ECOS: An Energy-Efficient Cluster Storage System

Xiaojun Ruan, Shu Yin, Adam Manzanares, Jiong Xie, Zhiyang Ding,
James Majors, Xiao Qin†
Computer Science and Software Engineering
Auburn University, Auburn, AL 36849
Email: {xzr0001, szy0004, acm0008, jzx0009, dingzhi, majorjh, xqin}@eng.auburn.edu

***Abstract*—*Cluster storage systems are essential building blocks for many high-end computing infrastructures. Although energy conservation techniques have been intensively studied in the context of clusters and disk arrays, improving energy efficiency of cluster storage systems remains an open issue. To address this problem, we describe in this paper an approach to implementing an energy-efficient cluster storage system or ECOS for short. ECOS relies on the architecture of cluster storage systems in which each I/O node manages multiple disks - one buffer disk and several data disks. Given an I/O node, the key idea behind ECOS is to redirect disk requests from data disks to the buffer disk. To balance I/O load among I/O nodes, ECOS might redirect requests from one I/O node into the others. Redirecting requests is a driving force of energy saving, and the reason is two-fold. First, ECOS makes an effort to keep buffer disks active while placing data disks into standby in a long time period to conserve energy. Second, ECOS reduces the number of disk spin downs/ups in I/O nodes. The idea of ECOS was implemented in a Linux cluster, where each I/O node contains one buffer disk and two data disks. Experimental results show that ECOS improves the energy efficiency of traditional cluster storage systems where buffer disks are not employed. Adding one extra buffer disk into each I/O node seemingly has negative impact on energy saving. Interestingly, our results indicate that ECOS equipped with extra buffer disks is more energy efficient than the same cluster storage system without the buffer disks. The implication of the experiments is that using existing data disks in I/O nodes to perform as buffer disks can achieve even higher energy efficiency.*

I. Introduction

Cluster storage systems - essential building blocks in many high-performance computers - have been widely adopted to support data-intensive applications running on high-performance computing platforms. Optimizing energy consumption in cluster storage systems has strong impacts on the cost of backup power-generation and cooling equipment in cost-effective cluster computing infrastructures. We were motivated to address the energy saving issues in cluster storage systems, because a significant fraction of the operation cost of data centers is due to energy consumption in storage systems. For example, the average power consumption of TOP 10 supercomputing systems is 1.32 Mwatt, in which a large portion is contributed by storage systems [3]. Dell Texas Data Center reported that 37 percent of the energy consumed by supercomputers is cost by storage systems [2]. In addition to emerging high-performance disk drives with high power needs, increasing storage requirements imposed by data-intensive applications make it desirable to design energy-efficient cluster storage systems. Several novel techniques proposed to conserve energy in storage systems include dynamic power management schemes [7] [17], power-aware cache management strategies [31], power-aware prefetching schemes [24], software-directed power management techniques [25], redundancy techniques [22], and multi-speed settings [10] [11] [15]. A few innovative techniques have been developed to substantially reduce energy dissipation in traditional server clusters [21] [13] [5] [4] [8] [9]. However, the research on the improvement of energy efficiency in cluster storage systems is still in its infancy. It is imperative to develop new cluster storage systems that can exhibit high energy efficiency and I/O performance for high-end data-intensive computing.

In this paper, we detail an approach to implementing an energy-efficient cluster storage system called ECOS. To achieve high aggregate I/O bandwidth under heavey workloads, we design a cluster storage system where each I/O node embraces multiple disks - one buffer disk and several data disks. The basic idea behind ECOS is to redirect disk requests from data disks to buffer disks within I/O nodes. Redirecting requests to buffer disks is a driving force of energy saving, because I/O load is skewed toward buffer disks so that data disks can be placed into standby in a long time period to conserve energy. Spinning down/up disks inevitably introduce extra energy overhead. As such, adding a buffer disk in each I/O node aims to reduce the number of disk spin downs/ups. To balance I/O load among I/O nodes, ECOS attempts to redirect disk requests from a heavily loaded I/O node into other I/O nodes with light load.

One of our recent studies was focused on an algorithm - DARAW - handling writes in parallel storage systems with buffer disk [23]. Having been extended to deal with writes in the context of cluster storage systems, the DARAW algorithm was implemented as a core component in the ECOS system. When it comes to large write requests (e.g., larger than 500MB), data should be be issued directly to data disks. In contrast, small write requests have to be sent to an active buffer disk. Once the data of a write request is transferred to buffer or data disks, an acknowledgement is returned to an application that issued the request.

When a buffer disk in an I/O node is overloaded, then the corresponding data disks within the I/O node need to be spinned up to balance I/O accesses. The challenging issue in this component of the research is to determine the optimal number of standby data disks to be activated in respond to high I/O traffic. The goal is to spin up as few data disks as possible,

978-1-4244-5736-6/09/$26.00 ©2009 IEEE

keeping the utilization of each disk below 100 percent.

MAID [6], PDC [20] [28], and BUD [23] [18] - four existing energy-efficient parallel disk systems - are conducive to achieving high energy efficiency with a small fraction of I/O delays. Our ECOS system is fundamentally different from these parallel storage systems, because ECOS is a cluster storage system with loosely-coupled parallel disks cross multiple I/O nodes whereas the other four systems contain tightly-coupled parallel disks (e.g., disk arrays).

Compared with other disk energy conservation techniques, the ECOS cluster storage system has the following three unique features.

• First of all, it has no need to modify data-intensive applications when they are ported from traditional cluster storage systems to ECOS.

• Second, it is not necessary to add extra hardware such as flash drives into cluster storage systems.

• Third, ECOS maintains an acceptable level of I/O performance by the virtue of parallel buffer disks across multiple I/O nodes.

