Autonomic Exploration of Trade-offs between Power and Performance in Disk Drives

Alma Riska Evgenia Smirni
College of William and Mary,
Williamsburg, VA 23187, USA
{riska,esmirni}@cs.wm.edu

ABSTRACT

Over-provisioning is a standard capacity planning practice that leads to disk drives that operate mostly under very low utilization (as low as single digit utilization) but that are consuming disproportional amounts of power. Methodologies that place the disk drive into a low power mode during idle times can assist in conserving power. This is a challenging problem because the performance of future jobs cannot be compromised, yet there is no knowledge of future disk arrivals. In this paper we explore the above problem by exploring ranges and trade offs of possible power savings and performance within a set of enterprise storage traces. We demonstrate the difficulty of obtaining significant power savings even in traces where overall utilization is less than 5% and explore the feasibility of popular schemes such as workload shaping for power savings. We also propose an autonomic algorithm that suggests when and for how long a power savings mode should be activated given an acceptable performance degradation target that is user provided. The robustness of the algorithm is illustrated via extensive experimentation.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Design Studies

General Terms

Algorithms, Design, Management, Performance

Keywords

Performance guarantees, power savings, disk drives, continuous data histograms

1. INTRODUCTION

The problem of power consumption and energy inefficiency in data centers that often host thousands of disks is indisputably a prevailing one as systems are routinely configured in order to meet peak user demands. User demands are often characterized as bursty, resulting in temporal loads of orders of magnitude higher than the average load. Given such workloads, standard capacity planning promotes over-provisioned systems that operate most of the time under low average utilization but that unfortunately consume disproportionally high power.

Idle periods in under-utilized disk drives offer opportunities for saving power in a straight forward manner: the system can selectively put disk drives in a low power operation mode (or idle mode) during idle times [14]. Doing this *transparently* to the end user is a challenging task: requests that arrive while the disk is in a power saving mode are to be inevitably delayed as the system requires a recovery period before the disk drive is mechanically set to a state that allows it to serve jobs again. The challenge here is to strike a balance between two clearly conflicting targets: achieve as high energy savings as possible while restraining response time degradation as much as possible (or to within predefined limits). To meet the above targets, the following questions must be answered.

> *When should an idle mode be activated?* Practical reasoning presumes that the disk may not be put on low power mode immediately after an idle interval is detected, as future arrivals of disk requests are largely unknown a priori. Borrowing ideas from background scheduling [4] it may be desirable to leave the disk in active state for a period of time, in anticipation of an upcoming request arrival.
>
> *How long should an idle mode last?* In light of the fact that future arrivals are again unknown, it may not be desirable from the performance perspective to stay in an idle mode long, in anticipation of future arrivals and of the require time to “activate" the disk in order to bring it in a state where it can serve requests.

The above issues highlight the conundrum that system designers face. A workload that results in a highly underutilized disk drive is not always an excellent candidate for power savings. If requests are spaced such that there is hardly time to bring the system in low power mode and bring it up timely to serve requests, it may simply not be possible to mine the idle times for power savings. As we demonstrate later with a simple example, simple metrics such as average disk utilization may give a distorted view of possible power savings.

In this paper, we present a solution to this difficult problem by leveraging on a schedulability framework that is initially proposed for the general problem of scheduling background jobs in disk drives with performance guarantees [13]. This framework utilizes histograms of idle intervals and the anticipated duration of background jobs that are non-preemptable to best serve the latter within the available idle periods. The basic assumption is that there is no a priori information about the arrival timestamps and/or the duration of upcoming background jobs, nor about the length of upcoming

ICAC'10, June 7–11, 2010, Washington, DC, USA.

idle intervals except of statistical information in the form of histograms from observations of past workloads.

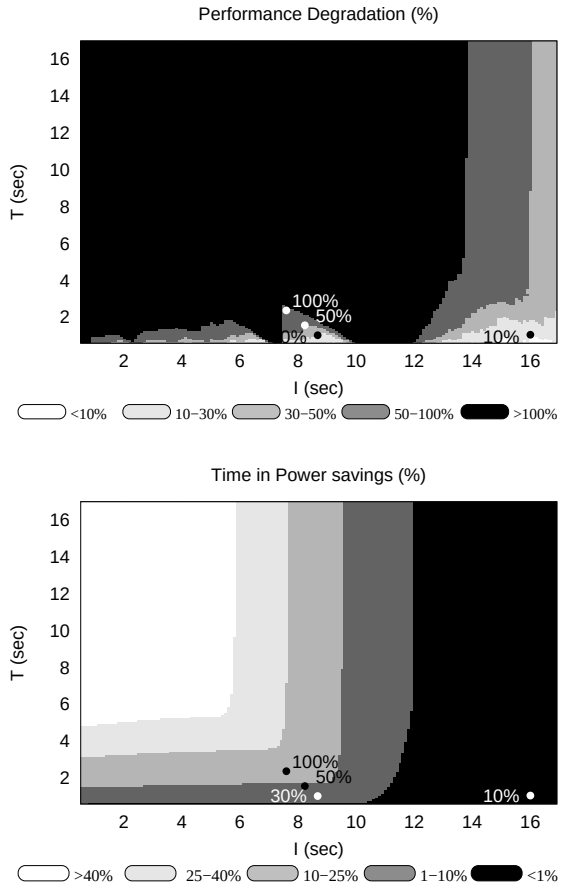

Figure 1: Performance slowdown (top) and the respective power savings potential (bottom) in a test case involving a single disk drive in a storage system used by a file server. Power savings is presented as the proportion of time that the system stays in power saving mode over the duration of the trace. The solid dots identify the (I, T) pairs that are computed by the framework. Each dot is associated with a percentage value D which an input parameter and signifies the acceptable performance degradation of jobs as defined by the user and/or system administrator.

Here, we treat the time that that the disk is placed on idle mode as a preemptable background job, but with the additional complication that the disk drive requires a recovery overhead to bring the system to the active mode. We use concepts and ideas of the schedulability framework presented in [13] to create a robust power saving prediction methodology that uses a selection mechanism to determine with simple calculations *which* idle intervals, if any, should be utilized for saving power. Naturally, the methodology avoids using short idle intervals for power savings, since that would impact negatively both performance and reliability of the storage system. The schedule of power saving in a disk drive is guided by two parameters: the time I that elapses during an idle interval before the power saving mode kicks in and the total time T that the system is put into a power saving mode. All idle intervals that are shorter than I units of time will not be utilized for power savings. Autonomically determining the values of the (I, T) pair that meet the power and performance goals of the system is the purpose of this paper.

