Thus, buffer disks aim to serve a majority of incoming write requests, attempting to reduce the large number of disk spinning operations by keeping data disks in standby for long period times. Interestingly, the write buffer disks not only can achieve high energy efficiency in parallel I/O systems, but also can shorten response times of write requests. To evaluate the performance and energy efficiency of our parallel I/O systems with buffer disks, we implemented a prototype using a cluster storage system as a testbed. Experimental results show that under light and moderate I/O load, buffer disks can be employed to significantly reduce energy dissipation in parallel I/O systems without adverse impacts on I/O performance.**

I. INTRODUCTION

In the past decade, large-scale storage systems have been developed to achieve high I/O performance and large storage capacity for a wide variety of data-intensive applications [13][6][10][7]. Much attention has been paid to the issues of performance and security in storage systems [21][19][2]. Making data disks active even when they are sitting idle is an important avenue to maintain high performance, because disks can immediately start serving newly arriving disk requests. This is a practical approach in some cases where there are high-end computing servers require extremely high I/O performance at the cost of high energy consumption. This approach, however, can waste a huge amount of energy in large-scale parallel disk systems; such a low energy-efficiency problem becomes more pronounced when there are many long idle periods. Traditional energy conservation techniques (e.g., dynamic power management) improve disk I/O energy efficiency by turning disks into the low-power state if the disks are sitting idle. Unfortunately, the conventional dynamic power management strategies for single disk systems are inadequate for parallel disk systems because of the following three reasons. (1) Idle periods under some workload conditions are too short to turn disks into a low-power state to conserve energy. (2) Although energy can be conserved by frequently place disks into the low-power state, an excessive number of power-state transitions inevitably have adverse impacts on the reliability of parallel I/O systems. (3) Numerous power-state transitions impose significant energy overhead as well as response time penalties. Making the traditional power management strategy energy effective largely depends on workload conditions. If workload of disk requests is relatively low and energy consumption of spinning operations is much lower, then it makes sense to apply the traditional power management strategy to aggressively spin down idle disks. Unfortunately, workloads of parallel I/O systems are usually high, leaving few opportunities for the dynamic power management strategies to conserve energy in many cases.

It is evident that the existing dynamic power management schemes ultimately encounter the problem of long power-state-transition times and noticeable power-state-transition energy overhead. Although disk active times in the parallel storage system can be shortened, energy dissipation in the storage system may not necessarily be reduced. This is because power-state transitions introduce a significant amount of energy overhead, which is referred to as penalties of spin-up and spin-down operations. Turning on a disk from the low-power mode does not only need to power the disk up, but also requires the disk to speed up its rotation speed - a physical movement, which consumes much more energy than electrical operations. In addition, if a new request arrives when the disk has been recently shut down, the new request has to wait for an unnecessary period of spin-up time. To remedy this deficiency, in this research we developed an innovative approach to significantly reducing unnecessary spinning operations while shortening response times in parallel I/O systems.

Recognizing that energy overhead and response-time penalties induced by power-state transitions negatively affect energy efficiency of parallel I/O, we seek to reduce the number of power-state transitions for writes issued to a parallel disk system. We focus on write requests, because there are a considerable number of write-intensive applications like transaction processing, log file updates, and data collection [19]. In this paper, we present the implementation of buffer-

* Corresponding Author. xqin@auburn.edu http://www.eng.auburn.edu/~xqin

0190-3918/09 $26.00 © 2009 IEEE
DOI 10.1109/ICPP.2009.10

disk-based parallel storage systems processing write requests. Specifically, we develop a dynamic request allocation algorithm for writes or DARAW, which dynamically and energy efficiently allocates buffer disks and data disks to serve write requests. Request allocations depend on not only data sets residing in buffer disks but also the power states of data disks. Data sets in buffer disks will be transferred to corresponding data disks when a set of conditions are satisfied. These conditions may be configured by system administrators to tune the performance of storage systems. Experimental results show that DARAW is conducive to conserving energy consumption in parallel storage systems without adverse impacts on I/O response times of write requests. Please note that our approach is that it can be readily applied to large-scale networked storage systems, where storage nodes are aggregated together into a larger cohesive storage system [4].

II. Related Work

Multi-speed Disks. Energy conservation techniques for disk systems have attracted much attention in the past few years. For example, energy dissipation in hard drives can be efficiently reduced by applying multi-speed disks because of very low power-state transition penalties. Song and Kandemir developed novel energy-aware compilers for multi-speed disks [18]. Although next-generation disks are likely to have multiple speeds, most disks utilized today are non-multi-speed disks. We believe that future generation multi-speed disks will be more expensive than conventional disks. Our proposed buffer-disk-based parallel I/O systems do not rely on multi-speed disks. In other words, the energy conservation technique investigated in this research is orthogonal with energy-efficient multi-speed disks. It is expected that further energy savings can be achieved by integrating our approach with the existing energy-saving techniques using multi-speed disks.

Cache Memory. Disk I/O performance has been substantially improved by applying cache [6]. The parallel I/O architecture considered in this study uses disks rather than cache memory as I/O buffer. Compared with cache, hard disks are slower and less energy efficient. However, disks are very cost effective and could buffer a whole lot more data than cache. Besides cost effective-ness, hard disks are non-volatile storage, meaning that buffered data are safe even after a power failure occurs in disks. A research for non-volatile caches is done by Gill and Modha [12]; the research focused on single disk, RAID-10 and RAID-5. It is possible to expand the research to energy-aware parallel storage systems.

Buffer Disks. To improve parallel disk buffer management, Kallahalla and Varman leveraged a shared buffer to improve I/O performance [8]. Goyal *et al.* explored the issue of quality of service in the context of storage system caches [4]. Rangaswami *et al.* investigated a way of employing disks to buffer data for streaming media servers in order to bridge the widening performance gap between dynamic random access memory and disk drives in the memory hierarchy [13]. The fundamental difference between our approach and the above three schemes is that the goal of our research is to reduce energy consumption in parallel I/O systems.