A prototype of ECOS was implemented in a Linux cluster, where each I/O node contains one buffer disk and two data disks. The power manager in ECOS relies on a system call in the Linux kernel to spin down and spin up disk drives. Experimental results show that ECOS improves the energy efficiency of traditional cluster storage systems without using buffer disks. Adding one extra buffer disk into each I/O node seemingly has negative impact on energy saving. Interestingly, our results indicate that ECOS equipped with extra buffer disks is more energy efficient than the same cluster storage system without the buffer disks. The implication of the experiments is that using existing data disks in I/O nodes to perform as buffer disks can achieve even higher energy efficiency.

In summary, the main contributions of this study are:

• We designed an energy-efficient disk architecture to reduce energy dissipation in cluster storage disk systems;

• We developed a disk power model for cluster storage systems; and

• We implemented an energy-efficient cluster storage system that consists of modules like disk request processing, data movement, data replacement, and power management for I/O nodes.

The remainder of the paper is organized as follows. Section II describes related work. After the presentation of an energy-efficient architecture for cluster storage systems, Section III details our evaluation methodology and a testbed used to implement ECOS. Section IV presents experimental results. Finally, Section V concludes this paper with future research directions.

II. Related Work

An array of techniques were developed to save energy in disk storage systems. Energy consumption of single disks can be reduced at either I/O level (e.g., dynamic power management [7] [17] and multi-speed disks [10] [11] [15]) or operating system level (e.g., power-aware cache management strategies [31], power-aware prefetching schemes [24]). Apart from energy-saving techniques at the levels of I/O and operating systems, energy efficiency can be optimized at the application level [25] [29]. For example, Weiel et al. developed an I/O semantics called Cooperative I/O for energy-aware applications. The design of our ECOS is orthogonal to the aforementioned schemes. therefore, incorporating these techniques in ECOS can ultimately improve the energy efficiency of cluster storage systems.

Buffer management has been widely used to boost performance of parallel disk systems [1] [27]. Previous studies showed that data buffers significantly reduce the number of disk accesses in parallel disk systems [30]. More importantly, it is observed from the previous studies that traffic of small reads and writes becomes a performance bottleneck of disk systems, especially when RAM sizes for data buffers are increased rapidly [30]. It is expected that small disk requests dominate energy dissipation in cluster storage systems supporting data-intensive applications like remote-sensing applications and on-line transaction processing systems [16] [26]. Our approach differs itself from the traditional buffer management schemes in the sense that the goal of ECOS is to leverage buffer disks to reduce energy dissipation in cluster storage systems.

Colarelli and Grunwald proposed the Massive Array of Idle Disks (MAID) as a replacement for old tape backup archives with hundreds or even thousands of tapes [6]. It is observed that only a small part of the archive would be active at a time, the idea behind MAID is to copy the required data to a set of cache disks while placing all the other disks in the standby mode to conserve energy. I/O accesses to the archive may retrieve data from the cache disks rather than from standby disks. Pinheiro and Bianchini designed a Popular Data Concentration (PDC) scheme to reduce energy consumption in a network server by skewing I/O load toward a few of all the disks in the server [20]. The design of PDC is based on an observation that network server workloads in many cases exhibit files with widely different popularities (e.g., Web server workloads exhibit highly skewed popularity towards a small set of files.). Taking into account the skewed data popularity, PDC dynamically migrates frequently accessed data to a subset of disks in a disk array. However, the bandwidth of disks limits the performance of PDC system. The reason is that PDC moves popular data to one disk which means the workload is extremely unbalanced. We observed from our experiments that data migration overhead is non-negligible in a highly dynamic I/O workloads [23] [18]. In addition, the popular data disk has a strong likelihood to become I/O bottleneck. ECOS attempts to balance I/O load by evenly distributing popular data among multiple buffer disks. It is worth noting that both MAID and PDC are focused on energy efficiency issues in tightly-coupled parallel disks like disk arrays. Unlike MIAD and PDC, ECOS is an energy-efficient cluster storage system where I/O nodes are loosely connected to provide high aggregate I/O bandwidth.

Flash drives can be employed to buffer and cache popular data for I/O nodes in clusters [12] [14]. Flash-drive-based cluster storage systems are likely to be the most energy efficient storage systems for cluster computing infrastructures because flash drives have a very low power consumption compared to hard drives. However, due to limited access times, the reliability of flash drives will have to be addressed when one integrates flash drives into cluster storage systems. A recent study shows that intensive and dynamic accesses for a long period of time can substantially shorten the lifetime of a flash drive [19].

III. Design and Implimentation of ECOS

A cluster storage system is comprised of an array of I/O nodes connected by a high-speed network. In recent years, most research efforts on reducing energy consumption in parallel disk systems. However, the issue of using buffer-disk architectures to reduce energy consumption in cluster storage systems has not investigated well. Our long-term goal is to develop fundamental techniques to save energy of large-scale cluster storage systems. The objective of this study, which is paving a way towards that goal, is to design and implement the ECOS system - an energy-efficient cluster storage system in which disk request processing, data movement/placement strategies, power management, and prefetching schemes are holistically integrated to save energy. The rationale for this study is that the development of ECOS will promote more energy-efficient resource management techniques for storage systems in general and cluster storage systems in particular. In this section, we first detail design issues including the system architecture and hardware configuration of the ECOS storage system. Then, we describe various implementation issues in ECOS.

The architecture of ECOS (see Fig. 1) is an extension of the architecture of traditional cluster storage systems, where each I/O node manages one local disk. Like the traditional cluster storage systems, ECOS has large files striped across a number of I/O nodes connected through a high-speed network. Since each I/O node in ECOS contains a buffer disk and multiple data disks, files might be distributed across a number of disks within one I/O node. All the compute nodes in the system can directly access I/O nodes through the nextwork.

Disk I/O parallelisms can be provided in forms of inter-request and intra-request parallelism. Inter-request parallelism allows multiple independent requests to be served simultaneously by multiple I/O nodes in ECOS, whereas intra-request parallelism enables a single disk request to be processed by multiple I/O nodes in parallel. A parallelism degree of a data request is the number of I/O nodes to which the requested data is striped. In the design of ECOS, we consider two different configurations to deal with inter-request and intra-request I/O parallelisms, respectively.

