Efficient Data Migration to Conserve Energy in Streaming Media Storage Systems

Yunpeng Chai, Zhihui Du, *Member*, *IEEE*, David A. Bader, *Fellow*, *IEEE*, and
Xiao Qin, *Senior Member*, *IEEE*

Abstract—Reducing energy consumption has been an important design issue for large-scale streaming media storage systems. Existing energy conservation techniques are inadequate to achieve high energy efficiency for streaming media computing environments due to high data migration overhead. To address this problem, we propose in this paper a new energy-efficient method called Explicit Energy Saving Disk Cooling or EESDC. EESDC significantly reduces data migration overhead because of two reasons. First, a set of disks referred to Explicit Energy Saving Disks (EESD) is *explicitly* fixed according to temporal system load. Second, all the migrated data in EESDC directly contribute on extending the idle time of EESD to conserve more energy efficiently. Therefore, the EESDC method is conducive to saving more energy by quickly achieving energy-efficient data layouts without unnecessary data migrations. We implement EESDC in a simulated disk system, which is validated against a prototype system powered by our EESDC. Our experimental results using both real-world traces and synthetic traces show that EESDC can save up to 28.13-29.33 percent energy consumption for typical streaming media traces. Energy efficiency of streaming media storage systems can be improved by 3.3-6.0 times when EESDC is coupled.

Index Terms—Energy conservation, data layout, streaming media, data migration, storage

1 Introduction

The energy efficiency has already been a major issue in the development of large-scale data centers. In 2008, 39.6TWh of electricity were needed to power data centers throughout Western Europe and the energy expenses were as high as €4.9 billion [1]. The energy consumption problem becomes even more critical now with a high up to 60 annual growth rate of storage system capacity [2]. In a disk-dominated storage system, disks may account for as much as 86 percent of the electricity cost in the entire system [13]. For the popular streaming media applications, disks are likely to dominate power consumption of storage systems due to large data storage capacities and heavy I/O load conditions. Therefore, improving energy efficiency of streaming media storage systems is a critical issue addressed in this study.

Although flash-based storage devices have excellent performance and low-power consumption, they have not been widely adopted due to high price and poor endurance for erasing operations, which is inevitable for rewriting. Compared with flash, multispeed disks have not even been a commercialized product due to the expensive and complex fabrications. In addition, the cache-based energy-saving solutions are inadequate for achieving good performance in the realm of streaming media systems because of the extremely wide capacity gap between disks and memory cache. Energy-efficient data layout algorithms are the most effective approach to conserve energy for streaming media storage systems built with conventional disks.

The effects of energy-efficient data layout algorithms depend on the following two key points:

- Unbalanced data layouts can keep more disks idle as long as possible to spin down for energy savings.
- Fast and efficient data migrations to achieve the unbalanced data layouts.

Motivation. The primary problem of existing energy-efficient disk data layout algorithms applied in streaming media systems is two-fold. First, data migration overhead of existing algorithms is too high when facing the huge data volume of streaming media. Large-scale data migrations are very time-consuming, thereby making it inefficient for existing techniques to conserve energy. Second, another negative factor that slows down the data migration process is that the I/O bandwidth reserved for data migrations is very limited, because most of the disk bandwidth is contributed to provide streaming media data for huge amounts of online users to guarantee high QoS requirements. Therefore, we address this problem by proposing efficient data migration schemes to substantially reduce data migration overhead.

Table 1 shows the simulated data migration overhead of *Popular Data Concentration* scheme or PDC [4]—a representative energy-efficient data layout algorithm. PDC features

- *Y. Chai is with the Key Laboratory of Data Engineering and Knowledge Engineering of Ministry of Education, School of Information, Renmin University of China, Beijing 100872, China. E-mail: ypchai@ruc.edu.cn.*
- *Z. Du is with the Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China. E-mail: duzh@tsinghua.edu.cn.*
- *D.A. Bader is with the College of Computing, Georgia Institute of Technology, North Avenue, Atlanta, GA 30332. Email: bader@cc.gatech.edu.*
- *X. Qin is with the Shelby Center for Engineering Technology, Department of Computer Science and Software Engineering, Samuel Ginn College of Engineering, Auburn University, Suite 3101E, Auburn, AL 36849-5347. E-mail: xqin@auburn.edu.*

Manuscript received 13 Mar. 2011; revised 1 Oct. 2011; accepted 3 Feb. 2012; published online 8 Feb. 2012.
Recommended for acceptance by Y. Hu.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2011-03-0150.
Digital Object Identifier no. 10.1109/TPDS.2012.63.

1045-9219/12/$31.00 © 2012 IEEE Published by the IEEE Computer Society

TABLE 1
Comparison of Data Migration Overhead in 24H

	PDC	EESDC
Migrated Block Num	81,436	7,340
Migrated Size	372.78 GB	33.60 GB
Ideal Migrating Time	22.3 minutes	2.0 minutes
Extra Consumed Power	902,853 J	81,377 J

distributing the data on a disk array by data popularity in descending order to save energy. We simulated a 10-disk PDC system with a 24h I/O trace from the real-world CCTV VOD (video-on-demand) system. Table 1 gives the number of migrated blocks, and the migrated data size measured in the simulation. In addition, assuming all the 10 disks are only used to transfer the migrated blocks at the same time without serving users, the migration time and energy caused by data migrations are also listed. The results indicate that PDC has to migrate 81,346 blocks of 372.78 GB in order to conserve energy for the VOD system. It ideally takes 22.3 minutes to migrate the data blocks, thereby consuming extra 902,853 J energy (see Section 5.5.1 for detailed experiments). However, the practical migration time will be on the order of hours filling most of a day, because complete parallelism among disks is impossible and most bandwidth must be reserved by serving users. We observed from this experiment that the time and energy overhead of PDC is very high for VOD systems.

Basic idea. The low data migration efficiency of traditional energy-efficient data layout algorithms weakens their energy saving effects in streaming media environment. The reason is their *implicit* way of energy conservation. They does not necessarily guarantee that which disks can be switched into the standby mode to save energy, so data migrations that cannot lead to save energy are unnecessary and causes the low efficiency. Specially, in MAID, copying the hot contents in the disks, which will not switch to standby mode, into the cache disks is unnecessary for energy saving. In PDC, the data migration is arranged on the whole disk array to form a specific data layout. Because the layout of the data with the lowest popularity is the key for energy saving, most data migrations in PDC that occur on the data with high or medium-level popularity contribute little to energy savings.

In this paper, we have proposed a new energy-efficient data layout algorithm called Explicit Energy Saving Disk Cooling (EESDC). EESDC first explicitly selects some appropriate disks as the Explicit Energy Saving Disks (EESD) according to temporal system load. Next, EESDC exchanges the “hot” data (i.e., data that are likely to be accessed) in EESD with the “cold” data in the non-EESD disks. In doing so, EESD can be cooled down by the virtue of data migrations to achieve high energy efficiency. EESDC can reduce data migration overhead significantly by explicitly confirming EESD and then focusing the data migrations only on a handful of important “hot” data on EESD that preventing EESD from spinning down.

Our trace-driven simulations under real-world traces show that EESDC outperforms other energy-efficient data layout algorithms, and saves up to 28.13-29.33 percent energy, which is 3.3-6.0 times of PDC, for typical practical streaming media workloads. Such an improvement in energy saving is possible because data migrations caused by EESDC are reduced by more than one order of magnitude compared with PDC (see Table 1). Moreover, practical hardware experiments validate that the simulation results are accurate and convincing.

This paper is organized as follows: in Section 2 we briefly describe the background and the related work. Section 3 gives an overview of the explicit energy saving disk cooling technique or EESDC. The implementation details of EESDC are presented in Section 4. In Section 5, we develop a simulator to evaluate EESDC by comparing it with other energy-efficient data layout algorithms. In addition, we present in this section a validation of the simulator to demonstrate the accuracy of our simulation studies. Finally, Section 6 concludes this paper with the discussion of future work.

