Energy efficient data migration techniques for mobile systems with multiple disks

ABSTRACT

Disks have been the most prevalent secondary storage devices and these days their usage is becoming more important in mobile systems due to I/O intensive applications such as multimedia applications and games. However, significant power consumption in the disk drives still limits battery lifetimes of mobile computing systems critically.

In this paper, we show that using several smaller disks (instead of one large disk) can be an energy-efficient secondary storage solution on typical mobile platforms without a significant performance penalty. Also, we propose a novel energy-efficient technique, which clusters related data into groups and migrates the correlated groups to the same disk. We compare this method with the existing data concentration scheme, and also combine the techniques. The experimental results show that the new technique saves the energy consumption up to 34% when a pair of 1.8″ disks is used instead of a single 2.5″ disk with a negligible increase in the average response time.

I. INTRODUCTION

Mobile and ubiquitous computing platforms are now widely accepted by the end-users, together with a background of network communications and high-performance processing environments. The ownership and use of cell phones, PDAs, PMPs, MP3 players, and other mobile devices are increasing by the year. The demand for disk drives with a small form-factor (2.5″ or less), embedded in or connected to such platforms, is also incrementally rising. At present, the demand for server and desktop PC disks is higher than that for smaller disks, but the two markets may soon become comparable [1].

In general, disk drives are known as significant power consumers in computer systems, and most mobile devices depend on battery power and so it is crucial to control the energy consumption of disk drives in mobiles. Therefore, a lot of effort has been put into reducing the amount of energy consumed in the disk drives.

However, existing energy-efficient techniques mainly have been constrained to the mobile systems with a single disk drive and have not investigated the potential of energy saving when using more than one smaller disk instead of a larger one. Compared with research on mobile storage systems, recently many studies of energy-conserving techniques based on arrays of multiple disks have been fulfilled in server storage systems [2,3,4,5,6]. Although such techniques are targeted to applications in data centers or network servers, the basic idea that distributing I/O accesses across multiple disks properly can lead to energy saving is sufficiently applicable to mobile storage systems.

On the top of such background, if we replace a larger high-power disk (e.g. 2.5″ disk) with a smaller low-power disk (e.g. 1.8″ disk) the energy conservation of the system will be obvious. But such replacement should assure that almost the same throughput and capacity can be guaranteed. To meet these constraints more than one disk may be used, and for this purpose we need to investigate various disks in terms of energy, performance, capacity, and so on. And, it is also important how to distribute I/O accesses well across multiple disks.

In this paper, our goal is to derive substantial energy saving in mobile storage systems with multiple smaller form-factor disks instead of one laptop disk by adopting a new dynamic data placement technique of correlated data groups classified from mobile workloads across disks. To step toward a solution, we first take a survey of which small form-factor disks can be exploited for energy conservation in mobile systems, and then describe the correlation between file requests leading to identifiable disk access patterns in mobile workloads.

A. Characteristics of Small Form-Factor Disks

Table 1 shows a comparison between five state-of-the-art small form-factor disks: the Travelstar 80GN [7], MK4004GAH [8], ST1.3 [9], Microdrive 3K8 [10], and MK4001MTD [11]. These disks have different parameters, such as capacity, speed, power, and size, which can be extracted from manufacturers' documents. The Travelstar 80GN is usually used in laptop computers and the other disks are used in handheld devices such as cell phones and PMPs. From Table 1, we see that the larger disks tend to consume more power but have better performance because their platters rotate more quickly. For instance, the Travelstar 80GN consumes 2.3 W in its active state, which is about 1.6 more than the MK4004GAH, and 3.8 times as much as the MK4001MTD, even when the low-power modes of the smaller disks are not used. In terms of performance, the Travelstar 80GN provides at least 25% less latency than all of the smaller disks, which we have estimated from the speed and average seek time, while average latency can be easily deduced from the disk speed. Consequently, when we consider replacing a larger high-power disk with multiples of smaller lower-power disks, the effects caused by the differences between these parameters on the systems should be considered sufficiently.

Table 1. Characteristics of laptop and smaller form factor disks

Form factor / Model		2.5″	1.8″	1″		0.85″
Parameters		Travelstar 80GN (Hitachi GST)	MK4004GAH (Toshiba)	ST1.3 (Seagate)	Microdrive 3K8 (Hitachi GST)	MK4001MTD (Toshiba)
Capacity (GB)		80	40	12	6, 8	4
Rotational speed (rpm)		4,200	4,200	N/A	3,600	3,600
Avg. seek time (ms)		12	15	N/A	12	16
Power (W)	Active	2.3	1.4	0.792	0.627	0.6
	Idle	0.95	0.4	0.254	N/A	0.45
	Standby	0.25	0.2	0.0429	N/A	0.12
Physical size	Weight (g)	99	62	14	13	8.5
	Area (cm^2)	70	42.39	12	12	7.68

