Sensitivity Based Power Management of Enterprise Storage Systems

Sriram Sankar† Sudhanva Gurumurthi† Mircea R. Stan‡

† Department of Computer Science
University of Virginia
Charlottesville, VA 22904
{ss2wn, gurumurthi}@cs.virginia.edu

‡ Department of Electrical and Computer Engg.
University of Virginia
Charlottesville, VA 22904
mircea@virginia.edu

Abstract

Energy-efficiency is a key requirement in data centers today. Storage systems constitute a significant fraction of the energy consumed in a data center and therefore enterprise storage systems need to deliver high performance in an energy-efficient manner. Static tuning of the storage system is not sufficient since energy consumption is strongly dependent on runtime variations in workload characteristics. Although dynamic disk power management can enable the storage system to adapt to varying workload conditions, prior work in this area has resorted to ad hoc heuristics that cannot guarantee that the system meets energy-efficiency goals. In this paper, we present a novel approach to storage power management that uses the sensitivity-based optimization technique. Our approach systematically balances the dynamic knobs in the disks to operate the storage-system at a desired performance level while maximizing the energy savings. We show that sensitivity-based power management can reduce the energy consumed by the storage system by over 20% for a set of commercial server workloads. We compare sensitivity-based power management to a previously proposed power management scheme for multi-RPM disk drives and show that our approach yields better performance and energy savings.

1. Introduction

Storage systems play a key role in affecting the performance of a large number of data-intensive applications, including transaction processing, business analytics, Internet search-engines, and e-mail services. Storage systems for these applications are built using a large number of disks, usually configured as RAID arrays, to provide the requisite I/O performance. On the other hand, power and cooling costs have become a significant concern in data centers and disk drives account for 13%-20% of this cost [22, 20]. Data center managers therefore face the dilemma of striking a good balance between meeting the performance demands of applications hosted on their servers and minimizing the power and cooling costs of operating those servers.

The power consumption of a disk drive is affected by both static design parameters as well as runtime (dynamic) factors [6]. An example of a static design parameter is the number of the platters, which can be made smaller or larger based on the power budget. Laptops tend to use fewer platters due to their highly constrained power budgets while server drives may use more platters for capacity reasons. However, the greatest opportunity for managing power consumption is at runtime via the use of dynamic "knobs". Common approaches to dynamic disk power management include turning off the disk during long periods of idleness [3, 8], varying the rotational speed (RPM) of the drive [5, 1, 22], and reducing seek activity [10]. The recent introduction of multi-RPM drives into the market [19, 9] provides opportunities to control all these dynamic knobs to reduce the energy consumed by the storage system. Prior work has explored the use of each of these knobs in isolation and a number of heuristic policies have been proposed for disk power management in servers. However, it is desirable to have a *systematic* way of *optimally* controlling several dynamic knobs in the storage system to maximize the energy savings for a given set of performance constraints. In this paper, we propose one such methodology.

Our approach is based on the sensitivity-based optimization technique, which was originally proposed for energy-delay optimizations in circuits [12]. The key idea behind this optimization technique is the fact that optimally trading off between performance and power is based on the desired level of performance that we wish to achieve or the energy that we are willing to consume. Therefore, there are multiple optimal points in the energy-performance space, thereby forming a Pareto curve. Sensitivity-based analysis facilitates identifying these optimal points by calculating the ratio of energy to delay sensitivities with respect to each knob in the system and ensuring that the sensitivity ratios are equal. A recent paper by Zhang et al. [21] demonstrated the applicability of sensitivity-based optimization for determining the optimal settings of the design-time parameters of disk drives. However, this previous work did not explore how sensitivity-based optimization techniques can be applied for runtime disk power management using dynamic knobs, where the optimizations need to be dynami-

cally adapted to variations in the workload behavior. In this paper, we use sensitivity-based optimization for dynamic disk power management.

Towards this end, this paper makes the following contributions:

- We present a sensitivity-based optimization technique for controlling multiple dynamic knobs in disk drives. Our technique measures the sensitivity of the various knobs at regular intervals at runtime and controls the settings of the various knobs based on this measurement.
- We show that our technique can reduce the energy consumption of the storage system by 20.6% on the average for a set of commercial server workloads. We compare our technique to a previously proposed heuristic for power management in multi-RPM disk drives [5] and show that our approach provides greater energy savings and delivers higher performance than the previous technique.
- We perform a sensitivity analysis of our technique to three storage system and power management policy parameters: the RPM step-size, the size of the sensitivity measurement window, and the performance threshold settings. We find that our power management technique is effective in reducing the storage system energy consumption for a wide range of parameter values.

The outline for the rest of the paper is as follows. The next section discusses the related work and Section 3 describes the experimental setup. Section 4 presents an overview of disk drives and sensitivity based optimization and describes our power management technique. Our experimental results are presented in Section 5 and Section 6 concludes this paper.

2. Related Work

The traditional approach to disk power management in single-user laptop and desktop systems is to spin down the disks into a standby mode during long periods of idleness [3]. However, this approach is challenging to apply in enterprise storage systems due to the lack of sufficient idleness between I/O requests to spin down the disks into the standby mode [7]. Three basic sets of techniques have been proposed to overcome this problem: (i) creating sufficient idleness in a subset of the disks by replicating and/or migrating their data into other disks [15, 2, 13]; (ii) dynamic modulation of the disk drive RPM (DRPM) to reduce the spindle power during periods of idleness or lighter I/O loads [5, 1, 19, 9]; (iii) minimizing seek activity by replicating data within the disks [10].

