Information Sciences 178 (2008) 4403–4417

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

EED: Energy Efficient Disk drive architecture ☆

Yuhui Deng [a],*, Frank Wang [a], Na Helian [b]

[a] Center for Grid Computing, Cambridge-Cranfield High Performance Computing Facilities, Cranfield University Campus, Bedfordshire MK43 0AL, United Kingdom
[b] Department of Computer Science, University of Hertfordshire, United Kingdom

ARTICLE INFO

Article history:
Received 30 December 2007
Received in revised form 18 June 2008
Accepted 26 July 2008

Keywords:
Disk drive
Energy efficiency
Data access pattern
Architecture
Non-volatile memory

ABSTRACT

Energy efficiency has become one of the most important challenges in designing future computing systems, and the storage system is one of the largest energy consumers within them. This paper proposes an Energy Efficient Disk (EED) drive architecture which integrates a relatively small-sized NAND flash memory into a traditional disk drive to explore the impact of the flash memory on the performance and energy consumption of the disk. The EED monitors data access patterns and moves the frequently accessed data from the magnetic disk to the flash memory. Due to the data migration, most of the data accesses can be satisfied with the flash memory, which extends the idle period of the disk drive and enables the disk drive to stay in a low power state for an extended period of time. Because flash memory consumes considerably less energy and the read access is much faster than a magnetic disk, the EED can save significant amounts of energy while reducing the average response time. Real trace driven simulations are employed to validate the proposed disk drive architecture. An energy coefficient, which is the product of the average response time and the average energy consumption, is proposed as a performance metric to measure the EED. The simulation results, along with the energy coefficient, show that the EED can achieve an 89.11% energy consumption reduction and a 2.04% average response time reduction with cello99 trace, a 7.5% energy consumption reduction and a 45.15% average response time reduction with cello96 trace, and a 20.06% energy consumption reduction and a 6.02% average response time reduction with TPC-D trace, respectively. Traditionally, energy conservation and performance improvement are contradictory. The EED strikes a good balance between conserving energy and improving performance.

1. Introduction

Energy saving has become one of the most important challenges in designing future computing systems. The increasing demands for higher performance, versatile functionality, and a better user interfaces have been escalating energy consumption at an unprecedented rate [12].

Over the last decade, many research efforts have been invested in energy conservation of disk drives for mobile systems, because the energy consumed by these disk drives determines the battery life of the system. Douglis et al. [8] employed a trace driven simulation to evaluate different energy saving policies. They found that threshold policies which spin down the disk drive after 1–10 s come close to the energy consumption of the optimal off-line algorithm. Their results also indicated that in some cases the threshold algorithms cause increased system delay. Lu and Micheli [26] proposed an adaptive algo-

☆ The initial work in this paper was conducted when Yuhui Deng was a Research Officer in Cranfield University. This paper was revised after Yuhui Deng joined EMC Research China as a Senior Research Scientist.

* Corresponding author.
E-mail addresses: deng_derek@emc.com, yuhuid@hotmail.com (Y. Deng).

0020-0255/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2008.07.022

rithm for dynamic power management. By adaptively adjusting the prediction of future requests, the algorithm can predict session length and spin down components between sessions to save energy. Compared with other approaches, the algorithm can reduce energy consumption with less impact on performance and reliability. Li et al. [24] performed a quantitative analysis of the potential costs and benefits of spinning down disk drives. They concluded that almost all the energy consumed by a disk drive can be conserved with little affect on performance. Helmbold et al. [16] used a simple machine learning algorithm which adapts to the pattern of recent disk activity to exploit the burst nature of user activity. The algorithm performs better than all other known algorithms, even outperforming the best fixed time-out mechanism. Dempsey [46] is a disk simulation environment which includes accurate modeling of disk energy consumption. Dempsey attempts to accurately estimate the energy consumption of specific disk stages including seeking, rotation, reading, writing, and idle periods. The results show that accurate modeling of disk behaviours during idle periods is critical to the accuracy of any energy model.

Recently, the research community has been very active in the area of energy conservation for high-end storage systems. Fan et al. [10] investigated the power consumption of the major components within a typical server. They reported that the peak power of one X86 CPU, one Motherboard, one PCI expansion slot, one IDE disk drive, one fan, and one DDR memory are 40 W, 25 W, 25 W, 12 W, 10 W, 9 W, respectively. From a power standpoint, it seems one disk drive is not a problem. Even the addition of several dozen disk drives would hardly be a concern. However, if hundreds or thousands of disk drives are put together, it will quickly become a big headache. One example shows the storage subsystem accounting for 27% of the energy consumed in a data centre [33]. To worsen the situation, this fraction is swiftly increasing as storage requirements are rising by 60% annually [29]. New data centres in the US were projected to demand 5GW of power (which is about 10% of the current generating capacity of California) and cost $4 billion/year to power in 2005 [4]. Popular Data Concentration (PDC) [32] migrates frequently accessed data to a subset of the disk drives. The goal is to skew the load towards a few of the disks, so that others can be transitioned to low power states. This policy can save energy only when the workload on the server is extremely low, but real-world workloads exhibit complex behaviour which is difficult to predict. Massive Array of Idle Disks (MAID) [7] uses a few additional cache disks running in the active state to hold recently accessed data blocks. Other disks can be put in low power state due to their extended idle periods. MAID is ideally suited for the storage of data with write-once/read-occasionally access patterns such as remote backup. However, RAID architectures are aimed primarily at achieving high performance by using multiple disk drives in parallel. MAID trades parallelism for energy conservation by using a few cache disk drives. Li and Wang [25] studied several redundancy based dynamic I/O request scheduling and cache management policies at the RAID controller level. The policies power down the redundant disks in RAID 1 and RAID 5 to save energy. Son et al. [42] proposed and evaluated a profile driven disk layout scheme which determines the disk number, strip unit, etc. to reduce energy consumption. Hibernator [47] is a disk array energy management system which combines several techniques (e.g. disks that can spin at different speeds, an approach for dynamically deciding which disk drive should spin at which speed, efficient ways to migrate the right data to an appropriate speed disk drive automatically, and automatic performance boosts if there is a risk that performance goals might not be met due to disk power management) to save energy while meeting performance goals.

There are a few isolated contributions which focus on the energy conservation of disk drive architecture in the community. Dynamic Rotations Per Minute (DRPM) [14] is a dynamic multi-speed disk model which spins server disks at different speeds in correlation with workloads to save energy without reducing performance. Carrera et al. [2] compared several techniques used for energy conservation. They discovered that only the multi-speed disk approach can really conserve energy on network servers. Active Disk [34] was first proposed to take advantage of the processing power of individual disk drives to run application level code. Having moving portions of an application's process execute directly on the disk drives can dramatically reduce data traffic. Due to the heat dissipation and thermal constraints, it is very difficult to further improve the data rate of disk drives. Like the Active Disk, Gurumurthi [15] suggested providing more powerful processors inside disk drives to expand computational capabilities, thus reducing the requirement of data rate to overcome the thermal constraints and boost performance.

Recently, a number of non-volatile storage technologies are emerging and bring opportunities to the architecture design of disk drives. Flash memory is a non-volatile memory which can be electrically erased and reprogrammed. Its major advantages including low energy consumption, non-volatility, and high performance have made it likely to replace disk drives in more and more systems, where size, energy, or performance are important [21]. Magnetic Random Access Memory (MRAM) combines a magnetic device with standard silicon based microelectronics to obtain the combined attributes of non-volatility, high performance, fast programming and unlimited program endurance. It provides random access with no refresh. MRAM is supposed to achieve the density of flash memory but at significantly faster write speeds and with unlimited endurance [45]. MicroElectroMechanical System (MEMS) is a very small-scale mechanical device which slides, bends, and deflects in response to electrostatic, electromagnetic, and external environmental forces. MEMS based storage is a non-volatile storage technology that merges magnetic recording material with thousands of probe based recording heads to provide online storage [40].