Motivating Example

To motivate the difficulty of the problem, we have exhaustively explored the I and T parameters for a disk drive trace from an enterprise storage system that has a disk utilization equal to only 0.5%. At a first glance, this trace looks like an excellent candidate for power savings. Figure 1 draws *regions* of different levels of performance slowdown (top plot) and the corresponding power savings (bottom plot) as a function of I (x-axis) and T (y-axis). By looking at these maps, one can immediately identify various (I, T) pairs that strike a good balance between performance and power savings. For instance, for the specific disk drive trace that we consider here, if one is interested in power savings greater than 30% (see the lightest regions in the power savings plot of Figure 1), this can be obtained by suffering performance slowdown greater than 100%. This is almost inconceivable, given that the average utilization of this trace is only 0.5%, which implies that this trace is 99.5% idle!

The specific example motivates the difficulty of the problem but also how one can be easily misled by looking at single parameter measures such as average utilizations. Yet, the figure illustrates that there is some room for power savings *if* we have such a map. Creating these maps is computationally expensive as it requires to exhaustively explore the entire state space of (I, T) pairs, running one simulation for every (I, T). Even if this were possible, it would still not be practical because workloads are dynamic and rarely known a priori.

The novelty of the work presented in this paper is the accurate identification of an (I, T) pair that is located in a feasible region within the maps *without* generating the above maps. This (I, T) pair represents a power savings schedule that meets the user-defined trade-off between performance and power savings. Given an acceptable average response time slowdown D as an input parameter (i.e., the user/system performance target), the framework that we present in this paper computes a suggested (I, T) pair *and* estimates the corresponding average power savings that can be achieved with this (I, T) pair while meeting the target of an average slowdown in response time of less than D. The framework's output, (i.e., the (I, T) pair) may not be the optimal one but it is consistently within the regions of the best possible scheduling choices. Indeed, in Figure 1 the various solid dot markings identify the (I, T) pairs that are suggested by our framework for different values of acceptable user slowdown D as given by the percentage value next to the dot. Note that all lie within the best region for the noted foreground slowdown target. The significance of the framework is that it manages to find an ideal (I, T) pair fast, based on a compact analytic model that is parameterized by simple observed past workload metrics. The overhead for monitoring the system metrics and the actual estimation procedure is minimal.

The case presented in Figure 1 highlights the difficulty of saving power in disk drives, even in the case where disk drives appear to be severely underutilized. Figure 1 clearly illustrates that power savings, if done haphazardly, may come with a dear cost: significant delays for the end-user. Workload shaping [15, 14] has been proposed in the literature as an alternative way to further create more room for power savings by moving some of the work (either in the form of reads or whites or both) to cache, to a buffer, or to another disk. Our framework helps in identifying whether there is additional room for power savings in the system or for reducing performance slowdowns by assessing whether it is possible to take advantage of workload shaping.

Contributions and Paper Organization

The contribution of this paper is the compact analytic model that is proposed which identifies accurately and efficiently good scheduling choices for power savings and performance *without* exhaustively exploring all the scheduling choices. The system allows the system to estimate *beforehand* performance vs. power savings trade-offs for multiple available choices such as which power saving mode to utilize in a disk drive or which workload shaping technique (if any) will extend power savings for the current workload. Ultimately, these estimations will guide the system to activate the technique that accurately reflects the system conditions and would yield a feasible trade-off between performance and power savings.

In addition to the example in Figure 1 that gives a preview of what our framework does, we illustrate the robustness of this modeling framework via trace driven simulations using four enterprise disk-level traces with very different characteristics. Our simulations show that our prediction for saving power that is based on monitoring simple system metrics is robust and always identifies the trade-off between potential power savings and system performance degradation. Ultimately, our framework answers the simple question whether workload shaping can be effective for power savings in a disk drive environment by quantifying the power savings giving a target workload slowdown.

This paper is organized as follows. Section 2 summarizes the power savings opportunities in disk drives and storage systems. In Section 3, we present the methodology that we propose to identify and estimate the power savings opportunities in a system under a given workload. We validate the effectiveness of the approach and illustrate its robustness in Sections 4 and 5 using trace-driven analysis and simulations. Section 6 positions our contributions relatively to related work. Conclusions and future work are given in Section 7.

2. POWER SAVINGS IN STORAGE

Disk drives consist of several mechanical and electronic components that consume power. First, the read/write heads (i.e., recording arm) fly at a very precise distance from the magnetic media. Second, the media platters rotate continuously at a constant speed (RPM). Third, the on-board electronics manage all the components of the drive and communicate continuously with the rest of the computer system. Consequently, power can be saved in a disk drive by stopping or slowing down any subset of these power consuming components, when they are not doing any useful work such as serving user requests.

There are several levels of power consumption in disk drives depending on the disk components that are active and operational. Unfortunately, when drive components operate in a power saving state, the disk drive itself is not active and it takes some time to bring it back up and ready to serve requests. Consequently, each level is distinguished by the amount of *power* it consumes and the amount of *time* it takes to get out of the power saving mode.

The exact amount of power savings and time it takes to get out of a power saving mode differs between drive families. The rotational speed, capacity, and drive form factor determine how much power is consumed and how much power can be saved in any power saving mode. Below we list all levels of power savings in a disk drive and the respective *expected* savings and penalties without focusing on a particularly disk drive family.

- **Level 1**: the drive is serving requests and it consumes power depending on the workload characteristics, such as sequential/random, and Reade's/WRITEs, with sequential WRITE workload consuming the highest amount of power.

- **Level 2**: the drive is idle but "active", which means that any new request gets served immediately without any delay, the amount of power saved is as much as 50% of the power consumed in Level 1. This means that even if the workload is managed such that the drive goes to extended periods of idleness, the amount of consumed power is reduced.

- **Level 3**: the drive heads are "parked" away from the drive platters (unloaded), without slowing the platter's rotation. With less drag from the heads, the drive consumes 15-20% less power than in "active" idle mode (i.e., Level 2). The penalty to reload the heads is about half a second.

- **Level 4**: the drive heads are "parked" away from the drive platter (unloaded), and the platter rotation is slowed down. With less drag from the heads, and less motor power to rotate the platters, the drive consumes 30% less power than in "active" idle mode (i.e., Level 2). The penalty to reload the heads and pick up the rotation speed is about a second.

- **Level 5**: the drive heads are "parked" away from the drive platter (unloaded) and the motor is stopped, i.e., the platters do not rotate at all. Only the electronics in the drive are on, to communicate with the host and receive requests. With no motor power, the drive consumes 50% less power than in "active" idle mode (i.e., Level 2). The penalty to reload the heads and turn on the motor to rotate the platters is about 8 seconds.

- **Level 6** the disk drive is spun down entirely cutting the power consumption almost entirely because neither mechanical nor electronic components in the disk drive are operational. However, to bring the disk drive back up takes as much as 25 seconds.