2 Background and Related Work

2.1 Streaming Media and Its Characteristics

In streaming media applications, users can enjoy videos or audios as they are being downloaded to computers. For streaming media servers, disk bandwidth is usually the performance bottleneck, because a large number of online users who need video data from server every second bring a mass of small-block random disk accesses. Generally speaking, streaming media services are considered as a data-intensive application. However, compared with traditional data-intensive applications, streaming media systems have their own special features:

- **High QoS constraints.** The Quality of Service (QoS) of streaming media applications is usually measured in startup delay and jitter rather than response time (see Section 5.2.2 for details). Moreover, the QoS constraints are much stringent than most online and all offline applications; low QoS services often mean unacceptable services for streaming media users. High bandwidth must be preserved to serve user requests to guarantee high QoS, so there is very limited bandwidth dedicated for data migrations among disks. In other words, *streaming media application has weak tolerance for high data migration overheads*.
- **Large and fast-growing storage demand.** In the world’s newly produced data each year, multimedia data, especially videos, are forming the largest category. Videos are accumulated extremely fast. For example, video materials about 12.6 million hours are published on YouTube each year [24]. The number of accumulated videos in Facebook grows up as high as 239 percent per year [8]. Such a large and fast-growing storage demand makes *streaming media application extraordinarily relies on highly cost-effective storage devices, i.e., cheap and large-capacity SATA disks* instead of expensive modern storage devices like flash and multispeed disks.

The storage capacity required by media libraries is tremendously large compared with storage space offered by the main memory. Another negative factor is that the streaming media users dynamically change their interests all the time (e.g., Yu et al. [16] discovered that after 1 hour, the changing rate of the top 200 videos may be high up to 60 percent). Such a high rate of changing and the limited capacity of memory compared with disks *make the cache hit rate very low.*

In addition, among such a large data volume of streaming media, a considerable part of videos are seldom or never accessed after being uploaded to servers [25]. Evidence indicates that *the primary data set of streaming media itself has great potential and requirements to save energy, even without relying on redundant storage.*

- **Distinctive behavior of streaming media users.** *The daily workload is drastically fluctuant for streaming media users. In a considerable part of daily time, the load is very high* (for example, there are a numerous of requests in the afternoon and evening, but few requests in the early morning [16]).

  In addition, *streaming media users usually abort watching much before the end of videos.* Yu et al. [16] pointed out that 37.44 percent users leaves in the first 5 minutes of watching, and the majority of partial sessions (52.55 percent) are terminated by users within the first 10 minutes. Therefore, the accuracy of prefetching streaming media data will be low in the prefetching-based energy saving algorithms [30] because of the numerous unpredicted early aborted users. In fact, much of the prefetched data may be useless at all, with the results of wasting much storage and transmission resources in streaming media applications.

2.2 Disk Power Management

Conventional disks spin at full speed regardless of the *active* or *idle* modes. The disks can completely stop spinning when they are placed in the *standby* mode to save energy without being able to serve any request.

The goal of disk power management (DPM) schemes is to conserve energy by turning disks to the low-power mode without adversely affecting I/O performance.

The FT scheme. The Fixed Threshold scheme or FT is the most popular DPM for conventional disks [4]. FT places a disk in the low-power mode after a fixed threshold time has elapsed since the last I/O request. The threshold is usually set to the *break-even* time-defined as the smallest period for which a disk must stay in the low-power mode to offset the extra energy spent in spinning the disk down and up.

2.3 Energy Conservation Techniques

2.3.1 Energy-Efficient Data Layouts

Colarelli and Grunwald [3] proposed the Massive Arrays of Idle Disks (MAID) scheme that uses extra cache disks to cache recently visited data. MAID directs some I/O requests onto cache disks, thereby can place some noncache disks with light load in the low-power mode. However, MAID's energy efficiency highly relies on a high cache hit rate of cache disks, because MAID makes no optimization on any data distribution for noncache disks. If the cache hit rate declines, the performance of MAID is similar to the FT scheme. In addition, extra cache disks in MAID introduce power consumption overheads. Therefore, energy savings offered by MAID is limited.

Popular Data Concentration [4] migrates data among disks according to data popularity in a descending order fixed by Multi Queue (MQ) algorithm [29]. All the disks' load distributes in a descending order. This makes the disks storing low popularity content gain greater opportunities to stay in the low-power mode. Unlike MAID, PDC relies on data migrations rather than copying data. Thus, PDC not only optimizes disk data distribution for energy savings, but also avoids extra cache disks. The experimental results show that PDC outperforms MAID in terms of energy savings [4]. However, the data migration overheads caused by PDC in streaming media systems are too high, making the systems unlikely to finish streaming requests in time. The high migration overheads are inevitable because PDC makes uniform and stringent data layouts, which significantly weakens PDC's energy y-saving effect for streaming media applications.

To illustrate the differences among FT, MAID, PDC, and our proposed EESDC, we compare the disk I/O load distributions imposed by the four schemes in Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2012.63.

2.3.2 Other Energy Saving Approaches

Kgil and Mudge [27], Kgil et al. [28] proposed an energy-efficient architecture that relies on NAND flash memory to reduce the power of main memory and disk in web server platforms. The flash memory not only extremely outperforms disks, but also is much more energy efficient than disks. However, flash-memory-enabled storage systems are not widely adopted due to high price and weak endurance for erasing operations.

Multispeed disks are proposed to spin at different lower speeds to provide more opportunities to save energy. The shifting overhead between two rotational speeds of a multispeed disk is relatively smaller than that of power-state transitions between the active and standby mode in a conventional disk. Carrera et al. [11] and Gurumurthi et al. [12], respectively, investigated methods to determine disks' rotation speed. Zhu et al. [13] proposed Hibernator, whose key idea is to set the rotational speed of all disks to appropriate values according to the system load, and to distribute data blocks on different disk tiers according to the access possibility of the data blocks. Unfortunately, it is uncertain that multispeed disk products can be widely deployed in the not-too-distant future due to their prohibitive manufacturing cost.

Zhu and Zhou [10] studied approaches to conserve energy by caching parts of data in memory in order to create opportunities for some disks to be switched into or kept in low-power mode. Manzanares et al. developed an energy-efficient prefetching algorithm or PRE-BUD for parallel I/O systems with buffer disks [30]. Prefetching data in buffer disks provide ample opportunities to increase idle periods in data disks, thereby facilitating long standby times of disks. However, when they come to streaming media applications, it is very hard to cache or prefetch

TABLE 2
Comparisons of Energy-Efficient Approaches on the Adaptability of Streaming Media's Features

Energy-Efficient Approaches	High QoS Constraints	Large Volume of Data			Special Behavior	
	Weak tolerance for data migrations	Low cost	Low cache hit rate	Original data set	Fluctuating and high load	Sessions' early terminations
MAID [3]	×	√	×	√	×	√
PDC [4]	×	√	√	√	×	√
Flash-based energy-efficient storage [27, 28]	√	×	×	√	√	√
Multi-speed disks [11, 12, 13]	√	×	√	√	√	√
Energy-efficient caching [10]	√	√	×	√	√	√
Energy-efficient prefetching [30]	√	√	√	√	√	×
Energy-efficient redundant disks [5, 6, 7, 9, 14, 15]	√	√	√	×	√	√
Our EESDC proposed in this paper	√	√	√	√	√	√

accurate to-be-accessed data. Consequently, energy savings offered by those cache replacement/prefetching algorithms are very insignificant (See Appendix G.1, available in the online supplemental material, for detailed experiments).

In addition, some energy-efficient data layout algorithms like EERAID [5], DIV [6], eRAID [7], and PARAID [9] were developed to conserve energy for redundant disk arrays. Recently some research work [14], [15] focuses on power-proportional storage, which aims at providing a low minimum power, a high maximum performance, and fast, fine-grained scaling for redundant storage. Although these energy saving methods for redundant storage can be integrated together with our method to improve energy saving effects further, our EESDC approach proposed in this paper mainly focuses on saving energy for original data sets in streaming media applications.

2.3.3 *Comparison of Energy Conservation Techniques*

Table 2 summarizes a comparison between an array of existing energy-efficient approaches and our proposed EESDC tailored for the adaptability of streaming media's features. In this table, "×" means the approach has a conflict with one of the streaming media features, whereas "√" indicates that there is no conflict. According to Table 2, all the existing energy-efficient algorithms designed for general-purpose storage systems have conflicting items, while EESDC fits streaming media very well. Appendix B, available in the online supplemental material, gives a detailed explanation of this table.