In this paper, we propose an Energy Efficient Disk (EED) drive architecture which can reduce energy consumption significantly, while also improving performance. The architecture integrates a relatively small flash memory into a traditional disk drive and periodically moves the frequently accessed data from the slow magnetic disk to the fast flash memory when the disk is idle. Because most of the data accesses can be satisfied with the flash memory and the flash memory uses considerably less energy than the traditional disk drive, the disk drive can remain in the low power state much longer than the traditional disk drive, thus conserving significant amounts of energy. Due to the fast access of the flash memory, the architecture can also improve performance. Trace driven simulation validates that the architecture is effective in both energy conservation and performance improvement.

The remainder of this paper is organized as follows. Section 2 introduces the background. The architecture of the EED is illustrated in Section 3. The implementation of EED is depicted in Section 4. Section 5 evaluates the EED architecture through real trace driven simulation. Section 6 concludes the paper with remarks on its contributions. There is also a brief discussion of the work and indications of future research in Section 6.

2. Background

2.1. Energy conservation

Disk drives have two components that contribute to their overall energy demands. The first one is a 12 V spindle motor used to spin the platters and drive the head motors. The second one is a 5 V supply used to power the analog-to-digital converters, servo-control DSP's and interface logic [7]. Due to the mechanical nature, the hardware support for disk energy conservation has not been changed too much over the years. Most modern disk drives have four power states; namely active, idle, standby, and sleep. Disk drives only work in active state. When a data access is completed and there is no succeeding request, the disk drive is transferred to the idle state where the disk platters are still spinning, but the electronics may be partially un-powered, and the heads may be parked or unloaded. If the disk drive receives a request when it is in an idle state, the drive will be transferred to the active state. If the disk drive remains in the idle state for a certain amount of time, it transfers into the standby state where the disk platters are spun down and the head is moved off the disk. The sleep state powers off all remaining electronics. The disk drive is transferred back to the active state when a new request arrives [27,30].

Disk drives in standby state or sleep state use considerably less energy than disk drives in the active state. Many research efforts have gone into investigating the energy consumption of disk drives by taking advantage of this feature [8,16,24,26]. Generally, the existing approaches employed to save energy of disk drives can be classified into four categories [8,12,26]. The first one is a simple timeout strategy which has gained wide popularity and is currently implemented in many operating systems. Once a disk drive is idle for a specific period of time, which is longer than some given timeout threshold, the disk is spun down in an effort to save energy. Upon the arrival of a new request, the disk is spun up to serve the request. The timeout strategy offers good accuracy, but it wastes energy when the disk is waiting for the timeout period to expire. The second one is a dynamic prediction which is based on the behaviours of applications. For example, a series of events that are likely to happen again in the future. The method shuts down the disk drive immediately to eliminate the waiting time of the timeout strategy. However, so far it is less accurate than the simple timeout mechanism. The third one is a stochastic mechanism. The problem is that the approach usually requires offline pre-processing and the prediction could be inaccurate due to the fluctuant data access pattern. The last one is application aware power management. The mechanism can have very accurate information of the data access pattern. Unfortunately, it requires modifying the existing applications, which makes it impractical.

The above methods can incur a significant energy cost and time penalty as the disk platters have to be spun up to full speed and the heads have to be moved back before a request can be served, which requires servo calibration to accurately track the head as it moves over the drive. To justify this penalty, the energy saved by putting the disk in standby or sleep state has to be greater than the energy needed to spin it up again, and the disk has to stay in the low power state for a sufficiently long period of time to compensate for the energy overhead [47]. An important issue is that the methods cannot be applied directly to server disk drives, since the spinning down and spinning up time of the server disk drives are much longer than that of the desktop and laptop. Due to the intensive workload, it is also very difficult to find an idle interval which is long enough to spin down the server disk drives. Another important concern is that frequently spinning up and spinning down may reduce the effective life span of the drives. Therefore, a good energy conservation method should be able to strike a balance between energy consumption, performance, and life span of the disk drive.

2.2. Data access pattern

Data access patterns are a measure of how well data can be selected, retrieved, compactly stored, and reused for subsequent accesses. Data access patterns such as temporal locality and spatial locality normally have a significant impact on the storage system performance. Understanding the nature of data access patterns is crucial to properly optimizing and designing storage systems.

Staelin and Garcia-Molina [43] observed that there was a very high locality of reference on extremely large file systems. Some files in the file system have a much higher skew of accesses than others. Gomez and Santonja [13] found that some of the data blocks are extremely hot and popular, but others are rarely or never accessed in terms of the investigation of several real traces. Cherkasova and Gupta [5] reported that 14–30% of the media files accessed on the media server account for 90% of the media sessions and 92–94% of the bytes transferred. The files are viewed by 96–97% of the unique clients. They also reported that 16–19% of the media files are accessed only once. Cherkasova and Ciardo [6] addressed the characterization of web workloads, which shows that web traffic exhibits a strong concentration of references: 10% of the files accessed on the server typically account for 90% of the server requests and 90% of the bytes transferred. The above works indicate that the skew is a normal pattern of I/O workloads which cover diverse applications. The skew of accesses is often called the 90/10 rule, or the 80/10 rule. The 90/10 rule indicates that 90% I/O accesses accumulate in 10% storage capacity. The percentages are applied recursively. For example, 10% of the 10% storage resources serve 90% of the 90% I/O accesses [11].

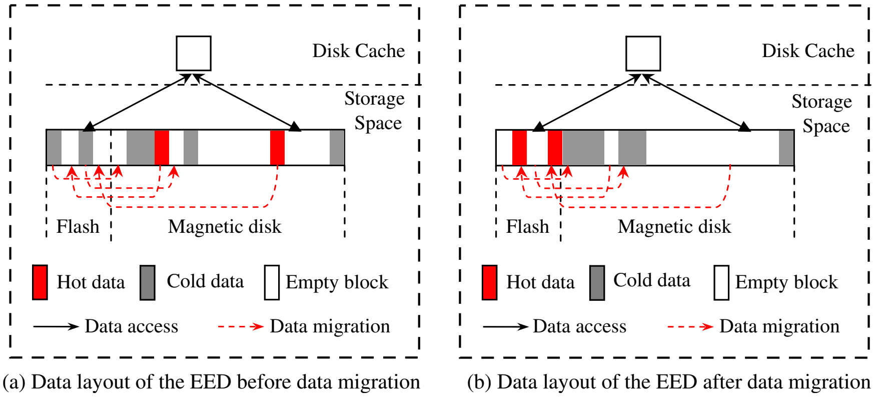
3. Architecture overview of the EED

According to the discussion in Section 2.1, the key principle of disk energy conservation is accurately predicting the idle time which can be employed to transfer the disk to low power state, or extending the length of disk idle phases and forcing transitions to standby state when this is likely to save significant amounts of energy. We employ the second method to design the EED drive architecture, which can conserve significant amounts of energy while improving performance.

The disk drive has long been a performance bottleneck of computer systems. Many research efforts have been devoted to alleviating this bottleneck. One of the most effective approaches is employing disk cache to reduce the number of disk I/Os [20,41]. Almost all modern disks employ a small amount of on-board disk cache (volatile SDRAM) to speed up data accesses (e.g. high performance disks normally employs 16 MB of disk cache). Disk cache can dramatically improve the performance of disk I/O by avoiding slow mechanical latencies if the data accesses are satisfied by the disk cache (cache hit), because accessing a byte of data in cache can be thousands of times faster than accessing a byte on the magnetic disk media. Another important effect is that the disk cache has an impact on slowing down the request rate which goes to the magnetic disk media. The effect can enlarge the interval between requests, thus extending the idle time. Due to the decreased miss ratio introduced by disk cache, disk performance can be improved and the idle time can be lengthened by increasing the disk cache size. However, while the miss ratio of disk cache decreases asymptotically with increased cache size, it begins to increase again after a certain point [41]. Therefore, we cannot improve the disk performance and extend the idle time too much by simply increasing the cache size. Furthermore, a big volatile disk cache could decrease the data reliability, especially during a power loss or a system crash. In addition, the cost of the disk cache would become a problem with the increase of the cache size.

