DRPM: Dynamic Speed Control for Power Management in Server Class Disks *

Sudhanva Gurumurthi[†]   Anand Sivasubramaniam[†]   Mahmut Kandemir[†]   Hubertus Franke[‡]

[†] Dept. of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802
{gurumurt,anand,kandemir}@cse.psu.edu

[‡] IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598
frankeh@us.ibm.com

Abstract

A large portion of the power budget in server environments goes into the I/O subsystem - the disk array in particular. Traditional approaches to disk power management involve completely stopping the disk rotation, which can take a considerable amount of time, making them less useful in cases where idle times between disk requests may not be long enough to outweigh the overheads. This paper presents a new approach called DRPM to modulate disk speed (RPM) dynamically, and gives a practical implementation to exploit this mechanism. Extensive simulations with different workload and hardware parameters show that DRPM can provide significant energy savings without compromising much on performance. This paper also discusses practical issues when implementing DRPM on server disks.

Keywords: Server Disks, Power Management.

1   Introduction

Data-centric services - file and media servers, web and e-commerce applications, and transaction processing systems to name a few - have become commonplace in the computing environments of large and small business enterprises, as well as research and academic institutions. In addition, other data-centric services such as search engines and data repositories on the Internet are sustaining the needs of thousands of users each day. The commercial consequences of the performance and/or disruption of such services have made performance, reliability and availability the main targets for optimization traditionally. However, power consumption is increasingly becoming a major concern in these systems [4, 2]. Optimizing for power has been understood to be important for extending battery life in embedded/mobile systems. It is only recently that the importance of power optimization in server environments has gained interest because of the cost of power delivery, cost of cooling the system components, and the impact of high operating temperatures on the stability and reliability of the components.

Several recent studies have pointed out that data centers can consume several Mega-watts of power [5]. It has been observed [5] that power densities of data centers could grow to over 100 Watts per square foot and that the capacity of new data centers for 2005 could require nearly 40 TWh (around $4B) per year. A considerable portion of this power budget on these servers is expected to be taken up by the disk subsystem, wherein a large number of disks are employed to handle the load and storage capacity requirements. Typically, some kind of I/O parallelism (RAID [30]) is employed to sustain the high bandwidth/throughput needs of such applications which are inherently data-centric. While one could keep adding disks for this purpose, at some point the consequent costs of increasing power consumption may overshadow the benefits in performance. Disks even when idle (spinning but not performing an operation), can drain a significant amount of power. For instance, a server class IBM Ultrastar 36ZX [17] disk is rated at 22.3 W (compare this to an Intel Xeon processor clocked at 1.6 GHz which is rated at 57.8 W). When we go to specific server configurations (e.g. a 4-way Intel Xeon SMP clocked at 1.6 GHz with 140 disks drawn from [36]), the disks consume 13.5 times more power than the processors.

One possible solution is to use a large cache, under the assumption that the I/O workload will exhibit good locality. Caching can also potentially be used to delay writes as proposed by Colarelli et al [6] for archival and backup systems. However, in most servers, though large caches are common, they are typically used for *prefetching* to hide disk latencies, since not all server workloads exhibit high temporal locality to effectively use the cache. Prefetching does not reduce the power consumption of the disks.

Another way of alleviating this problem is by shutting down disks or at least stop them from spinning since the spindle motor for the disks consume most of the power as will be described in section 3. Many disks offer different power modes and one could choose to transition them to a low power mode when not in use (idle) which achieves this functionality (e.g. stop the spinning). Such techniques have been effectively used [7, 25, 14, 26, 9, 38, 29, 13] in laptop environments where the goal is mainly to save battery energy. When in a low power mode, the disk needs to be spun up to full speed before a request can be serviced, and this latency is much more critical in servers than in a laptop setting. Further, the application of such traditional mode-control techniques for server environments is challenging, where one may not have enough idleness and where performance is more critical [11].

From the above discussion, we see two extremes in op-

---

eration - one that is performance efficient with disks spinning all the time, and the other where the goal is power optimization by stopping the spinning of the disk whenever there is a chance at the cost of performance. In this paper, we present a new option - Dynamic Rotations Per Minute (DRPM) - where one could choose to dynamically operate between these extremes, and adaptively move to whichever criterion is more important at any time. The basic idea is to dynamically modulate the speed at which the disk spins (RPM), thereby controlling the power expended in the spindle motor driving the platters. Slowing down the speed of spinning the platters can potentially provide quadratic (with respect to the change in RPM) power savings. However, a lower RPM can hurt rotational latencies and transfer costs when servicing a request (at best linearly). In addition to these rotational latencies and transfer costs, disk accesses incur seek overheads to position the head to the appropriate track, and this is not impacted by the RPM. Consequently, it is possible to benefit more from power than one may lose in performance from such RPM modulations. This DRPM mechanism provides the following benefits over the traditional power mode control techniques [25, 26] (referred to as TPM in this paper):

- Since TPM may need a lot more time to spin down the disk, remain in the low power mode and then spin the disk back up, there may not be a sufficient duration of idleness to cover all this time without delaying subsequent disk requests. On the other hand, DRPM does not need to fully spin down the disk, and can move down to a lower RPM and then back up again, if required, in a shorter time (RPM change costs are more or less linear with the amplitude of the change). The system can service requests more readily when they arrive.

- The disk does not necessarily have to be spun back up to its full speed before servicing a request as is done in TPM. One could choose to spin it up if needed to a higher speed than what it is at currently (taking lower time than getting it from 0 RPM to full speed), or service the request at the current speed itself. While opting to service the request at a speed less than the full speed may stretch the request service time, the exit latency from a lower power mode would be much lower than in TPM.

- DRPM provides the flexibility of dynamically choosing the operating point in power-performance trade-offs. It allows the server to use state-of-the-art disks (fastest in the market) and provides the ability to modulate their power when needed without having to live with a static choice of slower disks. It also provides a larger continuum of operating points for servicing requests than the two extremes of full speed or 0 RPM. This allows the disk subsystem to adapt itself to the load imposed on it to save energy and still provide the performance that is expected of it.

The DRPM approach is somewhat analogous to voltage/frequency scaling [31] in integrated circuits which provides more operating points for power-performance trade-offs than an on/off operation capability. A lower voltage (usually accompanied with a slower clock for letting circuits stabilize) provides quadratic power savings and the slower clock stretches response time linearly, thus providing energy savings during the overall execution. This is the first paper to propose and investigate a similar idea for disk power management.

The primary contribution of this paper is the DRPM mechanism itself, where we identify already available technology that allows disks to support multiple RPMs. More importantly, we develop a performance and power model for such disks based on this technology showing how costs for dynamic RPM changes can be modeled.

The rest of this paper looks at evaluating this mechanism across different workload behaviors. We first look at how well an optimal algorithm, called $DRPM_{perf}$ (that provides the maximum energy savings without any degradation in performance) performs under different workloads, and compare its pros and cons with an optimal version of TPM, called TPM_{perf} (which provides the maximum power savings for TPM without any degradation in performance). When the load is extremely high (i.e. there are very few idle periods), there is not much that can be done in terms of power savings if one does not want to compromise at all on performance, regardless of what technique one may want to use. At the other end of the spectrum, when there are very large idle periods, we find TPM_{pref} providing good power savings as is to be expected since it completely stops spinning the disks as opposed to $DRPM_{perf}$, which keeps them spinning albeit at a slow speed. However, there is a wide range of intermediate operating conditions when DRPM turns out to give much better (upto 60%) savings in the idle mode energy consumption, even if one does not wish to compromise at all on performance. It is also possible to integrate the DRPM and TPM approaches, wherein one could use TPM when idle times are very long and DRPM otherwise.

