Final Report for Period: 05/2010 - 03/2011 **Submitted on:** 03/25/2011
Principal Investigator: Qin, Xiao . **Award ID:** 0742187
Organization: Auburn University
Submitted By:
Qin, Xiao - Principal Investigator
Title:
BUD: A Buffer-Disk Architecture for Energy Conservation in Parallel Disk Systems

Project Participants

Senior Personnel

Name: Qin, Xiao
Worked for more than 160 Hours: Yes
Contribution to Project:

Name: Xie, Tao
Worked for more than 160 Hours: Yes
Contribution to Project:

Name: Yang, Yiming
Worked for more than 160 Hours: No
Contribution to Project:
Dr. Yang helped in designing the novel energy-aware parallel disk architecture using buffer disks.

Post-doc

Graduate Student

Name: Bellam, Kiranmai
Worked for more than 160 Hours: Yes
Contribution to Project:
Dissertation: Power, fault tolerance, and security issues in real-time systems
Power, fault tolerance, and security issues in modern real-time systems are of critical importance. This work is intended to seamlessly integrate security services and energy conservation techniques for real-time systems while endeavoring to achieving high system reliability.

Name: Manzanares, Adam
Worked for more than 160 Hours: Yes
Contribution to Project:
Dissertation: Energy Efficient Prefetching ? From Models to Implementation
The goal of this research is to bring down the cost of operating parallel disk systems. A computer hard drive can be in several states, including active, idle, or standby, and these states consume various amounts of energy. The project attempts to prefetch popular data into a small subset of the parallel disks (buffer disks) and allow the other disks in the disk system (data disks) to be placed in the standby state. When the data is moved it may cause the buffer disks to become a bottleneck, so the buffer disks must be carefully managed to prevent unacceptable degradations to response times.

Name: Zong, Ziliang
Worked for more than 160 Hours: Yes
Contribution to Project:
Dissertation: Energy-Efficient Resource Management for High-Performance Computing Platforms
Minimizing power dissipation is an important requirement in developing resource management systems for clusters. In this work, we investigate resource allocation solutions that conserve energy in clusters while retaining high performance. Our resource allocation approaches will judiciously allocate resources of a cluster computing system to satisfy performance needs of parallel

applications and achieve significant energy savings.

Undergraduate Student

Technician, Programmer

Other Participant

Research Experience for Undergraduates

Organizational Partners

San Diego State University

Dr. Tao Xie, the Co-PI of the project, is an assistant professor at San Diego State University. He plays an important role in this project by developing energy-aware data placement schemes.

Intel Corporation

Dr. Yiming Yang from Intel helped us to design the energy-efficient parallel disk architecture using buffer disks.

Other Collaborators or Contacts

Activities and Findings

Research and Education Activities:

Project team meetings held

The PI and Co-PI met together on several occasions during the past year: March 30, 2007, at Long Beach, California; and August 6, 2007, Arlington VA (during the NSF FSIO workshop). These project meetings are playing an important role in coordinating our collaborative research activities. In addition, the PI, Co-PI, and senior personnel made use of frequent emails and phone discussions to collaborate on the project.

A buffer-disk architecture

We designed an energy-efficient buffer-disk architecture called BUD for parallel disk systems. In this BUD configuration, referred to as interBUD aims to support inter-request parallelisms.

Power consumption models

We started the power consumption modeling effort by introducing the simplest model where each disk has only two power states: a working state Sw and a sleep state Ss. All the disks in a parallel disk system will be modeled by their power characteristics, i.e., D = (pw, ps, pu, pd, tu, td), where pw is the power consumed in the working state, ps is the power consumed in the working state, pu is the power consumed during wake up, pd is the power consumed during shutdown, tu is the transition time from the sleep state to working state. td is the transition time from the working state to sleep state. The total energy consumed by a disk can be calculated as , where Ew and Es are the energy consumed by the disk when it is in the working sleep states, Eu and Ed is the energy consumed when the disk transits from the sleep to working state and vice versa.

Energy-aware prefetching (PRE-BUD)

Our goal in this part of study was to use the interBUD architecture to aid in the reduction of the power consumption for these large-scale parallel disk systems. This work aimed at reducing the energy consumed of these systems by comparing two different approaches at reducing energy consumption using a disk management scheme called PRE-BUD. PRE-BUD is able to prefetch data into buffer disks with a desired consequence of reducing the total energy consumption of a large-scale parallel disk system. The first approach using PRE-BUD examined adds an extra disk, which becomes the buffer disk, and the other approach uses an existing disk as the buffer disk. This algorithm relies on the fact that in some applications a small percentage of the data is frequently accessed. Our goal was to place a small amount of frequently accessed data into the buffer disk reducing energy consumption. This work also focused on increasing the reliability of the disk system over typical energy aware disk management schemes. This can be achieved with a reduction in state transitions that the use of a buffer disk facilitates.

Write request processing

In the BUD architecture, large and small writes are processed in different ways. Large write requests are issued directly to data disks. In contrast, small write requests are 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 issuing the request. Our study confirmed that seek times of small disk request dominates disk I/O processing times. To alleviate this situation, we made use of sequential accesses, i.e., a log file system, in interBUD, thereby making the seek time of most write requests to be zero. This is because disk head of a sequential access disk is, in most cases, positioned on an empty track that is available for incoming write requests. The seek times of write requests handled by buffer disks are zero unless the buffer disks are in a process of moving data to data disks or responding read requests.

Energy-efficient data placement

We investigated data placement strategies, which place all data sets onto a disk array before they are accessed. Data placement is one of avenues that can significantly affect the overall performance of a parallel I/O system. We developed a data placement scheme that can simultaneously achieve high energy efficiency and quick response times through intelligent in our BUD architecture.

A Prefetching Scheme for Energy Conservation in Parallel Disk Systems

Large-scale parallel disk systems are frequently used to meet the demands of information systems requiring high storage capacities. A critical problem with these large-scale parallel disk systems is the fact that disks consume a significant amount of energy. To design economically attractive and environmentally friendly parallel disk systems, we developed two energy-aware prefetching strategies for parallel disk systems with disk buffers. First, we introduced a new buffer disk architecture that can provide significant energy savings for parallel disk systems while achieving high performance. Second, we designed a prefetching approach to utilize an extra disk to accommodate prefetched data sets that are frequently accessed. Third, we developed a second prefetching strategy that makes use of an existing disk in the parallel disk system as a buffer disk. Compared with the first prefetching scheme, the second approach lowers the capacity of the parallel disk system. However, the second approach is more cost-effective and energy-efficient than the first prefetching technique. Finally, we quantitatively compare both of our prefetching approaches against two conventional strategies including a dynamic power management technique and a non-energy-aware scheme. Using empirical results we show that our novel prefetching approaches are able to reduce energy dissipation in parallel disk systems by 44% and 50% when compared against a non-energy aware approach. Similarly, our strategies are capable of conserving 22% and 30% of the energy when compared to the dynamic power management technique.

Improving Reliability and Energy Efficiency of Disk Systems via Utilization Control
As disk drives become increasingly sophisticated and processing power increases, one of the most critical issues of designing modern disk systems is data reliability. Although numerous energy saving techniques are available for disk systems, most of energy conservation techniques are not effective in reliability critical environments due to their limitation of ignoring the reliability issue. A wide range of factors affect the reliability of disk systems; the most important factors ? disk utilization and ages ? are the focus of this study. We build a model to quantify the relationship among the disk age, utilization, and failure probabilities. Observing that the reliability of a disk heavily relies on both disk utilization and age, we propose a novel concept of safe utilization zone, where energy of the disk can be conserved without degrading reliability. We investigate an approach to improving both reliability and energy efficiency of disk systems via utilization control, where disk drives are operated in safe utilization zones to minimize the probability of disk failure. In this study, we integrate an existing energy consumption technique that operates the disks at different power modes with our proposed reliability approach. Experimental results show that our approach can significantly improve reliable while achieving high energy efficiency for disk systems.

DARAW: A New Write Buffer to Improve Parallel I/O Energy-Efficiency
In the past decades, parallel I/O systems have been used widely to support scientific and commercial applications. New data centers today employ huge quantities of I/O systems, which consume a large amount of energy. Most large-scale I/O systems have an array of hard disks working in parallel to meet performance requirements. Traditional energy conservation techniques attempt to place disks into low-power states when possible. In this part of research, we proposed a novel strategy, which aims to significantly conserve energy while reducing average I/O response times. This goal was achieved by making use of buffer disks in parallel I/O systems to accumulate small writes to form a log, which can be transferred to data disks in a batch way. We develop an algorithm - dynamic request allocation algorithm for writes or DARAW - to energy efficiently allocate and schedule write requests in a parallel I/O system. DARAW is able to improve parallel I/O energy efficiency by the virtue of leveraging buffer disks to serve a

majority of incoming write requests, thereby keeping data disks in low-power state for longer period times. Buffered requests are then written to data disks at a pre-determined time. Experimental results show that DARAW can significantly reduce energy dissipation in parallel I/O systems without adverse impacts on I/O performance.

Heat-Based Dynamic Data Caching: A Load Balancing Strategy for Energy-Efficient Parallel Storage Systems with Buffer Disks
Performance improvement and energy conservation are two conflicting objectives in large scale parallel storage systems. In this project, we proposed a novel solution to achieve the twin objectives of maximizing performance and minimizing energy consumption of parallel storage systems. Specifically, a buffer-disk based architecture (BUD for short) is designed to conserve energy. A heat-based dynamic data caching strategy is developed to improve performance. The BUD architecture strives to allocate as many requests as possible to buffer disks, thereby keeping a large number of idle data disks in low-power states. This can provide significant opportunities for energy conservation while making buffer disks a potential performance bottleneck. The heat-based data caching strategy aims to achieve good load balancing in buffer disks and alleviate overall performance degradation due to unbalanced workload. Our experimental results show that the proposed BUD framework and dynamic data caching strategy are able to conserve energy by 84.4% for small reads and 78.8% for large reads with slightly degraded response time.

An Adaptive Energy-Conserving Strategy for Parallel Disk Systems
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 a variety of parallel disk systems were developed to achieve high performance and energy efficiency, most existing parallel disk systems lack an adaptive way to conserve energy in dynamically changing workload conditions. To solve this problem, we develop an adaptive energy-conserving algorithm, or DCAPS, for parallel disk systems using the dynamic voltage scaling technique that dynamically choose the most appropriate voltage supplies for parallel disks while guaranteeing specified performance (i.e., desired response times) for disk requests. We conduct extensive experiments to quantitatively evaluate the performance of the proposed energy-conserving strategy. Experimental results consistently show that DCAPS significantly reduces energy consumption of parallel disk systems in a dynamic environment over the same disk systems without using the DCAPS strategy.

Energy-Efficient Data Placement
We investigated data placement strategies, which place all data sets onto a disk array before they are accessed. Data placement is one of avenues that can significantly affect the overall performance of a parallel I/O system. First, we developed a static non-partitioned file

assignment strategy for parallel I/O systems, called static round-robin (SOR), which is immune to the workload assumption. Next, to achieve energy-conservation and prompt responses simultaneously, we designed an energy-aware strategy, called striping-based energy-aware (SEA), which can be integrated into data placement in RAID-structured storage systems to noticeably save energy while providing quick responses. Finally, to illustrate the effectiveness of SEA, we implemented two SEA-powered striping-based data placement algorithms, SEA0 and SEA5, by incorporating the SEA strategy into RAID-0 and RAID-5, respectively.

An Energy-Aware Data Reconstruction Strategy for Mobile Disk Arrays
Compared with conventional stationary storage systems, mobile disk-array-based storage systems are more prone to disk failures due to their severe application environments. Further, they have very limited power supply. Therefore, data reconstruction algorithms, which are executed in the presence of disk failure, for mobile storage systems must be performance-driven, reliability-aware, and energy-efficient. We developed a reconstruction strategy, called multi-level caching-based reconstruction optimization (MICRO), which can be applied to RAID-structured mobile storage systems to noticeably shorten reconstruction times and user response times while saving energy.

Understanding the Relationship between Energy Conservation and Reliability in Parallel Disk Arrays
Power management and workload skew based energy conservation schemes for disk arrays inherently and adversely affect the reliability of disks due to either workload concentration or frequent disk speed transitions. A thorough understanding of the relationship between energy saving techniques and disk reliability is indispensible. We developed an empirical reliability model, called Predictor of Reliability for Energy Saving Schemes (PRESS). Fed by operating temperature, disk utilization, disk speed transition frequency, three energy-saving-related reliability affecting factors, PRESS estimates the reliability of entire disk array. Further, a new energy saving strategy with reliability awareness named Reliability and Energy Aware Distribution (READ) is developed in the light of the insights provided by PRESS.

Performance, Energy, and Reliability Balanced Dynamic Data Redistribution for Next Generation Disk Arrays
Contemporary disk arrays consist purely of hard disk drives, which normally provide huge storage capacities with low-cost and high-throughput for data-intensive applications. Nevertheless, they have some inherent disadvantages such as long access latencies, high annual disk replacement rates, fragile physical characteristics, and energy-inefficiency due to their build-in mechanical and electronic mechanisms. Flash-memory based solid state disks, on the other hand, although currently more expensive and inadequate in write cycles, offer much faster read accesses and are much more robust and energy efficient. To combine the complementary merits of hard disks and flash disks, in this research we developed a hybrid disk array architecture named HIT (hybrid disk storage) for data-intensive applications. Next, a dynamic data redistribution strategy called PEARL (performance,

energy, and reliability balanced), which can periodically redistribute data between flash and hard disks to adapt to the changing data access patterns, is developed on top of the HIT architecture.

ECOS: An Energy-Efficient Cluster Storage System
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 study 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.

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

HYBUD: An Energy-Efficient Architecture for Hybrid Parallel Disk Systems
Although flash memory is very energy-efficient compared to disk drives, flash memory is too expensive to use as a major component in large-scale storage systems. In other words, it is not a cost-effective way to make use of large flash memory to build energy-efficient storage systems. To address this problem, in this study we proposed a hybrid disk architecture or HYBUD that integrates a non-volatile flash memory with buffer disks to build cost-effective and energy-efficient parallel disk systems. While the most popular data sets are cached in flash memory, the second most popular data sets can be stored and retrieved from buffer disks. HYBUD is energy efficient because flash memory coupled with buffer disks can serve a majority of incoming disk requests, thereby keeping a large number of other data disks in the low-power state for longer period times. Furthermore, HYBUD is cost-effective by the virtue of inexpensive buffer disks assisting flash memory to cache a huge amount of popular data. Experimental results demonstratively show that compared with two existing non-hybrid architectures, HYBUD provides significant energy savings for parallel disk systems in a very cost effective way.

Energy-Aware Prefetching for Parallel Disk Systems
In this study we design and evaluate an energy-aware prefetching strategy for parallel disk systems consisting of a small number of buffer disks and large number of data disks. Using buffer disks to temporarily handle requests for data disks, we can keep data disks in the low-power mode as long as possible. Our prefetching algorithm aims to group many small idle periods in data disks to form large idle periods, which in turn allow data disks to remain in the standby state to save energy. To achieve this goal, we utilize buffer disks to aggressively fetch popular data from regular data disks into buffer disks, thereby putting data disks into the standby state for longer time intervals. A centrepiece in the prefetcing mechanism is an energy-saving prediction model, based on which we implement the energy-saving calculation module that is invoked in the prefetching algorithm. We quantitatively compare our energy-aware prefetching mechanism against existing solutions, including the dynamic power management strategy. Experimental results confirm that the buffer-disk-based prefetching can significantly reduce energy consumption in parallel disk systems by up to 50 percent. In addition, we systematically investigate the energy efficiency impact that varying disk power parameters has on our prefetching algorithm.

DORA: A Dynamic File Assignment Strategy with Replication
Compared with numerous static file assignment algorithms proposed in the literature, very few investigations on the dynamic file allocation problem have been accomplished. Moreover, none of them has integrated file replication techniques into file assignment algorithms in a highly dynamic file system where files are created or deleted on the fly and their access patterns varied over time. We argue that file replication and file assignment can act in concert to boost the performance of parallel disk systems. In this study, we propose a new dynamic file assignment strategy called DORA (dynamic round robin with replication). The advantages of DORA can be attributed to its two main

characteristics. First, it takes the dynamic nature of file access patterns into account to adapt to a changing workload condition. Second, it utilizes file replication techniques to complement file assignment schemes so that system performance can be further improved. Experimental results demonstrate that DORA performs consistently better than existing algorithms.

Collaboration-Oriented Data Recovery for Mobile Disk Arrays
Mobile disk arrays, disk arrays located in mobile data centers, are crucial for mobile applications such as disaster recovery. Due to their unusual application domains, mobile disk arrays face several new challenges including harsh operating environments, very limited power supply, and extremely small number of spare disks. Consequently, data reconstruction schemes for mobile disk arrays must be performance-driven, reliability-aware, and energy-efficient. In this study, we develop a flash assisted data reconstruction strategy called CORE (collaboration-oriented reconstruction) on top of a hybrid disk array architecture, where hard disks and flash disks collaborate to shorten data reconstruction time, alleviate performance degradation during disk recovery. Experimental results demonstrate that CORE noticeably improves the performance and energy-efficiency over existing schemes.
A File Assignment Strategy Independent of Workload Characteristic Assumptions
The problem of statically assigning non-partitioned files in a parallel I/O system has been extensively investigated. A basic workload characteristic assumption of most existing solutions to the problem is that there exists a strong inverse correlation between file access frequency and file size. In other words, the most popular files are typically small in size, while the large files are relatively unpopular. Recent studies on the characteristics of web proxy traces suggested, however, the correlation, if any, is so weak that it can be ignored. Hence, the following two questions arise naturally. First, can existing algorithms still perform well when the workload assumption does not hold? Second, if not, can one develop a new file assignment strategy that is immune to the workload assumption? To answer these questions, we first evaluate the performance of three well-known file assignment algorithms with and without the workload assumption, respectively. Next, we develop a novel static non-partitioned file assignment strategy for parallel I/O systems, called static round-robin (SOR), which is immune to the workload assumption. Comprehensive experimental results show that SOR consistently improves the performance in terms of mean response time over the existing schemes.

New course development
Our education activities include the development of a new course ? Advanced Computer Security - for graduate students. This course was intended to give students a strong background in understanding design and development of secure systems in general and secure storage systems in particular.

Findings:

A focus of the research activities carried out in the last year is the design of an energy efficient parallel disk architecture for inter-request parallelisms.

The architecture consists of four major components: a RAM buffer, m buffer disks, n data disks, and an energy-aware buffer-disk controller. The new BUD disk architecture is diagramed below [Zong et al., 2007]:

The RAM buffer with a size ranging from several megabytes to gigabytes is residing in the main memory. The buffer-disk controller carefully coordinates energy-related reliability model, data partitioning, disk request processing, data movement/placement strategies, power management, and prefetching schemes. Please refer to Section 3 for details of how the controller will be designed and developed. We choose to use log disks as buffer disks, because data can be written onto the log disks in a sequential manner to improve performance of disk systems. 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.

We introduced a model to calculating energy consumption for disk drives in a parallel disk system . [Zong et al., 2007a] The basic power model for this study is a summation of all power states multiplied by the time 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 share similar power consumption. Let Ttr 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 Ptr. Thus, energy Etr consumed by the disk when it enters and exits the inactive state is expressed as . Similarly, let Tactive and Tidle are the time intervals when the disk is in the active and inactive states, respectively. We denote the energy consumption rates of the disk when it is active and inactive by Pactive and Pidle, respectively. Hence, the energy dissipation Eactive of the disk when it is in the active state can be written as , and the energy Eidle of the disk when it is sitting idle can be expressed as . The total energy consumed by the disk system can be calculated as

.

Let Tai and Tia denote times a disk spends in entering and exiting the inactive state, Pai and Pia are the power consumption rates when the disk enters the inactive and active states. Let Nai and Nia be the number of times the disk enters and exits the inactive state. Then, the transition time Ttr and power Ptr can be computed as follows

,

.

In most cases where the values of Nai and Nia are both equal to Ntr, Ttr can be written as

The time interval Tactive when the disk is in active state is the sum of serving times of disk requests submitted to the disk system.

where n is the total number of submitted disk requests, and is the serving time of the ith disk request. can be modeled as

where Tseek is the amount of time spent seeking the desired cylinder, Trot is the rotational delay and Ttrans is the amount of time spent actually reading from or writing to the disk. Now we quantify energy saved by power management policies as below

[Zong et al., 2007a] Write requests can be divided into small write requests and large write requests. Whenever the controller receives a write request, it will first check the size. If the request is a large write, say over 10MB or more, it is sent directly to the corresponding data disk. Otherwise, the controller will send this request to the RAM buffer that buffers small writes from the host and form a log of data to be written into one of the buffer disks later. Once the data are transferred to the RAM buffer, the controller will send a 'write complete' acknowledgement message to the sender. Then the controller will test the state of all the buffer disks. If one buffer disk is not busy with writing a previous log or reading or transferring data, the data copy will be sent to this buffer disk to ensure that a reliable copy resides on one of the buffer disks. In order to guarantee the correctness and consistency of different data version, the controller is always trying to match the data with same block to the same buffer disk unless it is known that the data block is already outdated. In other words, 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. The most important scheduling strategy between RAM buffer and buffer disk is that rather than wait until the RAM is full, the data are written into the buffer disks whenever they are available. This policy has two major advantages. First, data re guaranteed to be written into one of the reliable buffer disks in the shortest time period, which is very important to ensure the reliability and availability of data. Second, the RAM buffer can have more available room to buffer a large burst of new requests because previous data are always quickly moved from the RAM to the buffer disks. Here we should note that the total storage space of each buffer disk is divided into equal n parts (n is the number of data disks) which are used to buffer the data requests corresponding to each data disk. For example, if we have two 10GB buffer disks and ten 100GB data disks, each buffer disk will have 1GB as the buffer space for each data disk. All the small write requests to data disk 1 will be buffered in the corresponding buffer space reserved for disk 1. The reason we split the buffer disks into small pieces for each data disk is to improve the response performance of the whole system. In the case when one buffer disk is busy writing or moving data, the other disk could serve the incoming requests immediately.

Handling read operations is kind of simple and straightforward in the BUD framework. When a read request arrives, the controller first searches the RAM buffer. If the data is still in the RAM buffer then the data is immediately sent back to the requester. Otherwise, the controller will do a seek operation in the buffer disks. If the required data can not be found in the buffer disks, a miss message will be sent back to controller and the controller will send a read request to the corresponding data disk and finally the data will be transferred to the requester by the data disk. Using this policy, the read performance should be similar to or sometimes better than that of traditional disk because most of the requests will be sent to the data disk and reading from RAM or buffer disks seldom occurs in real applications.

Our preliminary results consist of first developing a simulator which meets all project specifications and running this simulator with a trace to get some preliminary results [Zong et al., 2007b]. So far the simulator completed all the main functions that are necessary in order to model our distributed system. That is the program takes data from a trace. Then the program moves data to appropriate virtual disks that use disksim to derive there timing information. These virtual disks use a simple model to calculate total energy. Finally, both timing and energy data are reported to the user in the form of the two respective totals. Our results from the simulator consist of two parallel disk systems. To simulate with these two systems we used a simple trace that came with disksim. The first system that was simulated was a simple disk system which is used today by many storage systems. It is basically a RAID 1 system consisting of 31 disks. This is basically a baseline system in which to compare our results. The second simulated disk system is similar to section 3 with 6 buffer disks each one acting as a buffer for a group of approximately 5 disks.

The results from running these systems showed that both took 25.55 min. Table below shows the energy consumption of the disk system without employing buffer disks and the disk system using buffer disks to conserve energy. Specifically, the traditional system without buffer disks consumed 189279.78 J (0.05 KWH) with all disks starting from off and being turned on when needed. In contrast, the parallel disk system with buffer disks only used 117345.99 J (0.03 KWH), which is substantially less. This resulted in overall power savings of 38%. Our results have shown substantial gains can be made by using buffer disks for very specific workload conditions. In order to further develop our storage algorithm we need to test our results in many more simulations and with more varied workloads. The simulator also needs to check with analytic calculations and the code needs to include the transition time and power from switching from mode to mode.

Simulated Disk System	Energy Consumption
Disk system without buffer disks	189279.78 J
Disk system with buffer disks	117345.99 J
Energy saving	71933.79 J
Energy consumption reduced by 38%	

[Zong et al., 2007c] High performance data grids are increasingly becoming popular platforms to support data-intensive applications. Reducing high energy consumption caused by data grids is a challenging issue. Most previous studies in grid computing focused on performance and reliability without taking energy conservation into account. As such, designing energy-efficient data grid systems became highly desirable. In this project, we proposed a framework to simulate energy-efficient data grids. We presented an approach to integrate energy-aware allocation strategies into energy-efficient data grids. Our framework aims at simulating a data grid that can conserve energy for data-intensive applications running on data grids.

A data grid can be envisioned as a complicated distributed system, which consists of the following four major layers: application layer, middleware layer, resource layer and network layer. Fig. 1 depicts the four layers and their relations for data grids. It is worth noting that the simulation framework is constructed in accordance with the system architecture outlined as follows:

Data grids are complex multivariate environments, which are made up of numerous grid entities that need to be automatically managed. In order to make coherent and coordinated use of ubiquitous and heterogeneous data and storage resources, resource management is a centerpiece in the simulation framework. In this section, we present a framework to simulate data grids that are energy efficient in nature. In general, a data grid should energy efficiently handle two important components in the system: storage resources and data-intensive jobs. A data grid has to ad-dress the following issues. First, it is of paramount importance to find available storage resources within a short pe-riod of time. Second, it must allocate data-intensive jobs to available resources. We proposed a simulation framework for data grids:

In this framework, the global level scheduler (or grid level scheduler) coordinates mul-tiple local scheduling or helps to select the most appropri-ate resources for a job among different possible resources. Typically, a global level scheduler itself has no direct control over computing and storage resources. Therefore, the global level scheduler has to communicate with and appropriately trigger several local level schedulers to complete data-intensive tasks submitted by users. Those local level schedulers either control resources directly or have certain access to their local resources. The global level scheduler is also responsible for collaborating with other important supportive middleware like information services, communication services, and reliability control modules.

[Zong et al., 2007c] As long as the grid scheduling module collected all the information of currently available computing and storage resources, it can judiciously choose target recourses based on its scheduling policy and allocate the tasks analyzed by the task analyzer to these chosen resources for parallel execution. We designed the job scheduling flow in the simulation framework:

During the process of execution, the results collector will periodically check the randomly returned sub results and transfer these sub results to Grid level scheduler. The scheduler in the framework passes the latest information to all tasks, which can guarantee that the tasks with dependency could immediately be executed once they get the necessary sub results.

[Ruan et al., 2007] In the past decade cluster computing platforms have been widely applied to support a variety of scientific and commercial applications, many of which are parallel in nature. However, scheduling parallel applications on large scale clusters is technically challenging due to significant communication latencies and high energy consumption. As such, shortening schedule length and conserving energy consumption are two major concerns in designing economical and environmentally friendly clusters. In this study, we proposed an energy-efficient scheduling algorithm (TADVS) using the dynamic voltage scaling technique to provide significant energy savings for clusters. The TADVS algorithm aims at judiciously leveraging processor idle times to lower processor voltages (i.e., the dynamic voltage scaling technique or DVS), thereby reducing energy consumption experienced by parallel applications running on clusters. Reducing processor voltages, however, can inevitably lead to increased execution times of parallel task. The salient feature of the TADVS algorithm is to tackle this problem by exploiting tasks precedence constraints. Thus, TADVS applies the DVS technique to parallel tasks followed by idle processor times to conserve energy consumption without increasing schedule lengths of parallel applications. Experimental results clearly show that the TADVS algorithm is conducive to reducing energy dissipation in large-scale clusters without adversely affecting system performance.

In the first set of experiments, we varied CCR from 0.1 to 1 to examine the performance impacts of communication intensity on our TADVS scheduling strategy. This year, we evaluate the performance of TADVS algorithm by comparing the traditional NDS scheme:

TADVS consistently consumes less energy regardless of the value of CCR (Communication-Computation Ratio) [Ruan et al., 2007]. For example, TADVS conserves the energy consumption for the SPA application by up to 16.8% with an average of 10.7%. When one increases CCR from 0.1 to 1, the energy consumption gradually goes up. This can be explained by the fact that a high CCR results in high communication cost, which in turn leads to the increased total energy consumption. More interestingly, we observe from Fig. 4 that energy savings achieved by the TADVS strategy become more pronounced when the communication intensity is relatively low. This result clearly indicates that low communication intensity offers more space for TADVS to reduce voltage supplies of computing nodes to significantly conserve energy. In other words, applications with low communication intensities can greatly benefit from the TADVS scheduling scheme.

Figure below shows the energy consumption caused by the Gaussian application on the cluster with Intel Pentium 4 processors, whereas First of all, the experimental results reveal that TADVS can save energy consumption for the Gaussian application by up to 14.8% with an average of 9.6%. Second, the results plotted in Figs. 5 and 6 show that compared the Gaussian application with the SPA application, the energy saving rate of TADVS is less sensitive to the communication intensity. The empirical results suggest that the sensitivity of the energy saving rate of TADVS on communication intensity partially relies on the characteristics of parallel applications. Note that parallel applications' characteristics include parallelism degrees, number of messages, average message size, and the like. Compared with the SPA application (Fig. 1), the Gaussian application has a higher parallelism degree. More specifically, we concluded from the experimental results shown in Figs. 4 and 6 that the energy saving rate of TADVS is less sensitive to the communication intensity of parallel applications with higher parallel degrees. Moreover, parallel applications with higher parallelism degrees are able to take more advantages from the TADVS in terms of energy conservation. A practical implication of this observation is that although high communication intensities of parallel applications tends to reduce energy saving rates of TADVS, increasing parallel degrees of the parallel applications can potentially and noticeably boost up the energy saving rates.

[Zong et al., 2007d] High performance clusters and parallel computing technology are experiencing their golden ages because of the convergence of four critical momentums: high performance microprocessors, high-speed networks, middleware tools, and highly increased needs of computing capability. With forceful aid of cluster computing technology, complicated scientific and commercial applications like human genome sequence programs, universe dark matter observation and the Google search engine have been widely deployed and applied. Although clusters are cost-effective high-performance computing platforms, energy dissipation in large clusters is excessively high. Most previous studies in cluster computing focused on performance, security, and reliability, completely ignoring the issue of energy conservation. Therefore, designing energy-efficient algorithms for clusters, especially for heterogeneous clusters, becomes highly desirable. This year, we developed two novel scheduling strategies, called Energy-Efficient Task Duplication Scheduling (EETDS) and Heterogeneous Energy-Aware Duplication Scheduling (HEADUS), which attempt to make the best tradeoffs between performance and energy savings for parallel applications running on heterogeneous clusters. Our algorithms are based on the duplication-based heuristics, which are efficient solutions to minimize communication overheads among precedence constrained parallel tasks. Our algorithms consist of two major phases. Phase one is used to optimize performance of parallel applications and the second phase aims to provide significant energy savings. We present extensive simulation results using realistic parallel applications to prove the efficiency of our algorithm.

[Zong et al., to be published] Improve performance and conserve energy are two conflict objectives in parallel storage systems. In this project, we proposed a novel solution to achieve the twin objectives of maximizing performance and minimizing energy consumption of large scale parallel disk arrays. We observed that buffer disks can be a performance bottleneck of an energy-efficient parallel disk system. We developed a heat-based and duplication-enabled load balancing strategies to successfully overcome the natural shortcoming of the BUD architecture, in which the limited number of buffer disks are very likely become the bottleneck.

The basic idea of heat-based mapping is that blocks in data disks will be mapped to buffer disks based on their heat (access frequency). Our goal is to make the accumulated heat of data blocks allocated to each buffer disk is exactly the same or at least close. In other words, the temperature of each buffer disk should be in the same level. Here the temperature of a buffer disk means the total heat of all blocks existing in this buffer disk. For example, in current request queue, the heat of block 1-6 are 5, 4, 1, 2, 1, 2 respectively. Therefore, block 1 is cashed to buffer disk 1, block 2 and 3 are copied to buffer disk 2 and block 4, 5 and 6 are mapped to buffer disk 3. In this way, the temperature of each buffer disk is 5. The following figure depicts the dispatch results of heat-based load balancing strategy. Note that there are 15 requests cashed in the RAM buffer and they are going to be dispatched to different buffer disks by the controller. Requests have different colors, which represents they will access different data blocks. For example, request 1 will access the data block1 existing in data disk 1 and request 6 will access the data block6 existing in data disk 6.

The following figure shows another way to do load balancing, i.e. we can duplicate the most popular data blocks to several buffer disks. The basic idea of duplication based load balancing is to move multiple replicas of 'hot' data blocks to different buffer disks, which allow multiple buffer disks to serve the requests in parallel thereby improving the performance. In this example, the controller generates a load balanced dispatching by duplicating block 3 in each of the three buffer disks. To decide which block should be duplicated, we also need to calculate and order the heat of data blocks.

To validate the efficiency of the proposed framework and load balancing strategies, we conduct extensive simulations using both synthetic workload and real-life traces.

[Manzanares et al., to be published] Large-scale parallel disk systems are frequently used to meet the demands of information systems requiring high storage capacities. A critical problem with these large-scale parallel disk systems is the fact that disks consume a significant amount of energy. To design economically attractive and environmentally friendly parallel disk systems, we developed two energy-aware prefetching strategies (or PRE-BUD for short) for parallel I/O systems with disk buffers.

First, we studied a concrete example, which is based on the synthetic disk trace presented in the table below.

Synthetic Trace

Time 0 5 10 15 20 25 30 35
40 45 50 55 60 65 70 75 80
85 90 95
Block A1 A2 B1 B2 D1 D2 C1 C2
B2 B1 A1 A2 C1 C2 D1 D2 B2
B1 A1 A2

Disk Parameters (IBM 36Z15)
X = Transfer Rate =55 MB/s PAct = Power Active = 13.5 W
PIdle = Power Idle = 10.2 W PStdby = Power Standby = 2.5 W
EAS = Energy Active to Sleep = 13.0 J ESA = Energy Sleep to Active = 135.0 J
TAS = Time Active to Sleep = 1.5 s TSA = Time Sleep to Active = 10.9

The requests all have the size of 275MB. This means each request will take approximately 5s to complete. This length was chosen, so seek and rotational delays would be negligible. There are N=4 disks used in this example, where each disk is given a unique letter. Each disk has two different data sections requested multiple times throughout the example. The example demonstrates only large sequential reads. This was also chosen to simplify our example to allow us to demonstrate the potential benefits of our approach. We also assume that all the data can be buffered which causes a small percentage of data to be accessed 100% of the time. This is only used for our motivational example and our simulation results vary this parameter to model real-world conditions. We also assume these strategies can be handled off-line meaning we have prior knowledge of the complete disk request pattern.

Non-Energy Aware Results
TIdle(A) 70s TIdle(B) 70s
TIdle(C) 80s TIdle(D) 80s
TAct(A) 30s TAct(B) 30s
TAct(C) 20s TAct(D) 20s
ETrans(A) 0J ETrans(B) 0J
ETrans(C) 0J ETrans(D) 0J
TE(A) 1119J TE(B) 1119J
TE(C) 1086J TE(D) 1086J
TES 4410J

Energy Aware Results
TIdle(A) 0s TIdle(B) 10s
TIdle(C) 0s TIdle(D) 0s
TAct(A) 30s TAct(B) 30s
TAct(C) 20s TAct(D) 20s
TSleep(A) 45.2s TSleep(B) 33.7s
TSleep(C) 53.7s TSleep(D) 53.7s
ETrans(A) 296J ETrans(B) 309J
ETrans(C) 309J ETrans(D) 309J
TE(A) 814J TE(B) 900.25J
TE(C) 713.25J TE(D) 713.25J
TES 3140.75J

PRE-BUD Approach 1
TIdle(A) 0s TIdle(B) 0s
TIdle(C) 0s TIdle(D) 0s
TAct(A) 10s TAct(B) 10s
TAct(C) 10s TAct(D) 10s
TSleep(A) 100s TSleep(B) 100s
TSleep(C) 100s TSleep(D) 100s
ETrans(A) 13J ETrans(B) 13J
ETrans(C) 13J ETrans(D) 13J
TE(A) 398J TE(B) 398J
TE(C) 398J TE(D) 398J

BDEPre 540 BDETot 1350J
TES 3482J

PRE-Bud Approach 2
TIdle(A) 0s TIdle(B) 0s
TIdle(C) 0s TIdle(D) 0s
TAct(A) 140s TAct(B) 10s
TAct(C) 10s TAct(D) 10s
TSleep(A) 0s TSleep(B) 100s
TSleep(C) 100s TSleep(D) 100s
ETrans(A) 0J ETrans(B) 13J
ETrans(C) 13J ETrans(D) 13J
TE(A) 1890J TE(B) 398J
TE(C) 398J TE(D) 398J
BDEPre 540 BDETot 1350J
TES 3084J

The results presented in the above four Tables gave us some promising initial results. The two approaches using a buffer disk provided significant energy savings over the non-energy aware parallel disk storage system. This has the benefit of not impacting the capacity of the large-scale parallel disk system. The other main benefit of our first approach is the fact that state transitions are lowered as compared to the energy aware baseline.

Second, we design a prefetching approach that utilizes an extra disk to accommodate prefetched data. Third, we develop a second prefetching strategy that makes use of an existing disk in the parallel disk system as a buffer disk. Compared with the first prefetching scheme, the second approach lowers the capacity of the parallel disk system. However, the second approach is more cost-effective and energy-efficient than the first prefetching technique.

Finally, we quantitatively compare both of our prefetching approaches against two conventional strategies including a dynamic power management technique and a non-energy-aware scheme. The results obtained from the Figure below held the number of disks to 4 and also kept the data size of each request at 275MB. We have omitted the results of the non-energy aware approach, since they are constant and higher than the energy aware strategy. As expected the performance of both of our strategies were lowered when the hit rate was decreased. This is expected since our motivational example demonstrated a best case scenario. Disk sleep times are lowered once a miss is encountered. This is due to the fact that a disk has to wake up to serve the request. This will increase the energy consumption of disks that have to serve the missed requests. This leads to an increase in the total energy consumption of the entire system. Our buffered large-scale parallel disk system is still able to consume less energy than the energy aware approach. The energy aware and non-energy aware disk systems are not affected by buffer disk miss rates.

The first buffer disk approach begins to approach the same level of performance as the energy aware strategy. It is only able to save 10% energy over the energy aware strategy when the hit rate is 75%. This is because adding the extra disk puts extra energy requirements on the system, and lowering the hit rate further impacts the energy benefits of the first strategy. The second buffered disk approach is still able perform 25% better than the energy aware approach. This is because there is not the extra energy penalty of adding an extra disk. The capacity of your disk system will be lowered using this approach.

The hit rate becomes a very important factor in the performance of our approaches. If the buffer disk is constantly missing requests then both strategies will eventually downgrade to the energy aware approach. Fortunately applications have been documented to request 20% of the data available 80% of the time. Our heuristic based approach would work considerably well in this case. This is modeled by the 80% hit rate. The buffered disk approach one and two are able to save 12% and 26% energy over the energy aware strategy when the hit rate is 80%. Similarly, they are able to save 37% and 47% of the total energy compared to the non-energy aware approach.

The first buffer disk approach downgrades more quickly than the second approach as the hit rate is decreased as compared to the energy aware approach. This is not that great of a concern since the first strategy is still able to have a positive impact on the reliability of the as compared to the energy aware approach. The first buffer disk approach is still able to produce significant energy savings over the non-energy aware approach without compromising the reliability of the system. The energy savings performance of the second approach does not diminish as quickly as the first approach, but there will be an impact on the capacity of the system. The second approach is also able to reduce the number of state transitions.

From the above Figure we are able to see that the non-energy aware approach wastes a considerably larger amount of energy as compared to all of the energy aware approaches. This is expected since the non-energy aware approach is not able to place disks in the standby mode. Buffer strategy 1 was able to produce a 12% increase in energy savings over the energy aware strategy when 10 disks were simulated. Similarly, buffer strategy performed even better with an 18% increase. This is expected again because of the energy overhead adding an extra disk Buffer Strategy 1 requires.

[Xie and Sun, 2008a] Mainstream energy conservation schemes for disk arrays inherently affect the reliability of disks. A thorough understanding of the relationship between energy saving techniques and disk reliability is still an open problem, which prevents effective design of new energy saving techniques and application of existing approaches in reliability-critical environments. As one step towards solving this problem, we investigated an empirical reliability model, called Predictor of Reliability for Energy Saving Schemes (PRESS). The architecture of the PRESS model is given below:

Fed by three energy-saving-related reliability-affecting factors, operating temperature, utilization, and disk speed transition frequency, PRESS estimates the reliability of entire disk array. In what follows, we present two 3-dimennsional figures to represent the PRESS model at operating temperature 40 C (Figure 5a) and 50 C (Figure

5b), respectively.

Further, we developed a new energy saving strategy with reliability awareness called Reliability and Energy Aware Distribution (READ) is developed in the light of the insights provided by PRESS. Experimental results demonstrate that compared with existing energy saving schemes, MAID and PDC, READ consistently performs better in performance and reliability while achieving a comparable level of energy consumption.

[Xie and Sun, 2007] Many real-world applications like Video-On-Demand (VOD) and Web servers require prompt responses to access requests. However, with an explosive increase of data volume and the emerging of faster disks with higher power requirements, energy consumption of disk based storage systems has become a salient issue. To achieve energy-conservation and prompt responses simultaneously, in this study we propose a novel energy-saving data placement strategy, called Striping-based Energy-Aware (SEA), which can be applied to RAID-structured storage systems to noticeably save energy while providing quick responses. Further, we implement two SEA-powered RAID-based data placement algorithms, SEA0 and SEA5, by incorporating the SEA strategy into RAID-0 and RAID-5, respectively. Extensive experimental results demonstrate that (see the three figures below) compared with three well-known data placement algorithms Greedy, SP, and HP, SEA0 and SEA5 reduce mean response time on average at least 52.15% and 48.04% while saving energy on average no less than 10.12% and 9.35%, respectively.

[Xie, 2007] and [Madathil et al., 2008] The problem of statically assigning nonpartitioned files in a parallel I/O system has been extensively investigated. A basic workload characteristic assumption of existing solutions to the problem is that there exists a strong inverse correlation between file access frequency and file size. In other words, the most popular files are typically small in size, while the large files are relatively unpopular. Recent studies on the characteristics of web proxy traces suggested, however, the correlation, if any, is so weak that it can be ignored.

Hence, in this part of study, we raised the following two questions. First, can existing algorithms still perform well when the workload assumption does not hold? Second, if not, can one develop a new file assignment strategy that is immune to the workload assumption? To answer these questions, in this project we first evaluated the performance of three well-known file assignment algorithms with and without the workload assumption, respectively. Next, we developed a novel static file assignment strategy for parallel I/O systems, called static round-robin (SOR), which is immune to the workload assumption.