The hierarchy of storage in current computer architectures is designed to take advantage of data access locality to improve overall performance. Each level of the hierarchy has higher speed, lower latency, and smaller size than lower levels. A traditional disk drive has two levels, including a disk cache level and a magnetic disk media level. Motivated by the emerging non-volatile storage technologies (e.g. flash memory, MRAM, and MEMS which offer low energy consumption and high performance) and the highly skewed data access pattern, we propose to design an Energy Efficient Disk drive architecture which integrates a relatively small non-volatile memory into a traditional disk drive.

A disk drive presents itself as a sequence of Logic Block Addresses (LBA) to the above disk file system. When a request arrives, the LBA is converted to a physical block address Cylinder/Head/Sector (CHS) to locate the data. The storage space of the proposed EED is divided into two areas in terms of the LBA: one is the magnetic disk media, the other is the non-volatile memory. Both areas are in the same linear storage space, marked by LBA, and are exclusive (see Fig. 1). For example, if the non-volatile memory is 10 GB, and the magnetic disk media is 100 GB, the overall storage capacity of the EED will be 110 GB. The I/O requests from disk cache can go to the non-volatile memory or the magnetic media, depending on the data location. The non-volatile memory used in EED is different from a second level cache. For the system which employs non-volatile memory as a second level cache, the data residing in the cache has a copy stored in the magnetic disk media. Therefore, the storage space of the cache and the magnetic disk media is inclusive. It indicates that integrating a 10 GB flash memory into a 100 GB disk drive can only obtain 100 GB storage capacity. Thus, with the decreasing price and the increasing capacity of flash memory, the method is not cost-effective in comparison with the EED.



(a) Data layout of the EED before data migration (b) Data layout of the EED after data migration

Fig. 1. Data layout and data migration of the EED.

The EED can automatically move the frequently accessed data from the magnetic disk media to the non-volatile memory, or move the data from the non-volatile memory to the magnetic disk media when it gets cold. Fig. 1(a) illustrates the architecture and the data layout of EED before data migration. It shows that some hot data (frequently accessed data) and cold data (rarely accessed data) are distributed across the magnetic disk media. Fig. 1(b) depicts that the hot data is migrated from the magnetic disk media to the non-volatile memory, and the cold data is moved from the non-volatile memory to the disk media. Because most of the frequently accessed data can be satisfied from the non-volatile memory, due to the data migration, the disk can be spun down and remain in the low power state much longer than a traditional disk, thus saving significant amounts of energy. Furthermore, because the read accesses of the non-volatile memory are much faster than that of the magnetic disk media, the architecture can improve performance as well.

4. Implementation

According to the discussion in Section 3, the system design mainly consists of four components: non-volatile memory, frequency monitoring, data migration, and energy calculation.

4.1. Non-volatile memory

EED is a general architecture which can integrate different non-volatile storage media. Flash memory is a non-volatile memory which can be electrically erased and reprogrammed. There are two major types of flash memory which are available on the market following different logic schemes, namely NOR and NAND. Compared with the NOR flash memory, NAND flash memory has faster erasing and write times, along with higher data density, which makes NAND flash a better candidate for data storage. A NAND flash memory is composed of a fixed number of blocks, where each block consists of a number of pages and each page has a fixed-sized main data area and a spare data area. Data on NAND flash memory is read or written at the page level, and the erasing is performed at the block level. Flash memory is also much cheaper than volatile memories such as SDRAM. For example, 1 Gbit of NAND flash memory costs 3.75$, while the same size of low power SRAM and fast SRAM cost 320$ and 614$, respectively [31,38]. We adopted NAND flash as the non-volatile memory in the proposed EED. In the following discussion, the size of non-volatile memory is 10% of the overall storage space in terms of the 90/10 rule. Therefore, if we can integrate a 32 GB flash memory into a disk drive, the overall storage capacity of EED can reach 352 GB.

4.2. Frequency monitoring

A key component of this work is frequency tracking and identification of the most frequently accessed data blocks. Data blocks are normally correlated by semantics. For example, a file block is correlated to its inode block. A tree node in a database is correlated to its parent node and its ancestor nodes. The correlations can be used to improve the effectiveness of storage caching, prefetching, data layout and disk scheduling. Many algorithms can be used to extract the correlations [17,22,23]. C-Miner [23] employs a data mining technique called frequent sequence mining to discover the block correlations in storage systems. C-Miner is an effective method to discover block correlations, especially when such correlations are complex in nature. However, storage systems normally show very simple block correlations. Spatial locality states that the probability of accessing a piece of data is higher if the data near it was just accessed. The spatial locality is an example of simple block correlation. This is why sequential prefetching is employed by most current storage systems. Ruemmler and Wilkes [35] confirmed that the simple locality is an inherent characteristic of disk drive workloads. To keep the block correlations, we cluster the consecutive data blocks into objects and move the objects, as opposed to moving the data blocks individually. An object is defined as a group of consecutive blocks in terms of LBA. Block rearrangement can incur significant overhead due to quantity, while object reorganization is much faster in comparison. It is also much easier to track the frequency of object use.

A data structure including the fields of original object location, current object location, frequency, and object status is used to describe each object. The system monitors each I/O request and checks the corresponding LBA. If the LBA falls into the address range of an object, the frequency of that object will be increased by one.

There are several typical cache replacement algorithms including Random Replacement (RR), Least Frequently Used (LFU), and Least Recently Used (LRU) [20]. RR replaces cache lines by randomly selecting a cache line to evict. The policy is very fast, requires no extra storage, and is the easiest to implement. However, it performs poorly because it does not take advantage of the spatial and temporal locality. LFU is based on the access counts of the cache lines. The cache lines which have been used least frequently are evicted. Unfortunately, the recently active but currently cold cache lines tend to remain entrenched in the cache. Therefore, the inactive data increases the miss ratio and reduces the cache performance. LRU evicts the cache lines used least in the recent past on the assumption that it will not be used in the near future. LRU is the most frequently used algorithm because it is simple and easy to implement, and offers very good performance. Therefore, we use two fixed length LRU lists including a hot list and a recent list to identify the most frequently and recently accessed objects. When the system receives a request, the corresponding object will be recorded on the recent list. If the object on the recent list is accessed again in a short period of time, the object will be promoted to the hot list. If the promoted object is already on the hot list, the object will be moved to the head of the hot list. If the hot list is full, the last object on the hot list will be degraded to the recent list. If the recent list is full, the last object on the recent list will be discarded. This method is very effective in our experiment.

4.3. Data migration

One of the main design issues involves reorganizing data with minimal impact on the foreground workload. A fixed reorganization interval presents a trade-off. If the interval is too short, frequent data reorganization may introduce too much overhead, thus decreasing disk performance. If the interval is too long, performance degradation will be incurred as the data layout becomes less well adapted to the current data access patterns. Because frequency distribution is relatively stable, it is unnecessary to keep swapping the data between the non-volatile memory and the magnetic disk media. According to an actual trace investigation, Ruemmler and Wilkes [35] reported that requests arrive at an idle disk over 70% of the time. Therefore, the data reorganization can be done when the disk is idle. Other methods can also be applied to do the data migration. For example, Lumb et al. [28] proposed a free block scheduling to replace a disk drive's rotational latency with useful background media transfers, potentially allowing background disk I/O to occur with no impact on foreground service times. For the EED, when the disk is idle, a process running in the background examines the hot list discussed in Section 4.2. If the objects on the hot list are stored in magnetic disk media, the objects will be moved to the non-volatile memory. If the objects have already existed in the non-volatile memory, the object locations will be kept unchanged. When the non-volatile memory runs out of its 90% capacity, the objects which are not on the hot list will be migrated to the magnetic disk media.

After reorganizing disk layouts, we need to redirect the requests to the new physical locations. A mapping list consisting of the data structures which are described in Section 4.2 is maintained to achieve this goal. The fields (original location and current location) of the data structure indicate the mapping information. When a read request or a rewrite request arrives, the system will examine the corresponding object. If the original location and the current location are equal, the request will go to the original location. Otherwise, the request will go to the current location. When a write request arrives, if the destination address is occupied by the migrated objects, the data will be written to an available location which is close to the original location, and the information will be recorded by the corresponding data structure.

