Feasibility Regions: Exploiting Trade-offs between Power and Performance in Disk Drives*

Alma Riska[1] Ningfang Mi[2] Giuliano Casale[3] Evgenia Smirni[2]
[1] Seagate Research, alma.riska@seagate.com
[2] College of William Mary, {ningfang,esmirni}@cs.wm.edu
[3] SAP Research Belfast, giuliano.casale@sap.com

ABSTRACT

Low utilization immediately suggests that placing the system into a low power mode during idle times may considerably decrease power consumption. As future workload remains largely unknown, “when” to initiate a power saving mode and for “how long” to stay in this mode remains a challenging open problem, given that performance degradation of future jobs should not be compromised. We present a model and an algorithm that manages to successfully explore feasible regions of power and performance, and expose the system limitations according to both measures. Extensive analysis on a set of enterprise storage traces shows the algorithm’s robustness for successfully identifying “when” and for “how long” one should activate a power saving mode given a set of power/performance targets that are provided by the user.

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 keep consuming disproportionally high power resources.

Idle periods in disks of low utilization offer opportunities for saving power in a straight forward manner: one could put the disk in a low power mode during idle times [4]. Yet, this should be done transparently to the end user: requests that arrive while the disk is in a power saving mode are to be inevitably delayed as the system requires a recovery time before the disk is mechanically set to a state that allows serving 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 to within predefined limits.

In this paper, we present a solution to this problem leveraging on a schedulability framework that is initially proposed for scheduling background jobs in disk drives [3]. This framework relies on the stochastic characteristics of idle intervals and the anticipated duration of background jobs (background jobs are considered non-preemptable) to best serve them within idle periods. The performance degradation of foreground jobs is regulated by an input parameter furnished by the user.

The schedulability framework presented in [3] is used here to create a robust power saving prediction methodology that uses a selection mechanism to determine *which* idle intervals should be utilized for saving power. We express power savings within an idle interval as a function of two parameters: the time I that elapses in 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. By *exhaustively* exploring these two parameters, it is possible to construct a figure that looks similar to a geographic map (see Figure 1 for an example of a disk in a file server). The map draws regions of different levels of performance slowdown (top plot) and power savings (bottom plot) as a function of I and T. By looking at these maps, one can immediately identify an (I, T) pair that achieves both performance and power saving targets. Creating these maps though is computationally expensive as it requires running one simulation for every (I, T) pair to plot the results.

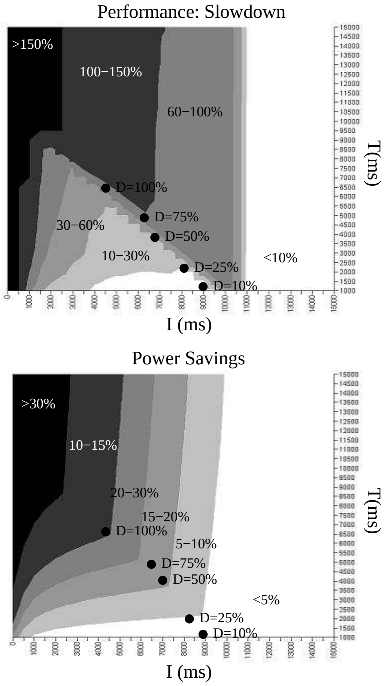

Figure 1: Performance slowdown (top) and power savings potential (bottom) in a test case (a disk in a file server).

The novelty of this work is the accurate identification of an (I, T) pair that is located in a *feasible region* within the maps. This (I, T) pair corresponds to the user-defined trade-off between performance and power savings, given the system targets. Given an acceptable average response time delay D as an input parameter, the framework provides an (I, T) pair *and* the average power saving that

*This work is supported by NSF grants CNS-0720699 and CCF-0811417.

can be achieved with this (I, T). Conversely, given a target power saving, the system can provide the average response time delay that must be tolerated. This allows the user to select the best operating mode as well as assess the limitations of the system.

We stress that the user does not have to exhaustively explore all parameters to create the power/performance map. Instead, our modeling framework manages to quickly identify the target regions *without* having to create the map. Given a predefined target D and a power saving mode, our framework outputs an (I, T) pair. The framework's output is consistency with the best possible choices. Indeed, in Figure 1 the various D markings identify the (I, T) pairs that are suggested by the framework and all lie within the best region for the noted foreground slowdown target. The significance of the framework is that it is compact and introduces minimal overhead for monitoring system metrics and the actual estimation procedure.

In addition to the example in Figure 1, we illustrate the robustness of this modeling framework via trace driven simulations using three 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.