Finally, this paper presents a simple heuristic that dynamically modulates disk speed using the DRPM mechanism and evaluates how well it performs with respect to $DRPM_{perf}$ where one has perfect knowledge of the future. One could modulate this algorithm by setting tolerance levels for degradation in response times, to amplify the power savings. We find that this solution comes fairly close to the power savings of $DRPM_{perf}$ (which does not incur any response time degradation) without significant penalties in response time, and can sometimes even do better in terms of power savings.

The rest of this paper is organized as follows. The next section gives an overview of the sources of energy consumption in a disk and prior techniques for power optimization. Section 3 presents the DRPM mechanism and the cost models for its implementation. Section 4 gives the experimental setup and section 5 gives results with $DRPM_{perf}$, comparing its potential with TPM and conducts a sensitivity analysis. The details of our heuristic for online speed setting and its evaluation are given in section 6. Section 7 discusses some issues that arise when implementing a real DRPM disk. Finally, section 8 summarizes the contributions of this paper.

2   Disk Power and TPM

There are several components at the disk that contribute to its overall power consumption. These include the spindle motor that is responsible for spinning the platters, the actuator that is responsible for the head movements (seeks), the electrical components that are involved in the transfer operations, the disk cache, and other electronic circuitry. Of these, the first two are mechanical components and typically overshadow the others, and of these the spindle motor is the most dominant. Studies of power measurements on different disks have shown that the spindle motor accounts for nearly 50% of the overall idle power for a two-platter disk, and this can be as high as 81.34% for a ten-platter server class disk [12]. Consequently, traditional power management techniques at the higher level focus on addressing this issue by shutting down this motor when not in active use.

Disk power management has been extensively studied in the context of single disk systems, particularly for the laptop/desktop environment. Many current disks offer different power modes of operation, such as *active* - when the disk is servicing a request, *idle* - when it is spinning and not serving a request, and one or more low power modes that consume less energy than idle (where the disk platters do not spin). Managing the energy consumption of the disk consists of two steps, namely, detecting suitable idle periods and then spinning down the disk to a low power mode whenever it is predicted that the action would save energy. Detection of idle periods usually involves tracking some kind of history to make predictions on how long the next idle period would last. If this period is long enough (to outweigh spindown/spinup costs), the disk is explicitly spun down to the low power mode. When an I/O request comes to a disk in the spundown state, the disk first needs to be spun up to service this request (incurring additional exit latencies and power costs in the process). Many idle time predictors use a time-threshold to find out the duration of the next idle period. A fixed threshold is used in [25], wherein if the idle period lasts over 2 seconds, the disk is spun down, and spun back up only when the next request arrives. The threshold could itself be varied adaptively over the execution of the program [7, 14]. A detailed study of idle-time predictors and their effectiveness in disk power management has been conducted in [9]. Lu et al. [26] provide an experimental comparison of several disk power management schemes proposed in literature on a single disk platform.

We broadly refer to these previous power mode-control mechanisms as TPM in this paper. It is to be noted that TPM has the disk spinning at either its full speed or fully stationary and does not allow intermediate RPMs.

Another power saving approach, though orthogonal to this work, is to replace a single disk with multiple smaller form-factor disks that consume lower power as in [40].

3 Dynamic RPM (DRPM)

The TPM techniques (and our DRPM mechanism) can be used in conjunction with other techniques that can reduce disk accesses (by aggregation and/or caching using hardware/OS/application support) or place data to reduce head movements [15] (which can save actuator power) to further the power savings. However, as was mentioned earlier, the spindle motor power needed to spin the disks is still the major power consumer [12], which is expended even when the disk is not serving a request (and is spinning). As can be seen in Figure 1, which shows the RPMs and power consumption of different IBM server class disks over the years, there appears to be a strong correlation between the rotational speed and the idle power in these disks (though it should be noted that RPM is not the only variation across the technologies employed). This motivates us to investigate the possibility of modulating the RPM dynamically to adjust the power consumption.

3.1 Basics of Disk Spindle Motors

A detailed exposition of disk spindle motors (SPMs) can be found in [19, 34]. Disk SPMs are permanent magnet DC brushless motors. In order to operate as a brushless motor, sensors are required inside the motor to provide the pulses necessary for commutation (i.e., rotation). These sensors may either be Hall-Effect sensors or back-EMF sensors. Speed control of the motors can be achieved by using Pulse-Width Modulation (PWM) techniques, which make use of the data from the sensors.
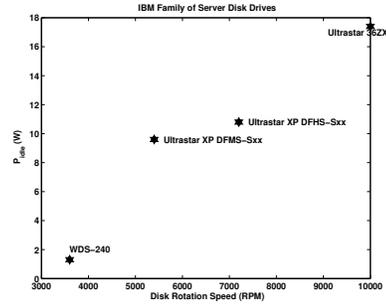

Figure 1. IBM Server Disks - Idle Power Consumption. For each disk, the form-factor was fixed at 3.5" and the largest capacity configuration was chosen. The idle power is relatively independent of the form-factor [12, 34]. We found that the idle power was not that strongly related to the capacity. For instance, two other IBM disks, the Ultrastar 9ZX and the Ultrastar 18ZX, are both 10,000 RPM disks with 9.1 and 18.2 GB capacity respectively, while their idle power consumption is 16.5 W and 16.3 W respectively.

A large accelerating torque is first needed to make the disks start spinning. This high torque is essentially required to overcome the stiction forces caused by the heads sticking to the surface of the disk platter. The use of technologies like Load/Unload [18] can ameliorate this problem by lifting the disk-arm from the surface of the platter. These technologies also provide power benefits and are used for example in IBM hard disk drives [18] to implement the special IDLE-3 mode. In addition to providing the starting torque, the SPM also needs to sustain its RPM once it reaches the intended speed.

One traditional approach in improving disk performance over the years has been to increase the RPM (which reduces rotational latencies and transfer times), which can prove beneficial in bandwidth-bound applications. However, such increases can cause concern in additional issues such as noise and Non-Repeatable Run-Outs (NRROs). (NRROs are off-track errors that can occur at higher RPMs, especially at high track densities.) These design considerations in the development of high RPM disks have been addressed by the use of advanced motor-bearing technologies like fluid [1, 16] and air-bearings [37].

However, the power associated with high RPMs still remains and this paper focuses on this specific aspect.

3.2 Analytical Formulations for Motor Dynamics

Our DRPM solution dynamically controls the spindle motor to change the RPM of the spinning platters. The RPM-selection capability can be provided by allowing the spindle-motor control block [35] of the hard-disk controller [20] to be programmable. For example, the desired RPM can be input via a programmable register in the hard-disk controller. The value read from this register can inturn be used by the spindle-motor driver [3] to generate the requisite signals for operating the disk at that RPM.

We now present the time overhead needed to effect an RPM change, and the power of the resulting state as a function of the RPM.

3.2.1 Calculating RPM Transition Times

In order to calculate the time required for a speed-change, we need some physical data of the spindle-motor. This information for a specific disk is usually proprietary, but there are DC brushless motors commercially available that we can use for this purpose. We have obtained the necessary information from the datasheet of a Maxon EC-20 20 mm flat brushless permanent magnet DC motor [27], whose physical characteristics closely match those of a hard disk spindle motor. Table 1 summarizes the basic mechanical characteristics of this motor.