There have been two prior works on providing performance guarantees while optimizing power in server storage systems using DRPM. Li et al. [11] proposed performance-directed energy management for main memory and disks. Their disk algorithm periodically adjusts the values of certain thresholds used in the DRPM control policy (proposed in [5]) to meet performance goals. Their study showed that self-tuning the parameters in the DRPM control policy is challenging due to the large number of parameters. Zhu et al. [22] propose an alternative approach to providing performance guarantees. Their Hibernator scheme takes as input a given maximum allowable response time as a Service Level Agreement (SLA) and uses a combination of data layout optimizations and RPM modulation to ensure that the storage system meets delivers this performance while minimizing energy consumption. In Hibernator, the RPM speed-setting decisions are made infrequently (once every few hours) which would make their scheme less adaptive to variations in the workload characteristics over smaller timescales. On the other hand, we make power management decisions at a finer granularity and demonstrate that we can get significant energy savings by adapting to variations in the workload behavior.

In general, all of these previous works tackle disk power management by using only a single dynamic knob - spindowns, RPM modulation, or disk-arm control. In this paper, we explore the use of multiple dynamic knobs and use sensitivity-based optimization to guide the setting of these knobs at runtime to maximize energy efficiency.

Finally, the energy consumed by the storage system can be reduced by replacing hard disk drives with solid-state disks (e.g., flash). Indeed, flash memory is already used in a variety of consumer electronic products and is also becoming popular in laptop computers. However, the cost per megabyte of solid state memory is significantly higher than those of hard disk drives and disk drives are expected to be the primary medium of storage in servers for at least another decade [16].

3. Experimental Setup and Workloads

Our experiments are carried out using Disksim [4], which is a widely used simulator for studying storage systems. We incorporate the disk power models equivalent to those given in [21] into the simulator, after validating them against real data from several manufacturer data sheets. We use a set of commercial I/O traces to measure the performance and power characteristics of our approach. Table 1 provides details about these workloads and the original storage systems on which they were collected. Financial is a trace of an On-Line Transaction Processing (OLTP) application collected at a large financial institution. Websearch is an I/O trace collected from a popular Internet search-engine server [18]. The TPC-H trace was collected on an 8-way IBM Netfinity SMP machine with 15 disks and running the IBM DB2 EE edition database management software. The workload was run in the power test mode in which the 22

queries of the benchmark are run consecutively. Openmail is a trace of an HP OpenMail e-mail server at the Atlanta Response Center [17].

Workload	Requests	Disks	Capacity (GB)	RPM	Platters
Financial	5,334,945	24	19.07	10000	4
Websearch	4,579,809	6	19.07	10000	4
TPC-H	4,228,725	15	35.96	7200	6
Openmail	3,053,745	8	9.29	10000	1

Table 1. Workloads and the configuration of the original storage systems on which the traces were collected.

We assume that the storage system of each workload uses disk drives that are identical to those listed in Table 1, except that they are multi-RPM drives, where the disk can perform I/O at each RPM level and can dynamically transition from one RPM to another based on a control policy [5]. We model the transition time characteristics of our drives using a linear-fit of the transition time characteristics of two real multi-RPM drives - the Sony Multi-Mode disk drive [14] and the Hitachi Deskstar 7K400 [9]. The power consumed due to the transition between any two RPM levels is assumed to be the average of the power consumption at those two levels. As with previous work on multi-RPM drives, we assume that RPM transitions are done in steps and our default model assumes a total of 10 RPM-levels between 6000 RPM and 15,000 RPM with an RPM step-size of 1000 RPM.

4. Disk Drives and Sensitivity based Optimization

4.1. Basics of Disk Drives

A hard disk drive consists of a stack of circular platters that store data, which are mounted on a central spindle and rotated by a Spindle Motor (SPM) at a certain Rotations Per Minute (RPM). The data on the platter surface is organized into sectors and tracks and is read from and written to using read/write heads. The heads are mounted on sliders, which are connected to a centrally controlled actuator/arm assembly, whose motion is effected by the Voice-Coil Motor (VCM). In addition to these two electro-mechanical components, modern disk drives also have a variety of onboard electronics such as data channels, motor drivers, and a cache.

The main operating modes of a disk drive are: (i) *seek:* movement of the disk head (and arms) to desired location on a platter, effected by the VCM; (ii) *rotational latency:* the time during which the desired sector rotates under the head; (iii) *transfer:* the data is read from or written to the platters; (iv) *idle:* the disk is not servicing a request, and is waiting to service future requests. The platters continue to spin during this phase and therefore consume power.

Since the bulk of the power in a disk drive is consumed by the electro-mechanical components, most storage power management schemes attempt to reduce the power consumed by these parts of the drive.

4.2. Sensitivity and Energy-Delay Optimization

The goal of the energy-delay optimization problem is to maximize energy savings subject to a given delay constraint. Let us assume that there are two knobs in the system, x and y, both of which affect energy and delay. Then, the energy-delay optimization problem can be formally stated as:

$$\textit{min Energy(x, y) s.t. Delay(x, y)} = D_0$$

where D_0 is the delay constraint (i.e., performance guarantee). Although the knobs can be static or dynamic, since our goal is on runtime power management, we will restrict our discussion to dynamic knobs in this paper.

The sensitivity based optimization technique [12, 23] can be used to solve this optimization problem. The key idea behind this technique is that energy-efficiency can be attained when the ratio of sensitivities (partial derivatives) of energy (E) to delay (D) with respect to each knob is balanced. For knobs x and y, we can state this mathematically as:

$$\frac{\frac{\partial E}{\partial x}}{\frac{\partial D}{\partial x}} = \frac{\frac{\partial E}{\partial y}}{\frac{\partial D}{\partial y}} \tag{1}$$

We now briefly explain how we can identify such optimal points. Let θ_i denote the "potential for energy reduction" using knob i at a given instant. θ_i is the ratio of the percentage change in energy to a percentage change in delay using knob i at that particular instant. For example, $\theta_i = 2$ means that a 2% change in energy consumption will produce a 1% change in performance if i is used as the knob at the given instant. This means that, among the various knobs in the system, turning down the knob that has the highest value of θ_i would provide the best opportunity for energy savings. Another key point to note is that θ_i depends upon the actual energy (E) and delay (D) of the system, which vary over time based on the characteristics of the workload using the system. Therefore, θ_i varies over time as well.