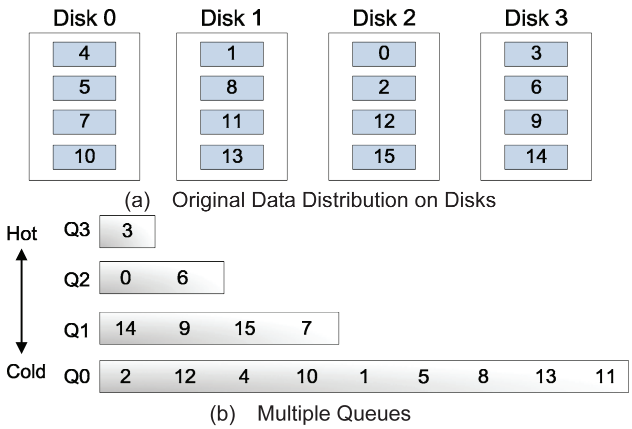
3 Overview of EESDC

3.1 A Motivational Example

First of all, we use an example to demonstrate the problem of existing methods. In this section, let us take a close look at an example of PDC. In this example shown as Fig. 1, 16 blocks (0-15) are randomly distributed on four disks (see Fig. 1a). Fig. 1b illustrates the block distributions made by the MQ algorithm. A new data distribution decision generated by PDC is given as: Block 3, 0, 6, 14 on Disk 0; Block 9, 15, 7, 2 on Disk 1; Block 12, 4, 10, 1 on Disk 2; and Block 5, 8, 13, 11 on Disk 3. In this case, 15 blocks are stored on different disks in the new distribution compared with the original one. In other words, most of the stored blocks may be involved in the data migrations according to PDC in many cases.

In streaming media systems which have weak tolerance for data migration overheads, such low-data migration efficiency cannot lead to good energy saving effects. However, what is the reason of such low-data migration efficiency?

The reason lies in that existing energy conservation techniques like MAID and PDC save energy in an *implicit* way, which means that energy saving is just their implicit hidden goal achieved by pursuing the explicit goal of caching hot data in cache disks (MAID) or skewed data layouts on all disks (PDC). In other words, those algorithms cannot determine which disks are involved in the energy saving goal, thus much of the data migration determined by them is not related to the disks which are going to standby, but only causes data blocks traveling among some active disks. For example, MAID may cause many unnecessary additional read operations on the disks that will not go into standby; PDC may force data migrations occur on disks with high or medium-level load. These unnecessary data migrations may account for a considerable part of all the



(a) Original Data Distribution on Disks

(b) Multiple Queues

Fig. 1. An example of data migration overhead in PDC.

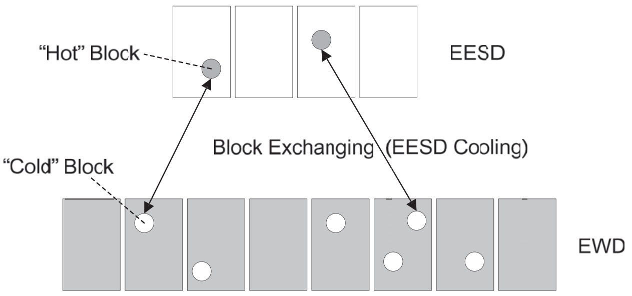


Fig. 2. Explicit energy saving disk cooling (EESDC)—an explicit goal-driven approach.

data migrations. Therefore, MAID and PDC involve too many inefficient data migrations, most of which are unnecessary or contribute little to energy savings.

Specially, the traditional implicit goal-driven energy-saving approaches have the following limitations:

- Since data migrations must involve all disks in a system, excessive data movement overheads become inevitable. Data migrations caused by disks that have few opportunities to conserve energy are unnecessary and adverse for energy efficiency, because the unnecessary data migrations can waste I/O bandwidth resource of streaming media systems.
- The implicit goal-driven approaches only indirectly control disks' load. In case that the I/O loads of a majority of disks in a streaming media system are high, these approaches are unable to efficiently save energy.

3.2 Principles of EESDC

In EESDC, disks that lead to energy savings are called Explicit Energy Saving Disks or EESD; non-EESD disks are called Explicit Working Disks (EWD). EESDC explicitly controls I/O loads of EESD, making EESD stay in the low-power state as long as possible while keeping EWD disks active all the time to serve user requests. In our approach, the number of EESD and EWD varies with dynamic workload conditions. When system load is high, an increasing number of disks will perform as EWD to provide high I/O bandwidth. When the system load is low, many disks will be marked as EESD to conserve energy.

Fig. 2 illustrates an example in which EESDC directly exchanges "hot" blocks in EESD with "cold" blocks in EWD. Note that "hot" blocks have high accessed possibilities; "cold" blocks have low accessed chances. Such a data exchanging procedure among the EESD and EWD disks are referred to as the "EESD cooling" technique, which cools down EESD disks by always making streaming media contents on EESD less likely to be accessed.

Choosing EESD disks—a nontrivial procedure—affects energy efficiency of streaming media systems powered by EESDC. Please refer to Section 4.2 for implementation details on the procedure of selecting EESD disks. Section 4.3 will discuss the implementation details of the block exchanging between EESD and EWD disks.

Unlike the implicit goal-driven energy-saving algorithms, our EESDC—an *explicit* goal-driven approach—explicitly identifies EESD disks that can save energy based on temporal system loads and keeps EESD disks at low-power state longer by very limited data migrations.

3.3 Benefits of EESDC

A streaming media system can achieve the following two benefits from the explicit goal-driven EESDC approach:

- Only hot data blocks residing in EESD disks are eligible for data migrations to save energy. In this way, our approach can significantly reduce data migration overheads.
- The unbalanced data distribution controlled by EESDC is driven only by the explicit goal of energy conservation. Our method which directly manages the I/O load of EESD disks can offer more energy savings.

3.3.1 Reducing Data Migration Overheads

When it comes to EESDC, it is possible to reduce data migration overheads because of the following three reasons:

1. Unlike PDC that maintains stringent data distributions on all disks, EESDC allows flexible and coarse-grained data distributions. Data blocks are not required to be distributed on a particular disk; rather, data blocks can be allocated to any disks inside EWD or EESD. In general, it is practical to place hot data on most EWDs, or to place cold data on any EESD. For example, if a block's popularity is promoted only a bit, the block may have to be migrated to an adjacent disk based on PDC's stringent data distribution policy. In this case, however, EESDC does not cause any data migration. Thus, the flexible and coarse-grained data distribution scheme enables EESDC to significantly reduce the amount of migrated data.
2. Because EESDC has explicit energy-saving target disks, and uses a disk-oriented way to trigger data migration activities, only identified hot blocks in EESD are eligible for data migrations to directly conserve energy. Unnecessary and inefficient data migrations are eliminated by EESDC.
3. EESDC dynamically adjusts the amount of data migrated between EESD and EWD based on temporal system workloads. When the system loads are light, EESD contains more disks and the amount of migrated data may increases because it will keep more disks to save energy. Under high system loads, the number of EESD disks is small and then only small amount of related data migrations are needed. Hence, unnecessary data migrations are eliminated. On the contrary, the implicit goal-driven PDC always treats data migrations in the same way regardless of dynamic system loads.

To cut data migration overhead cost, our EESDC scheme dramatically reduces the total amount of data migrated. In order to compare with PDC's data migration efficiency, we use the same example shown in Fig. 1. For a fair comparison, we assume that access rates of the blocks are in the same order as those used in the case of the MQ algorithm. According to EESDC, one disk should be chosen as an EESD disk under the given system load; EESDC selects Disk 1 as EESD because data blocks on Disk 1 are the coldest. In this case, only Block 1 on Disk 1 should be exchanged with another cold block. This example intuitively illustrates data migration overhead reduction offered by EESDC. Please see Section 5.5 for detailed results on data migrations.

3.3.2 *Directly Controlling the I/O Load of EESD*

Except for reducing data migration overhead significantly, another benefit of EESDC is to manage the I/O load of EESD disks directly. Thus, the load of EESD controlled by EESDC tends to be very low, which means better energy saving efficiency.

A disk load comparison between EESDC and other approaches is presented in Appendix A, available in the online supplemental material. It illustrates the EESDC's advantage of efficiently and directly controlling EESD's load in a very low level. In addition, Appendix C, available in the online supplemental material, gives a detailed quantitative analysis to verify that low disk load of EESD leads to high energy saving efficiency.