Dynamic Power Management. The traditional dynamic power management strategy is an efficient energy conservation technique for large idle time periods, which make it worthwhile to spin down disks when they are sitting idle. However, small and sequential data requests in modern scientific applications are very prevalent [7], making it less likely to observe large idle time intervals among requests. Moreover, small writes cause not only an energy consumption problem but also an I/O performance problem [1]. Hence, it is imperative to develop energy-saving techniques geared for writes, especially small writes, issued to parallel I/O systems.

III. Architecture and Algorithm for Buffer Disks

In this section, we first introduce a buffer-disk-based disk system architecture [17], which is energy-efficient in nature. Then, we present a dynamic request allocation algorithm for writes (or DARAW in short). Finally, we build an energy consumption model to quantify energy dissipation in parallel I/O systems.

A. Parallel I/O Systems with Buffer Disks

Compared with conventional parallel I/O systems, our disk architecture with buffer disks is energy efficient. Disks in this architecture are grouped into two categories - buffer disks and data disks. Write requests issued to a buffer-disk-based parallel I/O system are processed by buffer disks and; then buffered data sets are transferred back to target data disks in a batch manner.

Each disk, regardless of buffer or data disks, has its own queue to handle incoming requests. In addition, there is a central queue, from which all requests are dispatched to individual disks in the storage system. The number of buffer disks is, as a common case, less than the number of data disks, because our goal is to save energy by keeping a small number of active buffer disks while placing a large number of data disks into standby. The number of data disks and buffer disks (i.e., buffer/data disk ratio), of course, can largely affect the energy efficiency of the parallel I/O system. Ideally, the number of buffer/data disks can be adjusted on the fly in accordance with workload conditions. Thus, we evaluate in this study impacts of this buffer/data disk ratio on energy efficiency of parallel I/O systems.

B. The Algorithm for Buffer Disks

Now we describe the DARAW algorithm, which was designed in light of the novel disk architecture outlined in Section III(A). DARAW dynamically handles disk write requests without knowing disk access patterns in a priori. A buffer-disk-based parallel I/O system consists of a buffer-disk layer and a data disk layer. Thus, DARAW contains two components - a buffer-disk management scheme and a data-disk management scheme.

Given an incoming write request, the buffer-disk management scheme in DARAW has to choose the most

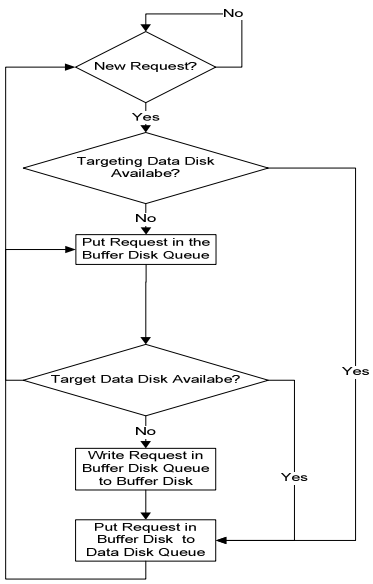

Figure 1 Buffer-disk management in DARAW.

appropriate buffer disk to serve the request. Decisions of choosing buffer disks largely depend on dynamically changing I/O workloads as well as the power states of data disks. For example, if a target data disk of a write request remains active, the request should be directly responded by the targeted data disk instead of a buffer disk. In doing so, unnecessary data transfer from buffer disks to data disks can be eliminated.

Fig. 1 outlines the buffer-disk management scheme in DARAW. A write request may be served by a buffer disk or the corresponding data disk. The buffer-disk management scheme directly allocates the request to its target data disk if it is active, thereby keeping the target disk in the active power state without spinning down the data disk until all dispatched requests are completed by the data disk. In this case, the requests targeting at active data disks can be written in the disks without being served by buffer disks. However, if a target data disk is standby, DARAW has to dispatch the request to a buffer disk. In case of redirecting a new write request to buffer disks, DARAW picks up a buffer disk containing a list of waiting requests targeting at the same data disk as that of the new request. This policy of choosing buffer disks aims to improve performance of transferring data from buffer disks back to data disks. In other words, buffering data with the same target disk into one buffer disk makes it possible to move the data back to the target disk in a batch manner. If no candidate buffer disk is identified, DARAW will pick a buffer disk with the lightest load. Note that I/O load of a buffer disk is quantified by the data amount to be buffered by the disk.

Note that incoming write requests are separated into two groups handled via different writing paths. In the first processing path, requests are served by buffer disks in which buffered data sets will be later on transferred to target data disks. Requests in the second processing path are directly handled by the data disk layer.

To facilitate the development of DARAW, in what follows we define an important scheduling-control parameter called *Sum of Requests in Buffers*, which is referred to as *SRB* throughout this paper. The *SRB* value of a data disk indicates the number of buffered requests targeting at this data disk. The *SRB* values help in keeping track of buffered data that need to be moved back to destination data disks. After a request is processed by a buffer disk, the corresponding *SRB* is increased by 1. When data is transferred back from to a data disk, the corresponding *SRB* is decreased. It is clear that requests going through the second path do not affect the *SRB* values, because these requests are not served by buffer disks. We set up a threshold value SRB_{th} for *SRBs* to initiate the data transfer process of moving data back to the destination hard disks. For example, if the *SRB* value of a data disk exceeds SRB_{th}, then DARAW has to spin up the data disk, to which buffered data starts being transferred from the buffer disk layer.