The above four figures show the simulation results for the four algorithms on a parallel I/O disk array with 16 disk drives. We observe that SOR consistently outperforms the three exiting approaches in terms of mean response time. This is because SOR considers both minimizing variance of service time for each disk and fine-tuning load balancing degree. Consequently, the sorted files were continuously assigned to disks such that a more evenly distributed workload allocation scheme was generated. SP takes the second place in mean response time metric, which is consistent with our expectation because it is one of the best existing static file assignment heuristics. To clearly demonstrate the performance improvement, Fig. 2b provides mean response time decrease gained by SOR compared with Greedy, SP, HP, respectively. In particular, SOR can reduce mean response time on average by 1118.3, 1052.8, and 269.6 seconds, compared with HP, Greedy, and SP, respectively. An interesting observation is that the mean response time improvement becomes more significant when the overall workload represented by the aggregate access rate increases. The implication is that SOR exhibits its strength in situations where system workload is heavy. In terms of mean slowdown, SOR also performs best among the four heuristics (Fig. c), which is consistent with the results shown in Fig. a. Since the total workload is relatively heavy, the mean disk utilization in Fig. d quickly arises to 1 when aggregate access rate is larger than 25 (1/second).

[Zong et al., 2008] [Zong et al., to be published] To conserve energy, BUD makes an effort to place most data disks will run in the low power state, thereby directing most traffic to a limited number of buffer disks. This can potentially make the buffer disks overloaded and become the performance bottleneck. Load balancing is one of the best solutions for the inherent shortcoming of the BUD architecture. Basically, there are three types of load balancing strategies called non-random load balancing, random load balancing, and redundancy load balancing. Sequential mapping belongs to non-random load balancing, because the buffer disks have fixed mapping relationships with specific data disks. The round-robin mapping is a typical random load balancing strategy by allocating data to each buffer disk with equal portions and in order. Redundancy load balancing strategies for storage systems include EERAID, eRAID, and RIMAC. We designed a heat-based load balancing strategy, which is also a random load balancing strategy. The primary objective of our strategy is to minimize the overall response time of disk requests by keeping all buffer disks equally loaded.

In contrast to sequential and round robin mapping algorithms, a heat-based mapping strategy is proposed to achieve load balancing among buffer disks. The basic idea of heat-based mapping is that blocks in data disks are mapped to buffer disks based on their heat, i.e. frequencies of accesses. Our goal is to make the accumulated heat of data blocks allocated to each buffer disk the same or close to this ideal situation. In other words, the temperature, or the workload of each buffer disk should be the same. The temperature of a buffer disk is the total heat of all blocks existing in the buffer disk. For example, suppose all blocks have the same data size, the heat of blocks 1-6 is 5, 4, 1, 2, 1, and 2, respectively. Then, block 1 is cached to buffer disk 1, blocks 2 and 3 are copied to buffer disk 2 and blocks 4, 5 and 6 are mapped to buffer disk 3. With this mapping in place, the temperature of each buffer disk is 5.

This algorithm will periodically collect the requests waiting in the queue, analyze the target block of each read request, and calculate the heat of each unique block. If the target block cannot be found in the buffer disk, the controller initiates a data miss command. This in turn will wake up the corresponding data disk in order to copy the block to the buffer disk that has the lowest temperature. In a special case, the selected buffer disk may not have free space to store a new data block. The controller will seek the next buffer disk with a temperature that is higher than the initial buffer disk selected, but still lower than any other buffer disks. In the worst case, no candidate buffer disk will be found because all buffer disks are full. A data replacement function using the Least Recently Used (LRU) algorithm will be executed to evict some existing data blocks. If the target block has already been cached in one of the buffer disks, then that buffer disk will serve the corresponding request. Once the algorithm has made the decision how to dispatch these requests, the block heat and buffer disk temperature need to be recalculated and updated accordingly. Since this is an online algorithm, the decision made at the current time period relies on the heat and temperature information collected in the last time period.

This set of experimental results aims at evaluating the energy efficiency of the buffer disk based parallel storage systems. To fairly compare the results, we generated and executed a large number of requests and simulated both large reads (average data size is 64MB) and small reads (average data size is 64KB). The following two Figures plot the total energy consumption of NO-buffer and Heat-BUD running 2000, 5000, 10000, and 20000 large read requests and small read requests, respectively. There are three important observations here. First, the BUD can significantly conserve energy compared with No-Buffer parallel storage systems. Second, the more requests BUD serves, the more potential power savings is revealed. For example, BUD outperforms No-Buffer in terms of energy conservation by 75.83%, 77.89%, 80.18% and 81.16% for 2000, 5000, 10000, and 20000 large reads, respectively. This is expected because more requests lead to more opportunities for BUD to keep the data disks in the sleep state. Third, BUD performs better for small reads (average 84.4% improvement) than large reads (average 78.77% improvement). The reason is that BUD consumes more energy when moving large data blocks to buffer disks.

Energy consumption for large reads

Energy consumption for small reads

In this part of the study, we evaluated the load balancing ability of the heat-based algorithm. Recall that the temperature of a buffer disk clearly indicates how busy it is. The Figure below records the temperature of three buffer disks when BUD is processing 800 requests. We can see that the three temperature curves merge together most of the time, which means that the three buffer disks are almost equally loaded for most of the simulation time.

Temperature tracking trace

In order to identify the information hidden in the above Figure and understand how the dynamic load balancing works, we plot the initial stage, intermediate stage of the temperature tracking trace in the following two Figures. At the initial stage, the three buffer disks are not load balanced. Buffer disk 2 is the busiest disk and buffer disk 1 is lightly loaded. Therefore, the heat-based algorithm will keep allocating requests to buffer disk 1. We observe that the temperature of buffer disk 1 keeps growing and it catches buffer disk 3 first. After that, the temperatures of buffer disk 1 and 3 cross-rise for a while and then they catch buffer disk 2. At this point, the system is load balanced for the first time. Fig. 5 shows that the entire system is perfectly load balanced in the intermediate stage because the temperatures of three buffer disks rise in turns.

Temperatures in initial stage

Temperatures in intermediate stage

To compare the load balancing efficiency of sequential mapping, round robin mapping, and heat-based mapping, we tested 2500 requests with average data size of 4MB using these three mapping strategies. The simulation results depicted in the Figure below prove that our heat-based mapping is the most efficient algorithm that achieves load balancing. In addition, the random mapping method (e.g., round robin mapping) outperforms non-random mapping strategies (e.g., sequential mapping) in terms of load balancing.

Load balancing comparison

[Manzanares et al., 2008a] The prefetching algorithm uses the frequency that a block is requested as a heuristic. The first step of the algorithm iterates over all of the requests and counts the references for each unique block. Then it sorts the list of unique blocks by the number of references to each block. At this point the algorithm puts the highly requested blocks into the buffer until it is full.

The last part of the algorithm also iterates over all requests trying to figure out how long each disk can sleep. If a block requested for a disk is in the buffer disk, the corresponding data disk can sleep longer. The buffer disk handles the request and the distance between requests on the same disk becomes cumulative. If a requested block is not in the buffer the disk must be woken up to serve the request, this is handled by the power management mechanism. The distance is then set to zero since the disk had to be woken up. Using the frequently accessed heuristic the PRE-BUD strategy should have a small performance impact on the system. Almost all steps of the algorithm run linearly with respect to the number of requests. The only step that is not linear is the phase that sorts the list of requests according to their frequency. Sorting is a common procedure and is known to have a best-case run-time of O(nlogn), where n is the number of data request to be sorted. The PRE-BUD strategy is able to have a run time of O(n + nlogn) using an efficient sorting algorithm. The PRE-BUD strategy is not assumed to be optimal, since the requested blocks are sorted using their frequency. The frequency is used as a heuristic to select blocks to be placed in the buffer. An optimal strategies goal would be to select the requests to be placed in the buffer that produce the largest impact on the standby time of disks. The standby time of each disk is directly related to the energy savings of the system.

The results displayed in the Figure below held the number of disks to 4 and also kept the data size of each request at 275MB. We have omitted the results of the non-energy aware approach, since they are constant and higher than the energy aware strategy. Buffer Strategy 1 adds an extra disk and Buffer Strategy 2 uses an existing disk as the buffer disk. As expected the performance of both of our strategies were lowered when the hit rate was decreased. Disk sleep times are lowered once a miss is encountered. This is due to the fact that a disk has to wake up to serve the request. This will increase the energy consumption of disks that have to serve the missed requests. This leads to an increase in the total energy consumption of the entire system. Our buffered large-scale parallel disk system is still able to consume less energy than the energy aware approach. The energy aware and non-energy aware disk systems are not affected by buffer disk miss rates.

Energy Savings (J) vs. Hit Rate

As the hit rate is lowered, the first buffer disk approach begins to approach the same level of performance as the energy aware strategy. It is only able to save 10% energy over the energy aware strategy when the hit rate is 75%. This is because adding the extra disk puts extra energy requirements on the system, and lowering the hit rate further impacts the energy benefits of the first strategy. The second buffered disk approach is still able perform 25% better than the energy aware approach. This is because there is not the extra energy penalty of adding an extra disk. The capacity of your disk system will be lowered using this approach.

The hit rate becomes a very important factor in the performance of our approaches. If the buffer disk is constantly missing requests then both strategies will eventually downgrade to the energy aware approach. Fortunately applications have been documented to request 20% of the data available 80% of the time. Our heuristic based approach would work considerably well in this case. This is modeled by the 80% hit rate. The buffered disk approach one and two are able to save 12% and 26% energy over the energy aware strategy when the hit rate is 80%. Similarly, they are able to save 37% and 47% of the total energy compared to the non-energy aware approach.

The first buffer disk approach downgrades more quickly than the second approach as the hit rate is decreased as compared to the energy aware approach. This is not that great of a concern since the first strategy is still able to have a positive impact on the reliability of the system as compared to the energy aware approach. The first buffer disk approach is still able to produce significant energy savings over the non-energy aware approach without compromising the reliability of the system. The energy savings performance of the second approach does not diminish as quickly as the first approach, but there will be an impact on the capacity of the system. The second approach is also able to reduce the number of state transitions.

Fig. 8 Energy Savings (J) vs. Number of Disks

From the above Figure we are able to see that the non-energy aware approach wastes a considerably larger amount of energy as compared to all of the energy aware approaches. This is expected since the non-energy aware approach is not able to place disks in the standby mode. Buffer strategy 1 was able to produce a 12% increase in energy savings over the energy aware strategy when 10 disks were simulated. Similarly, buffer strategy performed even better with an 18% increase. This is expected again because of the energy overhead adding an extra disk Buffer Strategy 1 requires. Our approach produces promising results as the number of disks is increased. This is an important observation, since our target system is a large-scale parallel system. This leads us to believe our system will produce energy benefits regardless of the number of disks in a system.

[Bellam et al., 2008] RAID 1 is popular and is widely used for disk drives. RAID 1 is implemented with a minimum of two disks, which are the primary and back disks. Initially the data is stored to the primary disk and then it is mirrored to the backup disk. This mirroring helps to recover the data when there is a failure in the primary disk. It also helps to increase the performance of the RAID 1 system by sharing the workload between the disks. We considered RAID 1 for all of our experiments.
The processor in the system generates the I/O stream, which is queued to the buffer. The utilization of the disk is calculated using the request arrival rate. Please refer to Section 3 for details of the description for the disk utilization model.
It should be noted that all requests here are considered as read requests. At any given point of time the disks can be in the following three states.
? State 1: Both the disks in sleep mode
? State 2: Primary disk active and backup disk in sleep mode
? State 3: Both the disks in active mode and share the load.
Let us consider that the disks are in state 1 at the beginning. Once the utilization is calculated, it is compared with the safe utilization zone range. If the calculated value falls below the range then disks stay in state 1. If the calculated value is within the range, then the primary disk is made active while the backup disk continues to stay in the sleep mode. This represents a transition to state 2. If the calculated value is beyond the range then both the disks are made active and both of them share the load, which corresponds to state 3. Transition of states from one power mode to another involves disk spin up and/or spin down. The disk spin ups and spin downs also consume a lot of energy.

RAID 1 is used in our experiments. RAID 1 uses a minimum of two disks,

one as a primary and one as the backup. We conducted the experiments on three types of disks from IBM.
The experimental results are compared against two traditional state of the art methods. In the first method, load balancing, both the disks are always made active. Load balancing achieves very high performance because both the disks share the load. The second method, traditional method, is where the primary disk is made always active and the backup disk is kept in sleep mode. The backup disk is made active only when the utilization of the primary disk exceeds 100%, also known as saturation. In what follows, we term our approach as RAREE (Reliability aware energy efficient approach).

Energy consumed by a 2 year old disk

The experimental data generated from the simulations is plotted in the above Figure, which represents the energy consumed by the 2 year old disk respectively. RAREE is compared against load balancing and the traditional method. From the above Figure it is observed that for the IBM 36Z15 disk the power consumed by RAREE falls in between the load balancing and traditional techniques. Even for the IBM 73LZX the trend is similar, but the difference in values is not as high as IBM 36Z15.
For the IBM 40GNX the power consumed by RAREE is smaller than the traditional and load balancing power consumption values because disk spin up and spin down values are much smaller for the IBM 40GNX when compared with the other two disks. It should be observed that the disk spin down and disk spin up values play a vital role in the energy consumption.

Energy consumed by a IBM 40GNX disk with different ages

The above Figure shows the performance of RAREE on Travelstar disks of different ages. It can be observed from figure that RAREE consumes less energy when compared to traditional and load balancing.

Impact of spin up power on energy

The Figure above shows the effects on energy when the spin up energy is varied for a 2 year old disk. The RAREE energy consumption falls in between traditional and load balancing techniques. Though the energy consumed by RAREE is a little higher than traditional technique, here we are also gaining good amount reliability.

Impact of active power on energy

The above Figure shows the change in energy as the active power is varied. Here RAREE energy consumption is definitely less than the two existing techniques, because RAREE makes the disks go to sleep mode as soon there are no requests unlike the other techniques. When idle power is changed unlike the active power the energy consumed by the RAREE again falls between the two techniques. This is because RAREE makes the system go to sleep mode very often depending on the conditions. It should be observed here though the RAREE consumed a bit higher energy than traditional it can be neglected as we are achieving reliability.

Reliability vs. disk ages

The above Figure is a very important graph here it shows the reliability in terms of annual failure rate percentile. It can be observed from the graph that RAREE achieves a very high reliability when compared to load balancing and traditional. Only one bar is shown for load balancing and traditional techniques because both have the same reliability levels as they don't pay special attention to reliability. We also found an interesting observation that when RAREE is applied to IBM40GNX, which is a travelstar, it definitely consumes much less energy than the other two Ultrastars, which are high performance disks. This makes it clear that RAREE gives best results when it is used on mobile disks instead of high performance disks. This doesn't limit the usage of RAREE to mobile disks because though Ultrastar consumes a little more energy than traditional technique we still get a good reliability at a marginal cost of energy. Simulation results prove that on an average roughly 20% of energy can be saved when RAREe is used instead of load balancing. When RAREe is used instead of the traditional method an excess of 3% of energy is saved, it is not a very significant amount but along with a very little energy saving we are also achieving high reliability which makes it significant.

[Roth et al., 2008] Energy conservation has become a critical problem for real-time embedded storage systems. Although a variety of approaches to reducing energy consumption has been extensively studied, energy conservation for real-time embedded storage systems is still an open problem. In this research, we propose an energy management strategy (IBEC) using I/O burstiness for real time embedded storage systems. Our approach aims at blending the IBEC energy management strategy with low level disk scheduling mechanism to conserve energy consumed by storage systems.

The above Figure shows the power consumption of these four algorithms when average request deadline varies from 75 ms to 25000 ms. We observe from this Figure that each of the four algorithms consumes the same amount of power at the maximal level when the average request deadline is less than 3500 ms. This is because the hard-disk has to be kept active all the time to service the arrival disk requests which have very tight deadlines. In other words, there is no opportunity for

IBEC to conserve some power. Therefore, IBEC gracefully degraded to existing power-aware scheduling algorithms like DP-EDF and PA-EDF. When the average request deadline is equal to or larger than 3500 ms, however, IBEC starts to conserve some energy while the three baseline algorithms remain the same performance in power consumption. We attribute the PF improvement of IBEC over the three baseline algorithms to the fact that IBEC judiciously employ the loose deadlines to conserve some energy. More interestingly, the improvement of IBEC over the three existing schemes in terms of PF is more pronounced when the deadline becomes looser for IBEC can further improve its power consumption performance when more slack time is available. On average, IBEC can save 10.8% power compared with the baseline algorithms.

The above Figure plots GR of the four algorithms when the deadline is increased from 75 to 25000 ms. It reveals that IBEC performs exactly the same, with respect to GR, as all the rest approaches when the deadline is less than 1000 ms. The reason is that the relatively high workload along with the tight deadlines make IBEC only concentrate on guaranteeing arrival requests' timing constraints, which have a higher priority than power conservation requirement. However, the GR performance of IBEC suddenly drops off when the deadline is 10,000 ms. In fact, this is an artifact of our specific implementation of the IBEC algorithm. In order to keep simulation times manageable and to closely approximate a real system where no infinite amount of time is available to re-evaluate the schedulability of a queue, we limited the maximum number of requests that IBEC would ensure their deadline constraints. In particular, when the length of the waiting queue of requests is larger than 1,000, our implementation of IBEC will no longer guarantee the schedulability of requests after the 1,000th.

From the following Figure, we can make three important observations. First, all algorithms perform identically in power consumption under the Normal Distribution. Second, the three power-aware algorithms noticeably outperform the EDF scheme, which has no power-awareness at all, when the Sparse Distributions were applied. This is because the nature of the Sparse Distribution decides a relatively large time interval between two continuous disk requests, which in turn gives the three power-aware algorithms chances to switch the hard-disk to 'sleep' mode to save energy. Furthermore, IBEC slightly outperforms DP-EDF and PA-EDF, two na?ve power-aware algorithms. The rationale behind this phenomenon is that IBEC can make most use of the slack time of each arrival request. Put it in another way, IBEC only wakes up the disk at the last second from which all arrived requests' deadlines can still be met, while DP-EDF only lets the disk sleep for a fixed period of time no matter whether a request is waiting for service or not. Third, for clustered workloads, IBEC and the two naive power-aware techniques perform comparably, and they both significantly perform better than EDF. This is due to the fact that IBEC, DP-EDF, and PA-EDF can put the disk to 'sleep' status completely between clusters of requests. Thus, the three power-aware algorithms can substantially save power compared with EDF. The reason why IBEC ties with DP-EDF and PA-EDF is that the performance improvement of IBEC in terms of power consumption essentially depends on slack times of arrival requests rather than the arrival patterns.

The results reported in the Figure below reveal that all of the four algorithms deliver a 100% guarantee ratio under the Sparse Distribution. The reason for this is that the average request deadline is generally much shorter than the sparse idle threshold, which means that even though IBEC aggressively slows down the processing pace of disk requests, their deadlines can still be satisfied. When we applied a Cluster Distribution pattern, the performance of the four algorithms goes down when the parameters of the Cluster Distribution increase. This is because there were a large number of requests arrived during a burst of incoming requests. Consequently, all the algorithms can only guarantee the deadlines of a small part of them.

[Liu et al., 2008a] Reducing energy consumption has become a pressing issue in cluster computing systems not only for minimizing electricity cost, but also for improving system reliability. Therefore, it is highly desirable to design energy-efficient scheduling algorithms for applications running on clusters. In this project, we address the problem of non-preemptively scheduling mixed tasks on power-aware clusters. We developed an algorithm called Power Aware Slack Scheduler (PASS) for tasks with different priorities and deadlines. PASS attempts to minimize energy consumption in addition to maximizing the number of tasks completed before their deadlines. To achieve this goal, high-priority tasks are scheduled first in order to meet their deadlines. Moreover, PASS explores slacks into which low-priority tasks can be inserted so that their deadlines can be guaranteed. The dynamic voltage scaling (DVS) technique is used to reduce energy consumption by exploiting available slacks and adjusting appropriate voltage levels accordingly.

The following Figure shows the total energy consumed to execute 1600 tasks. We observe that PASS saves up to 60 percent of the energy over CC-EDF. The reason that PASS can achieve such significant energy savings is because PASS creates large integrated slacks by scheduling tasks to the latest possible start time. CC-EDF schedules tasks according to the rule of Minimum Completion Time (MCT) which always schedules a task to its earliest possible start time. In this way, EDF hardly leaves any slack time that may be used by the DVS technique.

Total energy consumption (Execute 1600 tasks)

Hard task acceptance ratio.

Now we compare the performance of PASS against CC-EDF. We performed simulations for different task loads in order to examine the performance consistency. With respect to Hard Task Acceptance Ratio, from the above Figure we observe that PASS yields 10% better performance on average than CC-EDF. When the number of tasks is below 6400, PASS guarantees that all hard tasks can meet their deadlines. As the number of tasks further increases, PASS is still able to schedule most of the hard tasks while EDF is no longer able to reach similar performance level. The reason is because PASS always schedules hard tasks first, which helps meet hard tasks' deadlines. CC-EDF does not consider task priority, which results in a number of un-schedulable hard tasks.

The Figure below shows both algorithms can not schedule all tasks when the number of tasks exceeds 3200. It becomes unfair if we compare two algorithms using the Total Energy Consumption metric, since the number of accepted tasks by using PASS is different from the number by using CC-EDF. Instead, we use the Energy Consumption per Task as the metric. In this way, we are able to study the effect brought by the number of tasks on energy consumption.

Overall acceptance ratio.

As shown in the following Figure, PASS consistently performs better than CC-EDF with respect to Energy Consumption per Task. It is again because PASS decreases the processor speed for each task by utilizing the corresponding slack time. An interesting observation is that as the number of tasks increases, the Energy Consumption per Task achieved by PASS also increases. Once there are more incoming tasks, less slack times will be available since more tasks need to be scheduled within those slack times.

Energy consumption per task ( )

[Nijim et al., 2008a] 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 a variety of parallel disk systems were developed to achieve high performance and energy efficiency, most existing parallel disk systems lack an adaptive way to conserve energy in dynamically changing workload conditions. To solve this problem, we develop an adaptive energy-conserving algorithm, or DCAPS, for parallel disk systems using the dynamic voltage scaling technique that dynamically choose the most appropriate voltage supplies for parallel disks while guaranteeing specified performance (i.e., desired response times) for disk requests.

The DCAPS algorithm aims at judiciously lower the parallel disk system voltage using dynamic voltage scaling technique or DVS, thereby reducing the energy consumption experienced by disk requests running on parallel disk systems. The processing algorithm separately repeats the process of controlling the energy by specifying the most appropriate voltage for each disk request. Thus, the algorithm is geared to adaptively choose the most appropriate voltage for stripe units of a disk request while warranting the desired response time of the request.

The above two Figures plot the satisfied ratios, normalized energy consumption, and energy conservation ratio of the parallel disk systems with and without DCAPS. Figs 3(a) reveals that the DCAPS scheme yields satisfied ratios that are very close to those of the parallel disk system without employing DCAPS. This is essentially because DCAPS endeavors to save energy consumption at the marginal cost of satisfied ratio. More importantly, Figs. 3(b) and 4 show that DCAPS significantly reduces the energy dissipation in the parallel disk system by up to 71% with an average of 52.6%. The improvement in energy efficiency can be attributed to the fact that DCAPS reduces the disk supply voltages in the parallel disk system while making the best effort to guarantee desired response times of the disk requests. Furthermore, it is observed that as the disk request arrival rate increases, the energy consumption of the both parallel disk systems soars.

The Figure below shows that as the load increases, the energy conservation ratio tends to decrease. This result is not surprising because high arrival rates lead to heavily utilized disks, forcing the DCAPS to boos disk voltages to process larger number of requests within their corresponding desired response times. Increasing number of disk request and scaled-up voltages in turn give rise to the increased energy dissipations in the parallel disk systems.

[Nijim et al., 2008b] Cluster storage systems have emerged as high-performance and cost-effective storage infrastructures for large-scale data-intensive applications. Although a large number of cluster storage systems have been implemented, the existing cluster storage systems lack a means to optimize quality of security in dynamically changing environments. We solve this problem by developing a security-aware cache management mechanism (or CaPaS for short) for cluster storage systems. CaPaS aims at achieving high security and desired performance for data-intensive applications running on clusters. CaPaS is used in combination with a security control mechanism that can adapt to changing security requirements and workload conditions, thereby providing high quality of security for cluster storage systems. CaPaS is comprised of a cache partitioning scheme, a response-time estimator, and an adaptive security quality controller. These three components help in increasing quality of security of cluster storage systems while allowing disk requests to be finished before their desired response times. To prove the efficiency of CaPaS, we simulate a cluster storage system into which CaPaS, eight cryptographic, and seven integrity services are integrated. Empirical results show that CaPaS significantly improves overall performance over two baseline strategies by up to 73% with an average of 52% (see the above four Figures).

[Liu et al., 2008b]

Although data duplications may be able to improve the performance of data-intensive applications on data grids, a large number of data replicas inevitably increase energy dissipation in storage resources on the data grids. In order to implement a data grid with high energy efficiency, we address in this study the issue of energy-efficient scheduling for data grids supporting real-time and data-intensive applications. Taking into account both data locations and application properties, we design a novel Distributed Energy-Efficient Scheduler (or DEES for short) that aims to seamlessly integrate the process of scheduling tasks with data placement strategies to provide energy savings. DEES is distributed in the essence - it can successfully schedule tasks and save energy without knowledge of a complete grid state. DEES encompasses three main components: energy-aware ranking, performance-aware scheduling, and energy-aware dispatching. By reducing the amount of data replications and task transfers, DEES effectively saves energy.

The following Figure shows the performance of DEES using different (&#949;, &#956;) value pairs with respect to Guarantee Ratio. It is observed that DEES (2, 1) gives the best performance. This is because DEES (2, 1) takes both goals of meeting deadline and saving energy into account, and put more weight onto the deadline meeting part. Neighbors that can schedule more tasks are given preference. We conclude it is better to give preference to neighbors that can schedule more tasks while consuming satisfactory amount of energy.

Guarantee Ratio by ranking coefficients
With respect to Normalized Average Energy Consumption, as shown in the following Figure, we observe that DEES (2, 1) consumes the least amount of energy while DEES (0, 1) consumes the most. DEES (2, 1) considers both energy consumption and deadline constraints when dispatching tasks to neighbors. Doing so can reduce the energy cost per task. On the other hand, DEES (0, 1) schedules fewer tasks since it only cares about energy consumption when dispatching tasks. Moreover, given that more tasks miss their deadlines at each site; additional data replications may be needed. Therefore it relatively consumes more energy to replicate data and transfer the tasks.

Normalized Average Energy Consumption by ranking coefficients

In this experiment set, we compared the performance of DEES with Close-to-Files and Performance-driven algorithms under different task loads. From the Figure below, we observe that DEES yields better performance than Close-to-Files and achieves the same performance level as the Performance-driven algorithm does. The Performance-driven algorithm always schedules a task to a globally best resource that gives the best performance. Since it only focuses on performance but not other factors such as data locality, it yields very good performance with respect to Guarantee Ratio. But the fact that DEES gives similar performance as the Performance-driven algorithm is importance. Thus, DEES not only reduces energy consumption, but it does so without degrading the Guarantee Ratio. One reason is because DEES always schedules tasks with shorter deadlines first. The final criteria for judging whether a task can be scheduled are the task deadlines. Scheduling those tasks with shorter deadlines first makes more tasks schedulable. Moreover, DEES is fully distributed, which is expected to improve the performance when compared to a centralized algorithm, such as the Performance-driven algorithm, especially when the task load is heavy. Given that DESS is fully distributed, while Close-to-Files and Performance-driven algorithms need knowledge of a complete state of the grid, the results make DEES more favorable.

Guarantee Ratio by task loads

With respect to Normalized Average Energy Consumption, as shown in the Figure below, we see that DEES consumes much less energy per task than Close-to-Files does. On average DEES saves over 35% of energy consumed when compared to the other algorithms. This is because DEES considers the energy consumed to transfer both tasks and data during dispatching. Moreover, DEES groups tasks according to their data accesses and processes tasks on a group basis. Doing so limits the number of data replicas. This is because whenever data is replicated to a remote site, DEES always maximizes utilization of the data replicated by scheduling as many tasks as possible to that remote site. On the other hand, Close-to-Files makes dispatching decisions on a single task basis, which may result in unnecessary data replications. Furthermore, since DEES schedules more tasks than Close-to-Files does, the energy cost per task is expected to be less. The Performance-driven algorithm consumes the most amount of energy due to the fact that it is a greedy algorithm that always schedules a task to a resource giving the best performance, regardless of how much data are needed to be replicated and transferred.

Normalized Average Energy Consumption by task loads

[Ruan et al., 2009] In the past decades, parallel I/O systems have been used widely to support scientific and commercial applications. New data centers today employ huge quantities of I/O systems, which consume a large amount of energy. Most large-scale I/O systems have an array of hard disks working in parallel to meet performance requirements. Traditional energy conservation techniques attempt to place disks into low-power states when possible. In this work we propose a novel strategy, which aims to significantly conserve energy while reducing average I/O response times. This goal is achieved by making use of buffer disks in parallel I/O systems to accumulate small writes to form a log, which can be transferred to data disks in a batch way. We develop an algorithm - dynamic request allocation algorithm for writes or DARAW - to energy efficiently allocate and schedule write requests in a parallel I/O system. DARAW is able to improve parallel I/O energy efficiency by the virtue of leveraging buffer disks to serve a majority of incoming write requests, thereby keeping data disks in low-power state for longer period times. Buffered requests are then written to data disks at a pre-determined time. Experimental results show that DARAW can significantly reduce energy dissipation in parallel I/O systems without adverse impacts on I/O performance.

The Figure below shows the energy consumption and average response time of a parallel disk system with DARAW and the same disk system without DARAW. The results indicate that when we increase SRB, more energy can be saved. The results were expected since when the SRB grows, the system can write more requests into data disks with reduced number of power state transitions. However, we also observe that when the SRB equals to one, the energy consumption is even greater than the disk system without DARAW. This interesting tend can be explained as follows. Our parallel disk system has a buffer-disk layer that also consumes energy. If there is insufficient number of requests written into a data disk when a power-state transition occurs, energy conserved cannot offset energy overhead introduced by the buffer disk. When we did the experiment with a trace generated by increasing values of &#955;, we observe that energy consumptions in both the non-DARAW parallel disk system and the system with DARAW decrease.

IBM 40GNX Travelstar. Energy Consumption and Average Response Time Compare

Note that all the traces have the same number of disk requests. This implies the fact that when &#955; is high, all requests are arriving at the system within a shorter period of time, making all the disks stay in the active state for a shortened time interval. This is the reason behind the result that energy consumption of the system with DARAW when &#955; is set to 0.02 is slightly smaller than that of the system when &#955; is 0.01. However, the power consumption of the non-DARAW disk system is significantly smaller when &#955; is 0.01 as compared to &#955; = 0.02. Once the arrival rate goes up, each data disk in the non-DARAW system has greater probability to receive a request when it is working. Thus, the number of power-state transitions can be noticeably reduced. When &#955; is set to 0.02, there is less of an opportunity to simultaneously save energy and satisfy response times. When we increase the number of buffer disks from 5 to 20, DARAW can conserve energy while guaranteeing reasonably short response times. An appealing result shown in the above Figure is that compared with the parallel I/O system without DARAW, our approach not only achieves significant energy savings, but also reduces response times. In DARAW, the response time is the time when a request is written in to a data or buffer disk. Since buffer disks can serve coming requests when data disks are sleeping, the response time can be noticeably shortened.

Our results show that DARAW works well for parallel I/O systems with both high performance disks and mobile disks. DARAW achieves promising results when the arrival rate is low. When the request arrival rate rises, we can either use high-performance hard drives or add more buffer disks to boost I/O performance. If the arrival rate is high, all data disks are busy serving requests, leaving no opportunity to save energy. As the SRB parameter grows, DARAW is given a greater window of opportunity to conserve energy. However, if the SRB is too large, it may cause a 'traffic jam' inside the parallel I/O system with buffer disks.

[Xie and Sun, 2008a] Mainstream energy conservation schemes for disk arrays inherently affect the reliability of disks. A thorough understanding of the relationship between energy saving techniques and disk reliability is still an open problem, which prevents effective design of new energy saving techniques and application of existing approaches in reliability-critical environments. As one step towards solving this problem, we investigated an empirical reliability model, called Predictor of Reliability for Energy Saving Schemes (PRESS). The architecture of the PRESS model is given below:

Fed by three energy-saving-related reliability-affecting factors, operating temperature, utilization, and disk speed transition frequency, PRESS estimates the reliability of entire disk array. In what follows, we present two 3-dimennsional figures to represent the PRESS model at operating temperature 40 C (Figure 5a) and 50 C (Figure 5b), respectively.

Further, we developed a new energy saving strategy with reliability awareness called Reliability and Energy Aware Distribution (READ) is developed in the light of the insights provided by PRESS. Experimental results demonstrate that compared with existing energy saving schemes, MAID and PDC, READ consistently performs better in performance and reliability while achieving a comparable level of energy consumption.

[Xie and Wang, 2008] High performance, highly reliable, and energy-efficient storage systems are essential for mobile data-intensive applications such as remote surgery and mobile data center. Compared with conventional stationary storage systems, mobile disk-array-based storage systems are more prone to disk failures due to their severe application environments. Further, they have very limited power supply. Therefore, data reconstruction algorithms, which are executed in the presence of disk failure, for mobile storage systems must be performance-driven, reliability-aware, and energy-efficient. In this project we developed a novel reconstruction strategy, called multi-level caching-based reconstruction optimization (MICRO), which can be applied to RAID-structured mobile storage systems to noticeably shorten reconstruction times and user response times while saving energy. MICRO collaboratively utilizes storage cache and disk array controller cache to diminish the number of physical disk accesses caused by reconstruction. Experimental results demonstrate that compared with two representative algorithms DOR and PRO, MICRO reduces reconstruction times on average 20.22% and 9.34%, while saving energy no less than 30.4% and 13%, respectively.

[Xie and Sun, 2008b] Contemporary disk arrays consist purely of hard disk drives, which normally provide huge storage capacities with low-cost and high-throughput for data-intensive applications. Nevertheless, they have some inherent disadvantages such as long access latencies, fragile physical characteristics, and energy-inefficiency due to their build-in mechanical and electronic mechanisms. Flash-memory based solid state disks, on the other hand, although currently more expensive and inadequate in write cycles, offer much faster read accesses and are much more robust and energy efficient. To combine the complementary merits of hard disks and flash disks, in this study we propose a hybrid disk array architecture named HIT (hybrid disk storage) for data-intensive applications. Next, a dynamic data redistribution strategy called PEARL (performance, energy, and reliability balanced), which can periodically redistribute data between flash disks and hard disks to adapt to the changing data access patterns, is developed on top of the HIT architecture. Comprehensive simulations using real-life block-level traces demonstrate that compared with existing data placement techniques, PEARL exhibits its strength in both performance and energy consumption without impairing flash disk reliability.

ECOS: An Energy-Efficient Cluster Storage System [Ruan et al. 2009a] The disks we apply in our prototype are different for the purpose of testing performance of DARAW in different devices. The traces we use are synthetic traces. The arrival rates are generated by exponential distribution. To reflect the real world cases, the traces contain burstnesses and idle time gap inside. Each burstness contains a group of requests whose arrival rates based on exponential distribution. The most appropriate time for buffer disk to dump data to data disk is during idle time gaps. Hence, we will test traces with different idle time gaps and analyse the results.

In the above figure, up to more than 20% energy could be conserved when idle time gap between each request burstness is 300 seconds or 200 seconds. When the idle time between each burstness is as short as 50 seconds, the energy conservation rate is not very obvious by using DARAW strategy to buffer data on buffer disks from energy conservation perspective. In the figure, the Sum of Request in Buffer, or SRB, is another important parameter in our experiment. According to DARAW strategy, if the number of bufferred requests which target at one same data disk equals SRB, then the targetted data disk spins up and those requests will be dumped to it. The data disk will spin down when there is no request writting on it. Basically, the larger SRB we set up in DARAW, the more energy we can conserve in the system. However, trace features also can affect the energy conservation rate. The most appropriate time of dumping data to data disks is during idle time gaps. DARAW only dumps data to data disks when SRB requirements resatisfied. If there are too many dumping operations happen during burstness, energy conservation rate will be reduced.

Because idle time gap is low which means workload is high, so dumping data operation is more likely happen during burstnesses (see the above figure). Node 2 is faster than node 1, so the dumping operations will not extend the processing time of buffer disk. In node 1, when dumping operations happen during burstness, since the disks speed in node 1 is not as fast as node 2, the buffer disk needs more time to finish dumping data during burstness. Hence, node 1 consumes more energy in this experiment.

Performance Evaluation of Energy-Efficient Parallel I/O Systems with Write Buffer Disks [Ruan et al. 2009b] [Ruan et al. 2009c]
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.

The above figure plots energy efficiency and performance of the baseline algorithm applied to a traditional parallel I/O system without buffer disks. Results plotted in Fig. (a) show that the I/O load increases significantly as the arrival rate (i.e., &#955;) grows. For example, 1000 requests are issued to the simulated parallel I/O system within 100,000 milliseconds if &#955; is set to 0.01No./ms., whereas 1000 requests have arrived in the system within 50,000 milliseconds when &#955; is doubled.

An interesting counterintuitive observation drawn from Fig. (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 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. (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 &#955; 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. (d) depicts the energy consumption trend for the IBM Ultrastar and Travalstar disks. In what follows, we describe two important observations. First, Fig. (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 &#955; 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. c). It is worth noting that in each experiment, we fix the total number of requests (e.g, 1000).

HyBUD: An Energy-Efficient Architecture for Hybrid Parallel Disk Systems [Nijim et al. 2009]

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

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

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

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

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 compared 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. 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.

The above figure 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 HYBUD strategy consumes less energy than the other two alternatives strategy. We will discuss each strategy separately. When the disk request is submitted to the flash memory and if there is a miss, then the total energy cost will be the energy cost of read and the energy cost of writing the request into the flash under the assumption that this request will be frequently used in the future, and the cost of keep waking up the corresponding data disk every time the read request is made which leads to a huge energy consumption. For the BUD strategy, it consumes less energy than using only the flash memory strategy. This can be explained by the fact that the BUD strategy when miss occurs, it clusters read requests together if the disk is in sleeping mode. As a result, it provides a long disk idle times.

Finally the HYBUD strategy consumes the least energy. When the read request is submitted to the flash memory, they read request will be served immediately if the data disk is in active mode, otherwise, the read request will be written to the flash assuming that these requests will be frequently used. When the flash is full, the dirty data will be flushed to the buffer disks and all the miss requests will be clustered together, which leads to less energy consumption.

In this experiment we compared the three strategies in term of the size of data block. The above figure 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. In case of small data size, flash memory is no longer able to save energy, because the flash memory keep waking up the disk, which results in huge energy consumption. The HYBUD strategy can save up to 51% over the flash memory and 22% over BUD.

Energy-Aware Prefetching for Parallel Disk Systems [Manzanares et al. 2009]
For our experimental results we implemented a parallel disk simulator in JAVA. The first set of experiments we conducted varied the hit rate and the data size of the requests. The hit rate in these experiments is defined as the percentage of all the requests that can be served by the buffer disk and the data size is defined as the data size of each request. We generated random disk requests and varied the inter-arrival delay of the requests. The inter-arrival rate must be fairly low to produce energy savings or disks will never be placed in the sleep state. If the inter-arrival rate is high all disks must be active to serve the requests. The results of the first set of experiments are summarized in the Figure below.

Total Energy Consumption of Disk System while Data Size is varied for four different values of hit rate: (a) 85 %, (b) 90 %, (c) 95 %, and (d) 100 % hit rate.
There are two main observations we can draw from this figure, one being that as data size increases energy savings increases, and second, as the hit rate is increased energy savings increases. As the data size increases the time to serve the request increases. If multiple requests can be served from the buffer disk than the data disks have a greater opportunity to transition to the sleep state. Similarly as the hit rate increases the buffer disk serves a greater number of consecutive hits allowing data disks to sit idle for longer periods of time. The goal of our energy-efficient prefetcher is to increase the number and length of idle periods to allow a data disk to transition to the sleep state. This can be achieved by increasing the hit rate or increasing the data size of requests. This leads us to believe that many web and multimedia applications would be suitable for our energy saving techniques.

Total Energy consumption for various values of the following disk parameters: (a) power active, (b) power idle, and (c) power standby

The second set of experiments conducted focuses on the impact that varying disk power parameters has on the energy savings. We varied the power characteristics of our simulated IBM36Z15 disk. For each figure we only vary one disk energy parameter. The number of disks was fixed at four and the data size is 25MB. From the above figure we realize that lowering the Power Active, which is the energy consumed while the disk is in the active state, will decrease the energy consumption for all the strategies we compared. Lowering Power Active also impacts the relative energy savings that the PRE-BUD strategies are able to produce. If Power Active is 9.5W PRE-BUD2 saves 15.1 % energy over DPM. If it is increased to 17.5W PRE-BUD2 only saves 13% energy over DPM. Fig. 4 (a) is similar to Fig. 4 (b) but now we see that Power Idle has a greater impact on energy savings as compared to Power Active. If Power Idle, the energy consumed while the disk is idle, is very low PRE-BUD 2 has a negative impact, but if it's increased to 14.2 W PRE-BUD 2 now saves 25 % of energy as compared to DPM. The last set of experiments varied the Power Sleep parameter, which represents the energy consumed while the disk is in the sleep state, also has significant impact on PRE-BUD strategies. The percentage change in energy savings starts at 16.3% and drops to 11.7% with increasing Power Sleep. The results illustrated in Fig. 4 indicate that parallel disks with low active power, high idle power, and low standby power can produce the best energy-savings benefit. This is because PRE-BUD allows disks to be spun down to the standby state during times they would be idle using DPM. The greater the discrepancy between idle and standby power, the more beneficial PRE-BUD becomes.

[Tjioe, Widjaja, Lee, and Xie, 2009] In a completely dynamic environment where a sub-set of files are extremely popular and receive a dominant percentage of user requests, a dynamic file assignment algorithm may no longer be helpful. The reason is that no matter where it places these hot files the load imbalance across the disks cannot be solved. In this situation, file replication techniques can be employed to make replicas for these popular files and to distribute them onto other disks. We developed a new dynamic file assignment strategy called DORA (dynamic round robin with replication), which integrates file replication techniques into file assignment schemes for a user access pattern changing environment. DORA first sorts all files according to file size. Next, it assigns the files to disks in a round-robin fashion so as to distribute the load of all files evenly across all disks. Finally, DORA dynamically keeps track of the load of all files and the load on each disk. For some extremely hot files, it then creates replicas to effectively distribute request accesses on these files across all disks in a disk array. Using extensive simulations, we evaluated the performance of DORA by comparing it with one of the best existing dynamic file assignment algorithms, C-V.