For the knobs x and y, we can express θ_x and θ_y as:

$$\theta_x = -\frac{D}{E}\left(\frac{\frac{\partial E}{\partial x}}{\frac{\partial D}{\partial x}}\right) = -\frac{\frac{\partial E}{\partial x}/E}{\frac{\partial D}{\partial x}/D} \tag{2}$$

$$\theta_y = -\frac{D}{E}\left(\frac{\frac{\partial E}{\partial y}}{\frac{\partial D}{\partial y}}\right) = -\frac{\frac{\partial E}{\partial y}/E}{\frac{\partial D}{\partial y}/D} \tag{3}$$

Let us define "Tradeoff Factor" (T_f) as the ratio of the potentials for energy reduction of the knobs:

$$T_f = \theta_x : \theta_y \quad (4)$$

If we wish to operate the system at an optimal point, where we satisfy a specific delay constraint with minimal energy, we would like to have:

$$T_f = \theta_x : \theta_y = 1 \quad (5)$$

Substituting equations 2 and 3 in equation 5 yields equation 1. Sensitivity-based optimization attempts to achieve this balance by turning down the knob that has the highest value of θ_i or by turning up the knob that has the smallest value of θ_i.

4.3 Sensitivity Based Optimization of Disk Drives

In order to apply sensitivity based optimization to disk drives, we first need to select the set of dynamic knobs that we wish to use. In this paper, we use two dynamic knobs: SPM speed (RPM) and the VCM speed. Controlling the speed of these two motors can be achieved by varying their voltages. We do not use disk spindowns because all the workloads that we use this study have insufficient idleness for us to be able to apply this technique directly. Idleness can be increased by using our power management scheme in conjunction with replication and migration techniques [15, 2, 13].

Given these two knobs, the energy-delay optimization problem for disk drives for a given performance constraint D_0 can be written as:

min Energy(SPM, VCM) s.t. Delay(SPM, VCM) = D_0

Let us denote the potential for energy reduction due to the SPM and the VCM speeds as θ_{SPM} and θ_{VCM}.

4.4 Crafting Power Management Policies

Since θ_i varies over time for a given workload, we need to periodically measure the energy and delay, and compute the ratio of sensitivities $(\partial E/\partial i)/(\partial D/\partial i)$ to the knob i. In our power management policies, we measure the energy and delay and compute the sensitivities after every n requests to the storage system. We call this n-request window as the "sample window". (We study the impact of varying the size of the sample window in Section 5.2). To compute the sensitivities, we vary the value of each knob by a small amount above and below its current setting and determine the corresponding change in E and D as a result of this variation. We then input these new RPM and VCM speed values into analytical disk power and performance models to estimate the change in energy and delay. We use analytical models that are equivalent to those developed by Zhang et al. [21]. Once we have obtained the values of θ_{SPM} and θ_{VCM} and given a particular delay constraint, we can implement power management policies for the storage system.

We have implemented a Sensitivity-Based Power Management scheme (which we call ***SBPM***) that works as follows. We profile a workload running on the storage system for k I/O requests without performing any power management and calculate the average response time of the I/O requests over this window. During this phase, we set the RPM of the disk drives to those used in their original storage system configurations given in Table 1. In our experiments, we choose k to be the first 100,000 I/O requests of each workload. We use this average response time value as the basis for the performance constraint (D_0) to use in the optimization. (Note that a data center manager may craft this performance constraint in a different way for her SLA. For example, the value for D_0 might be arrived at through negotiations with the client whose application is to be hosted on her servers, or she may choose a different performance metric, such as, the maximum or minimum response time of the I/O requests over the profiling window). In addition to D_0, the ***SBPM*** scheme also uses two additional thresholds that specify the range of acceptable deviation in performance of the storage system from D_0: an upper threshold (UT) and a lower threshold (LT), which are expressed as a percentage.

We then measure the average response time of the storage system (RT) every n requests and calculate θ_{SPM}, θ_{VCM}, and T_f. Based on the values of RT, D_0, UT, and LT, there are three possibilities:

- $100(\frac{RT-D_0}{D_0}) > UT$: This condition indicates that the storage system is operating below the acceptable level of performance and therefore we need to turn up the knob settings to improve performance.
- $100(\frac{RT-D_0}{D_0}) < LT$: This condition indicates that the storage system is operating at higher performance than the desired level and therefore we can save energy by turning down the knob settings.
- If the difference in the response times is between LT and UT, then no power management actions are taken.

For the first two cases, the magnitude and direction of the SPM and VCM knobs are modulated based on the ratio of the potentials for energy reduction such that we bring T_f to one.

We compare the effectiveness of ***SBPM*** to the power management heuristic proposed by Gurumurthi et al. for DRPM drives [5]. That heuristic focuses solely on the RPM knob for power management and we denote this heuristic as ***DRPM***. ***DRPM*** is an ad hoc policy implemented at two-levels in the system: a performance-centric component that is implemented at the storage array controller and an energy-centric component at the disks. The array controller measures the average response time of the storage system

across n-request sample windows and compares the difference in the response times between successive windows. As with *SBPM*, *DRPM* also uses upper and lower performance thresholds. In *DRPM*, the difference in response times between two consecutive windows is compared against predefined values of UT and LT. If the difference is greater than UT, then all the disks are ramped up to the full-speed RPM. If the difference is less than LT, then disks are allowed to lower their RPMs to a level that is proportional to the difference between the response-time change and LT. This information is conveyed to the disks via a watermark value. The disks periodically check whether they have any I/O requests pending in their input queues and scale down their RPM by one step if the queue is empty to reduce energy consumption, eventually saturating their RPM at the watermark level.

In our experiments, we use the same values of UT and LT for both *SBPM* and *DRPM*.