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 Section 4 using trace-driven analysis and simulations. Conclusions and future work are given in Section 5.

2. POWER SAVINGS IN STORAGE SYSTEMS

There is a host of power saving methodologies in the storage systems/disk drives literature including algorithms that explore relationships among accessed data to improve latency while reducing energy by decreasing disk arm movement [2], use of multi-speed disks [7] that operate on different spin rates depending on the intensity of the workload, and selectively spinning up or down subsets of disks in large storage systems borrowing ideas from cache management [1]. Data migration between disks in order to create hot data on a few disks has been examined in [5] and has been also exploited in the form of write off-loading in [4].

Disk drives consist of several mechanical components such as the read/write heads (recording arm) which fly (at a very precise distance) over the continuously rotating magnetic media platters. Power can be saved in a disk drive by stopping or slowing down any of the components. There are several levels of power consumption in disk drives depending on the disk components that are active and operational. Unfortunately, when drive components are shut down, it takes some time to bring them back up and to be 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 determine how much power is consumed and how much power can be saved in any power saving mode. Below we list all the levels of power savings in a disk drive and the respective expected savings and penalties.

- **Level 1**: the drive is serving requests and it consumes power depending on the workload characteristics, such as sequential/random, and READs/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 in the system the workload is managed such that the drive goes in 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 (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 (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 any more. 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 (Level 2). The penalty to reload the heads and turn on the motor to rotate the platters is about 8 seconds.

- **Level 6** the drive is spun down entirely cutting the power consumption almost entirely, but bringing the disk back up takes as much as 25 seconds.

Among the above levels of power savings, we are interested in those that have smaller penalties such as levels 3 through 5. We capture their respective power savings and time-to-ready penalties in Table 1.

	Power savings relative to "active idle"	Time to active
Level 3	18%	0.5 sec
Level 4	30%	1 sec
Level 5	50%	8 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 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, however, is left to the overall system management unit, because it depends on how sensitive the system is to performance degradation.

3. ALGORITHMIC FRAMEWORK

The utilization of disk drives, even in demanding enterprise environments, is low to medium. In particular, disk drives that are being used for back-ups and archiving (e.g., low-end enterprise systems) are accessed only occasionally, and because of the large amount of data in archives and back-up systems, there is a massive amount of disks with very low utilization, which can be exploited for saving power [1]. However, with the explosion of the on-line data centers that support high-end enterprise systems, it may be desirable to exploit power savings opportunities even in such a non-traditional domain. The issue though is that power savings in disk drives may cause significant delay to some of the requests, if done haphazardly. While delays may be acceptable for archival systems, they are certainly not desirable for high-end systems.

Here we first give an overview of the algorithmic framework in [3] and show how it can be adapted for power savings. Pivotal for the success of the methodology is monitoring of the current system workload. Specifically, the framework monitors (1) the length of idle intervals and constructs their corresponding continuous data histogram[1], and (2) the response time of user requests, and uses as user input the acceptable slowdown in the user request performance attributed to the background jobs.

Based on the above information, the system determines "when" and for "how long" an idle interval can be used for background work. Naturally, the above scenario can be adapted for power savings: the background job is the time the disk drive or any of its components is shut down to preserve power. The penalty of the background jobs is the time it takes the disk drive or its components to become active, based on the selected level of the power conservation. The acceptable slowdown in performance depends on the system. It is expected that an archival system has an acceptable slowdown larger than a file server or database server.

The framework is general such that it may be used to optimize for different metrics in a system that serves background jobs. The main goal is to control the performance degradation in the system close to a pre-defined target. Secondary goals are to maximize the amount of background work served and/or the service rate of the background work. In the case of power savings, the system needs to control the degradation in performance while *maximizing* the amount of time the disk or its components are turned off.

Instead of monitoring the incoming workload and its characteristics, we monitor the idle intervals that result in the system while it serves that workload. As a result, we reduce the complexity and the overhead of the estimation procedure. Furthermore, because the histogram of idle times is the main data structure, it contributes to the accuracy of the framework as the actual performance degradation and background work completed are always close to the estimated ones.

The outcome of the framework is the pair (I, T), where I indicates when to initiate a power saving mode at the disk and T indicates for how long to keep the drive in that power saving mode. One of the strengths of this framework is the ability to estimate various performance metrics using the histogram of idle times, particularly the amount of work B completed during idle intervals. Here, the amount of work is measured by time. Therefore, B is also referred to the amount of time in power saving mode.