[Xie and Sharma, 2009] Mobile disk arrays, disk arrays located in mobile data centers, are crucial for mobile applications such as disaster recovery. Due to their unusual application domains, mobile disk arrays face several new challenges including harsh operating environments, very limited power supply, and extremely small number of spare disks. Consequently, data reconstruction schemes for mobile disk arrays must be performance-driven, reliability-aware, and energy-efficient. In this paper, we develop a flash assisted data reconstruction strategy called CORE (collaboration-oriented reconstruction) on top of a hybrid disk array architecture, where hard disks and flash disks collaborate to shorten data reconstruction time, alleviate performance degradation during disk recovery. Experimental results demonstrate that CORE noticeably improves the performance and energy-efficiency over existing schemes. Compared with DOR, CORE on average reduces reconstruction duration and mean user response time during reconstruction by 50.4% and 65.3%, respectively. In terms of energy consumption, CORE on average saves energy by 43.4%. Compared with PRO, CORE on average shrinks reconstruction duration and mean user response time during reconstruction by 48.2% and 61.9%, respectively. In addition, CORE saves energy on average by 42.5%.

[Xie and Sun, 2009] The problem of statically assigning non-partitioned files in a parallel I/O system has been extensively investigated. A basic workload characteristic assumption of most existing solutions to the problem is that there exists a strong inverse correlation between file access frequency and file size. In other words, the most popular files are typically small in size, while the large files are relatively unpopular. Recent studies on the characteristics of web proxy traces suggested, however, the correlation, if any, is so weak that it can be ignored. Hence, the following two questions arise naturally. First, can existing algorithms still perform well when the workload assumption does not hold? Second, if not, can one develop a new file assignment strategy that is immune to the workload assumption? To answer these questions, we first evaluate the performance of three well-known file assignment algorithms with and without the workload assumption, respectively. Next, we develop a novel static non-partitioned file assignment strategy for parallel I/O systems, called static round-robin (SOR), which is immune to the workload assumption. Comprehensive experimental results show that SOR consistently improves the performance in terms of mean response time over the existing schemes. Experimental results show that when the distribution of access rates across the files and the distribution of file sizes were inversely correlated with the same skew parameter &#952;, SOR consistently improves the performance of parallel I/O systems in terms of mean response time over three well-known file assignment algorithms. Compared to SP, one of the best existing static non-partitioned file assignment algorithms, SOR obviously achieves improvement in mean response time. When the correlation between file access frequency and file size is negligible, SOR still consistently performs better when file size exhibits a uniform distribution.

References

[Zong et al. 2007a] Z.-L. Zong, M.E. Briggs, N.W. O'Connor, and X. Qin, 'An Energy-Efficient Framework for Large-Scale Parallel Storage Systems,' Proc. 21st Int'l Parallel and Distributed Processing Symp. (IPDPS), 8th IEEE Int'l Workshop Parallel and Distributed Scientific and Engineering Computing, Long Beach, CA, March 2007.

[Zong et al., 2007b] Z.-L. Zong, M.E. Briggs, N.W. O'Connor, X. Qin, M. Alghamdi, and Y.-M. Yang, 'Design and Performance Analysis of Energy-Efficient Parallel Storage Systems,' the Commodity Cluster Symposium 2007 (CCS), Annapolis, Maryland, July 2007.

[Zong et al., 2007c] Z.-L. Zong, K. Bellam, X.-J. Ruan, A. Manzanares, X. Qin, and Y.-M Yang, 'A Simulation Framework for Energy-efficient Data Grids,' Proc. Winter Simulation Conference, Washington, D.C., Dec. 2007.

[Zong et al., 2007d] Z.-L. Zong, X. Qin, M. Nijim, X.-J. Ruan, K. Bellam, and M. Alghamdi, 'Energy-Efficient Scheduling for Parallel Applications Running on Heterogeneous Clusters,' Proc. 36th International Conference on Parallel Processing (ICPP), Sept. 2007.

[Ruan et al., 2007] X.-J. Ruan, X. Qin, M. Nijim, Z.-L. Zong, and K. Bellam, 'An Energy-Efficient Scheduling Algorithm Using Dynamic Voltage Scaling for Parallel Applications on Clusters,' Proc. 16th IEEE Int'l Conference on Computer Communications and Networks (ICCCN), Honolulu, Hawaii, Aug. 2007.

[Zong et al., to be published] Z.-L. Zong, A. Manzanares, X. Qin, 'Load-Balancing Strategies for Energy-Efficient Parallel Storage Systems with Buffer Disks.'

[Manzanares et al., to be published] A. Manzanares, K. Bellam, and X. Qin, 'Energy-Efficient Prefetching for Parallel I/O Systems with Buffer Disks.'

[Xie and Sun, 2008] T. Xie and Y. Sun, 'Sacrificing Reliability for Energy Saving: Is It Worthwhile for Disk Arrays?,' Proc. 22nd IEEE International Parallel and Distributed Processing Symposium (IPDPS 2008), Miami, Florida, USA, April 14-18, 2008.

[Madathil et al., 2008] D. K. Madathil, R. B. Thota, P. Paul, and Tao Xie 'A Static Data Placement Strategy towards Perfect Load-Balancing for Distributed Storage Clusters,' The 7th International Workshop on Performance Modeling, Evaluation, and Optimization of Ubiquitous Computing and Networked Systems (PMEO UCNS 2008), in conjunction with the 22nd IEEE International Parallel and Distributed Processing Symposium (IPDPS 2008), Miami, Florida, USA, April 14-18, 2008.

[Xie and Sun, 2007] T. Xie and Y. Sun, 'No More Energy-Performance Trade-Off: A New Data Placement Strategy for RAID-Structured Storage Systems,' Proc. 14th Annual IEEE International Conference on High Performance Computing (HiPC 2007), Lecture Notes in Computer Science (LNCS 3834), pp.35-46, Goa, India, December 18-21, 2007.

[Xie, 2007] T. Xie, 'SOR: A Static File Assignment Strategy Immune to Workload Characteristic Assumptions in Parallel I/O Systems,' Proc. 36th International Conference on Parallel Processing (ICPP 2007), XiAn, China, September 10-14, 2007.

[Bellam et al., 2008] K. Bellam, A. Manzanares, X. Ruan, X. Qin, and Y.-M. Yang, 'Improving Reliability and Energy Efficiency of Disk Systems via Utilization Control,' Proc. IEEE Symposium on Computers and Communications (ISCC'08), July 2008.

[Liu et al., 2008a] C. Liu, X. Qin, and S. Li, 'PASS: Power-Aware Scheduling of Mixed Applications with Deadline Constraints on Clusters,' Proc. the 17th Int'l Conf. Computer Communications and Networks (ICCCN), St. Thomas, Virgin Islands, Aug. 2008.

[Liu et al., 2008b] C. Liu, X. Qin, S. Kulkarni, C.-J. Wang, S. Li, A. Manzanares, and S. Baskiyar, 'Distributed Energy-Efficient Scheduling for Data-Intensive Applications with Deadline Constraints on Data Grids,' Proc. 27th IEEE International Performance Computing and Communications Conference (IPCCC), Dec. 2008.

[Manzanares et al., 2008a] A. Manzanares, K. Bellam, and X. Qin, 'A Prefetching Scheme for Energy Conservation in Parallel Disk Systems,' Proc. NSF Next Generation Software Program Workshop, April 2008.

[Manzanares et al., 2008b] A. Manzanares, D. Hamilton, and X. Qin, 'The Relationship Between Software Architecture and Visual Programming Languages,' Proc. Grand Challenges in Modeling & Simulation, Edinburgh, Scotland, June 2008.

[Nijim et al., 2008a] M. Nijim, A. Manzanares, and X. Qin, 'An

Adaptive Energy-Conserving Strategy for Parallel Disk Systems,' Proc. the 12th IEEE Int'l Symp. Distributed Simulation and Real Time Applications (DS-RT), Oct. 2008.

[Nijim et al., 2008b] M. Nijim, Z.-L. Zong, K. Bellam, X.-J. Ruan and X. Qin, 'Security-Aware Cache Management for Cluster Storage Systems,' Proc. the 17th Int'l Conf. Computer Communications and Networks (ICCCN), St. Thomas, Virgin Islands, Aug. 2008.

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

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

[Xie and Wang, 2008] T. Xie and H. Wang, 'MICRO: A Multi-level Caching-based Reconstruction Optimization for Mobile Storage Systems,' IEEE Transactions on Computers, Vol. 57, No. 10, pp. 1386-1398, October 2008.

[Xie and Sun, 2008b] T. Xie and Y. Sun, 'PEARL: Performance, Energy, and Reliability Balanced Dynamic Data Redistribution for Next Generation Disk Arrays,' The 16th Annual Meeting of the IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Baltimore, Maryland, USA, September 8-10, 2008.

[Zong et al., 2008] Z.-L. Zong, M. Nijim, and X. Qin, 'Energy-Efficient Scheduling for Parallel Applications on Mobile Clusters,' Cluster Computing: The Journal of Networks, Software Tools and Applications, vol. 11, no. 1, pp. 91 - 113, March 2008.

[Ruan et al. 2009a] X.-J. Ruan, S. Yin, A. Manzanares, J. Xie, Z.-Y. Ding, J. Majors, and X. Qin, 'ECOS: An Energy-Efficient Cluster Storage System,' Proc. the 28th International Performance Computing and Communications Conference (IPCCC), Phoenix, Arizona, Dec. 2009. (40%, 8 pages)

[Ruan et al. 2009b] X.-J. Ruan, A. Manzanares, S. Yin, Z. -L. Zong, and X. Qin, 'Performance Evaluation of Energy-Efficient Parallel I/O Systems with Write Buffer Disks,' Proc. the 38th Int'l Conf. on Parallel Processing (ICPP), Vienna, Austria, Sept. 2009. (Acceptance Rate: 32.3%, 71/220) (40%, 8 pages)

[Nijim et al. 2009] M. Nijim, A. Manzanares, X.-J. Ruan, and X. Qin, 'HYBUD: An Energy-Efficient Architecture for Hybrid Parallel Disk Systems,' Proc. the 18th Int'l Conf. on Computer Communications and Networks (ICCCN), San Francisco, CA, Aug. 2009. (Acceptance Rate: 29%). (30%, 8 pages)

[Manzanares et al. 2009] BUD'09 A. Manzanares, X.-J. Ruan, S. Yin, M. Nijim, X. Qin, and W. Luo, 'Energy-Aware Prefetching for Parallel Disk Systems: Algorithms, Models, and Evaluation,' Proc. the 8th IEEE

International Symposium on Network Computing and Applications (NCA), July 2009. (50%, 8 pages)

[Ruan et al. 2009c] X.-J. Ruan, A. Manzanares, K. Bellam, X. Qin, 'DARAW: A New Write Buffer to Improve Parallel I/O Energy-Efficiency,' Proc. the 24th Annual ACM Symposium on Applied Computing (SAC), March 2009. (Acceptance Rate: 29%) (50%, 8 pages)

[Tjioe, Widjaja, Lee, and Xie, 2009] J. Tjioe, R. Widjaja, A. Lee, and T. Xie, 'DORA: A Dynamic File Assignment Strategy with Replication,' The 38th International Conference on Parallel Processing (ICPP 2009), Vienna, Austria, September 22-25, 2009.

[Xie and Sharma, 2009] T. Xie and A. Sharma, 'Collaboration-Oriented Data Recovery for Mobile Disk Arrays,' The 29th International Conference on Distributed Computing Systems (ICDCS 2009), Montreal, Quebec, Canada, June 22-26, 2009.

[Xie and Sun, 2009] T. Xie and Y. Sun, 'A File Assignment Strategy Independent of Workload Characteristic Assumptions,' ACM Transactions on Storage, Vol. 5, Issue 3, Article 10, November 2009.

[Lewis et al., 2010] J. Lewis*, M. I. Alghamdi*, M. A. Assaf*, X.-J. Ruan*, Z.-Y. Ding*, and X. Qin, 'An Automatic Prefetching and Caching System,' Proc. the 29th International Performance Computing and Communications Conference (IPCCC), Albuquerque, New Mexico, Dec. 2010.

[Ruan, et al. 2010] X.-J. Ruan*, Q. Yang*, M. I. Alghamdi*, S. Yin*, Z.-Y. Ding*, J. Xie*, J. Lewis*, and X. Qin, 'ES-MPICH2: A Message Passing Interface with Enhanced Security,' Proc. the 29th International Performance Computing and Communications Conference (IPCCC), Albuquerque, New Mexico, Dec. 2010. (40%, 8 pages)

[Qiu, et al. 2010] M.-K. Qiu, J.-W. Niu, L. T. Yang, X. Qin, S.-L. Zhang, and B. Wang, 'Energy-Aware Loop Parallelism Maximization for Multi-Core DSP Architectures,' Proc. IEEE/ACM International Conference on Green Computing and Communications (GreenCom-2010), Hangzhou, China, Dec 18-20, 2010. (Best Paper Award. 16%, 8 pages)

[Yang, et al. 2010] Q. Yang*, X.-J. Ruan*, A. Lim, and X. Qin, 'Location Privacy Protection in Contention Based Forwarding for VANETs,' Proc. IEEE Globecom 2010 Wireless Networking Symposium, Miami, FL, Dec. 6-10, 2010. (Acceptance Rate: 35%, 1300/3688) (10%, 8 pages)

[Manzanares, et al. 2010] A. Manzanares*, X.-J. Ruan*, S. Yin*, J. Xie*, Z.-Y. Ding*, Y. Tian*, J. Majors*, and X. Qin, 'Energy Efficient Prefetching with Buffer Disks for Cluster File Systems,' Proc. IEEE International Conference on Parallel Processing (ICPP), San Diego, CA, Sept. 13-16, 2010. (40%, 10 pages)

[Nijim, et al. 2010] M. Nijim, Z.-L. Zong, X. Qin, Y. Nijim, 'Multi-Layer Prefetching for Hybrid Storage Systems: Algorithms, Models, and Evaluations,' Proc. IEEE International Conference on Parallel Processing Workshops (ICPPW), San Diego, CA, Sept. 13-16, 2010. (10%, 6 pages)

[Yin, et al. 2010] S. Yin*, M. I. Alghamdi*, X.-J. Ruan*, M. Nijim, A. Tamilarasan*, Z.-L. Zong*, X. Qin, and Y.-M. Yang, 'Improving Energy Efficiency and Security for Disk Systems,' Proc. 12th IEEE International Conference on High Performance Computing and Communications (HPCC-10), Melbourne, Australia, September 1-3, 2010. (Acceptance Rate: 19%, 58/304) (40%, 8 pages).

[Liu, et al. 2010] Z. Liu*, F. Wu, X. Qin, C.-S. Xie, J. Zhou*, and J.-Z. Wang*, 'TRACER: A Trace-Replay Based Load-controllable Scheme for Evaluating Energy-efficiency of Mass Storage Systems,' Proc. IEEE International Conference on Cluster Computing (CLUSTER), Heraklion, Crete, Greece, Sept. 20-24, 2010. (40%, 10 pages)

[Qiu, et al. 2010] J. Li, M. Qiu, J. Niu, W. Gao, Z. Zong, and X. Qin, 'Feedback Dynamic Algorithms for Preemptable Job Scheduling in Cloud Systems', Proc. 2010 IEEE/WIC/ACM International Conference on Web Intelligence, pp. 561-564. Toronto, Canada, Sep. 2010. (10%, 4 pages)

[Wang, et al. 2010] P. Wang*, Diqing Hu, C.-S. Xie, J.-Z. Wang*, and X. Qin, 'A Fine-grained Data Reconstruction Algorithm for Solid-state Disks,' Proc. the 5th IEEE International Conference on Networking, Architecture, and Storage (NAS), July 2010. (Acceptance Rate: 24%, 43/178) (20%, 8 pages)

[Xie, et al. 2010] J. Xie*, S. Yin*, X.-J. Ruan*, Z.-Y. Ding*, Y. Tian*, J. Majors*, A. Manzanares*, and X. Qin, 'Improving MapReduce Performance via Data Placement in Heterogeneous Hadoop Clusters,' Proc. 19th International Heterogeneity in Computing Workshop, Atlanta, Georgia, April 2010. (40%, 8 pages)

[Zong et al., 2011] Z.-L. Zong, A. Manzanares, X.J. Ruan, K. Bellam, X. Qin, 'Heat-Based Dynamic Data Caching: A Load Balancing Strategy for Energy-Efficient Parallel Storage Systems with Buffer Disks.' Proc. the 27th IEEE Symposium on Massive Storage Systems and Technologies: Research Track (MSST), May 2011.

Training and Development:

Student Support

This project has directly supported about 8 students including 4 graduate students and and 4 undergraduate students. The project also indirectly contributed to approximately 65 undergraduate students.

The BUD project has directly supported about 30 students and indirectly contributed to approximately 100 students. We developed a Java-based disk simulator to evaluate our proposed energy conservation techniques for disks.

We developed a graduate level class - COMP7970 Storage Systems ? to train graduate students to implement disk simulators and conduct research in the realm of storage systems. The following table summaries the evaluation of this training class.

Rating system: 5- Agree Strongly 4-Agree 3-Unsure 2-Disagree 1-

Disagree strongly 0-Not applicable

I have learnt the following concepts and techniques: Max Min
Average
Energy-efficient storage systems 5 4 4.6
Dynamic power management 5 3 4.6
Design of low-power devices 5 0 3.5
Disk simulations 5 2 4.1
Reliable storage systems 5 3 4.2
Disk arrays 5 4 4.4
I/O-aware load balancing 5 4 4.6
BUD: Energy-efficient parallel disk systems with buffer disks 5
4 4.6

General Course Observations
I like this course. 5 4 4.8
I will recommend this course to others. 5 4 4.8

The graduate students who took the training course were very satisfied with the material covered in the class. The students are very likely to recommend this class to other students in the department.

We developed a Java-based disk simulator to evaluate our proposed reliability models for energy-efficient storage systems.

We developed COMP7370 ? Advanced Computer Security - for graduate students. This course was intended to give students a strong background in understanding design and development of secure systems in general and secure storage systems in particular.

The PI at Auburn educated graduate students to investigate the issues of energy efficiency and reliability in parallel disk systems. Graduate students also learnt how to use disk simulators to conduct research in the area of parallel disk systems. More than 93% of the students who took the storage systems class decided to recommend this class to other undergraduate and graduate students in the department of computer science and software engineering at Auburn University. All the students who took the class claimed that they like this type of research projects focusing on storage systems.

One of the major education objectives in this project is to recruit new undergraduate students, especially women and minorities, to conduct research in the area of storage systems and energy conservation technology in computer systems. To attract both the best undergraduate to participate in this project, in April 2008 we organized a workshop (see the following photo) that offered an opportunity for minority and women undergradate students to learn basic concepts in reliable and energy-efficient storge systems. In the long run, we plan to recuit monority students to conduct intensive research in fault-tolerant storage systems and reliability analysis with the PI. In this workshop, the PI emphasized new techniques and exciting findings of building mathematical reliability models and reducing energy dissipation in parallel disk systems.

A laboratory tour was held after the workshop. In the tour, the underrepresented students who participated in the workshop visited our new storage systems laboratory (see the photo below). This workshop along with the laboratory tour aims to offer the undergraduate students an opportunity to gain first-hand experience in designing and implementing reliable and energy-efficient storage systems.

Research Experience for Undergraduate Students
To recruit new undergraduate students, especially women and minorities, to conduct research in the area of computer security, we designed a research program that offers ample opportunity to undergraduate students to do intensive research in information assurance with the PIs. In particular, students and the PIs are brought together to conduct research experiments in the field of secure and energy-efficient storage systems. The photo below shows two undergraduate students - Tsukasa Ogihara (right) and Joshua Lewis (middle) - are building a cluster computing system using commodity-off-the-shelf (COTS) hardware components.

The cluster system (see the photo below) built by our undergraduate research assistants will be used as a high-performance computing platform to support our computer security education. The cluster recently build in our department at Auburn supports security middleware services for secure software applications. We will use this cluster computing platform to design and implement study how to improve software applications' quality-of-security without adversely affecting performance.

Advanced Computer Security for Graduate Students
We developed COMP7370 ? Advanced Computer Security - for graduate students. This course was intended to give students a strong background in understanding design and development of secure systems in general and secure storage systems in particular.

The PI at Auburn educated graduate students to investigate the issues of energy efficiency and reliability in parallel disk systems. Graduate students also learnt how to use disk simulators to conduct research in the area of parallel disk systems. More than 93% of the students who took the storage systems class decided to recommend this class to other undergraduate and graduate students in the department of computer science and software engineering at Auburn University. All the students who took the class claimed that they like this type of research projects focusing on storage systems.

Contributions to Courses
This project has directly and indirectly contributed to the following classes:
COMP7970: Storage Systems
COMP4300: Computer Architecture
COMP4370: Computer and Network Security
COMP7370: Advance Computer and Network Security
CS696 Advanced Distributed Systems
CS325: Principles of Operating Systems
CS331: Computer Architecture
CS531: Advanced Computer Architecture

Outreach Activities:

The BUD outreach activities include curriculum enrichment presentations, engineering clubs, and tutorial services.

We offered a presentation in the Commodity Cluster Symposium 2007 in Baltimore, MD. In this presentation, we described the design and performance analysis of an energy-efficient parallel storage systems tailored for cluster computing platforms. In addition, we give a presentation in the IEEE International Symposium on Parallel and Distributed Processing 2007. We described to the audience the design of the BUD disk architecture. We also shared our experience of carrying out the BUD project with the audience.

To attract the best minority undergraduate to participate in this project, the PI offered an undergraduate course ? introduction to computer and network security ? at the Alabama State University, which is one of the historically black colleges and universities (HBCU). Secured storage systems were introduced in this class. The PI educated the minority students about important security issues in in large-scale storage systems. The PI also shared our experience of reliable and energy-efficient parallel disk systems with the African-American students who took this class.

Dr. Tao Xie, a co-PI of this project, gave a presentation entitled 'Understanding the Relationship Between Energy-Saving and Disk Reliability,' at the Computer Science & Engineering Department at the University of California, Riverside on May 30, 2008.

In the past year, the PI had given the following research talks related to this NSF funded project.
1. 'An Application-Oriented Approach for Computer Security Education,' invited talk at the Information Security and Computer Applications (ISCA2011) Conference, Feb. 25, 2011.
2. 'A Novel Application-Oriented Approach to Teaching Computer Security Courses'. Poster Session at NSF CCLI/TUES Conference, January 27, 2011.
3. 'Energy Efficient Prefetching ? From models to Implementation'. Seminar talk at Huazhong University of Science and Technology, Wuhan, Hubei, China. June 2010.
4. 'How to Read Papers?' Training Session for REU students at Auburn University, May 18, 2010.
5. 'How to Succeed in the AU REU Program?' Training Session for REU students at Auburn University, May 17, 2010. 'Improving Energy-

Efficiency and Reliability of Storage Systems,' Seminar talk at the University of New Orleans, Sept. 4, 2009.
6. 'Can We Improve Energy Efficiency of Secure Disk Systems without Modifying Security Mechanisms?' the IEEE NAS'09 Conference, ZhangJiaJie, China, July 10, 2009.
7. 'Security-Aware Scheduling for Real-Time Parallel Applications on Clusters,' Lecture at Huazhong University of Science and Technology, Wuhan, Hubei, China. June 22, 2009.
8. 'How to Read Papers?' Seminar talk at Wuhan National Laboratory for Optoelectronics, Wuhan, China, June 17, 2009.
9. 'Energy Efficient Scheduling for High-Performance Clusters,' Seminar talk at Huazhong University of Science and Technology, Wuhan, Hubei, China. June 8, 2009.
10. 'An Overview of Auburn University,' Seminar talk at Nanjing University of Information Science and Technology, Nanjing, China, June 3, 2009.
11. 'Thinking About Going to Graduate School?' Seminar talk at Nanjing University of Information Science and Technology, Nanjing, China, June 3, 2009.
12. 'How to Write Research Papers, Part 1 ? General Principles,' Seminar talk at Taiyuan University of Science and Technology, Taiyuan, Shanxi, China, May 27, 2009.
13. 'Energy Efficient Scheduling for High-Performance Clusters,' Seminar talk at Taiyuan University of Science and Technology, Taiyuan, Shanxi, China. May 26, 2009.
14. 'Energy Efficient Resource Management for High-Performance Computing Platforms,' Seminar talk at Wuhan National Laboratory for Optoelectronics, Wuhan, China, May 15, 2009.
15. 'How to Write Research Papers, Part 1 ? General Principles,' Seminar talk at Wuhan National Laboratory for Optoelectronics, Wuhan, China, May 12, 2009.

Journal Publications

Tao Xie and Xiao Qin, "An Energy-Delay Tunable Task Allocation Strategy for Collaborative Applications in Networked Embedded Systems", IEEE Transactions on Computers, p. 329-343, vol. 57, (2008). Accepted,

Z.-L. Zong, M. Nijim, and X. Qin, "Energy-Efficient Scheduling for Parallel Applications on Mobile Clusters", Cluster Computing: The Journal of Networks, Software Tools and Applications, p. , vol. , (2008). Accepted,

Tao Xie and Xiao Qin, "Allocation of Tasks with Availability Constraints in Heterogeneous Systems", IEEE Transactions on Computers, p. 188-199, vol. 57, (2008). Published,

Tao Xie and Xiao Qin, "Availability-Aware Stochastic Scheduling for Heterogeneous Clusters", Cluster Computing: The Journal of Networks, Software Tools and Applications, p. , vol. , (2008). Published,

X. Qin, M. Alghamdi, M. Nijim, Z.-L. Zong, X.-J. Ruan, K. Bellam, and A. A. Manzanares,, "Improving Security of Real-Time Wireless Networks Through Packet Scheduling", IEEE Transactions on Wireless Communications, p. 3273-3279, vol. 7, (2008). Published,

Z.-L. Zong, M. Nijim, and X. Qin, "Energy-Efficient Scheduling for Parallel Applications on Mobile Clusters", Cluster Computing: The Journal of Networks, Software Tools and Applications, p. 91-113, vol. 11, (2008). Published,

X. Qin, "Performance Comparisons of Load Balancing Algorithms for I/O-Intensive Workloads on Clusters", Journal of Network and Computer Applications, p. 32-46, vol. 31, (2008). Published,

X. Qin, "Design and Analysis of a Load Balancing Strategy in Data Grids", Future Generation Computer Systems: The Int'l Journal of Grid Computing, p. 131-137, vol. 23, (2007). Published,

X.-J. Ruan, A. Manzanares, K. Bellam, X. Qin, "DARAW: A New Write Buffer to Improve Parallel I/O Energy-Efficiency", Proc. the 24th Annual ACM Symposium on Applied Computing, p. , vol. , (2009). Published,

C. Liu, X. Qin, S. Kulkarni, C.-J. Wang, S. Li, A. Manzanares, and S. Baskiyar, "Distributed Energy-Efficient Scheduling for Data-Intensive Applications with Deadline Constraints on Data Grids", Proc. 27th IEEE International Performance Computing and Communications Conference (IPCCC), p. , vol. , (2008). Published,

M. Nijim, A. Manzanares, and X. Qin, "An Adaptive Energy-Conserving Strategy for Parallel Disk Systems", Proc. the 12th IEEE Int'l Symp. Distributed Simulation and Real Time Applications (DS-RT), p. , vol. , (2008). Published,

M. Nijim, Z.-L. Zong, K. Bellam, X.-J. Ruan and X. Qin, "Security-Aware Cache Management for Cluster Storage Systems", Proc. the 17th Int'l Conf. Computer Communications and Networks (ICCCN), p. , vol. , (2008). Published,

C. Liu, X. Qin, and S. Li,, "PASS: Power-Aware Scheduling of Mixed Applications with Deadline Constraints on Clusters", Proc. the 17th Int'l Conf. Computer Communications and Networks (ICCCN), p. , vol. , (2008). Published,

A. Roth, A. Manzanares, K. Bellam, M. Nijim, and X. Qin, "Energy Conservation for Real-Time Disk Systems with I/O Burstiness", Proc. IEEE Int'l Workshop Next Generation Autonomous Storage and High Performance Computing, p. , vol. , (2008). Published,

A. Manzanares, D. Hamilton, and X. Qin, "The Relationship Between Software Architecture and Visual Programming Languages", Proc. Grand Challenges in Modeling & Simulation, Edinburgh, p. , vol. , (2008). Published,

K. Bellam, A. Manzanares, X. Ruan, X. Qin, and Y.-M. Yang, "Improving Reliability and Energy Efficiency of Disk Systems via Utilization Control", Proc. IEEE Symposium on Computers and Communications, p. , vol. , (2008). Published,

A. Manzanares, K. Bellam, and X. Qin, "A Prefetching Scheme for Energy Conservation in Parallel Disk Systems", Proc. NSF Next Generation Software Program Workshop, p. , vol. , (2008). Published,

K. Bellam, A. Manzanares, and X. Qin, "Improving Reliability and Energy Efficiency of Disk Systems", Proc. 46th ACM Southeast Conference, p. , vol. , (2008). Published,

Z.-L. Zong, K. Bellam, X.-J. Ruan, A. Manzanares, X. Qin, and Y.-M Yang, "A Simulation Framework for Energy-efficient Data Grids", Proc. Winter Simulation Conference, p. , vol. , (2008). Published,

Z.-L. Zong, X. Qin&#61482;, M. Nijim, X.-J. Ruan, K. Bellam, and M. Alghamdi, "Energy-Efficient Scheduling for Parallel Applications Running on Heterogeneous Clusters", Proc. 36th International Conference on Parallel Processing (ICPP), p. , vol. , (2007). Published,

X.-J. Ruan, X. Qin, M. Nijim, Z.-L. Zong, and K. Bellam, "An Energy-Efficient Scheduling Algorithm Using Dynamic Voltage Scaling for Parallel Applications on Clusters", Proc. 16th IEEE Int'l Conference on Computer Communications and Networks (ICCCN), p. , vol. , (2007). Published,

K. Bellam, R.K. Vudata, X. Qin, Z.-L. Zong, M. Nijim, and X.-J. Ruan, "Interplay of Security and Reliability using Non-Uniform Checkpoints", Proc. 16th IEEE Int'l Conference on Computer Communications and Networks (ICCCN), p. , vol. , (2007). Published,

W. Luo, F.-M. Yang, L.-P. Pang, G. Tu, and X. Qin, "TERCOS: A Novel Approach to Exploiting Redundancies in Fault-Tolerant and Real-Time Distributed Systems", Proc. 13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, p. , vol. , (2007). Published,

Z.-L. Zong, M.E. Briggs, N.W. O'Connor, X. Qin, M. Alghamdi, and Y.-M. Yang, "Design and Performance Analysis of Energy-Efficient Parallel Storage Systems", Commodity Cluster Symposium, p. , vol. , (2007). Published,

X. Qin, M. Alghamdi, M. Nijim, and Z.-L. Zong, "Scheduling of Periodic Packets in Energy-Aware Wireless Networks", Proc. the 26th IEEE Int'l Performance Computing and Communications Conf., p. , vol. , (2007). Published,

Z.-L.Zong, M.E. Briggs, N.W. O'Connor, and X. Qin, "An Energy-Efficient Framework for Large-Scale Parallel Storage Systems", Proc. 21st Int'l Parallel and Distributed Processing Symp. (IPDPS), p. , vol. , (2007). Published,

T. Xie and H. Wang, "MICRO: A Multi-level Caching-based Reconstruction Optimization for Mobile Storage Systems", IEEE Transactions on Computers, p. 1386-1398, vol. 57, (2008). Published,

T. Xie and Y. Sun, "PEARL: Performance, Energy, and Reliability Balanced Dynamic Data Redistribution for Next Generation Disk Arrays", Proc. the 16th Annual Meeting of the IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), p. , vol. , (2008). Published,

J. Tjioe, R. Widjaja, A. Lee, and T. Xie, "DORA: A Dynamic File Assignment Strategy with Replication", The 38th International Conference on Parallel Processing, p. , vol. , (2009). Published,

T. Xie and A. Sharma, "Collaboration- Oriented Data Recovery for Mobile Disk Arrays", The 29th International Conference on Distributed Computing Systems (ICDCS 2009), p. , vol. , (2009). Published,

T. Xie and Y. Sun, "A File Assignment Strategy Independent of Workload Characteristic Assumptions", ACM Transactions on Storage, p. , vol. 5, (2009). Published,

X.-J. Ruan, A. Manzanares, K. Bellam, X. Qin, "DARAW: A New Write Buffer to Improve Parallel I/O Energy-Efficiency", the 24th Annual ACM Symposium on Applied Computing, p. , vol. , (2009). Published,

A. Manzanares, X.-J. Ruan, S. Yin, M. Nijim, X. Qin, and W. Luo, "Energy-Aware Prefetching for Parallel Disk Systems: Algorithms, Models, and Evaluation", Proc. the 8th IEEE International Symposium on Network Computing and Applications, p. , vol. , (2009). Published,

M. Nijim, A. Manzanares, X.-J. Ruan, and X. Qin, "HYBUD: An Energy- Efficient Architecture for Hybrid Parallel Disk Systems", Proc. the 18th Int'l Conf. on Computer Communications and Networks, p. , vol. , (2009). Published,

X.-J. Ruan*, A. Manzanares, S. Yin, Z. -L. Zong, and X. Qin, "Performance Evaluation of Energy-Efficient Parallel I/O Systems with Write Buffer Disks", Proc. the 38th Int'l Conf. on Parallel Processing, p. , vol. , (2009). Published,

X.-J. Ruan, S. Yin, A. Manzanares, J. Xie, Z.-Y. Ding, J. Majors, and X. Qin, "ECOS: An Energy- Efficient Cluster Storage System", Proc. the 28th International Performance Computing and Communications Conference, p. , vol. , (2009). Published,

W. Luo, X. Qin, X.-C. Tan, K. Qin, and A. Manzanares, "Exploiting Redundancies to Enhance Schedulability in Fault-Tolerant and Real-Time Distributed Systems", IEEE Transactions on Systems Man & Cybernetics, Part A: Systems and Humans, p. 626, vol. 39, (2009). Published,

A. Manzanares, A. Roth, X.-J Ruan, S. Yin, M. Nijim, and X. Qin, "Conserving Energy in Real- Time Storage Systems with I/O Burstiness", ACM Transactions on Embedded Computing Systems, p. , vol. 9, (2010). Published,

X. Qin, H. Jiang, A. Manzanares, X.-J Ruan, and S. Yin, "Communication-Aware Load Balancing for Parallel Applications on Clusters", IEEE Transactions on Computers, p. 42, vol. 59, (2010). Published,

X. Qin, H. Jiang, A. Manzanares, X.-J Ruan, and S. Yin, "Dynamic Load Balancing for I/O-Intensive Applications on Clusters", ACM Transactions on Storage, p. , vol. 5, (2009). Published,

Z.-L. Zong, X.-J. Ruan, A. Manzanares, and X. Qin, "Energy-Aware Duplication Strategies for Scheduling Precedence Constrained Parallel Tasks on Clusters", IEEE Transactions on Computers, p. 360, vol. 60, (2011). Published, X.-J. Ruan, S. Yin, A. Manzanares, M. Alghamdi, and X. Qin, "A Message Scheduling Scheme for Energy Conservation in Multimedia Wireless Systems", IEEE Trans. on Systems Man & Cybernetics, p. 272, vol. 41, (2011). Published,

Books or Other One-time Publications

Z.-L. Zong, K. Bellam, X.-J. Ruan, A. Manzanares, X. Qin, and Y.-M Yang, "A Simulation Framework for Energy-efficient Data Grids", (2007). Proceedings, Published
Bibliography: Proc. Winter Simulation Conference, Washington, D.C., Dec. 2007.

Z.-L. Zong, X. Qin&#61482;, M. Nijim, X.-J. Ruan, K. Bellam, and M. Alghamdi,, "Energy-Efficient Scheduling for Parallel Applications Running on Heterogeneous Clusters", (2007). Proceedings, Published
Bibliography: Proc. 36th International Conference on Parallel Processing (ICPP), Sept. 2007.

X.-J. Ruan, X. Qin, M. Nijim, Z.-L. Zong, and K. Bellam, "An Energy-Efficient Scheduling Algorithm Using Dynamic Voltage Scaling for Parallel Applications on Clusters", (2007). Proceedings, Published
Bibliography: Proc. 16th IEEE Int'l Conference on Computer Communications and Networks (ICCCN), Honolulu, Hawaii, Aug. 2007.

Tao Xie, Yao Sun, "Sacrificing Reliability for Energy Saving: Is It Worthwhile for Disk Arrays?", (2008). Proceedings, Published
Bibliography: The 22nd IEEE International Parallel and Distributed Processing Symposium (IPDPS 2008), Miami, Florida, USA, April 14-18, 2008

Deepthi K.Madathil, Rajani B. Thota, Paulina Paul, and Tao Xie, "A Static Data Placement Strategy towards Perfect Load-Balancing for Distributed Storage Clusters", ( ). Proceedings, Accepted
Bibliography: The 7th International Workshop on Performance Modeling, Evaluation, and Optimization of Ubiquitous Computing and Networked Systems (PMEO UCNS 2008)

? Tao Xie, Yao Sun, "No More Energy-Performance Trade-Off: A New Data Placement Strategy for RAID-Structured Storage Systems", (2007). Proceedings, Published
Bibliography: The 14th Annual IEEE International Conference on High Performance Computing (HiPC 2007), Lecture Notes in Computer Science (LNCS 3834), pp.35-46, Goa, India, December 18-21, 2007

Tao Xie, "SOR: A Static File Assignment Strategy Immune to Workload Characteristic Assumptions in Parallel I/O Systems", (2007). Proceedings, Published
Bibliography: The 36th International Conference on Parallel Processing (ICPP 2007), XiAn, China, September 10-14, 2007.

K. Bellam, R.K. Vudata, X. Qin, Z.-L. Zong, M. Nijim, and X.-J. Ruan, "Interplay of Security and Reliability using Non-Uniform Checkpoints", (2007). Proceedings, Published
Bibliography: Proc. 16th IEEE Int'l Conference on Computer Communications and Networks (ICCCN), Honolulu, Hawaii, Aug. 2007

X. Qin, M. Alghamdi, M. Nijim, and Z.-L. Zong, "Scheduling of Periodic Packets in Energy-Aware Wireless Networks", (2007). Proceedings, Published
Bibliography: Proc. the 26th IEEE Int'l Performance Computing and Communications Conf. (IPCCC'07), New Orleans, Louisiana, April 2007.

Z.-L.Zong, M.E. Briggs, N.W. O'Connor, and X. Qin, "An Energy-Efficient Framework for Large-Scale Parallel Storage Systems", (2007). Proceedings, Published
Bibliography: Proc. 21st Int'l Parallel and Distributed Processing Symp. (IPDPS), 2007.

Web/Internet Site

URL(s):
http://www.eng.auburn.edu/~xqin/research/bud/index.html

Description:
This is the website designed to facilitate the dissemination of new findings, methods, experience from the BUD project.

Other Specific Products

Product Type:

Software (or netware)

Product Description:
A Java-based disk simulator used to simulate energy efficient disk systems.
Sharing Information:
The simulator can be downloaded from the BUD website.

Product Type:

Software (or netware)

Product Description:
ioBalanceSim 1.0 is a software package simulating an array of I/O-aware load balancing policies in parallel (e.g., Clusters) and distributed (e.g., computational grids) computing platforms.
Sharing Information:
This software can be downloaded from the project website at:
http://www.eng.auburn.edu/~xqin/software/ioBalanceSim/

Contributions

Contributions within Discipline:
In this project, we developed fundamental and innovative energy conservation techniques for modern parallel disk systems. Unlike sequential disk systems where energy conservation problems can be tackled by the dynamic power management and prefetching schemes, large-scale parallel disk systems require energy-efficient coordination of hundreds or thousands of concurrent disk devices to meet energy-saving and high-performance requirements. We designed and developed in this project a novel energy-efficient buffer-disk archiecture, energy-related reliability models, disk request processing, data placement strategies, power management for buffer disks, and energy-aware prefetching schemes.

Contributions to Other Disciplines:
In the long term, the proposed architecture and the energy-conservation techniques coming from this research can be transferable to embedded disk systems, where power constraints are more severe than conventional disk systems.

Contributions to Human Resource Development:
Two doctoral students - Mais Nijim and Ziliang Zong - who were supported by this grant graduated from the PI's research group. Now Mais Nijim is a tenure-track assistant professor at the University of Southern Mississippi. Ziliang Zong is a tenure-track assistant professor at the South Dakota School of Mines and Technology.

In addition, two Master students who were supported by this grant graduated. These students became experts in the development of energy-efficient storage systems.

Contributions to Resources for Research and Education:
This project integrates the research efforts into the educational process. Specifically, we developed storage systems curriculum to better address the new issues raised in today's parallel disk systems, energy conservation technology, cluster computing, and the like.

Contributions Beyond Science and Engineering:
The BUD project benefits society by developing economically attractive and environmentally friendly parallel disk systems, which can lower electricity bills and reduce emissions of air pollutants by providing significant energy savings.

Conference Proceedings

Categories for which nothing is reported:

Any Conference

1. Research and Education Activities

Project team meetings held

The PI and Co-PI met together on several occasions during the past year: March 30, 2007, at Long Beach, California; and August 6, 2007, Arlington VA (during the NSF FSIO workshop). These project meetings are playing an important role in coordinating our collaborative research activities. In addition, the PI, Co-PI, and senior personnel made use of frequent emails and phone discussions to collaborate on the project.

A buffer-disk architecture

We designed an energy-efficient buffer-disk architecture called BUD for parallel disk systems. In this BUD configuration, referred to as interBUD aims to support inter-request parallelisms.

Power consumption models

We started the power consumption modeling effort by introducing the simplest model where each disk has only two power states: a working state S_w and a sleep state S_s. All the disks in a parallel disk system will be modeled by their power characteristics, i.e., $D = (p_w, p_s, p_u, p_d, t_u, t_d)$, where p_w is the power consumed in the working state, p_s is the power consumed in the working state, p_u is the power consumed during wake up, p_d is the power consumed during shutdown, t_u is the transition time from the sleep state to working state. t_d is the transition time from the working state to sleep state. The total energy consumed by a disk can be calculated as $E = E_w + E_s + E_u + E_d$, where E_w and E_s are the energy consumed by the disk when it is in the working sleep states, E_u and E_d is the energy consumed when the disk transits from the sleep to working state and vice versa.

Energy-aware prefetching (PRE-BUD)

Our goal in this part of study was to use the interBUD architecture to aid in the reduction of the power consumption for these large-scale parallel disk systems. This work aimed at reducing the energy consumed of these systems by comparing two different approaches at reducing energy consumption using a disk management scheme called PRE-BUD. PRE-BUD is able to prefetch data into buffer disks with a desired consequence of reducing the total energy consumption of a large-scale parallel disk system. The first approach using PRE-BUD examined adds an extra disk, which becomes the buffer disk, and the other approach uses an existing disk as the buffer disk. This algorithm relies on the fact that in some applications a small percentage of the data is frequently accessed. Our goal was to place a small amount of frequently accessed data into the buffer disk reducing energy consumption. This work also focused on increasing the reliability of the disk system over typical energy aware disk management schemes. This can be achieved with a reduction in state transitions that the use of a buffer disk facilitates.