The SRB parameter plays a vital role in minimizing energy consumption of parallel disk systems, because *SRB* values track how many data sets have been buffered. It is evident that energy overhead incurred by power-state transitions may diminish energy conserved by placing disks into standby. DARAW can solve this problem by using SRB values to keep track of the number of buffered write requests, aiming to substantially reduce the number of unnecessary power transitions in data disks.

```
if a request comes from overall queue then
    if targeting data disk is not sleeping then
        write the request into targeting data disk
    else
        if buffer disk i having same targeting requests
            write the request in buffer disk i
        else write the lowest load buffer disk
        end if
    end if
end if
if more than 3 working buffer disks are blank then
    spin down
end if
for each data disk i in Parallel Storage System
    if SRB_i >= SRB_th then
        spin on data disk i
        write all requests targeting at i into disk i
        spin off data disk i
    end if
end for
```

Figure 2 The dynamic request processing algorithm for writes in DARAW.

Fig. 2 depicts the dynamic write request processing algorithm in DARAW. Fig. 2 shows that if the *SRB* value of a data disk is larger than SRB_{th}, DARAW writes all the buffered requests into the data disk in a batch manner after spinning up the data disk to the active state. A large SRB_{th} leads to a small number of power-state transitions, which ultimately results in lower energy consumption. Let SRB_i denote the *SRB* value of data disk i; let SRB_i^j be the number of requests targeting on

data disk i while being served by buffer disk j. SRB_i can be derived from SRB_i^j. In other words, SRB_i is the sum of SRB_i^j of all the buffer disks. Thus, we have

$$SRB_i = \sum_{j=1}^{n} SRB_i^j , \quad (1)$$

where n is the number of buffer disks. Note that each SRB value in a parallel I/O system is updated continuously.

C. Energy Consumption Analysis

In a process of building an energy consumption model, we consider two types of energy dissipation in parallel I/O systems with buffer disks. The first one is energy consumption of disks when they are either active or standby. The second type of energy dissipation is induced by power-state transitions, including disk spinning down and up. When a disk frequently transitions between the two power states, energy and time overhead may adversely affect energy savings. Such negative impacts are mainly contributed by spinning down and up disks with physical movements (e.g., spinning disks from 0 rotation per minute or RPM to 7200 or 15000 RPM).

Now we analyze the energy impact of DARAW on parallel disk systems. Total energy consumption in a parallel disk system is comprised of energy consumption caused by serving all requests in data disks and buffer disks, idle energy consumption of all data disks and buffer disks, and energy consumption of spinning up and down disks. We model seek time and rotational delay using average values, because both seek times and rotational delays are very small in our experiments. Note that similar approaches were used to model seek times and rotational delays for small disk requests [19].

Let e_{total} be the total energy dissipation in a parallel disk system. e_{total} can be expressed by Eq. (2) below:

$$e_{total} = e_A^B + e_I^B + e_S^B + e_A^D + e_I^D + e_S^D + e_O^D + e_O^D , \quad (2)$$

where $e_A{}^B$, $e_I{}^B$ and $e_S{}^B$ are energy consumption of buffer disks when they are active, idle, and standby; $e_A{}^D$, $e_I{}^D$ and $e_S{}^D$ are energy consumption of data disks in the active, idle, and standby modes; $e_O{}^B$ and $e_O{}^D$ are extra energy overhead experienced by buffer and data disks. The energy consumption of buffer and data disks when they are active can be written as:

$$\begin{aligned} e_A^B + e_A^D &= \sum_{i=1}^{m}\sum_{k=1}^{l}\left(x_{i,k}^B \cdot P_{A,i}^B \cdot T_{i,k}^B\right) + \sum_{j=1}^{n}\sum_{k=1}^{l}\left(x_{j,k}^D \cdot P_{A,j}^D \cdot T_{j,k}^D\right) \\ &= \sum_{i=1}^{m}\left(P_{A,i}^B \cdot \sum_{k=1}^{l}\left(x_{i,k}^B \cdot \left(t_{i,k}^{SK} + t_{i,k}^{RT} + \frac{s_k}{B_i^B}\right)\right)\right) \\ &\quad + \sum_{j=1}^{n}\left(P_{A,j}^D \cdot \sum_{k=1}^{l}\left(x_{j,k}^D \cdot \left(t_{j,k}^{SK} + t_{j,k}^{RT} + \frac{s_k}{B_j^D}\right)\right)\right), \end{aligned} \quad (3)$$

where element $x^B_{i,k}$ is "1" if request k is responded by buffer disk i and is "0", otherwise. Similarly, $x^D_{j,k}$ is "1" if request k is responded by data disk j and is "0", otherwise. $P^B_{A,i}$ and $P^D_{A,j}$ are the power of active buffer disk i and active data disk j. $T^B_{i,k}$ and $T^D_{j,k}$ are the service times of requests k on the ith buffer disk and the jth data disk. $T^B_{i,k}$ and $T^D_{j,k}$ are the summation of seek time (i.e., $t^{SK}_{i,k}$ and $t^{SK}_{j,k}$), rotational latency (i.e. $t^{RT}_{i,k}$ and $t^{RT}_{j,k}$), and data transfer time (i.e., s_k/B^B_i and s_k/B^D_j). The data transfer times s_k/B^B_i and s_k/B^D_j depend on the data size s_k and the transfer rate B^B_i and B^D_j of the disks. In a homogeneous parallel disk system where all the buffer and data disks are identical, Eq. (3) can be simplified as follows:

$$e_A^B + e_A^D = P_A \cdot \left(\sum_{i=1}^{m}\sum_{k=1}^{l}\left(x_{i,k}^B \cdot T_{i,k}^B\right) + \sum_{j=1}^{n}\sum_{k=1}^{l}\left(x_{j,k}^D \cdot T_{j,k}^D\right)\right). \quad (4)$$

where P_A is the power of active disks.