5. Results

We present two sets of experimental results. The first set of results show the energy and performance characteristics of *SBPM* for the four workloads and compare them to the original storage systems of each of the workloads (which we denote as *Baseline*) and those that use the *DRPM* power management scheme. In the second set of results, we perform a sensitivity analysis of *SBPM* to various storage system and power management policy parameters. The default parameters used in the experiments are given in Table 2.

Parameters	Values in Experiment
RPM Step-Size	1000
Upper Threshold	15%
Lower Threshold	5%
Sample Window size	10000

Table 2. Default parameters used in the experiments.

5.1. Energy and Performance Characteristics of *SBPM*

The energy consumption characteristics of *SBPM*, *DRPM*, and *Baseline* are given in Figure 1 and the corresponding performance results are given in Figure 2. Figure 1 shows the energy consumption of the two power management schemes normalized to the *Baseline* system. We present the performance results in Figure 2 as Cumulative Distribution Functions (CDFs) of the response time. CDFs show the fraction of I/O requests whose response times are less than or equal to a given value on the x-axis. CDFs allow us to visualize the scenario where a large number of I/O requests may be experiencing relatively short response times whereas a few other requests may have very long response times.

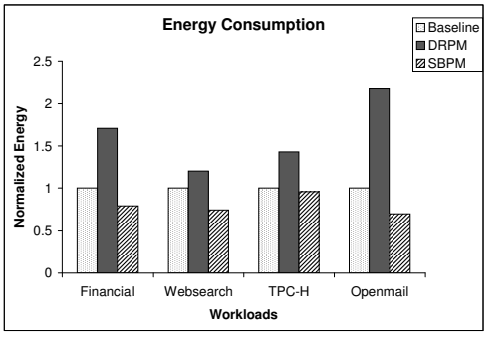

Figure 1. Energy consumption of *DRPM* and *SBPM* normalized to the *Baseline* energy consumption.

When we look at Figure 1, we can see that *SBPM* reduces the energy consumption of the storage system from the baseline. The energy savings for Financial, Websearch, TPC-H, and Openmail are 21.31%, 26.14%, 4.34%, and 30.75% respectively. *SBPM* also delivers performance comparable to *Baseline* for Financial, Websearch, and TPC-H as shown in Figure 2. Since we choose to allow up to a 15% degradation in the average response time to save energy (via the UT parameter), the *SBPM* CDFs are slightly shifted below the *Baseline* CDFs.

However, we find that *DRPM* consumes *more energy* than *Baseline* for all four workloads and its performance is worse than *SBPM*. The energy result is surprising given that this heuristic has been shown in prior work to be effective for managing power [5, 11]. The main reason for this difference is due to our assumptions about the time taken to transition between RPM levels. The prior work assumed the transition times between RPMs to be in the millisecond range. However real multi-RPM drives [9, 14], which we use as the basis for our transition time model, have latencies in the order of seconds to shift between RPM levels. This order of magnitude difference in the transition latencies has a profound impact on the performance and energy costs of shifting RPM levels. We will provide a deeper analysis of why *SBPM* fares better than *DRPM* shortly.

One factor that has a direct impact on performance is the inter-arrival time of the I/O requests. We find that the average inter-arrival times of Financial, Websearch, TPC-H, and Openmail to be 8.19 ms, 2.96 ms, 8.76 ms, and 1.18 ms respectively. Indeed, as we can see in Figure 2, both *SBPM* and *DRPM* for those workloads with very short inter-arrival times (Websearch and Openmail) experience a larger reduction in performance than those with longer inter-arrival times. However, beyond this high-level trend, although both power management schemes use the same values for UT and LT, *SBPM* consistently outperforms *DRPM*, thereby

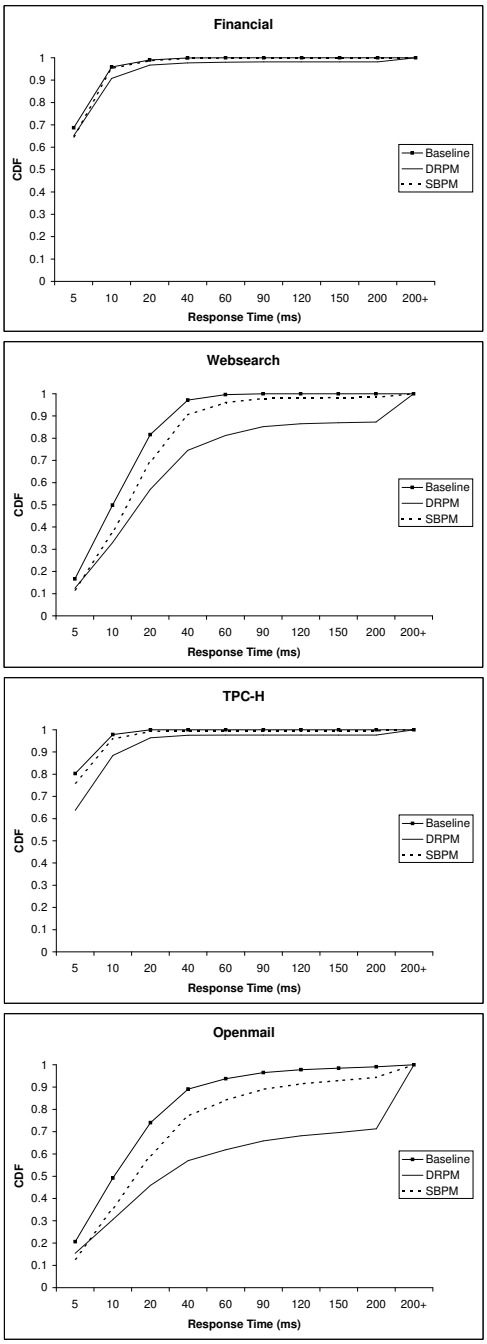

Figure 2. Performance of *SBPM*, *DRPM*, and *Baseline*.

suggesting that the latter might be making more sub-optimal RPM transition decisions and hence incurring larger performance penalties and energy overheads.

