The Pennsylvania State University

The Graduate School

Department of Computer Science and Engineering

POWER MANAGEMENT OF ENTERPRISE STORAGE SYSTEMS

A Thesis in

Computer Science and Engineering

by

Sudhanva Gurumurthi

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

August 2005

The thesis of Sudhanva Gurumurthi has been reviewed and approved* by the following:

Anand Sivasubramaniam
Professor of Computer Science and Engineering
Thesis Adviser
Chair of Committee

Mary Jane Irwin
A. Robert Noll Chair of Engineering, Computer Science and Engineering

N. Vijaykrishnan
Associate Professor of Computer Science and Engineering

Mahmut Kandemir
Associate Professor of Computer Science and Engineering

Natarajan Gautam
Associate Professor of Industrial and Manufacturing Engineering

Raj Acharya
Professor of Computer Science and Engineering
Head of the Department of Computer Science and Engineering

*Signatures are on file in the Graduate School.

Abstract

Data-centric services, such as transaction processing systems and search-engines, sustain the demands of millions of users each day. These services rely heavily on the I/O subsystem for their data storage and processing requirements. Technological improvements in hard disk drive densities and data-rates have been key enablers in the realization of these storage systems. However, server storage systems consume a large amount of power, leading to higher running costs, increased stresses on the power supply, higher failure rates and detrimental environmental impacts.

This thesis makes four contributions towards understanding the nature of the power problem and developing effective solutions to combat its effects. First, it shows that power management is a challenging problem for enterprise storage systems and traditional techniques to reduce power are relatively ineffective in such systems. Second, it shows that the heat that is dissipated due to the high power consumption would significantly restrict the ability to sustain the pace of performance growth of disks in the near future. Third, it proposes a novel disk drive architecture called *DRPM* that can provide significant savings in energy with very little loss in delivered performance. It also shows how such savings can be attained in practice via a control-policy and considers engineering issues in building such a device. Finally, the thesis shows how DRPM can be leveraged to provide a spectrum of dynamic thermal management policies which would pave the way for maintaining good performance growth in future disk drives.

Table of Contents

List of Tables

List of Figures

Acknowledgments

My years at Penn State have been highly memorable and there are several people to whom I am thankful to for making it so. As a complete list of all the people who played a role in this would require a book as long this thesis itself, I would have to restrict my acknowledgments to just a handful of individuals.

I am highly indebted to my advisor, Prof. Anand Sivasubramaniam, who provided me the support to carry out my research and made my stay at Penn State productive and enjoyable. He has always made time to talk to me about both research and career issues. He was a great mentor and a good friend. Professors Mary Jane Irwin, N. Vijaykrishnan, and Mahmut Kandemir were always supportive and helpful. I would like to thank Prof. Raj Acharya for providing excellent research facilities and also for his encouragement throughout my stay in the department.

The time I spent at industry on internships were one of the key highlights of my graduate-school life. It was a pleasure working with Karthick Rajamani and Tom Keller at IBM Austin. Shubu Mukherjee (Intel) was a great mentor and taught me many things about conducting disciplined research and presenting it clearly.

I have been fortunate to have a marvelous set of friends and colleagues. My colleagues at CSL have been great folks to work with and the countless hours that I have spent in collaborative work and other time-sinks with them made the hours spent in the lab both lively and enriching. In particular, I would like to acknowledge Ning An, Yanyong Zhang, Murali Vilayannur, Anshuman Parashar, Gokul Kandiraju, Chun Liu, Jianyong Zhang, Youngjae Kim, Partho Nath, and Amitayu Das. I am grateful to Anand Jayaraman and Madhavi Vuppalapati for being a steady source of friendship and support. Aravind Killampalli and Manthram Sivasubramaniam, who have been friends from my high-school days, continued be close, inspite of the geographic separation and our different career paths. I would like to thank Hai Huang, Soraya Ghiasi, Raz

Cheveresan, Paul Racunas, and Nick Wang for making my internship stints fun and tolerating me as I dragged them along on my travels in search of interesting vistas and good restaurants.

My parents have been the most important people in my life. They have been my primary source of emotional support and encouragement, through both good times and testing periods, even though we are on opposite sides of the world. I dedicate this thesis to them.

I would like to thank the National Science Foundation for funding my research over the years through their various grants.

Chapter 1

Introduction

The growth of business enterprises and the emergence of the Internet has lead to a proliferation of server-centric applications. Data-centric services, such as file servers, web portals, and transaction processing systems have become commonplace in not just large corporations, but also smaller business enterprises and academic institutions. Further, search engines, e-mail servers, stock trading and financial transactions, and other Internet-based commercial services sustain the needs of potentially millions of users each day. All these services are highly data-centric and rely heavily on the Input/Output (I/O) subsystem for their data storage and processing requirements.

There has been a tremendous growth in the storage demands in the enterprise sector, as depicted in Figure 1.1 [25]. The graph shows the growth broken down into different server market segments. Not surprisingly, the market is dominated by Windows and Unix-based systems.

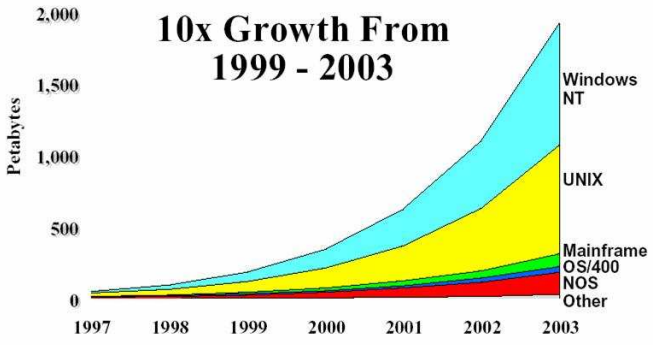


Fig. 1.1. Growth in Enterprise Storage Market

This brisk growth rate can be attributed to the fact that most institutions generate large volumes of data that needs to be stored reliably, cost-effectively, and that must also be accessed efficiently. The usual way that this is achieved is by forming storage arrays using a set of commodity disk-drives. These storage arrays, that use some form of RAID [80], provide the necessary capacity, bandwidth, and fault-resilience that is expected of servers. Moreover, increasingly stringent compliance requirements force these institutions to retain the generated data for extended periods of time, thereby leading to a significant investment in storage. These trends are expected to continue into the future as well.

However, with such steep growth comes the problem of high power consumption. Data centers, which are special buildings that house these server systems, already consume several Megawatts of power. This leads to large electricity bills that can run into millions or even billions of Dollars a year, nearly a third of which is due to the disk-drives [88]. Such high electricity costs increase the Total Cost of Ownership (TCO) which eventually get passed on to the customers who use them to host their applications. A second problem with such high power consumption is heat. With increased power consumption, more heat is generated. This heat needs to be removed from the system. This is done using component-level cooling solutions like heat-sinks in conjunction with external cooling such as fans and air-conditioning. Over the years, the heat-density (expressed in Watts-per-square-foot) has been steadily increasing in server storage products [41] and we have been able to keep pace with it by provisioning the necessary amount of cooling. However cooling systems for the higher heat densities are prohibitively expensive [110], whereby the storage architects have to cut back on the aggressiveness of their designs, which also entails a performance penalty.

Furthermore, cooling systems already constitute a significant fraction of the power cost of running a data center. This is illustrated by the power consumption breakdown of a typical data center shown in Figure 1.2. The pie chart shows the electricity use of different components in the data center of a large financial corporation in New York City [107]. The overall power

consumption of this data center is roughly 4.8 Megawatts. Thus, increasing the amount of cooling that we provision, in terms of air-conditioning systems, is neither cost-effective nor scalable.

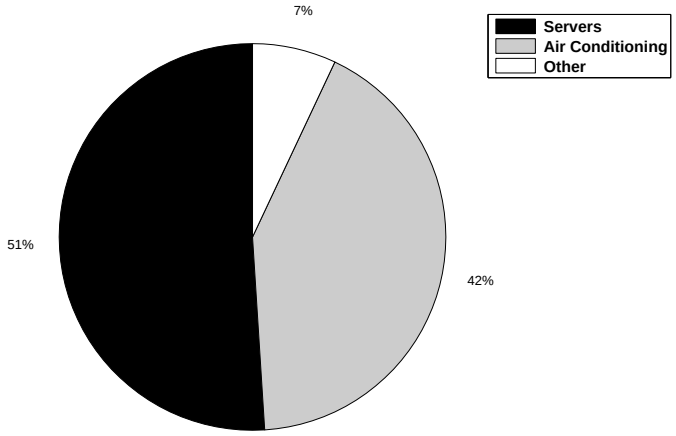


Fig. 1.2. Electricity Use in a Data Center

Finally, cutting back on the amount of cooling that we provide is not possible since the reliability of the server components, especially the disk drives, is highly dependent on the temperature. High temperatures can cause off-track errors due to thermal tilt of the disk stack and actuator, or even cause head crashes due to the out-gassing of spindle and voice-coil motor lubricants. Elevated temperatures can also increase wear-out of the bearings [44]. It has been shown that even a fifteen degree Celsius rise from the ambient temperature can double the failure rate of a disk drive [4]. This makes it imperative for the disk to operate below a *thermal envelope* and designing the disk to operate within this envelope is one of the key determinants in drive design and consequently the I/O performance.

This thesis makes several contributions towards understanding this problem and designing solutions to overcome the impediments posed by power to the I/O performance. These contributions (and the overall organization of this thesis) are briefly summarized below:

- Chapter 2 undertakes a detailed analysis of the I/O behavior of popular server workloads like On-Line Transaction Processing (OLTP) and On-Line Analytical Processing (OLAP) to explore the feasibility of applying traditional spindown-based disk power management techniques. The analysis shows that traditional power management is ineffective for server systems due to the nature of the workloads and also the physical characteristics of the disk drives themselves. The chapter also presents some simple approaches to tradeoff power and performance in large-scale storage systems using RAID parameter tuning.

- Chapter 3 presents an analysis of the thermal behavior of disk-drives, capturing several aspects of their design such as geometry, data-organization, ECC/servo overheads etc. This study shows that we are approaching a point where is going to be very difficult to sustain the performance growth that we have enjoyed for nearly the past two decades due to the increased heat dissipation of the higher performance designs. The fall-off from the projected roadmap is quantified and the impact of changes in other technology trends, like the cooling system and smaller form-factors on the disk-drive "thermal roadmap" are analyzed.

- In Chapter 4, a novel disk-drive design technique called Dynamic RPM (*DRPM*) is presented, which allows for run-time rotation-speed modulation. First, the characteristics of the device are modeled followed by an analysis of its potential. A control-policy for leveraging the DRPM technology is proposed and evaluated, and finally the engineering issues involved in building such a disk drive are discussed.

- The design constraints posed by the thermal envelope are tackled in Chapter 5. First, the need for faster disks in the future is motivated via a workload-driven study. Then, a set of

Dynamic Thermal Management (DTM) techniques are developed and their applicability in different drive-design scenarios is presented.

Finally, Chapter 6 concludes this thesis and identifies future research directions.

Chapter 2

The Challenge of Power Management for Enterprise Storage Systems

2.1 Introduction

Disk power management has been an active area of research in single-disk systems like laptops and desktops [22, 67, 43, 42, 71, 31]. Before venturing into a detailed analysis of power management, it is important to understand what consumes power in a disk drive and the basic principles behind any power-management technique. A hard disk drive is an electro-mechanical device. The two main sources of power consumption in a disk drive are the spindle-motor (SPM), which is used to rotate the platter assembly and the voice-coil motor (VCM), which is activated when the arms need to be moved. When the disk is just spinning and not servicing any requests, it is said to be in an *idle* power mode, and the bulk of the power is consumed by the SPM. However, when a request comes to the disk and a seek needs to be performed, the VCM needs to be activated as well (*seek* mode). The actual power consumed by the seek operation depends on the amount of time needed to accelerate and decelerate the arms, for which time the VCM is active (deceleration is performed by reversing the current in the VCM). The actual transfer of bits between the magnetic media and the electronic components in the drive takes place when the drive is in the *active* mode, where the read/write channel (also known as the data channel) in the drive electronics is enabled and thus consumes additional power. Overall, across all these three disk modes, the SPM is always active and the bulk of the power is also consumed by it. Thus, the traditional approach to disk power management has been to turn off the SPM and physically spin the platters down into a low-power state, known as the *standby* mode.

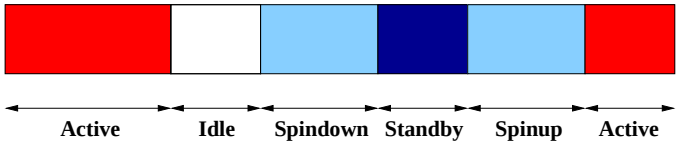


Fig. 2.1. Traditional Disk Power Management. No I/O access to the physical-media is possible when the disk is in the *Spindown*, *Standby*, and *Spinup* modes.

A typical timeline of how this traditional power management is implemented is depicted in Figure 2.1. Disk power management involves two steps, namely, detecting suitable idle periods and then spinning down the disk into the standby mode whenever it is predicted that the action would save energy. Detection of idle periods usually involves tracking some kind of history to make the 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). One could also pro-actively spin up the disk ahead of the next request if predictions can be made accurately, although most prior studies have not done this. Many idle time predictors use a time-threshold to find out the duration of the next idle period and use it to decide whether to effect a transition to the standby mode to save power.

There are several differences between single-user laptop/desktop storage systems and those in servers. First, servers do not have just a single disk but a multiplicity of them, typically configured into RAID arrays. Second, the workloads in server environments are significantly different from those on laptops/desktops environments. Servers have to deal with several users/transactions at the same time, with workloads being much more I/O intensive. The server workloads typically have a continuous request-stream that needs to be serviced, instead of the relatively intermittent activity that is characteristic of the more interactive desktop and laptop

environments. Further, response time is a much more critical issue in web-servers and database-servers in the corporate world (compared to desktop workloads), and we need to be careful when there is the issue of degrading performance to gain power savings. In addition, the I/O sub-systems in server environments offer several more parameters for tuning (RAID configuration, number of disks, striping unit, etc.) as opposed to single disk systems. Typically, these have been tuned for performance, and it is not clear whether those values are power efficient as well. Finally, server disks are physically different from their laptop and desktop counterparts. They have much larger spinup and spindown times and are designed for continuous operation and higher vibration-tolerance while serving I/O requests. All these differences in system environments and workload behavior warrant a rethinking of the way power-management needs to be done for these systems.

Transaction processing workloads are amongst the most common and I/O intensive, of the commercial applications. The focus of this work is on the TPC-C and TPC-H workloads [106]. These workloads are extensively used in the commercial world to benchmark hardware and software systems. TPC-C is an On-Line Transaction Processing (OLTP) benchmark, that uses queries to update and lookup data warehouses. TPC-H, in contrast, involves longer-lived queries that analyze the data for decision-making (On-Line Analytical Processing - OLAP).

In this chapter, a detailed and systematic study of the characteristics of server workloads, especially those that influence disk power management is presented, and the interplay between power and performance for these workloads is examined. This is conducted with a trace-driven simulation using the DiskSim [30] simulation infrastructure. DiskSim provides a detailed disk-timing model that has shown to be accurate [28], which is augmented for power measurements.

First, we show that traditional disk power management schemes proposed in desktop/laptop environments are not very successful in these server workloads, even if we can design the disks to spinup and spindown very fast and predict the idle periods accurately. Then, we investigate the effect of tuning different RAID parameters to balance power and performance when configuring

such storage arrays. We shall demonstrate that tuning the RAID configuration, number of disks, and stripe size have more impact from the power angle and that the values of system parameters for best performance are not necessarily those that consume the least power, and vice-versa.

The rest of the chapter is organized as follows. Section 2.2 presents the related work. Section 2.3 provides a brief overview of the RAID configurations used in this paper. Section 2.4 describes the workloads and metrics used in the evaluation, along with details of the simulated hardware. Section 2.5 presents the results of the study.

2.2 Related Work

Disk power management has been extensively studied in the context of single disk systems, particularly for the mobile/laptop environment [21, 70, 78]. A fixed threshold idle-predictor is used in [67], 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 spindown threshold could itself be varied adaptively over the execution of the program [22, 43]. A detailed study of idle-time predictors and their effectiveness in disk power management has been conducted in [31]. Lu et al. [70] provide an experimental comparison of several disk power management schemes proposed in literature on a single disk platform. The IBM Adaptive Battery Life Extender (ABLE) [2] provides power management for mobile disks.

If we move to high-end server class systems, previous work on power management has mainly focused on clusters, employing techniques such as shutting off entire server nodes [12], dynamic voltage scaling [7], or a combination of both [24]. There has also been work to reduce the power consumption by balancing the load between different nodes of a cluster [85]. Shutting off nodes is possible if the computational load can be re-distributed or there exist mirrors for the data. The application of voltage scaling can reduce the CPU energy consumption, and has indeed been shown to reduce energy up to 36% for web-server workloads [7]. Investigation of power optimization for SMP servers has looked into optimizing cache and bus energy [74].

The focus of this work is on power optimizations for the I/O (disk) subsystem, particularly in the context of transaction processing workloads. In web server workloads (where there are fewer writes), duplication of files on different nodes offers the ability to shut down entire nodes of the web serving cluster (both CPU and its disks) completely. Further, a web page (file) can be serviced by a single node (disks at that node), and the parallelism on a cluster is more to handle several requests at the same time rather than parallelizing each request. Interestingly, there has been a study on the the feasibility of performing disk power management in network servers [9] and the conclusions are quite similar to the ones presented in this chapter.

There have been some studies that have attempted power management in the context of disk arrays. [115] looked into minimizing the disk energy consumption of a laptop disk by replacing it with an array of smaller form-factor disks. Since the power consumed by a disk is a direct function of its size (by nearly the fifth-power), if a single disk is replaced with smaller disks, forming an array, and we keep as many disks in a spundown state for as long as possible, one could obtain good energy savings. This was done by using a Log-Structured File-System (LFS) [94]. The LFS cache was used to delay writes until a disk had to be spunup and also to buffer reads. However, this study does not look at striping the data across these disks for I/O parallelism, and is thus not applicable to server environments. In [17], the authors have proposed replacing a tape-backup system with an array of disks that are kept in the spundown state as long as possible. This work targets archival and backup systems, where idleness of the disks is much easier to exploit, and writes overwhelm the reads.

2.3 RAID Overview

Redundant Array of Independent/Inexpensive Disks (RAID) [80] employs a bunch of disks to serve a request in parallel, while providing the view of a single device to the request. If there are n disks, with each disk having a capacity of B blocks, then the RAID address space can be visualized as a linear address-space from 0 to $nB - 1$. The unit of data distribution across

the disks is called the *striping-unit* or just *stripe*. A stripe consists of a set of consecutive blocks of user data. Since there are multiple disks present, the reliability of the array can go down. Consequently, RAID configurations use either parity or mirroring for error detection and recovery. There are several RAID configurations based on how the data is striped and how the redundancy is maintained. We consider RAID levels 4, 5, and 10 (the latter two are among the more popular ones in use today).

In a RAID-4 array of n disks, $n-1$ disks store data, and the remaining disk stores parity. Logical stripe i thus falls on disk i mod $(n-1)$, i.e. the data stripes are assigned to the disks storing data in a cyclic fashion. The parity disk stores the parity information for stripes 1 through $n-1$, n through $2n-1$, and so on. When a read for a logical block is presented, the array controller will figure out the disk(s) and their blocks that are involved, and issue the requests to them. At the same time, the parity blocks for those will also need to be obtained to confirm the validity of the data. In a write operation, apart from involving the data disks, the parity may have to be both read and written (because it may need to be re-calculated). Usually, writes thus become a problem (especially small ones).

RAID-5 works the same way as RAID-4, with the difference being that there is no one disk dedicated for parity. Instead, each disk takes turns serving to hold the parity for a given set of stripes (this is usually called rotating parity - e.g. disk n serves as parity for data stripes 0 through $n-1$, disk 1 serves as parity for data stripes n through $2n-1$ and so on). Consequently, this can avoid the bottleneck of a single parity disk (especially for high small-write traffic), evening out the load across the array.

RAID-10, which has gained popularity in recent times, employs a combination of data *mirroring* (duplication of data on two different disks) and striping. Its specifications are rather loose, and the scheme used in our experiments is explained below. The disk array is split into two mirrors (each with an equal number of disks). Within each mirror, we simply stripe the data across the disks in a cyclic fashion (without any parity). A write will need to update both

mirrors. In the read implementation, we send the request to the mirror which has the shortest request queue at that instant [30]. Ties are broken by picking the mirror with the shortest seek time. RAID-10 provides greater reliability since it can tolerate multiple disk failures.

2.4 Experimental Setup

2.4.1 Workloads

Transaction processing workloads use database engines to store, process and analyze large volumes of data that are critical in several commercial environments. Many of these are also back-end servers for a web-based interface that is used to cater to the needs of several hundreds/thousands of users, who need low response times while sustaining high system throughput.

In this study, two important transaction processing workloads, identified by the Transaction Processing Council (TPC) [106], are used. While ideally one would like to run simulations with the workloads and database engines in their entirety in direct-execution mode, this would take an inordinate amount of time to collect data points with the numerous parameters that are varied in this study. Consequently, device level traces from executions on actual server platforms are used to drive the simulations.

- **TPC-C Benchmark:** TPC-C is an On-Line Transaction Processing (OLTP) benchmark. It simulates a set of users who perform transactions such as placing orders, checking the status of an order etc. Transactions in this benchmark are typically short, and involve both read and update operations. For more details on this benchmark, the reader is directed to [103]. The tracing was performed for a 20-warehouse configuration with 8 clients and consists of *6.15 million I/O references*. The traced system was a 2-way Dell PowerEdge SMP machine with 1.13 GHz Pentium-III processors and 4 10K rpm disks running IBM's

EEE DB-2 [55] on the Linux operating system.

- **TPC-H Benchmark:** This is an On-Line Analytical Processing (OLAP) benchmark and is used to capture decision-support transactions on a database [104]. There are 22 queries in this workload, and these queries typically read the relational tables to perform analysis for decision-support. The trace that is used in this study was collected on an IBM Netfinity SMP server with 8 700 Mhz Pentium III processors and 15 IBM Ultrastar 10K rpm disks, also running EEE DB-2 on Linux, and consists of *18 million I/O references*.

The traces have been collected at the device level and give the timestamp, type of request (read/write), logical block number and number of blocks. The logical block numbers are mapped to the physical disk blocks based on the RAID configuration.