We summarize the respective power savings and time-to-ready penalties of the various power saving modes in disk drives in Table 1. We remark that the associated time penalties associated with each power level are within representative ranges for disk drives [8, 18]. Among the above levels of power savings, we are interested in those that have smaller penalties such as levels 3 through 5, because in enterprise disk drive the average length of idle periods cannot accommodate shutting down a disk drive (i.e., Level 6) without significant degradation in performance.

	Power savings relative to "active idle"	Time to activate: "penalty"
Level 2	0%	0 sec
Level 3	18%	0.5 sec
Level 4	30%	1 sec
Level 5	50%	8 sec
Level 6	95%	25 sec

Table 1: Idle modes in a disk drive, their power savings relative to the "active idle" mode (level 2) and the time it takes the disk drive to become ready.

In the following section, we focus on estimating, for a given workload, the power savings and performance penalty for power saving levels 3 and 4. The choice of the appropriate power savings level, is left to the system management unit, because it depends on how sensitive a system is to performance degradation.

3. ALGORITHMIC FRAMEWORK

The power savings modes discussed in Section 2 are used commonly in mobile, personal, and archival storage devices and systems to limit the amount of power consumed by a disk drive [3, 2, 6]. However, with the explosion of the on-line data centers that support enterprise applications, it is desirable to exploit power savings opportunities even in such non-traditional domains [14]. The

issue though is that power savings in disk drives may cause significant delay to some of the requests, if done haphazardly. While performance degradation may be acceptable for archival systems, it is certainly not desirable for high-end systems and it should be controlled.

Here we propose a methodology that identifies power saving capabilities for a given workload in a disk drive such that the degradation caused in performance by the power saving modes lies within a threshold D set (dynamically or statically) by the user or the system itself. The goal of our methodology is to provide *a mechanism to estimate* power savings and the respective performance degradation before they take effect in the system. This methodology, can be used to quantify savings resulting from a power saving mode for a given workload and desired performance degradation D. As a result, the system can decide which, if any, power saving mode to activate in a disk drive.

There are widely accepted common practices when it comes to activating power saving modes in disk drives. First, the power saving mode is a low priority task in a disk drive. This means that the disk is put through a power saving mode when it is idle and it become active again at the latest when a new disk request arrives. We refer to the time the disk is set to remain in a power saving mode as T. Second, the disk drive should not be placed into a power saving mode immediately after it becomes idle. To avoid significant performance degradation some time I should elapse in idle active mode. Based on these two considerations, there are two main steps that we propose to follow in our estimation methodology for any given power saving mode

- **Step 1**: determine "when" (I) and for "how long" (T) the disk drive should be put into the power saving mode such that the degradation target D is not violated,

- **Step 2**: estimate power savings that result from activating the power mode based on the scheduling pair (I, T).

At the system level, multiple paver saving modes may need to be evaluated to determine which to activate. In that case, the above steps should repeat for each power saving mode and the most efficient one should be activated. We utilize the algorithmic framework proposed in [13] to complete the first step in our estimation methodology and develop the procedure for step 2 in this section.

3.1 Estimation of the pair (I, T)

The framework proposed in [13] represents a general algorithm that determines when to schedule tasks of low priority in a storage device or system such that performance of the high priority tasks does not degrade more than D. The outcome of the framework is the pair (I, T) that schedules low priority work for T units of time only after I unites of time have elapsed in an idle system.

In addition to the user/system input D, the algorithm in [13] uses as input workload metrics that are monitored in the system such as the length of idle intervals, the response time of user requests, the delay experienced by a high priority task that finds the system busy serving a low priority one. The accuracy and the flexibility of the algorithm is associated with using the histogram of monitored idle times in the system to determine the pair (I, T). Because the framework in [13] monitors the idle intervals continuously, it dynamically adapts its estimation to the current workload in the system and reflects any changes.

3.2 Estimation of power savings

The outcome of the framework in [13] is the pair (I, T), where I indicates when to initiate a given power saving mode at the disk and T indicates for how long to keep the drive in that power saving mode such that D is not violated. To estimate the power savings enabled by the scheduling pair (I, T) we continue to use here the histogram of idle time that is constructed to determine the pair (I, T). Lets denote by B the units of time that the disk drive is placed into the power saving mode, as determined by (I, T). One of the main consideration in estimating B is the penalty associated with taking the disk drive out of the power saving mode. This penalty is given in Table 1 and discussed in Section 2. The pair (I, T) indicates that the disk drive will spend at most T units of time every time the power saving mode is activated. However, the effective time during which the system saves power is $(T - P)$, where P denotes the penalty to bring a disk back into active mode after the power saving mode.

The average amount of time B in the power saving mode is estimated by categorizing the idle intervals as following

1. idle intervals shorter than I which can not contribute to saving power,

2. idle intervals of length R that fall between I and $I + T - P$, where the amount of time in power savings mode is exactly $R - I$, and

3. idle intervals of length R that are longer than $I + T - P$, where the amount of time B in power saving mode is $T - P$.

Figure 2 depicts how to use the histogram of idle times to estimate the amount of time B that the disk drive stays in the power saving mode with penalty P, which starts after I units of idle time have elapsed and ends T units of time later.

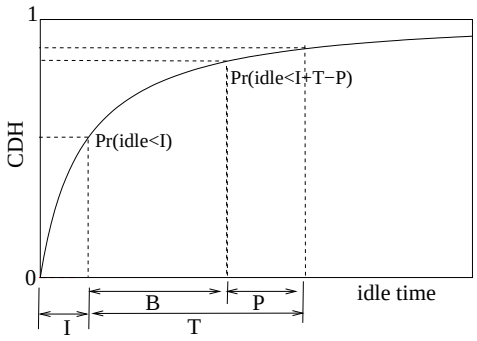

Figure 2: Estimation of the amount of time B that the disk stays in the power saving mode with penalty P which starts after I units of idle time have elapsed and ends T time units later.

The following equation captures how the amount of time in power savings is actually estimated using the idle times histogram

$$B = \int_{l=I}^{I+T-P} Pr(l) \cdot (l - I) + \int_{l=I+T-P}^{max} Pr(l) \cdot (T - P), \quad (1)$$

where $Pr(i)$ is the probability of an idle interval being of length l and max is the maximum length of the idle intervals in the system. **I changed the index here – now I think it is correct** Note that in the implementation of the algorithm, the integrals in the above equation are just finite sums. Eq. 1 gives the average amount of power savings per idle interval, although not every idle interval is utilized for power savings.

In order to estimate power savings under the scheduling pair (I, T) for a longer period of time, we take into consideration the total number of idle intervals observed during that period of time.

Specifically, the amount of power savings S over the period of time $Time$, is estimated using the following relation

$$S = \text{Savings over active idle} \cdot \frac{B \cdot \text{Number of Idle Intervals}}{Time}. \quad (2)$$

3.3 Applying the estimation procedure

The estimation procedure discussed in this section can be applied to estimate power savings in disk drives for different power saving modes and different workload shaping techniques that may be used to improve idleness in the disk drive (discussed in more detail in Section 5).