Parameter	Value	Units
Max. Permissible Speed	15000	rpm
Rotor Inertia (J_0)	3.84	gcm^2
Torque Constant (K_T)	9.1	mNm/A
Max. Continuous Current at 12K rpm (I)	0.708	A

Table 1. Maxon EC-20 Motor Characteristics

The motor specifications give a formula for calculating the time Δt (in ms) required for a speed-change of Δn RPM with a load inertia J_L as:

$$\Delta t = \frac{\pi}{300} \Delta n \frac{J_0 + J_L}{K_T I}$$

The load on the spindle motor is the platter assembly. We dismantled a 3.5" Quantum hard disk, and measured the weight of an individual platter using a sensitive balance and also its radius. Its weight m was found to be 14.65 gm and radius r was 4.7498 cm. Using these values, and assuming 10 platters per disk (as in [17], though we also have sensitivity results for different number of platters), we calculated the moment of inertia of the load J_L (in gcm^2) as:

$$J_L = \frac{1}{2} n_p m r^2 = \frac{1}{2} \times 10 \times 14.65 \times (4.7498)^2$$

where n_p is the number of platters.

$$\implies J_L = 1652.563 gcm^2$$

Therefore, we have

$$\Delta t = 2.693 \times 10^{-4} \Delta n \qquad (1)$$

This shows that the time cost of changing the RPM of the disk is directly proportional (linear) to the amplitude of the RPM change.

3.2.2 Calculating the Power Consumption at an RPM Level

We briefly explain the dependence between the power consumption of a motor and its rotation speed. A detailed exposition of this topic can be found in [23]. The motor voltage V is related to the angular velocity (rotation-speed) ω as

$$V = K_E \omega$$

where K_E is a constant. The power consumed by the motor, P, is
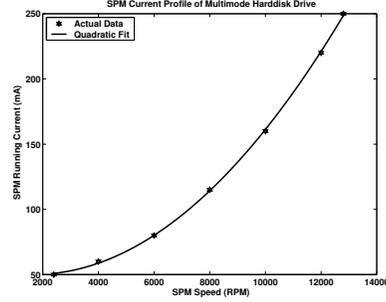
$$P = VI = \frac{V^2}{R}$$

Figure 2. Current Drawn by Sony Multimode Hard Disk

where R is the resistance of the motor. Therefore, we have

$$P = \frac{K_E^2 \omega^2}{R} \qquad (2)$$

This equation, similar to that relating the power and voltage for CMOS circuits, indicates that a change in the rotation-speed of the disk has a *quadratic* effect on its power consumption. In order to investigate whether this relationship holds true in the context of a hard disk, we used an experimental curve-fitting approach. There exists a commercial hard disk today - *the Multimode Hard Disk Drive* [28] from Sony - that indeed supports a variable speed spindle motor. The speed setting on such a disk is accomplished in a more *static (pre-configured) fashion*, rather than modulating this during the course of execution. The published current consumption values of this disk for different RPM values provides insight on how the current drawn varies with the RPM.

Figure 2 shows the current drawn by the SPM of the Multimode hard disk (repeated from [28]). A simple curve fit of these data points clearly shows this quadratic relationship. This relationship may appear somewhat different from the trends shown in Figure 1 where one could argue that the relationship between RPM and power is linear. However, Figure 1 shows the trend for disks of different generations, where it is not only the RPM that changes but the hardware itself. On the other hand, equation 2 and the Multimode hard disk current consumption profile (Figure 2) shows the relation between these two parameters is more quadratic for an individual disk drive.

This Multimode disk is composed of only two platters, while we are looking at server class disks that have several more platters (8-10 platters). Consequently, we cannot directly apply this model to our environment. On the other hand, a study from IBM [34] projects the relation between idle power and RPM for 3.5" server class IBM disks. In this study, other design factors such as the change in the number of disk-platters, have been considered besides just the RPM to make the projections for the individual disks and we depict the results from there by the points shown in Figure 3. In our power modeling strategy for a variable RPM disk, we employed two approaches, to capture a quadratic and linear relationship respectively:

- We took the points from the IBM study and used a quadratic curve to approximate their behavior as is shown by the solid curve in Figure 3 to model the idle power as

$$\begin{aligned} P_{idle} &= 1.318 \times 10^{-7} rpm^2 - 4.439 \times 10^{-4} rpm \\ &+ 8.643 \end{aligned} \qquad (3)$$
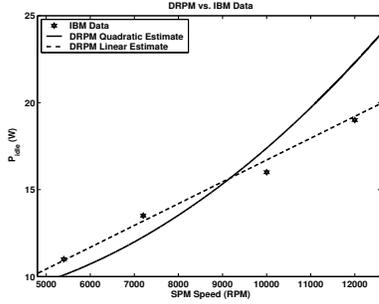
Figure 3. Comparison of DRPM Model to the IBM Projections

- We also performed a linear least squares fit through the points as is shown by the dotted line in Figure 3 to model the idle power as

$$P_{idle} = 0.0013rpm + 4.158 \qquad (4)$$

We have used both models in our experiments to study the potential of DRPM. In general, we find that the results are not very different in the ranges of RPMs that were studied (between 3600 to 12000) as is evident even from Figure 3 which shows the differences between linear and quadratic models within this range are not very significant.

Equations 1 and 3/4 provide the necessary information in modeling the RPM transition costs and power characteristics of our DRPM strategy. For the power costs of transitioning from one RPM to another, we conservatively assume that the power during this time is the same as that of the higher RPM state.

4 Experimental Setup and Workload Description

We conducted our evaluations using the DiskSim [8] simulator modeling a disk array for a server environment. DiskSim provides a large number of timing and configuration parameters for specifying disks and the controllers/buses for the I/O interface. The simulator was augmented with power models to record the energy consumption of the disks when performing operations like data-transfers, seeks, or when just idling. Our DRPM implementation accounts for the queuing and service delays caused by the changes in the RPM of the disks in the array. The default configuration parameters used in the simulations are given in Table 2, many of which have been taken from the data sheet of the IBM Ultrastar 36ZX [17] server hard disk. The power consumption of the standby mode was calculated by setting the spindle motor power consumption to 0 when calculating P_{idle} based on the method described in section 3. Note that this value for the power consumption is very aggressive as the actual power consumption even in this mode is typically much higher (for example its value is 12.72 W in the actual Ultrastar disk). However, as we shall later show, even with such a deep low-power standby mode that is used by TPM, DRPM surpasses TPM in terms of energy-benefits in several cases. Also, in our power models, we have accounted for the case that the power penalties for the active and seek modes (in addition to idle power) also depend upon the RPM of the disk. The modes exploited by

TPM, including the power consumption of each mode and the transition costs are illustrated in Figure 4.

We have considered several RPM operating levels, i.e. different resolutions for stepping up/down the speed of the spindle motor. These "step-sizes" are as low as 600 RPM, providing 13 steps between the extremes of 3600 and 12000 RPM (15 RPM levels in all). The default configuration that we use in our experiments is a 12-disk RAID-5 array, with a quadratic DRPM power-model and a step-size of 600 RPM.