The first ECOS configuration - aiming to support inter-request parallelisms - consists of four major components: a RAM buffer residing in compute nodes, m buffer disks, n data disks, and an energy-aware buffer-disk controller. The

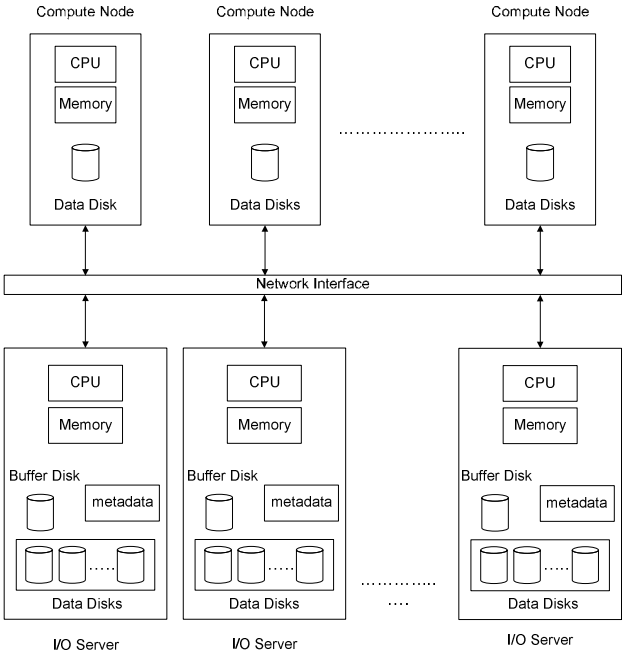

Fig. 1. The architecture of ECOS - an energy efficient cluster storage system. Each I/O node in ECOS contains a buffer disk and multiple data disks. Large files are striped across a number of I/O nodes connected through a high-speed network. Alternatively, large files might be distributed across a number of data disks within one I/O node.

RAM buffer with a size ranging from several megabytes to gigabytes is residing in the main memory. The buffer-disk controller carefully coordinates disk request processing, data movement/placement strategies, data striping, power management, and prefetching schemes. Please refer to the next subsection for details of how the controller is developed. It is to be noted that in most cases, the number of buffer disks m is smaller than the number of data disks n, and values of m and n are independent of one another for workloads with inter-request parallelisms.

The second ECOS configuration is designed for disk workloads with intra-request parallelisms. This configuration is similar to the previous one except that all the data disks in ECOS are conceptually partitioned into k groups of parallel disks each of which has m disk drives. Given m buffer disks, the second configuration is capable of serving disk requests with parallelism degrees as high as up to m.

There are two general ways of placing buffer disks. In the first approach, each I/O node only contains a single buffer disk serving all the other data disks within the I/O node. Alternatively, all the buffer disks can be grouped and placed into one or more I/O nodes. I/O nodes containing only buffer disks are called buffer I/O nodes; I/O nodes equipped with data disks are referred to as data I/O nodes. Comparing these two approaches, we advocate for the first one and the reason is two-fold. First, large popular files can be striped across multiple buffer disks residing in multiple I/O nodes. In doing so, any I/O node is most unlikely to become a performance bottleneck. Second, a buffer disk within an I/O

node can dedicate to data disks within the same I/O node, eliminating unnecessary communications between the buffer disk and data disks in other I/O nodes. Enforcing buffer disks to serve data disks within the same I/O node can reduce data transfers among I/O nodes through the network. As a result, evenly placing buffer disks across all the I/O node not only can achieve high aggregate I/O bandwidth, but also can improve network performance by reducing network traffic.

Buffer Disk Controller. The buffer disk controller, a centerpiece in the ECOS storage system, critically affects the overall performance and energy efficiency of I/O nodes. The buffer disk controller be designed and implemented to achieve the following specific goals. First, the buffer disk controller aims to minimize the number of active buffer disks while maintaining reasonably quick response times for disk requests. Second, the controller has to energy-efficiently deal with read and write requests issued to I/O nodes. Third, the controller must move data from buffer disks to home data disks in an energy-efficient way. Fourth, the controller is intended to incorporate an energy-aware prefetching strategy to dynamically fetch the most popular data into buffer disks, thereby allowing most data disks to be in the sleep mode to save energy. Design issues of the energy-aware buffer disk controller are discussed as follows.

Data Placement. Data placement, allocation of all popular files into buffer disks, can significantly affect energy efficiency and performance of ECOS. To fully exploit the capacity of parallel I/O, researchers have extensively investigated data placement algorithms for parallel disk systems. Conventional wisdom in the design of data placement mechanisms is to minimize a cost function while allocating data onto an array of independent disks. Most cost functions were focused on performance metrics (e.g., mean response time), ignoring the issue of energy saving. It is appealing to design data placement strategies to achieve high energy efficiency and quick response times in the context of cluster storage systems.

Energy Consumption Model. An energy cosumption model was implemented in ECOS to calculate energy dissipation in I/O nodes. We chose to use a model rather than an instrument to measure energy consumed by I/O nodes because the model allows us to evaluate impacts of a wide variety of disk drives on energy efficiency of ECOS.

For comparison purpose, we also implement an energy consumption model for a cluster storage system without employing buffer disks in I/O nodes. Before presenting the energy consumption models of the ECOS and non-ECOS systems, we first summarize the notation in Table I.

Let E_i^D and E_j^B be the energy dissipation in the ith data disk and jth buffer disk, respectively. The total energy consumption E_{ECOS} of the ECOS system is the sum of energy dissipation in n data disks and m buffer disks. Thus, the energy consumption in ECOS can be expressed by Eq. (1).