When comparing different power saving modes, we use the same histogram of idle times as the basis for our estimations. The main difference between the power saving modes taken into consideration is the penalty P. This penalty is an input parameter in the estimation of the scheduling pair (I, T). Consequently, for the same workload but different power saving modes there will be different scheduling pairs (I, T). The second difference between the power saving modes, which is savings over the active idle mode from Table 1 is used in Eq. 2 where the actual power saving S for a power mode is estimated.

For workload shaping techniques, which if active will change the shape of idleness in the disk drive, another histogram of idle times is constructed. Consequently, the differences in the estimated power savings here depend deeply on the changes in the histogram of idle times.

4. PERFORMANCE EVALUATION

Here, we evaluate the framework described in Section 3 via trace driven analysis and simulation. We use a set of traces measured at the disk level of two enterprise storage systems, an application development server ("Code") and a file server ("File") [17]. These traces record for each request that reaches the disk drive, the arrival time, the departure time, the type of the request (i.e., read or write), the length, and the location on the disk. The traces provide the highest level of detail with regard to the utilization of idle intervals for power savings, because the foreground busy periods and the idle intervals are captured *exactly*.

We give the high level trace characteristics in Table 2. Note that from the four traces presented in this table, the trace that is examined as a motivation example in Section 1 is "Code 2".

The traces indicate that the disks are underutilized but the idle intervals are highly variable (see the coefficient of variation, CV). Still if one had perfect knowledge of the length of idle intervals, the power savings would be around 10-17% for Level 3 power savings and between 15-28% for the Level 4 power savings. **E: I think that the reported results for level 3 and 4 are reversed. If you agree, please fix the numbers on the sentence above and on the table – switch columns. With that thinking, I added the following two sentences**. The "savings" column in Table 2 represents the ratio of the time that the system is set in low power mode over the trace duration. Since the penalty P to reactive the disk is higher with level 4 than with level 3 (see Table 1), it is natural that the reported time savings are smaller than level 3. We point out that these numbers give only an indication of actual power savings (we expect higher power savings in level 4 idle mode than level 3 but more power consumption to bring up the system in active mode).

As suggested in Table 2, the length of idle intervals in all traces is variable. In Figure 3, we show the distribution of the length of idle times for the traces of Table 2. The plot confirms that the distribution of the length of idle intervals has a long tail in all cases. The long tail indicates that there are some very long idle intervals

Trace	Mean Resp	Util (%)	Idle Length		Saving (%)	
			Mean	CV	Lev. 3	Lev. 4
Code 1	8.6	5.6	192.6	8.4	10	15
Code 2	8.6	0.5	1681.6	2.3	17	26
File 1	12.7	1.7	767.5	2.3	13	16
File 2	15.4	0.7	2000.2	3.8	17	28

Table 2: Trace characteristics: measurements are in milliseconds unless otherwise noted. The "Saving" columns indicate the upper bound on power savings under Level 3 and 4, achieved if *perfect* knowledge of the beginning and the duration of idle intervals is available. All traces are 12 hours long.

(several times longer than the idle interval mean) which can be exploited for power savings, particularly for traces "Code 2" and "File 2".

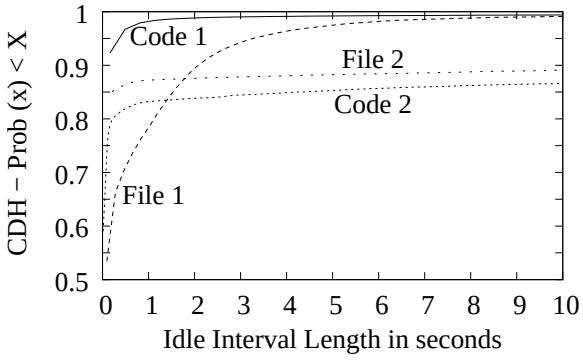

Figure 3: Histogram of idle times for our traces.

As explained in Section 2, there are multiple levels of power consumption in a disk drive and Table 1 lists the corresponding power savings and performance penalty for the ones of most interest in enterprise systems. We use the methodology laid out in Section 3 to identify the appropriate (I, T) pair given as input the target degradation D and the level of idle mode, and use Eq. 2 to estimate the power savings S. In addition, we run a trace driven simulation that puts the system in the selected power saving mode as guided by the selected (I, T) values and compare the estimated power savings with the actual power savings as given by the simulation. We also compare the user input performance degradation D with the actual performance degradation as given by the simulation.

We present these results in Tables 3, 4, 5, and 6 for traces "Code 1","Code 2", "File 1, and "File 2", respectively. Specifically, we show

Request Degradation: the average slowdown in user requests attributed to power savings (an input parameter, aimed to be closely met),

Time in Power Saving Mode: the ratio of the time in power saving mode to the duration of the trace.

The results in Tables 3, 4, 5, and 6 strongly suggest that our methodology estimates with very high accuracy the amount of time that the system can be put in a power saving mode for the given workload. Also, the suggested (I, T) pair yields performance that closely matches the user-set target.

One counter-intuitive observation in the results of Tables 3, 4, 5, and 6 is that "Code 1" has better power savings potential than "File 1" although the latter has more available idle time and generally

Level 3				Level 4			
Performance Degradation		Time in Power Saving Mode (S)		Performance Degradation		Time in Power Saving Mode (S)	
D	Sim.	Est.	Sim.	D	Sim.	Est.	Sim.
10	22	1.07	1.07	10	22	1.01	1.13
30	36	5.56	5.65	30	31	3.24	3.99
50	54	13.79	13.76	50	56	4.35	4.35
100	114	23.34	23.16	100	100	12.03	11.97

Table 3: Power savings estimated using our methodology (columns "Est.") and simulation (columns "Sim.") for trace "Code 1" and power savings Levels 3 and 4. The target performance degradation D is also reported, together with the achieved degradation (column "Sim."). All results are in (%).

Level 3				Level 4			
Performance Degradation		Time in Power Saving Mode (S)		Performance Degradation		Time in Power Saving Mode (S)	
D	Sim.	Est.	Sim.	D	Sim.	Est.	Sim.
10	12	0.04	0.04	10	18	0.01	0.01
30	28	4.24	4.24	30	31	0.04	0.04
50	51	8.57	8.57	50	58	0.10	0.10
100	95	15.12	15.12	100	109	5.95	5.95

Table 4: Power savings estimated using our methodology (columns "Est.") and simulation (columns "Sim.") for trace "Code 2" and power savings Levels 3 and 4. The target performance degradation D is also reported, together with the achieved degradation (column "Sim."). All results are in (%).