2.4.2 Simulation Environment

The vehicle for carrying out the experiments in this study is the DiskSim simulator [30], augmented with a disk power model. DiskSim provides a large number of timing and configuration parameters for specifying disks and the controllers/buses for the I/O interface. The default parameters that we use in this study are given in Table 2.1. The RPM and disk cache size have been chosen to reflect what was popular in state-of-the-art server systems at the time this study was undertaken.

The power values are taken from the data sheets of the IBM Ultrastar 36ZX [57] disk, which is used in several servers.

The simulated I/O subsystem architecture looks as shown in Figure 2.2 (a) for an 8-disk configuration. The disks in the RAID array are attached to an array-controller using 2 Ultra-3 SCSI buses. In our experiments, half the disks are on each bus (typically, each bus can sustain the bandwidth needs of up to 16 disks). The array controller stripes the data (as per the RAID

Parameter	Value
Number of Disks: 32 Stripe Size: 16 KB	
Capacity	33.6 GB
Rotation Speed	12000 rpm
Disk Cache Size	4 MB
Idle Power	22.3 W
Active (Read/Write) Power	39 W
Seek Power	39 W
Standby Power	12.72 W
Spinup Power	34.8 W
Spinup Time	26 secs.
Spindown Time	15 secs.
Disk-Arm Scheduling Algorithm	Elevator
Bus Type	Ultra-3 SCSI

Table 2.1. Default Disk Configuration Parameters. Many of these have been varied in our experiments.

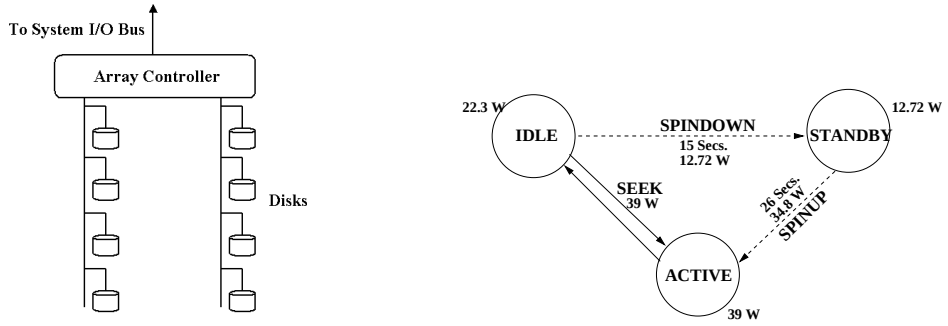
configuration) across all these disks. It also has an on-board cache, which can potentially avoid some of the disk accesses. The array controller is in turn interfaced to the main system via the I/O bus.

Figure 2.2 shows the power mode transitions for each disk. The parameters that are varied include the RAID configuration (4, 5 and 10), the number of disks, and the stripe size.

2.4.3 Metrics

The metrics used in this study are *total energy consumption over all the requests (E_{tot})*, *average energy consumption per I/O request (E)*, *response-time per I/O request (T)*, and *energy-response-time product ($E \times T$)*. 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 trace. All the disk activity (states) are monitored and their duration from the start to the end of the simulation, and this is used to calculate the overall energy consumption by the disks (integral of the power in each state over the



(a) Disk Array Configuration (b) Disk Power Modes

Fig. 2.2. RAID Configuration and Power Modes

duration in that state).

- The energy consumption per I/O request (E) is E_{tot} divided by the number of I/O requests. A previous study on energy management of server clusters also uses a similar metric (Joules per Operation) for capturing the impact of energy/power optimization for a given throughput [11].

- The response-time (T) is the average time between the request submission and the request completion. This directly has a bearing on the delivered system throughput.

- The product of the previous two ($E \times T$) measures the amount of energy or performance we can tradeoff for the other to have an overall beneficial effect. For instance, if we increase the number of disks in the array, and, get much more improvement in response time than the additional energy consumption, then we can consider this optimization to be a net

winner and the product would quantitatively capture this effect. Although computing the energy-delay product requires a complete system characterization to really quantify how the savings of response time in one hardware component can affect the energy of other hardware components (and vice-versa), in this study, the term is used to qualify the *relative importance* of energy and response time.

2.5 Results

In the following subsections, the possible benefits of traditional disk power management are explored using the traditional spindown-based approach for these server workloads and it will be shown that there is not much scope for energy savings with this approach. Subsequently, effect of tuning different hardware parameters on the power-performance behavior is analyzed.

2.5.1 Effectiveness of Conventional Disk Power Management

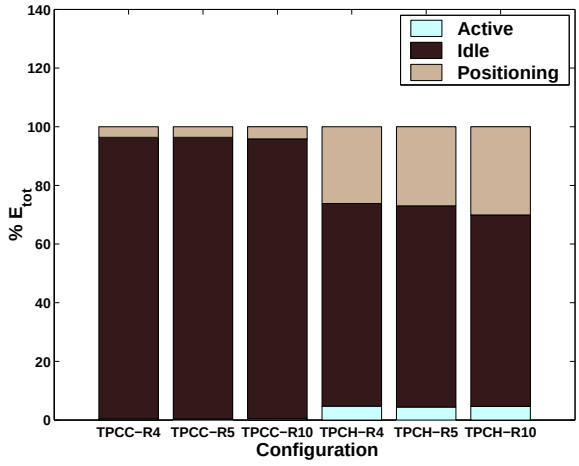


Fig. 2.3. Breakdown of Total Energy Consumption in Different Disk Modes (R4, R5 and R10 refer to RAID-4, RAID-5 and RAID-10 respectively).

In order to optimize disk energy, we need to first examine the energy profile of an actual execution. Figure 2.3 shows the energy consumption breakdown for the three RAID configurations and two workloads, in terms of the energy consumed when in active mode, idle mode, and during head movements (positioning). It can be seen that contrary to expectations, the least amount of energy is actually spent in the active mode. Most of the energy is expended when the disk is idle, and to a lesser extent for positioning the head. This suggests that one should optimize the idle mode for energy and one possibility is to apply traditional power management, which tries to put the disk in the standby mode when it is not performing any operation. This is explored in the next few experiments.

In the following results, the applicability of traditional power management techniques in server environments is investigated. First, the predictability of idle times between disk activities is examined. Next, the duration of idle times is analyzed to see how much scope is there for employing these techniques. Finally, an oracle predictor (that is accurate when predicting idle times both in terms of detecting when an idle period starts and what its duration would be) is employed to see the maximum savings that could be attained using these techniques.

2.5.1.1 The Predictability of Idle-Periods

Prediction of disk requests based on prior history has been a topic of previous research [105]. One commonly used technique is "autocorrelation analysis", wherein data with good correlations are conducive for fitting with ARIMA models [8] as a time-series. Essentially, an autocorrelation at lag i is computed between the observation pairs (idle time periods) $y(t)$ and $y(t+i)$ [31]. The resulting values are plotted as a graph for different lags. A sequence that lends itself to easier prediction models is characterized by a graph which has a very high value for small lags and then steeply falls to low values for larger lags (i.e. recent history has more say on the prediction making it easier to construct a model based on these). Note that observations can be negatively correlated as well. Further, there could be some repetitive sequences which can cause

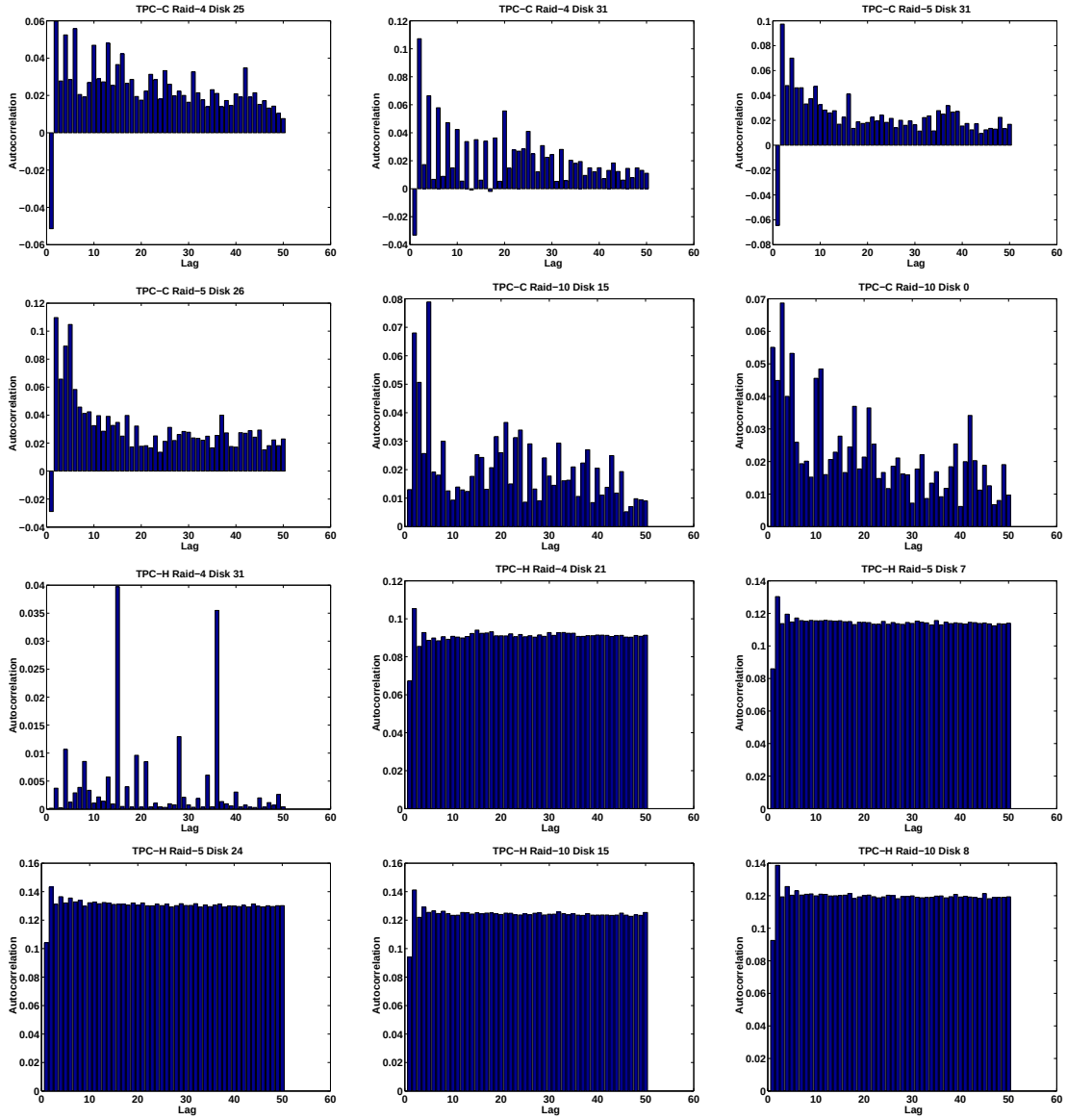


Fig. 2.4. Autocorrelation of Idle Periods for 50 Lags. For each RAID configuration and for each workload, the first graph pertains to the disk that has the least number of distinct idle-periods and the second to the one with the most. Note that the values either degrade slowly (e.g. TPC-H RAID 5 Disk 7) or degrades sharply but the absolute values are quite low (e.g. TPC-C RAID 10 Disk 0)

spikes in the graph, resulting in deviations from monotonic behavior. The reader is referred to [8] for further explanation on such time-series models.

An autocorrelation analysis of the idle periods of disks for 50 lags (lag 0 is not plotted) was conducted and the resulting graphs are shown in Figure 2.4 for the two workloads and all three RAID configurations. For each experiment, the graphs are shown for 2 disks: the disk with the minimum number of distinct idle periods, and the disk with the maximum number of distinct idle periods.

Overall, is can be observed that we do not have good correlation of idle periods. Either the values degrade slowly or degrade sharply but the absolute values are quite low. We show in Table 2.2, the mean (μ) and standard deviation (σ) across all the disks for the first five lags. As we can observe, except in a few cases, the mean does not cross 0.12 and the standard deviation is not very high either. All these results suggest that it is difficult to get good predictions of idle times based on previous (recent) history. These results are in contrast with those for normal workstation workloads, which have been shown to have higher correlations [31].

Benchmark	**RAID Level**	**Lag 1**		**Lag 2**		**Lag 3**		**Lag 4**		**Lag 5**	
		μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
TPC-C	**RAID-4**	-0.064	0.013	0.070	0.020	0.037	0.014	0.045	0.016	0.031	0.009
	RAID-5	-0.044	0.016	0.087	0.019	0.058	0.015	0.057	0.013	0.050	0.016
	RAID-10	0.014	0.020	0.076	0.019	0.057	0.015	0.043	0.014	0.049	0.015
TPC-H	**RAID-4**	0.066	0.012	0.101	0.017	0.083	0.015	0.090	0.014	0.085	0.015
	RAID-5	0.085	0.011	0.130	0.009	0.115	0.011	0.120	0.010	0.116	0.010
	RAID-10	0.092	0.005	0.139	0.005	0.118	0.005	0.125	0.005	0.121	0.005

Table 2.2. Autocorrelation Statistics of All Disks Over 5 Lags. For each lag, μ and σ denote the mean and standard-deviation of the autocorrelation at the given lag respectively.

Note that, though it is difficult to obtain good predictability of the idle periods using time-series analysis, which relies on the recent past for making predictions, it is possible that good prediction accuracy could be obtained by other means. For example, if the idle periods could be fitted to a probability distribution, it may be possible to predict the duration of an idle period with higher probability. However, as shall be shown in Section 2.5.1.3, even with perfect prediction, conventional disk power management does not provide much savings in the energy consumption.

2.5.1.2 The Duration of Idle-Periods

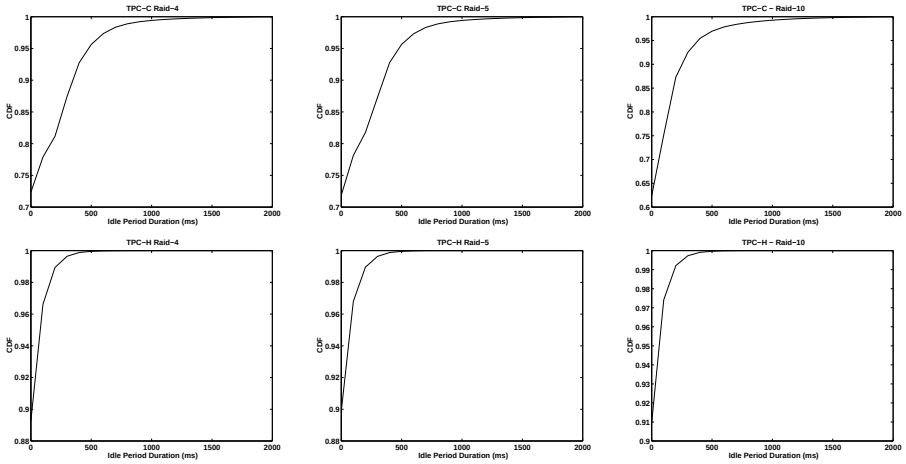
One could argue that even if we are not able to accurately predict the duration of the next idle period, it would suffice if we can estimate that it is larger than a certain value as far as power management is concerned. In order to ascertain the number of idle-periods that could potentially be exploited for power management, the idle periods of the disks is plotted as a Cumulative Density Function (CDF), shown in Figure 2.5.

It can be observed that, whether it be the overall results or that for an individual disk, idle times are extremely short. In fact, the configurations for TPC-H do not show any visible idle times greater than even 1 second. TPC-C, on the other hand, shows some idle times larger than 2 seconds, but this fraction is still quite small (less than 1% in most cases).

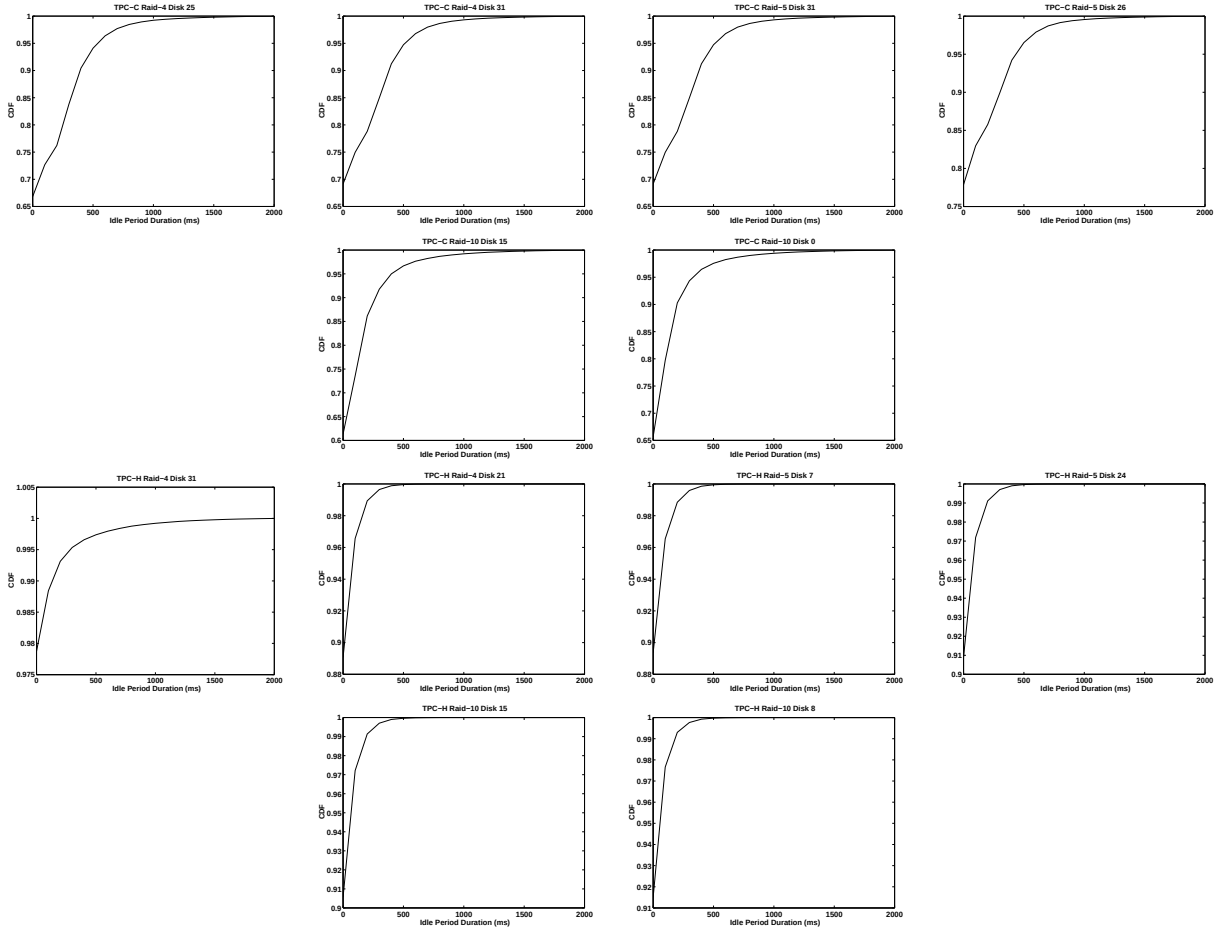
These results indicate that there is not much to be gained with traditional power management techniques, regardless of the predictability of the idle times, if spinup/spindown times are in the ranges indicated in Table 2.1.

2.5.1.3 Limits of Traditional Disk Power Management

Figure 2.6 (a) shows the maximum energy savings that we can hope to get for these workloads and RAID configurations without any degradation in response times for the spin-down/spinup values shown in Table 2.1 for the server disk under consideration. We are assuming

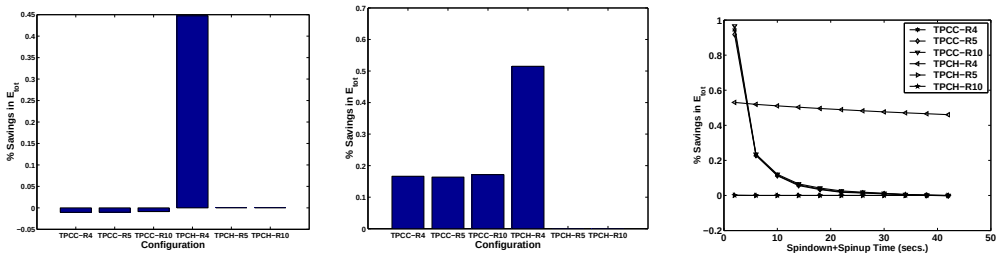


(a) For all Disks



(b) For Disks with the Minimum and Maximum Number of Idle Periods

Fig. 2.5. Cumulative Density Function (CDF) of Idle Periods. This captures what fraction of the total idle times are less than a given value on the x-axis



(a) (b) (c)

Fig. 2.6. Percentage Savings in the Total Energy Consumption with Disk Spindowns using a perfect idle time prediction oracle. The savings are given for (a) current server class disk (spindown+spinup = 41 secs), (b) an aggressive spinup + spindown value (9 secs, for a state-of-the-art IBM Travelstar disk used in Laptops), and (c) for different spinup + spindown values.

an oracle-predictor which has perfect knowledge of the next idle period for this calculation. For TPC-C, performing traditional disk power management actually hurts the overall energy consumption, as the durations of the long idle periods are not sufficient enough to overcome the energy cost of spinning up the disk. Even in the best case (RAID-4 for TPC-H), the percentage energy savings is quite negligible (less than 0.5%).

Another experiment was performed on how this situation would change if we had a laptop-type disk, whose spindown/spinup times are typically much lesser than those of server disks. Even when we assume values of *4.5 seconds* for spindown and spinup (which is in the range of state-of-the-art laptop disks [56]), the energy savings for these workloads are still quite small as is shown in Figure 2.6 (b). Figure 2.6 (c) plots the energy savings that can possibly be obtained for different values of spinup+spindown latencies. As can be seen, even if these latencies become smaller than 2 seconds, we get less than 1% improvement in the energy consumption. It is not clear if we can get these numbers down to those values for server class disks without any degradation in performance. Even if we do, these may need very powerful spindle motors, which can in-turn increase the energy consumption.