In order gain a deeper understanding of the relative differences between *SBPM* and *DRPM*, we use the sensitivity-based optimization as an analysis tool. As we discussed in Section 4.2, in order to operate the storage system in an energy-efficient manner, we need $T_f = (\theta_{SPM}/\theta_{VCM}) = 1$ at all times. Therefore, an analysis of the variation in the T_f value of the storage system can provide insights into the energy-efficiency of the system. A system whose T_f value stays closer to one will be more energy-efficient than one whose T_f value varies over a larger range. The variation in the T_f values for *SBPM* and *DRPM* for each workload are given in Figure 3, and the variation in the two potentials of energy reduction - θ_{SPM} and θ_{VCM} - are shown in Figure 4.

In Figure 3, the first row of graphs correspond to the T_f values for *DRPM* and the second row corresponds to *SBPM*. We can observe that the T_f values for *DRPM* oscillate significantly without ever reaching a steady-state value. This is due to the design of the *DRPM* heuristic, where the policy is to ramp up the RPM of all the drives to full speed if the performance degradation exceeds UT. As a result of such transitions to the highest RPM, the response time of the storage system improves significantly over the given sample window and, in many cases, the response time measurement in the next window exceeds the lower threshold LT. When the array controller observes this, it lowers the watermarks by a large value and the disks pull down their RPMs in order to reduce energy consumption. This results in excessive performance degradation and disks are forced to ramp back up again to full speed in the following sample window. As a result of these RPM oscillations, coupled with the performance and energy costs of transitioning between RPMs, the use of the *DRPM* policy leads to excessive energy consumption and poor performance, especially for those workloads that have very short inter-arrival times between I/O requests.

SBPM, on the other hand, is able to better balance the system by using both the SPM and VCM as knobs and we can see that the T_f values of all the workloads using this policy stay closer to one. Even for the Financial workload, which starts out at a sub-optimal state with a high T_f value, *SBPM* quickly balances the system and subsequent T_f values stay close to one. This process of balancing is shown in Figure 4, where, although θ_{VCM} starts out at a lower point, both the θ_{SPM} and θ_{VCM} values are equalized after about 50 sample windows. This equalization of the potentials of energy of the two knobs is also clearly visible for the Websearch workload.

For TPC-H, we can see that there are occasional spikes in the T_f values of *SBPM*. Although the frequency and magnitude of these spikes are lower than *DRPM*, such spikes lead to short durations of sub-optimal behavior. As Figure 4 shows, the θ_{VCM} values for TPC-H oscillate over a larger range in order to keep the system in balance. The reason for this behavior is due to the characteristics of disk seeks in this workload. We find that there is substantial variation in seek activity across the sample windows. For example, during sample windows 118-127, we find that seek time

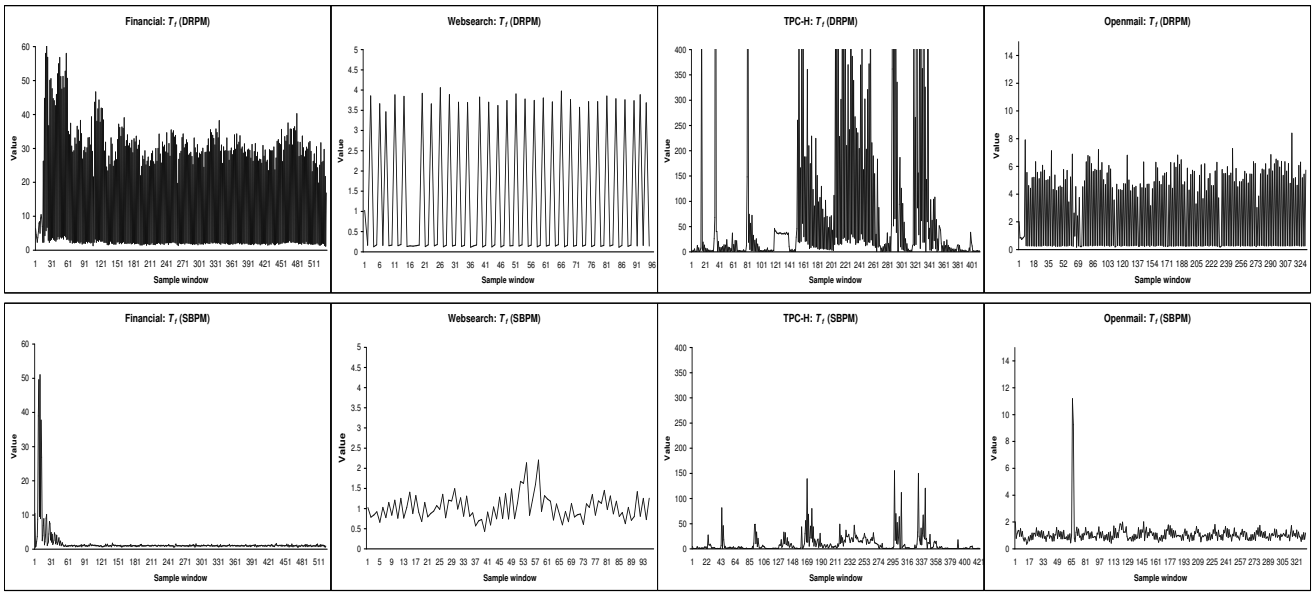

Figure 3. Comparison of T_f, the ratio of sensitivities, for *DRPM* and *SBPM*. For *SBPM*, T_f is close to the optimal value of 1 most of the time.

constituted 4% of the average response time whereas seeks accounted for 20% of the response time during the next ten sample windows. In order to quantify this behavior, we calculated the coefficient of variation in the seek time as a proportion of the overall response time over all the sample windows of each workload. The coefficient of variation is a normalized measure of dispersion in a probability distribution and is mathematically expressed as a percentage, $100(\sigma/\mu)$, where μ and σ are the mean and standard-deviations of the distribution respectively. We find that TPC-H has the highest coefficient of variation (62.01%), while those of Financial, Websearch, and Openmail are 18.57%, 10.88%, and 20.84% respectively. Since the VCM speed has a direct impact on the seek time and the impact of seeks on the response time varies significantly for this workload, the T_f of this workload is occasionally thrown off balance and the VCM knob is made to compensate for this and bring T_f closer to one. As a result of these frequent compensating actions, a large amount of energy is expended and therefore the energy savings using *SBPM* for TPC-H is lower than those for the other three workloads as shown in Figure 1. A similar region of deviation in seek behavior is seen for the Openmail workload at the 65^{th} sample window in Figure 3.