4.4. Energy conservation

As discussed in Section 2.1, though the timeout strategy does not save energy when the disk is waiting for timeout to expire, the policy is very straightforward and provides very good accuracy. We employed the simple timeout strategy to compare the energy consumption of the proposed EED drive architecture and a traditional disk drive. To save energy, a disk is spun down to the standby state if it is idle for a specified period of time. It can be spun up later to the active state for serving a request. Obviously, the longer the disk remains idle, the more energy can be conserved.

5. Experimental evaluation

Disksim [1], a trace driven simulator, is augmented to measure energy consumption and performance. We integrated four modules including non-volatile memory, frequency monitor, data migration, and energy saving into the simulator to validate the proposed EED architecture. The parameters and characteristics of the energy saving module are listed in Table 1, which are extracted from [2]. The simulated disk specification is based on a Quantum Atlas 10k disk with 10,025 rpm, 10,042 cylinders and 6 heads. We employed the parameters of Samsung NAND flash memory (K9F6408U0A) [36] to perform the simulation. The page size (main data area with 512 Bytes) of K9F6408U0A is equal to the sector size of the disk drives. The Block size is (8k + 256) Bytes. We adopted the random page read of 10 μs and the page program time of 200 μs to perform the read and write tests, respectively. For simplicity, we replaced the overhead of garbage collection with a penalty. The penalty is calculated as: penalty = (page size/block size) × (page program time).

5.1. Data access pattern

Our experiment employed three real traces including Cello99, Cello96, and TPC-D [44] to explore the impact of the EED on energy consumption and performance. Cello99 trace contains modern workloads which were collected in 1999. Cello96 trace was collected in 1996. The traces include accesses to 8 and 20 disks from multiple users and miscellaneous applications [23]. TPC-D is an Oracle trace of decision support processes collected in 1997. The three traces were collected from storage

Table 1
Main characteristics of power, energy, and time statistics

Parameter	IBM 40GNX	Flash memory
Power (Active)	3.0 W	0.03 W
Power (Idle)	0.82 W	N/A
Power (Standby)	0.25 W	N/A
Energy (Spin Down)	0.4 J	N/A
Energy (Spin Up)	8.7 J	N/A
Time (Spin Down)	0.5 s	N/A
Time (Spin Up)	3.5 s	N/A

systems consisting of multiple disk drives. Each line of a trace contains disk ID information. According to the disk ID, we extracted the requests which go to a specific disk and reconstructed three new traces. Because the storage capacity of disk drives used in Cello96 and TPC-D is smaller than the storage capacity of the disk model used in our experiment, we scaled the request address to fit the traces in the Quantum Atlas 10k disk model. In our experiments, we only used parts of the traces and modified the trace format to meet the requirements of Disksim. Table 2 illustrates the characteristics of the three traces used in our experiment, where the skews of 100/3.5, 96/10, 45/10 indicate that 100%, 96%, and 45% I/Os accumulate in 3.5%, 10%, and 10% storage space, respectively. As discussed in Section 2.2, the skew is recursive. Therefore, though the statistics in this section are based on a 10 GB disk drive (Quantum Atlas 10k), we believe that the results can be applied to disk drives which have much larger capacity (e.g. 500 GB). The inter-arrival time denotes the time between the arrival of the first request and the arrival of the next request.

Fig. 2 shows a data access pattern of Cello99 (part of Cello99) across a Quantum Atlas 10k disk. Fig. 2(a) illustrates that only a small number of cylinders are accessed with a high frequency (very skewed). It demonstrates a significant temporal locality. Fig. 2(b) shows the data access pattern after a data migration which moves the frequently accessed data from the magnetic disk media to the flash memory. Fig. 2(b) also shows that most of the data accesses are accumulated within a relatively small and specific area of the disk drive, which indicates that the data migration enhances the spatial locality. In our experiment, we will validate that grouping the most frequently accessed data blocks into the flash memory can save energy and improve performance.

5.2. The optimal object size

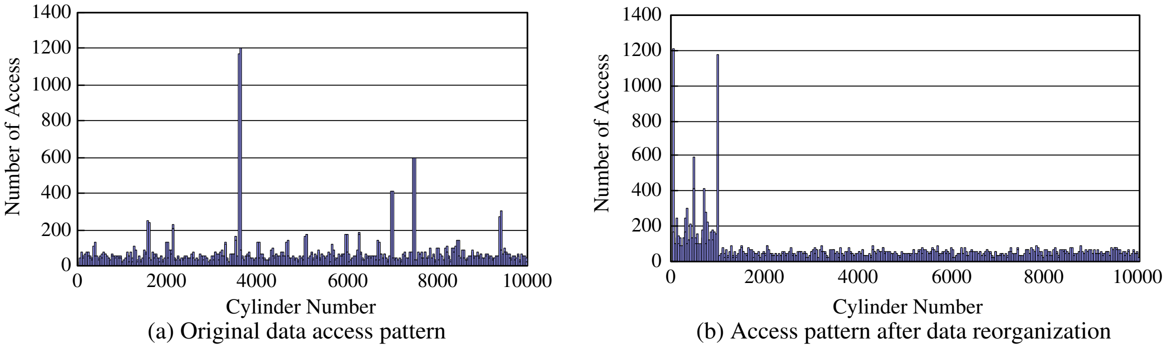
As discussed in Section 4.2, we adopted objects, which are groups of consecutive blocks on the disk, to reorganize the data blocks. We have tested the effects of moving objects with different object size (16, 32, 64, 128, and 256 blocks, respectively). According to the experimental results illustrated in Fig. 3, the general trend is that smaller objects perform better. However, it would incur more overhead to manage the objects with their increase in quantity. On the other hand, due to the small size of objects, the block correlations could be destroyed, and some sequential I/O accesses could be converted to random I/O accesses which normally incur significant performance penalties. In the case of very small objects, a single I/O may have to be split into two (consider a group of blocks requested by an I/O that spans an object boundary before moving). We found through the experiment that around 1.5% I/O accesses are split with an object size of 16 blocks. Based on the above analysis and the test results illustrated in Fig. 3, we employed 32 blocks as the optimal object size in the following tests.

5.3. Evaluating energy saving

The EED drive architecture is proposed to save energy by extending the length of disk idle phases and forcing transitions into the standby state. The idle time is the time between the end and the beginning of two consecutive busy periods within

Table 2
Characteristics of the three traces

Trace name	Cello99	Cello96	TPC-D
Request number	268426	354675	103395
Read percentage	46.12%	46.40%	98.28%
Average request size (KB)	8.025	7.75	56.65
Skew	100/3.5	96/10	45/10
Average Inter-arrival time (ms)	320.6	247.89	205.42



(a) Original data access pattern (b) Access pattern after data reorganization

Fig. 2. Data access pattern comparison.

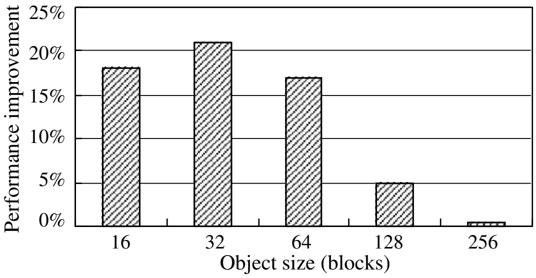


Fig. 3. Performance impact of differing object size.

the disk drives. Table 3 illustrates the average idle time of both the original disk and the proposed EED with different real traces. It shows that due to the proposed EED architecture, the average idle time of three different traces are extended by 3400%, 19.2%, 17.4%, respectively. The extended average idle time of Cello99 is the highest one among the three traces due to the highest skew, illustrated in Table 2. Figs. 4–6 show a slice (201 requests of the overall trace) of the idle time of the original disk and the EE disk with three different traces. The figures demonstrate that the idle times are generally increased due to the proposed EE disk architecture. We believe that longer average idle time indicates lower energy consumption.