4 Implementation Details

To implement the EESDC approach in a streaming media storage system, we must address the following four important issues:

- How to quantitatively measure the temperatures of streaming media blocks? Such measures are important because we need a way of identifying hot blocks.
- How to dynamically select EESD disks according to a streaming media system's load? EESD selections directly affect energy efficiency and QoS of the system.
- How to identify data blocks to be migrated? How to efficiently perform data migrations with low overheads? A good implementation of the data migration module in EESDC can quickly create energy-efficient data layouts with little cost.
- How to balance the load among the disks in EWD to avoid overloading in some disks?

4.1 Workflow and Overview of the Four Modules

Fig. 3 shows EESDC's workflow that contains four main modules, namely, a block processing module or BP, a dynamical EESD selecting module or DES, a data migration module (i.e., DMES) for energy savings, and a data migration module for assistant load balance or DMALB. These modules are implemented to address the aforementioned four issues. BP (see Section 4.2) is responsible for measuring blocks' temperature and sorting blocks in the increasing order of temperature. DES (see Section 4.3) takes charge of dynamically selecting EESD disks. DMES (see Section 4.4) migrates data blocks to cool down EESD disks and at the same time to balance I/O load among EWD disks as much as possible. DMALB (see Section 4.5) aims to further improve I/O performance by balancing load of the EWD disks.

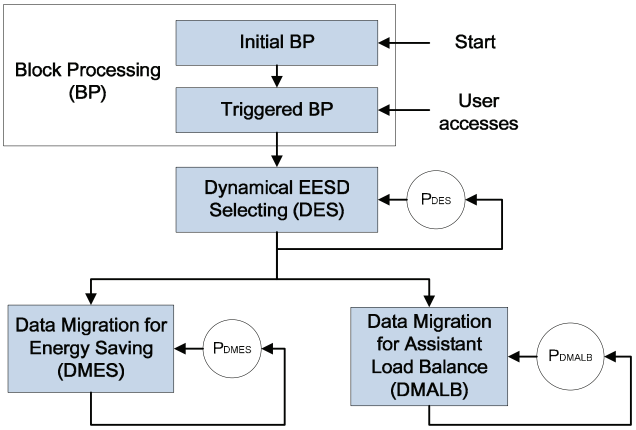

Block Processing (BP)
Initial BP
Start
Triggered BP
User accesses
Dynamical EESD Selecting (DES)
P_{DES}
Data Migration for Energy Saving (DMES)
P_{DMES}
Data Migration for Assistant Load Balance (DMALB)
P_{DMALB}

Fig. 3. Workflow of the EESDC mechanism.

The BP module contains two parts. The first part is executed in the initialization phase; the second part—triggered by user requests—updates block temperatures and maintains a sorted block list.

The DES, DMES, DMALB modules are all periodically driven. Noteworthy is that the period of DMES—P_{DMES}—should be set as a very small value to make DMES adapt to the constant changing of user interests in streaming media systems.

4.2 Block Processing (BP)

4.2.1 *Data Splitting and Distributions*

To balance load among all disks in streaming media storage systems, large streaming media files are usually split into multiple data blocks. Block sizes are normally between hundreds of KB to tens of MB. These blocks are generally distributed on all disks in a disk array by algorithms like JBOD [17] or Random [26]. In this BP module, we choose the most commonly used algorithm—the random distribution algorithm. The file splitting and block distributions are handled in the initial phase of BP.

We will show in Section 5.5.2 that block size can affect energy efficiency and QoS of energy-saving mechanisms.

4.2.2 *Temperatures of Data Blocks*

Data-block temperatures are calculated using (1), where AT_i is the total access time of Block i within a past fixed observation time window, $Size_i$ is the size of Block i, and δ is a prefix weight value. The temperature of Block i, i.e., Tp_i is proportional to AT_i and inversely proportional to $Size_i$. δ is a value larger than 1 to increase the temperature of the current block if Block i is the video prefix; otherwise it is equal to 1. Because most users watch videos from the very beginning, and the users often quit after watching the early part of a video [16], the higher temperature of a prefix block makes it easier to keep the block in EWD to improve QoS and to avoid waking some EESD disks up. A large temperature value of a block indicates that the bock is hot

$$Tp_i = \delta \cdot AT_i / Size_i. \tag{1}$$

Noteworthy is that the past fixed observation time window is usually set as a small value (e.g., 30 minutes) in EESDC, because user interests are constantly changing in streaming-media applications [16]. In addition, a small time window can reduce the overhead of recording and tracking user visits.

4.2.3 Data Block Sorting Based on Temperatures

Both the DES and DMES modules need a data block list sorted in an ascending order of temperature; Triggered BP maintains such a sorted list for DES and DMES.

To develop an efficient block-sorting procedure, we implement the Heap Sort Algorithm in the BP module. Except the initial sorting phase, the block-sorting procedure is very quick because the majority of operations are updating the sorted list (i.e., removing blocks with updated temperatures from the list; inserting the removed blocks to the list again). The time complexity of the implemented block-sorting procedure is log(M), where M is the total number of data blocks.

In EESDC, because the number of data blocks is much larger than the number of disks, time spent in sorting all blocks dominates the time complexity. If there are λ new user requests in a system, λ is the upper bound of the number of temperature changed blocks. Therefore, the time complexity of EESDC is $\lambda \cdot \log(M)$, where M is the number of all data blocks, and λ is much smaller than M for a typical streaming media storage system.

The block-sorting procedure in BP will not slow down system performance because of the follows reasons:

- Some user accesses may not trigger the block-sorting procedure, since the blocks in EWDs trigger the sorting procedure after accumulating a certain number of accesses. Only accesses to blocks in the EESD disks will immediately invoke the block-sorting procedure.
- The sizes of streaming media blocks are usually large; M is not a very large value in a typical system. Thus, the complexity of *log*(M) is reasonable.
- When it comes to a large-scale storage system, we can simply divide the system into a set of small subsystems, each of which has small number of storage nodes and disks. Each small-scale subsystem can be independently controlled to save energy. Therefore, M is not too large for individual subsystems.

4.3 Dynamical EESD Selecting (DES)

The overall goal of the DES module is to choose appropriate disks to serve as EESD disks to boost energy efficiency while ensuring EWD disks can offer good QoS to users with least data migrations. Too few EESDs provide very limited energy-saving opportunities. In contrast, a too large number of EESDs results in a small number of EWDs serving users and exchanging data with EESDs, thereby making low energy efficiency and poor QoS.

4.3.1 Two Phases in DES

DES is executed in two phases. The first step determines the number N_{EESD} of EESD disks and the number N_{EWD} of EWD disks according to temporal block temperature distributions and the estimated system load in the near future. The second step chooses disks to perform as EESDs and EWDs.

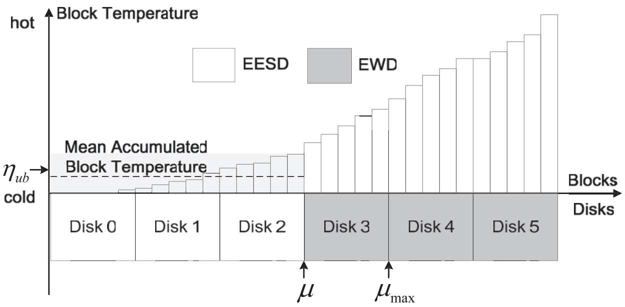


Fig. 4. Dynamical EESD selecting or DES in EESDC.

In fact, after the first phase is done, the second phase can be easily implemented as follows: first, mean temperature of all the blocks in each disk is calculated. Next, disks are sorted in an ascending order of the mean temperature. Finally, the first N_{EESD} disks are set as EESD and the rest of disks are EWD.

Therefore, a key issue in the DES module is to determine an appropriate value of μ—the proportion of disks serving as EESDs. The proportion of disks performing as EWD is consequently $(1 - \mu)$. Using ideal block temperature distributions on disks, DES can improve energy efficiency of streaming media systems. Fig. 4 gives an example of dynamic EESD selecting process. In this example, blocks are sorted in an increasing order of temperature. Blocks with the lowest temperature should be stored in EESD disks and the other ones should be residing in EWD disks. The portion μ should be set as large as possible to maximize energy savings under the following two conditions:

- The number N_{EWD} of EWD disks must be large enough to offer good performance (i.e., good QoS).
- The mean load of EESD disks must be low enough to have sufficient idle periods to conserve energy.