However, in general, *SBPM* is able to operate the storage system closer to the $T_f = 1$ point and therefore provides more energy savings and delivers higher performance than *DRPM*.

5.2. Sensitivity Analysis

Having seen the benefits of *SBPM*, we now analyze its behavior under a variety of system and policy parameters. We perform three sensitivity analysis experiments. The first experiment studies the impact of the RPM step-size on the effectiveness of *SBPM*. The second and third experiments explore the impact of two policy related parameters - the sample window size and the values of the performance thresholds UT and LT respectively.

5.2.1 Impact of RPM Step-Size

The RPM step-size can impact the effectiveness of power management. A smaller step-size gives finer granularity of control over energy and performance, but may be more challenging from the engineering viewpoint, whereas larger step-sizes are easier to implement but provide less control. To evaluate the impact of this parameter on *SBPM*, we consider a finer-grained step-size of 500 RPM and a coarser-grained step-size of 2500 RPM. The results from this experiment are given in Figure 5. Each set of bars in the Figure correspond the energy consumption of each step-size normalized to the default 1000 RPM step-size.

As we can see from Figure 5, using finer or coarser step-sizes does not significantly change the energy behavior. We also find the relative performance of all the configurations to be very similar and therefore do not show the response time CDF graphs. This result concurs with a previous study on DRPM drives that used an oracle (i.e., perfect) disk idleness predictor to evaluate the impact of RPM step-size [5].

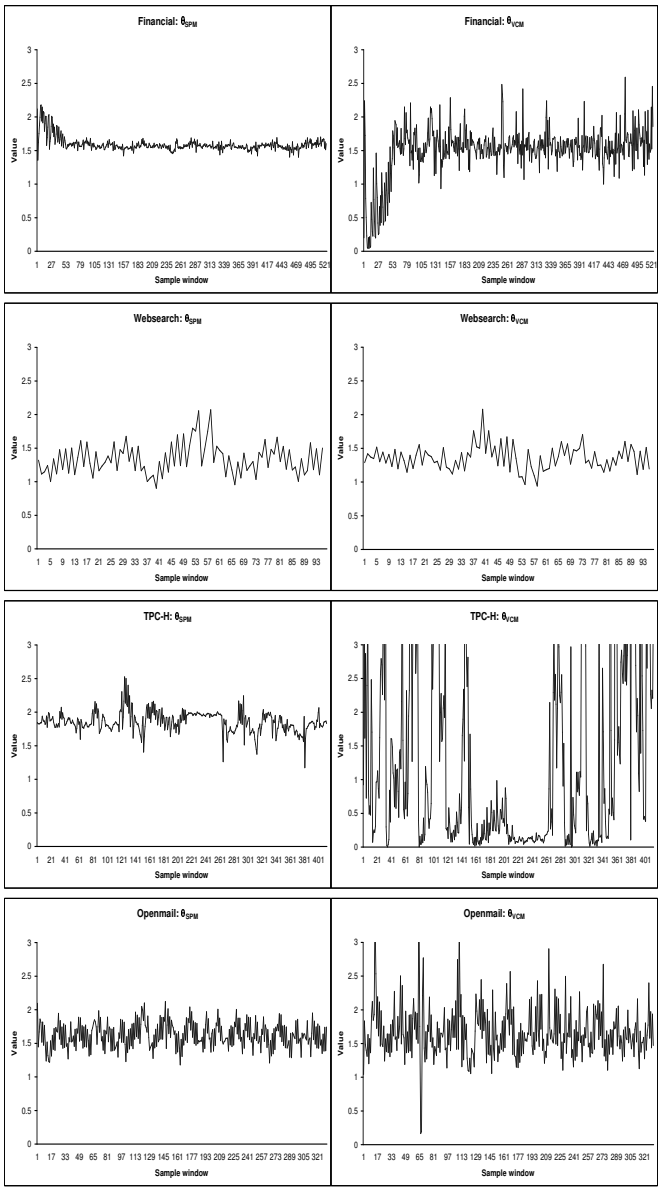

Figure 4. Variation in θ_{SPM} and θ_{VCM} for *SBPM*.

Although *SBPM* does not use any oracle, we still achieve energy savings across different RPM step-sizes.

5.2.2 Impact of Sample Window Size

The sample window size is a policy related parameter whose granularity can affect performance and energy. Too small a sample window can lead to frequent changes to the knob settings, which can have adverse consequences on performance and energy consumption. On the other hand, if the sample window is too large, then the system will be less responsive to workload variations and we might have to resort to large knob modulations to re-balance the system which can again entail performance and energy penalties. In order to study the effect of this parameter on *SBPM*, we consider two smaller sample windows of 1000 and 5000 I/O requests and two larger windows of 25000 and 50000 requests. The performance results corresponding to this experiment are given in Figure 6 and the energy results in Figure 7. Each set of bars in Figure 7 correspond the energy consumption of each sample window normalized to the default 10000-request window.