Despite the high idle power that was shown in Figure 2.3, we find that idle periods themselves are not very long. The contribution to the idle power is more due to the number of idle periods than the duration of each. This also suggests that it would be fruitful if we can develop techniques to coalesce the idle periods somehow (batching requests, etc.) to better exploit power mode control techniques. A recent study [75], conducted on the feasibility of employing aggressive prefetching to create these bursts has shown that applying traditional power management techniques to server systems is still quite challenging and relatively ineffective. Overcoming these limitations via a re-design of the disk-drive is one of the main contributions of this thesis.

2.5.2 Tuning the RAID Array for Power and Performance

Since power mode control does not appear very productive when we do not have the flexibility of extending response times, it is then interesting to examine what factors within the I/O architecture can influence its design for power-performance trade-offs. In the following discussion, the effect of three important parameters are investigated - the RAID level (4, 5 and 10), the number of disks across which the data is striped, and the stripe size. Traditional studies have looked at these parameters only from the performance angle.

2.5.2.1 Impact of Varying the Number of Disks

Figure 2.7 shows the T, E, and $E \times T$ (as defined in section 2.4.3) as a function of the number of disks that are used in the three RAID configurations for the two workloads. Please note that the third graph is normalized with respect to the leftmost point for each line. The energy-response time product has been normalized this way since we are more interested in the trend of a single line than a comparison across the lines. Figure 2.8 shows the total energy consumption (across all disks) broken down into the active, idle and positioning components.

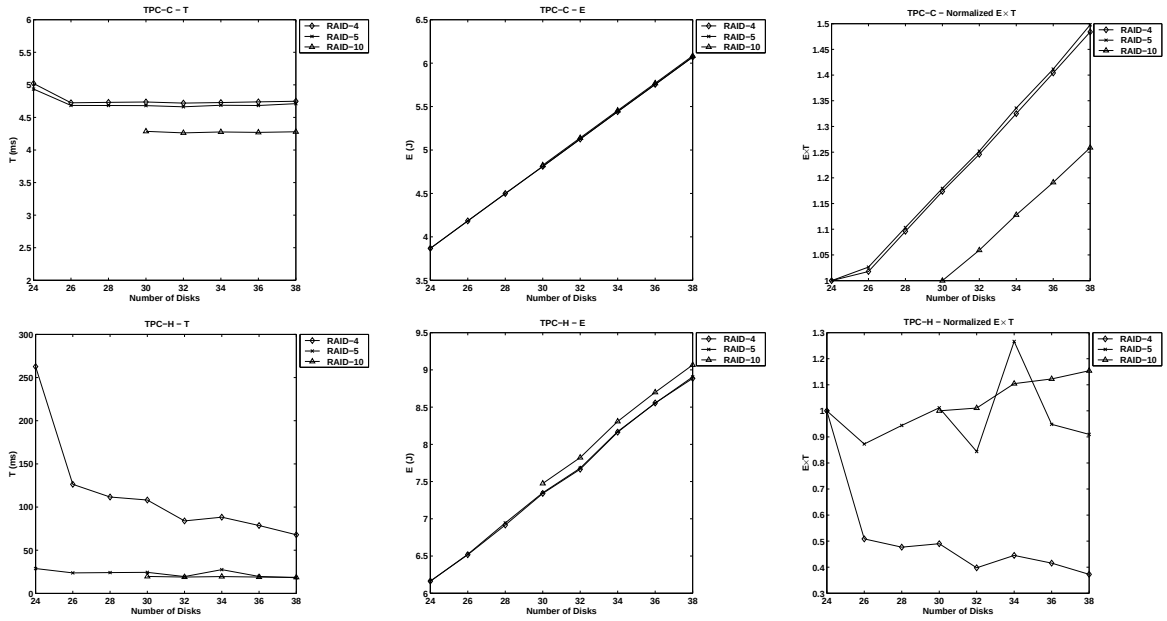


Fig. 2.7. Impact of the Number of Disks in the Array

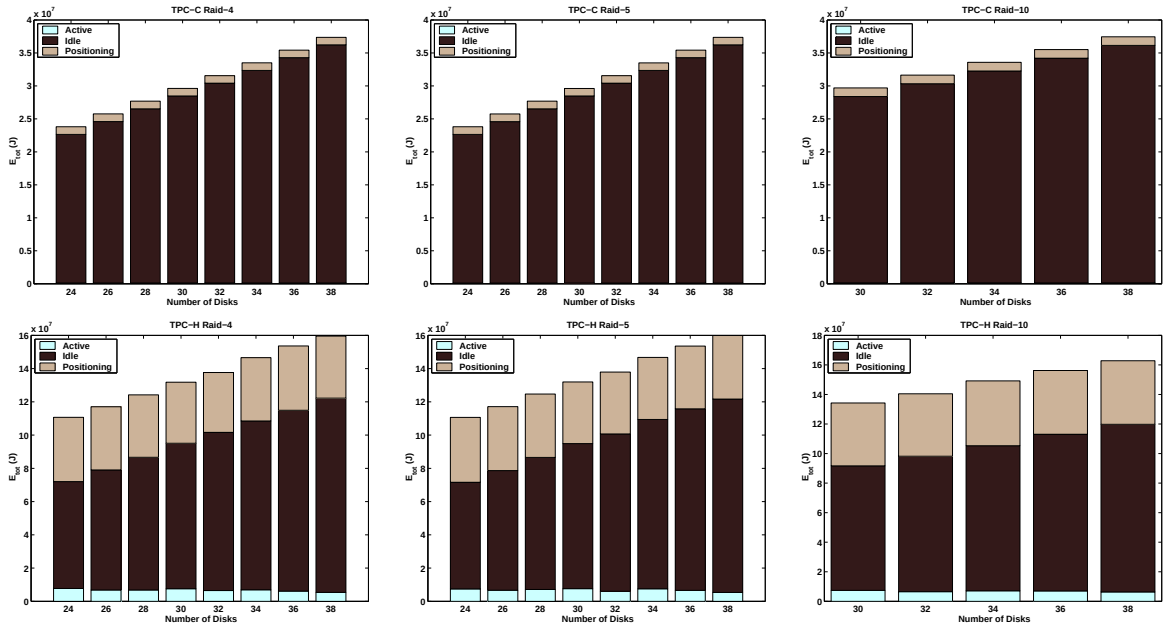


Fig. 2.8. Impact of the Number of Disks in the Array - Breakdown of Total Energy Consumption (E_{tot})

Benchmark	RAID Level	Number of Disks				
		30	**32**	**34**	**36**	**38**
TPC-C	**RAID-4**	1.010 (D) 1.049 (P)	1.010 (D) 1.047 (P)	1.010 (D) 1.047 (P)	1.010 (D) 1.046 (P)	1.010 (D) 1.046 (P)
	RAID-5	1.012	1.011	1.012	1.012	1.012
	RAID-10	1.0000 (M1) 1.0000 (M2)	1.0000 (M1) 1.0000 (M2)	1.0000 (M1) 1.0000 (M2)	1.0000 (M1) 1.0000 (M2)	1.0000 (M1) 1.0000 (M2)
TPC-H	**RAID-4**	1.400 (D) 75.142 (P)	1.263 (D) 56.627 (P)	1.027 (D) 63.248 (P)	1.011 (D) 56.157 (P)	1.011 (D) 46.961 (P)
	RAID-5	1.258	1.024	1.413	1.030	1.014
	RAID-10	1.010 (M1) 1.010 (M2)	1.004 (M1) 1.004 (M2)	1.014 (M1) 1.014 (M2)	1.005 (M1) 1.005 (M2)	1.002 (M1) 1.002 (M2)

Table 2.3. Instantaneous Queue Lengths of All Configuratons. For RAID-4, the average queue length of all the data disks (D) and that of the parity disk (P) are given. For RAID-5, the given value is the average queue length of all the disks in the array. For RAID-10, the average queue length of each mirror, M1 and M2, is presented.

Due the size of the I/O space addressed by the workloads, we chose configurations for RAID 10 starting from 30 disks.

When we examine the performance results, we notice little difference between the three RAID configurations for the TPC-C workload. Though the TPC-C workload has a high amount of write-traffic (14.56% of the total number of requests issued), even for the RAID-4 configuration, beyond 26 disks, there was little variation in the response time. On the other hand in TPC-H, that has a lesser percentage of write requests (8.76%), the RAID-4 configuration showed greater performance sensitivity when the number of disks were increased compared to the other two RAID configurations. This is due to a combination of two factors, namely, the inter-arrival time of the requests and the size of the data accessed per write request. It was found that the average inter-arrival time between requests for RAID-4 TPC-C was 119.78 ms whereas it was 59.01 ms for TPC-H. Further, for TPC-C, 98.17% of the writes spanned at most one stripe-unit whereas in TPC-H, 99.97% of the writes were over 2 stripe-units. These two factors caused a greater amount of pressure to be put on the parity disk. When the number of disks increases, there is some improvement across the three configurations (due to increase in parallelism), but this

improvement is marginal, except for RAID-4 TPC-H. It should be noted that there are some variations when increasing the disks because of several performance trade-offs. The benefit is the improvement in bandwidth with parallelism, and the downside is the additional overheads that are involved (latency for requests to more disks, and the SCSI bus contention). But these variations are not very significant across the range of disks studied here, and the main point to note from these results is that there is not much improvement in response time beyond a certain number of disks.

On the other hand, the energy consumption keeps rising with the number of disks that are used for the striping. If we look at the detailed energy profile in Figure 2.8, we observe that most of the energy is in the idle component (as mentioned earlier). When the number of disks is increased, the positioning component does not change much, but the idle component keeps increasing linearly, impacting the overall energy consumption trend as well.

Comparing the two workloads, we find that the active energy (though a small component in both) is more noticeable in TPC-H compared to TPC-C. This is because the former does much more data transfers than the latter. In terms of head positioning power, again TPC-H has a larger fraction of the overall budget, because the number of seeks and average seek distances are higher in this workload. This happens because TPC-H queries can manipulate several tables at the same time, while TPC-C queries are more localized. There are also differences in the relative dataset sizes between the two benchmarks used this study, which can contribute to this effect.

One interesting observation that can be seen in both the energy results in Figure 2.7 and Figure 2.8 is that the total and the breakdown are comparable across the three RAID configurations for a given number of disks. To investigate this further, in Table 2.3, we show the average queue length to different groups of disks for each RAID configuration: (i) for the normal data disks and the parity disk separately in RAID-4, (ii) average over all the disks for RAID-5, and (iii) for each of the two mirrors in RAID-10. We see that the load on the disks (except for the

parity disk in RAID-4 which is known to be a bottleneck) across the configurations is comparable, regardless of the number of disks in the range chosen. Since the idle energy is directly related to the load on the disks, and the loads are comparable, the overall energy (which is dominated by the idle energy) and its breakdown are more or less similar across the configurations.

If we are only interested in performance, one can keep increasing the number of disks. This is a common trend in the commercial world where vendors publish TPC results with large disk configurations (even though the improvements may be marginal). On the other hand, power dissipation gets worse with the number of disks. The relative influence of the two factors depends on the nature of the workload. The energy growth is in fact much more influential of these two factors for TPC-C and for RAID-10 in TPC-H, and is the determining factor in the energy-response time product graph. However, the performance gains of using more disks plays a more dominant role in the product for RAID-4 and RAID-5 TPC-H.

2.5.2.2 Impact of Stripe Size

Figure 2.9 shows the impact of varying stripe sizes on the three RAID configurations for the two workloads. In these experiments, the number of disks used for each configuration was chosen based on what gave the best energy-response time product in Figure 2.7 (24 for RAID levels 4 and 5 in TPC-C and 30 for RAID-10; 38,32, and 30 for RAID levels 4,5, and 10 of TPC-H respectively). Figure 2.10 breaks down the energy consumption in these executions into the idle, active and positioning components.

It is to be expected that a higher stripe size will lead to less head positioning latencies/overhead, providing better scope for sequential accesses and possibly involve fewer disks to satisfy a request. However, this can adversely affect the degree of parallelism, which can hurt performance. It can be observed that the latter effect is more significant in determining the performance. Between the two workloads, TPC-H requests are larger, and hence the point where the adverse effects become more significant tends to shift to the right for TPC-H. Of the RAID

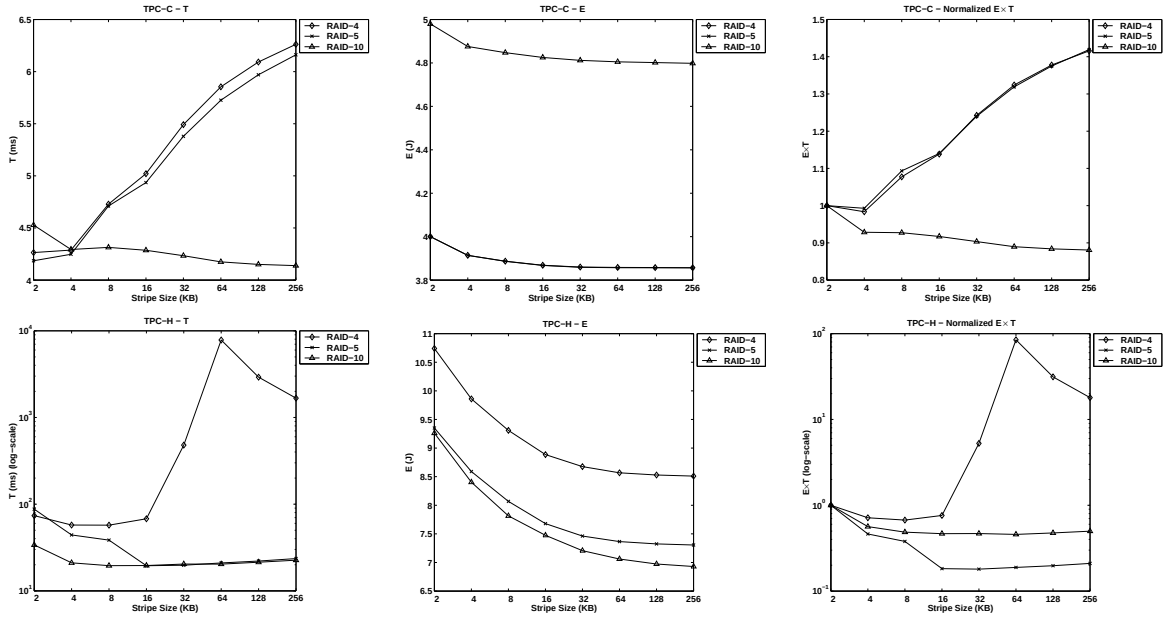


Fig. 2.9. Impact of the Stripe Size

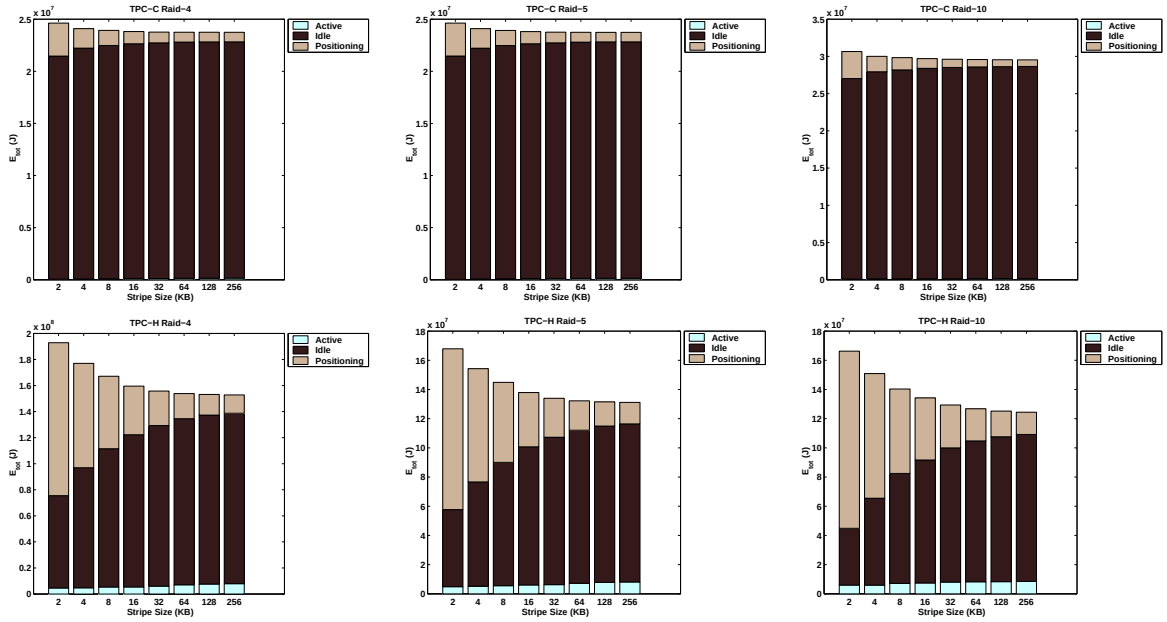


Fig. 2.10. Impact of the Stripe Size - Breakdown of Total Energy Consumption (E_{tot})

configurations, RAID-4 is more susceptible to stripe size changes because the sequentiality problem is higher there, i.e. one disk (the parity) can turn out to become a bottleneck. This becomes worse for TPC-H, which exercises the parity disk to a larger extent due to reasons explained in the previous section, making RAID-4 performance much worse. RAID-5 and RAID-10 for TPC-H are much better (note that the y-axis for TPC-H response time and energy-response time product graphs are in log-scale to enhance the readability of the RAID-5 and RAID-10 lines) though the overall above explanations with regard to stripe size still hold.

Increasing stripe size can lead to fewer disks being involved per request, and higher sequential accesses per disk (reducing seek overheads). Consequently, the head positioning overheads drop, having a consequence on its energy decrease as is shown in the energy profile graphs of Figure 2.10. This drop in positioning energy causes the overall energy to decrease as well. For both the workloads, the decrease in the energy consumption is not significant beyond a certain point. This is because of two other issues: (i) the idle component for the disks not involved in the transfer goes up (as can be seen in the increase in idle energy), and (ii) the active component for the disks involved in the transfer goes up (the number of accesses per disk does not linearly drop with the increase in stripe size, making this number degrade slower than ideal, while the transfer energy per request grows with the stripe size). These two offset the drop in the positioning component. This effect is much more pronounced for TPC-C compared to TPC-H, since in the latter the positioning overhead is much higher, as mentioned in section 2.5.2.1. Finally, the reader should note that despite these overall energy changes, the percentage variations are in fact quite small since the scale of the energy graphs in Figure 2.9 is quite magnified.

The energy-response time product indicates that response time is a much more significant factor in determining stripe size than energy variations components when stripe size is increased.

Different criteria thus warrant a different stripe size. If performance is the only goal, a smaller stripe size of around 4KB appears to be a good choice. If energy is the only goal, then wider stripes of around 256K seem better (though the energy savings may not be significantly

better than a smaller stripe). Overall, a stripe size of 4-16K seems a good choice from the energy-response time product perspective.

2.5.2.3 Implications

The results are put in perspective in Figure 2.11, which shows the trade-offs between performance tuning and energy tuning. The four graphs in this figure show: (a) the percentage increase (over the best-performing version) in response time for the best-energy (per I/O request) version; (b) the percentage increase (over the best-energy version) in energy consumption for the best-performing version; (c) the percentage increase (over the best-performing version) in response time for the best energy-response time product version; and (d) the percentage increase (over the best-energy version) in energy consumption for the best energy-response time product version. Table 2.4 gives the configurations that generate the best E, T, and $E \times T$ values. Overall, we observe that performance and energy optimizations can lead to very different choices of system configurations. The overall trends presented in this chapter are not very different even when we go for different dataset sizes.

Benchmark	**RAID Level**	**Best T**	**Best E**	**Best E × T**
TPC-C	**RAID-4**	32/4 KB	24/256 KB	24/4 KB
	RAID-5	32/4 KB	24/256 KB	24/4 KB
	RAID-10	32/4 KB	30/256 KB	30/256 KB
TPC-H	**RAID-4**	38/8 KB	24/256 KB	38/8 KB
	RAID-5	38/32 KB	24/256 KB	32/32 KB
	RAID-10	38/8 KB	30/256 KB	30/64 KB

Table 2.4. Optimal Configurations for the Workloads. For each configuration, the pair of values indicated give the number of disks used and the stripe-size employed.

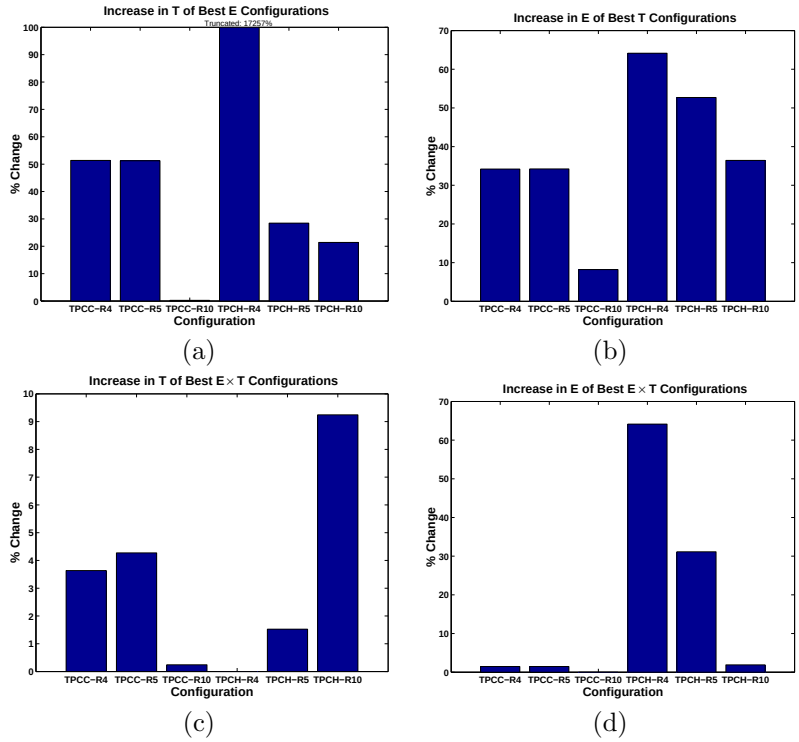


Fig. 2.11. The Effect of Tuning with different Performance and Energy Criteria

Chapter 3

Thermal Issues in Disk Drive Design

3.1 Introduction

Disk-drives lie at the heart of the storage-system and are the most significant determinant of the utility of the system, both from the performance and capacity viewpoints. Due to the large difference in speed between disks and the other levels in the memory hierarchy (main-memory, processor cache etc.), I/O performance plays a critical role for many server applications. There have been several improvements over the years to address the I/O bottleneck, including better caching/buffer management [82], parallelism in the form of RAID [81], and high bandwidth interconnects such as SAN. However, at the core of these extensive I/O subsystems lie the disk drives, whose performance advances have woefully lagged behind the rest of the system components. As Amdahl's law dictates, a single such laggard can eventually limit overall system performance. Further, it is quite possible that a faster drive can actually alleviate the need for going to expensive storage area networks and higher levels of parallelism when deploying balanced systems.