Parameter	Value
Parameters Common to TPM and DRPM	
Number of Disks in the Array	12,24
Stripe Size	16 KB
RAID Level	5,10
Individual Disk Capacity	33.6 GB
Disk Cache Size	4 MB
Max. Disk Rotation Speed	12000 RPM
Idle Power @ 12000 RPM	22.3 W
Active (R/W) Power @ 12000 RPM	39 W
Seek Power @ 12000 RPM	39 W
Standby Power	4.15 W
Spinup Power	34.8 W
Spinup Time	26 secs.
Spindown Time	15 secs.
Disk-Arm Scheduling	Elevator
Bus Type	Ultra-3 SCSI
DRPM-Specific Parameters	
Power Model Type	Quadratic,Linear
Minimum Disk Rotation Speed	3600 RPM
RPM Step-Size	600,2100 RPM

Table 2. Simulation Parameters with the default configurations underlined. Disk spinups and spindowns occur from 0 to 12000 RPM and vice-versa respectively.
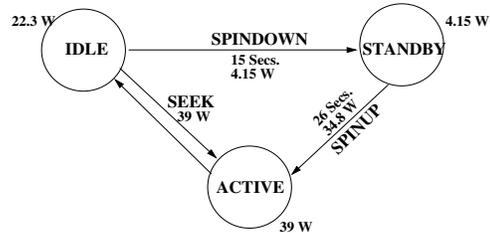

Figure 4. TPM Power Modes

Since we want to demonstrate the potential of the DRPM mechanism across a spectrum of operating conditions (different loads, long idle periods, bursts of I/O requests, etc.) that server disks may experience, and to evaluate the pros and cons of DRPM over other power saving approaches, we chose to conduct this study with several synthetic workloads where we could modulate such behavior. The synthetic workload generator injects a million I/O requests with different inter-arrival times, and request parameters (starting sector, request-size, and the type of access - read/write). All the workloads consist of 60% read requests and 20% of all requests are sequential in nature. These characteristics were chosen based on [33]. Since a closed-system simulation may alter the injected load based on service times of

the disk array for previous requests, we conducted an open-system simulation with these workloads.

We considered two types of distributions for the inter-arrival times, namely, exponential and Pareto. As is well-understood, exponential arrivals model a purely random Poisson process, and to a large extent models a regular traffic arrival behavior (without burstiness). On the other hand, the Pareto distribution introduces burstiness in arrivals, which can be controlled. The Pareto distribution is characterized by two parameters, namely, α, called the shape-parameter, and β, called the lower cutoff value (the smallest value a Pareto random-variable can take).[1] We chose a Pareto distribution with a finite mean and infinite variance.

For both distributions, we varied the mean inter-arrival time (in ms) as a parameter. In Pareto, there are different ways by which the traffic can be generated for a given mean. We set the β to 1 ms and varied α (i.e. when the mean is increased, the time between the bursts - idleness - tend to increase).

We use the term *workload* to define the combination of the distribution that is being used and the mean inter-arrival time for this distribution. For instance, the workload <Par,10> denotes a Pareto traffic with a mean inter-arrival time of 10 ms.

We compare the schemes for each workload using three metrics, namely, *total energy consumption over all the requests* (E_{tot}), *idle-mode energy consumption over all the requests* (E_{idle}), and *response-time per I/O request (T)*. These can be defined as follows:

- The total energy consumption (E_{tot}) is the energy consumed by all the disks in the array from the beginning to the end of the simulation period. We monitor all the disk activity (states) and their duration in each state, and use this to calculate the overall energy consumption by the disks (integral of the power in each state over the duration in that state).
- The idle-mode energy consumption (E_{idle}) is the energy consumed by all the disks in the array while not servicing an I/O request (i.e., while not performing seeks or data-transfers). This value is directly impacted by the spinning-speed of the spindle motor.
- The response-time (T) is the time between the request submission and the request completion averaged over all the requests. This directly has a bearing on the delivered system throughput.

Finally, we use the terms power and energy interchangeably sometimes.

5 Power Optimization without Performance Degradation

5.1 Energy Breakdown of the Workloads

Before we examine detailed results, it is important to understand where power is being drained over the course of execution i.e. when the disk is transferring data (Active), or positioning the head (Positioning) or when it is idling (Idle). Figure 5 gives the breakdown of energy consumption of two workloads from each of the inter-arrival time distributions - one at high and another at low load conditions - into these three components when there is no power saving technique
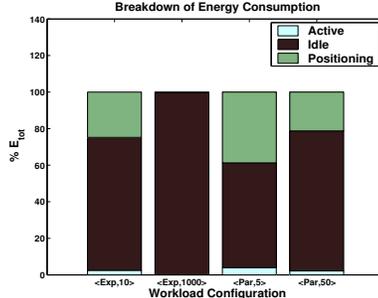
[1]The Pareto probability distribution function is given by $P(x) = \frac{\alpha\beta^{\alpha}}{x^{\alpha+1}}$, $x > \beta$, $\alpha > 0$ The mean is given by $E(x) = \frac{\alpha\beta}{\alpha-1}$.


Figure 5. Breakdown of E_{tot} for the different workloads. On the x-axis, each pair represents a workload defined by <Probability Distribution,Mean Inter-Arrival Time> pair.

employed. The high and low loads also indicate that idle periods are low and high respectively.

As is to be expected, when the load is light (<Exp,1000>, <Par,50>), the idle energy is the most dominant component. However, we find that even when we move to high load conditions (<Exp,10>, <Par,5>), the idle energy is still the most significant of the three. While the positioning energy does become important at these high loads, the results suggest that most of the benefits to gain are from optimizing the idle power (in particular, the spindle motor component which consumes 81.34% of this power). Consequently, our focus in the rest of this section is on the idle power component, by looking at how different schemes (TPM and DRPM) exploit the idleness for energy savings.

5.2 The Potential Benefits of DRPM ($DRPM_{perf}$)

The power saving, either with TPM or DRPM, is based on the idleness of disks between serving requests. While in the latter case, it is possible to get more savings by serving requests at a lower RPM, this may result in performance degradation. In the first set of results, we do not consider this to be an option, i.e. we define a scheme called $DRPM_{perf}$ whose performance is not any different from the original disk subsystem (which does not employ any power management technique). Further, to investigate what could be the potential of DRPM, we assume the existence of an idle-time prediction oracle, which can exactly predict when the next request will arrive after serving each request. Consequently, $DRPM_{perf}$ uses this prediction to find out how low an RPM it can go down to, and then come back up to full speed before servicing the next request (noting the times and energy required for doing such transitions).

To be fair, the same oracle can be used by TPM as well for effecting power mode transitions, and we call such a scheme TPM_{perf}, where the disk is transitioned to the standby mode if the time to the next request is long enough to accommodate the spindown followed by a spinup.

Note that $DRPM_{perf}$ can exploit much smaller idle times for power savings compared to TPM_{perf}. On the other hand, when the idle time is really long, TPM_{perf} can save more energy by stopping the spinning completely (while DRPM can take it down to only 3600 RPM). Therefore, in order to investigate the potential benefits if both these techniques were used in conjunction, we have also considered a scheme called *Combined*. In the *Combined* scheme, we use the oracle to determine which of the two
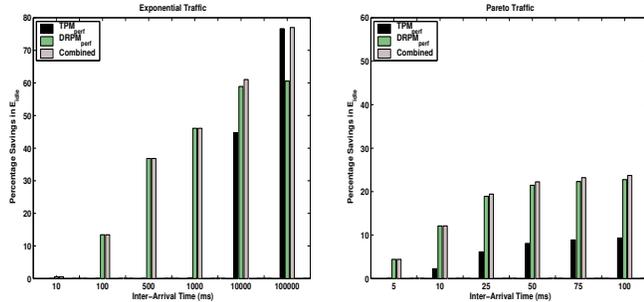
Figure 6. Savings in Idle Energy using TPM_{perf}, $DRPM_{perf}$, **and** $Combined$ **are presented for the quadratic power model.**

techniques saves the maximum energy, for each idle time-period.