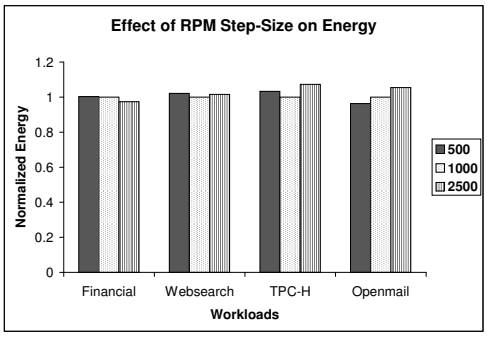

Figure 5. Impact of RPM Step-Size on *SBPM* - Energy consumption. The values are normalized to the 1000 RPM step-size configuration.

In our system, one of the most important factors that influences the choice of the sample window size is the transition time between RPM levels. In our disk drive model, the time taken to move from one RPM level to another is in the order of seconds, while the inter-arrival time between I/O requests for all our workloads are in the order of milliseconds. Since transitioning between RPM levels incurs performance and energy costs, it is important to amortize this cost over a large number of I/O requests.

Among the four workloads, Openmail has the shortest inter-arrival time between I/O requests (1.18 ms) and therefore its performance is most sensitive to RPM transitions. We find that the use of shorter sample windows leads to severe performance degradation and the system is overloaded within a short period of time. As a result of this behavior, we omit the data points for Openmail in the graphs. (Although a real system would handle such an overload condition at a higher level, for example, by dropping connections to the server, we do not attempt to modulate the arrival rate of the I/O requests to the storage system in this study. However, in a real system, we expect our power management scheme to be used in conjunction with admission-control schemes at the higher level to handle the overall system load). Websearch also has very short inter-arrival times (2.96 ms) and therefore using very small sample windows leads to significant performance degradation. Since the other workloads have longer inter-arrival times, their

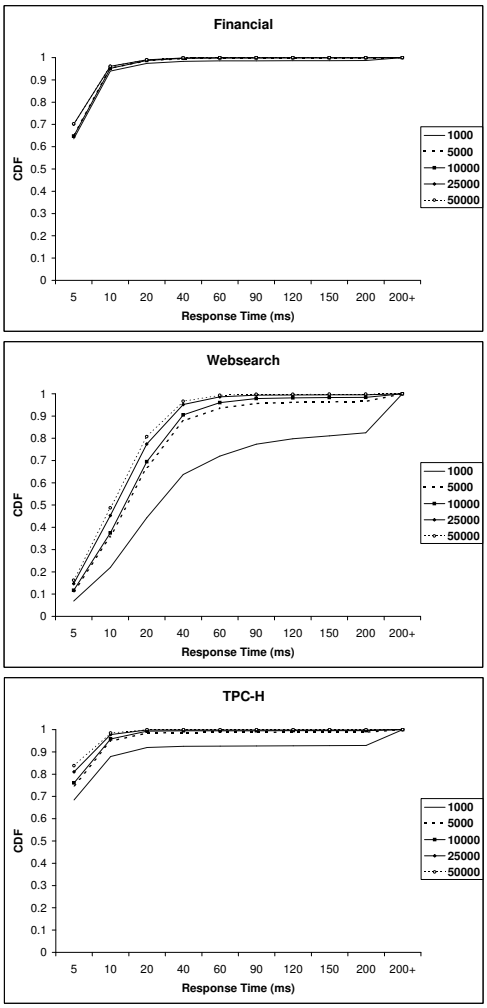

Figure 6. Impact of Sample Window Size on *SBPM* - Performance.

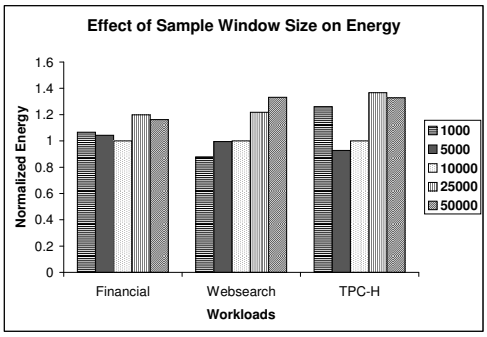

Figure 7. Impact of Sample Window Size on *SBPM* - Energy Consumption. The energy values are normalized to the 10000-request sample window configuration.

performance is less sensitive to the sample window size.

When we look at Figure 7, we can see that the smaller sample window sizes (1000, 5000, and 10000) are comparable in their energy consumption trends. The energy variations between the various configurations for each workload are within 10%. An interesting case is the TPC-H workload, whose energy consumption is significantly higher for the 1000-request sample window. The reason for this is due to a combination of two factors: (i) the transition time between RPM levels, and (ii) significant variations in the seek time. As we mentioned in Section 5.1, TPC-H exhibits significant variation in its seek characteristics compared to the other workloads and hence its θ_{VCM} varies significantly. For the 1000-request sample window, we find that the coefficient of variation of the seek time as a proportion of the overall response time for TPC-H is 55.6%, compared to 15.92% and 19.88% for Financial and Websearch respectively. Therefore, in each sample window, in addition to any latencies incurred due to RPM transitions, there is also energy expenditure due to VCM speed modulations, both of which combine to diminish the effectiveness of *SBPM* in reducing the energy consumption. The use of the larger 25000 and 50000-request sample windows, on the other hand, reduces the effectiveness of *SBPM* in adapting to the workload conditions and results in higher energy consumption as shown in Figure 7.

To summarize, when deploying a system that uses *SBPM*, it is important to choose sample windows that are large enough to amortize the cost of transition times while still being responsive to changes in the workload conditions.

5.2.3 Impact of Performance Thresholds

The choice of the performance thresholds UT and LT can have a significant impact on the energy-efficiency of the storage system. We experimented with two additional settings for these parameters: (UT=8%, LT=5%) and (UT=15%, LT=10%). The former setting biases the system towards higher performance whereas the latter is geared towards saving more energy. In both these cases, we found that the relative differences between *SBPM* and *DRPM* remain invariant and *SBPM* provides higher energy savings than *DRPM* for the same parameter settings. In the interest of space, we omit these graphs in the paper.