Write request processing

In the BUD architecture, large and small writes are processed in different ways. Large write requests are issued directly to data disks. In contrast, small write requests are 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 issuing the request. Our study confirmed that seek times of small disk request dominates disk I/O processing times. To alleviate this situation, we made use of sequential accesses, i.e., a log file system, in interBUD, thereby making the seek time of most write requests to be zero. This is because disk head of a sequential access disk is, in most cases, positioned on an empty track that is available for incoming write requests. The seek times of write requests handled by buffer disks are zero unless the buffer disks are in a process of moving data to data disks or responding read requests.

Energy-efficient data placement

We investigated data placement strategies, which place all data sets onto a disk array before they are accessed. Data placement is one of avenues that can significantly affect the overall performance of a parallel I/O system. We developed a data placement scheme that can simultaneously achieve high energy efficiency and quick response times through intelligent in our BUD architecture.

A Prefetching Scheme for Energy Conservation in Parallel Disk Systems

Large-scale parallel disk systems are frequently used to meet the demands of information systems requiring high storage capacities. A critical problem with these large-scale parallel disk systems is the fact that disks consume a significant amount of energy. To design economically attractive and environmentally friendly parallel disk systems, we developed two energy-aware prefetching strategies for parallel disk systems with disk buffers. First, we introduced a new buffer disk architecture that can provide significant energy savings for parallel disk systems while achieving high performance. Second, we designed a prefetching approach to utilize an extra disk to accommodate prefetched data sets that are frequently accessed. Third, we developed a second prefetching strategy that makes use of an existing disk in the parallel disk system as a buffer disk. Compared with the first prefetching scheme, the second approach lowers the capacity of the parallel disk system. However, the second approach is more cost-effective and energy-efficient than the first prefetching technique. Finally, we quantitatively compare both of our prefetching approaches against two conventional strategies including a dynamic power management technique and a non-energy-aware scheme. Using empirical results we show that our novel prefetching approaches are able to reduce energy dissipation in parallel disk systems by 44% and 50% when compared against a non-energy aware approach. Similarly, our strategies are capable of conserving 22% and 30% of the energy when compared to the dynamic power management technique.

Improving Reliability and Energy Efficiency of Disk Systems via Utilization Control

As disk drives become increasingly sophisticated and processing power increases, one of the most critical issues of designing modern disk systems is data reliability. Although numerous energy saving techniques are available for disk systems, most of energy conservation techniques are not effective in reliability critical environments due to their limitation of ignoring the reliability issue. A wide range of factors affect the reliability of disk systems; the most important factors – disk utilization and ages – are the focus of this study. We build a model to quantify the relationship among the disk age, utilization, and failure probabilities. Observing that the reliability of a disk heavily relies on both disk utilization and age, we propose a novel concept of safe utilization zone, where energy of the disk can be conserved without degrading reliability. We investigate an approach to improving both reliability and energy efficiency of disk systems via utilization control, where disk drives are operated in safe utilization zones to minimize the probability of disk failure. In this study, we integrate an existing energy consumption technique that operates the disks at different power modes with our proposed reliability approach. Experimental results show that our approach can significantly improve reliable while achieving high energy efficiency for disk systems.

DARAW: A New Write Buffer to Improve Parallel I/O Energy-Efficiency

In the past decades, parallel I/O systems have been used widely to support scientific and commercial applications. New data centers today employ huge quantities of I/O systems, which consume a large amount of energy. Most large-scale I/O systems have an array of hard disks working in parallel to meet performance requirements. Traditional energy conservation techniques attempt to place disks into low-power states when possible. In this part of research, we proposed a novel strategy, which aims to significantly conserve energy while reducing average I/O response times. This goal was achieved by making use of buffer disks in parallel I/O systems to accumulate small writes to form a log, which can be transferred to data disks in a batch way. We develop an algorithm - dynamic request allocation algorithm for writes or DARAW - to energy efficiently allocate and schedule write requests in a parallel I/O system. DARAW is able to improve parallel I/O energy efficiency by the virtue of leveraging buffer disks to serve a majority of incoming write requests, thereby keeping data disks in low-power state for longer period times. Buffered requests are then written to data disks at a pre-determined time. Experimental results show that DARAW can significantly reduce energy dissipation in parallel I/O systems without adverse impacts on I/O performance.

Heat-Based Dynamic Data Caching: A Load Balancing Strategy for Energy-Efficient Parallel Storage Systems with Buffer Disks

Performance improvement and energy conservation are two conflicting objectives in large scale parallel storage systems. In this project, we proposed a novel solution to achieve the twin objectives of maximizing performance and minimizing energy consumption of parallel storage systems. Specifically, a buffer-disk based architecture (BUD for short) is designed to conserve energy. A heat-based dynamic data caching strategy is developed to improve performance. The BUD architecture strives to allocate as many requests as possible to buffer disks, thereby keeping a large number of idle data disks in low-power states. This can provide significant opportunities for energy conservation while making buffer disks a potential performance bottleneck. The heat-based data caching strategy aims to achieve good load balancing in buffer disks and alleviate overall performance degradation due to unbalanced workload. Our experimental results show that the proposed BUD framework and dynamic data caching strategy are able to conserve energy by 84.4% for small reads and 78.8% for large reads with slightly degraded response time.

An Adaptive Energy-Conserving Strategy for Parallel Disk Systems

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 a variety of parallel disk systems were developed to achieve high performance and energy efficiency, most existing parallel disk systems lack an adaptive way to conserve energy in dynamically changing workload conditions. To solve this problem, we develop an adaptive energy-conserving algorithm, or DCAPS, for parallel disk systems using the dynamic voltage scaling technique that dynamically choose the most appropriate voltage supplies for parallel disks while guaranteeing specified performance (i.e., desired response times) for disk requests. We conduct extensive experiments to quantitatively evaluate the performance of the proposed energy-conserving strategy. Experimental results consistently show that DCAPS significantly reduces energy consumption of parallel disk systems in a dynamic environment over the same disk systems without using the DCAPS strategy.

Energy-Efficient Data Placement

We investigated data placement strategies, which place all data sets onto a disk array before they are accessed. Data placement is one of avenues that can significantly affect the overall performance of a parallel I/O system. First, we developed a static non-partitioned file assignment strategy for parallel I/O systems, called *s*tatic r*o*und-*r*obin (SOR), which is immune to the workload assumption. Next, to achieve energy-conservation and prompt responses simultaneously, we designed an energy-aware strategy, called *s*triping-based *e*nergy-*a*ware (SEA), which can be integrated into data placement in RAID-structured storage systems to noticeably save energy while providing quick responses. Finally, to illustrate the effectiveness of SEA, we implemented two SEA-powered striping-based data placement algorithms, SEA0 and SEA5, by incorporating the SEA strategy into RAID-0 and RAID-5, respectively.

An Energy-Aware Data Reconstruction Strategy for Mobile Disk Arrays

Compared with conventional stationary storage systems, mobile disk-array-based storage systems are more prone to disk failures due to their severe application environments. Further, they have very limited power supply. Therefore, data reconstruction algorithms, which are executed in the presence of disk failure, for mobile storage systems must be performance-driven, reliability-aware, and energy-efficient. We developed a reconstruction strategy, called *m*ult*i*-level *c*aching-based *r*econstruction *o*ptimization (MICRO), which can be applied to RAID-structured mobile storage systems to noticeably shorten reconstruction times and user response times while saving energy.

Understanding the Relationship between Energy Conservation and Reliability in Parallel Disk Arrays

Power management and workload skew based energy conservation schemes for disk arrays inherently and adversely affect the reliability of disks due to either workload concentration or frequent disk speed transitions. A thorough understanding of the relationship between energy saving techniques and disk reliability is indispensible. We developed an empirical reliability model, called Predictor of Reliability for Energy Saving Schemes (PRESS). Fed by operating temperature, disk utilization, disk speed transition frequency, three energy-saving-related reliability affecting factors, PRESS estimates the reliability of entire disk array. Further, a new energy saving strategy with reliability awareness named Reliability and Energy Aware Distribution (READ) is developed in the light of the insights provided by PRESS.

Performance, Energy, and Reliability Balanced Dynamic Data Redistribution for Next Generation Disk Arrays

Contemporary disk arrays consist purely of hard disk drives, which normally provide huge storage capacities with low-cost and high-throughput for data-intensive applications. Nevertheless, they have some inherent disadvantages such as long access latencies, high annual disk replacement rates, fragile physical characteristics, and energy-inefficiency due to their build-in mechanical and electronic mechanisms. Flash-memory based solid state disks, on the other hand, although currently more expensive and inadequate in write cycles, offer much faster read accesses and are much more robust and energy efficient. To combine the complementary merits of hard disks and flash disks, in this research we developed a hybrid disk array architecture named HIT (hybrid disk storage) for data-intensive applications. Next, a dynamic data redistribution strategy called PEARL (performance, energy, and reliability balanced), which can periodically redistribute data between flash and hard disks to adapt to the changing data access patterns, is developed on top of the HIT architecture.

ECOS: An Energy-Efficient Cluster Storage System

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 study 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.

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

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.

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

Although flash memory is very energy-efficient compared to disk drives, flash memory is too expensive to use as a major component in large-scale storage systems. In other words, it is not a cost-effective way to make use of large flash memory to build energy-efficient storage systems. To address this problem, in this study we proposed a hybrid disk architecture or HYBUD that integrates a non-volatile flash memory with buffer disks to build cost-effective and energy-efficient parallel disk systems. While the most popular data sets are cached in flash memory, the second most popular data sets can be stored and retrieved from buffer disks. HYBUD is energy efficient because flash memory coupled with buffer disks can serve a majority of incoming disk requests, thereby keeping a large number of other data disks in the low-power state for longer period times. Furthermore, HYBUD is cost-effective by the virtue of inexpensive buffer disks assisting flash memory to cache a huge amount of popular data. Experimental results demonstratively show that compared with two existing non-hybrid architectures, HYBUD provides significant energy savings for parallel disk systems in a very cost effective way.

Energy-Aware Prefetching for Parallel Disk Systems

In this study we design and evaluate an energy-aware prefetching strategy for parallel disk systems consisting of a small number of buffer disks and large number of data disks. Using buffer disks to temporarily handle requests for data disks, we can keep data disks in the low-power mode as long as possible. Our prefetching algorithm aims to group many small idle periods in data disks to form large idle periods, which in turn allow data disks to remain in the standby state to save energy. To achieve this goal, we utilize buffer disks to aggressively fetch popular data from regular data disks into buffer disks, thereby putting data disks into the standby state for longer time intervals. A centrepiece in the prefetcing mechanism is an energy-saving prediction model, based on which we implement the energy-saving calculation module that is invoked in the prefetching algorithm. We quantitatively compare our energy-aware prefetching mechanism against existing solutions, including the dynamic power management strategy. Experimental results confirm that the buffer-disk-based prefetching can significantly reduce energy consumption in parallel disk systems by up to 50 percent. In addition, we systematically investigate the energy efficiency impact that varying disk power parameters has on our prefetching algorithm.

DORA: A Dynamic File Assignment Strategy with Replication

Compared with numerous static file assignment algorithms proposed in the literature, very few investigations on the dynamic file allocation problem have been accomplished. Moreover, none of them has integrated file replication techniques into file assignment algorithms in a highly dynamic file system where files are created or deleted on the fly and their access patterns varied over time. We argue that file replication and file assignment can act in concert to boost the performance of parallel disk systems. In this study, we propose a new dynamic file assignment strategy called DORA (dynamic round robin with replication). The advantages of DORA can be attributed to its two main characteristics. First, it takes the dynamic nature of file access patterns into account to adapt to a changing workload condition. Second, it utilizes file replication techniques to complement file assignment schemes so that system performance can be further improved. Experimental results demonstrate that DORA performs consistently better than existing algorithms.

Collaboration-Oriented Data Recovery for Mobile Disk Arrays

Mobile disk arrays, disk arrays located in mobile data centers, are crucial for mobile applications such as disaster recovery. Due to their unusual application domains, mobile disk arrays face several new challenges including harsh operating environments, very limited power supply, and extremely small number of spare disks. Consequently, data reconstruction schemes for mobile disk arrays must be performance-driven, reliability-aware, and energy-efficient. In this study, we develop a flash assisted data reconstruction strategy called CORE (collaboration-oriented reconstruction) on top of a hybrid disk array architecture, where hard disks and flash disks collaborate to shorten data reconstruction time, alleviate performance degradation during disk recovery. Experimental results demonstrate that CORE noticeably improves the performance and energy-efficiency over existing schemes.

A File Assignment Strategy Independent of Workload Characteristic Assumptions

The problem of statically assigning non-partitioned files in a parallel I/O system has been extensively investigated. A basic workload characteristic assumption of most existing solutions to the problem is that there exists a strong inverse correlation between file access frequency and file size. In other words, the most popular files are typically small in size, while the large files are relatively unpopular. Recent studies on the characteristics of web proxy traces suggested, however, the correlation, if any, is so weak that it can be ignored. Hence, the following two questions arise naturally. First, can existing algorithms still perform well when the workload assumption does not hold? Second, if not, can one develop a new file assignment strategy that is immune to the workload assumption? To answer these questions, we first evaluate the performance of three well-known file assignment algorithms with and without the workload assumption, respectively. Next, we develop a novel static non-partitioned file assignment strategy for parallel I/O systems, called static round-robin (SOR), which is immune to the workload assumption. Comprehensive experimental results show that SOR consistently improves the performance in terms of mean response time over the existing schemes.

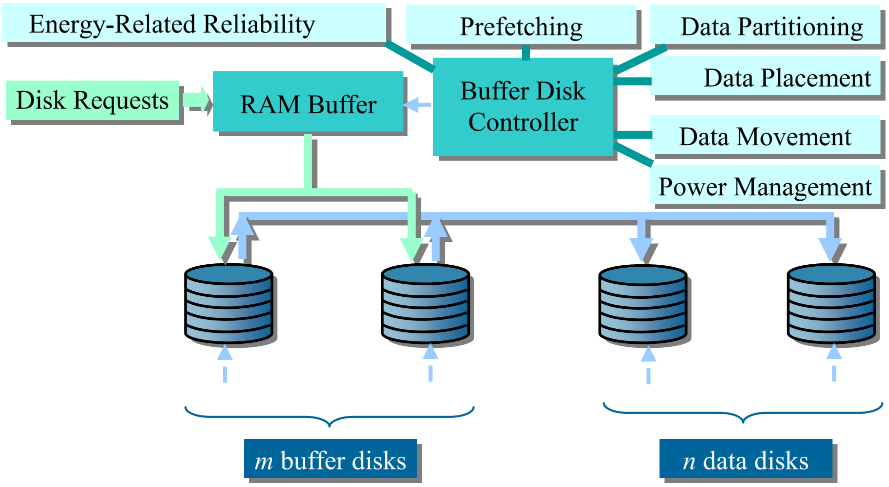
New course development

Our education activities include the development of a new course – Advanced Computer Security - for graduate students. This course was intended to give students a strong background in understanding design and development of secure systems in general and secure storage systems in particular.

2. Findings

A focus of the research activities carried out in the last year is the design of an energy efficient parallel disk architecture for inter-request parallelisms.

The architecture consists of four major components: a RAM buffer, *m* buffer disks, *n* data disks, and an energy-aware buffer-disk controller. The new BUD disk architecture is diagramed below [Zong et al., 2007]:



The RAM buffer with a size ranging from several megabytes to gigabytes is residing in the main memory. The buffer-disk controller carefully coordinates energy-related reliability model, data partitioning, disk request processing, data movement/placement strategies, power management, and prefetching schemes. Please refer to Section 3 for details of how the controller will be designed and developed. We choose to use log disks as buffer disks, because data can be written onto the log disks in a sequential manner to improve performance of disk systems. 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.

We introduced a model to calculating energy consumption for disk drives in a parallel disk system . [Zong et al., 2007a] The basic power model for this study is a summation of all power states multiplied by the time 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 share similar power consumption. Let T_{tr} 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_{tr}. Thus, energy E_{tr} consumed by the disk when it enters and exits the inactive state is expressed as $P_{tr} \cdot T_{tr}$. Similarly, let T_{active} and T_{idle} are the time intervals when the disk is in the active and inactive states, respectively. We denote the energy consumption rates of the disk when it is active and inactive by P_{active} and P_{idle}, respectively. Hence, the energy dissipation E_{active} of the disk when it is in the active state can be written as $P_{active} \cdot T_{active}$, and the energy E_{idle} of the disk when it is sitting idle can be expressed as $P_{idle} \cdot T_{idle}$. The total energy consumed by the disk system can be calculated as

$$
\begin{aligned}
E_{total} &= E_{tr} + E_{active} + E_{idle} \\
&= P_{tr} \cdot T_{tr} + P_{active} \cdot T_{active} + P_{idle} \cdot T_{idle}
\end{aligned}
\ .
$$

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

$$
\begin{aligned}
T_{tr} &= N_{ai} T_{ai} + N_{ia} T_{ia} \ , \\
P_{tr} &= \frac{T_{ai}}{T_{ai} + T_{ia}} P_{ai} + \frac{T_{ia}}{T_{ai} + T_{ia}} P_{ia} \ .
\end{aligned}
$$

In most cases where the values of N_{ai} and N_{ia} are both equal to N_{tr}, T_{tr} can be written as

$$
T_{tr} = N_{tr} \left(T_{ai} + T_{ia} \right),
$$

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

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

where *n* is the total number of submitted disk requests, and $T_{service}(i)$ is the serving time of the *i*th disk request. $T_{service}(i)$ can be modeled as

$$
T_{service}(i) = T_{seek}(i) + T_{rot}(i) + T_{trans}(i).
$$

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 the disk. Now we quantify energy saved by power management policies as below

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

[Zong et al., 2007a] Write requests can be divided into small write requests and large write requests. Whenever the controller receives a write request, it will first check the size. If the request is a large write, say over 10MB or more, it is sent directly to the corresponding data disk. Otherwise, the controller will send this request to the RAM buffer that buffers small writes from the host and form a log of data to be written into one of the buffer disks later. Once the data are transferred to the RAM buffer, the controller will send a “write complete” acknowledgement message to the sender. Then the controller will test the state of all the buffer disks. If one buffer disk is not busy with writing a previous log or reading or transferring data, the data copy will be sent to this buffer disk to ensure that a reliable copy resides on one of the buffer disks. In order to guarantee the correctness and consistency of different data version, the controller is always trying to match the data with same block to the same buffer disk unless it is known that the data block is already outdated. In other words, 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. The most important scheduling strategy between RAM buffer and buffer disk is that rather than wait until the RAM is full, the data are written into the buffer disks whenever they are available. This policy has two major advantages. First, data re guaranteed to be written into one of the reliable buffer disks in the shortest time period, which is very important to ensure the reliability and availability of data. Second, the RAM buffer can have more available room to buffer a large burst of new requests because previous data are always quickly moved from the RAM to the buffer disks. Here we should note that the total storage space of each buffer disk is divided into equal n parts (n is the number of data disks) which are used to buffer the data requests corresponding to each data disk. For example, if we have two 10GB buffer disks and ten 100GB data disks, each buffer disk will have 1GB as the buffer space for each data disk. All the small write requests to data disk 1 will be buffered in the corresponding buffer space reserved for disk 1. The reason we split the buffer disks into small pieces for each data disk is to improve the response performance of the whole system. In the case when one buffer disk is busy writing or moving data, the other disk could serve the incoming requests immediately.

Handling read operations is kind of simple and straightforward in the BUD framework. When a read request arrives, the controller first searches the RAM buffer. If the data is still in the RAM buffer then the data is immediately sent back to the requester. Otherwise, the controller will do a seek operation in the buffer disks. If the required data can not be found in the buffer disks, a miss message will be sent back to controller and the controller will send a read request to the corresponding data disk and finally the data will be transferred to the requester by the data disk. Using this policy, the read performance should be similar to or sometimes better than that of traditional disk because most of the requests will be sent to the data disk and reading from RAM or buffer disks seldom occurs in real applications.

Our preliminary results consist of first developing a simulator which meets all project specifications and running this simulator with a trace to get some preliminary results [Zong et al., 2007b]. So far the simulator completed all the main functions that are necessary in order to model our distributed system. That is the program takes data from a trace. Then the program moves data to appropriate virtual disks that use disksim to derive there timing information. These virtual disks use a simple model to calculate total energy. Finally, both timing and energy data are reported to the user in the form of the two respective totals. Our results from the simulator consist of two parallel disk systems. To simulate with these two systems we used a simple trace that came with disksim. The first system that was simulated was a simple disk system which is used today by many storage systems. It is basically a RAID 1 system consisting of 31 disks. This is basically a baseline system in which to compare our results. The second simulated disk system is similar to section 3 with 6 buffer disks each one acting as a buffer for a group of approximately 5 disks.

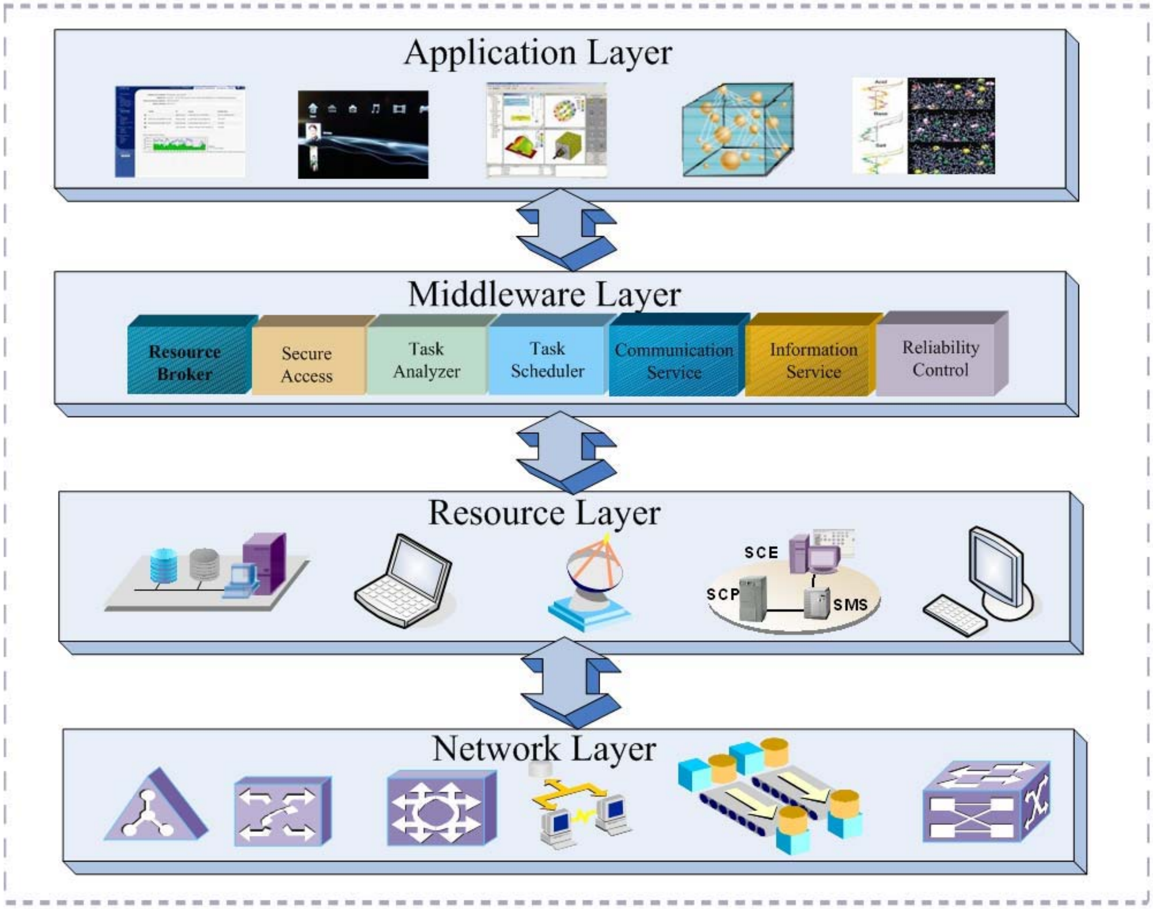
The results from running these systems showed that both took 25.55 min. Table below shows the energy consumption of the disk system without employing buffer disks and the disk system using buffer disks to conserve energy. Specifically, the traditional system without buffer disks consumed 189279.78 J (0.05 KWH) with all disks starting from off and being turned on when needed. In contrast, the parallel disk system with buffer disks only used 117345.99 J (0.03 KWH), which is substantially less. This resulted in overall power savings of 38%. Our results have shown substantial gains can be made by using buffer disks for very specific workload conditions.

In order to further develop our storage algorithm we need to test our results in many more simulations and with more varied workloads. The simulator also needs to check with analytic calculations and the code needs to include the transition time and power from switching from mode to mode.

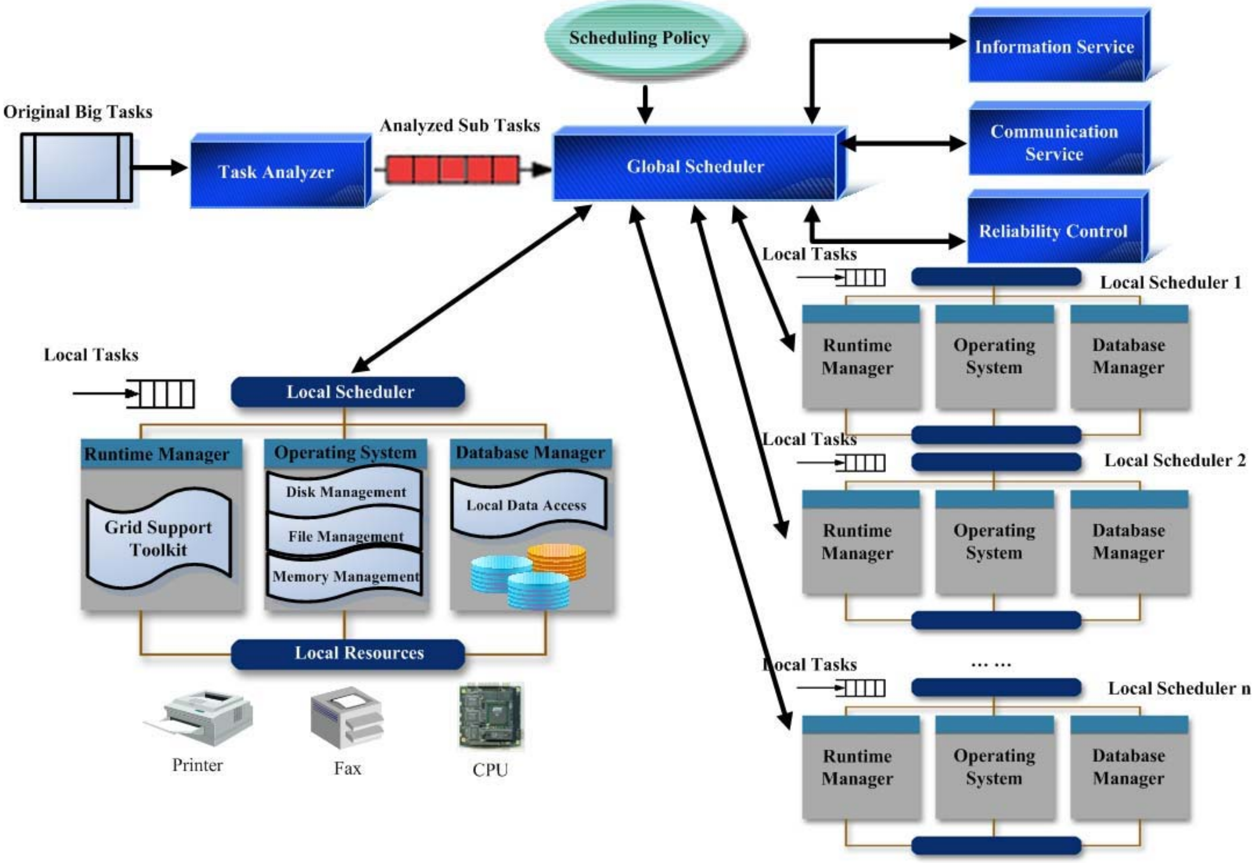
Simulated Disk System	Energy Consumption
Disk system without buffer disks	189279.78 J
Disk system with buffer disks	117345.99 J
Energy saving	71933.79 J
Energy consumption reduced by	38%

[Zong et al., 2007c] High performance data grids are increasingly becoming popular platforms to support data-intensive applications. Reducing high energy consumption caused by data grids is a challenging issue. Most previous studies in grid computing focused on performance and reliability without taking energy conservation into account. As such, designing energy-efficient data grid systems became highly desirable. In this project, we proposed a framework to simulate energy-efficient data grids. We presented an approach to integrate energy-aware allocation strategies into energy-efficient data grids. Our framework aims at simulating a data grid that can conserve energy for data-intensive applications running on data grids.

A data grid can be envisioned as a complicated distributed system, which consists of the following four major layers: application layer, middleware layer, resource layer and network layer. Fig. 1 depicts the four layers and their relations for data grids. It is worth noting that the simulation framework is constructed in accordance with the system architecture outlined as follows:

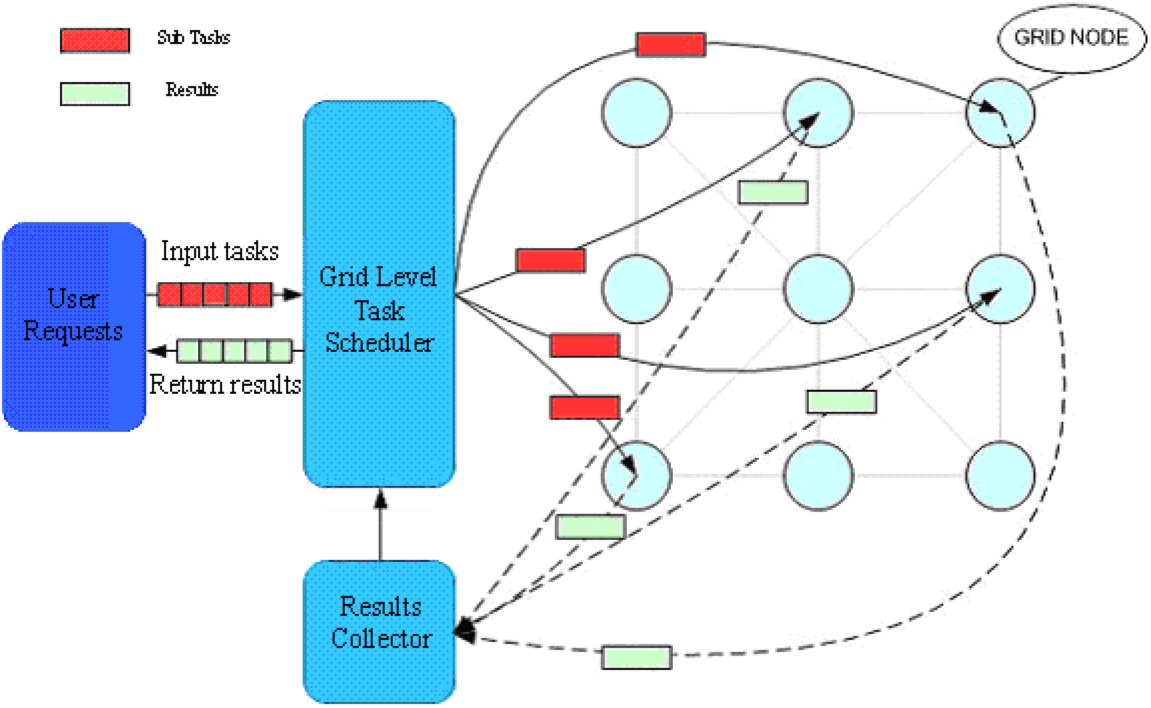


Data grids are complex multivariate environments, which are made up of numerous grid entities that need to be automatically managed. In order to make coherent and coordinated use of ubiquitous and heterogeneous data and storage resources, resource management is a centerpiece in the simulation framework. In this section, we present a framework to simulate data grids that are energy efficient in nature. In general, a data grid should energy efficiently handle two important components in the system: storage resources and data-intensive jobs. A data grid has to ad-dress the following issues. First, it is of paramount impor-tance to find available storage resources within a short pe-riod of time. Second, it must allocate data-intensive jobs to available resources. We proposed a simulation framework for data grids:



In this framework, the global level scheduler (or grid level scheduler) coordinates mul-tiple local scheduling or helps to select the most appropri-ate resources for a job among different possible resources. Typically, a global level scheduler itself has no direct control over computing and storage resources. Therefore, the global level scheduler has to communicate with and appropriately trigger several local level schedulers to complete data-intensive tasks submitted by users. Those local level schedulers either control resources directly or have certain access to their local resources. The global level scheduler is also responsible for collaborating with other important supportive middleware like information services, communication services, and reliability control modules.

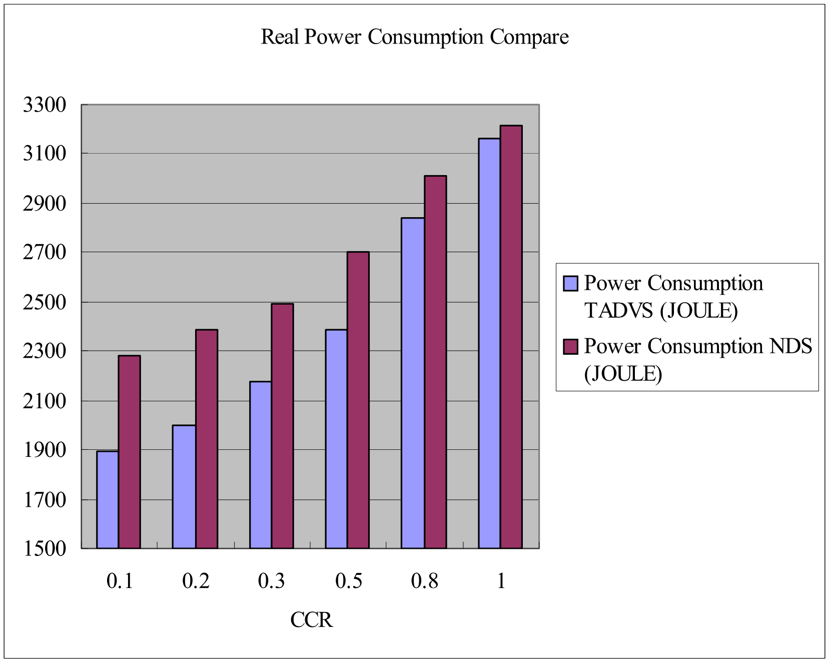
[Zong et al., 2007c] As long as the grid scheduling module collected all the information of currently available computing and storage resources, it can judiciously choose target recourses based on its scheduling policy and allocate the tasks analyzed by the task analyzer to these chosen resources for parallel execution. We designed the job scheduling flow in the simulation frame



During the process of execution, the results collector will periodically check the randomly returned sub results and transfer these sub results to Grid level scheduler. The scheduler in the framework passes the latest information to all tasks, which can guarantee that the tasks with dependency could immediately be executed once they get the necessary sub results.

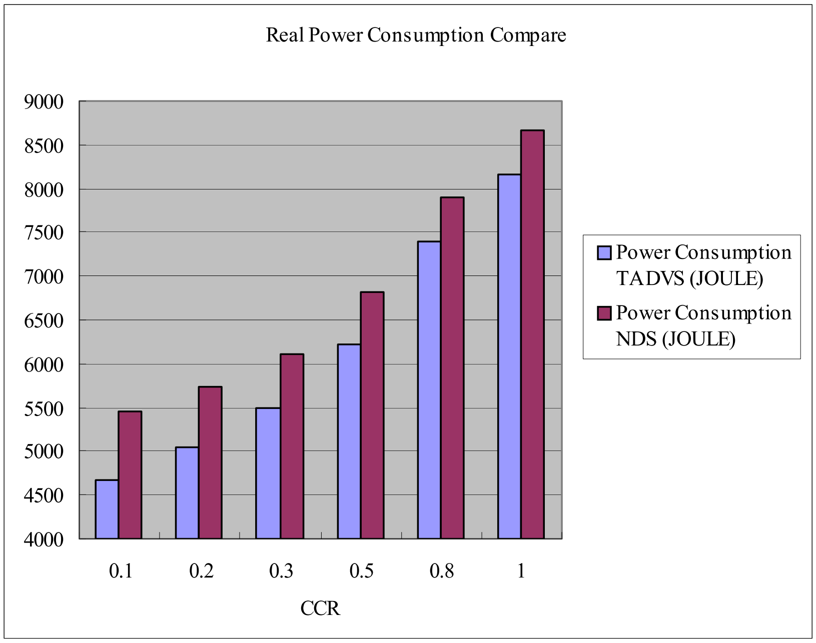
[Ruan et al., 2007] In the past decade cluster computing platforms have been widely applied to support a variety of scientific and commercial applications, many of which are parallel in nature. However, scheduling parallel applications on large scale clusters is technically challenging due to significant communication latencies and high energy consumption. As such, shortening schedule length and conserving energy consumption are two major concerns in designing economical and environmentally friendly clusters. In this study, we proposed an energy-efficient scheduling algorithm (TADVS) using the dynamic voltage scaling technique to provide significant energy savings for clusters. The TADVS algorithm aims at judiciously leveraging processor idle times to lower processor voltages (i.e., the dynamic voltage scaling technique or DVS), thereby reducing energy consumption experienced by parallel applications running on clusters. Reducing processor voltages, however, can inevitably lead to increased execution times of parallel task. The salient feature of the TADVS algorithm is to tackle this problem by exploiting tasks precedence constraints. Thus, TADVS applies the DVS technique to parallel tasks followed by idle processor times to conserve energy consumption without increasing schedule lengths of parallel applications. Experimental results clearly show that the TADVS algorithm is conducive to reducing energy dissipation in large-scale clusters without adversely affecting system performance.

In the first set of experiments, we varied CCR from 0.1 to 1 to examine the performance impacts of communication intensity on our TADVS scheduling strategy. This year, we evaluate the performance of TADVS algorithm by comparing the traditional NDS scheme:



TADVS consistently consumes less energy regardless of the value of CCR (Communication-Computation Ratio) [Ruan et al., 2007]. For example, TADVS conserves the energy consumption for the SPA application by up to 16.8% with an average of 10.7%. When one increases CCR from 0.1 to 1, the energy consumption gradually goes up. This can be explained by the fact that a high CCR results in high communication cost, which in turn leads to the increased total energy consumption. More interestingly, we observe from Fig. 4 that energy savings achieved by the TADVS strategy become more pronounced when the communication intensity is relatively low. This result clearly indicates that low communication intensity offers more space for TADVS to reduce voltage supplies of computing nodes to significantly conserve energy. In other words, applications with low communication intensities can greatly benefit from the TADVS scheduling scheme.

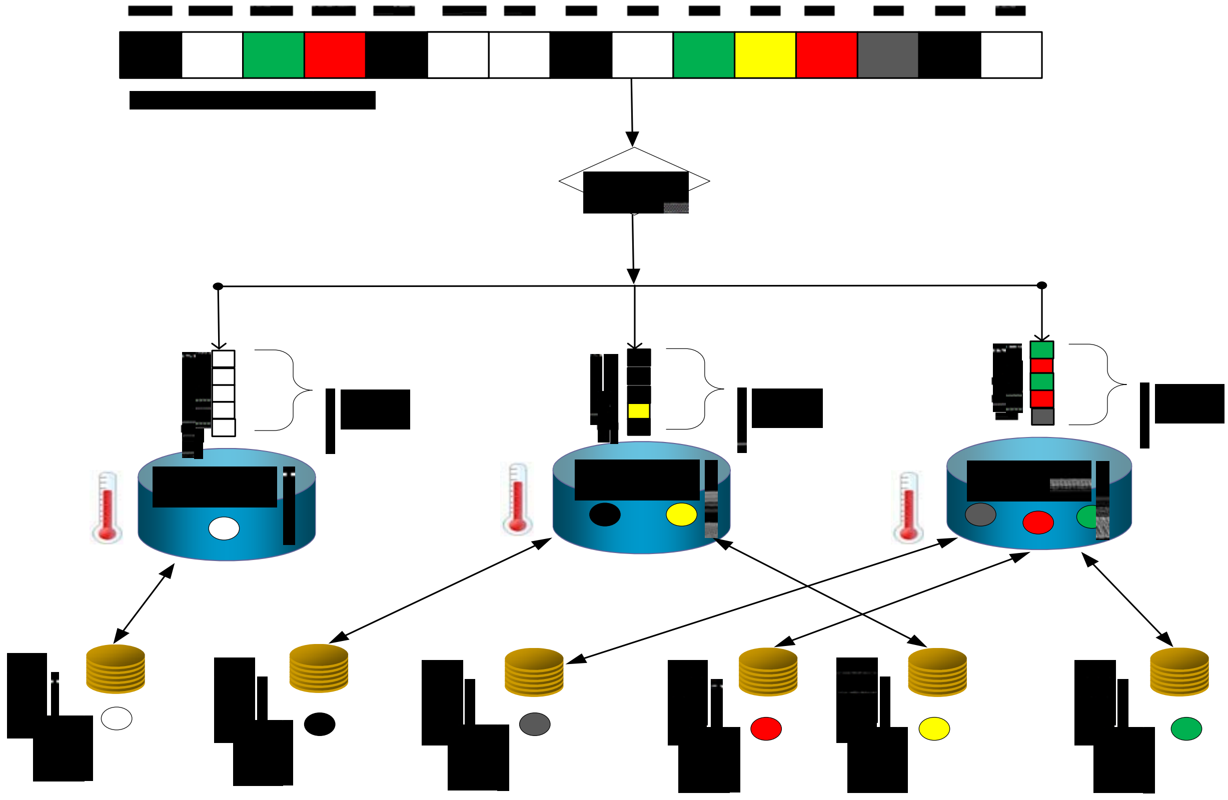
Figure below shows the energy consumption caused by the Gaussian application on the cluster with Intel Pentium 4 processors, whereas First of all, the experimental results reveal that TADVS can save energy consumption for the Gaussian application by up to 14.8% with an average of 9.6%. Second, the results plotted in Figs. 5 and 6 show that compared the Gaussian application with the SPA application, the energy saving rate of TADVS is less sensitive to the communication intensity. The empirical results suggest that the sensitivity of the energy saving rate of TADVS on communication intensity partially relies on the characteristics of parallel applications. Note that parallel applications' characteristics include parallelism degrees, number of messages, average message size, and the like. Compared with the SPA application (Fig. 1), the Gaussian application has a higher parallelism degree. More specifically, we concluded from the experimental results shown in Figs. 4 and 6 that the energy saving rate of TADVS is less sensitive to the communication intensity of parallel applications with higher parallel degrees. Moreover, parallel applications with higher parallelism degrees are able to take more advantages from the TADVS in terms of energy conservation. A practical implication of this observation is that although high communication intensities of parallel applications tends to reduce energy saving rates of TADVS, increasing parallel degrees of the parallel applications can potentially and noticeably boost up the energy saving rates.