The two conditions guide us to use (2) to set the value of μ, which suggests that the number of EWD disks must be at least $(1 - \mu_{max}) \cdot N$ to supply sufficient bandwidth

$$\max\ \mu, \quad s.t \begin{cases} \mu \le \mu_{\max} \\ BR_{EESD} \le \eta_{ub}, \end{cases} \tag{2}$$

where BR_{EESD} in (2) is the bandwidth ratio of EESD disks. The bandwidth ratio is defined as a ratio of current used bandwidth to the maximum possible disk bandwidth. η_{ub} is the upper bound of any EESD disk's bandwidth ratio to ensure good energy savings.

Fig. 4 suggests that the temperature of the blocks increases from the origin to the horizontal axis direction. Therefore, a large value of μ leads to a high mean temperature of all the blocks in EESD, i.e., a high ratio BR_{EESD}. Thus, the maximum valid μ makes BR_{EESD} be close to but not larger than η_{ub}.

μ_{max} and η_{ub} are two important parameters in the EESDC approach. Lowering μ_{max} can offer good QoS with reduced energy savings. η_{ub} is an empirical parameter, and should be set to an appropriate value to achieve the best energy efficiency. If η_{ub} is too high, there will be too few EWD disks to provide enough I/O bandwidth. In this case, EESD disks often have to spin up to serve requests, introducing high response delays and overheads of power transitions. If η_{ub} is too low, energy savings will be limited due to a small number of EESD disks.

4.3.2 Calculating μ_{max} and BR_{EESD}

Let m be the total number of data blocks. After data blocks are sorted in an ascending order of temperature, EWD disks should accommodate the last k blocks of all m video blocks in the sorted list whereas EESD disks should store the rest blocks in the list. The total required bandwidth of the EWD disks in a shortcoming period (i.e., B_{EWD}^{est}) can be calculated by (3), where B is the mean video bitrates. The future system load LD^{est} can be estimated using historical access patterns; the statistical visit time of block i is at_i

$$B_{EWD}^{est} = B{\cdot}LD^{est} \cdot \sum_{i=m-k}^{m-1} at_i \Big/ \sum_{i=0}^{m-1} at_i. \tag{3}$$

Let B_{max}^{disk} be the maximum bandwidth of disks. B_{EWD}^{actual}—the bandwidth that an EWD disk can achieve-is calculated as $(1-\mu)N \cdot B_{\max}^{disk}$. Considering the skewed I/O workloads among EWD and EESD disks, we make B_{EWD}^{est} equal to or less than $p \cdot B_{EWD}^{actual}$, where p is a constant in the range (0, 1). Configuring B_{EWD}^{est} in this way can avoid overloaded disks. Thus, μ_{max} can be calculated as (4) below

$$\mu_{max} = 1 - \frac{B_{EWD}^{est}}{p{\cdot}N{\cdot}B_{\max}^{disk}}, p \in (0,1). \tag{4}$$

The total bandwidth provided by EESD disks is $\mu \cdot B_{max}^{disk}$. Then, bandwidth ratio BR_{EESD} can be calculated as

$$BR_{EESD} = \frac{B{\cdot}LD^{est} \cdot \sum_{i=0}^{m-k-1} at_i / \sum_{i=0}^{m-1} at_i}{\mu N{\cdot}B_{\max}^{disk}}. \tag{5}$$

4.3.3 The Implementation of DES

DES is executed periodically (see Fig. 3). In each execution, DES first checks if the number of EESD disks derived from μ is appropriate for temporal conditions (we define it as Phase 0). If current value of μ has been correctly configured, this round of DES will be terminated; otherwise a new value of μ will be calculated in Phase 1 and a new set of EESD disks is decided in Phase 2 (Phases 1 and 2 correspond to the two phases in Section 4.3.1). Please see Appendix D, available in the online supplemental material, for the implementation details of Phase 0, 1, and 2.

4.4 Data Migrations for Energy Saving (DMES)

DMES is responsible for conserving energy through unbalanced data distributions achieved by direct block exchanging between EESD and EWD disks. In other words, disks are cooled down by migrating hot blocks from EESD to EWD disks and by moving cold blocks from EWD to EESD disks.

To implement the DMES module, we need to determine a *boundary temperature* (BOT) between EESD and EWD disks. BOT is an important parameter in DMES used to identify hot and cold blocks. Thus, blocks whose temperatures are higher than or equal to BOT should be placed on EWD disks; blocks whose temperatures are lower than BOT should be stored on EESD disks. In our implementation, BOT is calculated as follows.

DMES receives a sorted data block list SBL, in which blocks are sorted in an increasing order of their temperatures, from the BP module (see Section 4.2). If *len(SBL)* is the length of SBL, the value of BOT can be derived from current μ and *len(SBL)* as the approximate lowest temperature of all the temperatures of data blocks stored in EWD. Thus, BOT can be expressed as (6) below:

$$\begin{aligned} BOT &= Block\,Temperature\ of\ SBL[x], \\ x &= \lceil \mu \times len(SBL) \rceil. \end{aligned} \tag{6}$$

After BOT is determined, DMES can easily identify hot blocks residing in EESD disks. Specifically, all blocks whose temperatures are higher than BOT are hot blocks that may be migrated to EWD disks. Because DMES is repeatedly executed with very short period, DMES chooses the "hottest" data blocks in EESD disks to migrate each time. This mechanism can guarantee high data migration efficiency and quickly adaptively adjust itself to dynamically changing user interests in streaming media applications. In addition, to avoid the phenomenon that some EESDs containing many not-very-hot blocks are always kept away from the choices, we save some chances for EESDs that don't perform data migrations for a long time period larger than a threshold.

In addition, to balance load across EWD disks, DMES migrates hot blocks in EESDs to a target EWD with the lightest load among all the EWDs. This strategy speeds up the data exchanging, and improves load balancing among EWD disks to avoid overload.

4.5 Data Migrations for Assistant Load Balance (DMALB)

Although DMES can balance load among EWD disks to a certain extent by choosing lightly loaded EWDs as targets, DMES cannot deal with some special cases. For example, if one EWD's load suddenly increases, DMES cannot transfer this EWD's load to other EWD disks. On the contrary, the DMALB module directly migrates data blocks among EWD disks, thereby further balancing load among EWDs. DMALB performs the following two strategies:

- **Strategy 1.** If an individual EWD disk's load is higher than a threshold (i.e., Th_{heavy}), DMALB exchanges a hot block on this disk with a relatively cold block stored on another EWD with light load.
- **Strategy 2.** When some disks have extremely heavy loads (i.e., larger than $Th_{too\text{-}heavy}$), these disks should not be involved in the process of load balancing. This is because data migrations occurred on the extremely highly loaded disks can worsen QoS.

5 Performance Evaluation

5.1 Test Bed

We extended the widely used DiskSim simulator [22] by incorporating the file-level interface, file splitting and management, the "push" service manner of streaming media, the DPM algorithms, and a set of energy-efficient data-layout schemes. In our test bed, we simulated a streaming media storage system containing a set of 10 to 50 disks. The specifications for the disks used in our study are similar to

TABLE 3
Disk Parameters

	IBM Ultrastar 36Z15	Seagate Barracuda 7200.12
Interface	SCSI	SATA
Disk Rotation Speed	15000 RPM	7200 RPM
High Power	13.5 W	4.965 W
Low Power	2.5 W	0.816 W
Spinup Time	10.9 sec.	3.794 sec.
Spinup Energy	135 J	63.125 J
Spindown Time	1.5 sec.	0.291 sec.
Spindown Energy	13 J	4.419 J

those of the IBM Ultrastar 36Z15 disks (see [18] for a detailed disk data sheet). Table 3 shows key parameters of the simulated disks.

Four evaluated energy saving schemes are FT (see Section 2.2), PDC (see Sections 2.3), EESDC, and offline EESDC. The offline EESDC scheme is a variant of our EESDC in which accurate user future access patterns are utilized instead of estimating it through the methods in online EESDC. The offline version of EESDC can provide the approximate upper bound of energy savings offered by energy-efficient data layout algorithms.

We used both synthetic and real-world traces to drive our simulator. The synthetic traces are used to test the four schemes under a wide range of workload conditions; the real-world traces are used to evaluate the schemes in the context of practical streaming media applications.