TABLE I
NOTATION FOR MODELING ENERGY CONSUMPTION IN THE ECOS AND NON-ECOS SYSTEMS

Notation	Definition
n	Number of data disks
m	Number of buffer disks
E_{ECOS}	Total energy consumption of ECOS
E_i^D	Energy consumption of data disk i
E_j^B	Energy consumption of buffer disk j
$\alpha_{D,i}$	Energy penalty of spinning up data disk i
$\beta_{D,i}$	Energy penalty of spinning down data disk i
$\alpha_{B,i}$	Energy penalty of spinning up data disk i
$\beta_{B,i}$	Energy penalty of spinning down data disk i
$E_{non-ECOS}$	Total energy consumption of non-ECOS
$P_{D,i}^A$	Active power of data disk i
$P_{D,i}^S$	Standby power of data disk i
$P_{B,j}^A$	Active power of buffer disk j
$P_{B,i}^S$	Standby power of buffer disk i
$T_{D,i}^A$	Active time of data disk i
$T_{D,i}^S$	Standby time of data disk i
$T_{B,j}^A$	Active time of buffer disk j
$T_{B,j}^S$	Standby time of buffer disk j
$N_{D,i}^{up}$	Number of spin-ups of data disk i
$N_{D,i}^{down}$	Number of spin-downs of data disk i
$N_{B,i}^{up}$	Number of spin-ups of buffer disk i
$N_{B,i}^{down}$	Number of spin-downs buffer disk i
R	energy conservation Rate

$$\mathrm{E}_{ECOS} = \sum_{i=1}^{n} \mathrm{E}_i^D + \sum_{j=1}^{m} \mathrm{E}_j^B \tag{1}$$

The energy consumption E_i^D of data disk i in the ECOS system is the summation of the energy incurred by the data disk when it is in the active, idle, standby, and transition states. Thus, E_i^D can be calculated by Eq. (2)).

$$\begin{aligned}\mathrm{E}_i^D = \mathrm{P}_{D,i}^A \mathrm{T}_{D,i}^A + \mathrm{P}_{D,i}^I \mathrm{T}_{D,i}^I + \mathrm{P}_{D,i}^S \mathrm{T}_{D,i}^S + \\ + \mathrm{N}_{D,i}^{up} \alpha_{D,i} + \mathrm{N}_{D,i}^{down} \beta_{D,i}\end{aligned} \tag{2}$$

where $P_{D,i}^A$, $P_{D,i}^I$, and $P_{D,i}^S$ are the power of data disk i when the disk is in the active, idle, and standby mode; $T_{D,i}^A$, $T_{D,i}^I$, and $T_{D,i}^S$ are time intervals when the disk is in the three power states, $Nup_{D,i}$ and $Nown_{D,i}$ are the numbers of spin-ups and spin-downs; and $\alpha_{D,i}$ and $\beta_{D,i}$ are the energy penalty of spin-ups/downs. We observed that active power and idle power of many hard drives are very close and; therefore, we can simply the above equation by assuming that active power and idle power are identical (i.e., $P_{D,i}^A = P_{D,i}^I$. Thus, Eq. (2)) can be simplied as Eq. (3):

$$\mathrm{E}_i^D = \mathrm{P}_{D,i}^A \mathrm{T}_{D,i}^A + \mathrm{P}_{D,i}^S \mathrm{T}_{D,i}^S + \mathrm{N}_{D,i}^{up} \alpha_{D,i} + \mathrm{N}_{D,i}^{down} \beta_{D,i} \tag{3}$$

Energy dissipation E_j^B of buffer disk j in ECOS can be derived in the same means as that of data disks. Hence, E_j^B in Eq. (1) can be written as:

$$\mathrm{E}_i^B = \mathrm{P}_{B,i}^A \mathrm{T}_{B,i}^A + \mathrm{P}_{B,i}^I \mathrm{T}_{B,i}^I + \mathrm{P}_{B,i}^S \mathrm{T}_{B,i}^S + + \mathrm{N}_{B,i}^{up} \alpha_{B,i} + \mathrm{N}_{B,i}^{down} \beta_{B,i} \quad (4)$$

Under relatively high I/O workloads, it is unlikely to spin down any buffer disks. As a results, buffer disks are not placed into standby; all the buffer disks are kept in the active mode. The numbers of buffer disk spin-ups/downs are zero; there is no power penalty of spin-ups/downs. Consequently, the Eq. 5 can be simply as:

$$\mathrm{E}_{B,j} = \mathrm{P}_{B,j}^A \mathrm{T}_{B,j}^A \quad (5)$$

Now we are positioned to consider the energy consumption of a non-ECOS system. To make fair and conservation comparisons, we model a non-ECOS systems with n data disks. In other words, we remove m buffer disks from the ECOS system in order to turn the cluster storage system into a non-ECOS system. Let E_i denote the energy consumption of the ith disk in non-ECOS. Then, E_i can be derived from the energy incurred by the disk when it is in the active, idle, standby, and transition states. In case where the active power and idle power are very close, we can express the total energy consumption of the non-ECOS system using Eq. (6)

$$\begin{aligned} \mathrm{E}_{non-ECOS} &= \sum_{i=1}^{n} \mathrm{E}_i \\ &= \sum_{i=1}^{n} (\mathrm{P}_i^A \mathrm{T}_i^A + \mathrm{P}_i^I \mathrm{T}_i^I + \mathrm{P}_i^S \mathrm{T}_i^S + \mathrm{N}_i^{up} \alpha_i + \mathrm{N}_i^{down} \beta_i) \\ &\approx \sum_{i=1}^{n} (\mathrm{P}_i^A \mathrm{T}_i^A + \mathrm{P}_i^S \mathrm{T}_i^S + \mathrm{N}_i^{up} \alpha_i + \mathrm{N}_i^{down} \beta_i) \end{aligned} \quad (6)$$

where P_i^A, P_i^I, and P_i^S are the power of disk i when the disk is in the active, idle, and standby mode; T_i^A, T_i^I, and T_i^S are time intervals when the disk is in the three power states, $N^u p_i$ and $N^d own_i$ are the numbers of spin-ups and spin-downs; and α_i and β_i are the energy penalty of spin-ups/downs.