We would like to point out that $DRPM_{perf}$, TPM_{perf}, and $Combined$ do not put a bound on the energy savings that one can ever get. Rather, they give a bound when performance cannot be compromised. Figure 6 presents the idle energy savings (which was shown to be the major contributor of overall energy) for these schemes as a function of the inter-arrival times in the two distributions.

When we first examine the exponential traffic results, we note that the results confirm our earlier discussion wherein large inter-arrival times favor TPM_{perf}. At the other end of the spectrum, when inter-arrival times get very small, there is not really much scope for any of these schemes to save energy if performance compromise is not an option. However, between these two extremes, we find that $DRPM_{perf}$ provides much higher savings than TPM_{perf}. It finds more idle time opportunities to transition to a lower RPM mode, which may not be long enough for TPM. As is to be expected, the combined scheme approaches the better of the two across the spectrum of workloads.

When we next look at the Pareto traffic results, we find that the arrivals are fast enough (due to burstiness of this distribution) even at the higher mean values considered that $DRPM_{perf}$ consistently outperforms TPM_{perf} in the range under consideration. It is also this reason that makes the energy savings of all the schemes with this traffic distribution lower than that for exponential where the idle times are less varying.

The purpose of this exercise was to examine the potential of DRPM with respect to TPM while not compromising on performance. The rest of this section looks to understanding the sensitivity of the power savings with this approach to different hardware and workload parameters. Since the sensitivity of DRPM is more prominent at the intermediate load conditions (where it was shown to give better savings than TPM), we focus more on those regions in the rest of this paper.

5.3 Sensitivity Analysis of $DRPM_{perf}$

5.3.1 Number of Platters

Disks show significant variability in platter counts. At one end, the laptop disks have 1 or 2 platters, while server class disks can have as many as 8-10 platters. The number of platters has a consequence on the weight imposed on the

spindle motor, which has to spin them as was described earlier. In Figure 7 the effect of three different platter counts (4, 10 and 16) has been shown for the two types of traffic with different load conditions. It can be seen that as the number of platters increases, the savings drop. This is because a larger weight is imposed on the spindle motor, requiring a higher torque for RPM changes thereby incurring more overheads. Nevertheless, even at the 16-platter count, which is significantly higher than those in use today, we still find appreciable power savings even at high load conditions.
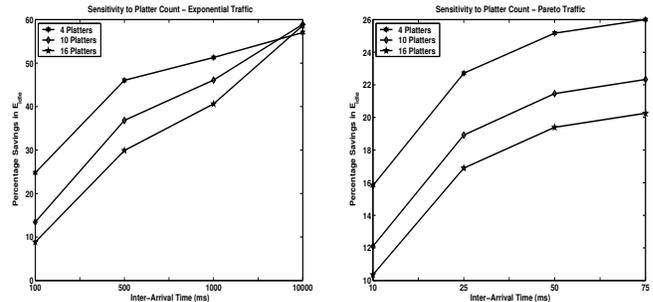

Figure 7. Sensitivity to Number of Platters in the Disk Assembly

While one may think that the need for storage-capacity increase over time may necessitate more platters, it is to be noted that this increase is usually achieved by denser media rather than by adding more platters. For instance, the IBM DFHS-Sxx (1994), 36ZX (1999), and 146Z10 (2002) have storage capacities of 4.51 GB, 36.7 GB and 146 GB respectively, but their platter counts are 8, 10, and 6. Therefore we do not expect platter counts to increase significantly.

5.3.2 Quadratic vs. Linear Power Model

As was discussed in section 3, we considered both quadratic and linear scaling models for the idle power consumption of the spindle motor at different RPMs. While the earlier results were presented with the quadratic model, we compare those results with the savings for $DRPM_{perf}$ with the linear model in Figure 8. We can observe that the differences between these two models are not very significant, though the linear model slightly under-performs that of the quadratic as is to be expected. This again confirms our earlier observations that the differences between a linear and quadratic model are not very different across these ranges of RPM values. Consequently, we find that $DRPM_{perf}$, even with a conservative linear power scaling model gives better energy savings than TPM_{perf} (compare with Figure 6).

We have also conducted similar sensitivity analysis for other factors such as the step-size employed for the spindle motor and the type of RAID configuration. The interested reader is referred to [10] for the details. In general, we find that $DRPM_{perf}$ can provide significant energy savings across a wide spectrum of disk and array configurations.

6 A Heuristic DRPM Algorithm

Having evaluated the potential of DRPM without any performance degradation, which requires an idle time pre-
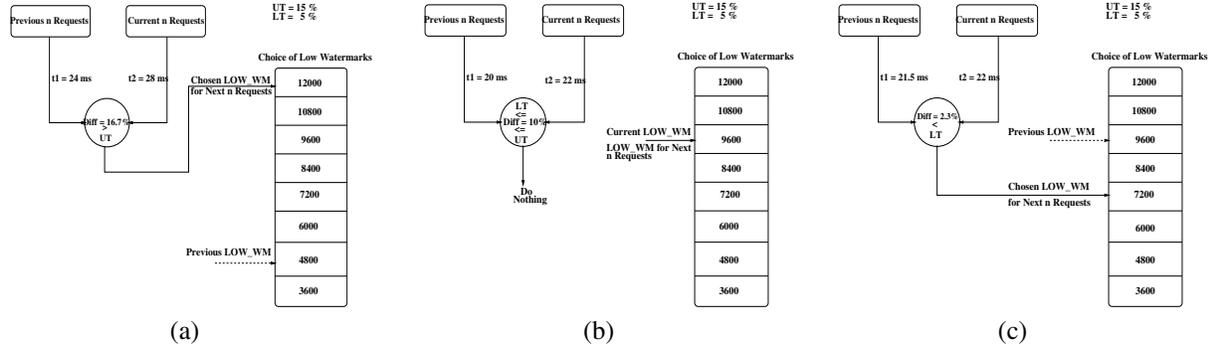
(a)   (b)   (c)

Figure 9. The operation of the DRPM heuristic for $UT = 15\%$ **and** $LT = 5\%$. **In each figure, for the choice of low watermarks, the dotted line shows where LOW_WM is before the heuristic is applied and the solid line shows the result of applying the scheme. The percentage difference in the response times,** t_1 **and** t_2 **between successive** n**-request windows,** $diff$, **is calculated. (a) If** $diff > UT$, **then LOW_WM is set to the maximum RPM for the next** n **requests. (b) If** $diff$ **lies between the two tolerance-limits, the current value of LOW_WM is retained. (c) If** $diff < LT$, **then the value of LOW_WM is set to a value less than the maximum RPM. Since** $diff$ **is higher than 50% of** LT **but lesser than 75% of** LT **in this example, it is set two levels lower than the previous LOW_WM. If it was between 75% and 87.5%, it would have been set three levels lower, and so on.**
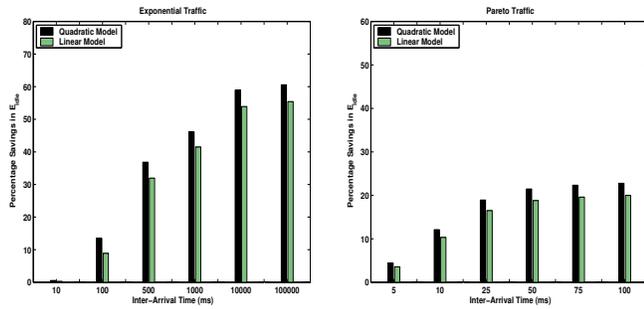

Figure 8. Behavior of $DRPM_{perf}$ **for a Power Model that relates the RPM and** P_{idle} **linearly**

diction oracle, we next move on to describe a scheme that can be used in practice to benefit from this mechanism. The goal is to save energy using the multiple RPM levels, without significantly degrading performance (response time).