B. Workloads on Mobile Platforms

Compared with the workloads on database or web servers, mobile workloads vary more between applications and also fluctuate due to frequent switchovers between programs. Nevertheless, regular patterns of workload occur in mobile computing environments, including laptop computers, suggesting that common applications are executed frequently and in sequence. For instance, if a laptop computer is taken on a business trip it is likely to be used to receive and answer emails, to edit presentations, or to connect to a remote server to perform file operations, and so on. A similar pattern of usage may appear for persons who usually utilize their laptops instead of desktop PCs, making repetitive use of email programs, tools for programming, document editors, multimedia players, and so on. To characterize the workloads frequently observed in mobile platforms Chen et al. [12] selected nine typical mobile applications and build two workload scenarios, called *programming* and *networking*, which simulate the workloads corresponding to these activities. The programming scenario consists of eight stages and during each stage the traces of rather programming-related applications such as *make* and *grep* are played. The networking scenario is similarly made up of three stages in which networking and multimedia applications such as *thunderbird* (an email client program), *ftp*, and *xmms* (an mp3 player) are replayed.

When people do these sorts of jobs on their laptops, data files are likely to be shared between applications. The shared files are likely to be accessed more frequently than other files. For instance, after searching and finding a file the user may try to edit and then compile it. These patterns, which do not occur in a server, suggest that correlated file requests (i.e. requests executed in the same order for related files within an application, or for files shared between applications) can be used to concentrate the load onto one disk in a multiple-disk storage system, so as to obtain greater energy savings. However, existing low-power load-skewing techniques like PDC [5] are designed for server workloads and are unlikely to yield satisfactory results when applied to mobiles.

Based on the above investigation, we propose a novel energy-saving technique for mobile systems, in which uses multiple small form-factor disks replacing one large disk. After identifying distinctive data access patterns in commonly used mobile platforms we show how they can lead to energy savings through data migration. Simulation results show that our method conserves disk energy considerably while maintaining reasonable performance, and is an improvement on existing low-power techniques used in storage systems based on disk arrays.

The rest of this paper is organized as follows. Section II presents a motivational example and Section III explains energy-conserving file placement techniques that use multiple disks including existing methods, our technique, and their combination. Section IV describes our simulator and presents simulation results. Section V describes related work and Section VI concludes the paper.

II. MOTIVATIONAL EXAMPLE

Let us assume that we replace a 2.5″ disk with multiple smaller form-factor disks for the purpose of energy saving while keeping acceptable performance. From Table 1, the MK4004GAH is the only disk that is comparable to the Travelstar 80GN in terms of both power and performance. Furthermore, MK4004GAH has sufficient capacity for mobile applications, which is not necessarily true of the three remaining disks. When we consider replacing a laptop disk with multiple smaller disks, the MK4004GAH will be a strong candidate because its nominal performance and capacity are comparable to but its power consumption is lower than a 2.5″ disk (the Travelstar 80GN). If we select the ST 1.3, which has the largest capacity except the MK4004GAH, as a candidate for replacement we will need 7 ST 1.3 disks due to the comparable capacity. Although the aggregate power with an elaborate low-power policy may match the power of a single Travelstar 80GN the number of disks is too large in a practical view. Therefore, as an initial survey into replacing one disk with multiple disks on a mobile platform, we will simply replace one 2.5″ disk with two 1.8″ disks. However the replacement need not be limited to specific disk models and we have used a range of off-the-shelf 2.5″ and 1.8″ disks.

Suppose that a 2.5″ disk processes I/O requests for 15% of the time and is idle for the remaining 85%. From the parameters in Table 1, the total energy consumption over ten minutes will be 691.5 J. If we replaced the 2.5″ disk with two 1.8″ disks and assume that the same workload is equally between the disks, then each will stay in the active state for 9.38% of the time, since a 1.8″ disk has a 25% larger response time. Total energy consumption of the two 1.8″ disks will be 592.56 J, indicating that about 14% of the energy is saved. But if the I/O accesses can be concentrated into a single 1.8″ disk, the other disk can enter its lower-power state (i.e. the standby state). This would save 32% of the energy, compared with the larger disk having no concentration applied. This result presents the upper bound level of energy gains which might be available through this sort of disk replacement. Since such perfect concentration is rather ideal, we now need to apply a realistic dynamic I/O distribution across two disks in the typical mobile workloads. And we do not consider the tradeoff of area and weight between different form-factor disks.

III. ENERGY-CONSERVING TECHNIQUES WITH MULTIPLE DISKS

As suggested in Section II, replacing one 2.5″ disk with two 1.8″ disks may lead to a substantial energy saving. But this requires an adequate data layout technique for mobile workloads. We will now describe a new energy-conserving technique (which we call COR) which skews correlated file requests onto one of a pair of disks and lets the other disk have more idle time. Next, we will review PDC from the viewpoint of energy and performance. Finally, we will describe a more robust and efficient energy-conserving technique that combines COR and PDC.