Idle energy consumption largely depends on idle time periods that can be derived from the serving time and the last request's finishing time. Eq. (5) below describes a way of quantifying energy consumption when disks are idle.

$$e_I^B + e_I^D = \sum_{i=1}^{m}\left(P_{I,i}^B \cdot T_{I,i}^B\right) + \sum_{j=1}^{n}\left(P_{I,j}^D \cdot T_{I,j}^D\right), \quad (5)$$

where $P^B_{I,i}$ and $P^D_{I,j}$ are the power of idle buffer disk i and idle data disk j. $T^B_{I,k}$ and $T^D_{I,k}$ are the total idle time periods on the ith buffer disk and the jth data disk, respectively. Suppose the number of idle periods on buffer disk i and data disk j is Ib_i and Id_j. Let $t^B_{I,i,1}$, …, $t^B_{I,i,bi}$ be the lengths of idle periods on buffer disk i. We denote $t^D_{I,j,1}$, …, $t^D_{I,j,dj}$ as the lengths of idle periods on data disk j. Thus, Eq. (5) can be rewritten as:

$$e_I^B + e_I^D = \sum_{i=1}^{m}\left(P_{I,i}^B \cdot \sum_{k=1}^{Ib_i} t_{I,i,k}^B\right) + \sum_{j=1}^{n}\left(P_{I,j}^D \cdot \sum_{k=1}^{Id_i} t_{I,j,k}^D\right). \quad (6)$$

Similarly, we represent the power of standby buffer disk i and data disk j as $P^B_{S,i}$ and $P^D_{S,j}$. In addition, we denote the idle period intervals on buffer disk i as $t^B_{S,i,1}$, …, $t^B_{S,i,Sbi}$; the lengths of idle periods on data disk j as $t^D_{S,j,1}$, …, $t^D_{S,j,Sdj}$. Then, the energy consumption when disks are placed into standby can be expressed as:

$$e_S^B + e_S^D = \sum_{i=1}^{m}\left(P_{S,i}^B \cdot \sum_{k=1}^{Sb_i} t_{S,i,k}^B\right) + \sum_{j=1}^{n}\left(P_{S,j}^D \cdot \sum_{k=1}^{Sd_i} t_{S,j,k}^D\right). \quad (7)$$

Now we calculate energy overheads caused by power-state transitions using Eqs. (8) and (9), where $y_{k,B,U,i}$ and $y_{k,B,D,i}$ are the number of disk spinning ups and downs in buffer disks; $z_{k,D,U,j}$, and $z_{k,D,D,j}$ are the number of disk spinning ups and downs in data disks. $E_{k,B,U,i}$, $E_{k,B,D,i}$, $E_{k,B,U,j}$, and $E_{k,D,D,j}$ are the energy consumed by each power-state transition.

$$\begin{aligned} e_O^B &= \sum_{i=1}^{m}\sum_{k=1}^{l}\left(y_{k,B,U,i} \cdot E_{P,B,U,i}\right) \\ &\quad + \sum_{i=1}^{m}\sum_{k=1}^{l}\left(y_{k,B,D,i} \cdot E_{P,B,D,i}\right), \end{aligned} \quad (8)$$

$$\begin{aligned} e_O^D &= \sum_{j=1}^{n}\sum_{k=1}^{l}\left(z_{k,D,U,j} \cdot E_{P,D,U,j}\right) \\ &\quad + \sum_{j=1}^{n}\sum_{k=1}^{l}\left(z_{k,D,D,j} \cdot E_{P,D,D,j}\right). \end{aligned} \quad (9)$$

TABLE I IBM 36Z15 ULTRASTAR

System Parameter.	Values
Rotations Per Minute	10000 RPM
Working Power	13.5 W
Standby Power	2.5 W
Spin up Energy	135 Joule
Spin down Energy	13 Joule
Spin up Time	10.9 sec
Spin Down Time	1.5 sec
Transfer Rate	52.8 MB/s

TABLE II IBM 40GNX TRAVALSTAR

System Parameter.	Values
Rotations Per Minute	5400 RPM
Working Power	3 W
Standby Power	0.25 W
Spin up Energy	8.7 Joule
Spin down Energy	0.4 Joule
Spin up Time	3.5 sec
Spin Down Time	0.5 sec
Transfer Rate	25 MB/s

IV. Performance Evaluation

To evaluate the performance of DARAW, we conducted extensive simulation experiments using various disk I/O traces representing real-world workload conditions with small writes. The trace file used in our simulation contains several important parameters such as arrival time, data size, cylinder number, targeting data disk, and arrival time.

Simulator Validation: We used synthetic I/O traces and real-world traces to validate the simulator against a prototype cluster storage system with 12 disks. The energy consumed by the storage system prototype matches closely (within 4 to 13%) to that of the simulated parallel disk system. The validation process gives us confidence that we can customize the simulator to evaluate intriguing energy-efficiency trends in parallel I/O systems by gradually changing system parameters.

For comparison purpose, we consider a baseline algorithm based on a parallel I/O system without the buffer-disks layer. This baseline algorithm attempts to spin up standby target disks upon the arrival of a request. Additionally, the baseline algorithm makes an effort to immediately spin down a disk after it is sitting idle for a period of time. Tables I and II summarize the parameters of two real-world disks (IBM 36z15 Ultrastar and IBM 40GNX Travalstar) simulated in our experiments.

Fig. 3 plots energy efficiency and performance of the baseline algorithm applied to a traditional parallel I/O system without buffer disks. Results plotted in Fig. 3(a) show that the I/O load increases significantly as the arrival rate (i.e., λ) grows. For example, 1000 requests are issued to the simulated parallel I/O system within 100,000 milliseconds if λ is set to 0.01No./ms., whereas 1000 requests have arrived in the system within 50,000 milliseconds when λ is doubled.