In this scheme, (i) the array controller communicates a set of operating RPM values to the individual disks based on how performance characteristics (response time) of the workload evolve. More specifically, the controller specifies *watermarks* for disk RPM extremes between which the disks should operate; (ii) subsequently, each disk uses local information to decide on RPM transitions.

Periodically each disk inspects its request queue to check the number of requests (N_{req}) waiting for it. If this number is less than or equal to a specific value N_{min}, this can indicate a lower load and the disk ramps down its speed by one step. It can so happen, that over a length of time the disks may gradually move down to a very low RPM, even with a high load, and do not move back up. Consequently, it is important to periodically limit how low an RPM the disks should be allowed to go to. This decision is made by the array controller at the higher level which can track re-

sponse times to find points when performance degradation becomes more significant to ramp up the disks (or to limit how low they can operate at those instants).

The array controller tracks average response times for n-request windows. At the end of each window, it calculates the percentage change in the response time over the past two windows. If this percentage change (ΔT_{resp}) is

- larger than an upper tolerance (UT) level, then the controller immediately issues a command to all the disks that are operating at lower RPMs to ramp up to the full speed. This is done by setting the LOW_WM (Low Watermark) at each disk to the full RPM, which says that the disks are not supposed to operate below this value.

- between an upper (UT) and lower (LT) tolerance level, the controller keeps the LOW_WM at where it is, since the response time is within the tolerance levels.

- less than the lower tolerance level (LT), in which case the LOW_WM can be lowered even further. The specific RPM that is used for the LOW_WM is calculated proportionally based on how much the response time change is lower than LT.

These three scenarios are depicted in Figure 9 which shows the choice of the LOW_WM for example differences in response time changes with $UT = 15\%$, $LT = 5\%$, and eight possible values for the LOW_WM. These are also the values used in the results to be presented, and window sizes are $n = 250, 500, 1000$, though we have experimented with a more comprehensive design space. In our experiments, we set $N_{min} = 0$, whereby the disks initiate a rampdown of their RPM based on whether their request-queue is empty or not.

6.1   Results with DRPM

We have conducted extensive experiments to evaluate how well the above heuristic (denoted as simply DRPM in

the rest of this paper) fares, not only in terms of its absolute energy savings and response time degradation, but also comparing it to the $DRPM_{perf}$ and static RPM choices (where non-DRPM disks of lower RPMs are used). The complete set of experimental results is given in [10] and we present the highlights here.

The first set of results in Figure 10 show the energy savings and response time degradation of our DRPM heuristic with respect to not performing any power optimization (referred to as Baseline). The energy savings are given with both the quadratic and linear power models discussed earlier for two different inter-arrival times in each of the two distributions. Note that these are E_{tot} savings, and not just those for the idle energy.
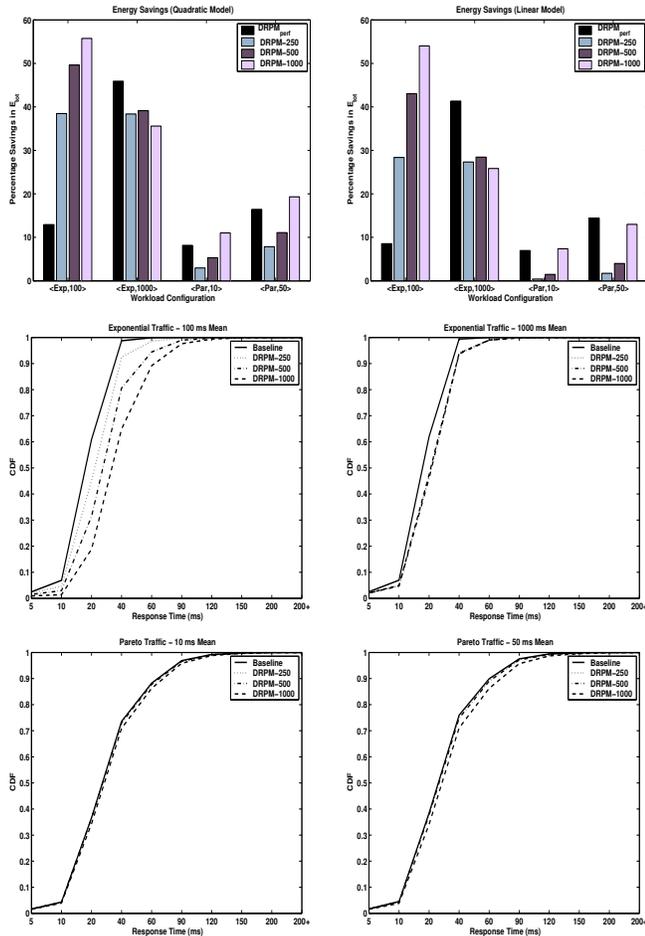

Figure 10. DRPM Heuristic Scheme Results.
$UT = 15\%$, $LT = 5\%$, $N_{min} = 0$. **The results are presented for** $n = 250, 500, 1000$, **referred to as DRPM-250, DRPM-500, and DRPM-1000 respectively.**

We observe that we can get as good savings, if not better in some cases (especially with higher loads) than $DRPM_{perf}$ which has already been shown to give good energy savings. Remember that $DRPM_{perf}$ services requests at the highest RPM even if it transitions to lower RPMs during idle periods. This results in higher active energy compared to the above heuristic which allows lower RPMs for serving requests, and also can incur higher transition costs in always getting back to the highest RPM. These effects are more significant at higher loads (smaller idle periods), causing our heuristic to in fact give better energy savings than $DRPM_{perf}$. At lighter loads, the long idle periods amortize such costs, and the knowledge of how long they are helps $DRPM_{perf}$ transition directly to the appropriate RPM instead of lingering at higher RPMs for longer times as is done in the heuristic scheme. Still the energy savings for the heuristic are quite good and are not far away from $DRPM_{perf}$, which has perfect knowledge of idle times. The results for the heuristic have been shown with different choices for n, the window of requests for which the LOW_WM is recalculated. A large window performs modulations at a coarser granularity, thus allowing the disks to linger at lower RPMs longer even when there may be some performance degradation. This can result in greater energy savings for larger n values as is observed in many cases.

The response time characteristics of the heuristic are shown as CDF plots in Figure 10, rather than as an average to more accurately capture the behavior through the execution. It can happen that a few requests get inordinately delayed while most of the requests incur very little delays. A CDF plot, which shows the fraction of requests that have response times lower than a given value on the x-axis, can capture such behavior while a simple average across requests cannot. These plots show the Baseline behavior which is the original execution without any power savings being employed, and is also the behavior of $DRPM_{perf}$ which does not alter the timing behavior of requests. The closeness of the CDF plots of the heuristic to the Baseline curve is an indication of how good a job it does of limiting degradation in response time.