IV. Experimental Results

In this section, we first describe a way of setting up a cluster as a testbed. Second, the experimental results are comprehensively analyzed.

A. Experiment Details

In the implemention of ECOS, each I/O node consists of three local disks including one buffer disk and two data disks. Buffer disks under relatively high I/O workloads are never spinned down; as a result, idle buffer disks can serve any incoming disk request immediately without paying spin-up penalty. To reduce network traffic incurred by communications among I/O nodes, we implemented the request processing module to ensure that a buffer disk within an I/O node gives high priorty to data disks within the same I/O node. By default, disk requests are redirected from a data disk to a buffer disk within the same I/O node. If one I/O node is heavily loaded while another one has high I/O load, requests might be redirected from the data disks in the heavily loaded node to the buffer disk in the lightly loaded one.

We developed a micro-benchmark running on a compute node. The micro-benchmark randomly issues disk requests based on the Poisson process. To reflect real-world I/O access patterns, we intentionally inserted idle periods between two consective request groups sent to I/O nodes in ECOS. In doing so, the micro-benchmark can issue a large number of disk requests representing I/O burstiness. It is observed that idle periods significantly affect energy efficiency of ECOS.

There is an important parameter referred to as Sum of Requests in Buffer or SRB, which affects the energy efficiency of ECOS. Hence, in our experiments, we focus on the impacts of SRB on the energy efficiency of the ECOS systems.

In addition to energy efficiency, energy conservation rate defined by Eq. (7) is used as a metric to quantitatively compare ECOS with non-ECOS.

$$\mathrm{R} = (1 - \frac{\mathrm{E}_{ECOS}}{\mathrm{E}_{nonECOS}}) \times 100\% \quad (7)$$

Table II summarizes the disk configuration of the tested cluster.

TABLE II
Disks Configuration

Disk Category	I/O Node 1
Buffer Disk	Maxtor DiamondMax Plus 9 80GB
Data Disk 1	WesternDigital 400 20GB
Data Disk 2	WeaternDigital 400 20GB
Disk Category	**I/O Node 2**
Buffer Disk	Seagate Barracuda 7200.7 80GB
Data Disk 1	WesternDigital 400 20GB
Data Disk 2	Maxtor D740X-6L 20GB

B. Performance Evaluation

In this subsection, let us present energy dissipation and energy conservation rate of ECOS. We first evalute the impacts of the SRB value and idle gap on energy conservation rate. In this experiment, SRB is decrease from 400 down to 25; the idle gap is varied from 50 to 300 Sec. Results plotted in Fig. 2 reveals that when I/O workloads are low, ECOS can significantly improve energy efficiency over non-ECOS. For example, ECOS conserves energy by up to more than 20% when the idle gap between two consective request groups is 300 or 200 Sec.

Recall that each I/O node in ECOS contains one buffer disk and two data disks. To make fair comparisons between ECOS and non-ECOS, we implemented a timeout policy in non-ECOS to place a disk into the standby mode if the disk has been sitting idle for a period of time (e.g., 20 Seconds). Compared with I/O nodes in non-ECOS, each I/O node in ECOS contains an extra buffer disk. Although adding a buffer disk in each I/O node seemingly imposes negative impact on energy efficiency, the results show that extra buffer disks can save a significant amount of energy. We contribute this trend