From a drive design perspective, two important performance metrics are the Internal Data Rate (IDR) and the seek-time. Over the past fifteen years, disk-drive manufacturers have been able to release products along a 40% IDR annual growth-rate curve. This growth has been provided by innovations in recording technology that have provided improved densities coupled with increases in the drive RPM. In order to compensate for the higher power dissipation due to the faster rotational speed of the platters and ensure that the device operates within the thermal design envelope, the platter-sizes have also been shrunk in the successive drive generations. This

process of designing to operate within the thermal envelope is critical because temperature is one of the most fundamental factors affecting the reliability of a disk drive. High temperatures can cause off-track errors due to thermal tilt of the disk stack and actuator, or even cause head crashes due to the out-gassing of spindle and voice-coil motor lubricants [44]. Disks are so sensitive to temperature that even a fifteen degree Celsius rise from the ambient temperature can double the failure rate of a disk drive [4].

While these techniques have been successful in providing us to scale the IDR of disk drives along the 40% growth-rate curve for nearly the past two decades, there are several impediments on the horizon:

- The growth rates of the linear and track densities are expected to slow down due to a variety of physical limitations. Further, lower Signal-to-Noise ratios at high areal densities require the use of stronger error correcting codes, which reduce the user-accessible capacity and user-perceived data-rate. These trends mandate more aggressive scaling of the drive RPM to meet the IDR targets, which also increases the amount of heat generated (by nearly the cubic power of the RPM).

- Going to smaller form-factors decreases the amount of heat that can be dissipated to the outside air.

- In high-density server configurations, which are typical in most machine-rooms and data centers today, the external ambient temperatures are becoming more difficult to contain to the pre-heating of the air by other components near the drives and provisioning more powerful cooling systems to mitigate it is very expensive (as was shown in Chapter 1).

In order to continue to make innovations, it is important to understand all these trade-offs and how they impact the disk drive roadmap over the next decade. Understanding and studying all these issues mandates comprehensive models of disk drives. Most of the previous published

work has been mainly on performance models [29], with a few isolated studies on modeling temperature within a drive [23].

The related work is given in Section 3.2. Detailed and inter-related models for the capacity and performance characteristics are developed and validated for real disks (Section 3.3). In Section 3.3.3, a detailed model of the thermal characteristics of disk-drives is presented along with details of how this model is adapted and validated for modern disks. Section 3.4 presents the expected trends in the magnetic recording technology over the next decade and develops a methodology to generate the disk drive roadmap. Section 3.4.1 charts out the roadmap when designing disks for operation within the thermal envelope. Section 3.4.2 studies the impact of changes in other related technologies on the roadmap.

3.2 Related Work

The research presented in this chapter spans cross-cutting topics in storage systems including drive physical and behavioral modeling, and high-density characteristics.

The importance of the I/O subsystem on the performance of server applications has resulted in fairly detailed performance models of the storage hierarchy (e.g. [30, 32]). The Disksim simulator [30] is one such publicly distributed tool that models the performance characteristics of the disk drive, controllers, caches and interconnects. The importance of modeling power has gained attention over the last decade, primarily in the context of conserving battery energy for drives in laptops [116, 49]. A tool called Dempsey is presented in [116] for detailed modeling of the energy consumption of mobile hard disks. A detailed breakdown of the power in the different components of a mobile disk is given in [40] and [49]. Temperature-aware design is becoming increasingly important [99] as well. In the context of disk drives, [16] describes a model of the thermal behavior of a drive based on parameters such as the dimensions, number of platters in the disk stack, their size, and properties of the constituent materials. This model is adapted to

study the thermal ramifications of current and future hard disk drives. There has also been some recent work on modeling and designing disk arrays in a temperature-aware manner [52].

There are also published papers on analyzing the dynamics of disk drives via macro-modeling using mixed-signal Hardware Description Languages (HDLs) such as VHDL-AMS [15, 20]. [111] presents a technique known as physical-effect modeling wherein the different components of a mechanical device are modeled separately in a HDL and are combined to model the entire device. [19] uses this technique to model the SPM and VCM of a hard-disk.

The historical evolution of different aspects of hard-disk drive design along with projections for their future trends are given in a set of papers published by industry [4, 34, 46, 92]. There have also been studies on the characteristics of designing future high density disk-drives. These papers cover issues such as the impact of bit-aspect ratios [13] and error-correcting code overheads [112] in such drives. There are several proposals [112, 72, 102] on how to build Terabit areal density drives, covering magnetic recording physics issues and also engineering considerations.

3.3 Modeling the Capacity, Performance and Thermal Characteristics of Disk drives

This section describes the models used in this study for capturing capacity, data rates and thermal characteristics of disk drives.

3.3.1 Modeling the Capacity

The model begins with an abstraction of the fundamental properties of recording technologies via two quantities, namely, the linear bit-density given in Bits-per-Inch (BPI) along a track, and the radial track-density, which is expressed in Tracks-per-Inch (TPI) (shown in Figure 3.1). BPI improvements are a result of technological advances in read/write head design and recording medium materials. TPI is improved by advances in the servo design, track misregistration reduction techniques, and more sophisticated heads [5]. The product of BPI and TPI

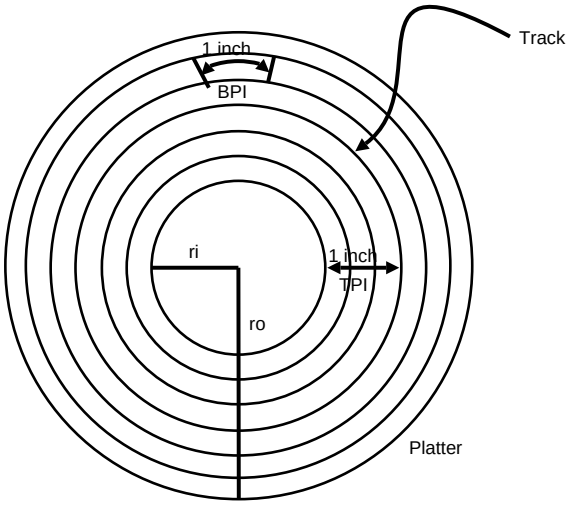


Fig. 3.1. Visual illustration of BPI and TPI. r_o and r_i denote the outer and inner platter-radii respectively.

is known as the *areal density* and is one of the most fundamental determinants of both drive speed and capacity. The ratio $\frac{BPI}{TPI}$ is known as the bit aspect-ratio (BAR) and will be used later in this study to set up the technology scaling predictive model. Another metric of interest to disk-drive designers is the Internal Data Rate (IDR), which is expressed in MB/s. The IDR is the actual speed at which data can be read from or written into the physical media. The IDR is affected by the BPI, platter size, and disk RPM.

Let us assume that we know the outer radius, r_o, of the disk drive. The inner radius is set to be half that of the outer radius,i.e., $r_i = \frac{r_o}{2}$. Although this rule of thumb was common in the past, modern disks may not necessarily follow this rule [5]. As the exact inner radius tends to be manufacturer specific and even varies across a single manufacturer's products, this rule is still used in the modeling.

Let n_{surf} denote the number of surfaces in the drive - this value is twice the number of platters. Then, the number of tracks on the disk surface, which is also denoted as the number of *cylinders* in the disk, n_{cylin}, is given by $n_{cylin} = \eta(r_o - r_i)TPI$, where η is the stroke efficiency,

which measures the fraction of the total platter surface that is user accessible. If $\eta = 1.0$, then the equation gives the number of tracks that can be laid out in the area between the innermost to the outermost edge of the platter. However, in practice, η is much lesser than 1 since portions of this real estate are dedicated for recalibration tracks, manufacturer reserved tracks, spare tracks (to recover from defects), landing zone for the head slider, and other manufacturing tolerances. The stroke is typically around $\frac{2}{3}$ [53], which is the value that is used in the models. From these, we can calculate the raw capacity (in bits), C_{max}, of the disk drive as

$$C_{max} = \eta \times n_{surf} \times \pi(r_o^2 - r_i^2)(BPI \times TPI)$$

In reality, even this C_{max} is not completely usable because:

1. Outer tracks can hold more sectors because of their longer perimeters. However, allocating storage on a per-track basis would require complex channel electronics to accommodate different data rates for each track [5]. Instead, *Zoned Bit Recording (ZBR) or Multi-Band Recording* is used, which can lead to some capacity loss.

2. In addition to user data, each sector needs additional storage for servo patterns and Error Correcting Codes (ECC) leading to a further reduction in capacity. These components are modeled as follows.

3.3.1.1 Capacity Adjustment due to Zoned Bit Recording (ZBR):

ZBR is a coarse-grained way of accommodating variable sized tracks, where the tracks are grouped into zones, with each track in a zone having the same number of sectors. Such grouping can provide good trade-offs between capacity and complexity of the electronics. ZBR can allow more data to reside on outer tracks to benefit from higher data rate due to a constant angular velocity, without extensively complicating the electronics.

Each track, j, has a raw bit capacity C_{t_j}, which is given by $C_{t_j} = 2\pi r_j BPI$, where r_j is the radius of track j. Let $j = 0, 1, ..., n_{cylin} - 1$, where 0 is the outermost track and $n_{cylin} - 1$ is the innermost. Then, for any two tracks m and n such that $m < n$, $C_{t_m} > C_{t_n}$ since $r_m > r_n$. Since the recordable area is within $r_o - r_i$, and we have n_{cylin} cylinders that are assumed to be uniformly spaced out, the perimeter of any given track j, denoted as P_{t_j} is given by

$$P_{t_j} = 2\pi[r_i + (\frac{r_o - r_i}{n_{cylin} - 1})(n_{cylin} - j - 1)] \tag{3.1}$$

This equation is easy to understand by considering the three cases where a track may be located on a surface:

1. If $j = 0$, then the corresponding track is that which is at the distance of r_o from the center of the platter. Thus, the perimeter of the track is $2\pi r_o$.

2. If $j = n_{cylin} - 1$, then the corresponding track is that which is closest to the spindle-motor assembly, at a distance of r_i from the center of the platter. Thus, the perimeter is $2\pi r_i$.

3. For the intermediate tracks, the distance between adjacent tracks is $(\frac{r_o - r_i}{n_{cylin}})$, i.e., the tracks are equally spaced out radially along the platter surface. Therefore, any given track j is located at a distance $(\frac{r_o - r_i}{n_{cylin}})(n_{cylin} - j)$ from the innermost track.

Assuming that each zone has an equal number of tracks, the number of tracks per zone, n_{tz}, is $n_{tz} = \frac{n_{cylin}}{n_{zones}}$, where n_{zones} is the desired number of zones, which is around 30 for modern disk-drives. Therefore, zone 0 would be composed of tracks 0 to $\frac{n_{cylin}}{n_{zones}} - 1$, zone 1 would have tracks $\frac{n_{cylin}}{n_{zones}}$ to $(\frac{2n_{cylin}}{n_{zones}} - 1)$ and so on. For each zone z, let the bit capacity of its smallest perimeter track in the zone be denoted as $C_{tz_{min}}$. In our ZBR model, we allocate, for every track in zone z, $C_{tz_{min}}$ bits (or $(\frac{C_{tz_{min}}}{8 \times 512})$ sectors). Thus each zone has a capacity of $n_{tz}(\frac{C_{tz_{min}}}{4096})$ sectors, making the total disk capacity (in 512-byte sectors), with losses due to ZBR taken into account, as

$$C_{ZBR} = n_{surf} \sum_{z=0}^{n_{zones}-1} n_{tz}(\frac{C_{tz_{min}}}{4096})$$

3.3.1.2 Capacity Adjustments due to Servo Information:

Servo are special patterns that are recorded on the platter surface to correctly position the head above the center of a track. In older drives, an entire surface (and head) used to be dedicated for servo information, leading to considerable loss of usable capacity. To mitigate this, modern drives make use of a technique known as *embedded servo*, where the servo patterns are stored along with each sector. There are no special servo surfaces and the read/write heads that are used for user data access are also used to read the servo information.

The storage-overheads for servo are modeled by considering the number of bits required to encode the track-identifier information for each servo-sector. Other fields in the servo information such as those for write-recovery (which signals the beginning of a servo pattern) and for generating the Position Error Signal (which indicates the position of the actuator with respect to a particular track) are not modeled due to the lack of information about their implementation in real disk drives. The servo model that we use is based on the information given in the patent [77]. The track-id information is encoded as a Gray Code, such that the fields for any two adjacent tracks differ only by a single bit. This enables fast and accurate seeks to be performed. Thus, the number of bits needed in the code to encode a track on a surface is $log_2(n_{cylin})$. As the servo information is embedded with each sector, the total number of bits used for storing servo in each sector, C_{servo} is given by

$$C_{servo} = \lceil log_2(n_{cylin}) \rceil \tag{3.2}$$

3.3.1.3 Capacity Adjustments due to Error-Correcting Codes:

One of the ramifications of technology scaling has been increased error rates. In order to understand why, a brief exposition of how higher densities are achieved is required.

Each bit cell in a track is composed of multiple magnetic grains (typically 50-100 grains/cell [47]). A bit storing a digital 'one' is composed of a region of grains that are uniformly polarized, and a region where there is a transition in the magnetic polarity represents a 'zero'. This is visually depicted in Figure 3.2.

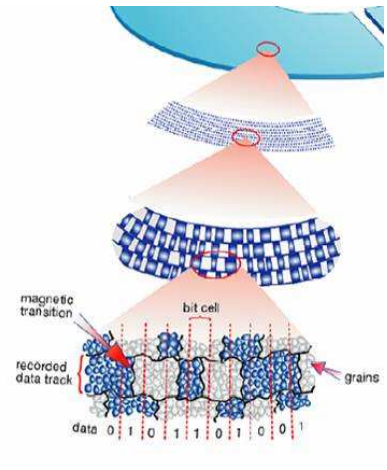


Fig. 3.2. A Magnetic Bit. A digital '0' is composed of a region of uniform polarity and a '1' by a boundary between regions of opposite magnetization. Image Source: Hitachi Global Storage Technologies (http://www.hitachigst.com/hdd/research/storage/pm/index.html)

When a write is performed on a bit cell, all the grains in the region have their magnetic polarity altered by the write head. To achieve higher areal density, the size of the bit cell needs to be reduced. In order to achieve this, we have two opportunities, namely, shrinking the grains themselves and reducing the number of grains in the cells. The standard approach has been to shrink the grain size. However, the superparamagnetic limit [10] imposes a minimum grain size

so that the signal energy stored in the grain does not drop below the ambient thermal energy. Otherwise, the magnetic grains would become thermally unstable and would flip their polarity within a short time span (effectively rendering disk drives as volatile storage!). One way to overcome this limit is to use a recording medium that is more coercive, and thus requiring a stronger field to change the state of the bits. Designing write heads to achieve this is quite challenging [53]. Therefore, in order to continue achieving areal density growth beyond this point, it would be necessary to reduce the number of grains per bit as well. The use of fewer grains in the bit cell leads to lower Signal-to-Noise Ratios (SNR). In order to accommodate such noisy conditions, Error-Correcting Codes (ECC) are required, and most modern disks use Reed-Solomon codes [91]. It has been shown that, for current disks, the ECC storage requirement is about 10% of the available capacity and would increase to 35% for disks whose areal densities are in the Terabit range [112]. Thus, in our model, the total capacity used by ECC (in bits), C_{ECC}, is 416 bits/sector for drives whose areal densities are less than 1 Tb/in^2, whereas those in the terabit range use 1440 bits/sector.

3.3.1.4 Derated Capacity Equation:

From the above discussions on ZBR, Servo and ECC costs, we can calculate their total space overhead (in bits/sector) as

$$\alpha = n_{tz}(\frac{C_{tz_{min}}}{4096})(C_{servo} + C_{ECC})$$

Therefore, the estimated capacity of the disk in terms of 512 byte sectors is given by

$$C_{actual} = n_{surf} \sum_{z=0}^{n_{zones}-1} n_{tz}(\frac{n_{tz}C_{tz_{min}} - \alpha}{4096}) \quad (3.3)$$

3.3.1.5 Validation:

To verify the accuracy of this derived model, the capacity estimated by the models are compared against actual values reported for a set of server disks of different configurations, manufacturers, and from different years (obtained from [4]). The result of this comparison is presented in Table 3.1. For most disks, the difference between the actual and estimated capacities is within 12%. The errors are primarily due to some of the assumptions made along the way, and also because of our assumption of 30 zones for each disk (which is optimistic for many of the older disks in the table which used only around 10-15 zones).

Model	**Cap. (GB)**	**Model Cap. (GB)**	**IDR (MB/s)**	**Model IDR (MB/s)**
Quantum Atlas 10K	18	17.6	39.3	46.5
IBM Ultrastar 36LZX	36	30.8	56.5	58.1
Seagate Cheetah X15	18	20.1	63.5	73.6
Quantum Atlas 10K II	18	12.8	59.8	61.9
IBM Ultrastar 36Z15	36	35.2	80.9	72.1
IBM Ultrastar 73LZX	36	34.7	86.3	85.2
Seagate Barracuda 180	180	203.5	63.5	71.8
Fujitsu AL-7LX	36	37.2	91.8	100.3
Seagate Cheetah X15-36LP	36	40.1	88.6	103.4
Seagate Cheetah 73LP	73	65.1	83.9	88.1
Fujitsu AL-7LE	73	67.6	84.1	88.1
Seagate Cheetah 10K.6	146	128.8	105.1	103.5
Seagate Cheetah 15K.3	73	74.8	111.4	114.4

Table 3.1. SCSI disk drives of different configurations from various manufacturers and year of introduction into the market. The capacities and IDR given by our model are compared against the corresponding values in the datasheets. It is assumed that $n_{zones} = 30$ for all the configurations. The detailed drive specifications are given in [37].

3.3.2 Modeling the Performance

Two main performance related drive parameters are the *seek time* and the *internal data rate*, whose modeling is presented below.

3.3.2.1 Seek Time:

The seek time depends on two factors, namely, the inertial power of the actuator voice-coil motor and the radial length of the data band on the platter [34]. Physically, a seek involves an acceleration-phase, when the VCM is turned on, followed by a coast-phase of constant velocity, and then a deceleration to stop the arms near the desired track. This is then followed by a head-settling period. For very short seeks, the settle-time dominates the overall seek time whereas for slightly longer seeks, the acceleration and deceleration phases dominate. Coasting is more significant for long seeks.

The model that is used is based on the one proposed by Worthington et al. [113], which uses three parameters, namely, the track-to-track, full-stroke, and average seek time values, which are usually specified in manufacturer datasheets. The track-to-track seek time is the time taken for the actuator to move to an adjacent track. The full-stroke seek time is the time taken for the actuator to move from one end of the data band to another. It has been observed that, except for very short seeks (less than 10 cylinders), a linear interpolation based on the above three parameters can accurately capture the seek time for a given seek distance [113], as the coast-time tends to be linear with the distance traversed. To determine these values for hard disk drives of the future that we will be evaluating later, a linear interpolation of data from actual devices of different platter sizes was used.

3.3.2.2 Calculating Internal Data Rate (IDR):

The maximum IDR would be experienced by tracks in the outermost zone (zone 0) of the disk drive, since there are more bits stored there while the angular velocity is the same across the tracks. Consequently, we can express the maximum IDR (in MB/sec) of the disk as:

$$IDR = (\frac{rpm}{60})(\frac{n_{tz0} \times 512}{1024 \times 1024}) \quad (3.4)$$

where n_{tz0} is the number of sectors/track in zone 0, and *rpm* is the angular velocity expressed as rotations-per-minute.

3.3.2.3 Validation:

The seek time models have already been verified in earlier work [113]. To validate the IDR model, its value is computed from the specifications for the disks listed in Table 3.1 using our models and compared against the manufacturer supplied IDR value. The resulting data is presented in the last two columns of Table 3.1. Again, it is assumed that each of the disks uses ZBR with 30 zones. From the table, it can be observed that for most of the disks, the IDR predicted by our model and the actual IDR are within 15%.

3.3.3 Modeling the Thermal Behavior

The thermal model that used in this study is based on the one developed by Eibeck et al [23]. This model evaluates the temperature distribution of the drive by calculating the amount of heat generated by components such as the SPM and the VCM, the conduction of heat along the solid components and the convection of heat to the air. It is assumed that the drive is completely enclosed and the only interaction with the external air is by the conduction of heat through the base and the cover and convection with the outside air. The outside air is assumed to be maintained at a constant temperature by some cooling mechanism. This is true in most modern systems where air flow is provided, typically using fans, to maintain a constant external temperature [98].

The model divides the hard disk into four components:

1. The internal drive air.
2. The SPM assembly that consists of the motor hub and the platters.
3. The base and cover.

4. The VCM and the disk arms.

The heat transfer rate over a time interval t, $\frac{dQ}{dt}$ (in Watts), through a cross-sectional area A is given by Newton's Law of Cooling as

$$\frac{dQ}{dt} = hA\Delta T$$

where h is the heat-transfer coefficient and ΔT is the temperature difference between the two entities. For solids, where heat is transferred via conduction, the heat transfer coefficient h depends upon the thermal conductivity k and the thickness of the material and is given by $\frac{k}{Thickness}$.

Between solids and fluids, the heat exchange takes place via convection, where the heat transfer coefficient depends on whether the fluid flow is laminar or turbulent, and also on the exact geometry of the solid components. The model makes use of empirical correlations for known geometries to calculate the heat transfer coefficient of the different solid components of the disk drive. The heat of the internal drive air is calculated as the sum of the heat energy convected to it by each of the solid components and the viscous dissipation (internal friction) in the air itself minus the heat that is lost through the cover to the outside. The viscous dissipation is related linearly to the number of platters in the disk stack, cubic (2.8-th power to be precise) with the disk RPM and to the fifth (4.6-th power to be precise) power of the platter diameter [16, 97].