Level 3				Level 4			
Performance Degradation		Time in Power Saving Mode (S)		Performance Degradation		Time in Power Saving Mode (S)	
D	Sim.	Est.	Sim.	D	Sim.	Est.	Sim.
10	16	0.33	0.33	na	na	na	na
30	37	2.00	2.00	30	27	0.16	0.16
50	57	3.28	3.27	50	59	1.14	1.14
100	101	6.06	6.05	100	115	2.71	2.70

Table 5: Power savings estimated using our methodology (columns "Est.") and simulation (columns "Sim.") for trace "File 1" and power savings Levels 3 and 4. The target performance degradation D is also reported, together with the achieved degradation (column "Sim."). All results are in (%).

Level 3				Level 4			
Performance Degradation		Time in Power Saving Mode (S)		Performance Degradation		Time in Power Saving Mode (S)	
D	Sim.	Est.	Sim.	D	Sim.	Est.	Sim.
10	11	4.56	4.56	na	na	na	na
30	34	10.08	10.07	30	31	4.70	4.70
50	50	12.46	12.46	50	51	7.53	7.53
100	124	17.82	17.82	100	100	11.35	11.34

Table 6: Power savings estimated using our methodology (columns "Est.") and simulation (columns "Sim.") for trace "File 2" and power savings Levels 3 and 4. The target performance degradation D is also reported, together with the achieved degradation (column "Sim."). All results are in (%).

longer idle intervals. However, the longer tail in the distribution of idle times of "Code 1" than in the distribution of "File 1" enables higher power saving in "Code 1", because power saving benefits significantly from the existence of long idle interval. We conclude that *single metrics such as utilization levels are not sufficient in indicating power saving capabilities for a given workload*. Our lightweight methodology, seamlessly incorporates many metrics that allow for an accurate estimation of power savings without violating performance targets.

While the results in Tables 3, 4, 5, and 6 show what portion of the time is utilized for power savings, the actual power savings are estimated using the data in Table 1. Specifically, actual power savings for Level 4 are about 40% higher than Level 3. However, for the enterprise environments that are represented by the four traces under consideration, Level 3 provides higher power savings than Level 4 for the performance targets used here. For the very low performance degradation target of 10% and traces "File 1" and "File 2" the estimation procedure did not find an appropriate scheduling pair (I, T) that would not violate the target under Level 4 power saving mode, which has the 1000ms penalty. We mark those cases in the respective tables as "na"

The estimated power savings in Tables 3, 4, 5, and 6 are among the highest possible power savings for the traces for a given performance target. We confirm this by exploring the entire scheduling state space of (I, T) pairs for "Code 2". The results for Level 3 power saving mode are given in Figure 1, in the motivation example presented in the Introduction Section, while the results for Level 4 power saving mode are given in Figure 4. In both figures, the lighter shades of color indicate better performance and more power savings. It is clear that there are limited regions where power savings are high and performance targets are met. Nevertheless, our methodology robustly identifies such regions even for the challenging case depicted in Figure 4 (with very few opportunities for power savings).

We also plot the distribution of delays in cause on foreground requests attributed to the disk drive being put to a power saving mode in Figure 5 for power saving mode level 3. The plots of Figure 5 show that although the average response time slowdown may be as high as 100%, the percentage of requests that experience delays accounts for only *a a small percentage* of the overall number of requests. For example, in trace "Code 1", even for response time target slowdowns as high as 100, the percentage of affected requests is always less than 2%. The cumulative distribution of delays attributed to power savings for the four traces further makes the point of the robustness of the framework.

5. WORKLOAD SHAPING

In storage systems that deploy from tens to thousands of disk drives, the power saving potential of individual disks is enhanced by further shaping their workload. The goal of workload shaping with the goal of improving power savings in the entire cluster is to re-direct part of the workload intended for one disk drive to another. Because the disk drive receives less workload that initially intended, its idle periods are now extended as are the opportunities for power savings on that drive.

Various workload shaping techniques have been proposed in the literature and deployed in high-end storage systems and data centers. For example, in [14] it was proposed to redirect the WRITE traffic received by enterprise-level arrays to other arrays in the storage cluster or data center to free up the disk drives and possibly spin them down for power savings. Particularly for backup and archival storage systems, where the workload is significantly less intensive than in enterprise systems, ways of re-directing the entire workload

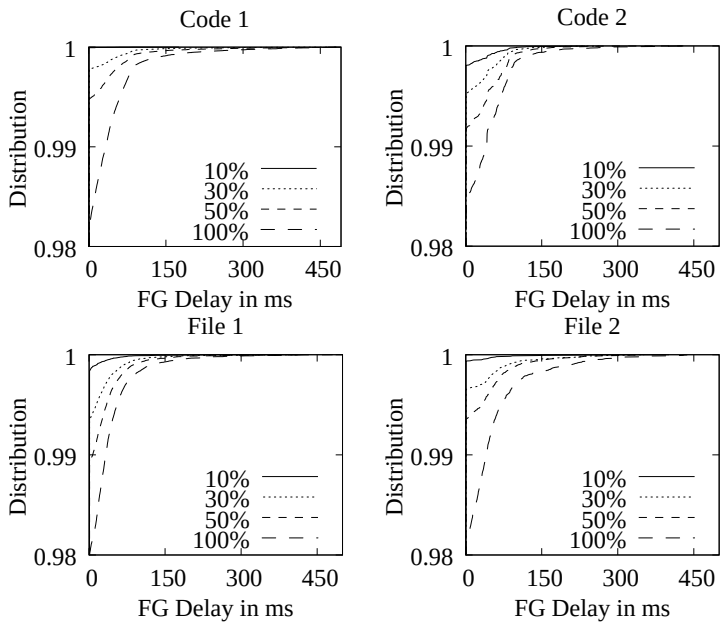

Figure 5: Cumulative distribution of delays in user requests attributed to power savings under level 3 for degradation targets 10, 30, 50, and 100 percent.

seen by disks or arrays to other parts of the system are explored [2]. In such cases, portions of data are moved or copied to other parts of the storage system before the disks or arrays are spun down.

For storage systems, it is critical to have an estimation of the efficiency of the workload shaping techniques for the current system workload, because each of them imposes extra work on the system. For example, WRITE offloading [14] requires write operations to be redirected to another disk/array (adding extra work to them) and after the power saving mode is over, copying the modified data into their original destination.

As explained in the previous sections, the potential for power savings in a disk drive depends on many factors, including tolerance for performance degradation on disk request and the distribution of idle intervals. Offloading part of the workload from one disk to another will certainly increase idleness in the disk drive, but that may not result in significant power savings to justify the added work and complexity in the system. Specifically, knowing the percentage of the workload that will be re-directed from one disk to another cannot determine the efficiency of the technique. It is the impact that this redirection has on the distribution of idle times that eventually determines the effectiveness of workload shaping.