In the case of power savings, the estimation of the useful amount of time that the disk stays in a power saving mode is different from the common background tasks, because the time P that it takes the disk drive to get out of the power saving mode is included in T and cannot be accounted for power savings. The amount of time B in 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 falls between I and $I + T - P$, where the amount of time in power savings mode is simply $R - I$,

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

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.

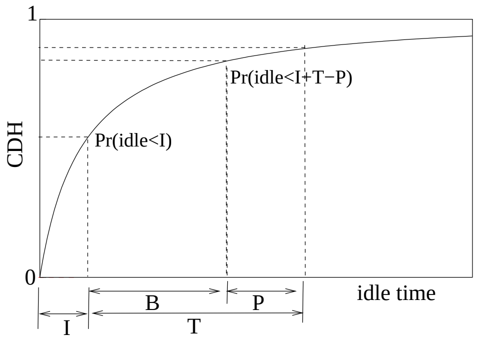

Figure 2: Estimation of the amount of time B that the disk stays in the power saving mode with penalty P which starts after I units of idle time have elapsed and ends T time units later.

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

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

where $Pr(i)$ is the probability of an idle interval being of length i and max is the maximum length of an idle interval in the system. 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, and although not every idle interval is utilized for power savings. To estimate the amount of power savings S over the period of time $Time$, we use the following relation

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

Eq. 2 enables the estimation of power savings for every power saving mode given the current workload in the system (as captured by the idle times histogram). For different power saving modes there are different penalties P and as a result also different pairs (I, T) that are the output of the framework. The power savings estimates that are obtained from Eq. 2 are associated with a given performance slowdown in the system.

Different (I, T) pairs can be computed for different performance slowdown targets, i.e., the (I, T) pairs are the independent variables of this analysis. For each pair (I, T), the corresponding power savings are also estimated using the data from Table 1 and Eq. 2. Such estimation can be done for each power mode. We remark that different (I, T) pairs are given distinctly for the different power modes. The result is a set of power savings and performance slowdowns, and the system can decide which one to utilize based on its priorities. Our methodology, not only estimates the maximum power savings for a given workload but also shows how to achieve them, i.e., when and for how long to initiate a power saving mode, but also which power saving mode to utilize.

As we will show in the evaluation section, our methodology finds accurately any power savings opportunities that exist in the system based on the current workload. Our methodology is flexible and does not use rigid thresholds that may cause either significant delays or unnecessary consumption of power.

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

[1] Here, we refer the reader to [3] for the details about how to online construct the histogram of idle times.

at the disk level of two enterprise storage systems, an application development server ("Code") and a file server ("File") [6]. These traces record the arrival time, the departure time, the type of each request, their length, and the position 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. 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.

Trace	Length (hrs)	Mean Resp.	Util (%)	Idle Length		Saving (%)	
				Mean	CV	Lev. 3	Lev. 4
Code 1	12	8.6	5.6	193	8.4	10	15
File 1	12	12.7	1.7	767	2.3	13	16
File 2	12	15.3	0.7	2000	3.8	17	28

Table 2: **Seagate trace characteristics: measurements are in milliseconds unless otherwise noted. The "Saving" columns indicate the bound on power savings under Level 3 and 4, *if we have perfect knowledge of the duration of idle intervals*.**

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 (several times the idle interval mean) which need to be exploited for power savings, particularly for trace "File 2". Trace "File 1" also indicates opportunities for power savings when compared to trace "Code 1", but as we show later in this section, this is not the case.

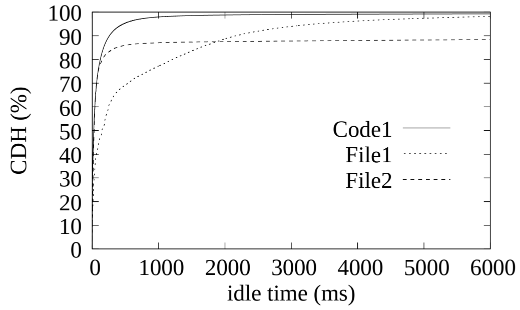

Figure 3: **Histogram of idle times for our traces.**

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 like the one we are evaluating in this section. For our evaluation, we use the methodology laid out in Section 3 to estimate the potential power savings under each workload (trace), an acceptable performance slowdown, and a power saving mode. We first use the framework to identify the appropriate (I, T) pair and then run a trace driven simulation that puts the system in a power saving mode as guided by the selected (I, T) values. We also compare the estimated results with the simulated ones.