An interesting counterintuitive observation drawn from Fig. 3(b) is that with respect to the baseline algorithm, the average response time of the high-performance disk (IBM 36z16 Ultrastar) is noticeably longer than that of the IBM 40GNX Travalstar - a low-performance disk. The rationale behind this observation is that the spin-up and spin-down time of IBM Ultrastar is much higher than those of IBM Travalstar. Thus, the overhead incurred by spin-up and spin-down in IBM Ultrastar is more expensive than in IBM Travalstar. Our traces contain a large number of small writes coupled with numerous small idle periods and; therefore, the overhead caused by disk spin up and spin down are even higher than I/O processing

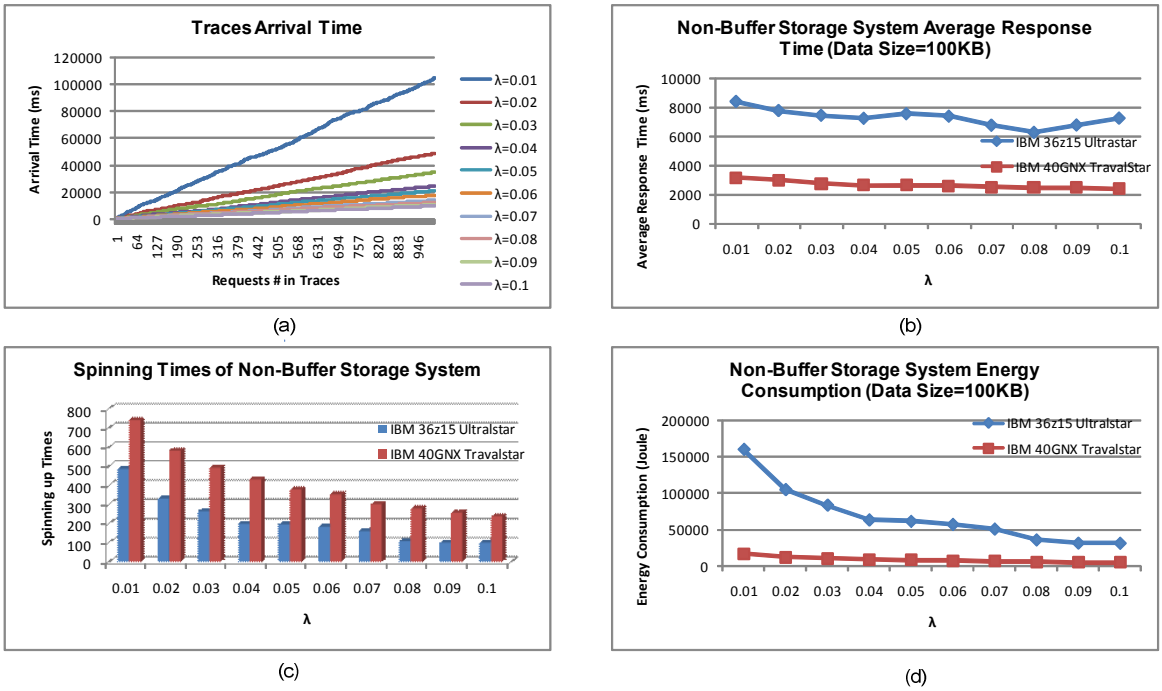

Figure 3 Energy efficiency and performance of the baseline algorithm based on a parallel I/O system without buffer disks.

times. In other words, the overhead of spin-ups and spin-downs dominates the average response time of disk requests in the parallel storage system.

Fig. 3(c) shows that the total spin-up times of the Ultrastar disks is smaller than those of the Travalstar disks. We attribute this trend to the fact that the spin-up delay of the IBM Ultrastar disks is much longer than that of the Travalstar disks. Compared with Travalstar, an Ultrastar disk is more likely to serve another request during the time between a spin-up and a consecutive spin-down. As the request arrival rate λ increases, the average inter-arrival time between two continuous requests decreases. In other words, the increasing I/O load gives rise to the decreasing number of spin-ups and spin-downs. Such a trend is apparent for both the IBM Ultrastar and Travalstar disks, because high I/O load can reduce the number of idle time periods, which in turn diminishes opportunities of spinning down disks to conserve energy.

Fig. 3(d) depicts the energy consumption trend for the IBM Ultrastar and Travalstar disks. In what follows, we describe two important observations. First, Fig. 3(d) reveals that under the same workload conditions, the overall energy consumption of Ultrastar is higher than that of Travalstar. The Ultrastar disks consume more energy, because compared with Travalstar, Ultrastar not only has higher active and standby power but also has higher spin-up and spin-down energy.. Second, when the request arrival rate λ increases (i.e. heavy workload), the energy consumption is reduced for both Ultrastar and Travalstar. The energy dissipation in the parallel disk system can be minimized by a high I/O load, because a high arrival rate results in low spin-up and spin-down overhead (see Fig. 3(c)). It is worth noting that in each experiment, we fix the total number of requests (e.g, 1000).

Fig. 4 below illustrates the energy consumption, average response time, total number of spin-ups, and energy saving rate of a buffer-disk-based parallel I/O system using DARAW. Since Fig. 3 demonstrates that IBM 40GNX Travalstar disks are more energy efficient than IBM 36Z15 Ultrastar, in this set of experiments we focused on the Travalstar disks. We observe from Figs. 4(a) and 4(d) that when λ is small, the DARAW algorithm can significant conserve energy. For example, when the workload is low, DARAW can save 50%-60% on energy consumption in the parallel disk systems with Travalstar. However, the energy conservation rate is dropping when the arrival rate is getting larger, meaning that there is not much space for energy conservation under high I/O load. This is simply because there are fewer idle periods that allow disks to be switched to standby when the workload is very heavy.