An important feature of EED is that most of the data accesses can be satisfied with the flash memory, which enables the magnetic disks to remain in the low power state much longer, thus saving energy. Furthermore, because the data access of flash memory is much faster than that of a traditional disk drive, the EED can improve performance as well. Fig. 7 shows how many requests access the magnetic disk with three traces. In this test, we did not apply the timeout policy. It illustrates that 77% of requests from Cello99, 90% of requests from Cello96, and 74% of requests from TPC-D go the magnetic disk of the original disk drive. The other requests are absorbed by the disk cache. The EED reduces the disk accesses to 3%, 13%, and 41%, respectively. The test results indicate that the EED can significantly decrease the number of disk accesses, thus saving energy and improving performance.

Figs. 8–10 show the energy consumption of the original disk and the EED with different traces and timeout thresholds. The rightmost bar labelled with NP in each figure indicates the baseline energy consumption of the Quantum Atlas 10k disk drive, which does not adopt any energy management policy. The figures confirm, through the observation of the three unique traces, that the proposed EED architecture can indeed save energy.

Fig. 11 shows the conserved energy of EED and the original disk with three different traces and differing timeout thresholds in comparison with the baseline energy consumption of the original disk drive. For both the EED and the original disk, the trend is that the amount of energy saved decreases with the growth of the threshold. The reason is that the disk spends more time in the active state with the increase of the threshold. For the original disk, Cello99 achieves energy use reduction

Table 3
Average idle time of the original disk and the EE disk with three different traces (ms)

	Cello99	Cello96	TPC-D
Original disk	317.6453	241.6937	190.8757
EE disk	11104.48	288.0996	224.1312

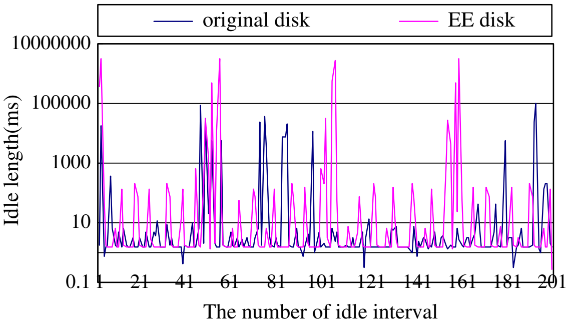


Fig. 4. Idle time of Cello99.

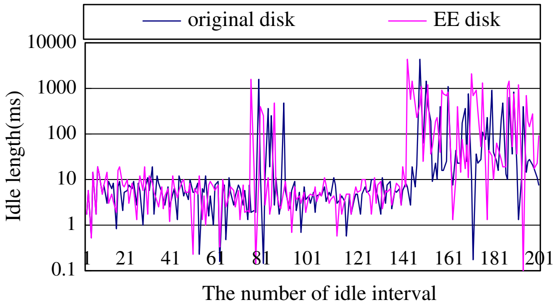


Fig. 5. Idle time of Cello96.

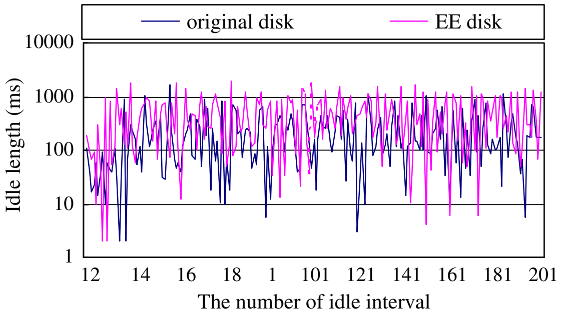


Fig. 6. Idle time of TPC-D.

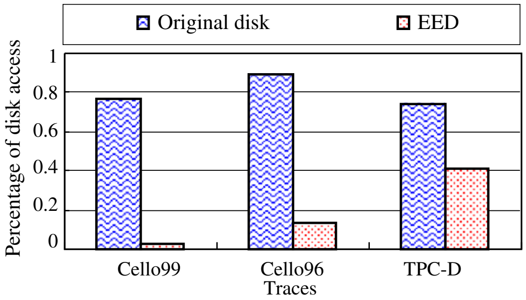


Fig. 7. Disk accesses of the original disk and the EED with three different traces.

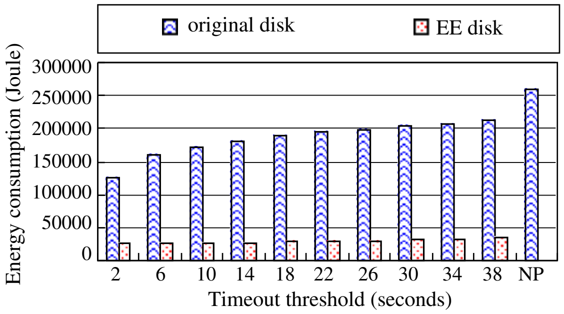


Fig. 8. Energy consumption with Cello99 trace.

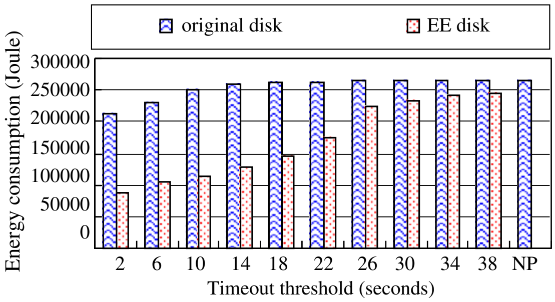


Fig. 9. Energy consumption with Cello96 trace.

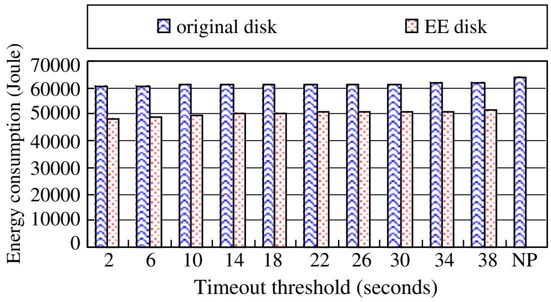


Fig. 10. Energy consumption with TPC-D trace.

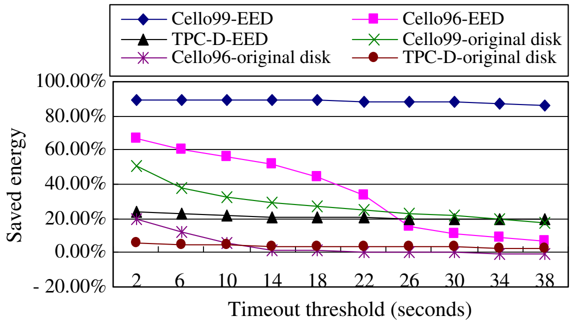


Fig. 11. Saved energy of the EED and the original disk with three different traces and differing timeout thresholds.

ranging from 17.72% to 51.09%. When the timeout threshold is 2 s, Cello96 obtains a 19.32% energy use reduction. However, the timeout policy begins to incur energy loss when the threshold reaches 26 s. TPC-D achieves negligible energy saving ranging from 2.57% to 5.37%. For EED, the saved energy of Cello99 and TPC-D are relatively stable. The reason is that both Cello99 and TPC-D have a relatively stable idle time (see Figs. 4 and 6). However, due to the longest average idle time, Cello99 results in a significant amount of conserved energy ranging from 86.27% to 89.63%. The saved energy of TPC-D only ranges from 19.26% to 23.96%. In contrast to Cello99 and TPC-D, the amount of energy conserved with Cello96 is decreased dramatically (from 67.05% to 7.19%) with the growth of the thresholds. This is because Cello96 has a dramatic variation in the length of idle time, illustrated in Fig. 5. The above measurements validate that the proposed EED architecture can save significant amounts of energy on account of the extended idle times of the disk.