A. COR: Load Skewing of Correlated Data

The idea of COR is to concentrate most of the correlated data onto one of two disks. *Correlated data* is data which is repeatedly occurred in the same order. This occurs when data requests are generated in the same order by an application (for instance, requests for libraries or configuration files) or files are sent to the disk system so that they can be shared between applications (for instance, when the output of one application is input to another). For example, if files are accessed in the order A, B, C, D, E, B, C and D, then B, C and D will be identified as a group of correlated data. If B, C and D are accessed frequently and all reside on one disk, the other disk is likely to enjoy a long idle time while these files are accessed, and can enter a lower-power state. Since many such groups of correlated data are found in mobile workloads, as mentioned in Section I, this method of load skewing can lead to significant energy savings. The goal of COR is to distribute data across two disks so that one stores most of the correlated data.

COR operates in two phases: first, data accesses that take place in the same order are identified and classified into groups; second, data is moved between disks. To identify and classify groups of correlated data, the system creates a pointer from each file accessed to the one to be accessed next. Although files may be accessed in different sequences, we only keep information for the sequence following the next recent access to determine whether the further accesses take place in the same order. Kroeger et al. [13] report that the probability that a file access will be followed by the same file that followed the last time it was accessed reaches 72% in modern I/O systems. This makes us believe that many access sequences can be detected by tracking the most recent file accesses. However, in future work we plan to find more correlated data using more complete history information.

When a sequence is detected the participating files are registered as a group of correlated data. At every file access, the current accesses are compared against the pre-registered groups of correlated data. If the current group already exists its counter is incremented by one. Lists with counters that grow beyond a pre-determined threshold value are assembled on one of the disks. To minimize the number of data blocks to be moved, the target disk is the one which already contains the largest number of blocks of files of correlated data. By repeating these operations, files in the same group will be located on the same disk and accesses will be concentrated on it.

B. PDC: Popular Data Concentration

PDC is proposed by Pinheiro et al. [5] to deal with highly skewed file access frequencies exhibited in the workloads of some network servers. For instance, the frequency of file accesses by a web server has been shown to conform to a Zipf distribution with a high coefficient. Zipf's law predicts that the frequency of access, or popularity, τ of a file is proportional to the inverse of a power of its rank r: $\tau = 1/r^{a}$. Workloads with a high value of a are known to show skewed popularity in disk accessing. The idea of PDC is to concentrate the most popular disk data by migrating it to a subset of the disks, so that the other disks can be sent to a low-power mode to conserve energy. PDC redistributes data across the disk array according to its popularity, in an orderly fashion. The first disk then stores the most popular data, the second disk stores the next most popular data, and so on. The least popular data and data that is apparently never accessed will be stored on the last few disks. Files are migrated to the target disk until it is full or the expected load approaches its maximum bandwidth.

However, if the frequency of file access varies significantly with time, PDC may cause a lot of file migration, which itself uses energy and also limits the possibility of energy conservation by idle disks. Furthermore, when new files are created they will be stored on the disk with the least popular disk data, interrupting the sleep of that disk. Therefore, if PDC is applied to mobile workloads with varying file popularity, the energy savings may be limited. Our technique is an alternative.

C. COMBINED: Combining COR and PDC

PDC moves popular data to a fixed subset of disks and this may involve an unnecessarily large number of migrations. Consider files on a disk which has been less active in the past but is accessed frequently now. Instead of reclassifying this disk as an active disk with popular data, PDC tries to migrate the popular files to a disk already marked as active (i.e. one of the first few disks). Inflexibly retaining the existing disk classification can lead to a lot of unnecessary data movement.

In this situation, if the recently accessed data on the less active disk can be identified as groups of correlated data, COR will make the less active disk the active disk and will not move the data. But the current version of COR has no way of identifying groups of correlated data on pre-assigned disks and may therefore miss additional chances of concentrating disk I/O. Furthermore, if the recently accessed data exhibits a low correlation, even though each file is frequently accessed, COR may lose the chance of skewing the load to conserve energy because it has no knowledge of the popularity of each file.

Therefore, we propose a complementary scheme called COMBINED, which combines COR and PDC to overcome these disadvantages. COMBINED is a simple combination of techniques: PDC lets file migration occur based on the frequency with which each file is accessed. At the same time, COR repeatedly tries to move groups of correlated data to the same disk. If a file migration request by PDC conflicts with one by COR, COR has priority. Thus, when file migrations result from popularity, only the files which are not registered in the current groups of correlated data will be moved.