Fig. 4 (c) clearly shows that DARAW slightly reduce the number of spin-ups and spin-downs. The entire parallel I/O system can benefit from the decreased number of power-state transitions, since power-state transitions not only consume energy but also degrade I/O performance. Figs. 4(a) and 4(b) show energy and performance impact of the number of buffer disks. Results plotted in Fig. 4(b) indicate that the increasing the number of buffer disks can boost aggregated I/O bandwidth, which in turn reduces the average response time. On the other hand, Fig. 4(a) shows that the energy consumption increases due to the large number of buffer disks. There exists a tradeoff problem between performance and energy conservation. A small number of buffer disks can merely provide a low I/O bandwidth; a large number of buffer disks ultimately consume more energy. In case of light

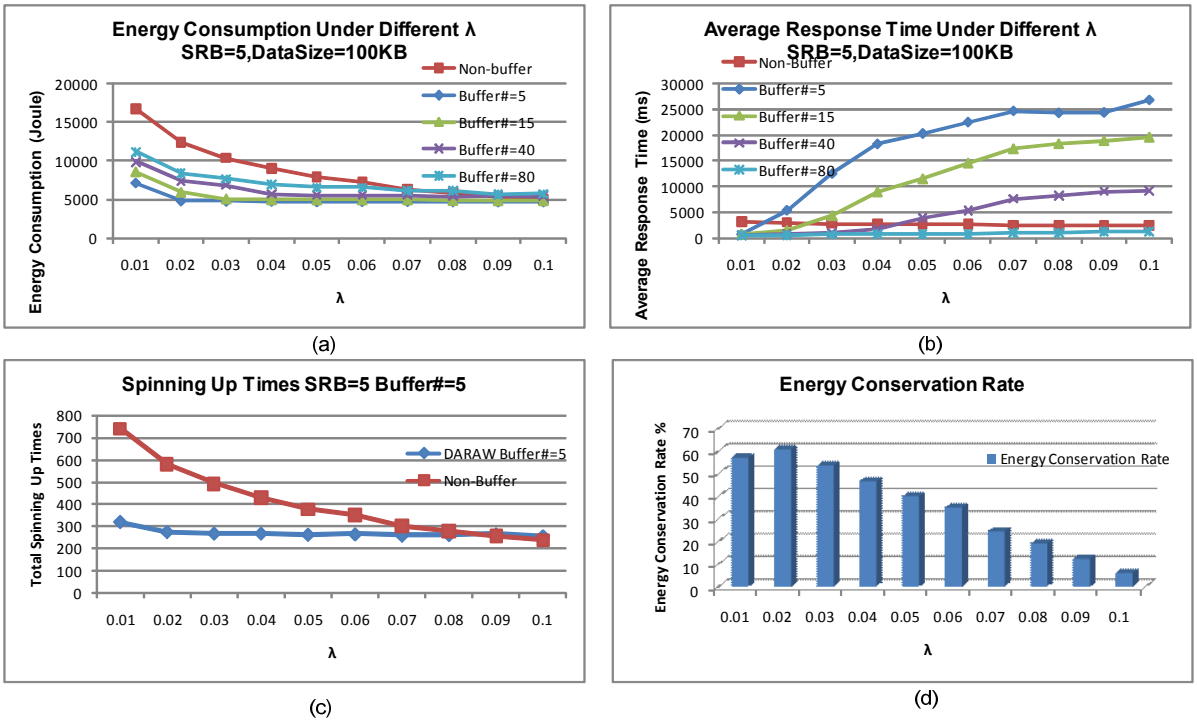

Figure 4 Energy efficiency and performance of DARAW. Buffer disks and Data Disks are IBM 40GNX Travalstar.

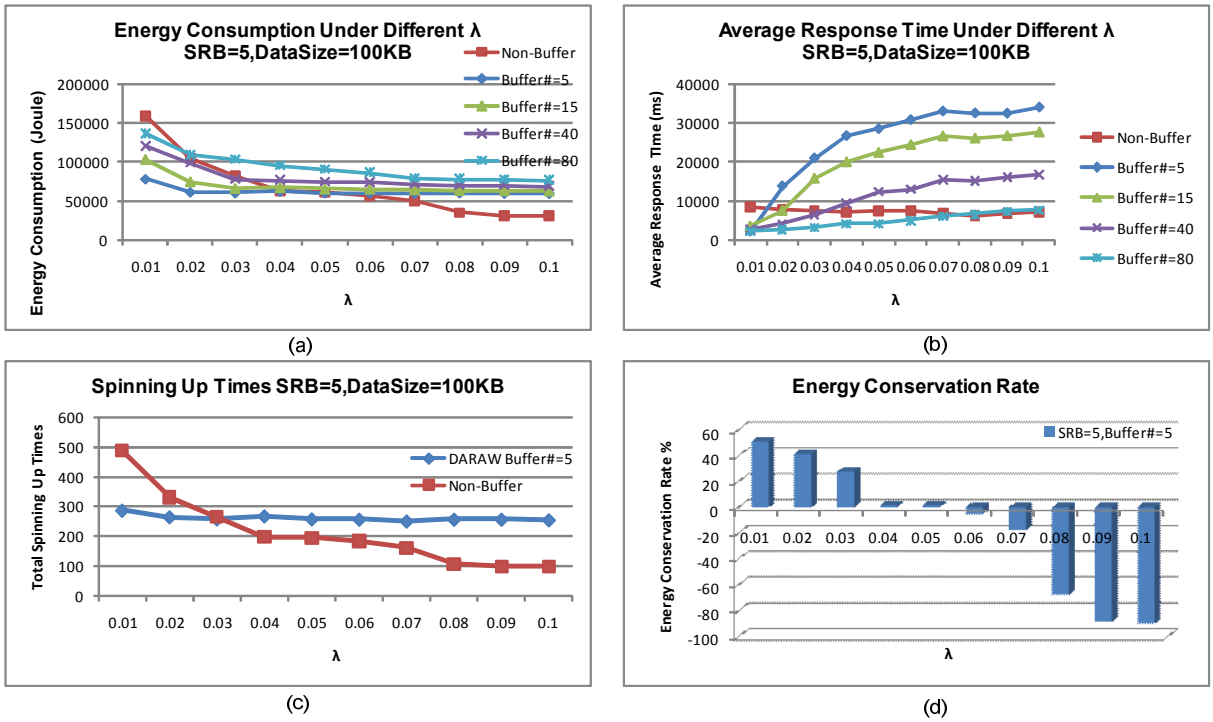

Figure 5 Energy efficiency and performance of DARAW. Buffer disks and Data Disks are IBM 36z15 Ultrastar.

workload (e.g., λ = 0.01), DARAW can significantly improve energy efficiency and performance measured in average response time. The overall performance tends to degrade when workload is high and the number of buffer disks is small.