5.1.1 Synthetic Trace Generator

To evaluate EESDC under various workloads, we have developed a synthetic-trace generator denoted as $WG(r(t), l, \alpha)$, where *r(t)* is user arrival rate at time t (i.e., how many users arrive per time unit around time t). The value of r dynamically changes at different time t, since streaming media systems have drastic load fluctuations. l is the mean session length of all users. Two parameters *r(t)* and l determine the number of online users (i.e., system load) in a streaming media system. α—data popularity—is a parameter in the Zipf distribution function expressed in (7) where F is the number of streaming media files

$$p(i, \alpha, F) = 1/i^{1+\alpha} \Big/ \sum_{i=1}^{F} 1/i^{1+\alpha}. \tag{7}$$

5.1.2 Two Real-World Traces

To validate experiments using synthetic traces, we tested the four energy-saving schemes by running the simulator with two 24-hour real-world traces.

The first trace represents a real-world streaming media system—the CCTV VOD (Video-On-Demand) system in Jan. 2005. User arrival rate of this trace is 11.88 per minute; the mean session length of all the users is 188.65 second. We observed from this trace that the CCTV VOD system's load dynamically changes—the number of online users during evenings is much larger than that in early mornings. The second trace represents the real-world UUSee systems [23]. The user arrival rate of this system is 3.55 per minute; the mean session length is 1094.48 second.

The CCTV VOD trace is a representative trace for short-session dominated streaming media applications like YouTube; the UUSee trace is a representative one for systems with many long sessions.

5.2 Performance Metrics

5.2.1 Measuring Energy Saving

Saved energy percentage is a metric to measure energy efficiency. Let E be the energy consumed by a system without using any energy saving technique, *E(A)* be the energy consumed by the same system supported by energy saving algorithm A. The saved energy percentage of algorithm A (i.e., $SE_{abs}(A)$) can be calculated as (8)

$$SE_{abs}(A) = (E - E(A))/E \times 100\%. \tag{8}$$

5.2.2 Measuring QoS

QoS in streaming media systems is quantified by two metrics—*startup delay* and *jitter*. Startup delay measures waiting time for users before video starts; *jitter* reflects variation in delay during playback procedure.

- *Mean startup delay.* Since most streaming media users are impatient, any energy saving schemes must conserve energy without significantly increasing startup delays. In this study, we calculate the Mean Startup Delay or MSD of all users as the first QoS metric.
- *Mean delay jitter.* Jitter measures the variability of delivery times in packet streams [19]. A small jitter (i.e., good QoS) means that users can watch videos fluently. The jitter used in streaming media applications is called delay jitter—a maximum difference in the total delay of different packets. We studied the Mean Delay Jitter or MDJ of all users to as the second QoS metric. Please see Appendix E, available in the online supplemental material, for the details of how to calculate the delay jitter.

5.3 Results of Synthetic Traces

The default system parameters of the synthetic traces are set as: the user arrival rate $= 0.2$ user/second, $l = 200$ second, $\alpha = 0.12$, and video bitrates $= 320$ Kbps.

5.3.1 Impacts of System Load

Fig. 5 shows the impacts of user arrival rate on saved energy percentages of the four schemes. It indicates that the saved energy decreases with the increasing arrival rate for all the methods. A higher user arrival rate—representing high system load—makes it hard for all the energy-saving schemes to conserve energy.

FT is very sensitive to the system load. When the arrival rate is larger than 0.25 user/second, FT's saved energy percentage is less than 2 percent. The reason is that FT does not change data layouts to conserve energy. PDC's energy efficiency is basically as poor as that of FT. PDC is even worse than FT when the user arrival rate is less than 0.15 user/second, because the disks lose some opportunities to stay in the standby mode due to a large number of data migrations arranged by PDC itself.

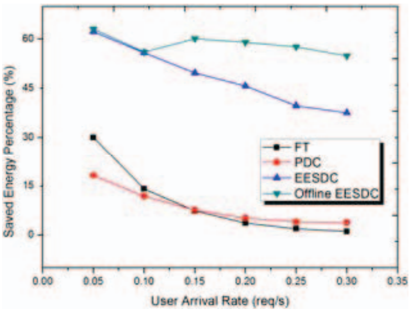


Fig. 5. Saved energy with different loads.

Compared with FT and PDC, EESDC offers a much higher energy-saving percentage (i.e., anywhere between 37.58 and 62.25 percent). Under light system load, energy conserved by EESDC is even very close to that of the offline scheme. Fig. 5 shows that unlike FT and PDC which can only save energy effectively under light load, EESDC can always maintain a high-energy saving rate under various workloads. Of course, the offline EESDC scheme has the highest saved energy percentage, which is between 54.88 and 63.04 percent.

In addition, more experimental results based on synthetic traces, such as the impacts of α in Zipf formula and the impacts of system scale when disk number increases from 10 to 50, are in Appendix F, available in the online supplemental material.

5.4 Results of Real-System Traces

We use two 24-hour real-world traces (see Section 5.1.2 for details on the two traces) to validate results obtained from the synthetic traces. Table 4 gives the saved energy and QoS of the four approaches for, respectively, the CCTV VOD and UUSee traces.

5.4.1 Energy Savings

Table 4 shows that FT always has the worst energy efficiency. PDC is more energy efficient than FT; PDC's saved energy percentage is only 2.2 and 2.6 times higher than that of FT for the CCTV VOD and UUSee traces. The saved energy percentage of EESDC is 13.2 and 6.0 times higher than those of FT and PDC in the case of the CCTV VOD system. When it comes to the UUSee system, EESDC's saved energy is 8.4 and 3.3 times higher than those of FT and PDC. EESDC outperforms FT and PDC because EESDC makes an effort to cut data migration overheads. The offline EESDC scheme, of course, has the best energy efficiency.

Many streaming media users only watch the beginning part of videos and then decide whether not continue watching the videos. Therefore, the beginning parts of video files attract many user accesses. The video number of UUSee is much smaller than that of the CCTV VOD, making the user accesses of UUSee are more skewed and focused than that of CCTV VOD. Thus, the energy saving effects of FT and PDC are better than under CCTV VOD.

TABLE 4
10-Disk System' 24h Overall Results

		FT	PDC	EESDC	Offline EESDC
CCTV VOD	Saved Energy	2.23%	4.91%	29.33%	55.35%
	MSD (ms)	250.5	419.1	497.0	56.1
	MDJ (ms)	246.8	1070.0	554.6	89.4
UUSee	Saved Energy	3.33%	8.54%	28.13%	53.69%
	MSD (ms)	196.3	489.2	393.7	53.1
	MDJ (ms)	1016.0	2588.9	839.1	298.9

MSD: Mean Startup Delay; MDJ: Mean Delay Jitter.

However, the average number of online users—a product of the mean user arrival rate and mean user session length—of UUSee is higher than that of CCTV VOD. Because UUSee has higher load than CCTV VOD, it's harder to achieve a high-energy saving rate in UUSee than in CCTV VOD. The consequence is that EESDC and offline EESDC save a bit less energy than under CCTV VOD trace.

5.4.2 QoS

In the case of the CCTV-VOD system, MSD and MDJ of FT are relatively smaller than those of PDC and EESDC. Because FT introduces no data migrations, FT's delay and jitter are only slowed down by disk power-state transitions. Compared with PDC, EESDC achieves similar startup delay and better delay jitter. Taken saved energy and QoS together, EESDC outperforms PDC significantly.

After simulating the UUSee system, we observed that the delay jitter increases significantly for all the algorithms compared with the CCTV VOD case. The reason is that the user session in UUSee is much longer than CCTV VOD, which significantly increases the possibility of a user encountering a video block stored in EESDs. In EESDC, EWD disks do not need to switch between high- and low-power state, thereby offering better QoS for the streaming media systems. In the PDC scheme, disks with both light and median-high load can be switched into the standby mode. Such power-state transitions in PDC have negative impacts on QoS. Therefore, EESDC can outperform PDC in terms of QoS significantly for UUSee.

5.5 Data Migration Overheads Analysis

5.5.1 The Amount of Migrated Data

The energy efficiencies of PDC and EESDC both rely on data migration overheads. Table 1 outlines the amount of migrated data for a simulated 24-hour CCTV system powered by PDC and EESDC. In what follows, we provide detailed results on data migration overheads.

Table 5 shows the number of the migrated data blocks or NMDB, and the migrated data amount per 1 percent saved energy (MDA per 1 percent SE) for PDC and EESDC in synthetic traces with different user arrival rates, the CCTV-VOD and UUSee traces. We observed that EESDC significantly reduced the migrated data amount by anywhere from 84.01 to 94.31 percent compared with PDC. What's more, the data migration efficiency for energy saving of EESDC is 15-22.1 times higher than that of PDC from the results of MDA per 1 percent SE.