IV. EXPERIMENTS

In this section, we first describe our simulation platform and disk models, and then use simulation to evaluate how the various techniques can skew the load through dynamic file disposition and go on to save energy on mobile platforms with multiple small disks. Although our current work employs only two disks, more disks would enhance generality and robustness, and we will consider this extension in future work.

A. Simulation Platform

We implemented a multi-disk power and performance simulator using real workloads from an evaluation board. Our simulator, shown in Figure 1, is composed of three parts: 1) a workload generator which simulates application patterns that are common on mobile systems and generates I/O traces for a disk; 2) a multi-disk file I/O simulation module which simulates variable file dispositions and I/O distribution across multiple disks using the I/O traces from the workload generator and the algorithms mentioned in Section III; 3) a multi-disk power and performance simulation module which estimates each disk's energy use and performance based on the pattern of disk accesses.

Workload generator. This module models usage patterns on specific mobile platforms and generates the corresponding I/O traces. We assume that the applications running on mobile platforms repeat predefined execution scenarios [12] and the types of application are limited to file transfer, email, file search, media play, and idling. Table 2 shows simplified usage patterns of the target applications in a PDA and a PMP, which are typical mobile computing systems. *File transfer* transmits disk files to or receives files from a network. *Email* reads mail messages from a mail box stored on a disk and moves them in another mail box. *File search* reads files stored on disks and searches for specified strings. *Madplay* plays multimedia files using a pre-defined play-list. *Sleep* models periods during which a user stops file operations for a while and no I/O requests are generated.

The workload generator creates I/O requests for a single 1.8″ disk (the MK4004GAH), executing applications repeatedly to follow the above scenarios. I/O requests are delivered to the disk through the page cache, file system, and device driver in Linux kernel 2.4. During these procedures, I/O requests which pass through the page cache and arrive at the disk are recorded on I/O traces. These traces contain the following information: a timestamp, a file identifier, the accessed block's offset within a file, the I/O data size, and the size of the accessed files.

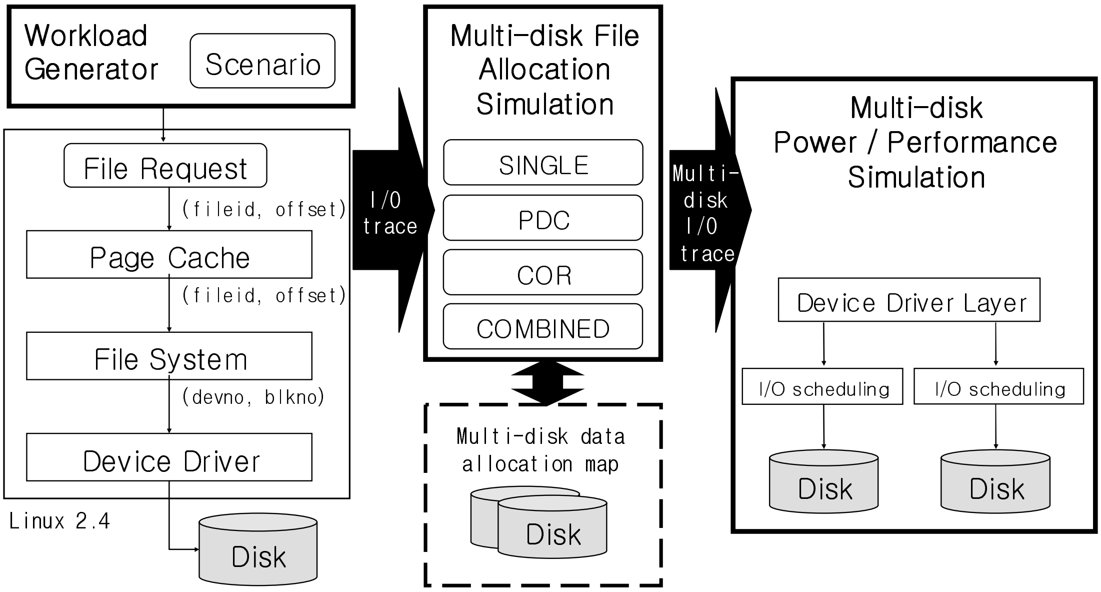

Figure 1. Simulator architecture

Table 2. Simplified usage scenarios

Mobile platform type	Execution order of applications
PDA	file transfer → email → file search → sleep
PMP	file transfer → madplay → sleep

Multi-disk file I/O simulation module. This module simulates static file disposition and dynamic file migration when the traces are forwarded to a mobile platform with multiple virtual disks. The static file disposition is the way in which the files are distributed across the disks before any disk accesses occur. As shown in Figure 1, a multi-disk allocation map is built during this procedure. This map can tell us how the files are distributed across the disks. Dynamic file migration is triggered whenever the energy-conserving techniques decide to move files between disks.