To solve the heat equations for each component, the model uses the finite difference method [65]. At each time step, the temperatures of all the components and the air are calculated, and this is iteratively revised at each subsequent time step until it converges to a steady state temperature. The air temperature is assumed to be uniform over the entire drive at each time step. The accuracy of the model depends upon the granularity of the time steps [16]. Using a coarse-grained time step provides a faster model (in terms of computation time), but the solution may not be accurate. On the other hand, an extremely fine-grained time step can provide an

accurate solution at the expense of a high computation time. A wide range of different sizes were experimented with and it was found that a value of 600 steps per minute gives a solution very close (numerically) to that of the finer-grained ones.

There are a number of input parameters to the thermal model. The first set of parameters relate to the disk geometry, such as the inner and outer radii of a platter, the enclosure dimensions, and the length of the disk arms. Another set of parameters pertain to the properties of the materials, such as the thermal conductivity and density. There are also operational parameters such as the number of platters, the RPM, the temperature of the outside air and the VCM power.

With regard to the materials, the platters on most current disk drives are typically made of an Aluminum-Magnesium alloy and the base/cover castings are Aluminum [45]. As the exact alloy that is employed tends to be proprietary information, it was assumed that the platters, together with the disk arm and spindle hub, are made of Aluminum. With regard to the operational parameters, the external ambient temperature was set to 28 C, which is the maximum operating wet-bulb temperature. The wet-bulb temperature measures the temperature with the humidity of the air taken into account. Many disks, including some of the thirteen that examined in the earlier two subsections (see Table 3.2), specify a maximum wet-bulb external temperature of 28-29.4 C. When calculating the power of the VCM, which is dependent on platter dimensions, previously published data was used [100]. This earlier work shows that the VCM power is roughly twice for a 95 mm (3.7") platter compared to that for a 65 mm (2.5") one, and nearly four times that for the 47 mm (1.8") size.

3.3.3.1 Validation and Setting a Thermal Envelope

The thermal model proposed earlier [23] was validated with disk drives that are over 15 years old. In addition to making sure that the model is still applicable for modern drives, we also need to define what should be the thermal envelope (i.e. the maximum operating temperature) for drive design when charting out the roadmap.

The Seagate Cheetah 15K.3 ST318453 SCSI disk drive [98] was modeled in detail. This disk-drive is composed of a single 2.6" platter (but a 3.5" form-factor enclosure) and rotates at 15K RPM (a 4-platter version of this disk is listed in the last row of Table 3.1). The disk drive was taken apart and its geometry studied in detail. This allows me to determine how the components are internally laid out and create geometry models parameterized for the platter-size and count. The physical drive parameters such as the length of the disk-arm, thickness of the platter, base, and cover etc., which are not considered by the capacity and performance models, were measured precisely using Vernier calipers. The VCM power of this disk is determined to be 3.9 W. The disk specifications (in their data sheets) typically include the maximum operating temperature which is the temperature that should not be exceeded even if the VCM and SPM are always on. In the validation experiment, the SPM and VCM are assumed to be always on, and the internal air temperature is calculated.

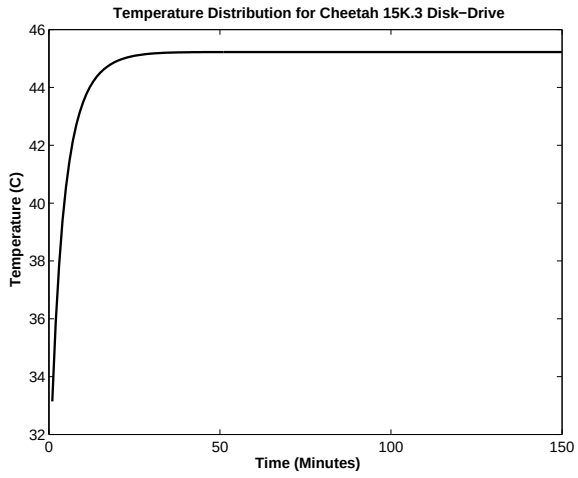


Fig. 3.3. Temperature of the modeled Cheetah ST318453 disk over time starting from an external temperature of 28 C.

The temperature of the internal air over the duration of the experiment is shown in Figure 3.3. All components are initially at the temperature of the outside air, namely, 28 C. The temperature rises from 28 C to 33 C within the first minute and rapidly increases thereafter. It then stabilizes and reaches a steady state of 45.22 C after about 48 minutes. Only the VCM and SPM are considered to be heat sources, and do not consider the heat generated by the on-board electronics. This is due to two factors, namely, the lack of publicly available data on the microarchitecture of the electronics, in addition to details of the technology roadmaps for ASICs. Consequently, the heat generated by these electronic components is discounted in all the results and the thermal envelope of operation is reduced accordingly. In fact, earlier research [52] has shown that on-board electronics can add about 10 C to the temperature within the drive. If we consider this additive factor (10 + 45.22 = 55.22 C), the results presented here come very close to the rated maximum operating temperature of this drive (which is 55 C), verifying the validity of this model.

Model	**Year**	**RPM**	**External Wet-bulb Temp.**	**Max. Oper. Temp.**
IBM Ultrastar 36LZX	1999	10K	29.4 C	50 C
Seagate Cheetah X15	2000	15K	28.0 C	55 C
IBM Ultrastar 36Z15	2001	15K	29.4 C	55 C
Seagate Barracuda 180	2001	7.2K	28.0 C	50 C

Table 3.2. Maximum operating temperatures for a specified external wet-bulb temperature.

It is to be noted that the thermal envelope - the maximum temperature within a drive for reliable operation - itself has negligible variance over time. This is reflected in the rated maximum operating temperatures of some of the disks in the list spanning different years (1999-2001) and different RPMs, procured from datasheets (see Table 3.2), which remains more or less constant. Consequently, I use the same *thermal envelope of 45.22 C* (obtained above without on-board

electronics) when laying out the roadmap over time across disks of different platter sizes and numbers.

3.4 Roadmap with Thermal Constraints

In the previous section, three drive models for capacity, performance and thermal characteristics were presented, which though explained independently are rather closely intertwined. This is because many of the parameters used by each model can depend on the results from another. For instance, the performance and thermal characteristics are closely dependent on the drive parameters (e.g. size and number of platters). The heat dissipation is closely dependent on the operating RPM. Finally, the capacity of drives is not only limited by recording technologies, but also by the thermal envelope (larger or more platters can lead to higher temperatures). It is important to ensure that we study all the inter-related factors together when charting out a disk drive technology roadmap.

This roadmap is driven by two fundamental factors: (i) the innovations in magnetic technologies to increase recording densities (the BPI and TPI in particular), and (ii) the growing demands for high data transfer rates (the IDR). The trends in growth of BPI, TPI and IDR over the past decade have been made available by Hitachi [46] where the values for each year, together with their Compound (annual) Growth Rate (*CGR*), are given. For instance, for the year 1999, the values for BPI, TPI, and IDR were 270 KBPI, 20 KTPI, and 47 MB/s, and their CGRs have been 30%, 50%, and 40% respectively. This growth rate in the linear and track densities has resulted in an areal density CGR of 100%. These past/projected growth rates are the starting points for the roadmap. Even though we have benefited from these growth rates over the past decade, it is going to be challenging to maintain these growth trends in the future:

- The growth in BPI is expected to slow down due to several factors. First, increases in linear density would require lower head fly heights. With current head fly heights already being

only a few nanometers from the platter surface, it is very difficult to reduce this further. Second, increasing the BPI requires higher recording medium coercivity, for which, as we mentioned in Section 3.3.1, it is not feasible to design a write head with currently known materials. Finally, the grain size is constrained by the superparamagnetic limit.

- The CGR for TPI is also expected to decline [13]. Increasing the TPI requires that tracks be narrower, which makes them more susceptible to media noise. Further, more closely spaced tracks can lead to inter-track interference effects. Finally, the track edges are noisier than the center region and the edge effects increase with narrower tracks.

As the BARs have also been dropping, there has been a larger slowdown in the BPI CGR than that for the TPI. It has been shown [13] that there exist optimal values for the BAR for a given areal density. The BAR is around 6-7 for disks today and is expected to drop to 4 or below in the future [46]. Furthermore, it is expected [33, 92] that the growth in areal density would slow down to 40-50%. Given this growth in areal density, the industry projections predict the availability of an areal density of 1 Tb/in^2 in the year 2010.

We studied a set of proposals for creating such a terabit density disk [112, 72, 102]. In particular, we are interested in the feasible values for BPI and TPI, given all the constraints related to the recording medium, head design, and noise margins, for constructing reliable terabit density disks. Among the proposals, we chose the one with more conservative assumptions about the BPI, since it does not scale as well as TPI, to obtain values of 1.85 MBPI and 540 KTPI giving a BAR of 3.42 (which agrees with current expectations). We then adjusted the CGRs for the BPI and TPI to achieve this areal density in the year 2010, together with the expected BAR trends. This provides a BPI and TPI CGR of 14% and 28% respectively (down from the original values of 30% and 50%), to give an areal density CGR of about 46% per year.

Once we have these fundamental parameters (the anticipated BPI and TPI for each year), a "roadmap" is then generated, starting from the year 2002, for a period of ten successive years,

i.e., upto the year 2012. The basic premise when doing this is that we are trying to sustain the expected IDR growth rate of *at least* 40% per year over the 11 year period. The steps when generating the temperature-dictated disk drive technology roadmap are given below:

1. For each year, the values for BPI and TPI are plugged in from the above estimates into the capacity model calculated in section 3.3.1 for a given platter size and number of platters - carried over from the previous year. For the resulting disk configuration, we can calculate its IDR for a given RPM (which is again carried over from the previous year), by putting in the appropriate values for n_{tz0} in equation 3.4. If the resulting IDR meets the projected 40% growth rate, then the new configuration would remain within the tolerable thermal envelope (since the same platter size, number of platters and RPM yielded a disk within the thermal envelope in the previous year).

2. However, if the disk from step 1 does not meet the target IDR for that year, one option is to see whether increasing the RPM can get us to this IDR (by putting this value in the LHS of equation 3.4 and finding the rpm). We can then use the resulting disk configuration and RPM value in the thermal model of the disk in section 3.3.3 to see whether this remains within the thermal envelope. If it does, then we have achieved the target IDR using the same number of platters and platter sizes as the previous year by increasing the RPM.

3. If the necessary RPM from step 2 does not keep the new disk within the thermal envelope, then the other option for still meeting the IDR target is to shrink the platter sizes. Recall that the viscous dissipation is proportional to the fifth power of the platter size, and the third power of the RPM. Further, a smaller platter size implies shorter seeks, thus reducing VCM power as well. Consequently, it is possible to remain within the thermal envelope by shrinking the platter size (note that the resulting n_{tz0} in equation 3.4 decreases) and increasing the RPM proportionally to compensate for the drop in IDR.

4. Shrinking the platter size as in step 3 results in a drop in the overall capacity. Over a period of time, such reductions in capacity can cumulate, causing a concern. To compensate for this reduction, it may become necessary to add platters at some point, causing all the steps enumerated above to be repeated.

Thus, the roadmap is not a single disk drive design point but is a spectrum of different platter sizes (and their corresponding RPMs) that try to sustain the IDR growth rate from year to year. When generating this roadmap, we consider the initial platter size (in the year 2002) to be 2.6", with two subsequent shrinks of 2.1" and 1.6" for later years. Smaller platter sizes are not considered due to the unavailability of VCM power correlations, and disk enclosure design considerations at such small media sizes. For each platter size in a given year, configurations with 1, 2, and 4 platters are considered. These represent disks for the low, medium, and high capacity market segments for the same technology generation. Increasing the number of platters also increases the viscous dissipation. This is taken into account and different external cooling budgets are provided for each of the three platter counts in order to use the same thermal envelope (45.22 C) for these higher platter disks at the beginning of the roadmap. We assume that the cooling technology remains invariant over time, and the disks need to be designed for the thermal envelope solely based on internal choices. The ramifications of changes in the cooling system are studied in Section 3.4.2.1.

3.4.1 Results

As a first step in the analysis, it is interesting to investigate what would be the disk speed required for a given platter size and its resultant thermal impact, when trying to meet the 40% IDR growth target. In the absence of any thermal constraints, if we are to meet the IDR target for a given year, we would use the largest platter size possible and merely modulate the RPM to reach the desired value (step 2 of the method). Tables 3.3, 3.4, and 3.5 give the RPM that is required in each year for the three platter sizes that we consider and the steady

state temperature that is reached for a one platter configuration. Trends for 2 and 4 platter configurations are similar.

Year	**IDR**$_{density}$	**RPM**	**Temperature (C)**	**IDR**$_{Required}$
2002	128.14	15098	45.24	128.97
2003	166.53	16263	45.47	180.56
2004	189.85	19972	46.46	252.78
2005	216.37	24534	48.26	353.89
2006	246.66	30130	51.48	495.44
2007	281.19	37001	57.18	693.62
2008	320.47	45452	67.27	971.07
2009	365.34	55819	85.04	1359.5
2010	300.23	95094	223.01	1903.3
2011	342.13	116826	360.40	2664.61
2012	390.03	143470	602.98	3730.46

Table 3.3. The thermal profile of the RPM required to meet the IDR CGR of 40% for the 2.6" platter-size. We assume a single-platter disk with $n_{zones} = 50$ and a 3.5" form-factor enclosure. The thermal envelope is 45.22 C.

Let us analyze these results by first focusing on the 2.6" platter size. The IDR requirements (shown as IDR$_{Required}$ in Table 3.3), from the year 2002 to 2012, increase nearly 29 times. A portion of the required increase is provided by the growth in the linear density, denoted in the Table as IDR$_{density}$ (i.e. the IDR obtainable with just the density growth without any RPM changes). Any demands beyond that has to be provided by an increase in the RPM. For instance, the RPM requirements grow nearly 9.5 times from the year 2002 to 2012. For a better understanding, let us sub-divide the timeline into three regions, namely, the years before 2004, where the BPI and TPI CGRs are 30% and 50% respectively, the years from 2004 to 2009, which are in the sub-terabit areal densities, and the region from 2010 to 2012. Recall that the growth rates in BPI and TPI slow down after 2003 to 14% and 28% respectively and the ECC overheads for terabit areal density disks would increase to 35%. The effects of these trends are shown in the Table, where there is only a 7.7% increase in the required RPM from 2002 to 2003, but the

Year	**IDR**$_{density}$	**RPM**	**Temperature (C)**	**IDR**$_{Required}$
2002	103.50	18692	43.56	128.97
2003	134.51	20135	43.69	180.56
2004	153.34	24728	44.37	252.78
2005	174.81	30367	45.61	353.89
2006	199.23	37303	47.85	495.44
2007	227.12	45811	51.81	693.62
2008	258.91	56259	58.81	971.07
2009	295.08	69109	71.17	1359.5
2010	242.49	117735	167.01	1903.3
2011	276.44	144586	262.19	2664.61
2012	315.02	177629	430.93	3730.46

Table 3.4. The thermal profile of the RPM required to meet the IDR CGR of 40% for the 2.1" platter-size. We assume a single-platter disk with $n_{zones} = 50$ and a 3.5" form-factor enclosure. The thermal envelope is 45.22 C.

Year	**IDR**$_{density}$	**RPM**	**Temperature (C)**	**IDR**$_{Required}$
2002	78.86	24533	41.64	128.97
2003	102.51	26420	41.74	180.56
2004	116.83	32455	42.15	252.78
2005	133.19	39857	42.93	353.89
2006	151.83	48947	44.29	495.44
2007	173.04	60127	46.73	693.62
2008	197.27	73840	51.04	971.07
2009	224.88	90680	58.63	1359.5
2010	184.75	154527	117.61	1903.3
2011	210.62	189769	176.20	2664.61
2012	240.11	233050	279.75	3730.46

Table 3.5. The thermal profile of the RPM required to meet the IDR CGR of 40% for the 1.6" platter-size. We assume a single-platter disk with $n_{zones} = 50$ and a 3.5" form-factor enclosure. The thermal envelope is 45.22 C.

required RPM growth increases to about 23% per-annum in the post-2003 region. During the terabit transition (from 2009 to 2010), a sudden 70% increase in RPM is required. This happens because of the way the impact of ECC is modeled, as a sudden increase from 10% to 35% when transitioning into the terabit region. Realistically, this transition would be more gradual. After this steep increase, the RPM growth rate again steadies out to 23% for the subsequent years.

When examining the thermal characteristics of the 2.6" drive, a similar trend is observable for the three temporal regions of the roadmap. The heat due to viscous dissipation increases from 0.91 W in 2002 to 1.13 W in 2003. In the second region, due to the higher rotational speed growth (and its relationship in the cubic power), the viscous dissipation grows from 2 W in 2004 to over 35.55 W in 2009, causing a significant rise in temperature, well beyond the thermal envelope of 45.22 C. Therefore, all other things remaining constant, it is clear that future single platter disk drives would not be able to provide the desired IDRs at the 2.6" platter size. The viscous dissipation increases even further from year 2010 onwards and reaches a value of 499.73 W in 2012, causing the internal drive air temperature to reach as high as 602.98 C for this platter size.

The effect of shrinking the platter (related in the fifth power to heat, which can compensate for certain amounts of increases in RPM) can be observed by examining the results for the 2.1" and 1.6" drives in Tables 3.4 and 3.5. Even though a smaller platter size implies a higher RPM is needed to meet the required IDR (than for the 2.6" drive), it can be seen that the higher RPMs can be somewhat offset by moving to the smaller sizes, helping us stay within the thermal envelope until around 2007. Beyond that, even the 1.6" size is too big to stay within the envelope.

Having seen that RPM increase is not always a viable option in drive design to achieve the target IDR, let us now analyze the impact of the thermal envelope in meeting the IDR requirements and the resulting capacity. Figure 3.4 shows the maximum achievable data rates (and the corresponding capacities) for the spectrum of disk designs where the points are all within the thermal envelope. For each experiment (with a given number of platters each of a given size), the maximum RPM that it can run at without exceeding the thermal envelope is calculated.

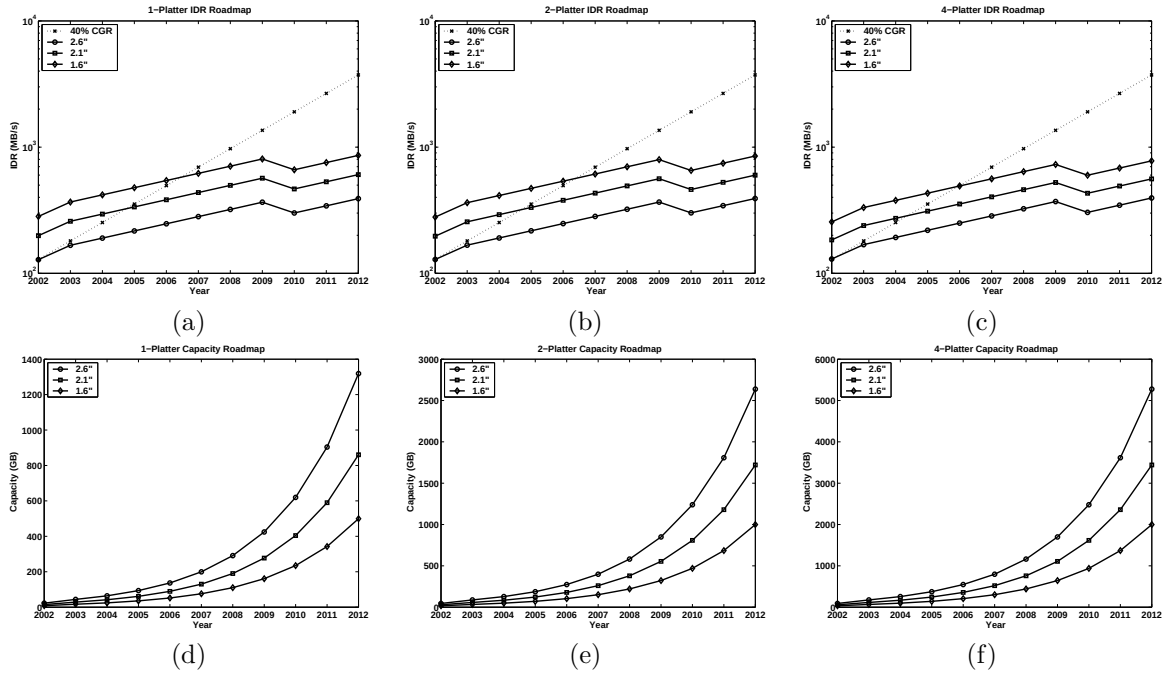


Fig. 3.4. Disk Drive Roadmap. Each solid curve (for a given platter size) gives the maximum attainable IDR (in the top 3 graphs) with that configuration which is within the thermal envelope of 45.22 C, and the corresponding capacity (in the bottom 3 graphs), for a given year. The dotted line indicates the 40% target growth rate in IDR over time. Any curve which falls below this dotted line fails to meet the target for those years.

This coupled with the density values for the corresponding year can be used to calculate the maximum IDR (and its capacity) that such a disk can sustain within the envelope. In addition to these lines, the IDR graphs also plot the 40% growth rate target (the dotted line). The IDR roadmap points which yield a value in any year larger than the corresponding value in the dotted line indicate that the corresponding configuration can yield a higher data rate than the target. Typically, in such years, the manufacturer of such a disk may opt to employ a lower RPM to just sustain the target IDR, rather than what is really possible. The more interesting points are where the roadmap lines intersect the target IDR line. Note that the y-axes of all IDR roadmap graphs are in log-scale.

Let us first consider the 1-platter roadmap. We can see that the 1.6" platter, and the 2.1" to a certain extent, are able to provide (and even exceed in the earlier years) the target IDR until about 2006. The 2.6" platter size, however, starts falling short of being able to meet the projections from 2003 onwards. The 2.1" and 1.6" sizes reach their maximum allowable RPMs in the 2004-2005 and 2006-2007 timeframes respectively, after which they fall short of the IDR requirements. At such points, the manufacturer is presented with three options:

- Sacrifice the data rate and retain capacity growth by maintaining the same platter size.
- Sacrifice capacity by reducing the platter size to achieve the higher data rate.
- Achieve the higher IDR by shrinking the platter but get the higher capacity by adding more platters.