Now we are positioned to evaluate energy-efficiency and performance impact of DARAW on IBM 36z15 Ultrastar - high performance disks. Energy consumption trends plotted in Fig. 5(a) are very close to those in Fig. 4(a). The energy consumption trend goes up even more dramatically in Fig. 5(a), because the spin-up and spin-down overhead of IBM 36z15 Ultrastar is higher than that of the IBM 40GNX Travalstar. As for low I/O load (e.g., λ = 0.01), DARAW is capable of conserving energy by 48% compared with the traditional strategy. Unfortunately, when the request arrival rate is relatively high, the energy efficiency of DARAW is lower than that of the baseline algorithm. In addition, the average response time and spinning up times are also reduced. Fig. 5(b) shows that if DARAW is applied to parallel I/O systems under high I/O workload, it is not a wise idea to maintain a large number of buffer disks without adversely affecting energy efficiency. With a fixed number of total disks, increasing the number of buffer disks can potentially incur high energy cost under relatively high I/O load. In contrast, when the workload is very light (e.g., λ = 0.01), DARAW can substantially reduce response time regardless of the number of buffer disks. The implication of these results is that DARAW is conducive to improving both energy efficiency and performance under light or moderate I/O load conditions.

In the last set of experiments, we let IBM Ultrastar perform as buffer disks and IBM Travalstar serve as data disks. Since the buffer-disk layer is likely to become the performance bottleneck in DARAW, we attempt to solve the bottleneck problem using high-performance hard drives as opposed to increasing the number of low-performance disks. Note that the Ultrastar disks consume more power than the Travalstar disks. Hence, we simply add a small number buffer disks using Ultrastar to maintain energy efficiency at a high level. Fig. 6(b) shows that the average response time is shortened by 62% after adding one Ultrastar buffer disk; whereas Fig. 6(c) confirms that the number of spin-ups is decreased by 66%. Figs. 6(a) and 6(d) show that DARAW reduce the energy by up to 60%.

V. Conclusion

In this study, we developed an algorithm - dynamic request allocation algorithm for writes or DARAW - to conserve energy dissipation in parallel I/O systems serving write requests. DARAW is conducive to improving parallel I/O energy efficiency by the virtue of employing a small number of buffer disks to serve a majority of write requests, thereby placing a large number of data disks in standby for long period of times. DARAW keeps track of the number of buffered requests targeting at each data disk. In doing so, DARAW can efficiently move buffered data sets to destination data disks in a batch manner. DARAW achieves both high energy-efficiency and performance under light and moderate I/O load. Under high I/O load, the performance of DARAW can be improved by either substituting high-performance hard drives for low-performance buffer disks or adding more buffer disks at the cost of energy. To quantify the energy efficiency and performance of DARAW, we implemented a prototype using a

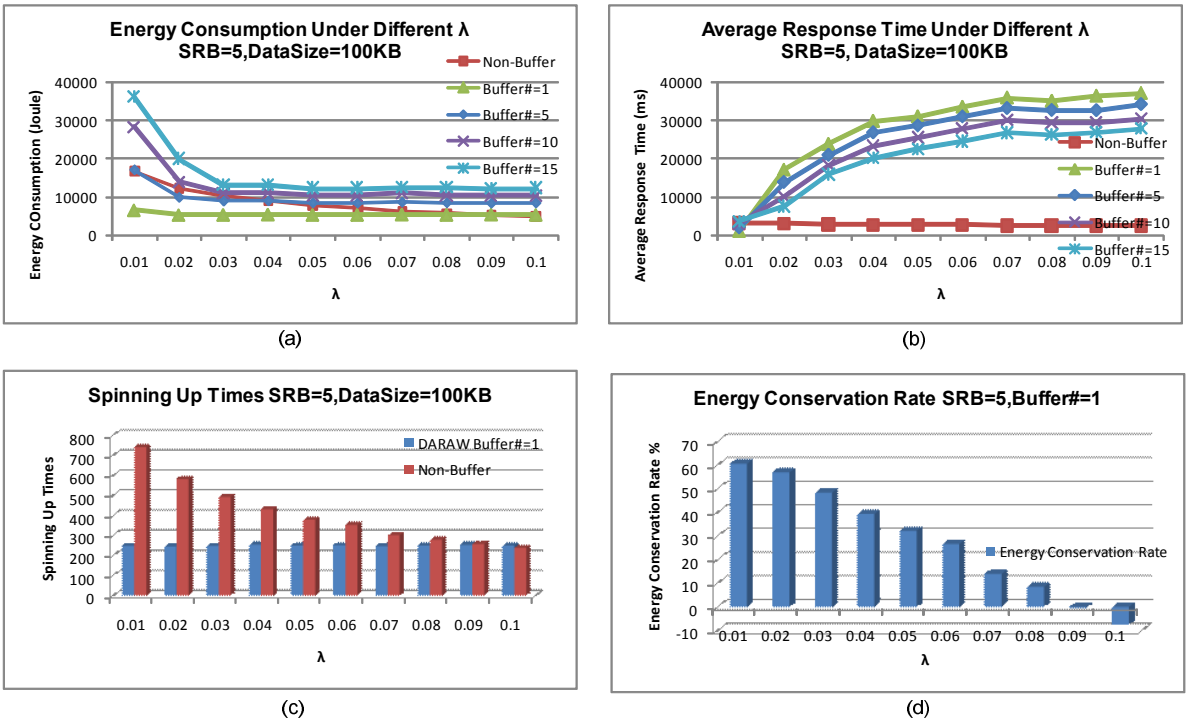

Figure 6 Energy efficiency and performance of DARAW. Buffer disks are IBM 36z15 Ultrastar; Data Disks are IBM 40GNX Travalstar.

cluster storage system as a testbed. Experimental results show that under light and moderate I/O workload conditions, DARAW can significantly reduce energy dissipation in parallel I/O systems without adverse impacts on I/O performance.