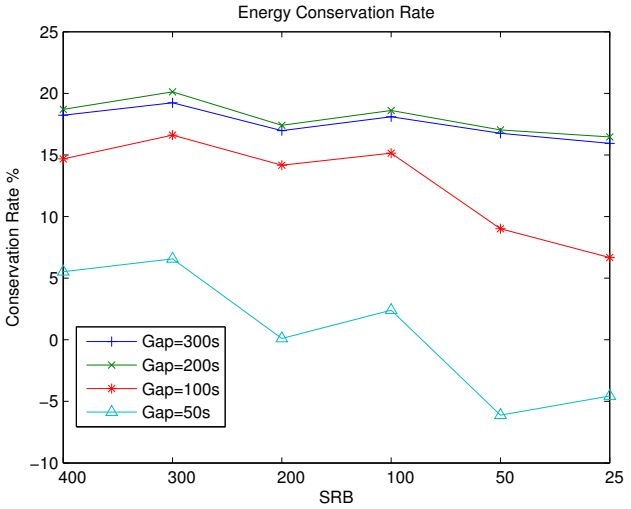

Fig. 2. Energy Conservation Rate

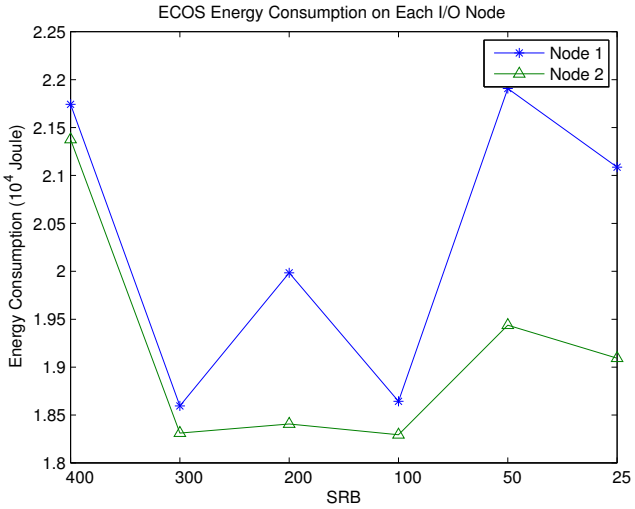

Fig. 3. Energy Consumption in I/O Nodes, idle time gap is 50s

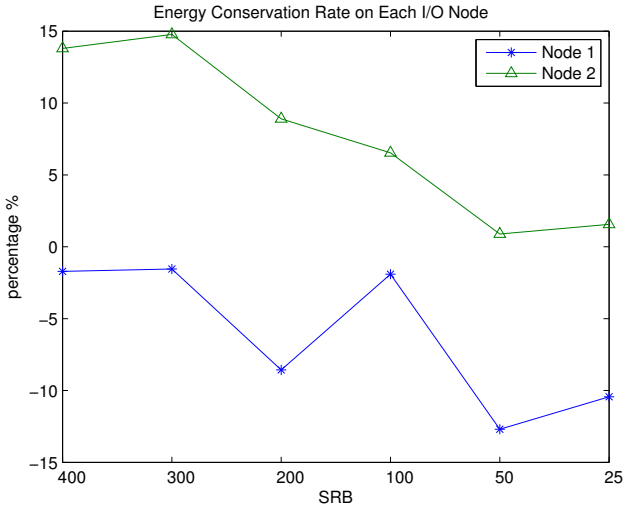

Fig. 4. Energy Conservation Rate in I/O Nodes, idle time gap is 50s

to the fact that energy saved by the buffer disks is larger than the energy overhead introduced by the extra buffer disks. In constrast to light I/O workloads, an extremely high I/O load can prevent ECOS from producing energy savings. High I/O workloads eliminate long idle periods in both buffer and data disks, thereby reducing the number of opportunities for data disks to be placed in the standby mode.

It is intriguing to observe from Fig. 2 that when the idle gap is as short as 50 Sec., the energy conservation rate becomes even negative, meaning that ECOS consumes more energy than non-ECOS. Recall that if the number of bufferred requests targeting at the same data disk equal to SRB, then the corresponding data disk must spin-up so that the bufferred data can be moved back to this data disk. The data disk is spun down to the standby mode under the following two conditions. (1) No read request is retriving data from the data disk; and (2) no write requests are currently being handled by the disk. The larger the SRB value, the more energy can be conserved in ECOS. However, I/O access patterns can greatly affect the energy conservation rate. The best time to transfer data between a buffer disk and data disks within an I/O node is at the time when the node is idle. ECOS forces buffer disks to copy data back to data disks when the SRB requirements are satisfied. Fig. 2 confirms that high frequency of data movement during I/O burstiness can reduce conservation rate.

It is worth noting that we implemented ECOS on a heterogenous cluster storage system, where the hard drives in the tested I/O nodes are not identical. As such, the goal of the second experiment is two-fold. First, we intend to show our approach can achieve high energy efficient for both homogeneous and heterogeneous cluster storage systems. Second, we plan to observe how overall system energy saving is affected by energy conservation provided in each I/O node. In what follows, we plot eight figures showing the energy consumption and energy conservation rate of the two individual I/O nodes in ECOS. Please note that the hard drives in I/O node 2 are more power consuming and faster than those in I/O node 1. Results depicted in Figs. 3-8 show although energy efficiency of the two I/O nodes are different, the energy-efficiency trends of the two nodes are quite similar.

Fig. 3 shows that the energy dissipation in I/O node 1 is larger than that of I/O node 2. The buffer disk in node 2 saves more energy than the buffer disk in node 1, because of the following four reasons. First, the workload is high due to a small value of idle gap (i.e., 50 Seconds). Second, the performance of the buffer disk in node 2 is higher than that of the buffer disk in node 1. Third, compared with node 1, node 2 can quickly move data back to the home data disks. This trend is more pronounced under high I/O workloads with enormous I/O burstiness. Last, data disks in node 1 are more likely to stay in the active state due to the slow process of moving data back to the home disks. Fig. 4 shows the energy