The methodology of Section 3 can be used effectively to evaluate the efficiency of the workload shaping techniques for the current system workload before any of them takes effect in the system. For that, one needs to monitor in addition to the distribution of idle times for the current workload also the distribution of idle intervals if a given workload shaping technique would be in place. For example, in the case of WRITE offloading [14], the distribution of idle intervals in the workload without the WRITEs (and the respective READs) can be constructed and be used to estimated the respective potential power savings by feeding it to the estimation methodology of Section 3.

Here we evaluate two general workload shaping techniques, namely WRITE offloading and READ offloading. The focus of this paper is not on optimizing these workload shaping techniques and the parameters associated with them. Instead, we aim to estimate the outcomes of a given approach for a workload shaping technique before it takes effect in the storage system. The WRITE offloading and READ offloading analyzed here are defined as follows

- **WRITE offloading**: all WRITEs, and any subsequent READs of the same data that arrive to a disk drive are re-directed somewhere else in the storage system.

- **READ offloading**: the most READ-accessed locations of the disk, and any subsequent WRITEs on the same locations, are copied to another disk drive in the storage system.

With regard to READ offloading, we assume that a 10 GBytes of buffer space is reserved for re-directed traffic in another storage device in the system (not cache). We believe that 10 GBytes of data will not present capacity or performance overhead to storage devices and/or arrays in the system. We stress that coming up with the right buffer size is outside the scope of this paper.

5.1 Effectiveness of workload shaping

The approaches that we consider for WRITE and READ offloading can be used easily to modify a given disk workload to generate the workload that the disk would see if the shaping technique was in effect. We apply both WRITE and READ offloading in all four traces that we have used so far in our evaluation, i.e., “Code 1”, “Code 2”, “File 1”, “File 2”, and construct the modified distribution of idle times for each trace and workload shaping technique. The resulting distributions of idle times are shown in Figure 6.

The two workload shaping techniques change the idleness in the system in different ways. Specifically, the distribution of idleness for trace “Code 1” is affected slightly only by READ offloading, while the WRITE offloading has barely any effect on it. The same behavior is observed also for trace “File 1".

On the other hand, READ offloading has a significant impact on the distribution of idle times for trace “Code 2”. The tail of the distribution in this case is extended by an order of magnitude. However READ offloading for trace “Code 2” causes the onset of many short idle intervals, which shorten the body of the distribution. Later in this subsection, we elaborate more on the impact

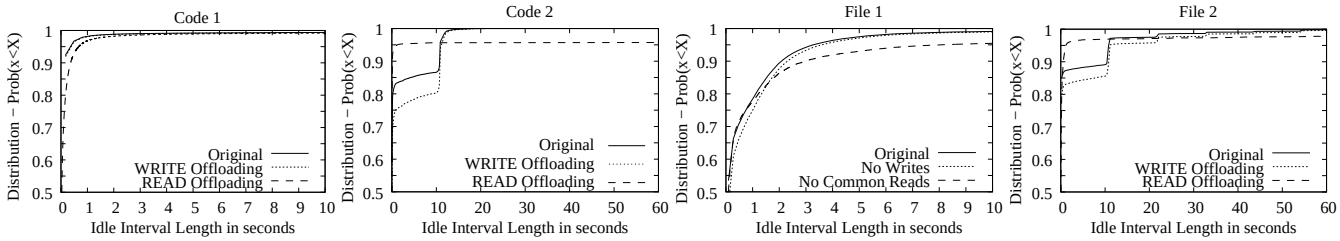

Figure 6: Distribution of idle interval lengths for the four traces and different workload shaping techniques.

that such transformation in the distribution of idle times has on the power saving capabilities in the disk drive. In contrary to the effect that READ offloading has on the distribution of idle times for "Code 2", WRITE offloading for "Code 2" extends idleness in the body of the distribution rather than the tail. Similar effects as those observed in trace "Code 2" are also observed in trace "File 2".

For the rest of the section, we focus only on traces "Code 1", "Code 2", and Level 3 power saving mode, traces "File 1", "File 2" and Level 4 power saving mode yield qualitatively similar results. In Tables 7 and 8, we show the power saving estimations for four levels of performance degradation (10%, 30%, 50%, and 100%) and the respective simulated performance for both WRITE offloading and READ offloading and traces "Code 1" and "Code 2", respectively.

WRITE Offloading				READ Offloading			
Performance Degradation		Time in Power Saving Mode (S)		Performance Degradation		Time in Power Saving Mode (S)	
D	Sim.	Est.	Sim.	D	Sim.	Est.	Sim.
10	25	2.99	2.99	10	16	3.47	3.47
30	35	4.65	4.65	30	32	15.92	15.92
50	52	14.72	14.72	50	54	15.96	15.96
100	99	28.32	28.32	100	117	22.24	22.24

Table 7: Estimated power savings for WRITE and READ offloading using our methodology (columns "Est.") and simulation (columns "Sim.") for trace "Code 1" and Level 3 power mode. The target performance degradation D is also reported, together with the achieved degradation (column "Sim."). All results are in (%).

For both traces, WRITE offloading is not effective when compared to the benefits of power savings mode for the original traces (see Tables 3 and 4), as it only changed slightly the distribution of idle times. Even the visible change in the body of distribution for trace "Code 2" and WRITE offloading seems to improve power savings only for medium slowdown (i.e., 30%).

READ offloading shows significant gains for "Code 2": for 100% degradation in delays the system is 50% of the time in power saving mode (see Table 8). We stress that under READ offloading trace "Code 2" becomes extremely idle, but it is the large number of very short intervals (see the very short body of the distribution in Figure 6) that does not allow to exploit more of that idleness for power savings.

In Figure 7, we explore the entire scheduling state space (i.e., the entire set of (I, T) pairs) by plotting the corresponding performance slowdown and time in power saving mode for "Code 2", Level 3 mode, and READ offloading. The light color shades indicate better performance or mode time in power saving. Certainly, the plots in Figure 7 capture a very different behavior from the same plots in Figures 1 and 4 where results for the original trace

WRITE Offloading				READ Offloading			
Performance Degradation		Time in Power Saving Mode (S)		Performance Degradation		Time in Power Saving Mode (S)	
D	Sim.	Est.	Sim.	D	Sim.	Est.	Sim.
10	10	0.90	0.90	10	3	18.47	18.47
30	31	5.23	5.23	30	15	39.62	39.62
50	49	9.75	9.75	50	83	44.58	44.58
100	82	15.53	15.53	100	89	50.46	50.46

Table 8: Estimated power savings for WRITE and READ offloading using our methodology (columns "Est.") and simulation (columns "Sim.") for trace "Code 2" and Level 3 power mode. The target performance degradation D is also reported, together with the achieved degradation (column "Sim."). All results are in (%).

"Code 2" are plotted. Although the power savings opportunities for trace "Code 2" are significant, because of the very short body and very long tail of the distribution of idle times, little changes in the values of the (I, T) may cause drastic change in system performance. With our methodology, we are able to identify scheduling pairs that achieve the desired performance while increasing time in power saving mode.