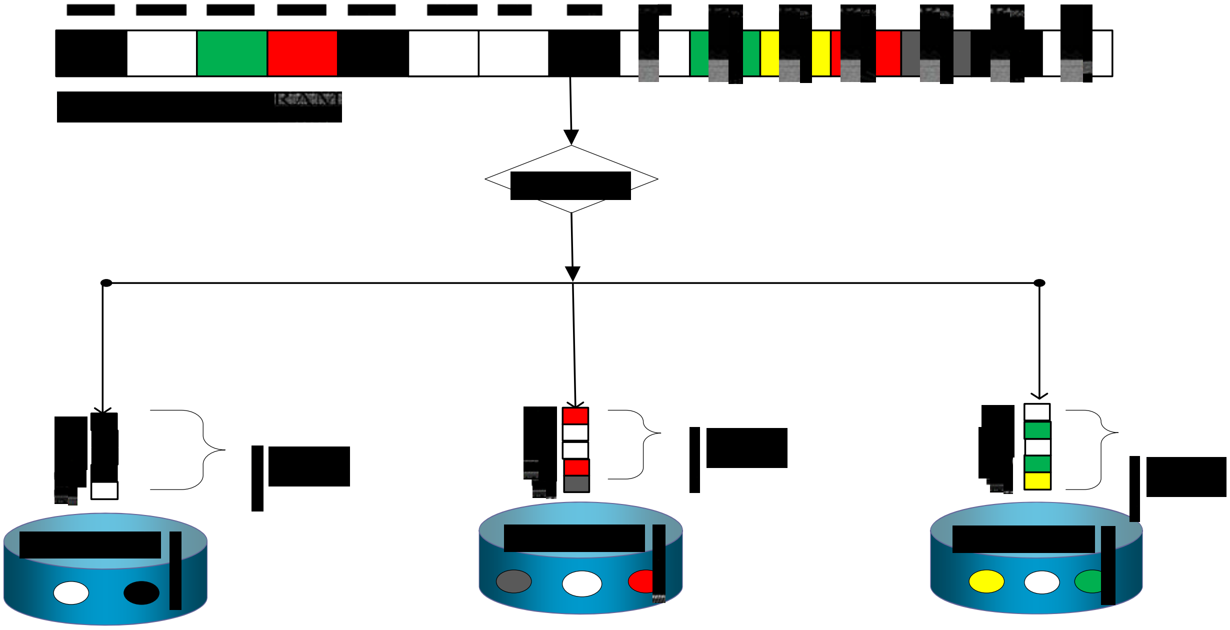
[Zong et al., 2007d] High performance clusters and parallel computing technology are experiencing their golden ages because of the convergence of four critical momentums: high performance microprocessors, high-speed networks, middleware tools, and highly increased needs of computing capability. With forceful aid of cluster computing technology, complicated scientific and commercial applications like human genome sequence programs, universe dark matter observation and the Google search engine have been widely deployed and applied. Although clusters are cost-effective high-performance computing platforms, energy dissipation in large clusters is excessively high. Most previous studies in cluster computing focused on performance, security, and reliability, completely ignoring the issue of energy conservation. Therefore, designing energy-efficient algorithms for clusters, especially for heterogeneous clusters, becomes highly desirable. This year, we developed two novel scheduling strategies, called Energy-Efficient Task Duplication Scheduling (EETDS) and Heterogeneous Energy-Aware Duplication Scheduling (HEADUS), which attempt to make the best tradeoffs between performance and energy savings for parallel applications running on heterogeneous clusters. Our algorithms are based on the duplication-based heuristics, which are efficient solutions to minimize communication overheads among precedence constrained parallel tasks. Our algorithms consist of two major phases. Phase one is used to optimize performance of parallel applications and the second phase aims to provide significant energy savings. We present extensive simulation results using realistic parallel applications to prove the efficiency of our algorithm.

[Zong et al., to be published] Improve performance and conserve energy are two conflict objectives in parallel storage systems. In this project, we proposed a novel solution to achieve the twin objectives of maximizing performance and minimizing energy consumption of large scale parallel disk arrays. We observed that buffer disks can be a performance bottleneck of an energy-efficient parallel disk system. We developed a heat-based and duplication-enabled load balancing strategies to successfully overcome the natural shortcoming of the BUD architecture, in which the limited number of buffer disks are very likely become the bottleneck.

The basic idea of heat-based mapping is that blocks in data disks will be mapped to buffer disks based on their heat (access frequency). Our goal is to make the accumulated heat of data blocks allocated to each buffer disk is exactly the same or at least close. In other words, the temperature of each buffer disk should be in the same level. Here the temperature of a buffer disk means the total heat of all blocks existing in this buffer disk. For example, in current request queue, the heat of block 1-6 are 5, 4, 1, 2, 1, 2 respectively. Therefore, block 1 is cashed to buffer disk 1, block 2 and 3 are copied to buffer disk 2 and block 4, 5 and 6 are mapped to buffer disk 3. In this way, the temperature of each buffer disk is 5. The following figure depicts the dispatch results of heat-based load balancing strategy. Note that there are 15 requests cashed in the RAM buffer and they are going to be dispatched to different buffer disks by the controller. Requests have different colors, which represents they will access different data blocks. For example, request 1 will access the data block1 existing in data disk 1 and request 6 will access the data block6 existing in data disk 6.



The following figure shows another way to do load balancing, i.e. we can duplicate the most popular data blocks to several buffer disks. The basic idea of duplication based load balancing is to move multiple replicas of “hot” data blocks to different buffer disks, which allow multiple buffer disks to serve the requests in parallel thereby improving the performance. In this example, the controller generates a load balanced dispatching by duplicating block 3 in each of the three buffer disks. To decide which block should be duplicated, we also need to calculate and order the heat of data blocks.



To validate the efficiency of the proposed framework and load balancing strategies, we conduct extensive simulations using both synthetic workload and real-life traces.

[Manzanares et al., to be published] Large-scale parallel disk systems are frequently used to meet the demands of information systems requiring high storage capacities. A critical problem with these large-scale parallel disk systems is the fact that disks consume a significant amount of energy. To design economically attractive and environmentally friendly parallel disk systems, we developed two energy-aware prefetching strategies (or PRE-BUD for short) for parallel I/O systems with disk buffers.

First, we studied a concrete example, which is based on the synthetic disk trace presented in the table below.

Synthetic Trace

Time	**0**	**5**	**10**	**15**	**20**	**25**	**30**	**35**	**40**	**45**	**50**	**55**	**60**	**65**	**70**	**75**	**80**	**85**	**90**	**95**
Block	**A1**	**A2**	**B1**	**B2**	**D1**	**D2**	**C1**	**C2**	**B2**	**B1**	**A1**	**A2**	**C1**	**C2**	**D1**	**D2**	**B2**	**B1**	**A1**	**A2**

Disk Parameters (IBM 36Z15)

X = Transfer Rate =55 MB/s	**P_{Act} = Power Active = 13.5 W**
P_{Idle} = Power Idle = 10.2 W	**P_{Stdby} = Power Standby = 2.5 W**
E_{AS} = Energy Active to Sleep = 13.0 J	**E_{SA} = Energy Sleep to Active = 135.0 J**
T_{AS} = Time Active to Sleep = 1.5 s	**T_{SA} = Time Sleep to Active = 10.9**

The requests all have the size of 275MB. This means each request will take approximately 5s to complete. This length was chosen, so seek and rotational delays would be negligible. There are N=4 disks used in this example, where each disk is given a unique letter. Each disk has two different data sections requested multiple times throughout the example. The example demonstrates only large sequential reads. This was also chosen to simplify our example to allow us to demonstrate the potential benefits of our approach. We also assume that all the data can be buffered which causes a small percentage of data to be accessed 100% of the time. This is only used for our motivational example and our simulation results vary this parameter to model real-world conditions. We also assume these strategies can be handled off-line meaning we have prior knowledge of the complete disk request pattern.

Non-Energy Aware Results

$T_{Idle(A)}$	70s	$T_{Idle(B)}$	70s
$T_{Idle(C)}$	80s	$T_{Idle(D)}$	80s
$T_{Act(A)}$	30s	$T_{Act(B)}$	30s
TAct(C)	20s	$T_{Act(D)}$	20s
ETrans(A)	0J	$E_{Trans(B)}$	0J
ETrans(C)	0J	$E_{Trans(D)}$	0J
$T_{E(A)}$	1119J	$T_{E(B)}$	1119J
$T_{E(C)}$	1086J	$T_{E(D)}$	1086J
T_{ES}	4410J		

Energy Aware Results

$T_{Idle(A)}$	0s	$T_{Idle(B)}$	10s
$T_{Idle(C)}$	0s	$T_{Idle(D)}$	0s
$T_{Act(A)}$	30s	$T_{Act(B)}$	30s
$T_{Act(C)}$	20s	$T_{Act(D)}$	20s
$T_{Sleep(A)}$	45.2s	$T_{Sleep(B)}$	33.7s
$T_{Sleep(C)}$	53.7s	$T_{Sleep(D)}$	53.7s
$E_{Trans(A)}$	296J	$E_{Trans(B)}$	309J
$E_{Trans(C)}$	309J	$E_{Trans(D)}$	309J
$T_{E(A)}$	814J	$T_{E(B)}$	900.25J
$T_{E(C)}$	713.25J	$T_{E(D)}$	713.25J
T_{ES}	3140.75J		

PRE-BUD Approach 1

$T_{Idle(A)}$	0s	$T_{Idle(B)}$	0s
$T_{Idle(C)}$	0s	$T_{Idle(D)}$	0s
$T_{Act(A)}$	10s	$T_{Act(B)}$	10s
$T_{Act(C)}$	10s	$T_{Act(D)}$	10s
$T_{Sleep(A)}$	100s	$T_{Sleep(B)}$	100s
$T_{Sleep(C)}$	100s	$T_{Sleep(D)}$	100s
$E_{Trans(A)}$	13J	$E_{Trans(B)}$	13J
$E_{Trans(C)}$	13J	$E_{Trans(D)}$	13J
$T_{E(A)}$	398J	$T_{E(B)}$	398J
$T_{E(C)}$	398J	$T_{E(D)}$	398J
BD_{EPre}	540	BD_{ETot}	1350J
T_{ES}	3482J		

PRE-Bud Approach 2

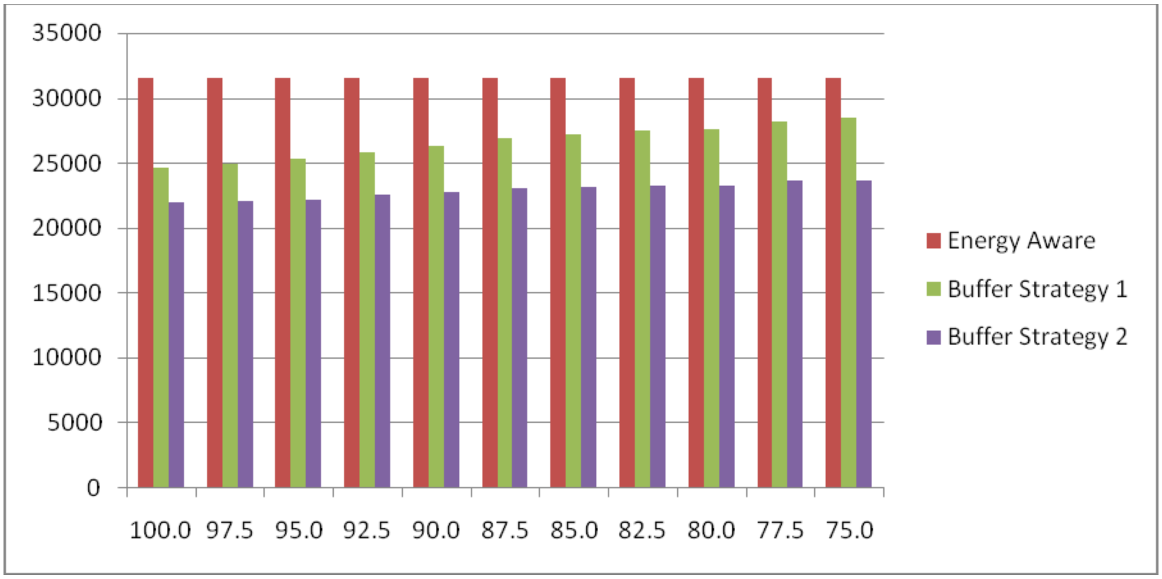
$T_{Idle(A)}$	0s	$T_{Idle(B)}$	0s
$T_{Idle(C)}$	0s	$T_{Idle(D)}$	0s

$T_{Act(A)}$	140s	$T_{Act(B)}$	10s
$T_{Act(C)}$	10s	$T_{Act(D)}$	10s
$T_{Sleep(A)}$	0s	$T_{Sleep(B)}$	100s
$T_{Sleep(C)}$	100s	$T_{Sleep(D)}$	100s
$E_{Trans(A)}$	0J	$E_{Trans(B)}$	13J
$E_{Trans(C)}$	13J	$E_{Trans(D)}$	13J
$T_{E(A)}$	1890J	$T_{E(B)}$	398J
$T_{E(C)}$	398J	$T_{E(D)}$	398J
BD_{EPre}	540	BD_{ETot}	1350J
T_{ES}	3084J		

The results presented in the above four Tables gave us some promising initial results. The two approaches using a buffer disk provided significant energy savings over the non-energy aware parallel disk storage system. This has the benefit of not impacting the capacity of the large-scale parallel disk system. The other main benefit of our first approach is the fact that state transitions are lowered as compared to the energy aware baseline.

Second, we design a prefetching approach that utilizes an extra disk to accommodate prefetched data. Third, we develop a second prefetching strategy that makes use of an existing disk in the parallel disk system as a buffer disk. Compared with the first prefetching scheme, the second approach lowers the capacity of the parallel disk system. However, the second approach is more cost-effective and energy-efficient than the first prefetching technique.

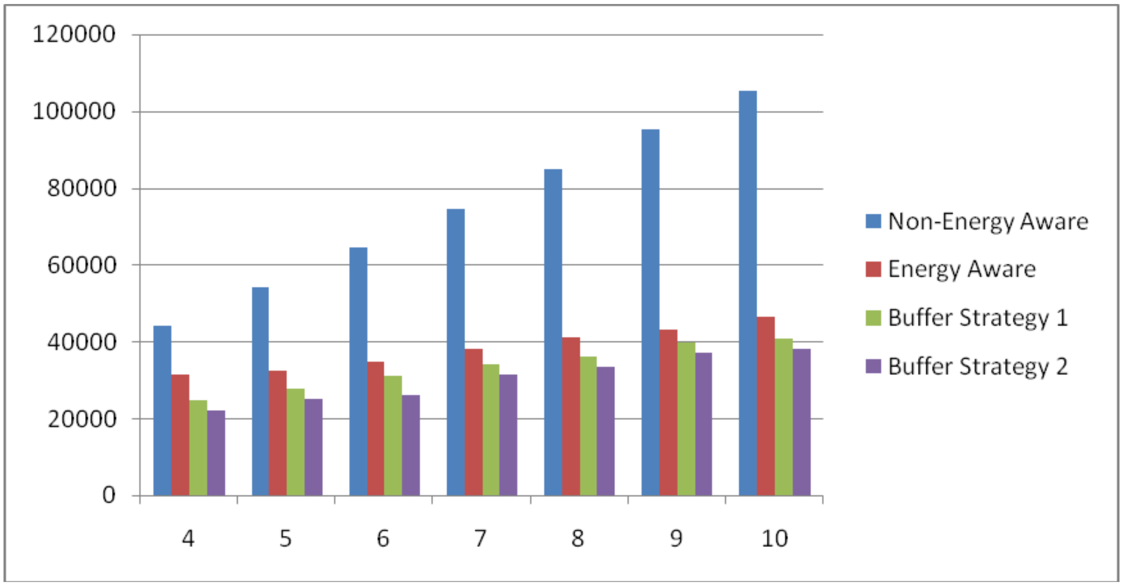
Finally, we quantitatively compare both of our prefetching approaches against two conventional strategies including a dynamic power management technique and a non-energy-aware scheme. The results obtained from the Figure below held the number of disks to 4 and also kept the data size of each request at 275MB. We have omitted the results of the non-energy aware approach, since they are constant and higher than the energy aware strategy. As expected the performance of both of our strategies were lowered when the hit rate was decreased. This is expected since our motivational example demonstrated a best case scenario. Disk sleep times are lowered once a miss is encountered. This is due to the fact that a disk has to wake up to serve the request. This will increase the energy consumption of disks that have to serve the missed requests. This leads to an increase in the total energy consumption of the entire system. Our buffered large-scale parallel disk system is still able to consume less energy than the energy aware approach. The energy aware and non-energy aware disk systems are not affected by buffer disk miss rates.



The first buffer disk approach begins to approach the same level of performance as the energy aware strategy. It is only able to save 10% energy over the energy aware strategy when the hit rate is 75%. This is because adding the extra disk puts extra energy requirements on the system, and lowering the hit rate further impacts the energy benefits of the first strategy. The second buffered disk approach is still able perform 25% better than the energy aware approach. This is because there is not the extra energy penalty of adding an extra disk. The capacity of your disk system will be lowered using this approach.

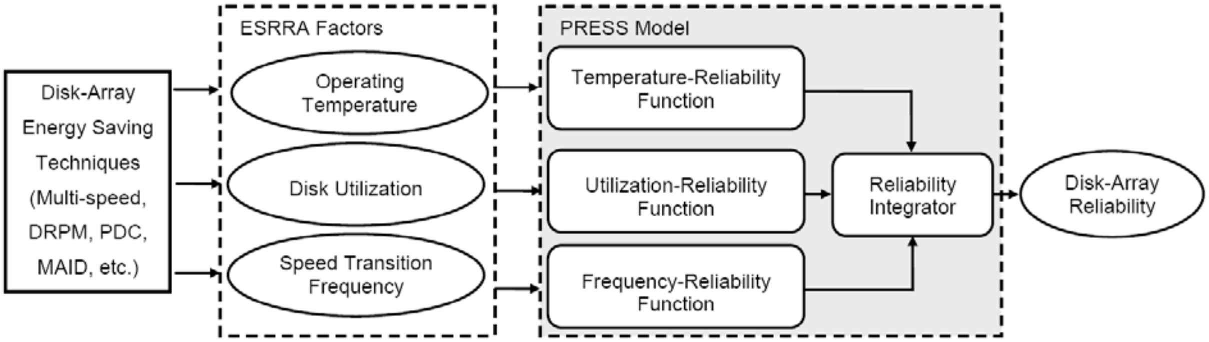
The hit rate becomes a very important factor in the performance of our approaches. If the buffer disk is constantly missing requests then both strategies will eventually downgrade to the energy aware approach. Fortunately applications have been documented to request 20% of the data available 80% of the time. Our heuristic based approach would work considerably well in this case. This is modeled by the 80% hit rate. The buffered disk approach one and two are able to save 12% and 26% energy over the energy aware strategy when the hit rate is 80%. Similarly, they are able to save 37% and 47% of the total energy compared to the non-energy aware approach.

The first buffer disk approach downgrades more quickly than the second approach as the hit rate is decreased as compared to the energy aware approach. This is not that great of a concern since the first strategy is still able to have a positive impact on the reliability of the as compared to the energy aware approach. The first buffer disk approach is still able to produce significant energy savings over the non-energy aware approach without compromising the reliability of the system. The energy savings performance of the second approach does not diminish as quickly as the first approach, but there will be an impact on the capacity of the system. The second approach is also able to reduce the number of state transitions.



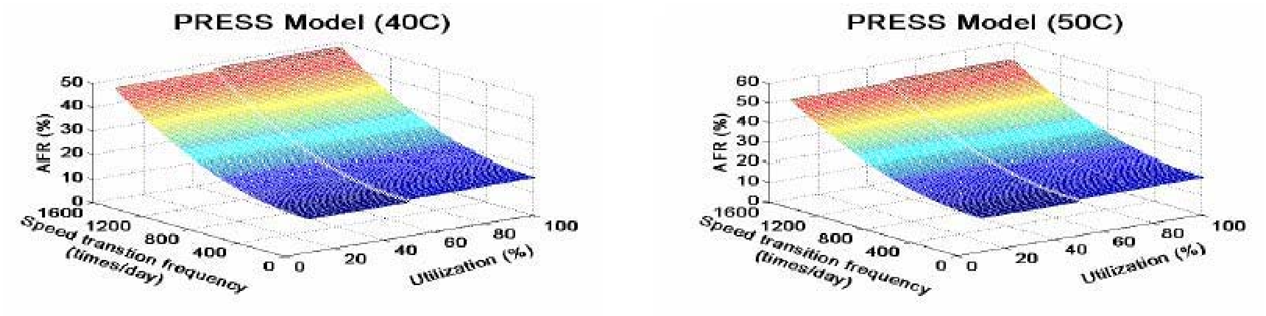
From the above Figure we are able to see that the non-energy aware approach wastes a considerably larger amount of energy as compared to all of the energy aware approaches. This is expected since the non-energy aware approach is not able to place disks in the standby mode. Buffer strategy 1 was able to produce a 12% increase in energy savings over the energy aware strategy when 10 disks were simulated. Similarly, buffer strategy performed even better with an 18% increase. This is expected again because of the energy overhead adding an extra disk Buffer Strategy 1 requires.

[Xie and Sun, 2008a] Mainstream energy conservation schemes for disk arrays inherently affect the reliability of disks. A thorough understanding of the relationship between energy saving techniques and disk reliability is still an open problem, which prevents effective design of new energy saving techniques and application of existing approaches in reliability-critical environments. As one step towards solving this problem, we investigated an empirical reliability model, called Predictor of Reliability for Energy Saving Schemes (PRESS). The architecture of the PRESS model is given below:



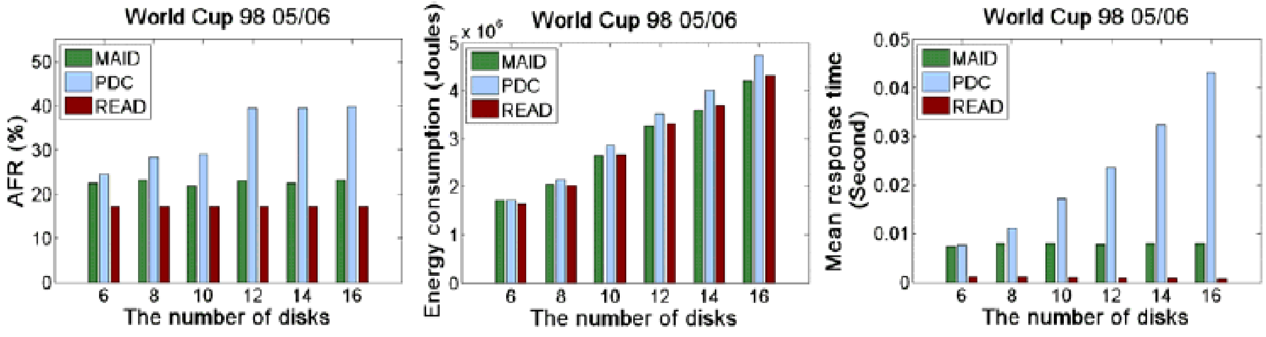
Disk-Array Energy Saving Techniques (Multi-speed, DRPM, PDC, MAID, etc.)
ESRRA Factors
Operating Temperature
Disk Utilization
Speed Transition Frequency
PRESS Model
Temperature-Reliability Function
Utilization-Reliability Function
Frequency-Reliability Function
Reliability Integrator
Disk-Array Reliability

Fed by three energy-saving-related reliability-affecting factors, operating temperature, utilization, and disk speed transition frequency, PRESS estimates the reliability of entire disk array. In what follows, we present two 3-dimennsional figures to represent the PRESS model at operating temperature 40 C (Figure 5a) and 50 C (Figure 5b), respectively.



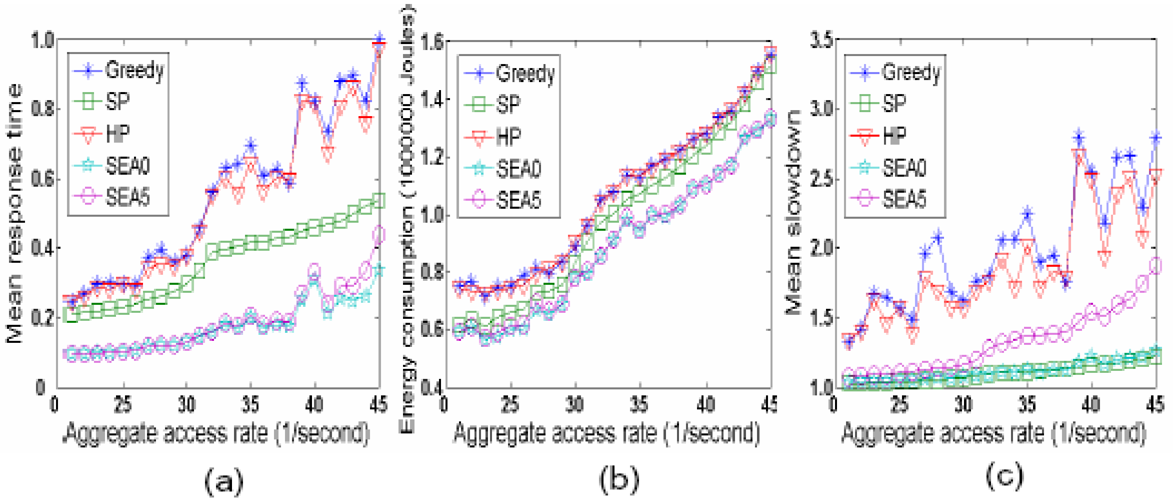
(a) (b)

Further, we developed a new energy saving strategy with reliability awareness called Reliability and Energy Aware Distribution (READ) is developed in the light of the insights provided by PRESS. Experimental results demonstrate that compared with existing energy saving schemes, MAID and PDC, READ consistently performs better in performance and reliability while achieving a comparable level of energy consumption.



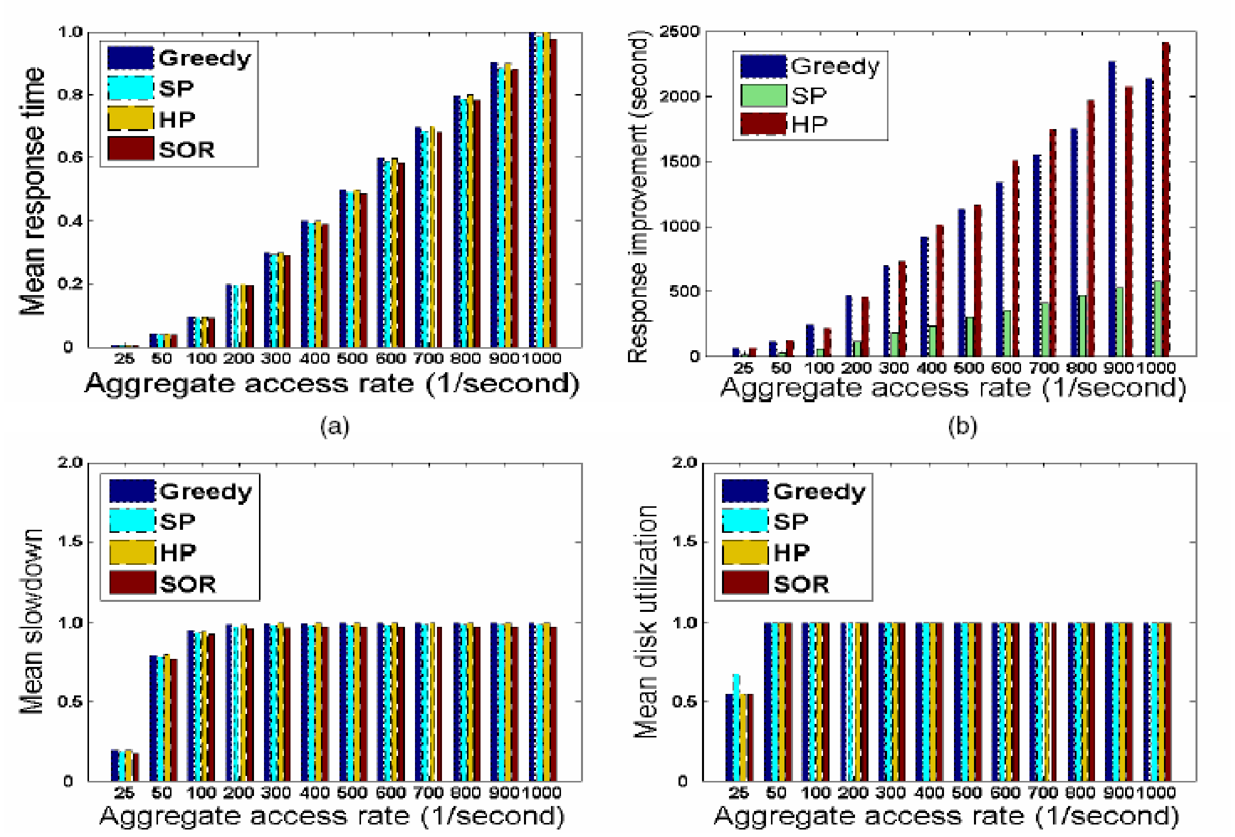
(a) Reliability (b) Energy consumption (c) Mean response time

[Xie and Sun, 2007] Many real-world applications like Video-On-Demand (VOD) and Web servers require prompt responses to access requests. However, with an explosive increase of data volume and the emerging of faster disks with higher power requirements, energy consumption of disk based storage systems has become a salient issue. To achieve energy-conservation and prompt responses simultaneously, in this study we propose a novel energy-saving data placement strategy, called Striping-based Energy-Aware (SEA), which can be applied to RAID-structured storage systems to noticeably save energy while providing quick responses. Further, we implement two SEA-powered RAID-based data placement algorithms, SEA0 and SEA5, by incorporating the SEA strategy into RAID-0 and RAID-5, respectively. Extensive experimental results demonstrate that (see the three figures below) compared with three well-known data placement algorithms Greedy, SP, and HP, SEA0 and SEA5 reduce mean response time on average at least 52.15% and 48.04% while saving energy on average no less than 10.12% and 9.35%, respectively.



[Xie, 2007] and [Madathil et al., 2008] The problem of statically assigning nonpartitioned files in a parallel I/O system has been extensively investigated. A basic workload characteristic assumption of existing solutions to the problem is that there exists a strong inverse correlation between file access frequency and file size. In other words, the most popular files are typically small in size, while the large files are relatively unpopular. Recent studies on the characteristics of web proxy traces suggested, however, the correlation, if any, is so weak that it can be ignored.

Hence, in this part of study, we raised the following two questions. First, can existing algorithms still perform well when the workload assumption does not hold? Second, if not, can one develop a new file assignment strategy that is immune to the workload assumption? To answer these questions, in this project we first evaluated the performance of three well-known file assignment algorithms with and without the workload assumption, respectively. Next, we developed a novel static file assignment strategy for parallel I/O systems, called static round-robin (SOR), which is immune to the workload assumption.



(a) (b)

(c) (d)

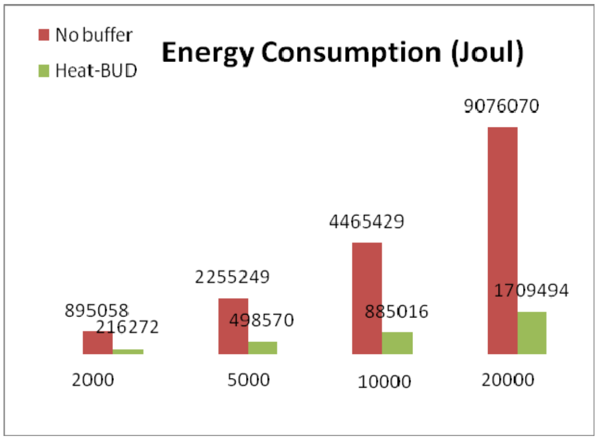
The above four figures show the simulation results for the four algorithms on a parallel I/O disk array with 16 disk drives. We observe that SOR consistently outperforms the three exiting approaches in terms of mean response time. This is because SOR considers both minimizing variance of service time for each disk and fine-tuning load balancing degree. Consequently, the sorted files were continuously assigned to disks such that a more evenly distributed workload allocation scheme was generated. SP takes the second place in mean response time metric, which is consistent with our expectation because it is one of the best existing static file assignment heuristics. To clearly demonstrate the performance improvement, Fig. 2b provides mean response time decrease gained by SOR compared with Greedy, SP, HP, respectively. In particular, SOR can reduce mean response time on average by 1118.3, 1052.8, and 269.6 seconds, compared with HP, Greedy, and SP, respectively. An interesting observation is that the mean response time improvement becomes more significant when the overall workload represented by the aggregate access rate increases. The implication is that SOR exhibits its strength in situations where system workload is heavy. In terms of mean slowdown, SOR also performs best among the four heuristics (Fig. c), which is consistent with the results shown in Fig. a. Since the total workload is relatively heavy, the mean disk utilization in Fig. d quickly arises to 1 when aggregate access rate is larger than 25 (1/second).

[Zong et al., 2008] [Zong et al., to be published] To conserve energy, BUD makes an effort to place most data disks will run in the low power state, thereby directing most traffic to a limited number of buffer disks. This can potentially make the buffer disks overloaded and become the performance bottleneck. Load balancing is one of the best solutions for the inherent shortcoming of the BUD architecture. Basically, there are three types of load balancing strategies called non-random load balancing, random load balancing, and redundancy load balancing. Sequential mapping belongs to non-random load balancing, because the buffer disks have fixed mapping relationships with specific data disks. The round-robin mapping is a typical random load balancing strategy by allocating data to each buffer disk with equal portions and in order. Redundancy load balancing strategies for storage systems include EERAID, eRAID, and RIMAC. We designed a heat-based load balancing strategy, which is also a random load balancing strategy. The primary objective of our strategy is to minimize the overall response time of disk requests by keeping all buffer disks equally loaded.

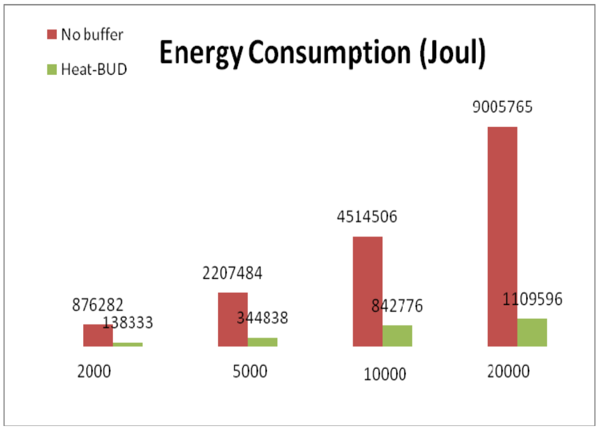
In contrast to sequential and round robin mapping algorithms, a heat-based mapping strategy is proposed to achieve load balancing among buffer disks. The basic idea of heat-based mapping is that blocks in data disks are mapped to buffer disks based on their heat, i.e. frequencies of accesses. Our goal is to make the accumulated heat of data blocks allocated to each buffer disk the same or close to this ideal situation. In other words, the temperature, or the workload of each buffer disk should be the same. The temperature of a buffer disk is the total heat of all blocks existing in the buffer disk. For example, suppose all blocks have the same data size, the heat of blocks 1-6 is 5, 4, 1, 2, 1, and 2, respectively. Then, block 1 is cached to buffer disk 1, blocks 2 and 3 are copied to buffer disk 2 and blocks 4, 5 and 6 are mapped to buffer disk 3. With this mapping in place, the temperature of each buffer disk is 5.

This algorithm will periodically collect the requests waiting in the queue, analyze the target block of each read request, and calculate the heat of each unique block. If the target block cannot be found in the buffer disk, the controller initiates a data miss command. This in turn will wake up the corresponding data disk in order to copy the block to the buffer disk that has the lowest temperature. In a special case, the selected buffer disk may not have free space to store a new data block. The controller will seek the next buffer disk with a temperature that is higher than the initial buffer disk selected, but still lower than any other buffer disks. In the worst case, no candidate buffer disk will be found because all buffer disks are full. A data replacement function using the Least Recently Used (LRU) algorithm will be executed to evict some existing data blocks. If the target block has already been cached in one of the buffer disks, then that buffer disk will serve the corresponding request. Once the algorithm has made the decision how to dispatch these requests, the block heat and buffer disk temperature need to be recalculated and updated accordingly. Since this is an online algorithm, the decision made at the current time period relies on the heat and temperature information collected in the last time period.

This set of experimental results aims at evaluating the energy efficiency of the buffer disk based parallel storage systems. To fairly compare the results, we generated and executed a large number of requests and simulated both large reads (average data size is 64MB) and small reads (average data size is 64KB). The following two Figures plot the total energy consumption of NO-buffer and Heat-BUD running 2000, 5000, 10000, and 20000 large read requests and small read requests, respectively. There are three important observations here. First, the BUD can significantly conserve energy compared with No-Buffer parallel storage systems. Second, the more requests BUD serves, the more potential power savings is revealed. For example, BUD outperforms No-Buffer in terms of energy conservation by 75.83%, 77.89%, 80.18% and 81.16% for 2000, 5000, 10000, and 20000 large reads, respectively. This is expected because more requests lead to more opportunities for BUD to keep the data disks in the sleep state. Third, BUD performs better for small reads (average 84.4% improvement) than large reads (average 78.77% improvement). The reason is that BUD consumes more energy when moving large data blocks to buffer disks.

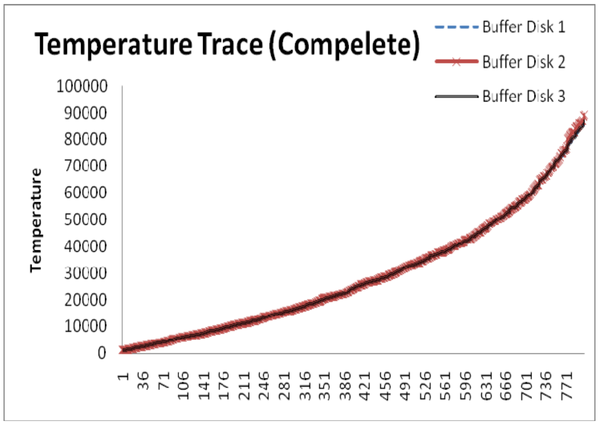


Energy consumption for large reads



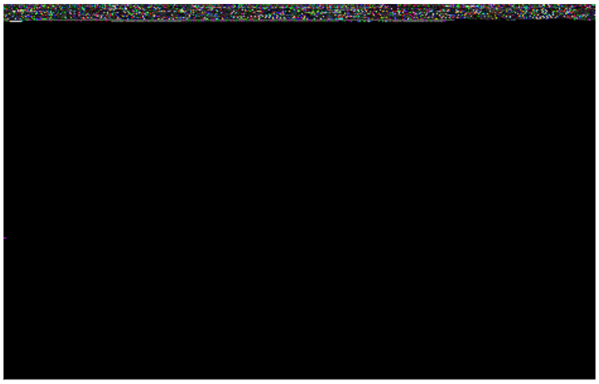
Energy consumption for small reads

In this part of the study, we evaluated the load balancing ability of the heat-based algorithm. Recall that the temperature of a buffer disk clearly indicates how busy it is. The Figure below records the temperature of three buffer disks when BUD is processing 800 requests. We can see that the three temperature curves merge together most of the time, which means that the three buffer disks are almost equally loaded for most of the simulation time.

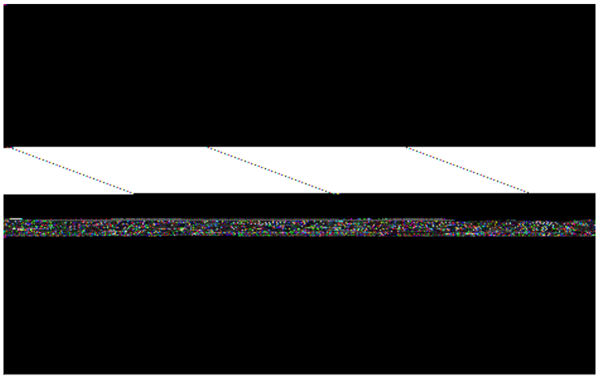


Temperature tracking trace

In order to identify the information hidden in the above Figure and understand how the dynamic load balancing works, we plot the initial stage, intermediate stage of the temperature tracking trace in the following two Figures. At the initial stage, the three buffer disks are not load balanced. Buffer disk 2 is the busiest disk and buffer disk 1 is lightly loaded. Therefore, the heat-based algorithm will keep allocating requests to buffer disk 1. We observe that the temperature of buffer disk 1 keeps growing and it catches buffer disk 3 first. After that, the temperatures of buffer disk 1 and 3 cross-rise for a while and then they catch buffer disk 2. At this point, the system is load balanced for the first time. Fig. 5 shows that the entire system is perfectly load balanced in the intermediate stage because the temperatures of three buffer disks rise in turns.

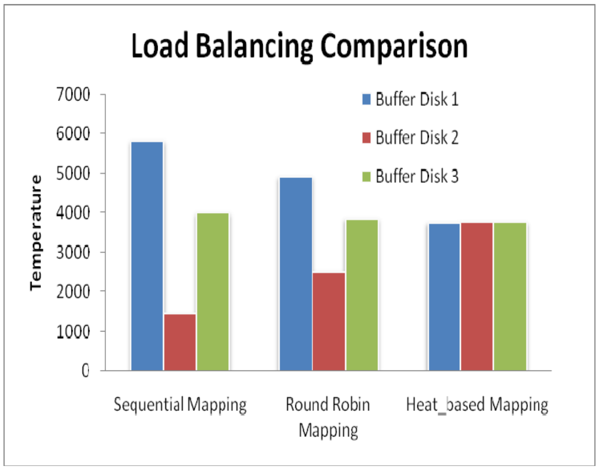


Temperatures in initial stage



Temperatures in intermediate stage

To compare the load balancing efficiency of sequential mapping, round robin mapping, and heat-based mapping, we tested 2500 requests with average data size of 4MB using these three mapping strategies. The simulation results depicted in the Figure below prove that our heat-based mapping is the most efficient algorithm that achieves load balancing. In addition, the random mapping method (e.g., round robin mapping) outperforms non-random mapping strategies (e.g., sequential mapping) in terms of load balancing.