At higher loads, it is more important to modulate the RPM levels (LOW_WM) at a finer granularity to ensure that the disks do not keep going down in RPMs arbitrarily. We see that a finer resolution ($n = 250$ requests) does tend to keep the response time CDF of the heuristic close to the Baseline. In <Par,10> and <Par,50>, one can hardly discern differences between the Baseline and the corresponding heuristic results. Remember that the Pareto traffic has bursts of I/O requests followed by longer idle periods. Since our heuristic modulates the LOW_WM based on the number of requests (rather than time), this modulation is done fast enough during the bursts so that the response time of those requests are not significantly compromised, and is done slow enough during the longer idle periods that the energy savings are obtained during those times. In the exponential traffic, while there are some deviations from the baseline, we are still able to keep over 90% of requests within a 5% response time degradation margin with a $n = 250$ window, while giving over 35% energy savings (in the quadratic model). Changing the power model from quadratic to linear does not change the trends as was pointed out earlier, and we still find over 25% energy savings.

6.2 Controlling UT and LT for Power-Performance Trade-offs

The DRPM heuristic provides two additional parameters (in addition to n already considered) - UT and LT - for modulating the RPM control. By keeping UT where it is, and moving LT up (closer to UT), we can allow the disks to transition to even lower RPM levels, thereby saving even more energy without compromising significantly on performance. This is shown by comparing the results for UT=15% and LT=10% in Figure 11 (a) with those of the results in

Figure 10 (at least for higher loads).

Similarly, one can bring the UT parameter closer to LT, to reduce response time degradation without significantly changing the energy results. This is shown by comparing the results for UT=8% and LT=5% in Figure 11 (b) with those of the results in Figure 10.

This heuristic thus provides an elegant approach for determining where one wants to operate in the power-performance profile.
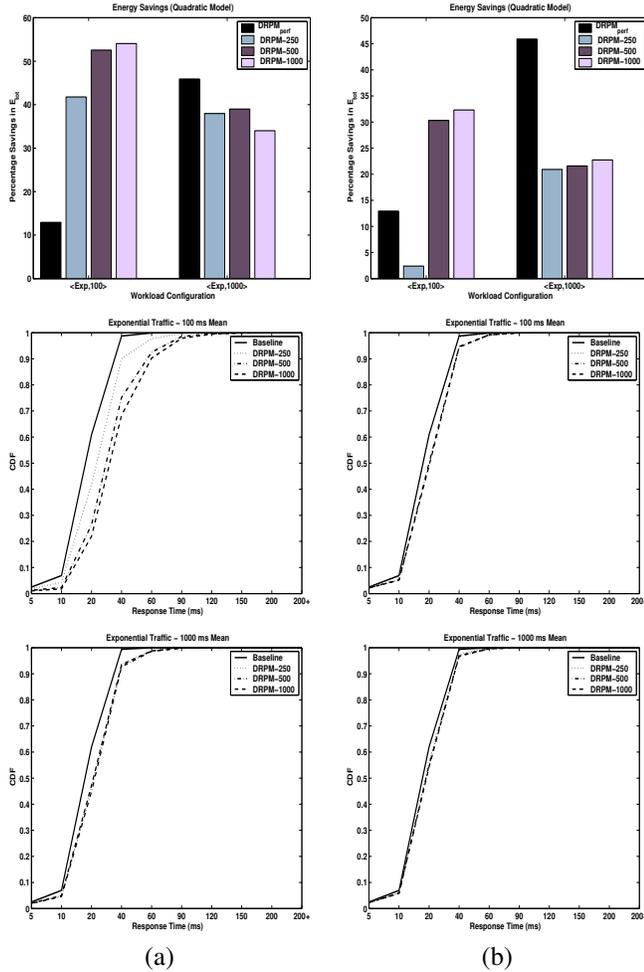

| (a) | (b) |

Figure 11. Controlling UT and LT for Power-Performance Tradeoffs. (a) presents the results for UT=15%,LT=10%. (b) presents the results for UT=8%,LT=5%.

7 Issues in Implementing DRPM Disks

Having demonstrated the potential of DRPM, it is important to understand some of the ramifications in its physical realization:

- **Providing Speed Control**

  As mentioned in section 3, speed control in DC brush-less PM motors can be achieved using PWM tech-niques. PWM achieves speed control by switching on and off the power supply to the motor at a certain fre-quency (called the duty cycle). The choice of duty cy-cle determines the motor speed. The design of such speed-control mechanisms can be found in [3].

- **Head Fly-Height**

  The height at which the disk head slider flies from the platter surface depends on the linear velocity of the spinning platter, v, which can be expressed as $v = 2\pi r f_{spin}$, where r is the radius of the disk and f_{spin} is the frequency of rotation (measured in RPM). The fly height needs to be more or less constant over the entire range of linear velocities supported by the given spindle system. The Papillon slider presented in [24] is capable of maintaining this constant fly height over the range of RPMs that we have considered.

- **Head Positioning Servo and Data Channel Design**

  In hard-disks, positioning the head requires accurate information about the location of the tracks. This in-formation is encoded as servo-signals on special servo-sectors, that are not accessible by normal read/write operations to the disk. This servo information is given to the actuator to accurately position the head over the center of the tracks. The servo information needs to be sampled at a certain frequency to position the head properly. As the storage density increases, the number of Tracks Per Inch (TPI) increases, requiring higher sampling frequencies. This sampling frequency is di-rectly proportional to the spinning speed of the disk f_{spin}. Therefore, at lower f_{spin} it might not be possi-ble to properly sample the servo information. [39] ad-dresses this problem by designing a servo system that can operate at both low and high disk RPMs along with a data channel that can operate over the entire range of data-rates over the different RPMs (the data rate of a channel is directly proportional to f_{spin}).

- **Idle-Time Activities**

  Server environments optimize idle periods in disks to perform other operations such as validating the disk contents and optimizing for any errors ([21]). The fre-quencies of such operations are much lower than the idle times themselves to really have a significant con-sequence on the effectiveness of power saving tech-niques. Still, it is possible that DRPM may be more useful for such activities, since it allows those per-formance non-critical operations to be undertaken at a relatively slow RPM (for energy savings), while tra-ditional power mode control of transitioning the disk completely to a standby state prevents such activities.

- **Smart Disk Capabilities**

  The anticipated smart disks [22, 32] provide an ex-cellent platform for implementing DRPM algorithms, and also provide the flexibility of modulating the algo-rithm parameters or even changing the algorithm en-tirely during the course of execution.

The effect of RPM modulation on disk reliability needs further investigation. On the one hand, we have been in-creasing the number of disks in arrays to not only enhance performance, but also for availability. This in turn has ac-centuated the power problem, which this paper has tried to address. In doing so, it is conceivable that we may need more disks for hot spares in case RPM modulation can worsen MTTFs. This vicious cycle between perfor-mance, power and availability warrants a further investiga-tion which we plan to undertake in the future.