The reduced data migration overheads make it possible for EESDC to achieve high energy efficiency in streaming media systems. All the data migrations made by EESDC contribute to energy conservation without paying much unnecessary data migration overheads.

TABLE 5
The Amount of Data Migration

UAR	PDC		EESDC	
	NMDB	MDA per 1%SE	NMDB	MDA per 1%SE
0.05	34298	2.86 GB	1950	0.143 GB
0.1	33480	4.31 GB	2880	0.236 GB
0.15	32678	6.54 GB	3630	0.334 GB
0.2	32500	9.55 GB	4242	0.424 GB
0.25	31618	11.78 GB	5018	0.578 GB
0.3	32140	12.75 GB	5140	0.626 GB
CCTV	81436	25.31 GB	7340	1.146 GB
UUSee	68640	12.12 GB	4974	0.809 GB

NMDB: the number of migrated data blocks. MDA per 1 percent SE: migrated data amount per 1 percent saved energy percentage. UAR: User arrival rate.

5.5.2 Impacts of Streaming Media Block Size

Block sizes affect disk access patterns and data migrations in streaming media systems, which in turn has impacts on energy savings and QoS. Fig. 6 shows the saved energy percentage and QoS of EESDC when block size ranges from 0.78 to 37.5 MB for CCTV VOD trace.

Fig. 6 indicates that the saved energy first sharply increases and then slightly falls down with the increasing block size. This trend can be explained by the fact that when the block size is large, exchanging hot blocks between EESD and EWD disks can better utilize spatial localities. However, when the block size is too big, the data migration overhead becomes too large to accomplish in a short time and most of the too-far-away data in the block will not be accessed. As a result, energy savings slightly go down. Fig. 6 suggests that the block size offering the best energy saving is approximately 9.38 MB, and the highest saved energy percentage is 30.82 percent.

Fig. 6 also shows that the mean startup delay (MSD) increases when the block size is increasing. The first accessed block in a streaming media may be stored in a standby disk, which gives rise to an increased startup delays. Large blocks lead to skewed workload conditions, which provide EESDC great opportunities to transit disks into the standby mode. Thus, the startup delay increases when the block size goes up.

Another phenomenon is that the mean delay jitter (MDJ) first declines and then ascends with increasing block size. During a playback, any block stored in a standby disk from the EESD set can cause delay jitter. When the block size is very small, a video file contains a great number of blocks. During an entire playback process, there will be a large number of accessed blocks stored in a standby disk. As such, delay jitters are large when block sizes are very small. In contrast, larger block sizes tend to lead to more skewed loads, thereby offering many disks chances to switch into the standby mode. Thus, the delay jitter becomes larger because users have larger possibility to encounter data blocks stored in standby disks when the block size increases from normal size to very large one. The above reasons explain why delay jitter first decreases and then increases when the block size increases.

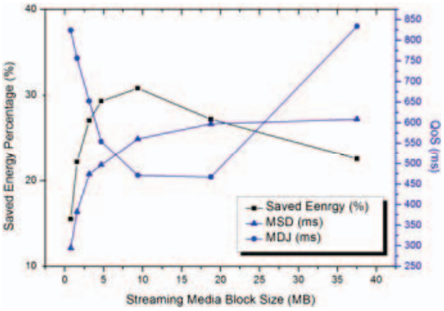


Fig. 6. Saved energy and QoS with different block sizes.

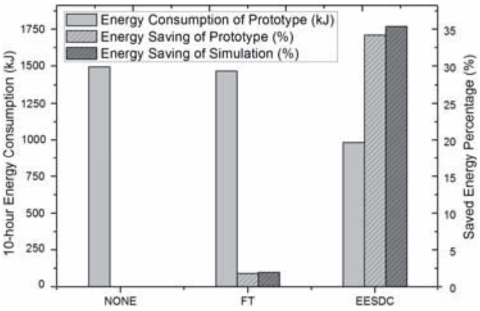


Fig. 7. Comparison experiment results.

In addition, more simulation results of the performance analysis, including the impacts of cache sizes, parameter *BT*, and I/O bandwidth limitation, are presented in Appendix G, available in the online supplemental material.

5.6 A Real Hardware Prototype and Experimental Results

To validate the simulation results, we first develop a prototype system in a real hardware testbed. Then, we measure the energy dissipation of our prototype under a typical real-world streaming media trace, and compare energy savings achieved in the testbed against those observed in the simulator. In our testbed, a computer contains an AMD Athlon II X2 240 Processor 2800 MHz 64 bits, 4 GB DDR2 Kingston memory, and eight disks connected to additional PCI-SATA cards separately powered by an extra power supply. A power meter is used to measure the energy consumption of the eight disks. The parameters of the eight Seagate Barracuda 7200.12 500 GB disks used in the experiments are measured and listed in Table 3.

We validate the accuracy of our simulator by comparing energy-saving results between the prototype system and our simulated system under the same configuration settings and the same typical real-world streaming media trace from CCTV VOD service.

The validation results indicate that for the 10-hour CCTV VOD trace, the prototype system consumes 1494.36 kJ when no energy-efficient algorithm is adopted; the prototype consumes 1467.0 kJ when FT is used and only 983.16 kJ when EESDC is employed. Fig. 7 shows the measured power consumption of prototype system, and energy-saving percentages under both the prototype system and simulated system when no energy-efficient scheme, FT, and EESDC are, respectively, adopted.

The validation results confirm that for both FT and EESDC schemes, energy-efficiency values observed from the simulator and the prototype agree with each other, even though the empirical results are a little bit lower than simulation results. In fact, simulated systems are constructed based on some primary factors in simulation model, but are not able to take all the detailed factors into account. This makes the little difference between our simulated results and prototype results. However, the difference is very small. The simulated results can be considered as accurate ones.

Importantly, Table 4 shows that the 10-hour CCTV VOD trace exhibits more energy savings than the 24-hour CCTV VOD trace replayed in the simulator. This is because the average load of the first 10-hour part is lighter than the whole 24-hour CCTV VOD trace.

In summary, the validation process confirms that EESDC can achieve high energy efficiency in a typical and practical streaming media environment, and the simulation results discussed in Sections 5.3-5.5 and Appendices F-G, available in the online supplemental material, are accurate.

6 Conclusions and Future Work

In this paper, we have presented a novel energy conservation technique called EESDC to save energy in streaming media systems with low data migration overheads. The three main contributions of this study are given as follows: first, we emphasize and quantify the importance of data migration efficiency for energy saving, especially in streaming media storage systems. Second, the explicit goal-driven EESDC both reduces the data migration overheads and improves energy efficiency and QoS. Finally, EESDC significantly improves energy efficiency by adapting itself to meet the characteristics of streaming media applications.

A potential concern about energy saving approaches is their impacts on storage systems' reliability. Recent studies has re-examined some assumptions regarding factors that affect disk lifetime [20], [21]. In the future, we can extend the EESDC approach by making a good tradeoff between energy efficiency and reliability of streaming media systems.

In addition, with the rapid increasing of disk storage's space and constant falling price, storage systems with redundant disks becomes increasingly common. As an energy saving approach for the primary data set, EESDC will be integrated with other redundancy-based energy-saving approaches to further improve energy efficiency of streaming media systems.

Acknowledgments

This work was supported in part by the National Excellent Courses Integration Project (No. JPKC-5), National Nature Science Foundation of China (No. 61073008, 60773148, and No. 60503039), NSF Grants CNS-0614915, CAREER CCF-0611589, an IBM Shared University Research (SUR) award, NSF awards CCF-0845257 (CAREER), CNS-0757778 (CSR), CCF-0742187 (CPA), CNS-0831502 (CyberTrust), OCI-0753305 (CI-TEAM), DUE-0837341 (CCLI), and DUE-0830831 (SFS), and an Intel gift (Number 2005-04-070) as well as an Auburn University startup grant. In addition, we thank Prof. Yuanchun Shi, Prof. Shiqiang Yang, Prof. Lifeng Sun, Xin Xiao, and Meng Zhang for the comments on the simulation experiments and the paper organization, and Peng Gao for the efforts on the prototype system.