Multi-disk power and performance simulation module. The multi-disk I/O traces handled by the multi-disk file I/O simulation module record how I/O accesses to each disk are distributed. From these traces, this module estimates the performance and energy consumption of each disk using a disk model and a power control policy. We use traditional threshold-based power management (TPM) as the power control policy. TPM spins down a disk after it has been idle for a fixed threshold period. The literature [2,4,5,6] suggests that a multiple-speed disk policy may be more beneficial for workloads with a much shorter idle time, but since multiple-speed disks are not available commercially we use a conventional disk model and a threshold-based power control policy.

Our performance model estimates the total request response time of each disk as the sum of a queue delay, a disk delay, and a service time per request, as follows:

$$T_{total\ response\ time} = \sum T_{queue\ delay} + \sum T_{disk\ delay} + \sum T_{service\ time}$$
$$T_{avg.\ response\ time} = T_{total\ response\ time} \Big/ N.$$

We obtain the average request response time by dividing the total response times by *N*, which is the number of requests in the I/O traces. At the device driver level, while I/O requests are waiting in the request queue they can be merged into a large I/O request if the blocks of a newly added request are contiguous to those of a request waiting in the queue. This is called I/O clustering in Linux and improves performance by reducing the number of I/O requests. But if requests stay too long in the queue the response time will increase greatly, so our simulator limits the maximum delay to 70 ms.

Each request sent from a queue to a disk is serviced with a service time, which is the service time needed to read or write *l* blocks from the starting block b , and can be estimated as follows:

$$T_{service}(b,l) = T_{seek}(b,b_{last}) + T_{xfer}(l) + T_{avg.\ rotation\ delay}.$$

The seek time $T_{seek}(b,b_{last})$ is the time which it takes to move the head located at block b_{last} to block b when an I/O occurs. We adopt a simple seek time model using the linear approximation $T_{seek}(b,b_{last}) = a_1|b - b_{last}| + a_0$, where a_0 and a_1 are constants and vary according to the characteristics of the disk. $T_{xfer}(l)$ is the time to actually read or write *l* blocks and is calculated by multiplying a unit transfer time by *l*. As the rotation delay fluctuates irregularly, an average rotation delay $T_{avg.\ rotation\ delay}$ is used.

The total energy consumption of disk *i* is modeled as the sum of the energy used by the disk in each state and the energy consumed during its transition periods:

$$E_i = \sum_j P_{ij} \cdot T_{ij} + \sum_k \sum_l N_{ikl} \cdot E_{ikl}.$$

The disk state *j* can be active, idle, or standby and in each state a disk has a different power consumption value P_{ij}. T_{ij} is the period during which disk *i* stays in each power state. E_{ikl} is the energy consumed by a transition from state *k* to state *l* , and N_{ikl} is the number of transitions from state *k* to state *l* on disk *i*.

B. Simulation Results

Simulation setup. The aim of our simulations is to assess the potential for energy conservation replacing a laptop disk with two smaller disks in a mobile platform, and to determine which load concentration technique can most improve energy consumption and performance. The simulations were conducted using the disk model and I/O traces described in the previous subsection. Detailed power and performance parameters for the different form-factor disks are given in Table 3. The performance and power parameters of both disks are the same as those given elsewhere [7][8], but the capacities of the 1.8″ and 2.5″ disks are bounded to 400MB and 800MB respectively.

We used a PXA255 embedded evaluation board with a 1.8″ Toshiba MK4004GAH disk and a Linux 2.4 kernel to run the workload generator and extract I/O traces. To generate I/O traces of mobile workloads, the scenarios in Table 2 were performed for 71 minutes for the PMP and 81 minutes for the PDA. The request rates were respectively about 3 requests/second and about 10 requests/second.

We compared the energy-efficiency and performance of five schemes, which have different data layouts and I/O distributions across the disks. SINGLE uses one 2.5″ disk and has no file migration. The other schemes all use two 1.8″ disks. SPAN also has no file migration. PDC, COR and COMBINED each incorporate the eponymous load-skewing technique. For PDC, we assume that file migration occurs every 5 minutes. The five schemes are listed in Table 4. We compared the energy savings and average request response times of all the other schemes against SINGLE, which is a baseline scheme representing an upper bound on energy consumption and a lower bound on average response time. All the schemes have a threshold-based power control policy as described in the previous subsection.

Simulation results. In the simulations we assume that files are initially located randomly but uniformly across the disk(s). Figure 2 shows the energy consumption of the five schemes for the PMP trace. As expected, SINGLE exhibits the highest energy consumption of 7874 J. By simply replacing the 2.5″ disk with two 1.8″ lower-power disks, SPAN achieves a 22% energy saving, consuming 6155 J. PDC does not achieve long idle times in this scenario and the energy saved by PDC is less than that saved by SPAN. COR and COMBINED save about 15% more energy than SPAN and 34% more than SINGLE. These results elicit two observations: first, replacing a high-power disk with two smaller disks can be beneficial by itself for energy conservation; second, file migration can save substantial additional energy. The energy-conserving effect of concentrating the I/O accesses that result from appropriate file migration can be confirmed by analyzing the time that disks spend in each state.