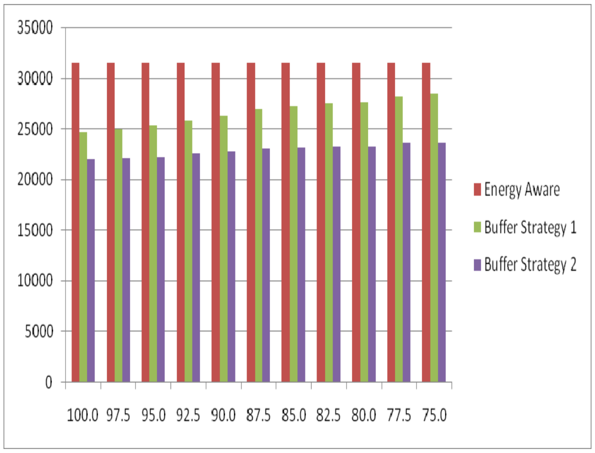


Load balancing comparison

[Manzanares et al., 2008a] The prefetching algorithm uses the frequency that a block is requested as a heuristic. The first step of the algorithm iterates over all of the requests and counts the references for each unique block. Then it sorts the list of unique blocks by the number of references to each block. At this point the algorithm puts the highly requested blocks into the buffer until it is full.

The last part of the algorithm also iterates over all requests trying to figure out how long each disk can sleep. If a block requested for a disk is in the buffer disk, the corresponding data disk can sleep longer. The buffer disk handles the request and the distance between requests on the same disk becomes cumulative. If a requested block is not in the buffer the disk must be woken up to serve the request, this is handled by the power management mechanism. The distance is then set to zero since the disk had to be woken up. Using the frequently accessed heuristic the PRE-BUD strategy should have a small performance impact on the system. Almost all steps of the algorithm run linearly with respect to the number of requests. The only step that is not linear is the phase that sorts the list of requests according to their frequency. Sorting is a common procedure and is known to have a best-case run-time of O(nlogn), where n is the number of data request to be sorted. The PRE-BUD strategy is able to have a run time of O(n + nlogn) using an efficient sorting algorithm. The PRE-BUD strategy is not assumed to be optimal, since the requested blocks are sorted using their frequency. The frequency is used as a heuristic to select blocks to be placed in the buffer. An optimal strategies goal would be to select the requests to be placed in the buffer that produce the largest impact on the standby time of disks. The standby time of each disk is directly related to the energy savings of the system.

The results displayed in the Figure below held the number of disks to 4 and also kept the data size of each request at 275MB. We have omitted the results of the non-energy aware approach, since they are constant and higher than the energy aware strategy. Buffer Strategy 1 adds an extra disk and Buffer Strategy 2 uses an existing disk as the buffer disk. As expected the performance of both of our strategies were lowered when the hit rate was decreased. Disk sleep times are lowered once a miss is encountered. This is due to the fact that a disk has to wake up to serve the request. This will increase the energy consumption of disks that have to serve the missed requests. This leads to an increase in the total energy consumption of the entire system. Our buffered large-scale parallel disk system is still able to consume less energy than the energy aware approach. The energy aware and non-energy aware disk systems are not affected by buffer disk miss rates.



Energy Savings (J) vs. Hit Rate

As the hit rate is lowered, the first buffer disk approach begins to approach the same level of performance as the energy aware strategy. It is only able to save 10% energy over the energy aware strategy when the hit rate is 75%. This is because adding the extra disk puts extra energy requirements on the system, and lowering the hit rate further impacts the energy benefits of the first strategy. The second buffered disk approach is still able perform 25% better than the energy aware approach. This is because there is not the extra energy penalty of adding an extra disk. The capacity of your disk system will be lowered using this approach.

The hit rate becomes a very important factor in the performance of our approaches. If the buffer disk is constantly missing requests then both strategies will eventually downgrade to the energy aware approach. Fortunately applications have been documented to request 20% of the data available 80% of the time. Our heuristic based approach would work considerably well in this case. This is modeled by the 80% hit rate. The buffered disk approach one and two are able to save 12% and 26% energy over the energy aware strategy when the hit rate is 80%. Similarly, they are able to save 37% and 47% of the total energy compared to the non-energy aware approach.

The first buffer disk approach downgrades more quickly than the second approach as the hit rate is decreased as compared to the energy aware approach. This is not that great of a concern since the first strategy is still able to have a positive impact on the reliability of the system as compared to the energy aware approach. The first buffer disk approach is still able to produce significant energy savings over the non-energy aware approach without compromising the reliability of the system. The energy savings performance of the second approach does not diminish as quickly as the first approach, but there will be an impact on the capacity of the system. The second approach is also able to reduce the number of state transitions.

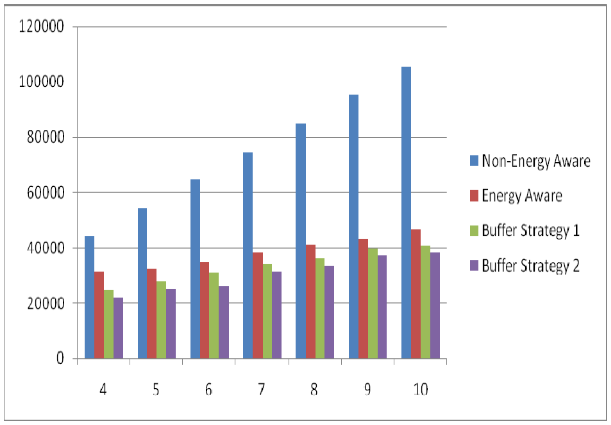


Fig. 8 Energy Savings (J) vs. Number of Disks

From the above Figure we are able to see that the non-energy aware approach wastes a considerably larger amount of energy as compared to all of the energy aware approaches. This is expected since the non-energy aware approach is not able to place disks in the standby mode. Buffer strategy 1 was able to produce a 12% increase in energy savings over the energy aware strategy when 10 disks were simulated. Similarly, buffer strategy performed even better with an 18% increase. This is expected again because of the energy overhead adding an extra disk Buffer Strategy 1 requires. Our approach produces promising results as the number of disks is increased. This is an important observation, since our target system is a large-scale parallel system. This leads us to believe our system will produce energy benefits regardless of the number of disks in a system.

[Bellam et al., 2008] RAID 1 is popular and is widely used for disk drives. RAID 1 is implemented with a minimum of two disks, which are the primary and back disks. Initially the data is stored to the primary disk and then it is mirrored to the backup disk. This mirroring helps to recover the data when there is a failure in the primary disk. It also helps to increase the performance of the RAID 1 system by sharing the workload between the disks. We considered RAID 1 for all of our experiments.

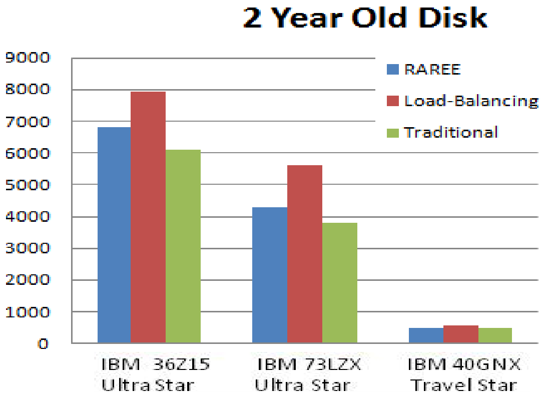
The processor in the system generates the I/O stream, which is queued to the buffer. The utilization of the disk is calculated using the request arrival rate. Please refer to Section 3 for details of the description for the disk utilization model.

It should be noted that all requests here are considered as read requests. At any given point of time the disks can be in the following three states.

- State 1: Both the disks in sleep mode
- State 2: Primary disk active and backup disk in sleep mode
- State 3: Both the disks in active mode and share the load.

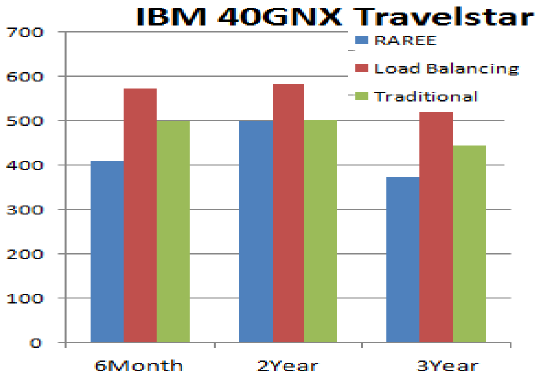
Let us consider that the disks are in state 1 at the beginning. Once the utilization is calculated, it is compared with the safe utilization zone range. If the calculated value falls below the range then disks stay in state 1. If the calculated value is within the range, then the primary disk is made active while the backup disk continues to stay in the sleep mode. This represents a transition to state 2. If the calculated value is beyond the range then both the disks are made active and both of them share the load, which corresponds to state 3. Transition of states from one power mode to another involves disk spin up and/or spin down. The disk spin ups and spin downs also consume a lot of energy.

RAID 1 is used in our experiments. RAID 1 uses a minimum of two disks, one as a primary and one as the backup. We conducted the experiments on three types of disks from IBM.
The experimental results are compared against two traditional state of the art methods. In the first method, load balancing, both the disks are always made active. Load balancing achieves very high performance because both the disks share the load. The second method, traditional method, is where the primary disk is made always active and the backup disk is kept in sleep mode. The backup disk is made active only when the utilization of the primary disk exceeds 100%, also known as saturation. In what follows, we term our approach as RAREE (Reliability aware energy efficient approach).



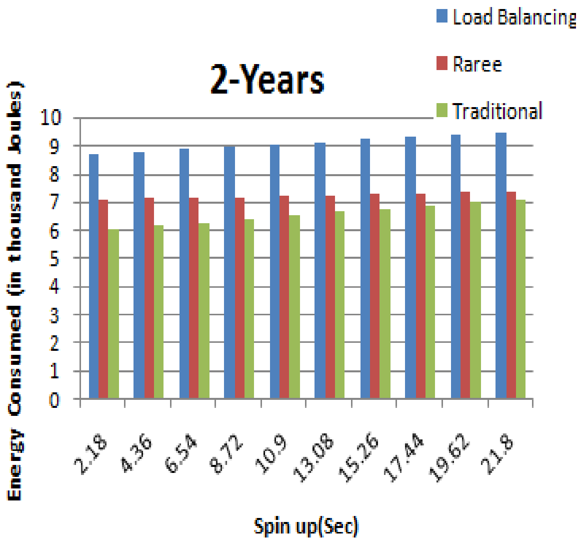
Energy consumed by a 2 year old disk

The experimental data generated from the simulations is plotted in the above Figure, which represents the energy consumed by the 2 year old disk respectively. RAREE is compared against load balancing and the traditional method. From the above Figure it is observed that for the IBM 36Z15 disk the power consumed by RAREE falls in between the load balancing and traditional techniques. Even for the IBM 73LZX the trend is similar, but the difference in values is not as high as IBM 36Z15. For the IBM 40GNX the power consumed by RAREE is smaller than the traditional and load balancing power consumption values because disk spin up and spin down values are much smaller for the IBM 40GNX when compared with the other two disks. It should be observed that the disk spin down and disk spin up values play a vital role in the energy consumption.



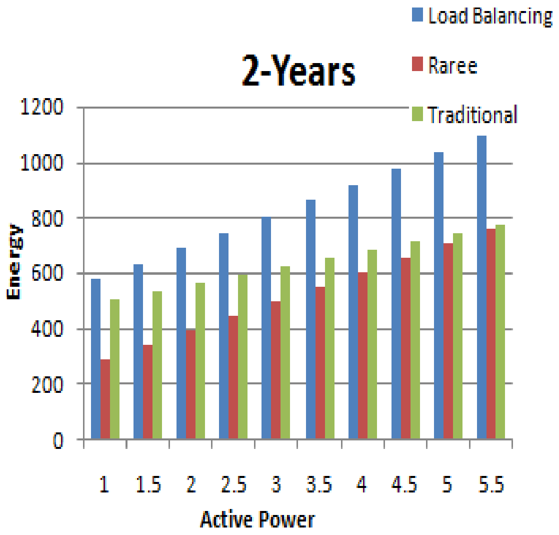
Energy consumed by a IBM 40GNX disk with different ages

The above Figure shows the performance of RAREE on Travelstar disks of different ages. It can be observed from figure that RAREE consumes less energy when compared to traditional and load balancing.



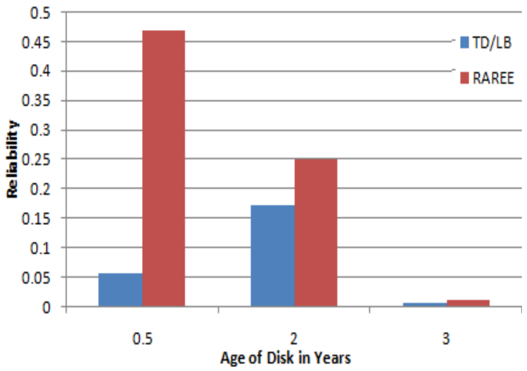
Impact of spin up power on energy

The Figure above shows the effects on energy when the spin up energy is varied for a 2 year old disk. The RAREE energy consumption falls in between traditional and load balancing techniques. Though the energy consumed by RAREE is a little higher than traditional technique, here we are also gaining good amount reliability.



Impact of active power on energy

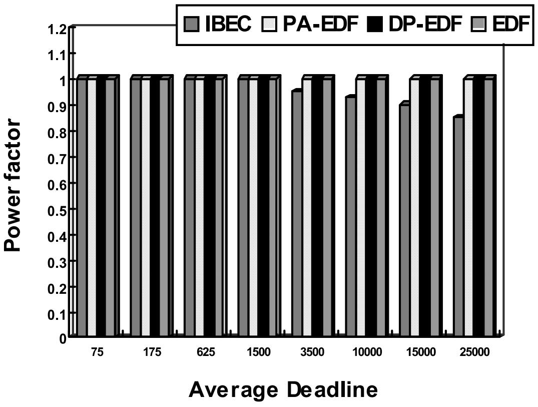
The above Figure shows the change in energy as the active power is varied. Here RAREE energy consumption is definitely less than the two existing techniques, because RAREE makes the disks go to sleep mode as soon there are no requests unlike the other techniques. When idle power is changed unlike the active power the energy consumed by the RAREE again falls between the two techniques. This is because RAREE makes the system go to sleep mode very often depending on the conditions. It should be observed here though the RAREE consumed a bit higher energy than traditional it can be neglected as we are achieving reliability.



Reliability vs. disk ages

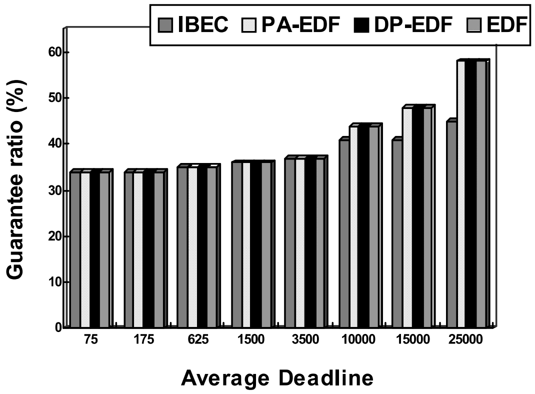
The above Figure is a very important graph here it shows the reliability in terms of annual failure rate percentile. It can be observed from the graph that RAREE achieves a very high reliability when compared to load balancing and traditional. Only one bar is shown for load balancing and traditional techniques because both have the same reliability levels as they don't pay special attention to reliability. We also found an interesting observation that when RAREE is applied to IBM40GNX, which is a travelstar, it definitely consumes much less energy than the other two Ultrastars, which are high performance disks. This makes it clear that RAREE gives best results when it is used on mobile disks instead of high performance disks. This doesn't limit the usage of RAREE to mobile disks because though Ultrastar consumes a little more energy than traditional technique we still get a good reliability at a marginal cost of energy. Simulation results prove that on an average roughly 20% of energy can be saved when RAREe is used instead of load balancing. When RAREe is used instead of the traditional method an excess of 3% of energy is saved, it is not a very significant amount but along with a very little energy saving we are also achieving high reliability which makes it significant.

[Roth et al., 2008] Energy conservation has become a critical problem for real-time embedded storage systems. Although a variety of approaches to reducing energy consumption has been extensively studied, energy conservation for real-time embedded storage systems is still an open problem. In this research, we propose an energy management strategy (IBEC) using I/O burstiness for real time embedded storage systems. Our approach aims at blending the IBEC energy management strategy with low level disk scheduling mechanism to conserve energy consumed by storage systems.



Power consumption vs. deadlines.

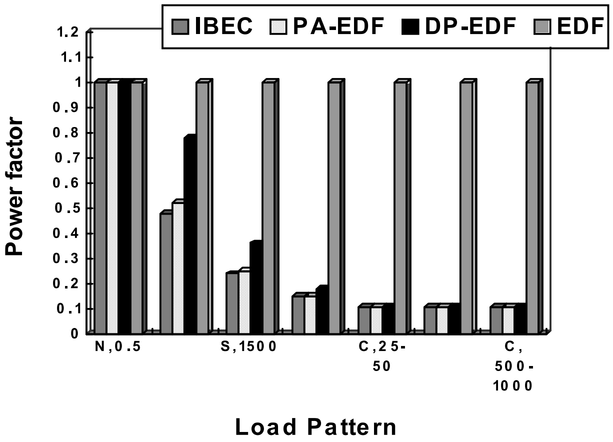
The above Figure shows the power consumption of these four algorithms when average request deadline varies from 75 ms to 25000 ms. We observe from this Figure that each of the four algorithms consumes the same amount of power at the maximal level when the average request deadline is less than 3500 ms. This is because the hard-disk has to be kept active all the time to service the arrival disk requests which have very tight deadlines. In other words, there is no opportunity for IBEC to conserve some power. Therefore, IBEC gracefully degraded to existing power-aware scheduling algorithms like DP-EDF and PA-EDF. When the average request deadline is equal to or larger than 3500 ms, however, IBEC starts to conserve some energy while the three baseline algorithms remain the same performance in power consumption. We attribute the *PF* improvement of IBEC over the three baseline algorithms to the fact that IBEC judiciously employ the loose deadlines to conserve some energy. More interestingly, the improvement of IBEC over the three existing schemes in terms of *PF* is more pronounced when the deadline becomes looser for IBEC can further improve its power consumption performance when more slack time is available. On average, IBEC can save 10.8% power compared with the baseline algorithms.



Guarantee ratio vs. deadlines.

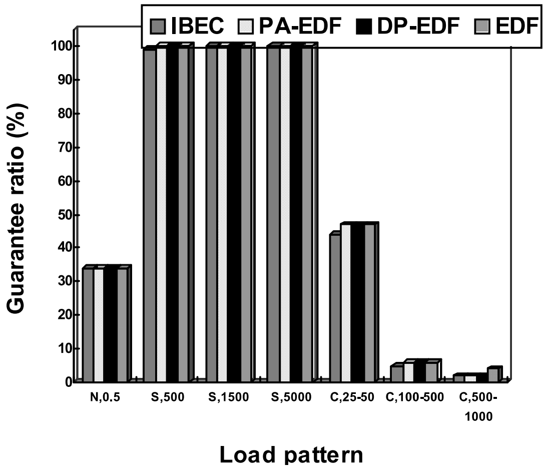
The above Figure plots *GR* of the four algorithms when the deadline is increased from 75 to 25000 ms. It reveals that IBEC performs exactly the same, with respect to *GR*, as all the rest approaches when the deadline is less than 1000 ms. The reason is that the relatively high workload along with the tight deadlines make IBEC only concentrate on guaranteeing arrival requests' timing constraints, which have a higher priority than power conservation requirement. However, the *GR* performance of IBEC suddenly drops off when the deadline is 10,000 ms. In fact, this is an artifact of our specific implementation of the IBEC algorithm. In order to keep simulation times manageable and to closely approximate a real system where no infinite amount of time is available to re-evaluate the schedulability of a queue, we limited the maximum number of requests that IBEC would ensure their deadline constraints. In particular, when the length of the waiting queue of requests is larger than 1,000, our implementation of IBEC will no longer guarantee the schedulability of requests after the 1,000th.

From the following Figure, we can make three important observations. First, all algorithms perform identically in power consumption under the Normal Distribution. Second, the three power-aware algorithms noticeably outperform the EDF scheme, which has no power-awareness at all, when the Sparse Distributions were applied. This is because the nature of the Sparse Distribution decides a relatively large time interval between two continuous disk requests, which in turn gives the three power-aware algorithms chances to switch the hard-disk to "sleep" mode to save energy. Furthermore, IBEC slightly outperforms DP-EDF and PA-EDF, two naïve power-aware algorithms. The rationale behind this phenomenon is that IBEC can make most use of the slack time of each arrival request. Put it in another way, IBEC only wakes up the disk at the last second from which all arrived requests' deadlines can still be met, while DP-EDF only lets the disk sleep for a fixed period of time no matter whether a request is waiting for service or not. Third, for clustered workloads, IBEC and the two naive power-aware techniques perform comparably, and they both significantly perform better than EDF. This is due to the fact that IBEC, DP-EDF, and PA-EDF can put the disk to "sleep" status completely between clusters of requests. Thus, the three power-aware algorithms can substantially save power compared with EDF. The reason why IBEC ties with DP-EDF and PA-EDF is that the performance improvement of IBEC in terms of power consumption essentially depends on slack times of arrival requests rather than the arrival patterns.



Energy vs. workload distribution.

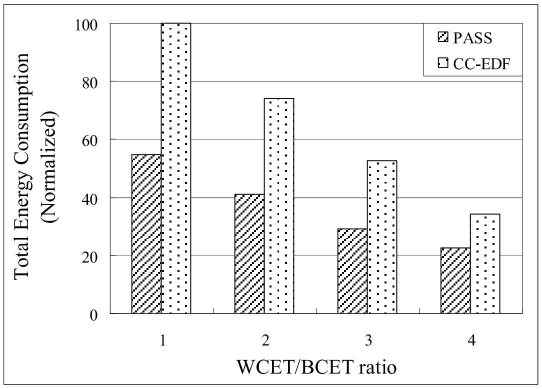
The results reported in the Figure below reveal that all of the four algorithms deliver a 100% guarantee ratio under the Sparse Distribution. The reason for this is that the average request deadline is generally much shorter than the sparse idle threshold, which means that even though IBEC aggressively slows down the processing pace of disk requests, their deadlines can still be satisfied. When we applied a Cluster Distribution pattern, the performance of the four algorithms goes down when the parameters of the Cluster Distribution increase. This is because there were a large number of requests arrived during a burst of incoming requests. Consequently, all the algorithms can only guarantee the deadlines of a small part of them.



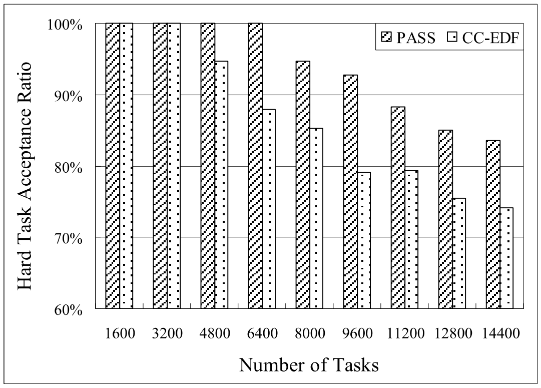
Guarantee ratio vs. workload distributions.

[Liu et al., 2008a] Reducing energy consumption has become a pressing issue in cluster computing systems not only for minimizing electricity cost, but also for improving system reliability. Therefore, it is highly desirable to design energy-efficient scheduling algorithms for applications running on clusters. In this project, we address the problem of non-preemptively scheduling mixed tasks on power-aware clusters. We developed an algorithm called Power Aware Slack Scheduler (PASS) for tasks with different priorities and deadlines. PASS attempts to minimize energy consumption in addition to maximizing the number of tasks completed before their deadlines. To achieve this goal, high-priority tasks are scheduled first in order to meet their deadlines. Moreover, PASS explores slacks into which low-priority tasks can be inserted so that their deadlines can be guaranteed. The dynamic voltage scaling (DVS) technique is used to reduce energy consumption by exploiting available slacks and adjusting appropriate voltage levels accordingly.

The following Figure shows the total energy consumed to execute 1600 tasks. We observe that *PASS* saves up to 60 percent of the energy over *CC-EDF*. The reason that *PASS* can achieve such significant energy savings is because *PASS* creates large integrated slacks by scheduling tasks to the latest possible start time. *CC-EDF* schedules tasks according to the rule of Minimum Completion Time (*MCT*) which always schedules a task to its earliest possible start time. In this way, *EDF* hardly leaves any slack time that may be used by the DVS technique.



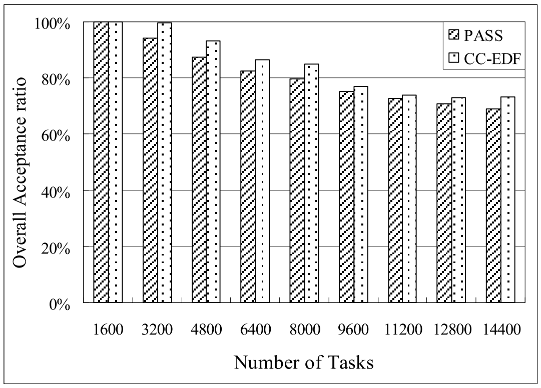
Total energy consumption (Execute 1600 tasks)



Hard task acceptance ratio.

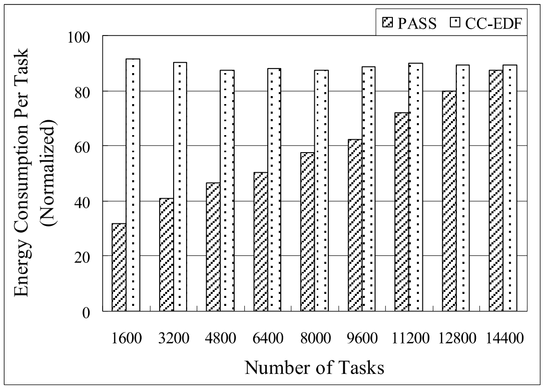
Now we compare the performance of *PASS* against *CC-EDF*. We performed simulations for different task loads in order to examine the performance consistency. With respect to *Hard Task Acceptance Ratio*, from the above Figure we observe that *PASS* yields 10% better performance on average than *CC-EDF*. When the number of tasks is below 6400, *PASS* guarantees that all hard tasks can meet their deadlines. As the number of tasks further increases, *PASS* is still able to schedule most of the hard tasks while *EDF* is no longer able to reach similar performance level. The reason is because *PASS* always schedules hard tasks first, which helps meet hard tasks' deadlines. *CC-EDF* does not consider task priority, which results in a number of un-schedulable hard tasks.

The Figure below shows both algorithms can not schedule all tasks when the number of tasks exceeds 3200. It becomes unfair if we compare two algorithms using the *Total Energy Consumption* metric, since the number of accepted tasks by using *PASS* is different from the number by using *CC-EDF*. Instead, we use the *Energy Consumption per Task* as the metric. In this way, we are able to study the effect brought by the number of tasks on energy consumption.



Overall acceptance ratio.

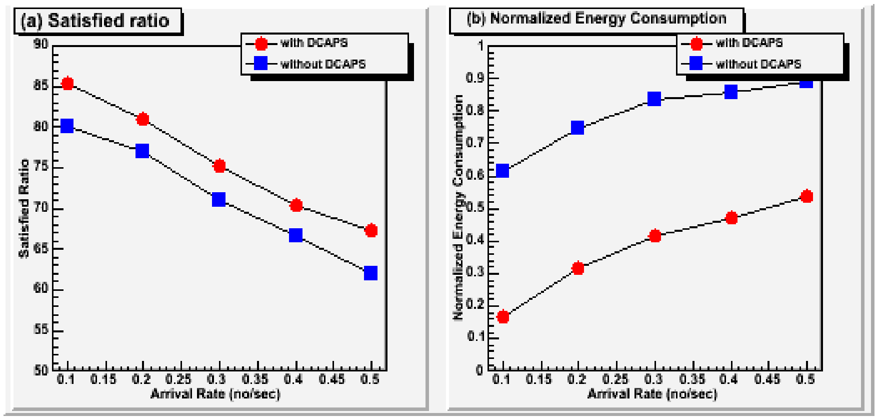
As shown in the following Figure, *PASS* consistently performs better than *CC-EDF* with respect to *Energy Consumption per Task*. It is again because *PASS* decreases the processor speed for each task by utilizing the corresponding slack time. An interesting observation is that as the number of tasks increases, the *Energy Consumption per Task* achieved by *PASS* also increases. Once there are more incoming tasks, less slack times will be available since more tasks need to be scheduled within those slack times.



Energy consumption per task ($\frac{WCET}{BCET} = 2$)

[Nijim et al., 2008a] 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 a variety of parallel disk systems were developed to achieve high performance and energy efficiency, most existing parallel disk systems lack an adaptive way to conserve energy in dynamically changing workload conditions. To solve this problem, we develop an adaptive energy-conserving algorithm, or DCAPS, for parallel disk systems using the dynamic voltage scaling technique that dynamically choose the most appropriate voltage supplies for parallel disks while guaranteeing specified performance (i.e., desired response times) for disk requests.

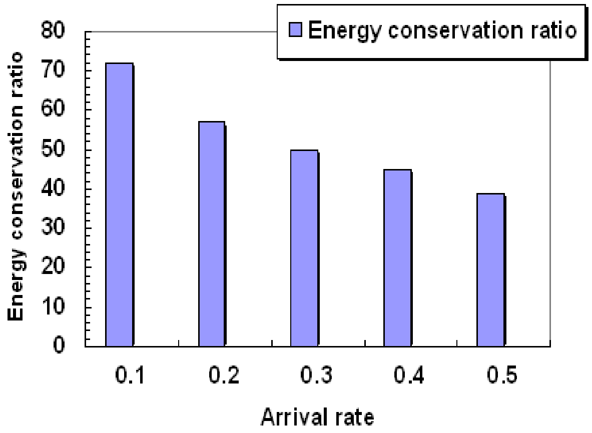
The DCAPS algorithm aims at judiciously lower the parallel disk system voltage using dynamic voltage scaling technique or DVS, thereby reducing the energy consumption experienced by disk requests running on parallel disk systems. The processing algorithm separately repeats the process of controlling the energy by specifying the most appropriate voltage for each disk request. Thus, the algorithm is geared to adaptively choose the most appropriate voltage for stripe units of a disk request while warranting the desired response time of the request.



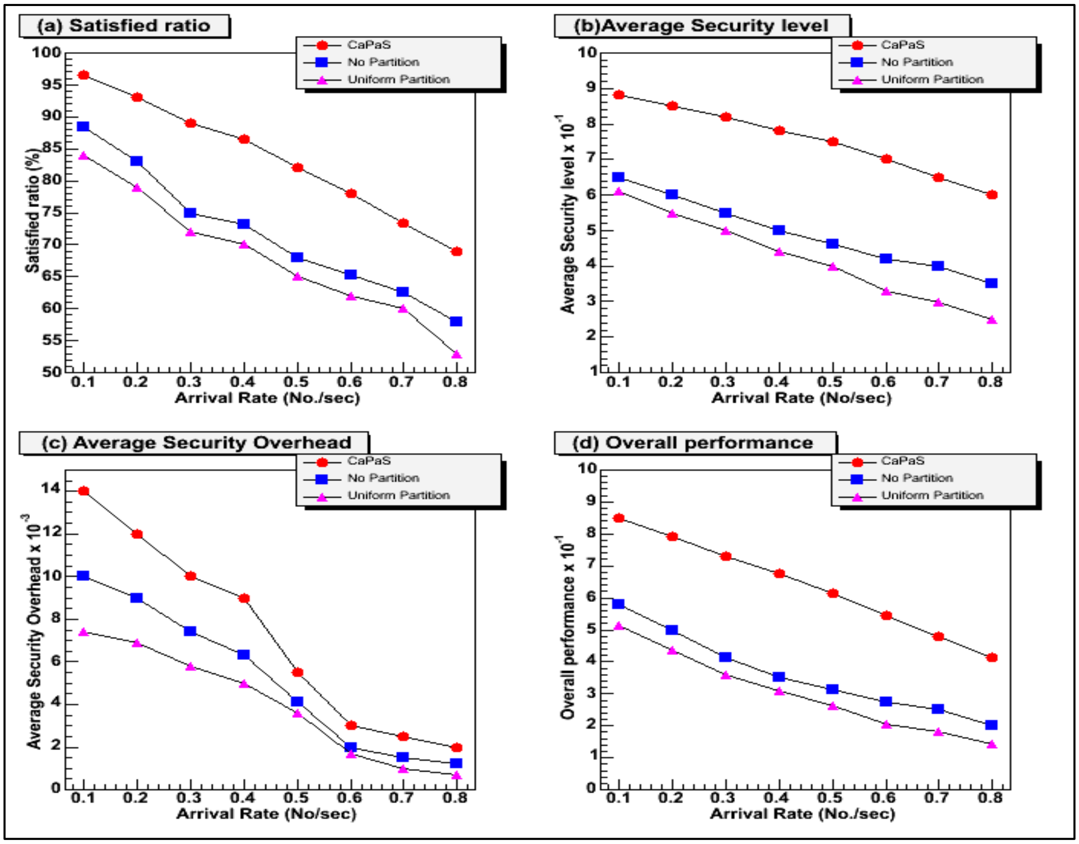
Impact of request arrival rate on satisfied ratio and normalized energy consumption when disk bandwidth is 30MB/sec.

The above two Figures plot the satisfied ratios, normalized energy consumption, and energy conservation ratio of the parallel disk systems with and without DCAPS. Figs 3(a) reveals that the DCAPS scheme yields satisfied ratios that are very close to those of the parallel disk system without employing DCAPS. This is essentially because DCAPS endeavors to save energy consumption at the marginal cost of satisfied ratio. More importantly, Figs. 3(b) and 4 show that DCAPS significantly reduces the energy dissipation in the parallel disk system by up to 71% with an average of 52.6%. The improvement in energy efficiency can be attributed to the fact that DCAPS reduces the disk supply voltages in the parallel disk system while making the best effort to guarantee desired response times of the disk requests. Furthermore, it is observed that as the disk request arrival rate increases, the energy consumption of the both parallel disk systems soars.

The Figure below shows that as the load increases, the energy conservation ratio tends to decrease. This result is not surprising because high arrival rates lead to heavily utilized disks, forcing the DCAPS to boos disk voltages to process larger number of requests within their corresponding desired response times. Increasing number of disk request and scaled-up voltages in turn give rise to the increased energy dissipations in the parallel disk systems.



Impact of request arrival rate on energy conservation ratio.



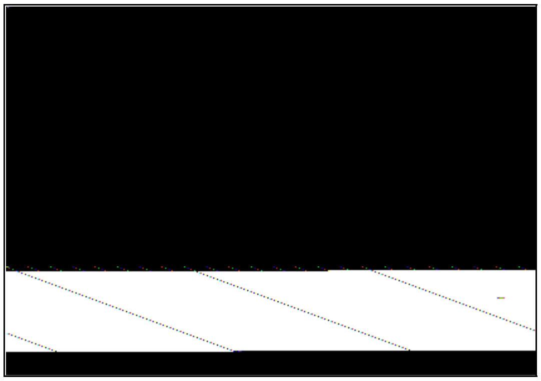
Performance impact of confidentiality services where data size = 300KB, disk bandwidth = 25MB/Sec, and cache size = 40 MB.

[Nijim et al., 2008b] Cluster storage systems have emerged as high-performance and cost-effective storage infrastructures for large-scale data-intensive applications. Although a large number of cluster storage systems have been implemented, the existing cluster storage systems lack a means to optimize quality of security in dynamically changing environments. We solve this problem by developing a security-aware cache management mechanism (or CaPaS for short) for cluster storage systems. CaPaS aims at achieving high security and desired performance for data-intensive applications running on clusters. CaPaS is used in combination with a security control mechanism that can adapt to changing security requirements and workload conditions, thereby providing high quality of security for cluster storage systems. CaPaS is comprised of a cache partitioning scheme, a response-time estimator, and an adaptive security quality controller. These three components help in increasing quality of security of cluster storage systems while allowing disk requests to be finished before their desired response times. To prove the efficiency of CaPaS, we simulate a cluster storage system into which CaPaS, eight cryptographic, and seven integrity services are integrated. Empirical results show that CaPaS significantly improves overall performance over two baseline strategies by up to 73% with an average of 52% (see the above four Figures).

[Liu et al., 2008b]

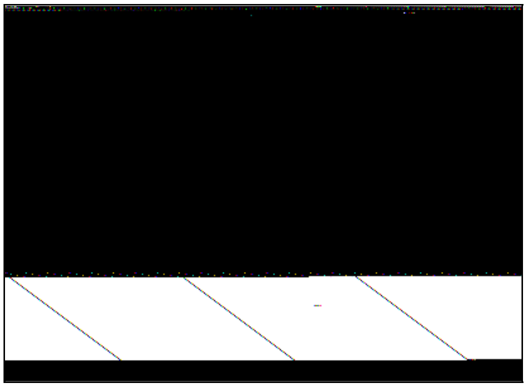
Although data duplications may be able to improve the performance of data-intensive applications on data grids, a large number of data replicas inevitably increase energy dissipation in storage resources on the data grids. In order to implement a data grid with high energy efficiency, we address in this study the issue of energy-efficient scheduling for data grids supporting real-time and data-intensive applications. Taking into account both data locations and application properties, we design a novel Distributed Energy-Efficient Scheduler (or DEES for short) that aims to seamlessly integrate the process of scheduling tasks with data placement strategies to provide energy savings. DEES is distributed in the essence - it can successfully schedule tasks and save energy without knowledge of a complete grid state. DEES encompasses three main components: energy-aware ranking, performance-aware scheduling, and energy-aware dispatching. By reducing the amount of data replications and task transfers, DEES effectively saves energy.

The following Figure shows the performance of *DEES* using different (*ε*, *μ*) value pairs with respect to *Guarantee Ratio*. It is observed that *DEES (2, 1)* gives the best performance. This is because *DEES (2, 1)* takes both goals of meeting deadline and saving energy into account, and put more weight onto the deadline meeting part. Neighbors that can schedule more tasks are given preference. We conclude it is better to give preference to neighbors that can schedule more tasks while consuming satisfactory amount of energy.



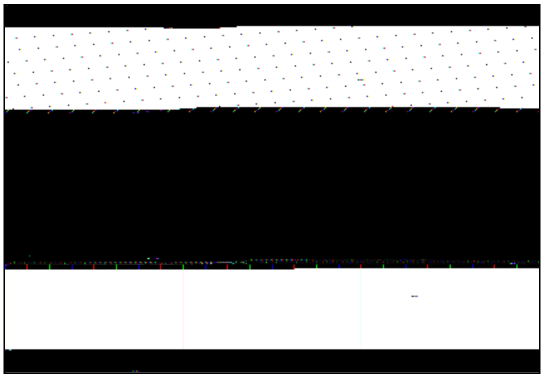
Guarantee Ratio by ranking coefficients

With respect to *Normalized Average Energy Consumption*, as shown in the following Figure, we observe that *DEES (2, 1)* consumes the least amount of energy while *DEES (0, 1)* consumes the most. *DEES (2, 1)* considers both energy consumption and deadline constraints when dispatching tasks to neighbors. Doing so can reduce the energy cost per task. On the other hand, *DEES (0, 1)* schedules fewer tasks since it only cares about energy consumption when dispatching tasks. Moreover, given that more tasks miss their deadlines at each site; additional data replications may be needed. Therefore it relatively consumes more energy to replicate data and transfer the tasks.



Normalized Average Energy Consumption by ranking coefficients

In this experiment set, we compared the performance of *DEES* with *Close-to-Files* and *Performance-driven* algorithms under different task loads. From the Figure below, we observe that *DEES* yields better performance than *Close-to-Files* and achieves the same performance level as the *Performance-driven* algorithm does. The *Performance-driven* algorithm always schedules a task to a globally best resource that gives the best performance. Since it only focuses on performance but not other factors such as data locality, it yields very good performance with respect to *Guarantee Ratio*. But the fact that DEES gives similar performance as the *Performance-driven* algorithm is importance. Thus, *DEES* not only reduces energy consumption, but it does so without degrading the *Guarantee Ratio*. One reason is because *DEES* always schedules tasks with shorter deadlines first. The final criteria for judging whether a task can be scheduled are the task deadlines. Scheduling those tasks with shorter deadlines first makes more tasks schedulable. Moreover, *DEES* is fully distributed, which is expected to improve the performance when compared to a centralized algorithm, such as the Performance-driven algorithm, especially when the task load is heavy. Given that *DESS* is fully distributed, while *Close-to-Files* and *Performance-driven* algorithms need knowledge of a complete state of the grid, the results make *DEES* more favorable.



Guarantee Ratio by task loads

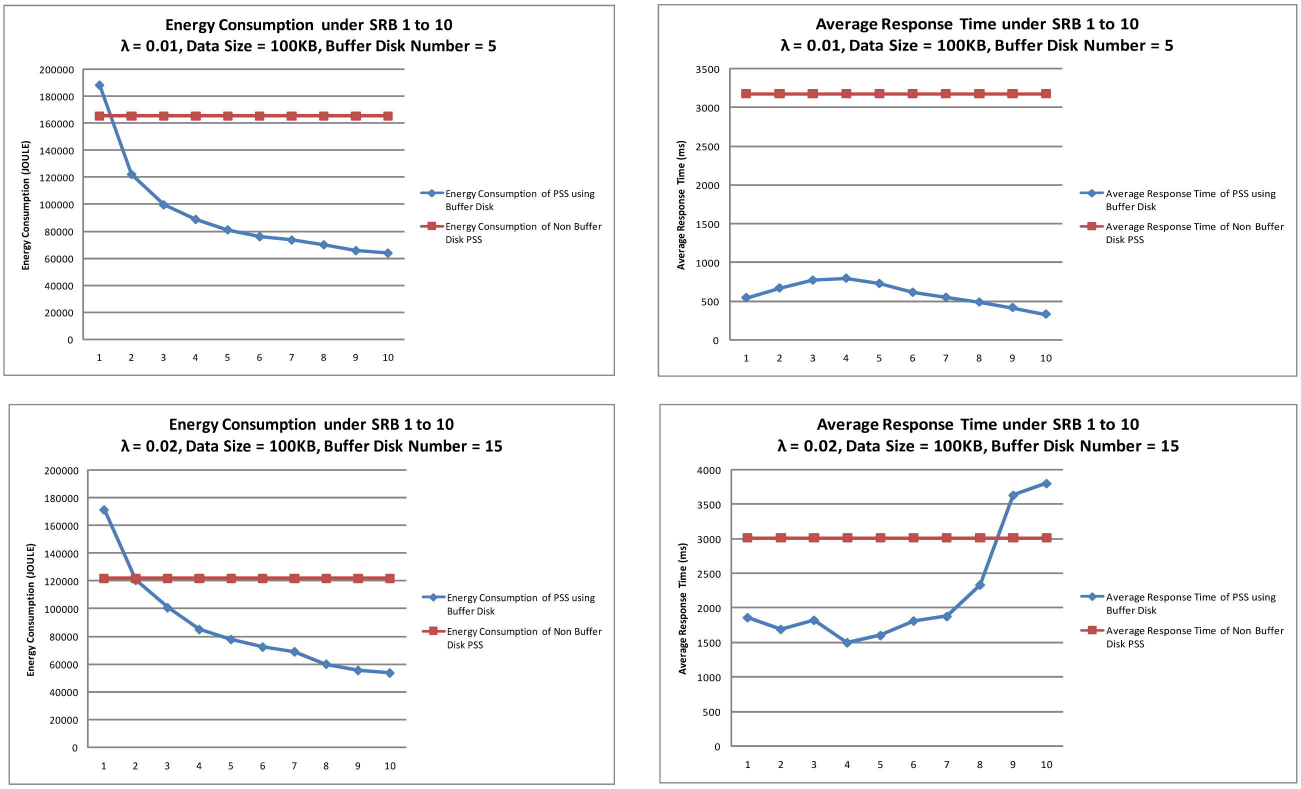
With respect to *Normalized Average Energy Consumption*, as shown in the Figure below, we see that *DEES* consumes much less energy per task than *Close-to-Files* does. On average *DEES* saves over 35% of energy consumed when compared to the other algorithms. This is because *DEES* considers the energy consumed to transfer both tasks and data during dispatching. Moreover, *DEES* groups tasks according to their data accesses and processes tasks on a group basis. Doing so limits the number of data replicas. This is because whenever data is replicated to a remote site, *DEES* always maximizes utilization of the data replicated by scheduling as many tasks as possible to that remote site. On the other hand, *Close-to-Files* makes dispatching decisions on a single task basis, which may result in unnecessary data replications. Furthermore, since *DEES* schedules more tasks than *Close-to-Files* does, the energy cost per task is expected to be less. The *Performance-driven* algorithm consumes the most amount of energy due to the fact that it is a greedy algorithm that always schedules a task to a resource giving the best performance, regardless of how much data are needed to be replicated and transferred.