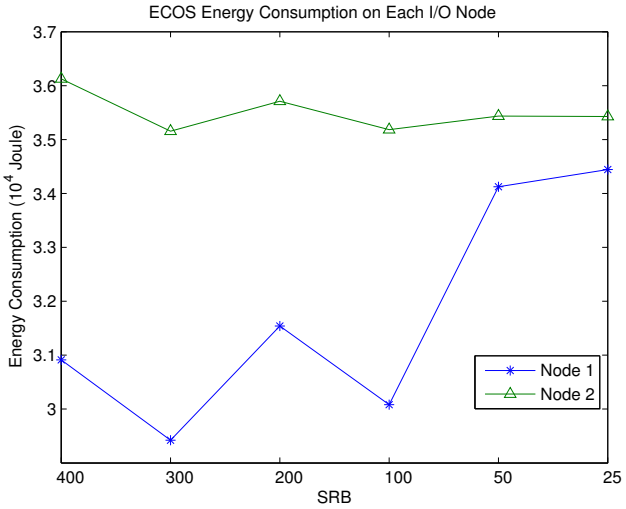

Fig. 5. Energy Consumption in I/O Nodes, idle time gap is 100s

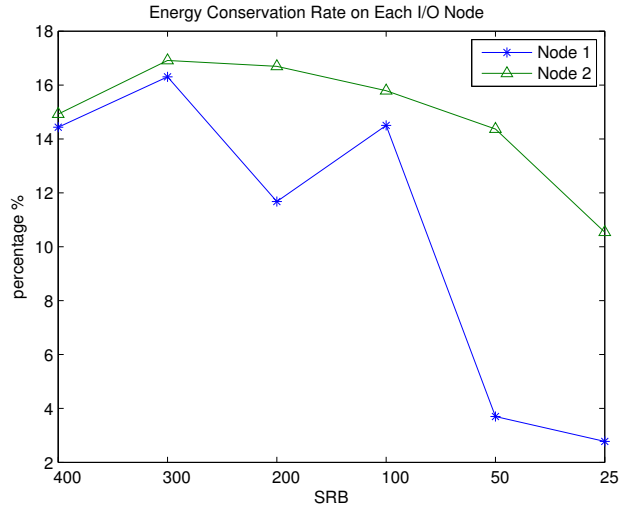

Fig. 6. Energy Conservation Rate in I/O Nodes, idle time gap is 100s

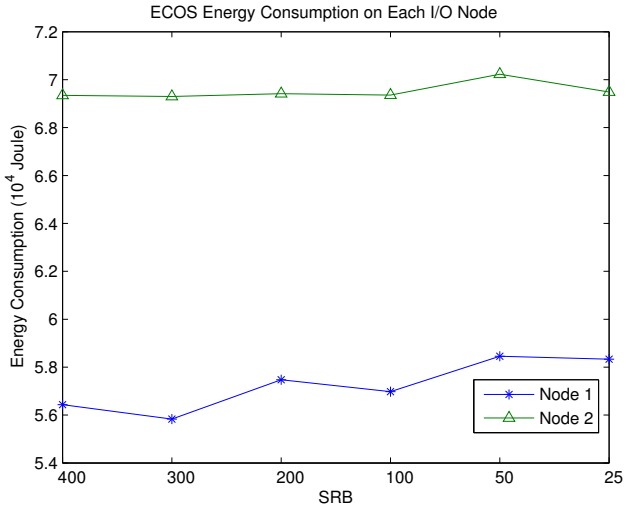

Fig. 7. Energy Consumption in I/O Nodes, idle time gap is 200s

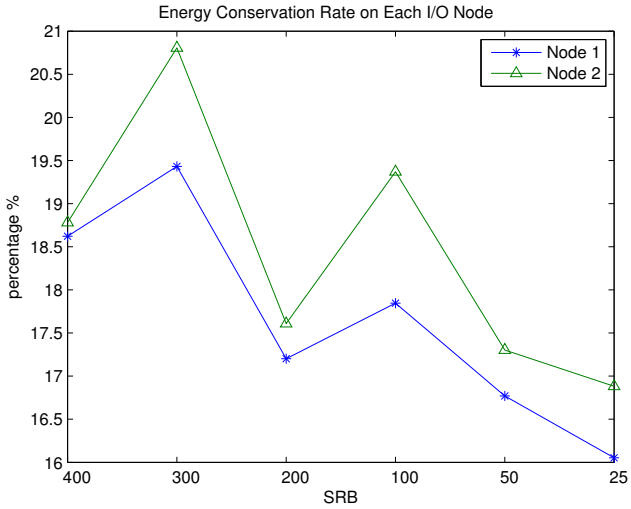

Fig. 8. Energy Conservation Rate in I/O Nodes, idle time gap is 200s

consumption rate of I/O node 2 is higher than that of I/O node 1. This is mainly because time spent in moving data from a buffer disk to data disks in node 2 is shorter than that spent in moving data in node 1. The implication behind this result is that fast data movement between a buffer disk and data disks can help in achieving high energy efficiency.

Figs. 5 and 6 plot energy consumption and energy conservation rate of the two I/O nodes. We observe from Figs. 5 and 6 that for a medium workload (e.g., idle gap is set to 100 Seconds), there is a low probability of transferring data between buffer and data disks during I/O burstiness. Therefore, the discrepancy between the data movement times of the two I/O nodes starts diminishing with decreasing I/O workload. Unlike high I/O load that makes node 2 more energy efficient than node 1, medium I/O allows node 1 exhibit more energy-efficient than node 2 (see Fig. 5). Furthermore, Fig. 5 indicates that node 1 is more sensitive to SRB than node 2; the sensitivity can be analyzed as follows. First, decreasing the SRB value increases the frequency of data movement between buffer disks and their corresponding data disks. Second, the high data movement frequency leads to a high probability of transferring data back and forth between a buffer disk and data disks during I/O burstiness.

When the idle gap increases to 200 Seconds, a small SRB does not significantly affect the energy consumption in both I/O nodes. Fig. 7 illustrates that energy consumption slowly increases when SRB decreases. Fig. 8 shows that under condition that SRB is 400 or 200, the data movement operations tend to occur within an idle gap between two consecutive request groups. This result indicates that to deliver the high performance, the data movement mechaism must be actuated at the time between two consecutive I/O burstiness.

V. Conclusion

Cluster storage systems are cost-effective building blocks for many high-end computing infrastructures. Optimizing energy efficiency of cluster storage systems remains an open issue. In this reserach, we designed and implemented an energy-efficient cluster storage system called ECOS. Each I/O node in ECOS controls multiple disks - one buffer disk and several data disks. The key idea behind ECOS is to redirect disk requests from data disks to the buffer disks. To improve I/O performance of buffer disks, ECOS attempts to balance I/O load among all I/O nodes in the cluster storage system. Redirecting requests is a driving force of energy saving and the reason is two-fold. First, ECOS makes an effort to keep buffer disks active while placing data disks into standby in a long time period to conserve energy. Second, ECOS reduces the number of disk spin downs/ups in I/O nodes.

Results show that ECOS improves energy efficiency of traditional cluster storage systems without using buffer disks. Interestingly, our results indicate that ECOS equipped with extra buffer disks is more energy efficient than the same cluster storage system without the buffer disks. Using existing data disks in I/O nodes to perform as buffer disks can achieve even higher energy efficiency.

Acknowledgments

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-0917137 (CSR), CNS-0831502 (CyberTrust), CNS-0855251 (CRI), 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] Striping and buffer caching for software raid file systems in workstation clusters. In *ICDCS '99: Proceedings of the 19th IEEE International Conference on Distributed Computing Systems*, page 544, Washington, DC, USA, 1999. IEEE Computer Society.

[2] Where does power go. http://www.greendataproject.org/, 2007.

[3] Power consumption of supercomputers. http://www.top500.org/lists/2008/06/highlights/power, June 2008.

[4] P. Bohrer, E. Elnozahy, T. Keller, M. Kistler, C. Lefurgy, C. McDowell, and R. Rajamony. *The Case for Power Management in Web Servers*. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[5] J. Chase, D. Anderson, P. Thackar, A. Vahdat, and R. Boyle. Managing energy and server resources in hosting centers. *Proc. 18th Symp. on Operating Systems Principles*, October 2001.