Table 3. Disk parameters

Form factor (″)		2.5	1.8
Capacity (Mbytes)		800	400
Rotation Speed (RPM)		4200	4200
Avg. seek time (ms)		12	15
Power (W)	Active	2.3	1.4
	Idle	0.95	0.4
	Standby	0.25	0.12
Active to idle energy/time (J/s)		1.15/0.5	0.7/0.5
Idle to active energy/time (J/s)		1.15/0.5	0.7/0.5
Active to standby energy/time (J/s)		2.94/2.3	2.05/3.1
Standby to active energy/time (J/s)		5.00/1.6	1.84/1.7
DPM threshold: idle/standby (s/s)		1/3.379	1/5.979
Seek time model (a1, a0)		(2.9^{-10}, 0.0072)	(3.6^{-10}, 0.0090)

Figure 3 shows how long each disk stays in each state for each scheme under the PMP scenario. SINGLE seldom has enough idle time to enter a lower power mode and it is active time is longer than any other schemes. Using SPAN, the I/O accesses are split between two disks, which achieve 30% and 18% standby time. Despite file migration PDC cannot concentrate the load and each disk shows almost the same disk time distribution, and consequently PDC does not have idle time enough to put one disk into a lower-power state than the other. COR achieves 36% and 50% standby on each disk by use of file migration based on correlated data, leading to lower energy consumption. COMBINED only achieves as much standby time as COR.

Table 5 shows the average time taken to respond to a request, which can be taken as a measure of the performance of each scheme. Due to the inherent difference between the performances of 2.5″ and 1.8″ disks, SPAN has double the response time of SINGLE. PDC shows a 2.6 times longer average response time than

SINGLE, due to the large number of file migrations. With fewer file migrations than PDC, COR and COMBINED have only 30ms more delay than SINGLE.

Table 4. List of the schemes

Scheme name	Description
SINGLE	Single 2.5″ disk and no file migration
SPAN	Two 1.8″ disks and no file migration
PDC	Two 1.8″ disks and PDC algorithm
COR	Two 1.8″ disks and COR algorithm
COMBINED	Two 1.8″ disks and COMBINED algorithm

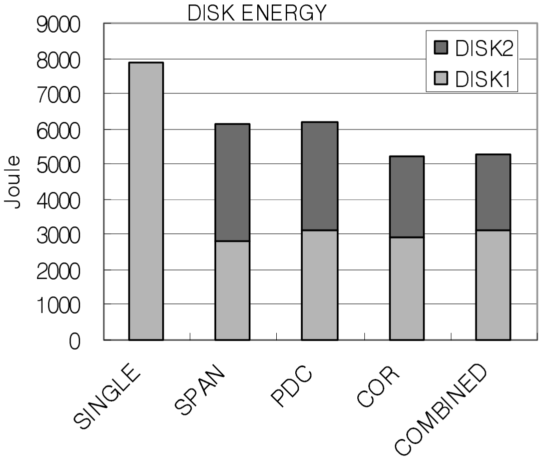

Figure 2. Energy consumption (PMP scenario)

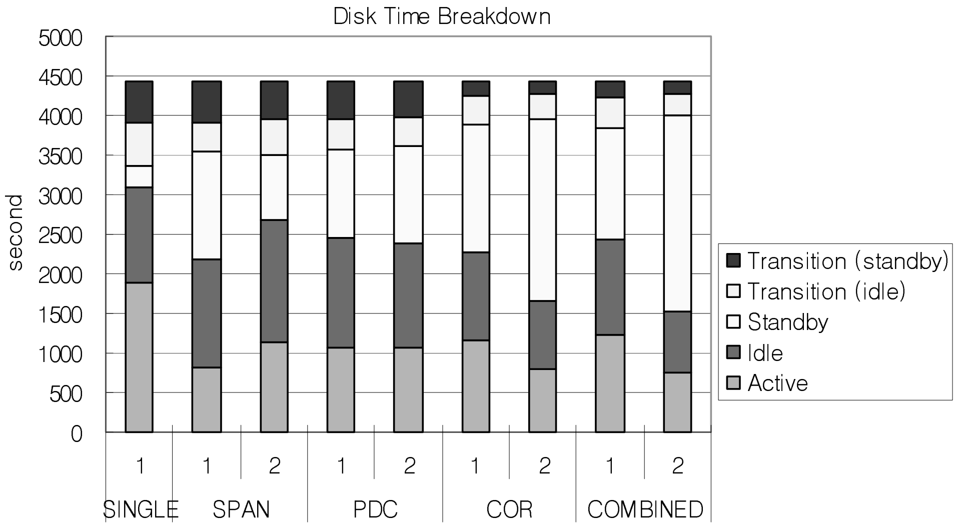

Figure 3. Disk time breakdown (PMP scenario)