The above experimental results also show that the highest energy use reduction of TPC-D is only 23.96%. This is mainly caused by three factors. The first one is that TPC-D is collected from a decision support process in which most of the data accesses are highly sequential. The second one is because the skew of TPC-D is much lower in comparison with that of Cello99 and Cello96. The third one is that the block correlations of TPC-D are broken to a certain degree, which can produce more I/O accesses going to the magnetic disk drive, thus incurring greater energy consumption. This is because the object size (32 KB) used to move the frequently accessed data blocks is smaller than the average request size of TPC-D. We believe an intelligent correlation detector could alleviate the impact of the object size. According to Table 2, the TPC-D has a much higher ratio of read/write. Because most of the frequently accessed data blocks in EED are moved to the flash memory, if a data stream is skewed (which, as discussed in Section 2.2, most of the I/O accesses are), we believe that there will be a direct correlation between the ratio of reads to writes and energy conservation. The reason is that most of the I/O accesses (read) are absorbed by the flash memory, thus keeping the magnetic disk in a low power state for an extended period of time.

5.4. Evaluating performance

Figs. 12–14 show the average response times of the original disk and the EED with different traces and thresholds. The rightmost bar labelled with NP in each figure indicates the baseline performance of the original disk drive, which does not employ any energy management policy. The *Y* axis of Figs. 12 and 13 are in logarithmic scale. Figs. 12 and 14 illustrate that for the same threshold, the EED shows improved performance (reducing the average response time) when compared with the original disk. The reason is that most of the frequently accessed requests can be served with the fast flash memory instead of the slow magnetic disk. Therefore, the time penalty of spin up and spin down is much smaller than the performance gains, due to the fast access of flash memory. It is very interesting to observe that Fig. 13 depicts different performance behaviours in comparison with Figs. 12 and 14. It shows a significant performance degradation when the thresholds are 10, 14, and 18 s. As explained in Section 5.3, the Cello96 exhibits dramatic variation in the length of idle time (see Fig. 5). For a certain thresholds, such as 10 s, the performance gain of fast flash memory access can not compensate for the performance penalty of the employed energy management strategy.

Fig. 15 describes the performance variation of the EED in comparison with the baseline performance of the original disk. The negative values denote performance degradation. The positive values indicate performance improvement. Fig. 15 validates that for three traces, the average response time is decreased with the growth of the threshold. It also shows that if the threshold is too small, it could incur significant performance degradation. For example, the threshold of 2 s results in 692 times performance degradation with Cello96 trace. The reason is that small thresholds produce more repetitions of spin downs and spin ups. Fig. 16 shows that when the threshold is increased from 2 to 38 s, the number of disk spin downs and spin ups is reduced from 525 to 182 for Cello99, from 6452 to 135 for Cello96, and from 527 to 27 for TPC-D. Please note that the *Y* axis of Fig. 16 is in logarithmic scale. The disk drive employed in our experiment has a spin down and spin up latency of 0.5 s and 3.5 s, respectively (see Table 1). The performance penalty incurred by the power management policy is much larger than the performance gains produced by the fast flash memory access. We can observe in Figs. 11 and 15 that the EED can save energy while improving system performance with a reasonable time threshold. For example, if a threshold of 38 s is adopted with Cello99, we can achieve an 86.27% energy use reduction and a 21.27% performance improvement.

As discussed in Section 3, we cannot simply increase the size of the volatile disk cache to improve the disk performance. Hsu and Smith [18] reported that disk cache in the megabyte range is sufficient, and for a very large disk cache, the hit ratio continues to slightly improve as the cache size is increased beyond 4% of the storage used. It indicates that if the disk cache size grows beyond a threshold, the increased cache only achieves a limited contribution to the hit ratio, which results in poor cost-efficiency. The EED employs flash memory as a sort of non-volatile cache to store the frequently accessed data. Therefore, it is important to investigate the impact of the size of the flash memory on the EED performance. We performed some measurements with 32 KB object size by using Cello99. When the size was increased from 2% to 5%, the performance improvement was about 19%. When the size was increased to 10%, the performance still obtained about 2% growth. However, when further increasing the flash memory size to 15% and 20%, we found a negligible performance growth. It seems the results are consistent with Hsu's suggestions. In order to further investigate the impact, we used the TPC-D trace to perform the second round of tests. The experiments showed that significant performance improvement was noted when the size of flash memory was increased from 2% to 5%, 10%, and 20%. We believe that different skews have different impacts on the size of the flash memory, since the skews of Cello99 and TPC-D are 100/3.5 and 45/10, respectively.

5.5. Energy coefficient

Sections 5.3 and 5.4 illustrate that a smaller threshold produces lower energy consumption, while incurring higher average response time. A reasonable threshold should be able to strike a good balance between energy conservation and

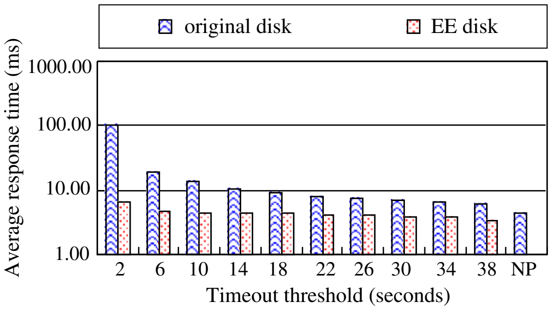


Fig. 12. Average response time with Cello99 trace.

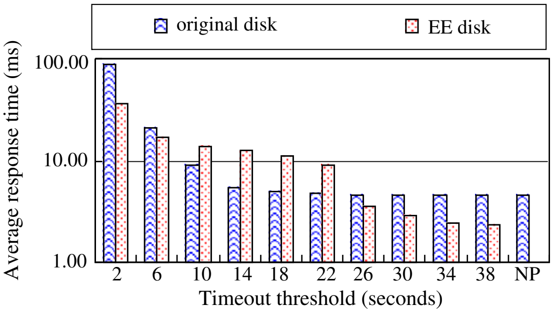


Fig. 13. Average response time with Cello96 trace.

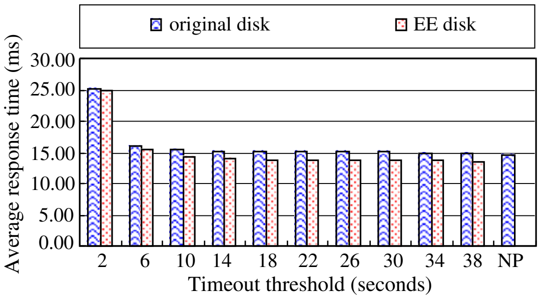


Fig. 14. Average response time with TPC-D trace.

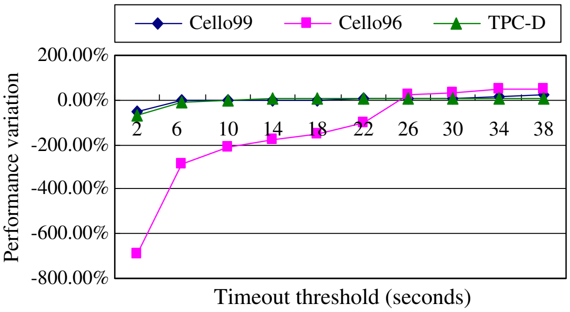


Fig. 15. Performance variation of EED (average response time).

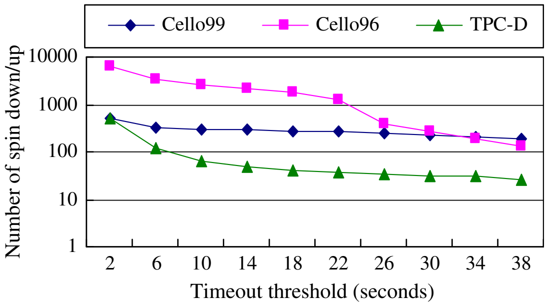


Fig. 16. Number of disk spin downs/spin ups for EED.

performance. Since what we want to achieve is minimal energy consumption accompanied by the lowest average response time possible, we define an energy coefficient as a metric to measure the original disk and the EED.