[6] D. Colarelli and D. Grunwald. Massive arrays of idle disks for storage archives. *Proc. ACM/IEEE Conf. on Supercomputing*, pages 1–11, 2002.

[7] Fred Douglis, P. Krishnan, and Brian Marsh. Thwarting the power-hungry disk. In *WTEC'94: Proceedings of the USENIX Winter 1994 Technical Conference on USENIX Winter 1994 Technical Conference*, pages 23–23, Berkeley, CA, USA, 1994. USENIX Association.

[8] E. N. Elnozahy, M. Kistler, and R. Rajamony. Energy-efficient server clusters. *Proc. 2nd Workshop on Power-Aware Computing Systems*, February 2002.

[9] M. Elnozahy, M. Kistler, and R. Rajamony. Energy conservation policies for web servers. *Proc. 4th USENIX Symp. on Internet Technologies and Systems*, March 2003.

[10] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, and H. Franke. Drpm: dynamic speed control for power management in server class disks. pages 169–179, June 2003.

[11] David P. Helmbold, Darrell D. E. Long, Tracey L. Sconyers, and Bruce Sherrod. Adaptive disk spin—down for mobile computers. *Mob. Netw. Appl.*, 5(4):285–297, 2000.

[12] J. W. Hsieh, T. W. Kuo, P. L. Wu, and Y. C. Huang. Energy-efficient and performance-enhanced disk using flash-memory cache. *Proc. Int'l Symp. on Low Power Electronics and Design*, pages 334–339, 2007.

[13] S. Jin and A. Bestavros. Gismo: A generator of internet streaming media objects and workloads. *ACM SIGMETRICS Performance Evaluation Review*, November 2001.

[14] N. Joukov and J. Sipek. Greenfs: Making enterprise computers greener by protecting them better. *Proc. ACM SIGOPS Operating Systems Review*, pages 69–80, 2008.

[15] P Krishnan, M P Long, and Scott J Vitter. Adaptive disk spindown via optimal rent-to-buy in probabilistic environments. Technical report, Durham, NC, USA, 1995.

[16] SangKeun Lee and Chong-Sun Hwang. Efficient, energy conserving transaction processing in wireless data broadcast. *IEEE Trans. on Knowl. and Data Eng.*, 18(9):1225–1238, 2006. Member-Kitsuregawa,, Masaru.

[17] Kester Li, Roger Kumpf, Paul Horton, and Thomas Anderson. A quantitative analysis of disk drive power management in portable computers. In *WTEC'94: Proceedings of the USENIX Winter 1994 Technical Conference on USENIX Winter 1994 Technical Conference*, pages 22–22, Berkeley, CA, USA, 1994. USENIX Association.

[18] Adam Manzanares, Xiaojun Ruan, Shu Yin, and Mais Nijim. Energy-aware prefetching for parallel disk systems: Algorithms, models, and evaluation. *IEEE Int'l Symp. on Network Computing and Applications*, 2009.

[19] Inc. Micron Technology. Wearing-leveling techniques in nand flash devices. *Micron Technology, Inc. Specification*, 2008.

[20] E. Pinheiro and R. Bianchini. Energy conservation techniques for disk array-based servers. *Int'l Conf. on Supercomputing*, pages 68–78, 2004.

[21] E. Pinheiro, R. Bianchini, E. Carrera, and T. Heath. Load balancing and unbalancing for power and performance in cluster-based systems. *Proc. Workshop on Compilers and Operating Systems for Low Power*, September 2001.

[22] Eduardo Pinheiro, Ricardo Bianchini, and Cezary Dubnicki. Exploiting redundancy to conserve energy in storage systems. *SIGMETRICS Perform. Eval. Rev.*, 34(1):15–26, 2006.

[23] X. J. Ruan, A. Manzanares, K. Bellam, Z. L. Zong, and X. Qin. Daraw: A new write buffer to improve parallel i/o energy-efficiency. *Proc. ACM Symp. on Applied Computing*, 2009.

[24] Seung Woo Son and Mahmut Kandemir. Energy-aware data prefetching for multi-speed disks. In *CF '06: Proceedings of the 3rd conference on Computing frontiers*, pages 105–114, New York, NY, USA, 2006. ACM.

[25] S.W. Son, M. Kandemir, and A. Choudhary. Software-directed disk power management for scientific applications. pages 4b–4b, April 2005.

[26] Daniel Stodolsky, Mark Holland, William V. Courtright, II, and Garth A. Gibson. Parity logging disk arrays. *ACM Trans. Comput. Syst.*, 12(3):206–235, 1994.

[27] Peter J. Varman and Rakesh M. Verma. Tight bounds for prefetching and buffer management algorithms for parallel i/o systems. *IEEE Trans. Parallel Distrib. Syst.*, 10(12):1262–1275, 1999.

[28] Jun Wang, Huijun Zhu, and Dong Li. eraid: Conserving energy in conventional disk-based raid system. *IEEE Transactions on Computers*, 57(3):359–374, 2008.

[29] Andreas Weissel, Björn Beutel, and Frank Bellosa. Cooperative i/o: a novel i/o semantics for energy-aware applications. In *OSDI '02: Proceedings of the 5th symposium on Operating systems design and implementation Due to copyright restrictions we are not able to make the PDFs for this conference available for downloading*, pages 117–129, New York, NY, USA, 2002. ACM.

[30] Qing Yang and Yiming Hu. Dcd — disk caching disk: A new approach for boosting i/o performance. pages 169–169, May 1996.

[31] Qingbo Zhu, Francis M. David, Christo F. Devaraj, Zhenmin Li, Yuanyuan Zhou, and Pei Cao. Reducing energy consumption of disk storage using power-aware cache management. In *HPCA '04: Proceedings of the 10th International Symposium on High Performance Computer Architecture*, page 118, Washington, DC, USA, 2004. IEEE Computer Society.