8   Concluding Remarks

This paper has presented a new approach to address the growing power problem in large disk arrays. Instead of completely spinning down disks, which can incur significant time and power costs, this paper proposes to modulate the RPM of disks dynamically. The resulting DRPM mechanism has been shown to find more scope for power savings when idle times are not very long compared to traditional power management (TPM) techniques that have been proposed for laptop/desktop disks. In addition, it also allows the option of servicing requests at a lower RPM when performance is not very critical, to provide additional power savings. Finally, it can be combined with TPM techniques to amplify the power savings.

We have proposed timing and power models for the DRPM mechanism, and have conducted a sensitivity analysis of different hardware parameters. In addition, we have presented a heuristic that can be used in practice to benefit from the DRPM mechanism to allow trade-offs between power savings and performance benefits. Detailed simulations have shown that we can get considerable energy savings without significantly compromising on performance.

It is to be noted that the heuristic presented here is one simple way of using the DRPM mechanism though it is conceivable that one can optimize/change this further to get higher power savings, or to limit the performance degradation.

References

[1] W. Blount. Fluid Dynamic Bearing Spindle Motors: Their Future in Hard Disk Drives. IBM White Paper.

[2] P. Bohrer, E. Elnozahy, T. Keller, M. Kistler, C. Lefurgy, C. McDowell, and R. Rajamony. *The Case for Power Management in Web Servers*, chapter 1. Kluwer Academic Publications, 2002.

[3] S. Cameron and F. Carobolante. Speed Control Techniques for Hard Disc Drive Spindle Motor Drivers. In *Proceedings of the Annual Symposium on Incremental Motion Control Systems and Devices*, pages 66–82, June 1993.

[4] J. Chase, D. Anderson, P. Thakur, A. Vahdat, and R. Doyle. Managing Energy and Server Resources in Hosting Centers. In *Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP'01)*, pages 103–116, October 2001.

[5] J. Chase and R. Doyle. Balance of Power: Energy Management for Server Clusters. In *Proceedings of the 8th Workshop on Hot Topics in Operating Systems (HotOS)*, May 2001.

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

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

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

[9] R. Golding, P. Bosch, and J. Wilkes. Idleness is not sloth. Technical Report HPL-96-140, HP Laboratories, October 1996.

[10] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, and H. Franke. Dynamic Speed Control for Server Class Disks. Technical Report CSE-03-007, The Pennsylvania State University, March 2003.

[11] 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.

[12] E. Harris, S. Depp, W. Pence, S. Kirkpatrick, M. Sri-Jayantha, and R. Troutman. Technology Directions for Portable Computers. *Proceedings of the IEEE*, 83(4):636–658, April 1995.

[13] T. Heath, E. Pinheiro, J. Hom, U. Kremer, and R. Bianchini. Application Transformations for Energy and Performance-Aware Device Management. In *Proceedings of the International Conference on Parallel Architectures and Compilation Techniques (PACT)*, pages 121–130, September 2002.

[14] 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.

[15] I. Hong and M. Potkonjak. Power Optimization in Disk-Based Real-Time Application Specific Systems. In *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*, pages 634–637, November 1996.

[16] Hydrodynamic Bearing Technology In Quantum Hard Disk Drives. http://www.maxtor.com/quantum/src/whitepapers/ wp_fbplusas.htm.

[17] IBM Hard Disk Drive - Ultrastar 36ZX. http://www.storage.ibm.com/ hdd/ ultra/ ul36zx.htm.

[18] IBM Hard Disk Drive Load/Unload Technology. http://www.storage.ibm.com/hdd/library/whitepap/ load/load.htm.

[19] M. Jabbar. Disk Drive Spindle Motors and Their Controls. *IEEE Transactions on Industrial Electronics*, 43(2):276–284, April 1996.

[20] J. Jeppesen, W. Allen, S. Anderson, and M. Pilsl. Hard Disk Controller: The Disk Drive's Brain and Body. In *Proceedings of the International Conference on Computer Design (ICCD)*, pages 262–267, September 2001.

[21] H. Kari, H. Saikkonen, and F. Lombardi. Detecting Latent Faults in Modern SCSI Disks. In *Proceedings of the International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 403–404, January 1994.

[22] K. Keeton, D. Patterson, and J. Hellerstein. A Case for Intelligent Disks (IDISKs). *SIGMOD Record*, 27(3):42–52, September 1998.

[23] T. Kenjo. *Electric Motors and Their Controls*. Oxford University Press, 1991.

[24] N. Kojima, K. Okada, M. Yotsuya, H. Ouchi, and K. Kawazoe. Flying characteristics of novel negative pressure slider "Papillon". *Journal of Applied Physics*, 81(8):5399–5401, April 1997.

[25] K. Li, R. Kumpf, P. Horton, and T. Anderson. Quantitative Analysis of Disk Drive Power Management in Portable Computers. In *Proceedings of the USENIX Winter Conference*, pages 279–291, 1994.

[26] Y.-H. Lu, E.-Y. Chung, T. Simunic, L. Benini, and G. Micheli. Quantitative Comparison of Power Management Algorithms. In *Proceedings of the Design Automation and Test in Europe (DATE)*, March 2000.

[27] Maxon motor. ftp://ftp.maxonmotor.com/Public/Download/ catalog_2002/Pdf/02_157_e.pdf.

[28] 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.

[29] A. Papathanasiou and M. Scott. Increasing disk Burstiness for Energy Efficiency. Technical Report 792, The University of Rochester, November 2002. D.

[30] D. Patterson, G. Gibson, and R. Katz. A Case for Redundant Arrays of Inexpensive Disks (RAID). In *Proceedings of ACM SIGMOD Conference on the Management of Data*, pages 109–116, 1988.

[31] T. Pering and R. Broderson. The Simulation and Evaluation of Dynamic Voltage Scaling Algorithms. In *Proceedings of the International Symposium on Low-Power Electronics and Design (ISLPED)*, pages 76–81, August 1998.

[32] E. Riedel, C. Faloutsos, G. Ganger, and D. Nagle. Data Mining on an OLTP System (Nearly) for Free. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 13–21, May 2000.

[33] C. Ruemmler and J. Wilkes. UNIX Disk Access Patterns. In *Proceedings of the USENIX Winter Technical Conference*, pages 405–420, January 1993.

[34] N. Schirle and D. Lieu. History and Trends in the Development of Motorized Spindles for Hard Disk Drives. *IEEE Transactions on Magnetics*, 32(3):1703–1708, May 1996.

[35] W. Sereinig. Motion-Control: The Power Side of Disk Drives. In *Proceedings of the International Conference on Computer Design (ICCD)*, pages 268–273, September 2001.

[36] TPC-C Executive Summary - Dell PowerEdge 6650/4/1.6GHz. http://www.tpc.org/tpcc/results/tpcc_result_detail.asp? id=102053101.

[37] Ultrastar 36xp negative air pressure bearing. http://www.almaden.ibm.com/sst/html/hdi/abs.htm.

[38] A. Weissel, B. Beutel, and F. Bellosa. Cooperative I/O - A Novel I/O Semantics for Enegy-Aware Applications. In *Proceedings of the Symposium on Operating Systems Design and Implementation (OSDI)*, August 2002.

[39] H. Yada, H. Ishioka, T. Yamakoshi, Y. Onuki, Y. Shimano, M. Uchida, H. Kanno, and N. Hayashi. Head positioning servo and data channel for hdd's with multiple spindle speeds. *IEEE Transactions on Magnetics*, 36(5):2213–2215, September 2000.

[40] R. Youssef. RAID for Mobile Computers. Master's thesis, Carnegie Mellon University Information Networking Institute, August 1995.