Definition. The energy coefficient is the product of the average response time and the average energy consumption.

The average energy consumption is the overall energy consumption divided by the number of requests. According to the above definition, the minimal energy coefficient indicates the optimal threshold. Table 4 shows the energy coefficient of the original disk and the EED. Due to the proposed energy coefficient, we can easily identify that the optimal thresholds of cello99, cello96, and TPC-D are 18 s, 38 s, and 26 s, respectively. The energy coefficient listed in Table 4 also demonstrates the behaviours of energy saving and performance discussed in Sections 5.3 and 5.4.

6. Discussion and conclusion

In this paper, we proposed a disk drive architecture which takes a significant step towards an energy efficient storage device by integrating flash memory into a traditional disk drive and moving the frequently accessed data from the magnetic disk media to the flash memory. The storage spaces of the flash memory and the magnetic storage media in the EED are exclusive. From the file system's point of view, they exist in the same linear space. This method is more cost-efficient than a traditional non-volatile second level cache. A power management strategy and a module of flash memory are integrated into the Disksim simulator to validate the proposed EED architecture. Real trace simulations show that the EED can save significant amounts of energy while improving performance with an appropriate threshold. An energy coefficient is proposed to measure the energy consumption and performance of the EED. The energy coefficient provides useful insights into the behaviours of the proposed EE disk.

In our experiments, the NAND flash memory used in EED is K9F6408U0A, which consumes about 0.03 W. The reason we chose this flash memory is because its page size (512 Bytes) is equal to the sector size of a traditional disk drive. We believe that the number of flash chips integrated in EED has an impact on its energy consumption, but these effects are minor. The flash memory consumes much less energy than a traditional disk drive. For example, the latest NAND flash memory chip (K9K8G08U0A), which provides 8 GB of storage space, consumes only 0.08 W [37]. This indicates that the energy consumed by the IBM 40GNX (see Table 1) can support thirty seven K9K8G08U0A flash memory. Please note that the IBM 40GNX is a laptop EIDE disk drive which consumes much less energy than a server disk drive.

We employed NAND flash memory as the non-volatile memory in the EED. A very important feature of NAND flash is that the pages cannot be rewritten. When a portion of data on a page is modified, the new version of the data must be written to an available page somewhere, and the old version is invalidated. When the storage capacity becomes low, garbage collection is triggered to recycle the invalidated pages. Because erasing is performed in blocks, the valid pages in the recycled blocks have to be copied to somewhere before erasing the blocks. Another important feature of the NAND flash memory is the endurance cycles. A block will wear-out after a specified number of program/erase cycles. A poor garbage collection policy could quickly wear out a block and a flash memory chip. The reader is referred to [3] for a comprehensive understanding of the NAND flash memory. Due to the above features, the NAND flash memory is unlikely to be used in high-end storage systems such as servers, though it can be deployed in laptops and desktops. The emerging storage media such as MRAM and MEMS discussed in Section 1 are not on the market currently. However, they may be good candidates for EED due to the features of non-volatility, low energy consumption, unlimited life span, etc.

The EED differs from a Solid State Drive (SSD). The traditional term solid state refers to semiconductor devices. Therefore, an SSD indicates the use of semiconductors to emulate a hard disk drive. SSD commonly consists of either DRAM volatile memory or NAND flash non-volatile memory. The DRAM based SSD requires an internal battery and backup disk drive to guarantee data persistence. This is why most of the current SSDs employ non-volatile flash memory as the storage media (e.g. USB memory sticks). SSD consumes much less energy than a traditional disk drive. Samsung announced a 32 GB SSD consisting of 16 2 GB NAND flash chips. The product is supposed to replace the mini laptop hard drives. However, it costs around 960$ to purchase the 32 GB SSD [38]. The price and storage capacity are the major hurdles for the customers to widely adopt the SSD. In contrast to SSD, the EED integrates a relatively small non-volatile flash memory into a traditional disk drive, which strikes a good balance among the storage capacity, price, performance, and power consumption.

Table 4
Energy coefficient of the original disk and the EE disk with different traces

Threshold (s)	Cello99		Cello96		TPC-D	
	Original disk	EE disk	Original disk	EE disk	Original disk	EE disk
2	48.1764	0.6673	53.3735	8.7239	14.7426	11.7142
6	11.6824	0.4592	13.5032	5.0300	9.4503	7.2626
10	8.5392	0.4543	6.2472	4.5202	9.1055	6.8826
14	7.0327	0.4477	3.8876	4.5709	9.0212	6.8021
18	6.2592	***0.4473***	3.6663	4.6035	8.9782	6.7272
22	5.7924	0.4511	3.4705	4.4976	8.9415	6.7155
26	5.4157	0.4515	3.3856	2.2577	8.9415	***6.7082***
30	5.1990	0.4571	3.3856	1.9100	8.9415	6.7281
34	4.9419	0.4530	3.3750	1.6623	8.9295	6.7302
38	4.7286	0.4533	3.3728	***1.6067***	8.9164	6.7377

Furthermore, the EED can provide a much longer life span than an SSD. Because the NAND flash memory has a limited program cycle, the EED focuses on read operations when employing the flash memory to provide storage space for the frequently accessed data, thus reducing the number of program/erase calls and extending the life span of the memory.

Due to the explosive growth of digital information, a large-scale IT infrastructure can involve millions of components. For example, one of the significant advances in cluster networks over the past several years has been that it is now practical to connect tens of thousands of nodes with networks that have massively scalable capacity. With the growth of the system scale, hardware component failures are becoming a big challenge to deal with [9]. Jiang et al. [19] analyzed the storage logs collected from about 39,000 storage systems commercially deployed at various customer sites and reported that disk failures contribute to 20–55% of storage subsystem failures. The data set covers a period of 44 months and includes about 1,800,000 disks hosted in about 155,000 storage shelf enclosures. Schroeder and Gibson [39] collected and analyzed seven data sets which vary in duration from one month to five years and cover in total a population of more than 100,000 drives from at least four different vendors. Their investigation shows that annual disk replacement rates typically exceed 1%, with 2–4% common and up to 13% observed on some systems. Media error and high temperature are two persistent causes of disk failure. For the EED, most of the data accesses can be performed by the flash memory, which reduces the number of physical accesses to the magnetic disk media and decreases thermal dissipation. Therefore, the EED can alleviate such disk failures. Furthermore, due to the hot swap facilities, it is easy to replace or upgrade hard disk drives without interrupting the system operation. Based on the above discussion, we believe that exploration into and design of an Energy Efficient Disk drive architecture has wide-ranged benefits.

Compressing the data can increase the effective size of the volatile disk cache, non-volatile flash memory, and magnetic disk. Compression allows for more data to be stored in a cache or flash memory of a given size. This improves the hit ratio of both the cache and the flash memory, and results in less disk drive accesses. Compression of the I/O data also increases the quantity of data that can be transferred per time unit. For a large volume of data, latency can be reduced by means of increased bandwidth, even in the presence of compression. Based on the above discussion, we believe that using compression in the EDD can further conserve energy. Possible directions for future work include an intelligent energy management policy, capable of predicting threshold and block correlations on the fly.

Acknowledgements

We would like to thank the anonymous reviewers for helping us refine this paper. Their constructive comments and suggestions were very helpful. The author wishes to express thanks to his friend Brandon Pung from the Massachusetts Institute of Technology, who polished the language of this article and offered a number of constructive comments. In addition, we are grateful to Prof. Witold Pedrycz for giving us the opportunity to clarify our thoughts.