For example, consider the year 2005. From Table 3.4, we notice that a speed of 30,367 RPM would be required to meet the IDR for the 2.1" size. However, this is 1,543 RPM in excess of what is required to be within the thermal envelope. If we shrink the platter to 1.6", we would be able to achieve this data rate with an RPM of 39,857. However, for the one platter device, the capacity drops from 61.13 GB to just 35.48 GB. If the manufacturer wishes to achieve a capacity that is closer to the 2.1" system, an additional platter may be added to push the capacity of the

1.6" drive to 70.97 GB. At this point, the roadmap would shift into the 2-platter system and consequently increase the cooling requirements for the product. In general, it can be seen that the IDR growth of 40% can be sustained till the year 2006. The growth from 2006 to 2007, for the 1.6" platter-size, dips to 25% and to only 14% per-annum subsequently.

When transitioning to terabit areal densities in the year 2010, due to the large increase in the ECC overheads, which is not offset by the BPI growth, the IDR drops from 805.24 MB/s in 2009 to 661.39 MB/s in 2010. After this, the IDR growth of 14% is resumed. By the year 2012, there is over a 2,870 MB/s gap between the 40% CGR point and the best data rate achievable from the design space of this study. Similar trends can be observed for the 2 and 4 platter roadmaps as well with the difference that the fall off from the roadmap is slightly steeper (despite conservatively assuming a higher cooling budget for them), since they cause a higher viscous dissipation making RPM an even bigger issue in restricting their maximum data rates.

3.4.2 Impact of Other Technological Considerations

The influence of different drive parameters such as the form factor, aggressiveness of ZBR, etc., and other external conditions such as the effectiveness of the cooling system, on the above roadmap is now presented.

3.4.2.1 Cooling System

The heat transfer rate from the drive depends on the temperature gradient between its enclosure surfaces and the ambient air. Thus, reducing the ambient external air temperature (by using a more powerful cooling system) allows more heat to be extracted from the inside. We consider two configurations that are 5 C and 10 C cooler than the values (referred to as Baseline) used in the previous experiments, and those results are shown in Figure 3.5.

It can be observed that better cooling can indeed provide benefits. Consider the 1-platter configuration. For the 2.6" platter-size, lowering the ambient temperature from 28 C to 23 C

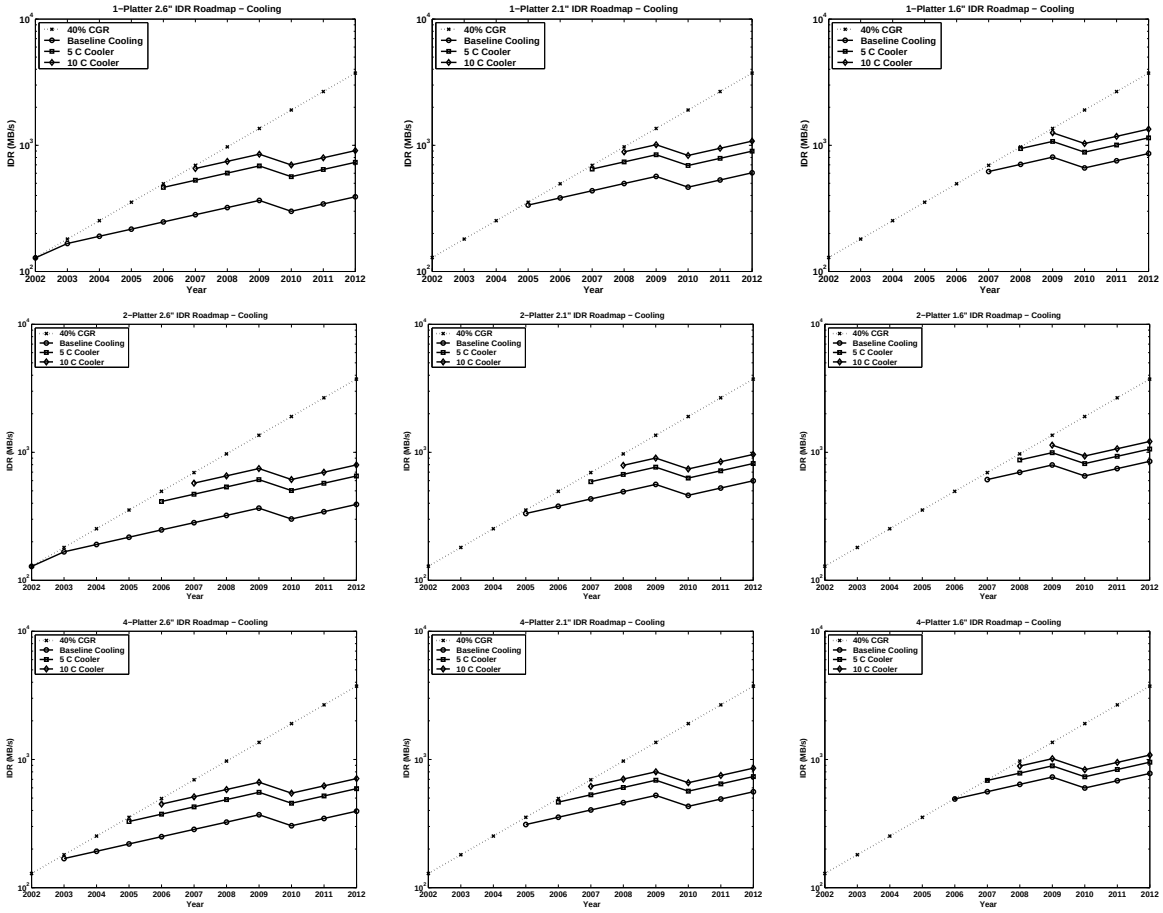


Fig. 3.5. Improvements in the Cooling System. Each row is for a particular platter-count, and each column for a particular platter size. Each graph shows the IDR in the original roadmap (Baseline), together with those when the ambient external air temperature is 5 C and 10 C lower. The curves are shown only for the data points where they fail to meet the target 40% CGR.

(approximately room temperature) allows it to meet the target data rates till the year 2005, while it was falling off at 2003 earlier. Under the original cooling provisions, only the 1.6" platter size would have been able to provide the IDR that satisfies the 40% CGR curve. The ability to use the 2.6" size, with improved cooling, would be able to provide a capacity of 93.67 GB compared to only 35.48 GB for the latter with the 1-platter system. Overall, the 5 C and 10 C temperature reductions allow the roadmap to be lengthened by one and two years respectively (2007 and 2008) using the 1.6" size. However, the terabit transition cannot be sustained even by the aggressive cooling systems. The trends are found to be similar for higher platter counts as well. For the 2-platter system, the 40% CGR roadmap can be maintained for as long as the single-platter one. Beyond 2007, the fall-off of the data-rates are sharper for the multi-platter drives. For instance, for the 5 C cooler configuration, the maximum attainable speed is only 66,240 RPM (with an IDR of 871.13 MB/s) whereas the 1-platter disk can provide 71,580 RPM (941.36 MB/s). A similar trend is observable for the 10 C configuration as well. This behavior is even more pronounced for the 4-platter system, where the roadmap itself drops off a year earlier. As the viscous dissipation is higher if there are more platters, they tend to limit the achievable RPM to a greater extent.

Though these enhancements can buy some more time in the roadmap (and enhance drive reliability as well [4]), increasing the cooling requirement is not that attractive from the practical viewpoint because:

1. It is not as easy to control ambient temperature in a desktop setting, since a large fraction of disk drives are sold as commodity.

2. Even if the drives can be housed in a machine room, that has more extensive cooling systems, the associated costs may be prohibitive [50, 110].

A more preferable alternative is to come up with designs that can deliver the required performance/capacity without increasing the cooling requirements.

3.4.2.2 Aggressive Zoned-Bit Recording

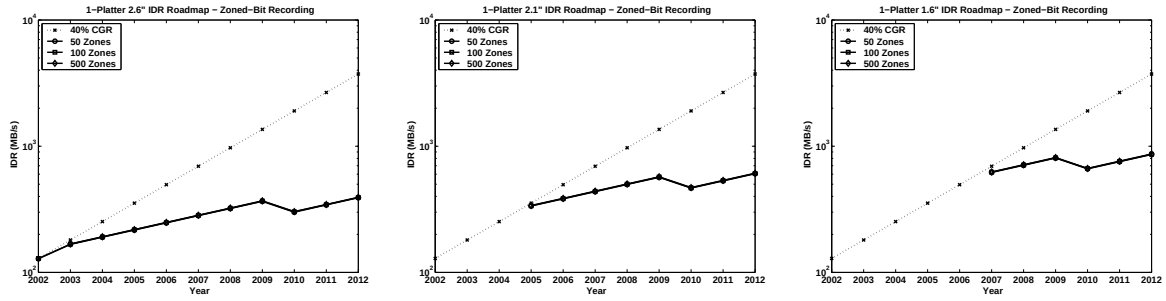


Fig. 3.6. Impact of Aggressive Zoned-Bit Recording

Another way of potentially getting higher data-rates, without increasing the disk RPM, is by increasing the number of zones. Having more zones allows for better utilization of the outer tracks on the platter surface. This provides both increased capacity and also higher data-rates. In the baseline roadmap, it is assumed that each disk has 50 zones/surface. In Figure 3.6 the performance of the 50-zone configuration is compared to those with 100 and 500 zones/surface respectively for the 1-platter roadmap. The results are similar for the other two platter-counts.

As seen from the graphs, using more zones has a negligible impact on the data-rate, indicating that the utilization of the disk-perimeter (and its associated bit allocation) is good enough with just 50 zones for the range of BPI, TPI, and platter-sizes in the roadmap. It is well-known that after a certain point adding more zones reaches a point of diminishing returns. Since the ideal-capacity configuration is also the one that maximizes the achievable IDR, the performance would show a diminished return. Moreover, increasing the number of zones also entails a significant increase in the complexity of the drive-electronics.

3.4.2.3 Form Factor of Drive Enclosure

Another trend in drive design is to use smaller enclosures, especially for disks with platters that are 2.6" or smaller. With a smaller form factor, the base and cover areas are reduced, lowering the heat transfer rate from the drive. This can cause the internal drive temperature to be higher than for one with a larger form factor. In order to compensate for this and ensure that the thermal-envelope would not be violated, more cooling would have to be provided. We studied how the roadmap would be affected if we employed a disk that uses the 2.5" form factor enclosure, instead of 3.5" (note that this is not the platter size). The dimensions for such a disk case would be 3.96"×2.75" [101], and would therefore still be able to house a 2.6" platter.

The 2.6" platter-size is equally ineffective at both form-factors in tracking the 40% CGR curve. Also, the larger form-factor delivers slightly better performance than the 2.5" one. This gap widens as we move into the higher RPM ranges of the roadmap. This is due to the limitations of the smaller base and cover to dissipate the internal drive heat to the ambient air. These limitations start becoming more serious as the internally generated heat increases for the higher RPMs. The differences were not found to be significant between the 1- and 2-platter models. However, the roll-off in the data-rates are slightly higher for the 4-platter configurations, due to the additional viscous heat that is generated.

In general, the results clearly indicate that we tend to fall off the roadmap even at 2002 by going for such a smaller enclosure, and a much more aggressive cooling system to cut the ambient temperature by at least another 15 C is needed before this becomes a comparable option.

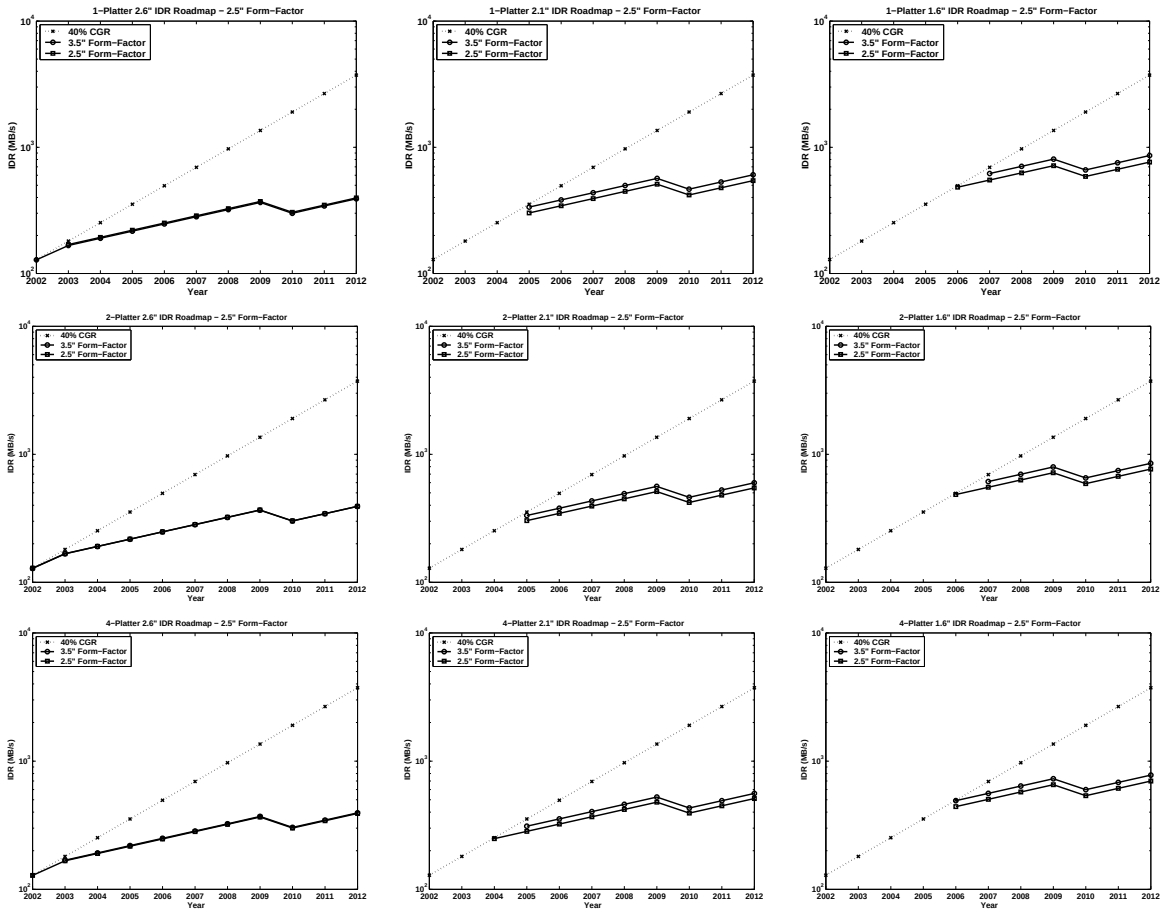


Fig. 3.7. Using a Smaller Drive Enclosure

Chapter 4

DRPM: Using Dynamic Speed Control for Power Management

4.1 Introduction

In Chapter 2, we have seen that disk power management, which involves spinning the disk down to the standby-state to save power followed by a spinup to service a request arriving subsequently, is relatively challenging to apply in server storage systems. This is primarily due to the nature of the idle-periods, which occur in large numbers but are very short in duration, rendering power management ineffective if no performance loss is desired. This is particularly important in servers where performance and availability are critical requirements. Thus, the objective is to increase the idleness such that power management operations could be effectively performed.

One possible solution is to use a large cache, under the assumption that the I/O workload will exhibit good locality. The cache can also potentially be used to delay writes as proposed by Colarelli et al [18] 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. Although prefetching does not directly reduce the power consumption of the disks, when done *aggressively* and *accurately*, could make the workload more bursty and thus potentially increase the idleness. Indeed, such techniques have been shown to be effective in the context of single-user systems [78]. However, it has been shown [75] that we need overly optimistic (and grossly unrealistic) levels (> 16) of perfect prefetching to get even moderate savings in energy using spindown-based disk power management.

The root cause of this inability to do power management stems from the fact that there are only two states of operation of the disk - 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 due to the unavailability of the device to service an I/O request presented to it. In this chapter a new option called *Dynamic Rotations Per Minute (DRPM)* is introduced, 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.

Recall from Chapter 3 that the three main factors affecting the power dissipation are the platter-size, RPM, and number of platters. At the time of designing the drive, all three could be varied in order to achieve a particular power budget. For instance, since laptop disks are designed for low-power operation, they have smaller and fewer platters and also a lower RPM. Server disks follow a different trend. However, among the three parameters, only the RPM could be potentially varied dynamically. This is the basic rationale behind DRPM.

Slowing down the speed of spinning the platters can provide at least quadratic (2.8th power in the best case) power savings, as presented in Chapter 3. (In this chapter, we have assumed a quadratic power model and an even more pessimistic linear model to evaluate DRPM. As will be shown later, even under these pessimistic assumptions, DRPM provides significant power savings for server workload scenarios). 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 [67, 70] (referred to hereafter as TPM):

- 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 [83, 35, 27] in modern microprocessors 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) provides quadratic power savings and the slower clock stretches response time linearly, thus providing energy savings during the overall execution.

In this chapter, behavioral models of DRPM disk drives, both in terms of performance and power, are developed. The rest of this chapter looks at evaluating this mechanism across different workload behaviors. First, an optimal algorithm, called $DRPM_{perf}$ is presented (that provides the maximum energy savings without any degradation in performance) and evaluated under different workloads, and its pros and cons are compared with an optimal version of TPM, called TPM_{perf} (which provides the maximum power savings for TPM without any degradation in performance). The sensitivity of DRPM to different physical/technological trends is also analyzed. Then, a simple heuristic is presented 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. Finally, the engineering issues in building DRPM disk drives are discussed.

4.2 Dynamic RPM (DRPM)

In Section 2.1, it was shown that the spindle-motor is the largest consumer of power in a disk-drive and hence TPM techniques attempt to turn off this component and transition the disk to the standby mode. Moreover, as presented in Chapter 3, the power consumption due to the rotating disks can be expressed as:

$$Power \propto (\#Platters) * (RPM)^{2.8} * (Diameter)^{4.6}$$

All three factors can varied at the design-time of the disk in order to satisfy a particular power *budget*. For instance, laptop disks are built to operate in a battery-powered system, where energy resources are scarce. Hence, such disks tend to be smaller, slower, and have fewer platters

whereas server disks follow an opposite trend. Among the three parameters, only the RPM could potentially be varied dynamically, providing nearly cubic reduction in the viscous dissipation.

4.2.1 Basics of Disk Spindle Motors

A detailed exposition of disk spindle motors (SPMs) can be found in [60, 97]. 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.

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 [58] 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/Hitachi hard disk drives [58] 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 the following other issues:

- *Acoustics:* Noise can increase at higher RPMs. Vibration reduction techniques in the motor assembly via structural damping with some form of viscous fluid [6, 54] or air-bearings [108], instead of the traditional ball-bearings, have been used to address this concern.

- *Non Repeatable Runout (NRRO):* With higher RPMs (and track densities also rising), off-track errors (called *Track Misregistration*) can increase. The damping technique employed has a significant consequence on such errors. Again, the use of fluid or air-bearings can reduce NRROs significantly.

As can be seen from the above discussion, technologies have alleviated some of these design considerations in the development of high RPM disks. However, the power associated with high RPMs still remains and this chapter focuses on this specific aspect.

4.2.2 Analytical Formulations for Motor Dynamics

DRPM modulates the rotational speed of the spindle motor to provide the energy savings. The time overhead needed to effect an RPM change, and the power of the resulting state as a function of the RPM are calculated in the following discussions.

4.2.2.1 Calculating RPM Transition Times

In order to calculate the time required for a speed-change, we need some physical data about the spindle-motor. As this information is proprietary, the characteristics had to be inferred by studying the characteristics of a variety of DC brushless motors. The necessary information was obtained from the datasheet of a Maxon EC-20 20 mm flat brushless permanent magnet DC motor [73], whose physical characteristics closely match those of a hard disk spindle motor. Table 4.1 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 4.1. Characteristics of Maxon EC-20 Motor

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:

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

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 [57]), we calculated the moment of inertia of the load J_L (in gcm^2) as:

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

where n_p is the number of platters. (The moment of inertia calculation approximated the platter-stack to be a solid cylinder).

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

Therefore, we have

$$\Delta t = 2.693 \times 10^{-4} \Delta n \tag{4.1}$$

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

4.2.2.2 Calculating the Power Consumption at an RPM Level

An experimental curve-fitting approach was used to find how the RPM is related to the SPM power consumption. There exists a commercial hard disk today called the Multimode Hard Disk Drive [76] from Sony that indeed supports a variable speed spindle motor. However, 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 usage values of this disk for different RPM values provides insight on how the current drawn varies with the RPM in a real implementation.

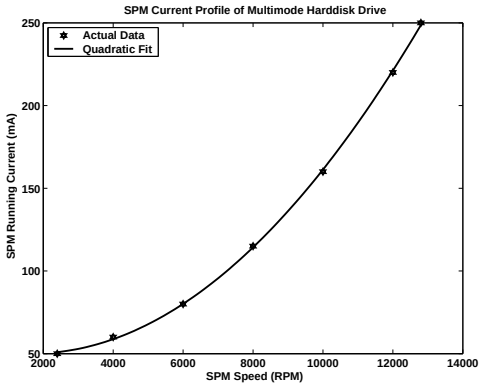


Fig. 4.1. Current Drawn by Sony Multimode Hard Disk

The impact of RPM on the current drawn by the SPM is shown in Figure 4.1 for the Multimode hard disk (repeated from [76]), as the distinct points of current values between 2400 and 12000 RPM. To find a general relation between these two variables (RPM and current draw), a simple curve fit of the above data points was attempted. The effect of this fit is shown by the curve in Figure 4.1, suggesting atleast a quadratic relationship between the two (although it can be as high as cubic).

This Multimode disk is composed of only two platters, while the server class disks can have several more platters. Consequently, this data cannot directly be applied in the models. On the other hand, a study from IBM [97] projects the relation between idle power and RPM for 3.5" server class IBM disks, shown by the points in Figure 4.2. In our power modeling strategy for a variable RPM disk, we employed two approaches, to capture a quadratic and a pessimistic linear relationship respectively:

- The points from the IBM study [97] were taken and a quadratic curve was used to approximate their behavior as is shown by the solid curve in Figure 4.2 to model the idle power as

$$P_{idle} = 1.3182 \times 10^{-7} rpm^2 - 4.4385 \times 10^{-4} rpm + 8.6425 \tag{4.2}$$

- A linear least squares fit through the points was also performed as is shown by the dotted line in Figure 4.2 to model the idle power as

$$P_{idle} = 0.0013 rpm + 4.158 \tag{4.3}$$

Both models are used in the 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 4.2 which shows the differences between linear and quadratic models within this range are not very significant.