We show these results in Tables 3, 4, and 5 for traces "Code 1", "File 1, and "File 2", respectively. For easy comparison, the simulation results are shown in parenthesis, next to the model's prediction. Specifically, we show

FG Resp. Slowdown: the slowdown in average foreground response time attributed to power savings (an input parameter, guaranteed to be met in our methodology),

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, and 5 show that our methodology estimates well the amount of time that the system under the given workload can be put in a power saving mode for the purpose of power savings. Note that the performance/power maps shown in Figure 1 correspond to trace File 1 and the (I, T) pairs that we use for saving power are also marked on Figure 1. The results from the simulations match reasonably well the estimated ones.

The estimated results in Tables 3, 4, and 5 are among the best possible trade-offs between potential power savings and performance slowdown. We confirm this by exploring the entire state space of (I, T) pairs. For trace "File 1", we present the state exploration in Figure 1. In Figure 1, we show that we identify the region in the map that gives the largest amount of power savings while meeting the performance targets.

One counter-intuitive observation in the results of Tables 3, 4, and 5 is that trace "Code 1" holds better power savings potential than trace "File 1" although the latter has more available idle time and generally longer idle intervals. However, the longer tail in the distribution of idle times of trace "Code 1" enables better power savings with long idle interval requirements. Most importantly, our methodology is able to identify these opportunities correctly because the decisions are made based on the histogram of idle times which captures correctly and efficiently distribution tails.

Level 3		Level 4	
FG Resp. Slowdown (D)	Time in Power Saving Mode	FG Resp. Slowdown (D)	Time in Power Saving Mode
10 (12)	6.80 (5.68)	10 (13)	2.06 (3.36)
15 (18)	10.02 (9.32)	15 (18)	3.31 (6.88)
50 (63)	21.93 (24.73)	50 (56)	12.96 (11.72)
100 (106)	27.46 (32.09)	100 (141)	20.17 (20.98)

Table 3: **Estimated performance under trace Code 1, under power savings Levels 3 and 4. The values presented in parentheses are the results obtained from the trace-driven simulations. All results are in (%).**

Level 3		Level 4	
FG Resp. Slowdown (D)	Time in Power Saving Mode	FG Resp. Slowdown (D)	Time in Power Saving Mode
10 (5)	1.11 (0.48)	10 (10)	0.17 (0.14)
15 (11)	1.85 (1.25)	15 (16)	0.17 (0.14)
50 (51)	5.66 (4.96)	50 (71)	2.27 (2.34)
100 (103)	9.08 (8.22)	100 (134)	4.85 (4.93)

Table 4: **Estimated performance under trace File 1, under power savings Levels 3 and 4. The values presented in parentheses are the results obtained from the trace-driven simulations. All results are in (%).**

We also analyze the distribution of delays in user requests attributed to power savings and plot them in the plots of Figures 4 and 5 for power savings level 3 and 4, respectively. The figures show that although the average response time slowdown may be high, the percentage of penalized requests is *very small*. For example, in trace Code 1, even response times target slowdowns are as

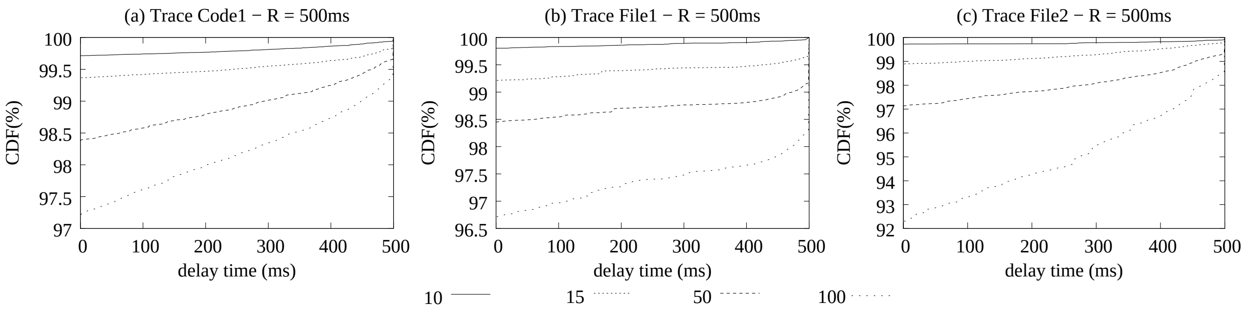

Figure 4: Distribution of delays in user requests attributed to power savings under level 3 for slowdown targets equal to 10, 15, 50, and 100.