Table 5. Average request response time (PMP scenario)

Scheme	Avg. request response time (s)
SINGLE	0.133
SPAN	0.279
PDC	0.342
COR	0.167
COMBINED	0.166

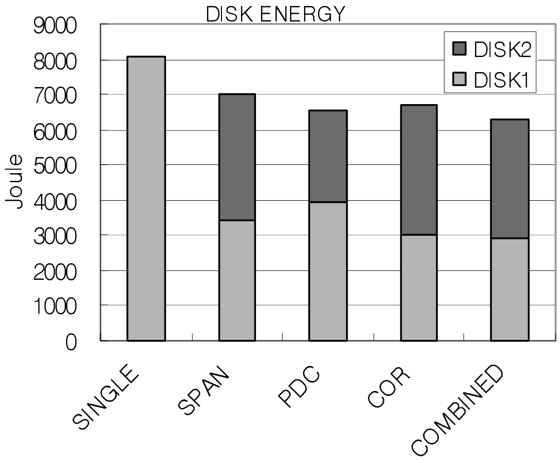

Figure 4. Energy consumption (PDA scenario)

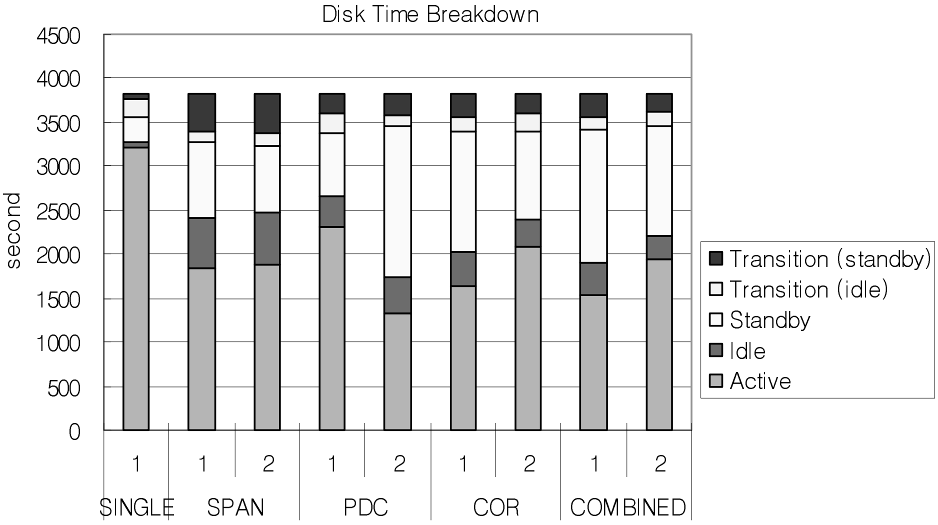

Figure 5. Disk time breakdown (PDA scenario)

Table 6. Average request response time (PDA scenario)

Scheme	Avg. request response time (s)
SINGLE	0.047
SPAN	0.123
PDC	1.176
COR	0.112
COMBINED	0.104

Figures 4 and 5 and Table 6 show the energy consumption, the time spent in each state, and the average request response time under the PDA scenario. These results are similar to those for PMP, but there are several differences worth mentioning.

First, PDC now exhibits better energy conservation than SPAN and even COR. But, due to the I/O overhead of a large increased number of file migrations (PDC has 80138 file migrations and COR has 10491), PDC has an average request response time that is ten times longer than COR. This indicates that PDA workloads favor algorithms based on file popularity, as does PDC, rather than file migration based on correlated data. Nevertheless, file migration overheads are still critical to the overall disk system performance.

Second, unlike PMP scenario COMBINED now has the best performance in terms of energy conservation, together with an adequate average response time. COMBINED saves about 23% more energy than SINGLE with a reasonable average performance delay of 60 ms or so. COMBINED saves 4.4% more energy than PDC with a much lower response time.

V. RELATED WORK

Carrera et al. [2] and Papathanasiou et al. [3] investigated the possibility of saving energy by replacing high-speed server disks with arrays of smaller form-factor disks with almost the same aggregate I/O throughput. Carrera et al. showed that a combination of laptop and SCSI disks can provide energy savings in network servers. But these authors do not look at replacing high-performance disks with a set of lower-power disks deeply, and they concentrate on using two-speed power management of SCSI disks to save energy.

Papathanasiou et al. propose using arrays of laptop disks in place of a server disk to save energy, using the tradeoff between energy efficiency and worst-case response time, but they simply assume that the original contents of a server disk are already mirrored on three laptop disks, and do not consider data migration. They also employ an optimal dynamic power management (DPM) policy.