The results presented in this section are indicative of how the workload may be shaped for power savings. Simulation and performance degradation/power saving maps were also done for all four traces and power modes but are not reported here due to lack of space. We stress that the results presented here are representative of all experiments.

6. RELATED WORK

There is a host of power saving methodologies in the storage systems/disk drives. From these works we first discuss those that investigate the effectiveness of multi-speed disks for power savings [1, 6, 23]. In [1] the authors advocate the use of multi-speed disks where each disk is slowed down to reduce energy consumption during low load periods and show that this method can provide power savings up to 23% for web and proxy servers. Dynamically setting the rotation speed in disk drives is proposed as a low-level hardware-based technique to save power within a drive, because the faster the disk drive spins the more power it consumes [6]. Several of the power savings techniques in storage systems and devices, including disk drives that rotate at different speeds and migration of data to the most feasible set of disk drives, are evaluated collectively in the Hibernator framework [23].

Power conservation by selectively spinning up or or down selected sets of disks has been explored first in mobile environments [3, 7] but has been also considered in large data storage archives where data is accessed infrequently. The Massive Array of Inexpensive Disks [2] borrows ideas from cache management to spin up se-

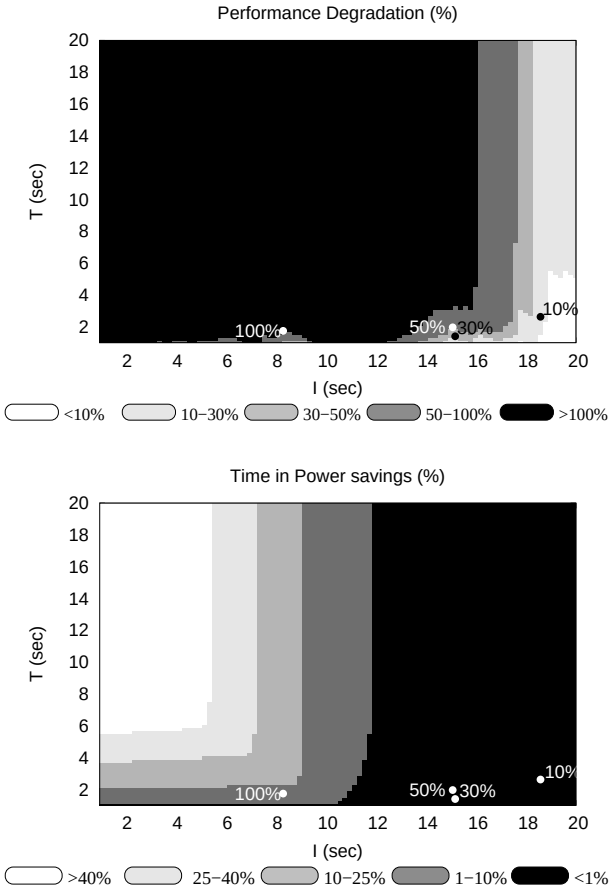

Figure 4: Exploration of the entire scheduling space as performance vs. savings for Level 4 mode and trace "Code 2". Results of our scheduling framework are marked. As the color shade becomes lighter both performance and time in power savings increase.

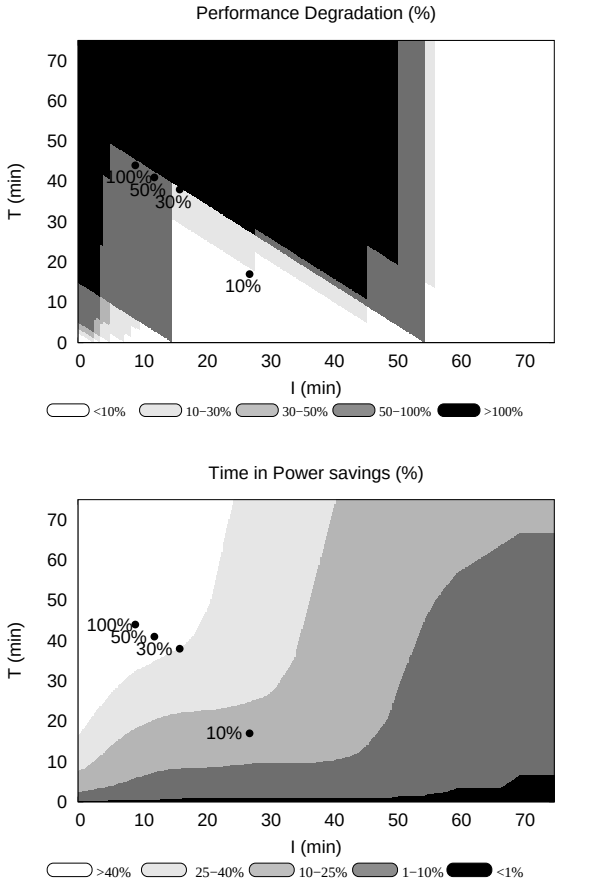

Figure 7: Exploration of the entire scheduling space as performance vs. savings for Level 3 mode, trace "Code 2" and READ offloading. Results of our scheduling framework are marked. As the color shade becomes lighter both performance and time in power savings increase.

lected disks that act as an LRU cache and selectively spins down subsets of inactive disks[2]. A technique called PDC that migrates frequently accessed data to a few disks allows the rest of the disks to be lowly loaded and be put on a low-power mode is proposed and power savings of PDC and MAID are more substantial when two-speed disks are used[15]. Power-aware cache management policies in data centers are considered in [24].

Redundancy has been also explored as a techniques to save energy in storage systems. EERAID [10] and eRAID [11] focus on RAID 1 and RAID 5 systems and achieve savings by request scheduling and cache replacement policies at the RAID controller. RIMAC [19] consider RAID 5 systems and achieve energy savings by exploiting parity redundancy in parity-based redundant disk arrays. Diverted Accesses have been proposed in [16] and implement redundancy driven by analytic models that quantify energy savings of difference redundancy configurations [16].

Data migration between disks in order to create hot data on a few disks has been examined in [15] and has been also exploited in the form of write off-loading in [14]. FS2 contains a runtime component responsible for dynamically reorganizing disk layout in order to improve disk performance and save power by reducing seek time and rotational delays [9]. Algorithms that explore relationships among accessed data to improve latency while reducing energy by decreasing disk arm movement [5]. PARAID uses a skewed striping pattern to adapt to the system load and varies the number of powered disks [20]. Changing the striping parameters, i.e., on which disks to stripe and the stripe unit has been proposed effective in high-performance environments [22]. Data reorganization and disk mapping algorithms that target scientific applications have shown promise for reducing energy consumption [21]. The bursty nature of storage workloads is exploited in [12] to reduce power consumptions in storage systems while maintaining QoS guarantees.