[redacted]

Normalized Average Energy Consumption by task loads

[Ruan et al., 2009] In the past decades, parallel I/O systems have been used widely to support scientific and commercial applications. New data centers today employ huge quantities of I/O systems, which consume a large amount of energy. Most large-scale I/O systems have an array of hard disks working in parallel to meet performance requirements. Traditional energy conservation techniques attempt to place disks into low-power states when possible. In this work we propose a novel strategy, which aims to significantly conserve energy while reducing average I/O response times. This goal is achieved by making use of buffer disks in parallel I/O systems to accumulate small writes to form a log, which can be transferred to data disks in a batch way. We develop an algorithm - dynamic request allocation algorithm for writes or DARAW - to energy efficiently allocate and schedule write requests in a parallel I/O system. DARAW is able to improve parallel I/O energy efficiency by the virtue of leveraging buffer disks to serve a majority of incoming write requests, thereby keeping data disks in low-power state for longer period times. Buffered requests are then written to data disks at a pre-determined time. Experimental results show that DARAW can significantly reduce energy dissipation in parallel I/O systems without adverse impacts on I/O performance.

The Figure below shows the energy consumption and average response time of a parallel disk system with DARAW and the same disk system without DARAW. The results indicate that when we increase SRB, more energy can be saved. The results were expected since when the SRB grows, the system can write more requests into data disks with reduced number of power state transitions. However, we also observe that when the SRB equals to one, the energy consumption is even greater than the disk system without DARAW. This interesting tend can be explained as follows. Our parallel disk system has a buffer-disk layer that also consumes energy. If there is insufficient number of requests written into a data disk when a power-state transition occurs, energy conserved cannot offset energy overhead introduced by the buffer disk. When we did the experiment with a trace generated by increasing values of λ, we observe that energy consumptions in both the non-DARAW parallel disk system and the system with DARAW decrease.

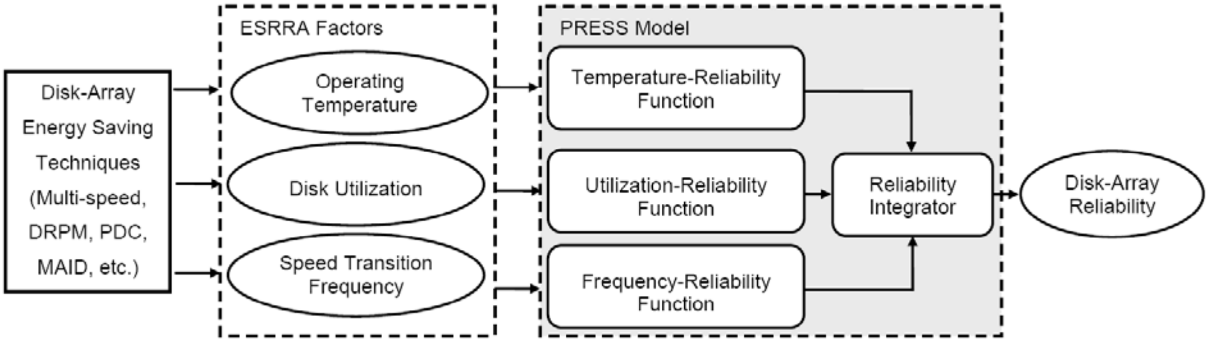


IBM 40GNX Travelstar. Energy Consumption and Average Response Time Compare

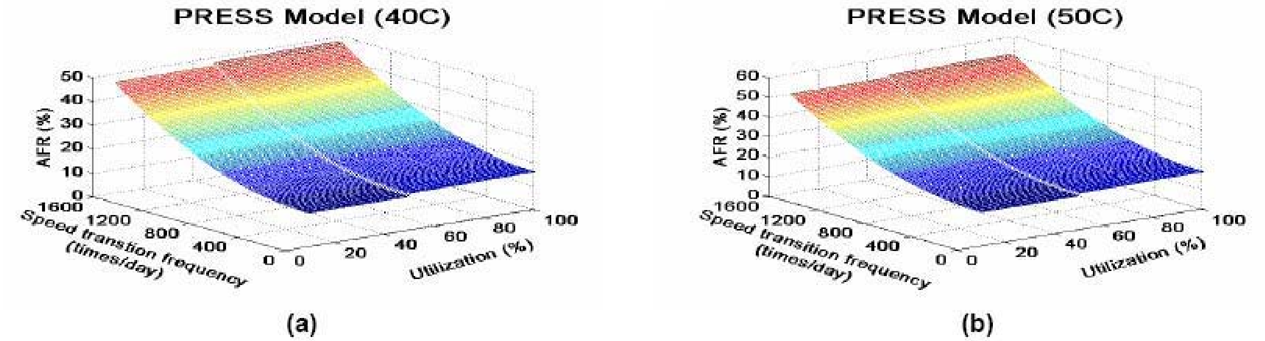
Note that all the traces have the same number of disk requests. This implies the fact that when λ is high, all requests are arriving at the system within a shorter period of time, making all the disks stay in the active state for a shortened time interval. This is the reason behind the result that energy consumption of the system with DARAW when λ is set to 0.02 is slightly smaller than that of the system when λ is 0.01. However, the power consumption of the non-DARAW disk system is significantly smaller when λ is 0.01 as compared to λ = 0.02. Once the arrival rate goes up, each data disk in the non-DARAW system has greater probability to receive a request when it is working. Thus, the number of power-state transitions can be noticeably reduced. When λ is set to 0.02, there is less of an opportunity to simultaneously save energy and satisfy response times. When we increase the number of buffer disks from 5 to 20, DARAW can conserve energy while guaranteeing reasonably short response times. An appealing result shown in the above Figure is that compared with the parallel I/O system without DARAW, our approach not only achieves significant energy savings, but also reduces response times. In DARAW, the response time is the time when a request is written in to a data or buffer disk. Since buffer disks can serve coming requests when data disks are sleeping, the response time can be noticeably shortened.

Our results show that DARAW works well for parallel I/O systems with both high performance disks and mobile disks. DARAW achieves promising results when the arrival rate is low. When the request arrival rate rises, we can either use high-performance hard drives or add more buffer disks to boost I/O performance. If the arrival rate is high, all data disks are busy serving requests, leaving no opportunity to save energy. As the SRB parameter grows, DARAW is given a greater window of opportunity to conserve energy. However, if the SRB is too large, it may cause a "traffic jam" inside the parallel I/O system with buffer disks.

[Xie and Sun, 2008a] Mainstream energy conservation schemes for disk arrays inherently affect the reliability of disks. A thorough understanding of the relationship between energy saving techniques and disk reliability is still an open problem, which prevents effective design of new energy saving techniques and application of existing approaches in reliability-critical environments. As one step towards solving this problem, we investigated an empirical reliability model, called Predictor of Reliability for Energy Saving Schemes (PRESS). The architecture of the PRESS model is given below:


Disk-Array Energy Saving Techniques (Multi-speed, DRPM, PDC, MAID, etc.)
ESRRA Factors
Operating Temperature
Disk Utilization
Speed Transition Frequency
PRESS Model
Temperature-Reliability Function
Utilization-Reliability Function
Frequency-Reliability Function
Reliability Integrator
Disk-Array Reliability

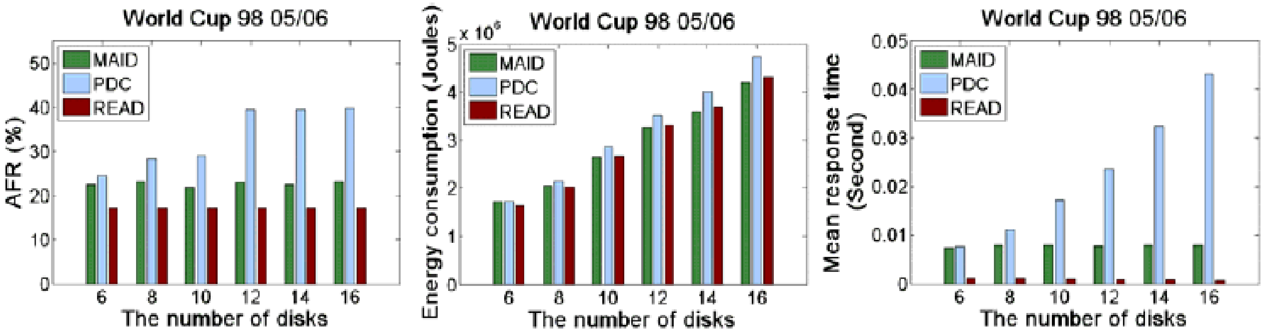
Fed by three energy-saving-related reliability-affecting factors, operating temperature, utilization, and disk speed transition frequency, PRESS estimates the reliability of entire disk array. In what follows, we present two 3-dimennsional figures to represent the PRESS model at operating temperature 40 C (Figure 5a) and 50 C (Figure 5b), respectively.



(a)

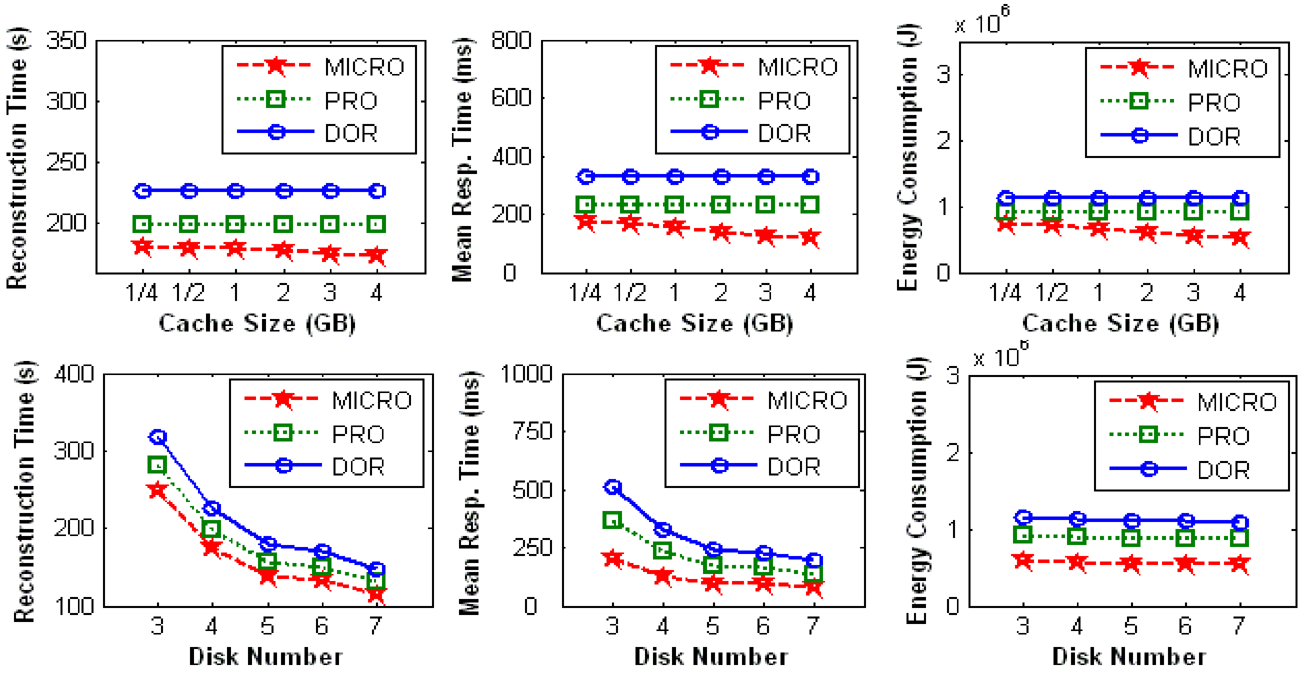
(b)

Further, we developed a new energy saving strategy with reliability awareness called Reliability and Energy Aware Distribution (READ) is developed in the light of the insights provided by PRESS. Experimental results demonstrate that compared with existing energy saving schemes, MAID and PDC, READ consistently performs better in performance and reliability while achieving a comparable level of energy consumption.

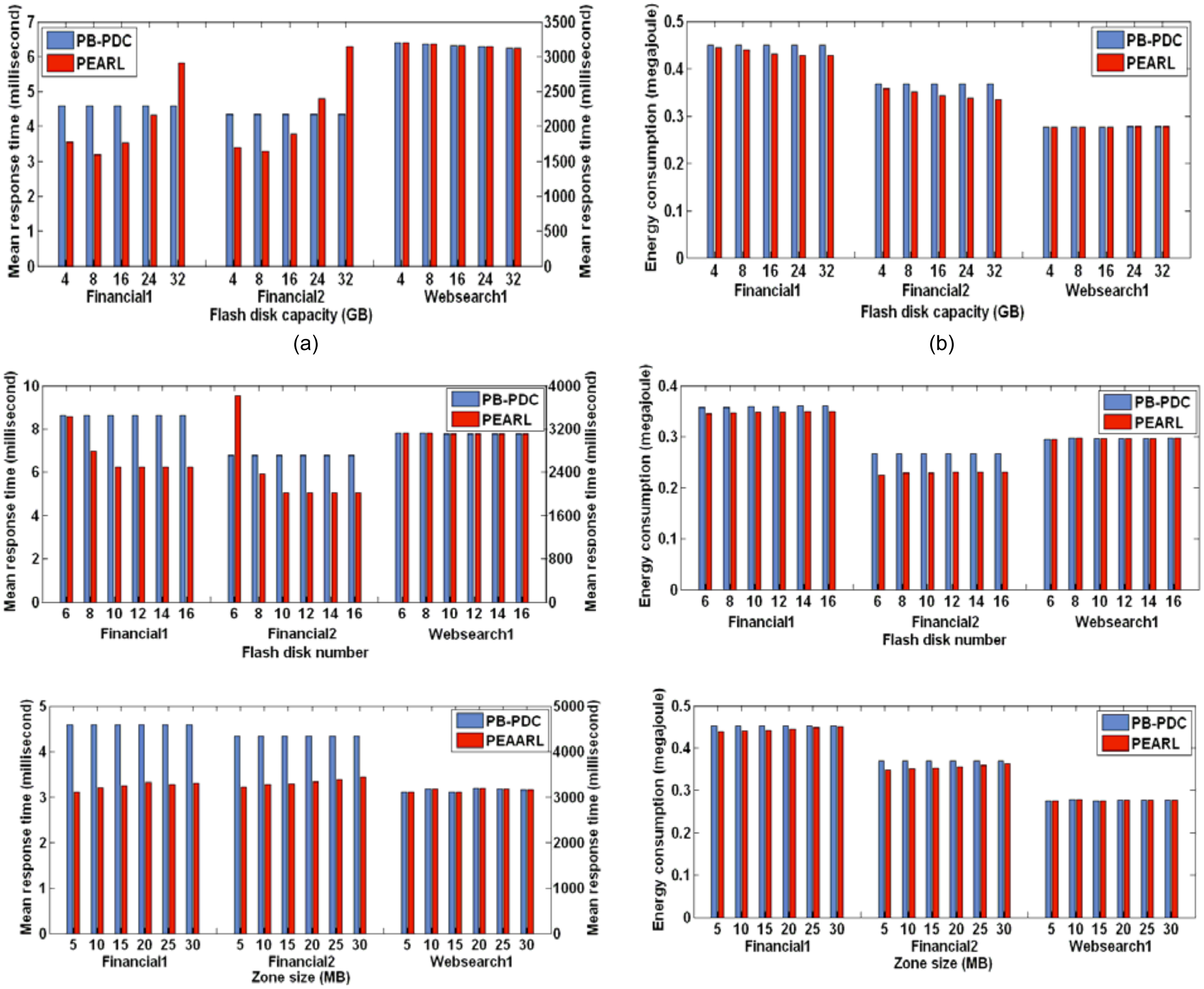


(a) Reliability (b) Energy consumption (c) Mean response time

[Xie and Wang, 2008] High performance, highly reliable, and energy-efficient storage systems are essential for mobile data-intensive applications such as remote surgery and mobile data center. Compared with conventional stationary storage systems, mobile disk-array-based storage systems are more prone to disk failures due to their severe application environments. Further, they have very limited power supply. Therefore, data reconstruction algorithms, which are executed in the presence of disk failure, for mobile storage systems must be performance-driven, reliability-aware, and energy-efficient. In this project we developed a novel reconstruction strategy, called multi-level caching-based reconstruction optimization (MICRO), which can be applied to RAID-structured mobile storage systems to noticeably shorten reconstruction times and user response times while saving energy. MICRO collaboratively utilizes storage cache and disk array controller cache to diminish the number of physical disk accesses caused by reconstruction. Experimental results demonstrate that compared with two representative algorithms DOR and PRO, MICRO reduces reconstruction times on average 20.22% and 9.34%, while saving energy no less than 30.4% and 13%, respectively.



[Xie and Sun, 2008b] Contemporary disk arrays consist purely of hard disk drives, which normally provide huge storage capacities with low-cost and high-throughput for data-intensive applications. Nevertheless, they have some inherent disadvantages such as long access latencies, fragile physical characteristics, and energy-inefficiency due to their build-in mechanical and electronic mechanisms. Flash-memory based solid state disks, on the other hand, although currently more expensive and inadequate in write cycles, offer much faster read accesses and are much more robust and energy efficient. To combine the complementary merits of hard disks and flash disks, in this study we propose a hybrid disk array architecture named HIT (hybrid disk storage) for data-intensive applications. Next, a dynamic data redistribution strategy called PEARL (performance, energy, and reliability balanced), which can periodically redistribute data between flash disks and hard disks to adapt to the changing data access patterns, is developed on top of the HIT architecture. Comprehensive simulations using real-life block-level traces demonstrate that compared with existing data placement techniques, PEARL exhibits its strength in both performance and energy consumption without impairing flash disk reliability.



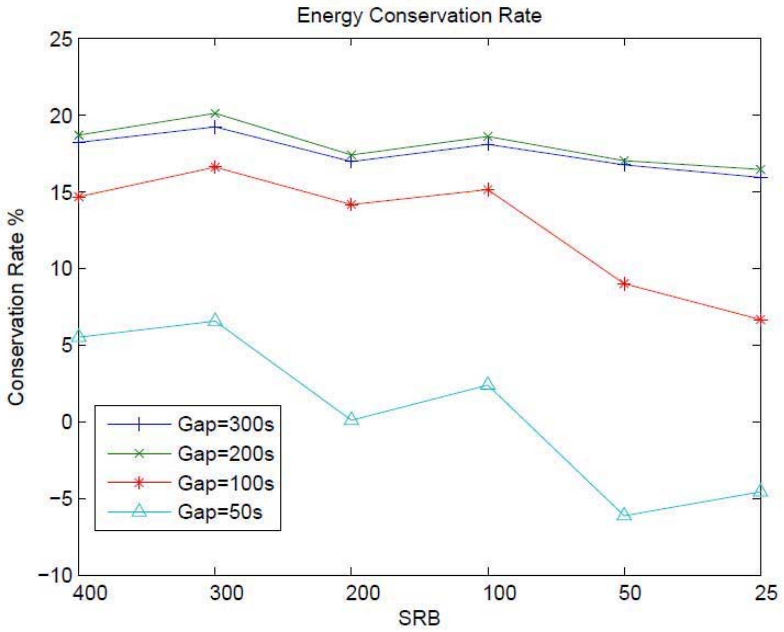
(a)

(b)

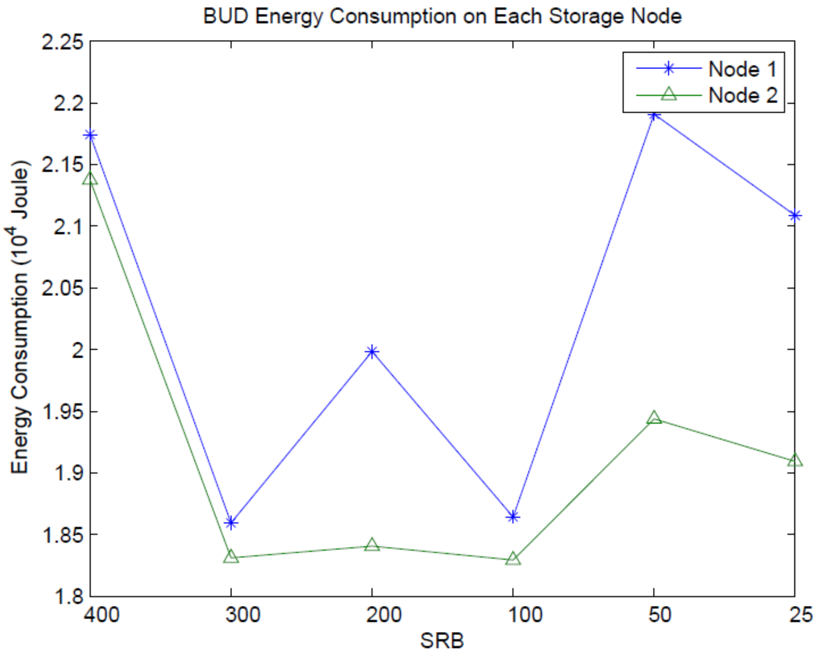
ECOS: An Energy-Efficient Cluster Storage System [Ruan et al. 2009a]

The disks we apply in our prototype are different for the purpose of testing performance of

DARAW in different devices. The traces we use are synthetic traces. The arrival rates are generated by exponential distribution. To reflect the real world cases, the traces contain burstnesses and idle time gap inside. Each burstness contains a group of requests whose arrival rates based on exponential distribution. The most appropriate time for buffer disk to dump data to data disk is during idle time gaps. Hence, we will test traces with different idle time gaps and analyse the results.



In the above figure, up to more than 20% energy could be conserved when idle time gap between each request burstness is 300 seconds or 200 seconds. When the idle time between each burstness is as short as 50 seconds, the energy conservation rate is not very obvious by using DARAW strategy to buffer data on buffer disks from energy conservation perspective. In the figure, the Sum of Request in Buffer, or SRB, is another important parameter in our experiment. According to DARAW strategy, if the number of bufferred requests which target at one same data disk equals SRB, then the targetted data disk spins up and those requests will be dumped to it. The data disk will spin down when there is no request writting on it. Basically, the larger SRB we set up in DARAW, the more energy we can conserve in the system. However, trace features also can affect the energy conservation rate. The most appropriate time of dumping data to data disks is during idle time gaps. DARAW only dumps data to data disks when SRB requirements resatisfied. If there are too many dumping operations happen during burstness, energy conservation rate will be reduced.



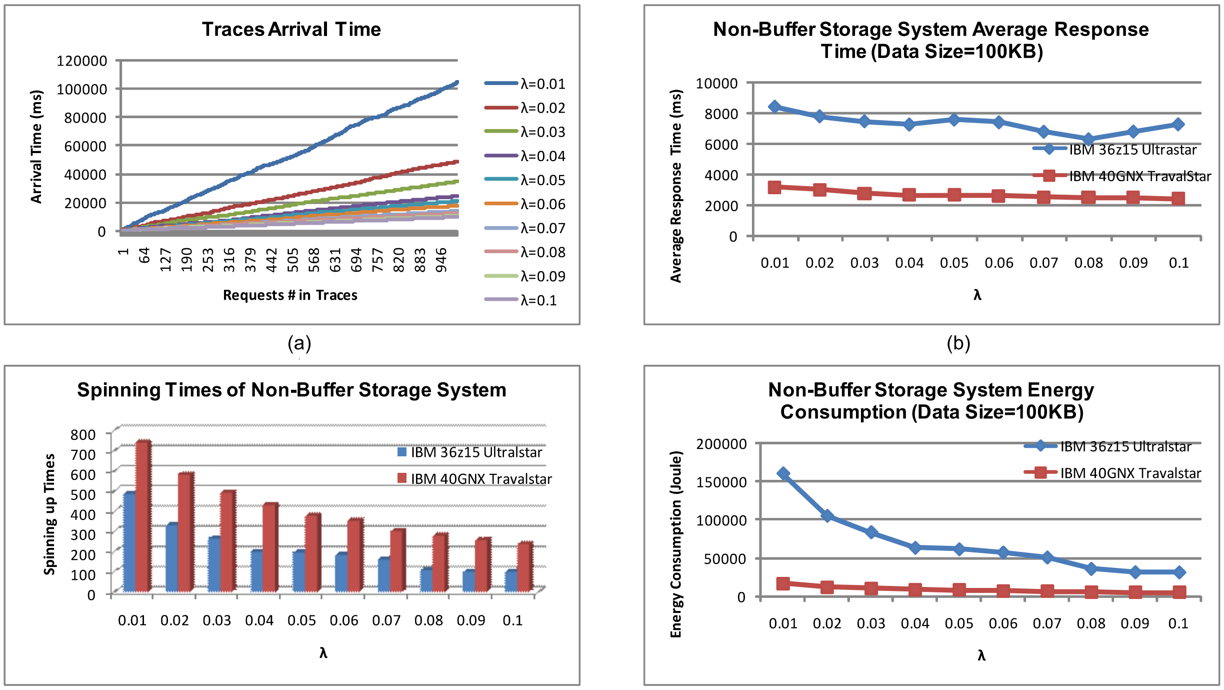
Because idle time gap is low which means workload is high, so dumping data operation is more likely happen during burstnesses (see the above figure). Node 2 is faster than node 1, so the dumping operations will not extend the processing time of buffer disk. In node 1, when dumping operations happen during burstness, since the disks speed in node 1 is not as fast as node 2, the buffer disk needs more time to finish dumping data during burstness. Hence, node 1 consumes more energy in this experiment.

Performance Evaluation of Energy-Efficient Parallel I/O Systems with Write Buffer Disks [Ruan et al. 2009b] [Ruan et al. 2009c]

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.



(a)

(b)

(c)

(d)

The above figure plots energy efficiency and performance of the baseline algorithm applied to a traditional parallel I/O system without buffer disks. Results plotted in Fig. (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.

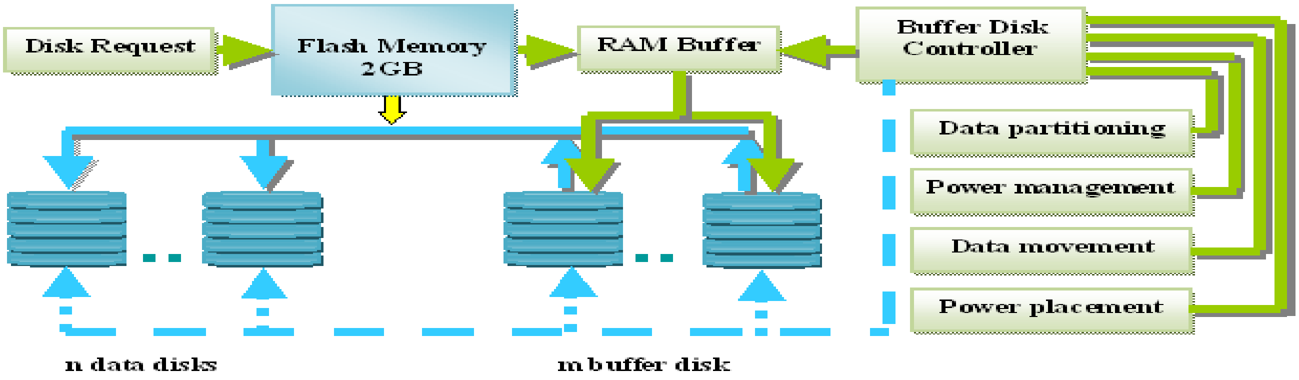
An interesting counterintuitive observation drawn from Fig. (b) is that with respect to the baseline algorithm, the average response time of the high-performance disk (IBM 36z16 Ultrastar) is noticeably longer than that of the IBM 40GNX Travalstar - a low-performance disk. The rationale behind this observation 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 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. (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. (d) depicts the energy consumption trend for the IBM Ultrastar and Travalstar disks. In what follows, we describe two important observations. First, Fig. (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. c). It is worth noting that in each experiment, we fix the total number of requests (e.g, 1000).

HyBUD: An Energy-Efficient Architecture for Hybrid Parallel Disk Systems [Nijim et al. 2009]

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



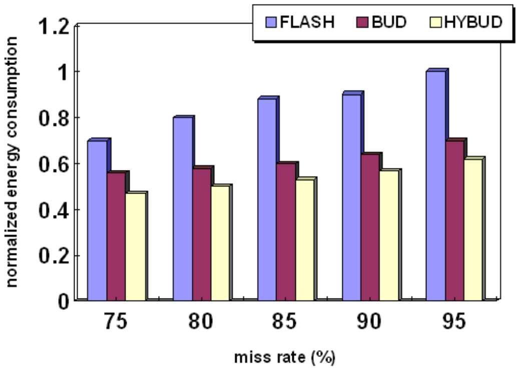
HYBUD: The architecture for cost-effective hybrid parallel disk systems with buffer disks. HYBUD contains a 2-GB flash drive, m buffer disks, n data disks, and an energy-aware flash drive/buffer disk controller.

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

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

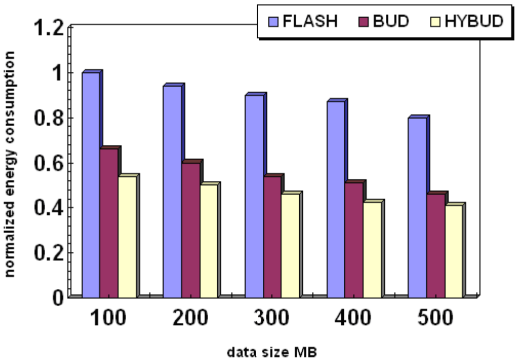
To improve I/O performance of buffer disks handing write requests, we chose to use a log file system that allows data to be written sequentially buffered in buffer disks to minimize disk seek times and rotational delays. We developed a buffer disk manager that is responsible for the following activities.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

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 compared 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. 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.



Energy consumption versus miss ratio

The above figure 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 HYBUD strategy consumes less energy than the other two alternatives strategy. We will discuss each strategy separately. When the disk request is submitted to the flash memory and if there is a miss, then the total energy cost will be the energy cost of read and the energy cost of writing the request into the flash under the assumption that this request will be frequently used in the future, and the cost of keep waking up the corresponding data disk every time the read request is made which leads to a huge energy consumption. For the BUD strategy, it consumes less energy than using only the flash memory strategy. This can be explained by the fact that the BUD strategy when miss occurs, it clusters read requests together if the disk is in sleeping mode. As a result, it provides a long disk idle times.



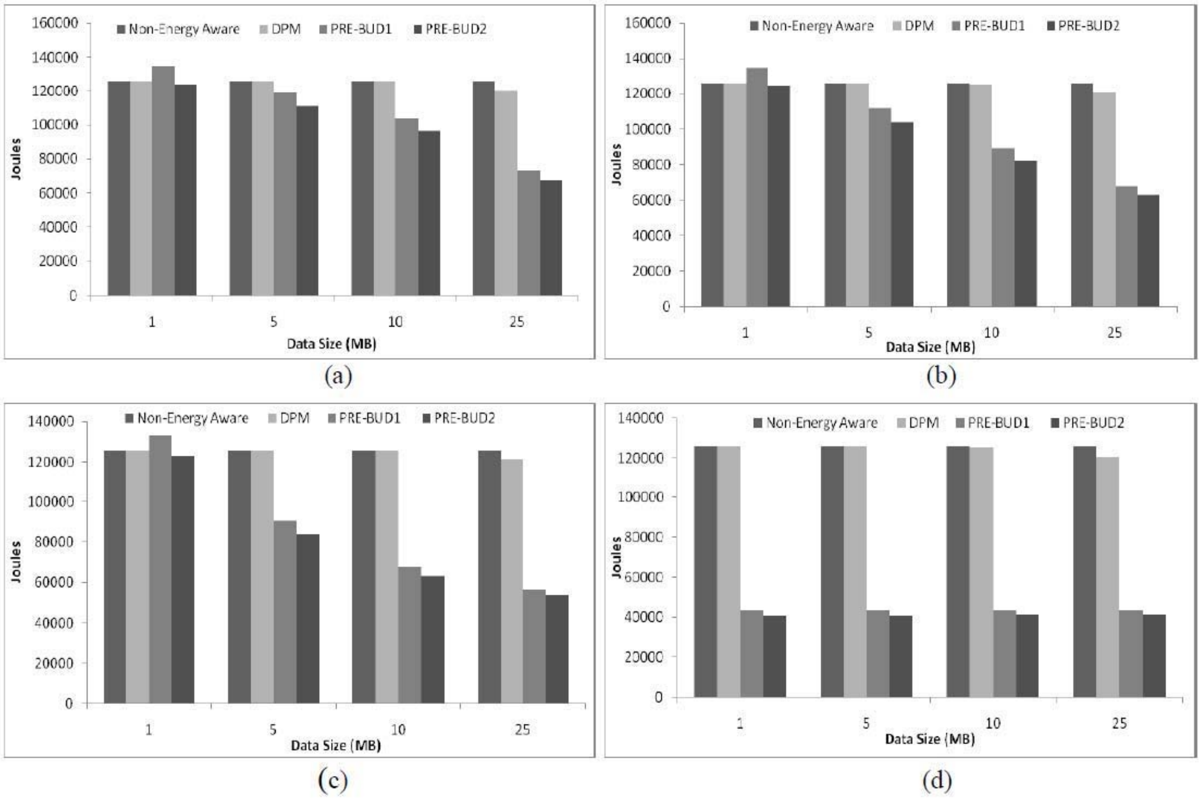
Energy consumption versus data size MB

Finally the HYBUD strategy consumes the least energy. When the read request is submitted to the flash memory, they read request will be served immediately if the data disk is in active mode, otherwise, the read request will be written to the flash assuming that these requests will be frequently used. When the flash is full, the dirty data will be flushed to the buffer disks and all the miss requests will be clustered together, which leads to less energy consumption.

In this experiment we compared the three strategies in term of the size of data block. The above figure 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. In case of small data size, flash memory is no longer able to save energy, because the flash memory keep waking up the disk, which results in huge energy consumption. The HYBUD strategy can save up to 51% over the flash memory and 22% over BUD.

Energy-Aware Prefetching for Parallel Disk Systems [Manzanares et al. 2009]

For our experimental results we implemented a parallel disk simulator in JAVA. The first set of experiments we conducted varied the hit rate and the data size of the requests. The hit rate in these experiments is defined as the percentage of all the requests that can be served by the buffer disk and the data size is defined as the data size of each request. We generated random disk requests and varied the inter-arrival delay of the requests. The inter-arrival rate must be fairly low to produce energy savings or disks will never be placed in the sleep state. If the inter-arrival rate is high all disks must be active to serve the requests. The results of the first set of experiments are summarized in the Figure below.

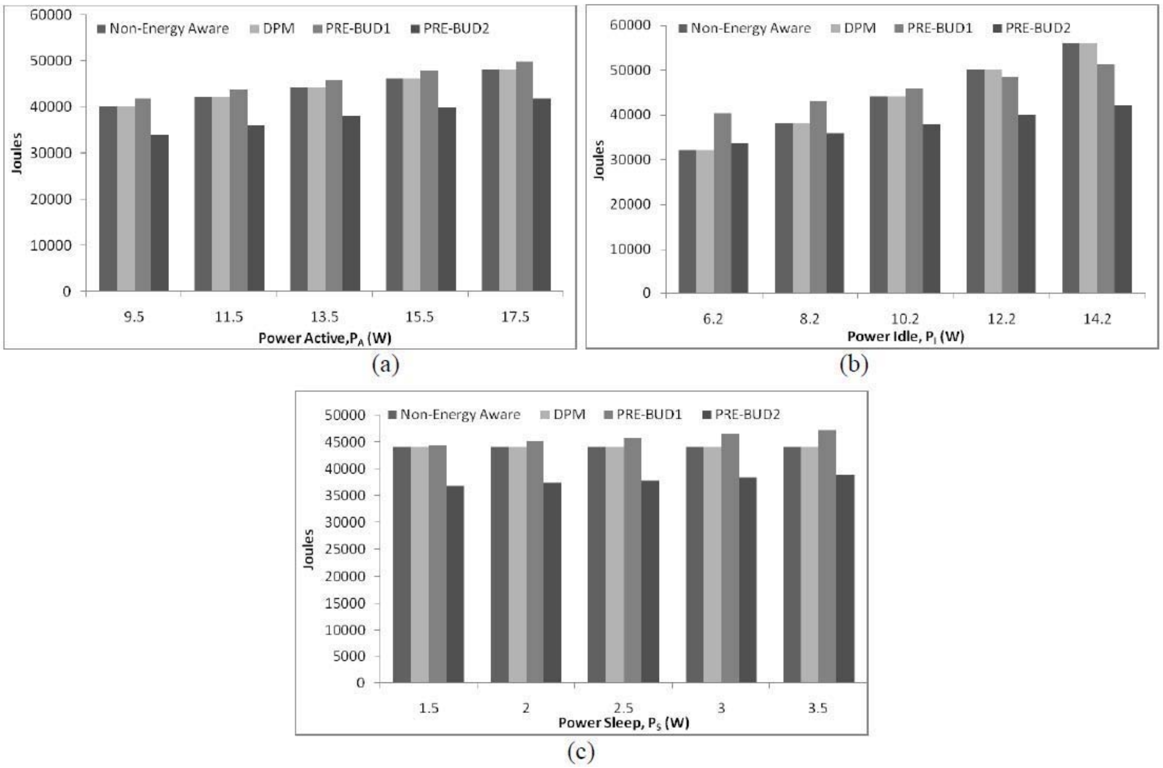


(a) (b)

(c) (d)

Total Energy Consumption of Disk System while Data Size is varied for four different values of hit rate: (a) 85 %, (b) 90 %, (c) 95 %, and (d) 100 % hit rate.

There are two main observations we can draw from this figure, one being that as data size increases energy savings increases, and second, as the hit rate is increased energy savings increases. As the data size increases the time to serve the request increases. If multiple requests can be served from the buffer disk than the data disks have a greater opportunity to transition to the sleep state. Similarly as the hit rate increases the buffer disk serves a greater number of consecutive hits allowing data disks to sit idle for longer periods of time. The goal of our energy-efficient prefetcher is to increase the number and length of idle periods to allow a data disk to transition to the sleep state. This can be achieved by increasing the hit rate or increasing the data size of requests. This leads us to believe that many web and multimedia applications would be suitable for our energy saving techniques.