Gurumurthi et al. [4] proposed a multiple-speed disk technique called dynamic rotations per minute (DRPM) for disk array based servers, which exploits access patterns that exhibit short idle intervals. DRPM varies disk speeds to match the required average response time and the length of the request queue, instead of I/O throughput, but there is no migration of data between disks.

Pinheiro et al. [5] proposed a technique called popular data concentration (PDC) that dynamically migrates popular disk data (i.e. frequently accessed data) to a subset of the disks in an array. Their aim is to skew the load towards a few of the disks, allowing the others to be sent to low-power modes. They assert that PDC can conserve a substantial amount of energy using two-speed disks on network servers, including web servers. But their simulation results with PDC show that the energy efficiency and performance delay can vary considerably, depending on parameters such as request rate and migration period.

Zhu et al. [6] proposed a combinational technique called Hibernator, which combines intelligent speed setting and data migration to conserve energy on multi-speed disk arrays, with response time guaranteed by a service-level agreement (SLA). Hibernator exploits a RAID5-like striping scheme to achieve redundancy, and its migration techniques are largely specific to database servers.

The above techniques are all targeted to server workloads and our method aims at mobile workloads. But, as described previously the basic idea of energy conservation using I/O concentration through data migration can be applied to mobile storage systems with multiple disks. And, we focus on improved energy conservation and acceptable response time, rather than looking to enhance reliability and availability using redundancy. This is because a mobile system can accommodate a limited numbers of disks due to space, weight and battery restrictions, while high throughput is not normally required.

VI. CONCLUSIONS

Despite the rise of flash memory hard disk drives remain the most significant mass storage systems for mobile and ubiquitous computing platforms, due to their capacity, performance and low cost. We expect that mobile platforms with disk arrays may soon be available.

We have proposed a novel energy-conserving technique, appropriate for the workloads of mobile devices, which uses multiple small form-factor disks instead of one larger disk. Our specific contribution is to suggest a more energy-efficient load-skewing technique across multiple small disks based on the I/O patterns frequently found in mobile-device workloads. Workload-based simulations showed that our scheme can save more than 34% energy consumption over a single disk solution, and also can save more than 14.8% of disk energy consumption and improve the average I/O response time by up to 6 times, compared with the existing PDC energy-conservation scheme.

We plan to extend our algorithms by introducing additional disks and we will also look more closely at aspects such as reliability, energy and performance. We are also interested in the study of energy-conserving techniques based on heterogeneous storage systems for mobile platforms.

REFERENCES

[1] L. D. Paulson, "Will hard drives finally stop shrinking?", *IEEE Computer*, vol. 38, no. 5, pp.14-16, 2005.

[2] E. V. Carrera, E. Pinheiro, and R. Bianchini, "Conserving disk energy in network servers", in *Proc. of* the *17th International Conference on Supercomputing*, June 2003.

[3] A. Papathanasiou and M. Scott, "Power-efficient server-class performance from arrays of laptop disks", Technical Report 837, Department of Computer Science, University of Rochester, May 2004.

[4] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, and H. Franke, "DRPM: Dynamic speed control for power management in server class disks", in *Proc. of the International Symposium on Computer Architecture*, June 2003.

[5] E. Pinheiro and R. Bianchini, "Energy conservation techniques for disk array-based servers", in *Proc. of the 18th International Conference on Supercomputing (ICS'04)*, June 2004.

[6] Q. Zhu, Z. Chen, L. Tan, Y. Zhou, K. Keeton, and J. Wilkes, "Hibernator: helping disk arrays sleep through the winter", in *Proc. of the 20th ACM Symposium on Operating Systems Principles*, Oct. 2005.

[7] Hitachi GST, Travelstar 80GN.

http://www.hitachigst.com/tech/techlib.nsf/products/Travelstar_80GN.

[8] Toshiba, MK4004GAH.

http://www3.toshiba.co.jp/storage/ english/spec/hdd/mk4004gs.htm.

[9] Seagate, ST1.3.

http://www.seagate.com/products/consumer_electronics/st1series.html.

[10] Hitachi GST, Microdrive 3K8.

http://www.hitachigst.com/tech/techlib.nsf/products/Microdrive_3K8.

[11] Toshiba, MK4001MTD.

http://www3.toshiba.co.jp/storage/english/spec/hdd/mk4001.htm.

[12] F. Chen, S. Jiang, and X. Zhang, “SmartSaver: turning flash drive into a disk energy saver for mobile computers”, in *Proc. of 11th ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED’06)*, Tegernsee, Germany, October 4-6, 2006.

[13] T. M. Kroeger and D. D. E. Long, “The case for efficient file access pattern modeling”, in *Proc. of the Seventh Workshop on Hot Topics in Operating Systems*, March 1999.