6. Conclusions and Future Work

Power is a major problem in data centers and storage systems account for a sizable portion of the overall energy consumption. However, many applications that run on servers housed in data centers also demand high performance. It is therefore important to develop techniques that can meet the performance demands of these applications while maximizing the energy savings. In this paper, we have presented

one such technique. Our approach makes use of sensitivity-based optimization to dynamically modulate various knobs in the storage system, such as the disk RPM and the VCM speed to adapt the storage system to varying workload conditions and save energy. We have shown how to craft power management policies based on sensitivity-based optimization and have demonstrated that our *SBPM* approach can provide significant energy savings for several server workloads. We have also shown that our approach is superior to the previously proposed *DRPM* heuristic for dynamic RPM modulation, especially for realistic RPM transition latencies. Our sensitivity analysis demonstrates the effectiveness of *SBPM* in reducing the storage energy consumption for a variety of RPM steps-sizes, sample window sizes, and performance threshold settings.

In future work, we plan to explore how sensitivity-based power management can be used in conjunction with idleness creation techniques [15, 2, 13] to save even more energy. We also plan to explore the use of sensitivity-based power management for other knobs within the storage system such as the disk and array controller cache sizes.

Acknowledgments: This research has been supported in part by NSF CAREER Award CCF-0643925, NSF grant CNS-0551630, a Marco IFC grant and gifts from HP and Google.

References

[1] E. Carrera, E. Pinheiro, and R. Bianchini. Conserving Disk Energy in Network Servers. In *Proceedings of the International Conference on Supercomputing (ICS)*, June 2003.

[2] D. Colarelli and D. Grunwald. Massive Arrays of Idle Disks for Storage Archives. In *Proceedings of Supercomputing*, November 2002.

[3] F. Douglis and P. Krishnan. Adaptive Disk Spin-Down Policies for Mobile Computers. *Computing Systems*, 8(4):381–413, 1995.

[4] G. Ganger, B. Worthington, and Y. Patt. *The DiskSim Simulation Environment Version 2.0 Reference Manual*. http://www.ece.cmu.edu/ ganger/disksim/.

[5] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, and H. Franke. DRPM: Dynamic Speed Control for Power Management in Server Class Disks. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*, pages 169–179, June 2003.

[6] S. Gurumurthi, A. Sivasubramaniam, and V. Natarajan. Disk Drive Roadmap from the Thermal Perspective: A Case for Dynamic Thermal Management. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*, pages 38–49, June 2005.

[7] S. Gurumurthi, J. Zhang, A. Sivasubramaniam, M. Kandemir, H. Franke, N. Vijaykrishnan, and M. Irwin. Interplay of Energy and Performance for Disk Arrays Running Transaction Processing Workloads. In *Proceedings of the International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pages 123–132, March 2003.

[8] D. Helmbold, D. Long, T. Sconyers, and B. Sherrod. Adaptive Disk Spin-Down for Mobile Computers. *ACM/Baltzer Mobile Networks and Applications (MONET) Journal*, 5(4):285–297, December 2000.

[9] Hitachi Power and Acoustic Management - Quietly Cool, March 2004. http://www.hitachigst.com/tech/techlib.nsf/productfamilies/White_Papers.

[10] H. Huang, W. Hung, and K. Shin. FS2: Dynamic Data Replication in Free Disk Space for Improving Disk Performance and Energy Consumption. In *Proceedings of the Symposium on Operating Systems Principles (SOSP)*, pages 263–276, October 2005.

[11] X. Li, Z. Li, F. David, P. Zhou, Y. Zhou, and S. Adve. Performance Directed Energy Management for Main Memory and Disks. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pages 271–283, October 2004.

[12] D. Marković, V. Stojanović, B. Nikolić, M. Horowitz, and R. Brodersen. Methods for True Energy-Performance Optimization. *IEEE Journal of Solid-State Circuits*, 39(8):1282–1293, August 2004.

[13] D. Narayanan, A. Donnelly, and A. Rowstron. Write Off-Loading: Practical Power Management for Enterprise Storage. In *Proceedings of the USENIX Conference on File and Storage Technologies (FAST08)*, February 2008.

[14] K. Okada, N. Kojima, and K. Yamashita. A novel drive architecture of HDD: "multimode hard disc drive". In *Proceedings of the International Conference on Consumer Electronics (ICCE)*, pages 92–93, June 2000.

[15] E. Pinheiro and R. Bianchini. Energy Conservation Techniques for Disk Array-Based Servers. In *Proceedings of the International Conference on Supercomputing (ICS)*, June 2004.

[16] J. Rydning, D. Reinsel, and W. Schlichting. Storage Technology Futures to 2015: Flash, Disk Drive, Holographic, and New Technology Road Maps and Applications Revealed. *IDC Special Study No:202056*, June 2006.

[17] The Openmail Trace. http://tesla.hpl.hp.com/private software/.

[18] UMass Trace Repository. http://traces.cs.umass.edu.

[19] WD GreenPower Hard Drives. http://www.wdc.com/en/company/greenpower.asp.

[20] S. Worth. Storage Networking Industry Association - Green Storage Tutorial, 2007. http://www.snia.org/forums/green/programs/SWWorth_Green_Storage.pdf.

[21] Y. Zhang, S. Gurumurthi, and M. Stan. SODA: Sensitivity Based Optimization of Disk Architecture. In *Proceedings of the Design Automation Conference (DAC)*, pages 865–870, June 2007.

[22] 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 Symposium on Operating Systems Principles (SOSP)*, pages 177–190, October 2005.

[23] V. Zyuban and P. Strenski. Balancing Hardware Intensity in Microprocessor Pipelines. In *IBM Journal of Research and Development*, volume 47, 5-6, pages 585–598, 2003.