(a)

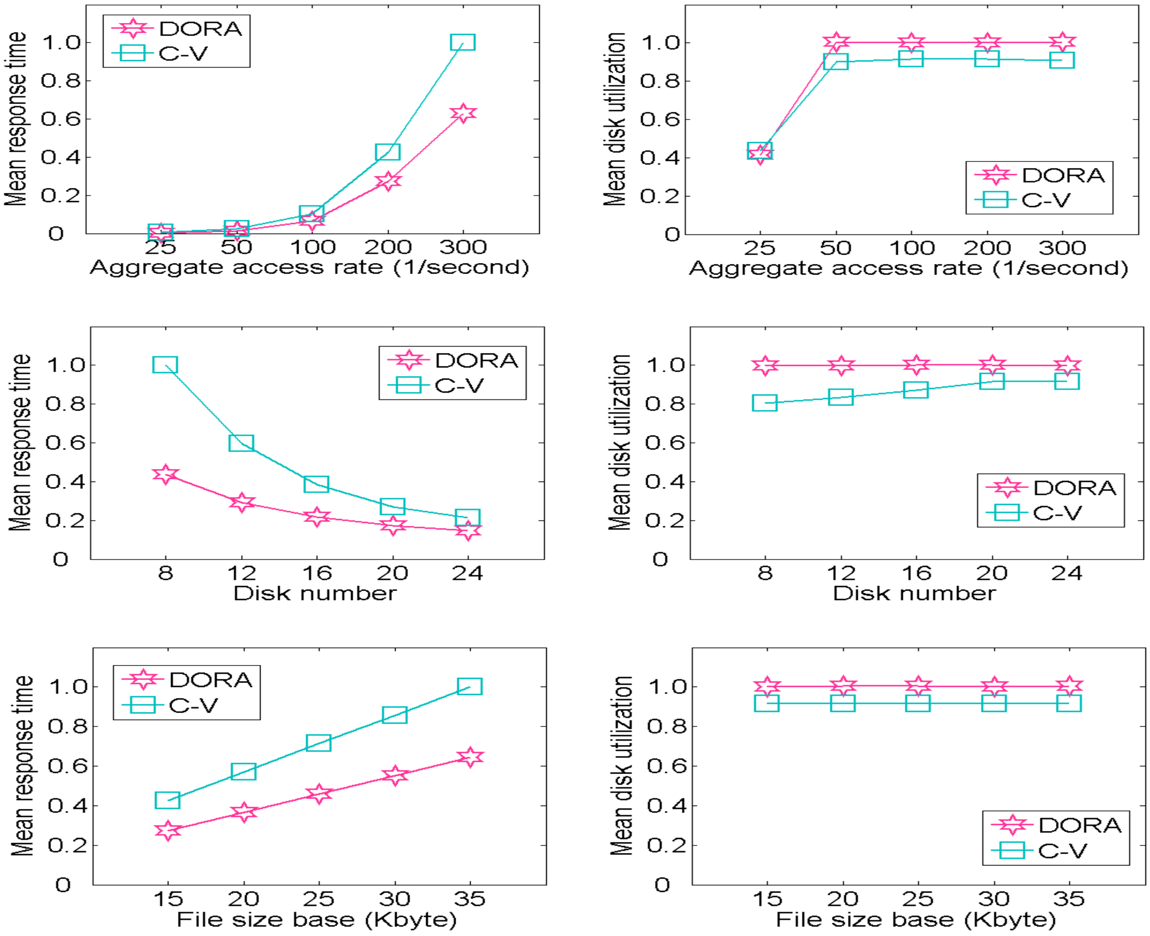
(b)

(c)

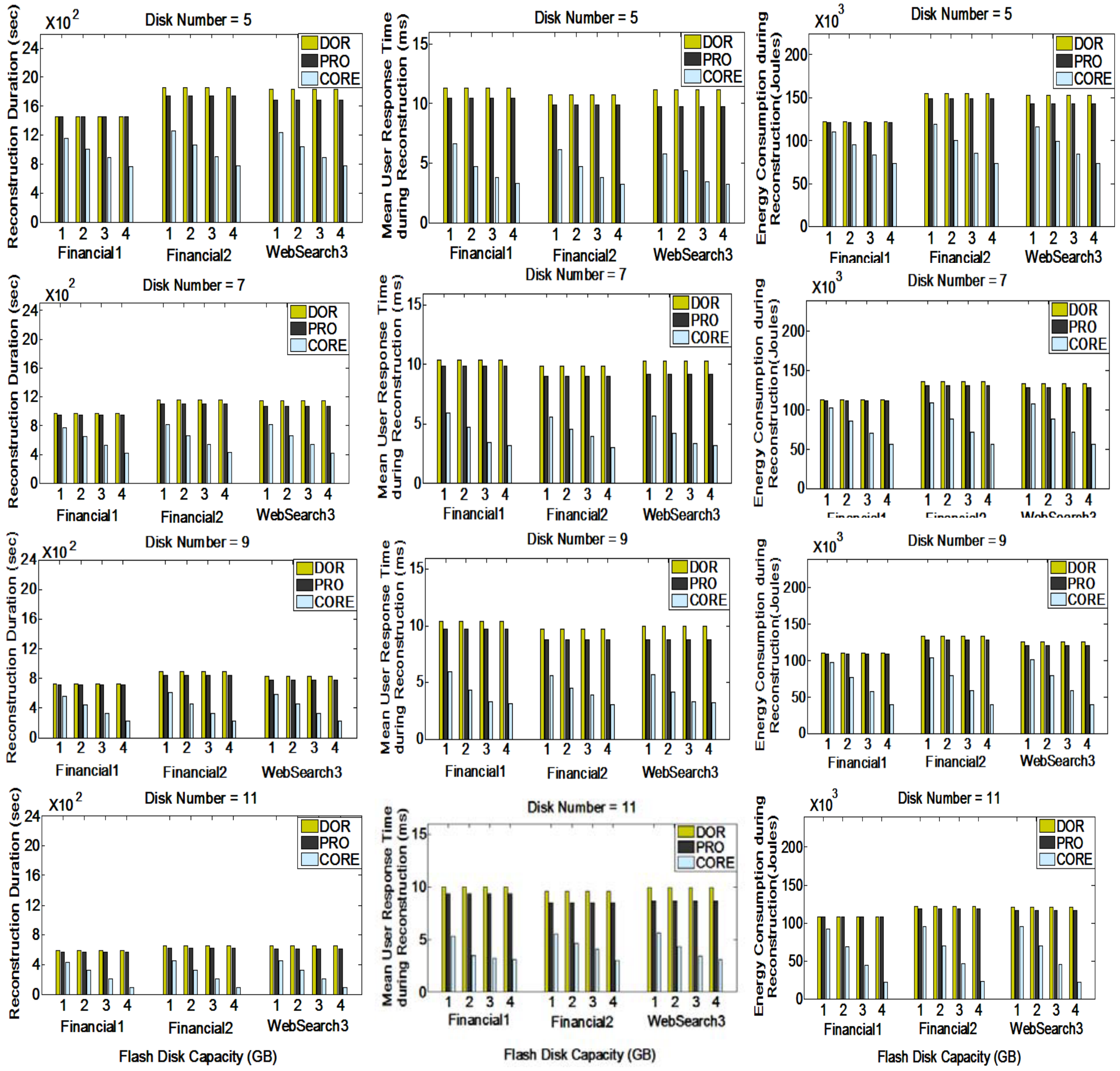
Total Energy consumption for various values of the following disk parameters: (a) power active, (b) power idle, and (c) power standby

The second set of experiments conducted focuses on the impact that varying disk power parameters has on the energy savings. We varied the power characteristics of our simulated IBM36Z15 disk. For each figure we only vary one disk energy parameter. The number of disks was fixed at four and the data size is 25MB. From the above figure we realize that lowering the Power Active, which is the energy consumed while the disk is in the active state, will decrease the energy consumption for all the strategies we compared. Lowering Power Active also impacts the relative energy savings that the PRE-BUD strategies are able to produce. If Power Active is 9.5W PRE-BUD2 saves 15.1 % energy over DPM. If it is increased to 17.5W PRE-BUD2 only saves 13% energy over DPM. Fig. 4 (a) is similar to Fig. 4 (b) but now we see that Power Idle has a greater impact on energy savings as compared to Power Active. If Power Idle, the energy consumed while the disk is idle, is very low PRE-BUD 2 has a negative impact, but if it's increased to 14.2 W PRE-BUD 2 now saves 25 % of energy as compared to DPM. The last set of experiments varied the Power Sleep parameter, which represents the energy consumed while the disk is in the sleep state, also has significant impact on PRE-BUD strategies. The percentage change in energy savings starts at 16.3% and drops to 11.7% with increasing Power Sleep. The results illustrated in Fig. 4 indicate that parallel disks with low active power, high idle power, and low standby power can produce the best energy-savings benefit. This is because PRE-BUD allows disks to be spun down to the standby state during times they would be idle using DPM. The greater the discrepancy between idle and standby power, the more beneficial PRE-BUD becomes.

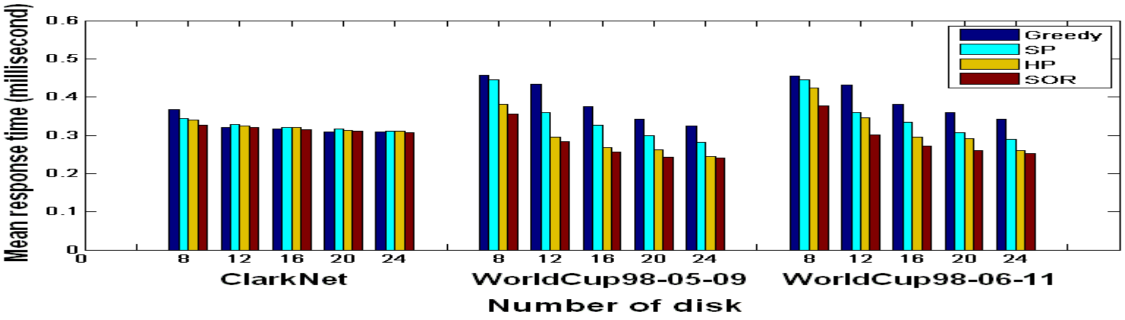
[Tjioe, Widjaja, Lee, and Xie, 2009] In a completely dynamic environment where a sub-set of files are extremely popular and receive a dominant percentage of user requests, a dynamic file assignment algorithm may no longer be helpful. The reason is that no matter where it places these hot files the load imbalance across the disks cannot be solved. In this situation, file replication techniques can be employed to make replicas for these popular files and to distribute them onto other disks. We developed a new dynamic file assignment strategy called DORA (*d*ynamic round *r*obin with replication), which integrates file replication techniques into file assignment schemes for a user access pattern changing environment. DORA first sorts all files according to file size. Next, it assigns the files to disks in a round-robin fashion so as to distribute the load of all files evenly across all disks. Finally, DORA dynamically keeps track of the load of all files and the load on each disk. For some extremely hot files, it then creates replicas to effectively distribute request accesses on these files across all disks in a disk array. Using extensive simulations, we evaluated the performance of DORA by comparing it with one of the best existing dynamic file assignment algorithms, C-V.

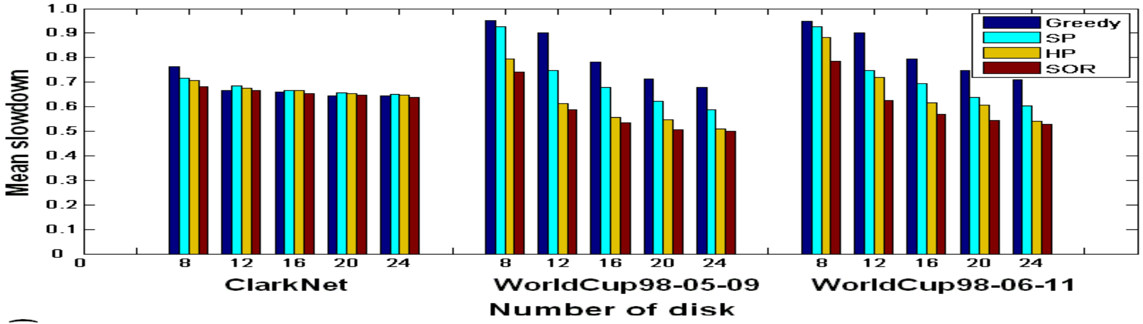


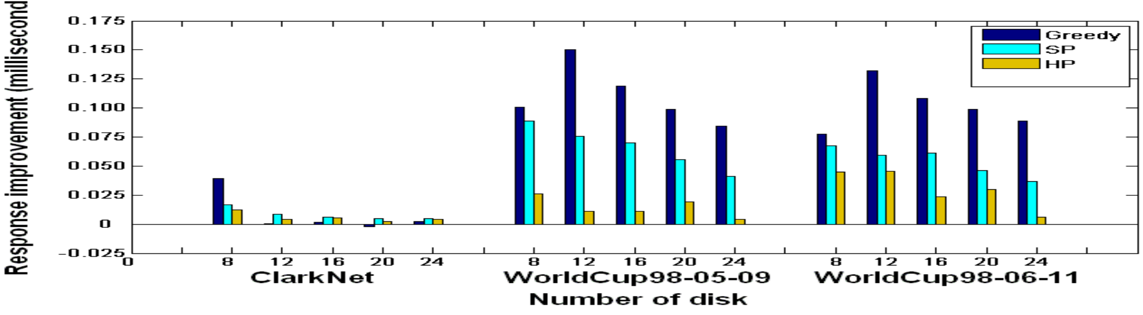
[Xie and Sharma, 2009] Mobile disk arrays, disk arrays located in mobile data centers, are crucial for mobile applications such as disaster recovery. Due to their unusual application domains, mobile disk arrays face several new challenges including harsh operating environments, very limited power supply, and extremely small number of spare disks. Consequently, data reconstruction schemes for mobile disk arrays must be performance-driven, reliability-aware, and energy-efficient. In this paper, we develop a flash assisted data reconstruction strategy called CORE (<u>c</u>ollaboration-<u>o</u>riented <u>re</u>construction) on top of a hybrid disk array architecture, where hard disks and flash disks collaborate to shorten data reconstruction time, alleviate performance degradation during disk recovery. Experimental results demonstrate that CORE noticeably improves the performance and energy-efficiency over existing schemes. Compared with DOR, CORE on average reduces reconstruction duration and mean user response time during reconstruction by 50.4% and 65.3%, respectively. In terms of energy consumption, CORE on average saves energy by 43.4%. Compared with PRO, CORE on average shrinks reconstruction duration and mean user response time during reconstruction by 48.2% and 61.9%, respectively. In addition, CORE saves energy on average by 42.5%.



[Xie and Sun, 2009] The problem of statically assigning non-partitioned files in a parallel I/O system has been extensively investigated. A basic workload characteristic assumption of most existing solutions to the problem is that there exists a strong inverse correlation between file access frequency and file size. In other words, the most popular files are typically small in size, while the large files are relatively unpopular. Recent studies on the characteristics of web proxy traces suggested, however, the correlation, if any, is so weak that it can be ignored. Hence, the following two questions arise naturally. First, can existing algorithms still perform well when the workload assumption does not hold? Second, if not, can one develop a new file assignment strategy that is immune to the workload assumption? To answer these questions, we first evaluate the performance of three well-known file assignment algorithms with and without the workload assumption, respectively. Next, we develop a novel static non-partitioned file assignment strategy for parallel I/O systems, called static round-robin (SOR), which is immune to the workload assumption. Comprehensive experimental results show that SOR consistently improves the performance in terms of mean response time over the existing schemes. Experimental results show that when the distribution of access rates across the files and the distribution of file sizes were inversely correlated with the same skew parameter θ, SOR consistently improves the performance of parallel I/O systems in terms of mean response time over three well-known file assignment algorithms. Compared to SP, one of the best existing static non-partitioned file assignment algorithms, SOR obviously achieves improvement in mean response time. When the correlation between file access frequency and file size is negligible, SOR still consistently performs better when file size exhibits a uniform distribution.







References

[Zong et al. 2007a] Z.-L. Zong, M.E. Briggs, N.W. O'Connor, and X. Qin, “An Energy-Efficient Framework for Large-Scale Parallel Storage Systems,” Proc. 21st Int'l Parallel and Distributed Processing Symp. (IPDPS), 8th IEEE Int'l Workshop Parallel and Distributed Scientific and Engineering Computing, Long Beach, CA, March 2007.

[Zong et al., 2007b] Z.-L. Zong, M.E. Briggs, N.W. O'Connor, X. Qin, M. Alghamdi, and Y.-M. Yang, “Design and Performance Analysis of Energy-Efficient Parallel Storage Systems,” the Commodity Cluster Symposium 2007 (CCS), Annapolis, Maryland, July 2007.

[Zong et al., 2007c] Z.-L. Zong, K. Bellam, X.-J. Ruan, A. Manzanares, X. Qin, and Y.-M Yang, “A Simulation Framework for Energy-efficient Data Grids,” Proc. Winter Simulation Conference, Washington, D.C., Dec. 2007.

[Zong et al., 2007d] Z.-L. Zong, X. Qin, M. Nijim, X.-J. Ruan, K. Bellam, and M. Alghamdi, “Energy-Efficient Scheduling for Parallel Applications Running on Heterogeneous Clusters,” Proc. 36th International Conference on Parallel Processing (ICPP), Sept. 2007.

[Ruan et al., 2007] X.-J. Ruan, X. Qin, M. Nijim, Z.-L. Zong, and K. Bellam, “An Energy-Efficient Scheduling Algorithm Using Dynamic Voltage Scaling for Parallel Applications on Clusters,” Proc. 16th IEEE Int'l Conference on Computer Communications and Networks (ICCCN), Honolulu, Hawaii, Aug. 2007.

[Zong et al., to be published] Z.-L. Zong, A. Manzanares, X. Qin, “Load-Balancing Strategies for Energy-Efficient Parallel Storage Systems with Buffer Disks.”

[Manzanares et al., to be published] A. Manzanares, K. Bellam, and X. Qin, “Energy-Efficient Prefetching for Parallel I/O Systems with Buffer Disks.”

[Xie and Sun, 2008] T. Xie and Y. Sun, “Sacrificing Reliability for Energy Saving: Is It Worthwhile for Disk Arrays?,” Proc. 22nd IEEE International Parallel and Distributed Processing Symposium (IPDPS 2008), Miami, Florida, USA, April 14-18, 2008.

[Madathil et al., 2008] D. K. Madathil, R. B. Thota, P. Paul, and Tao Xie “A Static Data Placement Strategy towards Perfect Load-Balancing for Distributed Storage Clusters," The 7th International Workshop on Performance Modeling, Evaluation, and Optimization of Ubiquitous Computing and Networked Systems (PMEO UCNS 2008), in conjunction with the 22nd IEEE International Parallel and Distributed Processing Symposium (IPDPS 2008), Miami, Florida, USA, April 14-18, 2008.

[Xie and Sun, 2007] T. Xie and Y. Sun, “No More Energy-Performance Trade-Off: A New Data Placement Strategy for RAID-Structured Storage Systems," Proc. 14th Annual IEEE International Conference on High Performance Computing (HiPC 2007), Lecture Notes in Computer Science (LNCS 3834), pp.35-46, Goa, India, December 18-21, 2007.

[Xie, 2007] T. Xie, "SOR: A Static File Assignment Strategy Immune to Workload Characteristic Assumptions in Parallel I/O Systems," Proc. 36th International Conference on Parallel Processing (ICPP 2007), XiAn, China, September 10-14, 2007.

[Bellam et al., 2008] K. Bellam, A. Manzanares, X. Ruan, X. Qin, and Y.-M. Yang, "Improving Reliability and Energy Efficiency of Disk Systems via Utilization Control," *Proc. IEEE Symposium on Computers and Communications* (ISCC'08), July 2008.

[Liu et al., 2008a] C. Liu, X. Qin, and S. Li, "PASS: Power-Aware Scheduling of Mixed Applications with Deadline Constraints on Clusters," *Proc. the 17th Int'l Conf. Computer Communications and Networks* (ICCCN), St. Thomas, Virgin Islands, Aug. 2008.

[Liu et al., 2008b] C. Liu, X. Qin, S. Kulkarni, C.-J. Wang, S. Li, A. Manzanares, and S. Baskiyar, "Distributed Energy-Efficient Scheduling for Data-Intensive Applications with Deadline Constraints on Data Grids," *Proc. 27th IEEE International Performance Computing and Communications Conference* (IPCCC), Dec. 2008.

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

[Manzanares et al., 2008b] A. Manzanares, D. Hamilton, and X. Qin, "The Relationship Between Software Architecture and Visual Programming Languages," *Proc. Grand Challenges in Modeling & Simulation*, Edinburgh, Scotland, June 2008.

[Nijim et al., 2008a] M. Nijim, A. Manzanares, and X. Qin, "An Adaptive Energy-Conserving Strategy for Parallel Disk Systems," *Proc. the 12th IEEE Int'l Symp. Distributed Simulation and Real Time Applications* (DS-RT), Oct. 2008.

[Nijim et al., 2008b] M. Nijim, Z.-L. Zong, K. Bellam, X.-J. Ruan and X. Qin, "Security-Aware Cache Management for Cluster Storage Systems," *Proc. the 17th Int'l Conf. Computer Communications and Networks* (ICCCN), St. Thomas, Virgin Islands, Aug. 2008.

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

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

[Xie and Wang, 2008] T. Xie and H. Wang, "MICRO: A Multi-level Caching-based Reconstruction Optimization for Mobile Storage Systems," IEEE Transactions on Computers, Vol. 57, No. 10, pp. 1386-1398, October 2008.

[Xie and Sun, 2008b] T. Xie and Y. Sun, "PEARL: Performance, Energy, and Reliability Balanced Dynamic Data Redistribution for Next Generation Disk Arrays," The 16th Annual Meeting of the IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and

Telecommunication Systems (MASCOTS), Baltimore, Maryland, USA, September 8-10, 2008.

[Zong et al., 2008] Z.-L. Zong, M. Nijim, and X. Qin, “Energy-Efficient Scheduling for Parallel Applications on Mobile Clusters,” *Cluster Computing: The Journal of Networks, Software Tools and Applications*, vol. 11, no. 1, pp. 91 - 113, March 2008.

[Ruan et al. 2009a] X.-J. Ruan, S. Yin, A. Manzanares, J. Xie, Z.-Y. Ding, J. Majors, and X. Qin, “ECOS: An Energy-Efficient Cluster Storage System,” Proc. the 28th International Performance Computing and Communications Conference (IPCCC), Phoenix, Arizona, Dec. 2009. (40%, 8 pages)

[Ruan et al. 2009b] X.-J. Ruan, A. Manzanares, S. Yin, Z. -L. Zong, and X. Qin, “Performance Evaluation of Energy-Efficient Parallel I/O Systems with Write Buffer Disks,” Proc. the 38th Int'l Conf. on Parallel Processing (ICPP), Vienna, Austria, Sept. 2009. (Acceptance Rate: 32.3%, 71/220) (40%, 8 pages)

[Nijim et al. 2009] M. Nijim, A. Manzanares, X.-J. Ruan, and X. Qin, "HYBUD: An Energy-Efficient Architecture for Hybrid Parallel Disk Systems," Proc. the 18th Int'l Conf. on Computer Communications and Networks (ICCCN), San Francisco, CA, Aug. 2009. (Acceptance Rate: 29%). (30%, 8 pages)

[Manzanares et al. 2009] BUD’09 A. Manzanares, X.-J. Ruan, S. Yin, M. Nijim, X. Qin, and W. Luo, “Energy-Aware Prefetching for Parallel Disk Systems: Algorithms, Models, and Evaluation,” Proc. the 8th IEEE International Symposium on Network Computing and Applications (NCA), July 2009. (50%, 8 pages)

[Ruan et al. 2009c] X.-J. Ruan, A. Manzanares, K. Bellam, X. Qin, “DARAW: A New Write Buffer to Improve Parallel I/O Energy-Efficiency,” Proc. the 24th Annual ACM Symposium on Applied Computing (SAC), March 2009. (Acceptance Rate: 29%) (50%, 8 pages)

[Tjioe, Widjaja, Lee, and Xie, 2009] J. Tjioe, R. Widjaja, A. Lee, and T. Xie, “DORA: A Dynamic File Assignment Strategy with Replication,” The 38th International Conference on Parallel Processing (ICPP 2009), Vienna, Austria, September 22-25, 2009.

[Xie and Sharma, 2009] T. Xie and A. Sharma, “Collaboration-Oriented Data Recovery for Mobile Disk Arrays,” The 29th International Conference on Distributed Computing Systems (ICDCS 2009), Montreal, Quebec, Canada, June 22-26, 2009.

[Xie and Sun, 2009] T. Xie and Y. Sun, “A File Assignment Strategy Independent of Workload Characteristic Assumptions,” ACM Transactions on Storage, Vol. 5, Issue 3, Article 10, November 2009.

[Lewis et al., 2010] J. Lewis*, M. I. Alghamdi*, M. A. Assaf*, X.-J. Ruan*, Z.-Y. Ding*, and X. Qin, “An Automatic Prefetching and Caching System,” *Proc. the 29th International Performance Computing and Communications Conference* (IPCCC), Albuquerque, New Mexico, Dec. 2010.

[Ruan, et al. 2010] X.-J. Ruan*, Q. Yang*, M. I. Alghamdi*, S. Yin*, Z.-Y. Ding*, J. Xie*, J. Lewis*, and X. Qin, “ES-MPICH2: A Message Passing Interface with Enhanced Security,” *Proc. the 29th International Performance Computing and Communications Conference* (IPCCC), Albuquerque, New Mexico, Dec. 2010. (40%, 8 pages)

[Qiu, et al. 2010] M.-K. Qiu, J.-W. Niu, L. T. Yang, X. Qin, S.-L. Zhang, and B. Wang, "Energy-Aware Loop Parallelism Maximization for Multi-Core DSP Architectures," *Proc. IEEE/ACM International Conference on Green Computing and Communications* (GreenCom-2010), Hangzhou, China, Dec 18-20, 2010. (Best Paper Award. 16%, 8 pages)

[Yang, et al. 2010] Q. Yang*, X.-J. Ruan*, A. Lim, and X. Qin, "Location Privacy Protection in Contention Based Forwarding for VANETs," *Proc. IEEE Globecom 2010 Wireless Networking Symposium*, Miami, FL, Dec. 6-10, 2010. (Acceptance Rate: 35%, 1300/3688) (10%, 8 pages)

[Manzanares, et al. 2010] A. Manzanares*, X.-J. Ruan*, S. Yin*, J. Xie*, Z.-Y. Ding*, Y. Tian*, J. Majors*, and X. Qin, "Energy Efficient Prefetching with Buffer Disks for Cluster File Systems," *Proc. IEEE International Conference on Parallel Processing* (ICPP), San Diego, CA, Sept. 13-16, 2010. (40%, 10 pages)

[Nijim, et al. 2010] M. Nijim, Z.-L. Zong, X. Qin, Y. Nijim, "Multi-Layer Prefetching for Hybrid Storage Systems: Algorithms, Models, and Evaluations," *Proc. IEEE International Conference on Parallel Processing Workshops* (ICPPW), San Diego, CA, Sept. 13-16, 2010. (10%, 6 pages)

[Yin, et al. 2010] S. Yin*, M. I. Alghamdi*, X.-J. Ruan*, M. Nijim, A. Tamilarasan*, Z.-L. Zong*, X. Qin, and Y.-M. Yang, "Improving Energy Efficiency and Security for Disk Systems," *Proc. 12th IEEE International Conference on High Performance Computing and Communications (*HPCC-10*)*, Melbourne, Australia, September 1-3, 2010. (Acceptance Rate: 19%, 58/304) (40%, 8 pages).

[Liu, et al. 2010] Z. Liu*, F. Wu, X. Qin, C.-S. Xie, J. Zhou*, and J.-Z. Wang*, "TRACER: A Trace-Replay Based Load-controllable Scheme for Evaluating Energy-efficiency of Mass Storage Systems," *Proc. IEEE International Conference on Cluster Computing* (CLUSTER), Heraklion, Crete, Greece, Sept. 20-24, 2010. (40%, 10 pages)

[Qiu, et al. 2010] J. Li, M. Qiu, J. Niu, W. Gao, Z. Zong, and X. Qin, "Feedback Dynamic Algorithms for Preemptable Job Scheduling in Cloud Systems", *Proc. 2010 IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 561-564. Toronto, Canada, Sep. 2010. (10%, 4 pages)

[Wang, et al. 2010] P. Wang*, Diqing Hu, C.-S. Xie, J.-Z. Wang*, and X. Qin, "A Fine-grained Data Reconstruction Algorithm for Solid-state Disks," *Proc. the 5th IEEE International Conference on Networking, Architecture, and Storage* (NAS), July 2010. (Acceptance Rate: 24%, 43/178) (20%, 8 pages)

[Xie, et al. 2010] J. Xie*, S. Yin*, X.-J. Ruan*, Z.-Y. Ding*, Y. Tian*, J. Majors*, A. Manzanares*, and X. Qin, "Improving MapReduce Performance via Data Placement in Heterogeneous Hadoop Clusters," *Proc. 19th International Heterogeneity in Computing Workshop*, Atlanta, Georgia, April 2010. (40%, 8 pages)

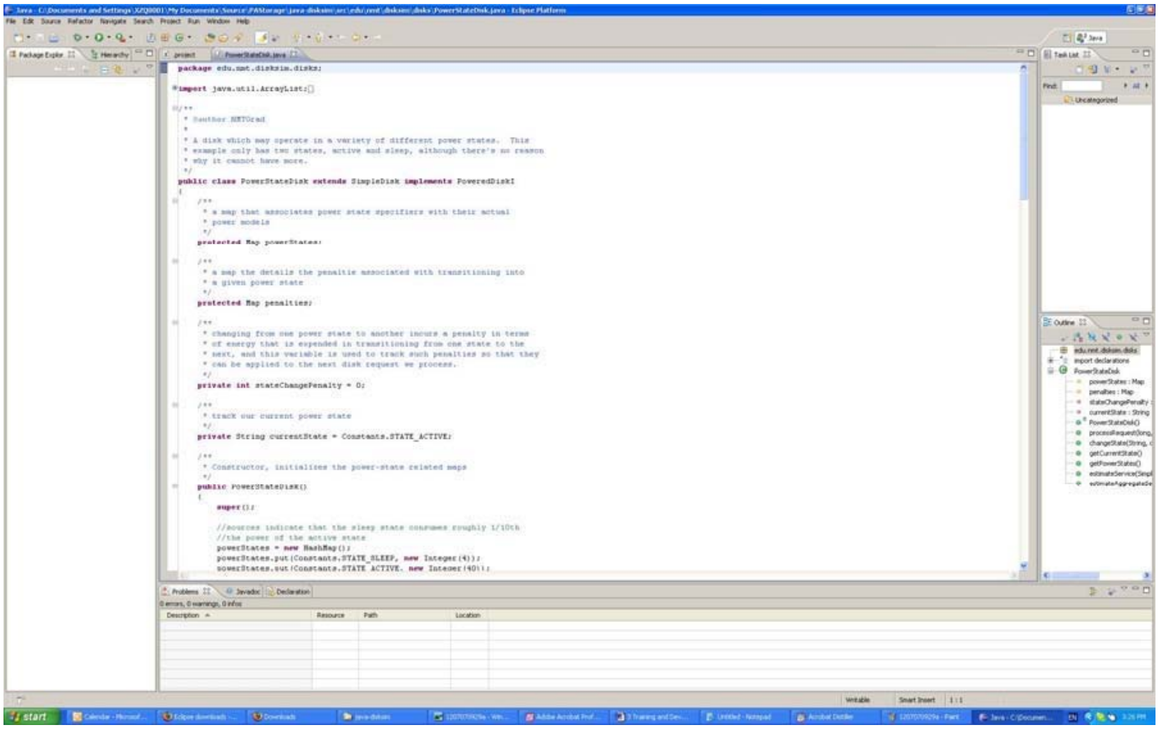
[Zong et al., 2011] Z.-L. Zong, A. Manzanares, X.J. Ruan, K. Bellam, X. Qin, "Heat-Based Dynamic Data Caching: A Load Balancing Strategy for Energy-Efficient Parallel Storage Systems with Buffer Disks." Proc. the 27th IEEE Symposium on Massive Storage Systems and Technologies: Research Track (MSST), May 2011.

3. Training and Development

Student Support

This project has directly supported about 8 students including 4 graduate students and and 4 undergraduate students. The project also indirectly contributed to approximately 65 undergraduate students.

The BUD project has directly supported about 30 students and indirectly contributed to approximately 100 students. We developed a Java-based disk simulator to evaluate our proposed energy conservation techniques for disks.



```java
package edu.mnt.disksim.disks;

import java.util.ArrayList;

/**
 * @author NETGrad
 *
 * A disk which may operate in a variety of different power states.  This
 * example only has two states, active and sleep, although there's no reason
 * why it cannot have more.
 */
public class PowerStateDisk extends SimpleDisk implements PoweredDiskI
{
    /**
     * a map that associates power state specifiers with their actual
     * power models
     */
    protected Map powerStates;

    /**
     * a map the details the penaltie associated with transitioning into
     * a given power state
     */
    protected Map penalties;

    /**
     * changing from one power state to another incurs a penalty in terms
     * of energy that is expended in transitioning from one state to the
     * next, and this variable is used to track such penalties so that they
     * can be applied to the next disk request we process.
     */
    private int stateChangePenalty = 0;

    /**
     * track our current power state
     */
    private String currentState = Constants.STATE_ACTIVE;

    /**
     * Constructor, initializes the power-state related maps
     */
    public PowerStateDisk()
    {
        super();

        //sources indicate that the sleep state consumes roughly 1/10th
        //the power of the active state
        powerStates = new HashMap();
        powerStates.put(Constants.STATE_SLEEP, new Integer(4));
        powerStates.put(Constants.STATE_ACTIVE, new Integer(40));
```

0 errors, 0 warnings, 0 infos

We developed a graduate level class - COMP7970 Storage Systems – to train graduate students to implement disk simulators and conduct research in the realm of storage systems. The following table summaries the evaluation of this training class.

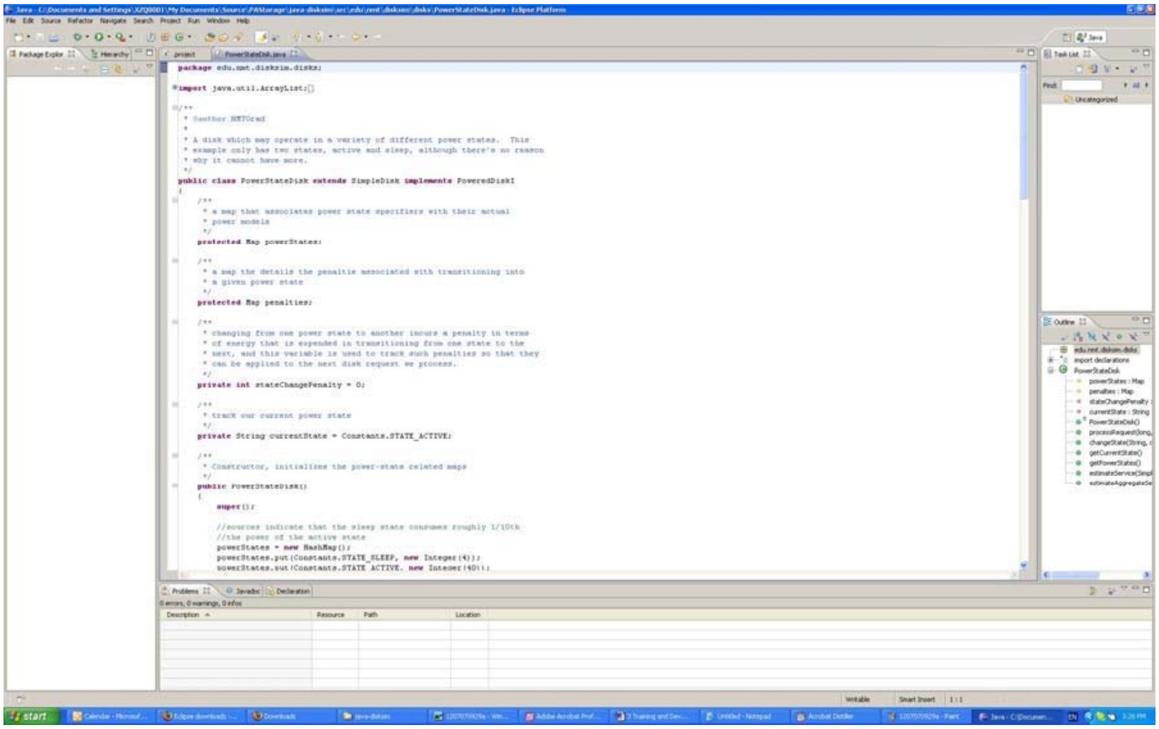
Rating system: 5- Agree Strongly 4-Agree 3-Unsure 2-Disagree 1-Disagree strongly 0-Not applicable

I have learnt the following concepts and techniques:	Max	Min	Average
Energy-efficient storage systems	5	4	4.6
Dynamic power management	5	3	4.6
Design of low-power devices	5	0	3.5
Disk simulations	5	2	4.1
Reliable storage systems	5	3	4.2
Disk arrays	5	4	4.4

I/O-aware load balancing	5	4	4.6
BUD: Energy-efficient parallel disk systems with buffer disks	5	4	4.6
General Course Observations			
I like this course.	5	4	4.8
I will recommend this course to others.	5	4	4.8

The graduate students who took the training course were very satisfied with the material covered in the class. The students are very likely to recommend this class to other students in the department.

We developed a Java-based disk simulator to evaluate our proposed reliability models for energy-efficient storage systems.



```
package edu.nmt.disksim.disks;

import java.util.ArrayList;

/**
 * @author NMTGrad
 *
 * A disk which may operate in a variety of different power states.  This
 * example only has two states, active and sleep, although there's no reason
 * why it cannot have more.
 */
public class PowerStateDisk extends SimpleDisk implements PoweredDiskI
{
    /**
     * a map that associates power state specifiers with their actual
     * power models
     */
    protected Map powerStates;

    /**
     * a map the details the penaltie associated with transitioning into
     * a given power state
     */
    protected Map penalties;

    /**
     * changing from one power state to another incurs a penalty in terms
     * of energy that is expended in transitioning from one state to the
     * next, and this variable is used to track such penalties so that they
     * can be applied to the next disk request we process.
     */
    private int stateChangePenalty = 0;

    /**
     * track our current power state
     */
    private String currentState = Constants.STATE_ACTIVE;

    /**
     * Constructor, initializes the power-state related maps
     */
    public PowerStateDisk()
    {
        super();

        //sources indicate that the sleep state consumes roughly 1/10th
        //the power of the active state
        powerStates = new HashMap();
        powerStates.put(Constants.STATE_SLEEP, new Integer(4));
        powerStates.put(Constants.STATE_ACTIVE, new Integer(40));
```

We developed COMP7370 – Advanced Computer Security - for graduate students. This course was intended to give students a strong background in understanding design and development of secure systems in general and secure storage systems in particular.

The PI at Auburn educated graduate students to investigate the issues of energy efficiency and reliability in parallel disk systems. Graduate students also learnt how to use disk simulators to conduct research in the area of parallel disk systems. More than 93% of the students who took the storage systems class decided to recommend this class to other undergraduate and graduate students in the department of computer science and software engineering at Auburn University. All the students who took the class claimed that they like this type of research projects focusing on storage systems.

One of the major education objectives in this project is to recruit new undergraduate students, especially women and minorities, to conduct research in the area of storage systems and energy conservation technology in computer systems. To attract both the best undergraduate to participate in this project, in April 2008 we organized a workshop (see the following photo) that offered an opportunity for minority and women undergradate students to learn basic concepts in reliable and energy-efficient storge systems. In the long run, we plan to recuit monority students to conduct intensive research in fault-tolerant storage systems and reliability analysis with the PI. In this workshop, the PI emphasized new techniques and exciting findings of building mathematical reliability models and reducing energy dissipation in parallel disk systems.

A laboratory tour was held after the workshop. In the tour, the underrepresented students who participated in the workshop visited our new storage systems laboratory (see the photo below). This workshop along with the laboratory tour aims to offer the undergraduate students an opportunity to gain first-hand experience in designing and implementing reliable and energy-efficient storage systems.

Research Experience for Undergraduate Students

To recruit new undergraduate students, especially women and minorities, to conduct research in the area of computer security, we designed a research program that offers ample opportunity to undergraduate students to do intensive research in information assurance with the PIs. In particular, students and the PIs are brought together to conduct research experiments in the field of secure and energy-efficient storage systems. The photo below shows two undergraduate students - Tsukasa Ogihara (right) and Joshua Lewis (middle) - are building a cluster computing system using commodity-off-the-shelf (COTS) hardware components.

The cluster system (see the photo below) built by our undergraduate research assistants will be used as a high-performance computing platform to support our computer security education. The cluster recently build in our department at Auburn supports security middleware services for secure software applications. We will use this cluster computing platform to design and implement study how to improve software applications' quality-of-security without adversely affecting performance.

Advanced Computer Security for Graduate Students

We developed COMP7370 – Advanced Computer Security - for graduate students. This course was intended to give students a strong background in understanding design and development of secure systems in general and secure storage systems in particular.

The PI at Auburn educated graduate students to investigate the issues of energy efficiency and reliability in parallel disk systems. Graduate students also learnt how to use disk simulators to conduct research in the area of parallel disk systems. More than 93% of the students who took the storage systems class decided to recommend this class to other undergraduate and graduate students in the department of computer science and software engineering at Auburn University. All the students who took the class claimed that they like this type of research projects focusing on storage systems.

Contributions to Courses

This project has directly and indirectly contributed to the following classes:
COMP7970: Storage Systems
COMP4300: Computer Architecture
COMP4370: Computer and Network Security
COMP7370: Advance Computer and Network Security
CS696 Advanced Distributed Systems
CS325: Principles of Operating Systems
CS331: Computer Architecture
CS531: Advanced Computer Architecture

4. Outreach Activities

The BUD outreach activities include curriculum enrichment presentations, engineering clubs, and tutorial services.

We offered a presentation in the Commodity Cluster Symposium 2007 in Baltimore, MD. In this presentation, we described the design and performance analysis of an energy-efficient parallel storage systems tailored for cluster computing platforms. In addition, we give a presentation in the IEEE International Symposium on Parallel and Distributed Processing 2007. We described to the audience the design of the BUD disk architecture. We also shared our experience of carrying out the BUD project with the audience.

To attract the best minority undergraduate to participate in this project, the PI offered an undergraduate course – introduction to computer and network security – at the Alabama State University, which is one of the historically black colleges and universities (HBCU).

Secured storage systems were introduced in this class. The PI educated the minority students about important security issues in in large-scale storage systems. The PI also shared our experience of reliable and energy-efficient parallel disk systems with the African-American students who took this class.

Dr. Tao Xie, a co-PI of this project, gave a presentation entitled “Understanding the Relationship Between Energy-Saving and Disk Reliability,” at the Computer Science & Engineering Department at the University of California, Riverside on May 30, 2008.

In the past year, the PI had given the following research talks related to this NSF funded project.

1. “An Application-Oriented Approach for Computer Security Education,” invited talk at the Information Security and Computer Applications (ISCA2011) Conference, Feb. 25, 2011.
2. “A Novel Application-Oriented Approach to Teaching Computer Security Courses”. Poster Session at NSF CCLI/TUES Conference, January 27, 2011.
3. “Energy Efficient Prefetching – From models to Implementation”. Seminar talk at Huazhong University of Science and Technology, Wuhan, Hubei, China. June 2010.
4. “How to Read Papers?” Training Session for REU students at Auburn University, May 18, 2010.
5. “How to Succeed in the AU REU Program?” Training Session for REU students at Auburn University, May 17, 2010. “Improving Energy-Efficiency and Reliability of Storage Systems,” Seminar talk at the University of New Orleans, Sept. 4, 2009.
6. “Can We Improve Energy Efficiency of Secure Disk Systems without Modifying Security Mechanisms?” the IEEE NAS’09 Conference, ZhangJiaJie, China, July 10, 2009.

7. “Security-Aware Scheduling for Real-Time Parallel Applications on Clusters,” Lecture at Huazhong University of Science and Technology, Wuhan, Hubei, China. June 22, 2009.
8. “How to Read Papers?” Seminar talk at Wuhan National Laboratory for Optoelectronics, Wuhan, China, June 17, 2009.
9. “Energy Efficient Scheduling for High-Performance Clusters,” Seminar talk at Huazhong University of Science and Technology, Wuhan, Hubei, China. June 8, 2009.
10. “An Overview of Auburn University,” Seminar talk at Nanjing University of Information Science and Technology, Nanjing, China, June 3, 2009.
11. “Thinking About Going to Graduate School?” Seminar talk at Nanjing University of Information Science and Technology, Nanjing, China, June 3, 2009.
12. “How to Write Research Papers, Part 1 – General Principles,” Seminar talk at Taiyuan University of Science and Technology, Taiyuan, Shanxi, China, May 27, 2009.
13. “Energy Efficient Scheduling for High-Performance Clusters,” Seminar talk at Taiyuan University of Science and Technology, Taiyuan, Shanxi, China. May 26, 2009.
14. “Energy Efficient Resource Management for High-Performance Computing Platforms,” Seminar talk at Wuhan National Laboratory for Optoelectronics, Wuhan, China, May 15, 2009.
15. “How to Write Research Papers, Part 1 – General Principles,” Seminar talk at Wuhan National Laboratory for Optoelectronics, Wuhan, China, May 12, 2009.