As a future direction of this study, we intend to develop a reliability model for parallel I/O systems with buffer disks. The reliability model will allow us to make tradeoffs between energy efficiency and reliability in parallel I/O systems.

ACKNOWLEDGMENT

The work reported in this paper was supported by the US National Science Foundation under Grants CCF-0845257 (CAREER), CNS-0757778 (CSR), CCF-0742187 (CPA), CNS-0831502 (CyberTrust), OCI-0753305 (CI-TEAM), DUE-0837341 (CCLI), and DUE-0830831 (SFS), as well as Auburn University under a startup grant and a gift (Number 2005-04-070) from the Intel Corporation.

REFERENCES

[1] S. H. Baek and K. H. Park, "Matrix-Stripe-Cache-Based Contiguity Transform for Fragmented Writes in RAID-5," *IEEE Trans. Comp., vol. 56, no. 8, pp. 1040-1054,* 2007.

[2] R. Barbe, M. Kallahalla, P. Varman and J.S. Vitter, "Competitive Parallel Disk Prefetching and Buffer Management," *Proc. IOPADS*, 1997.

[3] E. V. Carrera, E. Pinheiro, and Ricardo Bianchini, "Conserving Disk Energy in Network Servers," *Proc. Int'l Conf. Supercomputing,* 2003

[4] J. C. Chuang and M. A. Sirbu, "Distributed Network Storage Service with Quality-of Service Guarantees," *Proc. Conf. Internet Society INET'99,* June 1999.

[5] P. Goyal, D. Jadav, D. S. Modha, and R. Tewari, "CacheCOW: Qos for Storage System Caches," *Proc. Int'l Workshop QoS,* 2003

[6] B. C. Forney, A. C. Arpaci-Dusseau and R. H. Arpaci-Dusseau, "Storage-Aware Caching: Revisiting Caching," *Proc. Int'l Conf. File and Storage Technologies,* 2002.

[7] D. Hildebrand, L. Ward and P. Honeyman, "Large Files, Small Writes, and pNFS," *Proc. Int'l Conf. Supercomputing, 2006*.

[8] M. Kallahalla and P. J.Varman, "Improving Parallel-Disk Buffer Management using Randomized Writeback," *Proc. Int'l Conf. Parallel Proc.,* 1998.

[9] S. Lakshmanan, M. Ahamad, and H. Venkateswaran, "Responsive Security for Stored Data," *IEEE Trans. Parallel and Distr. Sys.,* vol. 14, no. 9, Sep. 2003.

[10] M. I. Lutwyche and M. Despont, et al, "Highly Parallel Data Storage System Based on Scanning Probe Arrays," *American Inst. Physics,* 2000.

[11] C. Ruemmler and J. Wilkes, "An Introduction to Disk Drive Modeling," *IEEE Computer*, Mar. 1994.

[12] B. S. Gill and D. S. Modha, "WOW: Wise Ordering for Writes-Combining Spatial and Temporal Locality in Non-Volatile Caches," *Proc. FAST, 2005.*

[13] M. Nijim, X. Qin, and T. Xie, "Modeling and Improving Security of a Local Disk System for Write-Intensive Workloads," *ACM Trans. Storage*, vol. 2, no. 4, pp. 400-423, Nov. 2006.

[14] X. Qin, "Performance Comparisons of Load Balancing Algorithms for I/O-Intensive Workloads on Clusters," *J. Network and Comp. App.,* vol.31, no.1, pp. 32-46, 2008.

[15] R. Rangaswami, Z. Dimitrijevic, E. Chang and K. Schauser, "MEMS-based Disk Buffer for Streaming Media Servers," *Proc. Int'l Conf. Data Eng.,* 2003.

[16] A. L. N. Reddy, J. Wyllie and K.B. R. Wijayaratne, "Disk Scheduling in a Multimedia I/O System," *ACM Trans. Multimedia Comp., Comm. and App., vol. 1, no. 1,* 2005.

[17] X.–J. Ruan, A. Manzanares, K. Bellam, X. Qin, and Z. –L. Zong, "DARAW: A New Write Buffer to Improve Parallel I/O Energy-Efficiency," Proc. 24th Annual ACM Symp. Applied Comp., Mar. 2009.

[18] S. W. Song, M. Kandemir, and A. Choudhary, "Software-directed disk power management for scientific applications," *Proc. IPDPS,* April 2005.

[19] W. Susilo, F. Zhang and Y. Mu, "Privacy-Enhanced Internet Storage," *Proc. AINA,* 2005.

[20] R. Wijayaratne and A. L. Narasimha Reddy, "Integrated QOS management for Disk I/O," *Proc. IEEE Int'l Conf. Multimedia Comp. and Sys.,* 1999.

[21] H. Zhang, W. Wu, X. Dong, D. Qian and L. Dai, "A Study on Data Placement of Extensible Parallel Storage System," *Proc. Int'l Conf. Computer and Information Science,* 2007.

[22] Z. Zong, M. Briggs, N. O'Connor, and X. Qin, "An Energy-Efficient Framework for Large-Scale Parallel Storage Systems," *Proc. Int'l Conf. Parallel and Distributed Processing Symp.,* Mar. 2007.