References

[1] J.S. Bucy, G.R. Ganger, The DiskSim simulation environment version 3.0 reference manual, Technical Report CMU-CS-03-102, January 2003.
[2] E.V. Carrera, E. Pinheiro, R. Bianchini, Conserving disk energy in network servers, in: Proceedings of the 17th International Conference on Supercomputing, June 2003, pp. 86–97.
[3] L. Chang, T. Kuo, Efficient management for large-scale flash-memory storage systems with resource conservation, ACM Transactions on Storage 1 (4) (2005) 381–418.
[4] J. Chase, R. Doyle, Balance of energy: energy management for server clusters, in: Proceedings of the 8th Workshop on Hot Topics in Operating Systems (HotOS), May 2001.
[5] L. Cherkasova, M. Gupta, Analysis of enterprise media server workloads: access patterns, locality, content evolution, and rates of change, IEEE/ACM Transactions on Networking 12 (5) (2004) 781–794.
[6] L. Cherkasova, G. Ciardo, Characterizing temporal locality and its impact on web server performance, Technical Report HPL-2000-82, Hewlett Packard Laboratories, July 2000.
[7] D. Colarelli, D. Grunwald, Massive arrays of idle disks for storage archives, in: Proceedings of the 2002 ACM/IEEE Conference on Supercomputing, 2002, pp. 1–11.
[8] F. Douglis, P. Krishnan, B. Marsh, Thwarting the energy-hungry disk, in: Proceedings of the Winter USENIX Conference, 1994, pp. 292–306.
[9] Y. Deng, RISC: a resilient interconnection network for scalable cluster storage systems, Journal of Systems Architecture 54 (1–2) (2008) 70–80.
[10] X. Fan, W. Weber, L.A. Barroso, Power provisioning for a warehouse-sized computer, in: Proceedings of the 34th Annual International Symposium on Computer Architecture, June 2007, pp. 13–23.
[11] G.R. Ganger, B.L. Worthington, R.Y. Hou, Y.N. Patt, Disk subsystem load balancing: disk striping vs. conventional data placement, in: Proceedings of the Hawaii International Conference on System Sciences, January 1993, pp. 40–49.
[12] C. Gniady, A.R. Butt, Y.C. Hu, Y. Lu, Program counter-based prediction techniques for dynamic energy management, IEEE Transactions on Computers 55 (6) (2006) 641–658.
[13] M.E. Gomez, V. Santonja, Characterizing temporal locality in I/O workload, in: Proceedings of the 2002 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'02), 2002.
[14] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, H. Franke, Reducing disk energy consumption in servers with DRPM, Computer 36 (12) (2003) 59–66.
[15] S. Gurumurthi, Should disks be speed demons or brainiacs?, ACM SIGOPS Operating Systems Review 41 (1) (2007) 33–36
[16] D.P. Helmbold, D.D.E. Long, T.L. Sconyers, B. Sherrod, Adaptive disk spin-down for mobile computers, Mobile Networks and Applications 15 (4) (2000) 285–297.
[17] C. Hsu, C. Chen, Y. Su, Hierarchical clustering of mixed data based on distance hierarchy, Information Sciences 177 (20) (2007) 4474–4492.
[18] W.W. Hsu, A.J. Smith, The performance impact of I/O optimizations and disk improvements, IBM Journal of Research and Development 48 (2) (2004) 255–289.
[19] W. Jiang, C. Hu, Y. Zhou, A. Kanevsky, Are disks the dominant contributor for storage failures? a comprehensive study of storage subsystem failure characteristics, in: Proceedings of the 6th USENIX Conference on File and Storage Technologies, 2008.
[20] R. Karedla, J.S. Love, B.G. Wherry, Caching strategies to improve disk system performance, Computer 27 (3) (1994) 38–46.

[21] G. Lawton, Improved flash memory grows in popularity, IEEE Computer 39 (1) (2006) 16–18.
[22] A.J.T. Lee, C. Wang, An efficient algorithm for mining frequent inter-transaction patterns, Information Sciences 177 (17) (2007) 3453–3476.
[23] Z. Li, Z. Chen, Y. Zhou, Mining block correlations to improve storage performance, ACM Transactions on Storage 1 (2) (2005) 213–245.
[24] K. Li, R. Kumpf, P. Horton, T.E. Anderson, Quantitative analysis of disk drive energy management in portable computers, in: Proceedings of the USENIX Winter Conference, 1994, pp. 279–291.
[25] D. Li, J. Wang, EERAID: energy-efficient redundant and inexpensive disk array, in: Proceedings of the 11th ACM SIGOPS European Workshop, September 2004.
[26] Y. Lu, G.D. Micheli, Adaptive hard disk energy management on personal computers, in: Proceedings of the IEEE Great Lakes Symposium, March 1999, pp. 50–53.
[27] Y. Lu, E. Chung, T. Simunic, L. Benini, G. Micheli, Quantitative comparison of power management algorithms, in: Proceedings of Design Automation and Test in Europe, March 2000.
[28] C.R. Lumb, J. Schindler, G.R. Ganger, Freeblock scheduling outside of disk firmware, in: Proceedings of the 1st USENIX Conference on File and Storage Technologies, 2002, pp. 275–288.
[29] F. Moore, More energy needed, Energy User News November 25, 2002.
[30] A.E. Papathanasiou, M.L. Scott, Energy efficient prefetching and caching, in: Proceedings of the USENIX Annual Technical Conference, 2004.
[31] C. Park, J. Seo, D. Seo, S. Kim, B. Kim, Cost-efficient memory architecture design of NAND flash memory embedded systems, in: Proceedings of the 21st International Conference on Computer Design, 2003, pp. 474–480.
[32] E. Pinheiro, R. Bianchini, Energy conservation techniques for disk array-based servers, in: Proceedings of the 18th International Conference on Supercomputing, June 2004, pp. 68–78.
[33] Power Heat, and Sledgehammer. White paper, maximum Throughput, Inc., April 2002.
[34] E. Riedel, G. Gibson, C. Faloutsos, Active storage for large-scale data mining and multimedia, in: Proceedings of the International Conference on Very Large Data Bases (VLDB), 1998, pp. 62–73.
[35] C. Ruemmler, J. Wilkes, Unix disk access patterns, in: Proceedings of 1993 Winter Usenix Conference, 1993, pp. 405–420.
[36] Samsung NAND Flash Memory, K9F6408U0A-TCB0 Datasheet.
[37] Samsung NAND Flash Memory, K9K8G08U0A Datasheet.
[38] Samsung unveils 32 GB Flash hard drive, http://news.zdnet.co.uk/hardware/.
[39] B. Schroeder, G.A. Gibson, Disk failures in the real world: What does an MTTF of 1,000,000 h mean to you?, in: Proceedings of the 5th USENIX Conference on File and Storage Technologies, 2007.
[40] S.W. Schlosser, J.L. Griffin, D.F. Nagle, G.R. Ganger, Designing computer systems with MEMS-based storage, in: Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), 2000, pp. 1– 12.
[41] A.J. Smith, Disk cache—miss ratio analysis and design considerations, ACM Transactions on Computer Systems 3 (3) (1985) 161–203.
[42] S.W. Son, G. Chen, M. Kandemir, Disk layout optimization for reducing energy consumption, in: Proceedings of the 19th International Conference on Supercomputing, 2005, pp. 274–283.
[43] C. Staelin, H. Garcia-Molina, Clustering active disk data to improve disk performance, Tech. Rep. CSTR-283-90, Department of Computer Science, Princeton University, 1990.
[44] Storage Systems Program HP Laboratories, http://tesla.hpl.hp.com/public_software/.
[45] C.K. Subramanian, T.W. Andre, J.J. Nahas, B.J. Garni, H.S. Lin, A. Omair, J.W.L. Martino, Design aspects of a 4 Mbit 0.18 μm 1T1MTJ toggle MRAM memory, in: Proceedings of the IEEE International Conference on Integrated Circuit Design and Technology, 2004, pp. 177–181.
[46] J. Zedlewski, S. Sobti, N. Garg, F. Zheng, A. Krishnamurthy, R. Wang, Modeling hard-disk energy consumption, in: Proceedings of the 2nd USENIX Conference on File and Storage Technology (FAST03), 2003, pp. 217–230.
[47] Q. Zhu, Z. Chen, L. Tan, Y. Zhou, K. Keeton, J. Wilkes, Hibernator: helping disk arrays sleep through the winter, in: Proceedings of the 20th ACM Symposium on Operating Systems Principles 2005 (SOSP 2005), 2005, pp. 177–190.