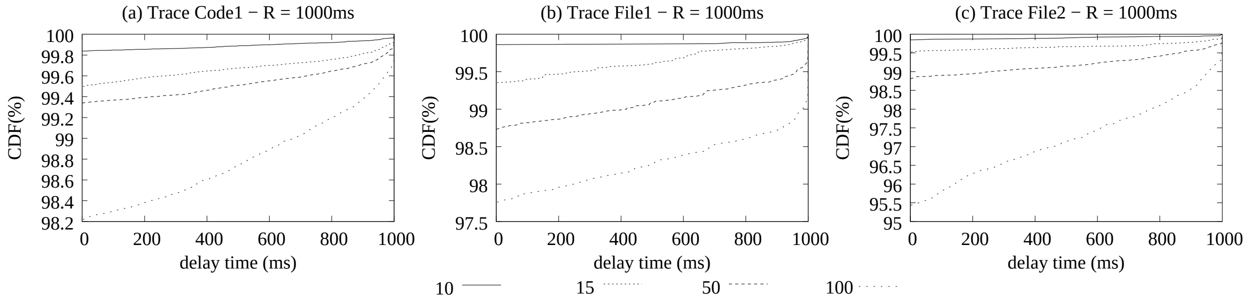

Figure 5: Distribution of delays in user requests attributed to power savings under level 4 for slowdown targets equal to 10, 15, 50, and 100.

Level 3		Level 4	
FG Resp. Slowdown (D)	Time in Power Saving Mode	FG Resp. Slowdown (D)	Time in Power Saving Mode
10 (7)	8.94 (8.32)	10 (6)	4.62 (4.62)
15 (11)	11.11 (10.48)	15 (11)	6.15 (6.65)
50 (65)	27.55 (24.79)	50 (53)	12.36 (12.37)
100 (162)	48.17 (45.29)	100 (184)	24.17 (24.58)

Table 5: Estimated performance under trace File 2, under power savings Levels 3 and 4. The values presented in parentheses are the results obtained from the trace-driven simulations. All results are in (%).

FG	Real Power Saving (%)					
	Level 3			Level 4		
Slw.	Code 1	File 1	File 2	Code 1	File 1	File 2
10	1.22	0.20	1.61	0.64	0.05	1.43
15	1.80	0.33	2.00	1.03	0.05	1.91
50	3.95	1.02	4.96	4.02	0.70	3.83
100	4.94	1.63	8.67	6.25	1.50	7.49

Table 6: Real power saving for our traces, for Level 3 and Level 4 savings. All results are in (%).

high as 100, the percentage of affected requests is always less than 3%. The CDF of the delay distribution for all three traces further make the point of the robustness of the framework.

While the results in Tables 3, 4, and 5 indicate what portion of the time is utilized for power savings, the actual power savings are estimated using the data in Table 1. Our findings are presented in Table 6. Not surprisingly, even in lightly utilized enterprise systems, it is difficult to reach high actual power savings (see the limits in the last two columns of Table 2 where we assume full knowledge of all future workload), especially if the system can tolerate low performance degradation. Nevertheless, our methodology is robust and identifies any potential savings. Without any knowledge of the future workload it can opportunistically exploit idle intervals based on the performance degradation level it can tolerate.

5. CONCLUSIONS

In this paper, we presented a framework that accurately finds any opportunities that exist in a storage system for power savings. It estimates power savings capabilities for each power saving mode in disk drives and performance degradation level. Based on the estimations, the system decides which power saving mode to utilize (if any) for power savings. The framework also determines "when" and for "how long" the idle period should be utilized by an idle power saving mode. The framework is robust and lightweight because it bases its decisions on workload characteristics such as the histogram of idle times.

6. REFERENCES

[1] D. Colarelli and D. Grunwald. Massive arrays of idle disks for storage archives. In *Supercomputing '02: Proceedings of the 2002 ACM/IEEE conference on Supercomputing*, pages 1–11, Los Alamitos, CA, USA, 2002. IEEE Computer Society Press.

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

[3] 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'09 conference*, 2009.

[4] D. Narayanan, A. Donnelly, and A. I. T. Rowstron. Write off-loading: Practical power management for enterprise storage. In *FAST*, pages 253–267, 2008.

[5] E. Pinheiro and R. Bianchini. Energy conservation techniques for disk array-based servers. In *ICS*, pages 68–78, 2004.

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

[7] Q. Zhu, Z. Chen, L. Tan, Y. Zhou, K. Keeton, and J. Wilkes. Hibernator: helping disk arrays sleep through the winter. In *SOSP '05: Proceedings of the twentieth ACM symposium on Operating systems principles*, pages 177–190, New York, NY, USA, 2005. ACM.