Equations 4.1 and 4.2/4.3 provide the necessary information in modeling the RPM transition costs and power behavior of the DRPM disk-drive. For the power costs of transitioning from one RPM to another, it is conservatively assumed that the power during this time is the same as that of the higher RPM state.

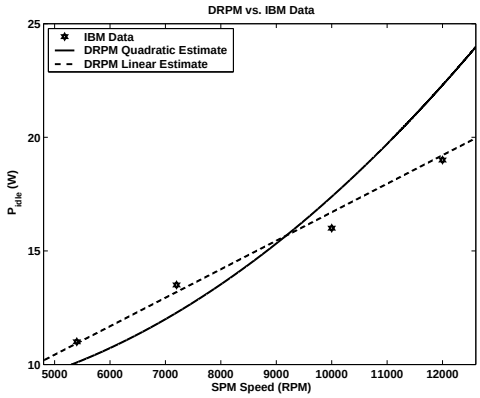


Fig. 4.2. Comparison of DRPM Model to IBM Projections given in [97]

4.3 Experimental Setup and Workload Description

The evaluations are conducted using the DiskSim [30] simulator, augmented with power models to record the energy consumption of the disks when performing operations like data-transfers, seeks, or when just idling. The DRPM model is implemented in this simulator and the queuing and service delays caused by the changes in the RPM of the disks in the array are accurately captured.

The default configuration parameters used in the simulations are given in Table 4.2, many of which have been taken from the data sheet of the IBM Ultrastar 36ZX [57] 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 4.2. 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). Also, in the power models, the power penalties for the active and seek modes (in addition to idle power) 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 4.3.

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
Maximum Disk Rotation Speed	12000 RPM
Idle Power @ 12000 RPM	22.3 W
Active (Read/Write) 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 Algorithm	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 4.2. Simulation Parameters with the default configurations underlined. Disk spinups and spindowns occur from 0 to 12000 RPM and vice-versa respectively.

Several RPM operating levels have been considered, 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 is used in the experiments is a 12-disk RAID-5 array, with a quadratic DRPM power-model and a step-size of 600 RPM.

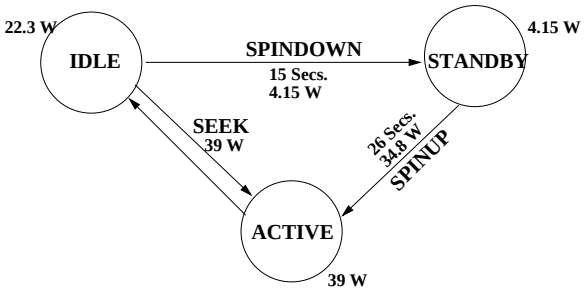


Fig. 4.3. TPM Power Modes

To demonstrate the potential of DRPM under a wide 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, synthetic workloads are employed. The synthetic workload generator injects a million I/O requests with different inter-arrival times, and request parameters (starting sector, request-size, and whether the access is a read or a write). All the workloads consist of 60% read requests and 20% of all requests are sequential in nature. These characteristics were chosen based on the observations of server workloads presented in the study by Ruemmler and Wilkes [96]. Since a closed-system simulation may alter the injected load based on service times of the disk array for previous requests, an open-system simulation was conducted (as in other studies [93, 87]).

Two types of distributions for the inter-arrival times were considered, namely, exponential and Pareto [61, 86]. 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).

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

$$E(x) = \frac{\alpha\beta}{\alpha - 1}$$

The Pareto distribution chosen for the experiments have a finite mean and infinite variance.

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

In the results that follow, the term *workload* to used 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.

In general, statistics to differentiate between the schemes are collected after the initial start up effects.

4.3.1 Metrics

The metrics that are used in the evaluations are: *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. All the disk activity (states) are monitored and their duration in each state is recorded. This is used 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 RPM 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.

4.4 Power Optimization without Performance Degradation

4.4.1 Energy Breakdown of the Workloads

Before examining the 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 4.4 gives the breakdown of

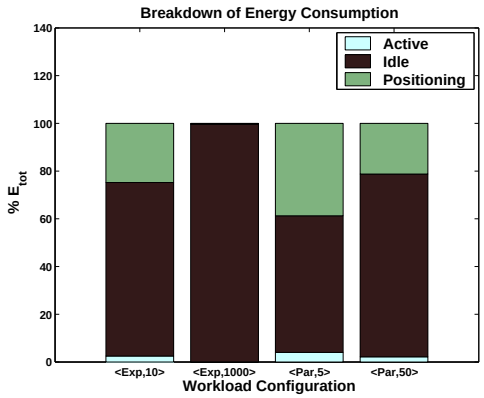


Fig. 4.4. Breakdown of E_{tot} for the different workloads. On the x-axis, each pair represents a workload defined by <Probability Distribution,Mean Inter-Arrival Time> pair.

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 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). Similar trends were observed for the real workloads that were studied in Chapter 2.

4.4.2 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. In the first set of results, the performance of TPM and DRPM are analyzed in the absence of any performance degradation. 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 turning off the SPM (while DRPM can take it down to only 3600 RPM).

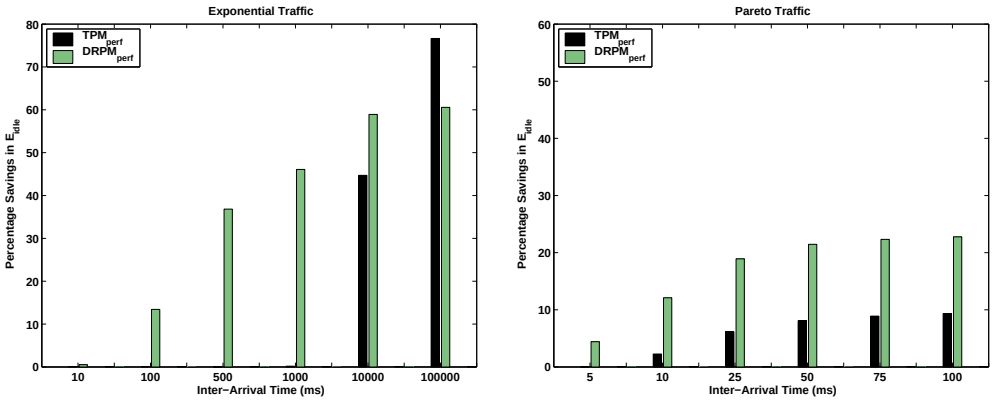


Fig. 4.5. Savings in Idle Energy using TPM_{perf} and $DRPM_{perf}$ are presented for the quadratic power model.

Note that $DRPM_{perf}$ and TPM_{perf} 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 4.5 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 the 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.

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), the ensuing discussions are focused on those regions.

4.4.3 Sensitivity Analysis of $DRPM_{perf}$

4.4.3.1 Number of Platters

Disks show significant variability in platter counts. At one end, the laptop disks have 1 or 2 platters, while high-capacity server class disks could 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 4.6 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.

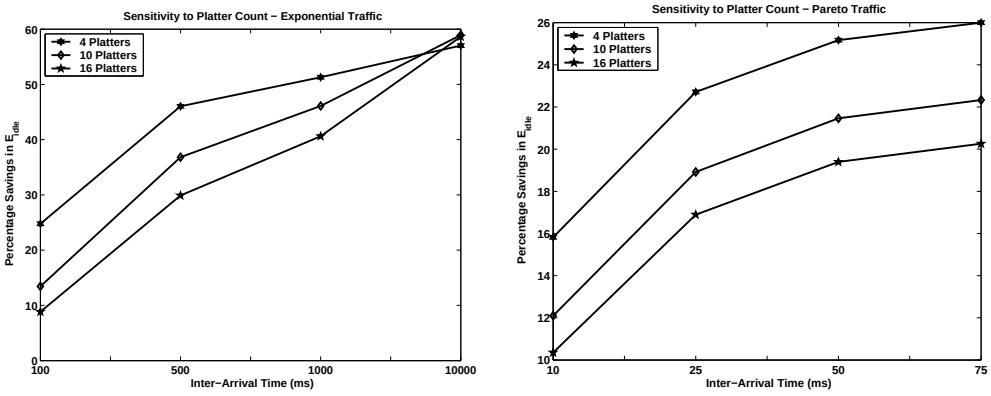


Fig. 4.6. Sensitivity to Number of Platters in the Disk Assembly

4.4.3.2 Step-Size of the Spindle Motor

The step-size of the spindle motor has both performance and energy-savings ramifications on the behavior of DRPM. The availability of a large number of steps (i.e. a smaller step-size) allows finer modularity of RPM control - it may allow lower RPM operations in certain cases which may not have been possible with a larger step-size, while still keeping transition costs low. This effect is seen in Figure 4.7, where the step-size of 600 RPM, which is used in our default configuration, provides slightly higher energy savings than a larger step-size of 2100 RPM, though the differences are not very significant. This suggests that even a few steps between the high and low RPM values can provide good energy savings. It has been shown that even using just two RPM-levels provide significant energy savings, even much more than TPM [9].

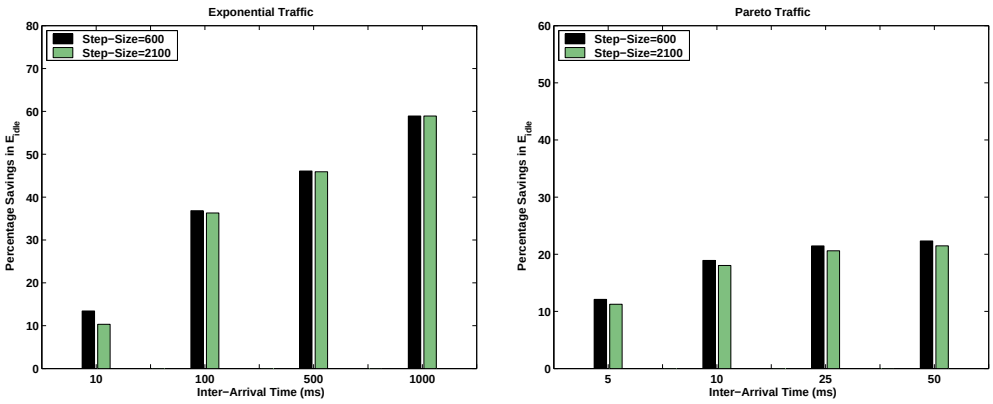


Fig. 4.7. Sensitivity to the Step-Size Used by DRPM

4.4.3.3 Quadratic vs. Linear Power Model

While the earlier results were presented with the quadratic model, the savings for $DRPM_{perf}$ with the linear model are shown in Figure 4.8. 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 the 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 4.5).

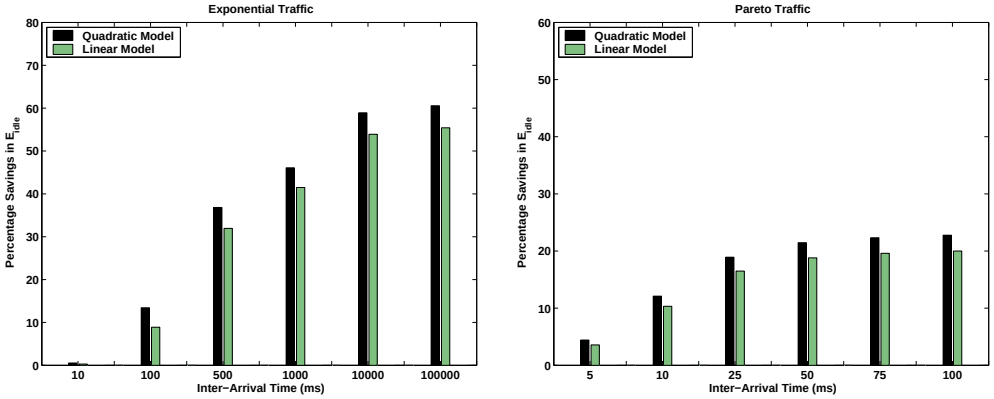


Fig. 4.8. Behavior of $DRPM_{perf}$ for a Power Model that relates the RPM and P_{idle} linearly

4.4.3.4 RAID Configuration

RAID-5 has been used in all previous experiments. Another popular configuration in servers is RAID-10 wherein two mirrored disk arrays are maintained, and a read request is sent to one with the lower load while a write needs to update both mirrors. Figure 4.9 compares the idle energy savings of $DRPM_{perf}$ with RAID-5 and RAID-10 (both have the same number of disks). No significant differences are found in the savings obtained for both RAID configurations.

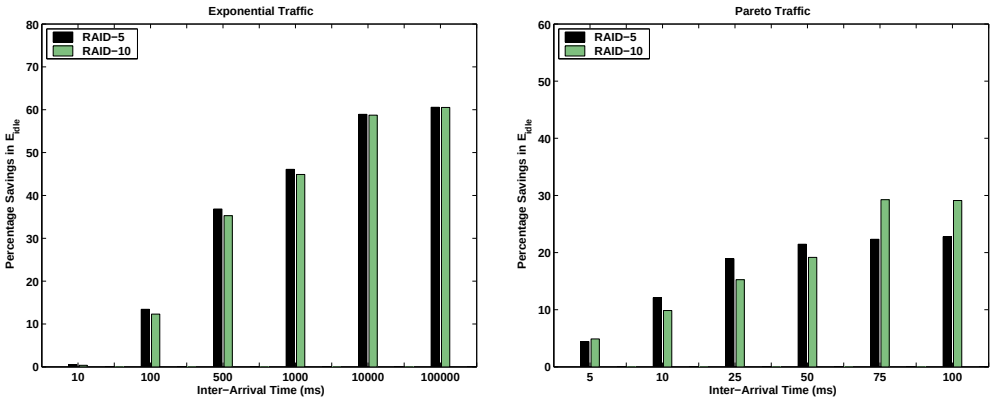


Fig. 4.9. Sensitivity of $DRPM_{perf}$ to RAID-Level.

4.4.3.5 Number of Disks

Servers are typically configured with several disks to both optimize performance (in terms of bandwidth) and also fault-tolerance (mirroring). Figure 4.10 shows the idle energy savings with $DRPM_{perf}$ for RAID-5 arrays with 12 and 24 disks. As the number of disks increases (with each individual disk having the same capacity), while there may be some benefit from performance, the power consumption definitely rises. At the same time, for a given load, the idle power would be higher on a larger disk configuration. Consequently, more savings are obtained with DRPM for a larger number of disks in the system. As idle times get very large, the relative differences in savings for different disk configurations would diminish since each of these curves is normalized with respect to the corresponding configuration without power saving approaches. These results motivate the need for power savings with larger disk arrays and the possible benefits of DRPM in such settings.

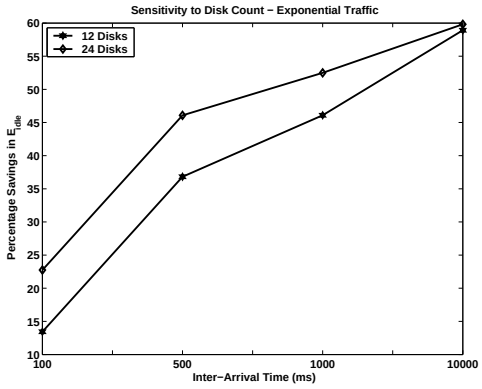


Fig. 4.10. Sensitivity to Number of Disks in the Array

4.5 A DRPM Control Policy

In the previous section, the potential benefits that DRPM could provide was quantified under the idealizing assumption that we had perfect knowledge of the idle periods. Now, a control-policy is presented, in order to choose the RPM to operate at in any given instant under a more realistic scenario, when such an idle prediction oracle is not available.

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 response 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.

The values for UT and LT can be specified in or be a function of a specification in a Service Level Agreement between the customer who wants to host the application and the data-center where the storage-system is operated and managed. The three scenarios listed above are depicted in Figure 4.11 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 $n = 250, 500, 1000$. In the 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.

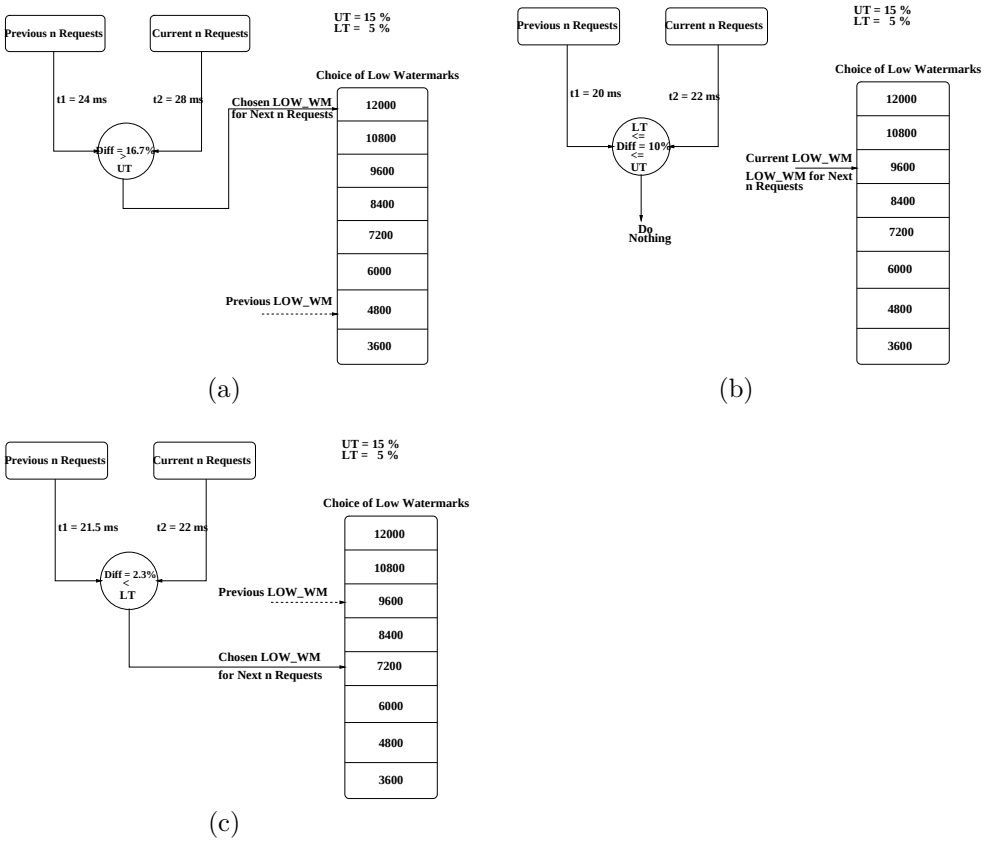


Fig. 4.11. The operation of the DRPM control policy for $UT = 15\%$ and $LT = 5\%$. In each figure, for the choice of low watermarks, the dotted line shows where LOW_WM is before the policy is applied and the solid line shows the result of applying the scheme. The percentage difference in the response 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.

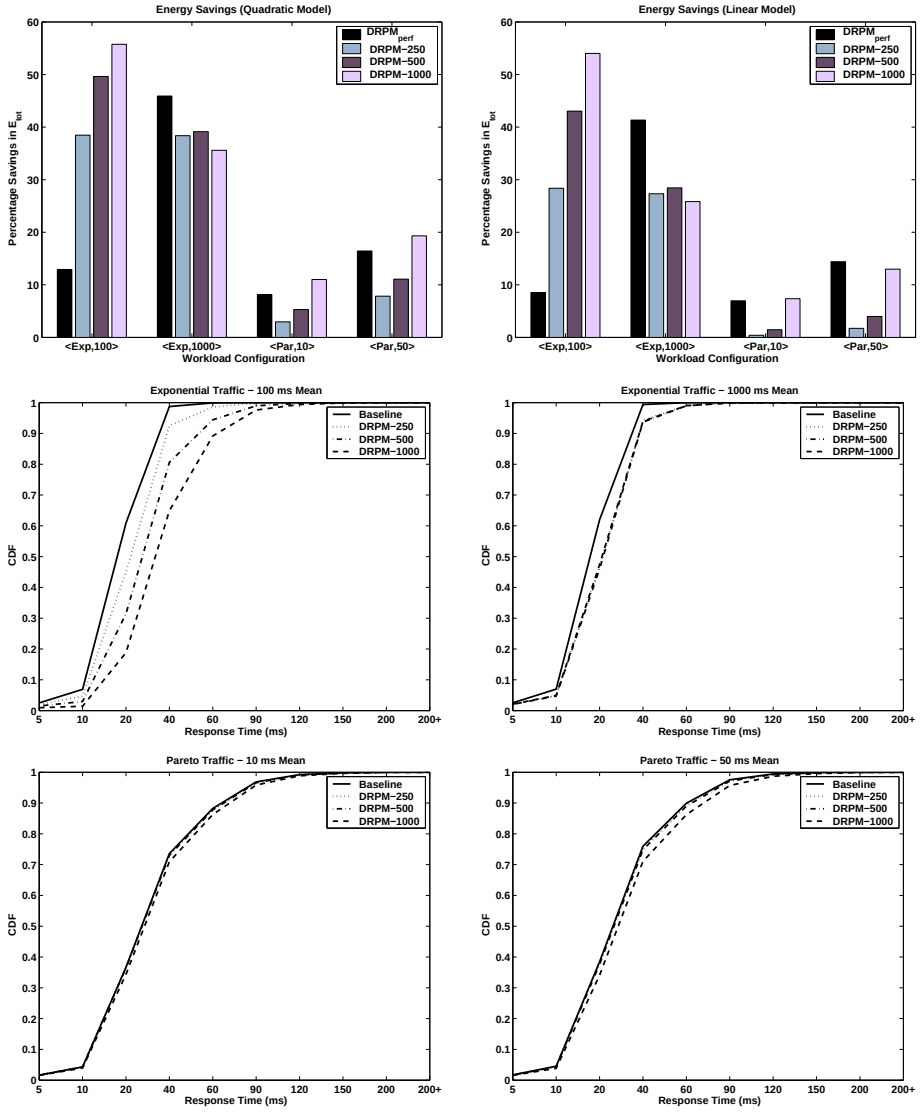


Fig. 4.12. DRPM Control Policy Scheme Results. $UT = 15\%$, $LT = 5\%$, $N_{min} = 0$. The results are presented for $n = 250, 500, 1000$, referred to as DRPM-250, DRPM-500, and DRPM-1000 respectively.

4.5.1 Results with DRPM Control Policy

The first set of results in Figure 4.12 show the energy savings and response time degradation of the DRPM control policy 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.