References

[1] G. Nebuloni, "Energy Footprint of the European Server Infrastructure, 2008, and 2009-2013 Forecast," http://www.idc.com/getdoc.jsp?containerId=GE11R9, Oct. 2009.

[2] B. Moore, "Taking the Data Center Power and Cooling Challenge," *Energy User News*, Aug. 2002.

[3] D. Colarelli and D. Grunwald, "Massive Arrays of Idle Disks for Storage Archives," *Proc. ACM/IEEE Conf. Supercomputing*, pp. 1-11, Nov. 2002.

[4] E. Pinheiro and R. Bianchini, "Energy Conservation Techniques for Disk Array-Based Servers," *Proc. 18th Int'l Conf. Supercomputing*, pp. 68-78, June 2004.

[5] D. Li, P. Gu, H. Cai, J. Wang, "EERAID: Energy Efficient Redundant and Inexpensive Disk Array," *Proc. 11th workshop ACM SIGOPS European Workshop*, 2004.

[6] E. Pinheiro, R. Bianchini, and C. Dubnicki, "Exploiting Redundancy to Conserve Energy in Storage Systems," *Proc. Joint Int'l Conf. Measurement and Modeling of Computer Systems (SIGMETRICS '06)*, pp. 15-26, June 2006.

[7] J. Wang, H. Zhu, and D. Li, "eRAID: Conserving Energy in Conventional Disk-Based RAID System" *IEEE Trans. Computers*, vol. 57, no. 3, pp. 359-374, Mar. 2008.

[8] J. Yarow, "Videos on Youtube Grew 123 Percent Year over Year, While Facebook Grew 239 Percent," http://www.strangelove.com/blog/2010/06/videos-on-youtube-grew-123-year-over-year-while-facebook-grew-239, June 2010.

[9] C. Weddle, M. Oldham, J. Qian, A.A. Wang, P. Reiher, and G. Kuenning, "PARAID: A Gear-Shifting Power-Aware RAID," *ACM Trans. Storage*, vol. 3, no. 3, pp. 245-260, Oct. 2007.

[10] Q. Zhu and Y. Zhou, "Power-Aware Storage Cache Management," *IEEE Trans. Computers*, vol. 54, no. 5, pp. 587-602, May 2005.

[11] E.V. Carrera, E. Pinheiro, and R. Bianchini, "Conserving Disk Energy in Network Servers," *Proc. 17th Int'l Conf. Supercomputing*, pp. 86-97, June 2003.

[12] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, and H. Franke, "DRPM: Dynamic Speed Control for Power Management in Server Class Disks," *Proc. 30th Ann. Int'l Symp. Computer Architecture*, pp. 169-179, June 2003.

[13] Q. Zhu, Z. Chen, L. Tan, Y. Zhou, K. Keeton, and J. Wilkes, "Hibernator: Helping Disk Arrays Sleep through the Winter," *Proc. ACM Symp. Operating Systems Principles*, pp. 177-190, Oct. 2005.

[14] H. Amur, J. Cipar, and V. Gupta, "Robust and Flexible Power-Proportional Storage," *Proc. First ACM Symp. Cloud Computing*, June 2010.

[15] E. Thereska, A. Donnelly, and D. Narayanan, "Sierra: Practical Power-Proportionality for Data Center Storage," *Proc. Sixth Conf. Computer Systems (EuroSys '11)*, Apr. 2011.

[16] H. Yu, D. Zheng, B. Zhao, and W. Zheng, "Understanding User Behavior in Large-Scale Video-on-Demand Systems," *Proc. First ACM SIGOPS/EuroSys Conf.*, pp. 333-344, Apr. 2006.

[17] J. Gray and P. Kukol, "Sequential Disk IO Tests for GBps Land Speed Record," http://research.microsoft.com/research/pubs/view.aspx?type=Technical%20Report&id=766, 2004.

[18] Hitachi, "Ultrastar 36Z15 Datasheet," http://www.hitachigst.-com/hdd/ultra/ul36z15.htm, Jan. 2003.

[19] Y. Mansour and B. Patt-Sharmir, "Jiiter Control in QoS Networks," *IEEE/ACM Trans. Networking*, vol. 9, no. 4, pp. 492-502, Aug. 2001.

[20] E. Pinheiro, W.D. Weber, and L.A. Barroso, "Failure Trends in a Large Disk Drive Population," *Proc. USENIX Conf. File and Storage Technologies (FAST)*, 2007.

[21] B. Schroeder and G.A. Gibson, "Disk Failures in the Real World: What does An MTTF of 1,000,000 Hours Mean to You?" *Proc. USENIX Conf. File and Storage Technologies (FAST)*, 2007.

[22] G. Ganger, B. Worthington, and Y. Patt, "The DiskSim Simulation Environment (V4.0)," http://www.pdl.cmu.edu/DiskSim, Sept. 2009.

[23] X. Xiao, Y. Shi, Q. Zhang, J. Shen, and Y. Gao, "Toward Systematical Data Scheduling for Layered Streaming in Peer-to-Peer Networks: Can We Go Farther?," *IEEE Trans. Parallel and Distributed Systems*, vol. 21, no. 5, pp.685-697, May 2010.

[24] C. Yates, "Top 10 Video Stats," http://www.huddleproductio-ns.com/?p=1014, June 2011.

[25] X. Cheng, C. Dale, and J. Liu, "Statistics and Social Network of Youtube Videos," *Proc. 16th Int'l Workshop Quality of Service*, pp. 229-238, 2008.

[26] J. Aerts, J. Korst, and S. Egner, "Random Duplicate Storage Strategies for Load, Balancing in Multimedia Servers," *Information Process Letters,* vol. 76, no. 1, pp. 51-59, 2000.

[27] T. Kgil and T. Mudge, "FlashCache: A NAND Flash Memory File Cache for Low Power Web Servers," *Proc. Int'l Conf. Compilers, Architecture and Synthesis for Embedded Systems (CASES '06),* Oct. 2006.

[28] T. Kgil, D. Roberts, and T. Mudge, "Improving NAND Flash Based Disk Caches," *Proc. 35th Ann. Int'l Symp. Computer Architecture (ISCA '08),* June 2008.

[29] Y. Zhou and J.F. Philbin, "The Multi-Queue Replacement Algorithm for Second Level Buffer Caches," *Proc. USENIX Technical Conf.,* pp. 91-104, June 2001.

[30] A. Manzanares, S. Yin, X.-J. Ruan, and X. Qin, "PRE-BUD: Prefetching for Energy-Efficient Parallel I/O Systems with Buffer Disks," *ACM Trans. Storage,* vol. 7, no. 1, article no. 3, 2010.

Yunpeng Chai received the BE and PhD degrees in computer science and technology from Tsinghua University in 2004 and 2009, respectively. Since 2009 he has been working at Renmin University of China as an assistant professor in College of Information. His research interests include cloud storage, streaming media service system, and energy conservation of storage systems.

Zhihui Du received the BE degree in 1992 in Computer Department from Tianjian University. He received the MS and PhD degrees in computer science, respectively, in 1995 and 1998, from Peking University. From 1998 to 2000, he worked at Tsinghua University as a postdoctoral researcher. Since 2001, he is working at Tsinghua University as an associate professor in the Department of Computer Science and Technology. His research areas include high performance computing and grid computing. He is a member of the IEEE.

David A. Bader received the PhD degree in 1996 from the University of Maryland and was awarded a US National Science Foundation (NSF) Postdoctoral Research Associateship in Experimental Computer Science. He is a professor in the School of Computational Science and Engineering and executive director of high performance computing at Georgia Institute of Technology. He is a professor in computational science and engineering, a school within the College of Computing, at the Georgia Institute of Technology. He has coauthored more than 100 articles in journals and conferences, and his main areas of research include parallel algorithms, combinatorial optimization, and computational biology and genomics. He is a fellow of the IEEE and a member of the ACM.

Xiao Qin (S'00-M'04-SM'09) received the BS and MS degrees in computer science from Huazhong University of Science and Technology, Wuhan, China, in 1996 and 1999, respectively, and the PhD degree in computer science from the University of Nebraska, Lincoln, in 2004. He is currently an associate professor of computer science at Auburn University. He won an NSF CAREER Award in 2009. His research interests include parallel and distributed systems, real-time computing, storage systems, fault tolerance, and performance evaluation. He is a senior member of the IEEE and the IEEE Computer Society.