In this paper we focus on power savings at the disk drive level. The work presented here differs from the above works in that it provides an analytic methodology for *predicting* power savings while user performance remains quantifiable and any degradation is bounded. Given a disk workload, we demonstrate that the actual (low) disk utilization levels are not the only indicators of the viable power savings but it is instead the stochastic characteristics and the composition of the workload that dictate the trade off between power savings and user-perceived performance. The analytic model that is proposed here effectively quantifies *a priori* the amount of possible power savings given a fixed allowable user delay (or alternatively, the expected user delay given a desirable amount of saved power). To the best of our knowledge, this is the first work that success-

fully bounds performance while quantifying power savings with a surprisingly effective analytic model.

7. CONCLUSIONS

We have presented a simple analytic model and its integration into an algorithmic framework that provides the following: given a performance target for the responsiveness of the storage system, it provides answers to the following difficult questions: “when" and for “how log" the system should be put in a power saving mode during idle times. Our results also illustrate that power savings in storage environments is not easy and perhaps counter-intuitive: we have shown traces of very low average disk utilization to not be fit for power savings. It is the entire distribution of idle times (rather than simple measures such like average idleness) that controls the ability to save power. We use a light-weight way to capture the distribution of idle times in the form of a histogram and use this histogram for autonomically determine (as well as predict) the possible power savings given a user-provided performance degradation target. The framework is robust, lightweight, and adaptive because it is based on a workload histogram that continuously adapts to changes in the monitored workload, providing thus a powerful way for autonomically identifying all opportunities for power savings and performance without any prior knowledge of future workload.

Acknowledgments

This work is supported by NSF grants CCF-0811417 and CCF-0937925. The authors thank Seagate Technology for providing the enterprise traces used for this work. A preliminary version of this paper (non-copyrighted) appeared in HotMetrics 2009.

8. REFERENCES

[1] E. Carrera, E. Pinheiro, and R. Bianchini. Conserving disk energy in network servers. In *Proceedings of ICS 2003*, pages 86–97, 2003.

[2] D. Colarelli and D. Grunwald. Massive arrays of idle disks for storage archives. In *Proceedings of the ACM/IEEE Conference on Supercomputing (SC)*, pages 1–11, 2002.

[3] F. Douglis, P. Krishnan, and B.N. Bershad Adaptive disk spin-down policies for mobile computers. In *Proceedings of the 2nd USENIX Symposium on Mobile and Location-Independent Computing*, pages 121–137, 1995.

[4] L. Eggert and J. D. Touch. Idletime scheduling with preemption intervals. In *Proceedings of the 20th ACM Symposium on Operating Systems Principles (SOSP)*, pages 249–262, 2005.

[5] D. Essary and A. Amer. Predictive data grouping: Defining the bounds of energy and latency reduction through predictive data grouping and replication. *ACM Transactions on Storage*, 4(1):1–23, 2008.

[6] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, and H. Franke. DRPM: Dynamic Speed Control for Power Management in Server Class Disks. In *Proceedings of the Annual International Symposium on Computer Architecture (ISCA)*, pages 169–180, 2003.

[7] D.P. Helmbold, D.D.E. Long, T.L. Sconyers, and B, Sherrod. Adaptive disk spin-down for mobile computers. MONET 5(4), pages 285-297, 2000.

[8] Hitachi Global Storage Technologies. Power and acustics management. White paper at http://www.hitachigst.com, 2007.

[9] H. Huang, W. Hung, and K. G. Shin. FS2:dynamic data replication in free disk space forimproving disk performance and energy consumption. In *Proceedings of the 20th ACM Symposium on Operating Systems Principles (SOSP)* pages 263–276, 2005.

[10] D. Li and J. Wang. EERAID: energy efficient redundant and inexpensive disk array. In *Proceedings of the 11th ACM SIGOPS European workshop*, pages 29, Sept. 2004,

[11] D. Li and J. Wang. Conserving Energy in RAID Systems with Conventional Disks. In *Proceedings of the 3rd International Workshop on Storage Network Architecture and Parallel I/Os*, Sept 2005.

[12] L. Lu and P. Varman Workload Decomposition for Power Efficient Storage Systems. In *Proceedings of the First Workshop on Power Aware Computing and Systems (HotPower 2008)*, 2008.

[13] N. Mi, A. Riska, X. Li, E. Smirni, and E. Riedel. Restrained utilization of idleness for transparent scheduling of background tasks. In *Proceedings of the joint ACM SIGMETRICS/Performance Conference*, pages 205–216, 2009.

[14] D. Narayanan, A. Donnelly, and A. I. T. Rowstron. Write off-loading: Practical power management for enterprise storage. In *Proceedings of the USENIX Conference on File And Storage Technologies (FAST))*, pages 253–267, 2008.

[15] E. Pinheiro and R. Bianchini. Energy conservation techniques for disk array-based servers. In *Proceedings of the Annual International Conference on Supercomputing (ICS)*, pages 68–78, 2004.

[16] E. Pinheiro, R. Bianchini, and C. Dubnicki. Exploiting redundancy to conserve energy in storage systems. In *Proceedings of SIGMETRICS/Performance 2006*, pages 15–26, 2006.

[17] A. Riska and E. Riedel. Disk drive level workload characterization. In *Proceedings of the USENIX Annual Technical Conference*, pages 97–103, May 2006.

[18] Seagate Technology. Constellation ES: High capacity storage designed for seamless enterprise integration Product overview at http://www.seagate.com, 2009.

[19] X. Yao and J. Wang. RIMAC: A Redundancy-based, Hierarchical I/O Architecture for Energy-Efficient Storage Systems. In *Proceedings of the 1st ACM EuroSys Conference*, Apr 2006.

[20] C. Weddle, M. Oldham, J. Qian, A. Wang, P. Reiher, and G. Kuenning. PARAID: A Gear-Shifting Power-Aware RAID In ACM Transactions on Storage, 7(3), page 13, 2007.

[21] S. W. Son and M. Kandemir. Integrated data reorganization and disk mapping for reducing disk energy consumption. In *Proceedings of Seventh IEEE International Symposium onCluster Computing and the Grid*, pages 557 - 564, 2007.

[22] S. W. Son, G. Chen, and M. Kandemir. Disk layout optimization for reducing energy consumption. In *Proceedings of the 19th Annual International Conference on Supercomputing ICS 2005*, pages 274–283, 2005.

[23] Q. Zhu, Z. Chen, L. Tan, Y. Zhou, K. Keeton, and J. Wilkes. Hibernator: helping disk arrays sleep through the winter. In *Proceedings of the ACM Symposium on Operating Systems Principles (SOSP)*, pages 177–190, 2005.

[24] Q. Zhu and Y. Zhou. Power-aware storage cache management. *IEEE Transactions on Computers*, 54(5):587–602, 2005.