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 policy 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 the control policy 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 control policy. Still the energy savings for the policy are quite good and are not far away from $DRPM_{perf}$, which has perfect knowledge of idle times. The results for the control policy 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 (see Figure 4.13 which shows the amount of time spent in different RPM levels for <Exp,100> with two different n values). This can result in greater energy savings for larger n values as is observed in many cases.

The response time characteristics of the control policy are shown as CDF plots in Figure 4.12, 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 control policy 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. It can be seen that a finer resolution ($n = 250$ requests) does tend to keep the response time CDF of the control policy close to the Baseline. In <Par,10> and <Par,50>, one can hardly discern differences between the Baseline and the corresponding control policy results. Remember that the Pareto traffic has bursts of I/O requests followed by longer idle periods. Since the control policy 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, and we still find over 25% energy savings.

4.5.2 Comparison with Static RPM Choices

While all the results presented so far have employed the DRPM technique, where the angular velocity of the disk is varied dynamically, another power saving approach could be to

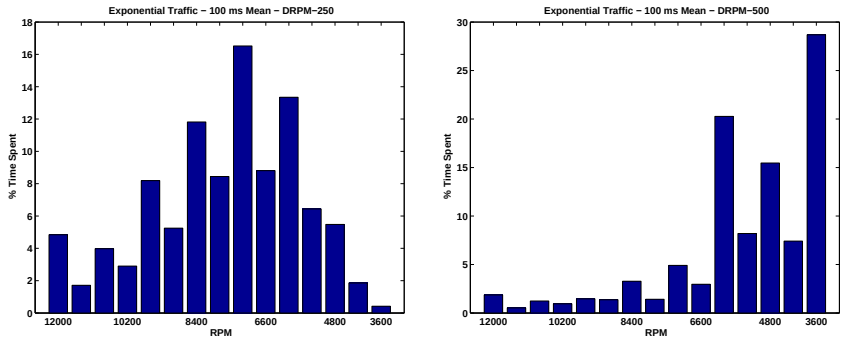


Fig. 4.13. Average Residence-Times in the Different RPMs for the DRPM control policy for $n = 250, 500$. The values presented have been averaged over all the disks in the array. A step-size of 600 RPM is used. Note that the lower RPMs are exercised more with a larger n.

simply use disks of a lower (but constant) RPM. We refer to such options as static RPM choices. For the static schemes, two options are considered - All and Half - where all the disks are of a lower RPM, or only half the disks are of lower RPM and the other half running at 12000 RPM. These schemes are denoted as All-3600, All-5400, All-7200, Half-3600, Half-5400 and Half-7200, depending on what speed disk they use statically.

The results for <Par,5> are given in Figure 4.14

As is to be expected, the static schemes can give considerable energy savings based on what RPM is chosen. Consequently, we see that the static schemes at 5400 and 7200 RPMs give 3-4 times more savings than DRPM. When we have very low RPM disks (3600), the execution time itself gets stretched considerably that active power becomes a big constituent of the overall energy, making All-3600 and Half-3600 to not give any more savings (or even increase energy consumption). On the other hand, the problem with the static schemes is the considerable degradation in response times as is shown in the figure, with their CDFs showing significant deviations from that of the Baseline (while our DRPM control policy closely matches the Baseline).

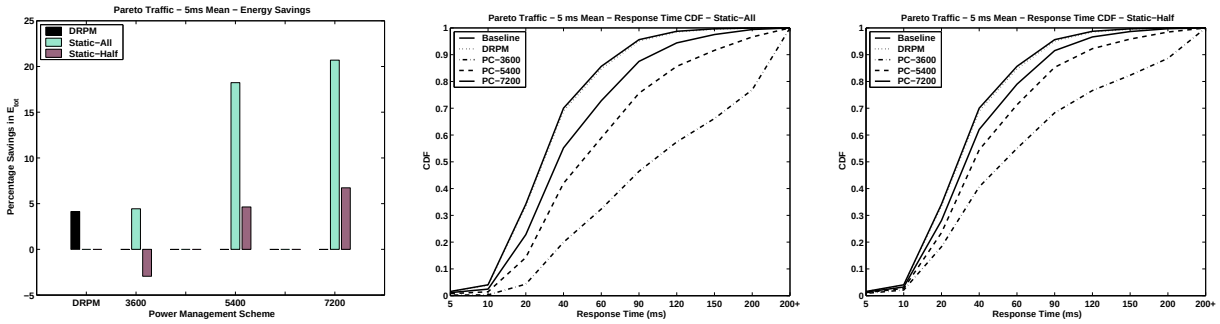


Fig. 4.14. Static RPM vs. DRPM. The workload is <Par,5> and the graphs are presented for the quadratic power model. For DRPM, we chose $n = 1000$. The "PC" in the response-time graphs stand for "Pre-Configuration".

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

The DRPM control policy 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 4.15 (a) with those of the results in Figure 4.12 (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 4.15 (b) with those of the results in Figure 4.12.

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

4.6 Issues in Implementing DRPM Disks

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

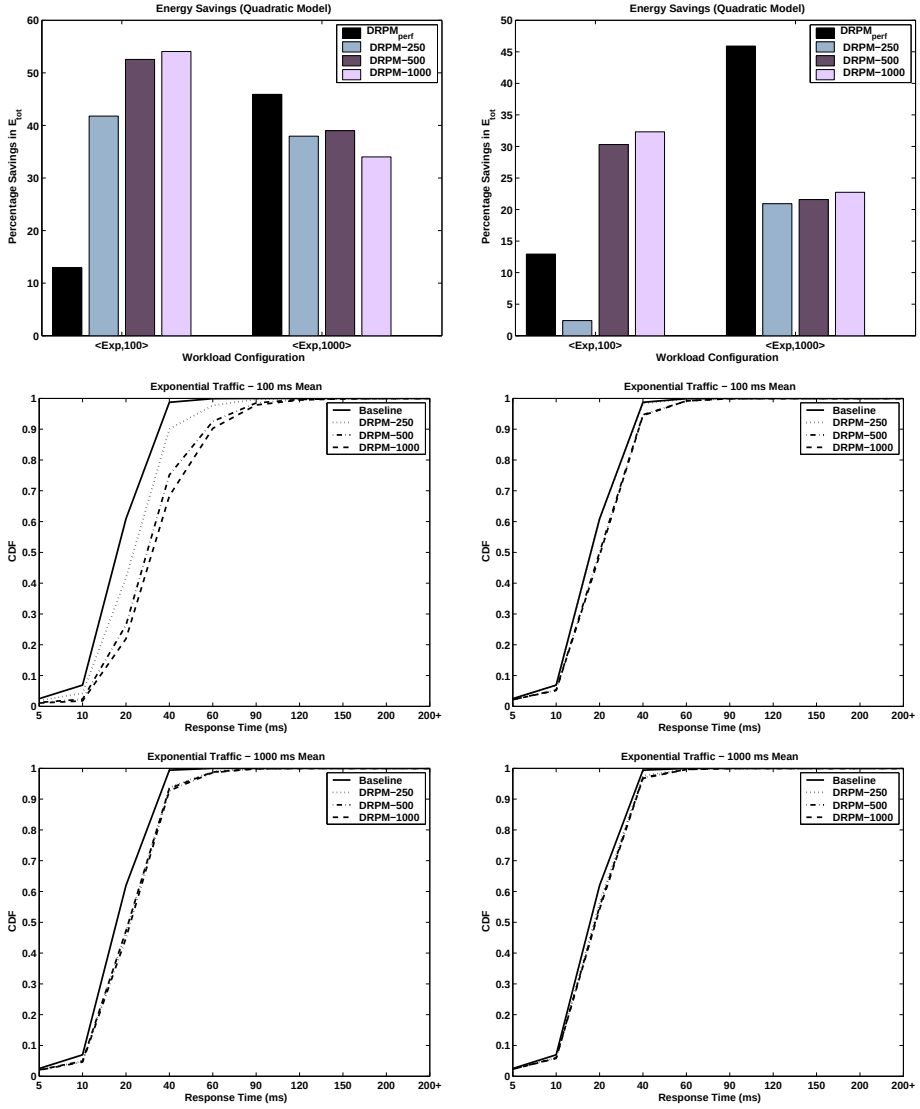


Fig. 4.15. Controlling UT and LT for Power-Performance Tradeoffs. (a) presents the results for UT=15%,LT=10%. (b) presents the results for UT=8%,LT=5%.

- **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 (RPMs) supported by the given spindle system. The linear velocity varies from the inner to the outer diameter of the disk and this tolerance is usually accommodated when designing the slider. However, in DRPM, since the angular velocity is also being varied, there is a larger variation in the linear velocities, whereby necessitating a re-design of the slider. The Papillon slider presented in [64] is capable of maintaining this constant fly height over the range of RPMs that has been considered in this work.

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

  In hard-disks, positioning the head requires accurate information about the location of the tracks, which 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 TPI increases, higher sampling frequencies are required. This sampling frequency is directly proportional to the spinning speed of the disk f_{spin}. Therefore, when f_{spin} varies, this frequency also needs to be changed accordingly in order to properly sample the servo information. [114] addresses 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}).

  In designing the data channel there are some possible tradeoffs between the design complexity and supporting a large number of data rates. Providing multiple data rates is not

new; all disks that employ Zoned-Bit Recording (which is typical of most products in the market today) already support multiple data rates since the number of bits in each zone are different. The complexity induced by DRPM is that the number of data rates to be supported is a product of the number of zones and the number of RPMs. If there are design constraints associated with providing more than a specified number of data-rates, one possible tradeoff is to reduce the number of zones to accommodate the additional speeds.

- **Providing Speed Control**

  As mentioned in section 4.2, speed control in DC brushless permanent-magnet motors can be achieved using PWM techniques. PWM achieves speed control by switching on and off the power supply to the motor at a certain frequency (called the duty cycle). The choice of duty cycle determines the motor speed.

- **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 ([59, 62]). The frequencies of such operations are much lower than the idle times themselves to really have a significant consequence on the effectiveness of power saving techniques. Still, it is possible that DRPM may be more useful for such activities, since it allows those performance non-critical operations to be undertaken at a relatively slow RPM (for energy savings), while traditional power mode control of transitioning the disk completely to a standby state prevents such activities.

- **Smart Disk Capabilities**

  The anticipated smart disks [1, 63, 93] provide an excellent platform for implementing DRPM algorithms, and also provide the flexibility of modulating the algorithm parameters or even changing the algorithm entirely during the course of execution.

Chapter 5

Overcoming Thermal Constraints via Dynamic Thermal Management

5.1 Introduction

In Chapter 3, it was shown that the thermal design envelope poses a challenge to the data-rate scaling that we enjoyed for nearly the past two decades. Although the brute-force approach to tackle this problem is to increase the cooling budget, this is not an attractive solution given the high cost of cooling and the nature of the disk drive market, where the devices are sold as commodity. It is therefore desirable to come up with a more elegant solution to the problem. In this chapter, we shall present a set of *Dynamic Thermal Management (DTM)* strategies that can be used to boost performance while still working under the constraints imposed by the thermal envelope. In particular, we propose and evaluate two DTM approaches outlined below:

1. Detecting thermal slack (difference between current temperature and the thermal envelope that the disk has been designed for), and exploiting it to temporarily ramp-up RPM (assuming that the disk can support DRPM) for better performance.

2. Deploying a disk that has been designed for the *average case* behavior to run it at a higher RPM than what the worst case would support most of the time, and use dynamic throttling techniques when getting close to thermal emergencies.

Before discussing these two possibilities, the question one may ask is whether such performance improvements are really needed from the application perspective. Consequently, a set of some commercial application traces are analyzed (section 5.1.1) to motivate the need for higher

data rates. Subsequently, the above two mechanisms are discussed as possible ways of achieving such data rates in sections 5.1.2 and 5.1.3 respectively.

5.1.1 The Need for Faster Disks

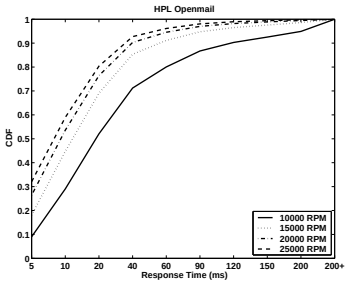
Even though it is apparent that higher data rates would help bandwidth limited workloads, one is still interested in finding out how helpful this can be in the context of realistic workloads. We conducted this evaluation using 5 commercial I/O traces given in Figure 5.1 (a). The TPC-C trace was collected on a 2-way Dell PowerEdge SMP machine, and the TPC-H trace was collected on a 8-way IBM Netfinity SMP system, both using DB2 on Linux as the underlying platform. The table also gives the approximate year when the traces were collected.

The models presented in Chapter 3 are used to capture some of the disk characteristics for the appropriate year (since this information was not always available). All the disks are assumed to have a 4 MB disk cache and ZBR with 30 zones/surface. For the RAID systems, RAID-5 was used with a stripe-size of 16 512-byte blocks. The performance of the disks was simulated using DiskSim [30] with the appropriate RPM in Figure 5.1 (a). The experiments were conducted by increasing RPM in steps of 5000 (without their thermal effects) to find the impact on response time. The objective of this experiment is merely to investigate if using higher RPM disks can provide significant storage system performance boosts.

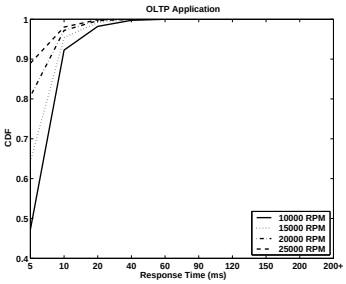
Results from these experiments are shown in Figure 5.1 in the form of CDF plots of the response times. In addition, below each graph, the average response times are indicated. It is observed that a 5000 RPM increase from the baselines provides significant benefit in the I/O response time. The average response-times improved by 20.8% for the OLTP Application to over 52.5% for Openmail. For the Openmail, Search-Engine, and TPC-C workloads, we notice that the entire CDF curve shifts to the left, indicating that most of the I/O requests benefited from the increased RPM. We find the higher RPM helping even those workloads with a considerable number of seeks such as Openmail, where there is an average seek distance of 1952 cylinders per

Workload	Year	# Req.	RPM	Disk Cap. (GB)	# Disks	RAID?
HPL Openmail [3]	2000	3,053,745	10000	9.29	8	Yes
OLTP Application [109]	1999	5,334,945	10000	19.07	24	No
Search-Engine [109]	1999	4,579,809	10000	19.07	6	No
TPC-C	2002	6,155,547	10000	37.17	4	Yes
TPC-H	2002	4,228,725	7200	35.96	15	No

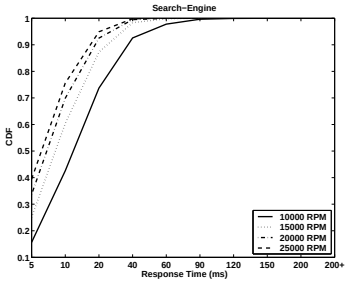
(a) Workloads Used



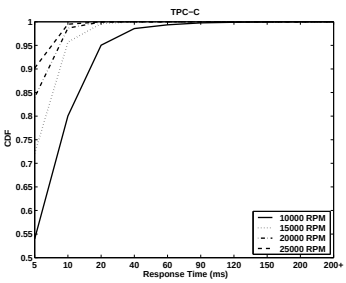
(b) Avg. Resp. Times = {54.54,25.93,18.61,15.35}



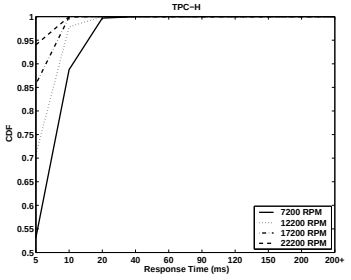
(c) Avg. Resp. Times = {5.66,4.48,3.91,3.57}



(d) Avg. Resp. Times = {16.22,10.72,8.63,7.55}



(e) Avg. Resp. Times = {6.50,3.23,2.46,2.06}



(f) Avg. Resp. Times = {4.91,3.25,2.64,2.32}

Fig. 5.1. Performance impact of faster disk drives for server workloads. Each graph shows the CDF of the response times for each RPM used in the constituent drives in the simulated systems. The average response times for each of the corresponding configurations is shown below each plot in the order of increasing RPMs.

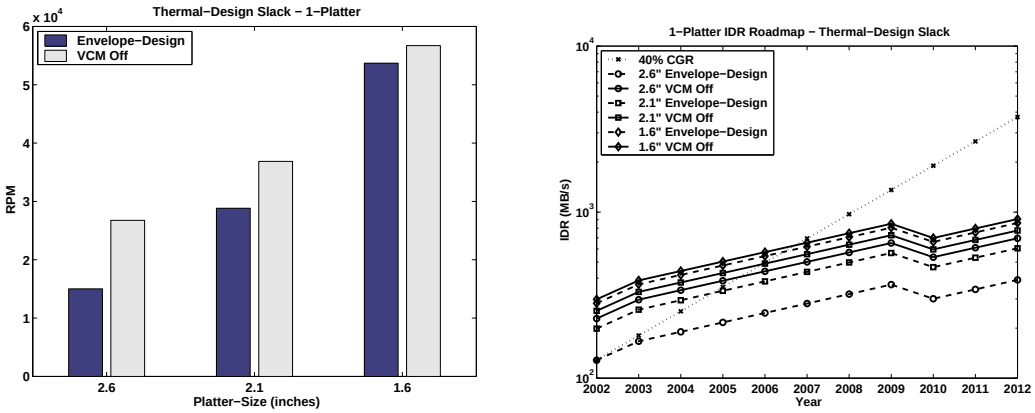
request with over 86% of all requests requiring a movement of the arm. This is because most requests span multiple successive blocks, thus benefiting from the higher RPM.

These results suggest that these workloads would have benefited from a higher RPM even in those systems where they were initially run, though one may not have been able to get there because of the thermal envelope. This makes a strong case for continuing to support higher RPM in future disk drives, even those beyond thermal limits as long as we can provision dynamic thermal management techniques to avoid hitting those limits. As the subsequent two mechanisms illustrate, it would be possible to supply the additional 5-15K RPM, which provided us with the bulk of the performance benefits in these traces, with DTM.

5.1.2 Exploiting Thermal Slack

Note that the thermal envelope was previously defined based on the temperature attained with both the VCM and the SPM being on (i.e. the disk is constantly performing seeks). However, during idle periods (when not serving requests), the VCM is off, thus generating less heat. Further, there could be sequentiality in requests, reducing seek activities. This implies that there is a "thermal slack" to be exploited between the thermal envelope and the temperatures that would be attained if the VCM was off. However, the disk drive has been pre-set with a maximum RPM for a thermal limit based on the VCM being on constantly. If on the other hand, the disk was DRPM-enabled, then we could temporarily push the RPM even higher during periods of no/few seeks without exceeding the thermal limits.

Figure 5.2 (a) shows the RPM that we can drive the design to (for different platter sizes) when exploiting this slack, compared to the original maximum RPM we could support assuming the VCM was always on. We see that there is plenty of slack for the 2.6" platter size, allowing its speed to increase up to 26,750 RPM from the 15,020 RPM with the original thermal envelope. In terms of the data rate, this boost allows it to exceed the 40% CGR curve until the 2005-2006 time frame (Figure 5.2(b)). Even after this time frame, the data rates are around 5.6% higher



(a) Maximum Achievable RPM for a platter-size (b) Revised IDR Roadmap

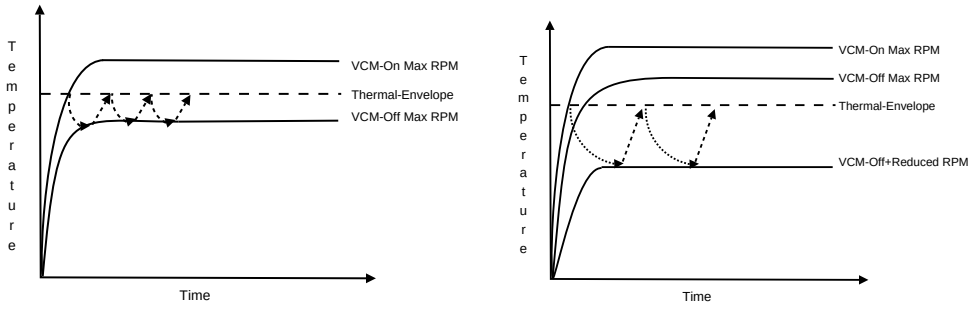
Fig. 5.2. Exploiting the Thermal Slack. 1-platter disk. VCM-off corresponds to the RPM and IDR values attainable when the thermal slack is exploited. Envelope-Design corresponds to the RPM/IDR values when the VCM is assumed to be always on.

than the non-slack based configuration. In fact, the slack for the 2.6" drive allows it to surpass a non-slack based 2.1" configuration, thus providing both better speed and higher capacity for the same thermal budget.

The amount of available slack decreases as the platter size is shrunk (see Figure 5.2(a)), since the VCM power is lower for smaller platter sizes (2.28 W for 2.1" vs. 0.618W for 1.6"). This makes the slack smaller to exploit in future designs with smaller platters. The next solution strategy can turn out to be more rewarding in such situations.

5.1.3 Dynamic Throttling

We shall now present two alternatives to design disk-drives without being constrained by the thermal design envelope by employing throttling techniques at runtime. These techniques are schematically shown in Figure 5.3. The basic idea is that by building a disk with higher RPM, we can benefit on performance in the average case, and throttle the system only when temperature limits are reached.



(a) VCM-alone (b) VCM+Lower RPM.

Fig. 5.3. Dynamic Throttling Scenarios in the context of disks designed with average case behavior rather than worst-case assumptions. In (a), only the VCM is turned off, with the disk continuing to spin at maximum RPM. In (b), the VCM is turned off and the disk is transitioned to a lower RPM.

Let us first consider the scenario in Figure 5.3(a). Here, if both the SPM and VCM are operating continuously, the temperature of the disk (depicted by the legend "VCM-On Max RPM") would violate the thermal envelope. In the absence of the VCM (either there are no requests issued to it, or the requests are sequential to avoid seeks), the temperature falls below the envelope (depicted by the legend "VCM-Off Max RPM"). The throttling mechanism can then operate as follows. Requests are sent to the disk, operating at this higher RPM (than permitted by the normal thermal envelope) until the temperature is close to the thermal limit. At that point, the requests are not issued to the disk for a while (t_{cool}), giving a thermal profile indicated by the downward-pointing dotted curve. After this period, requests (involving seeks) can be resumed and the disk would start heating up again (shown by the rising dotted curve), till it reaches close to the thermal envelope again in time t_{heat}.

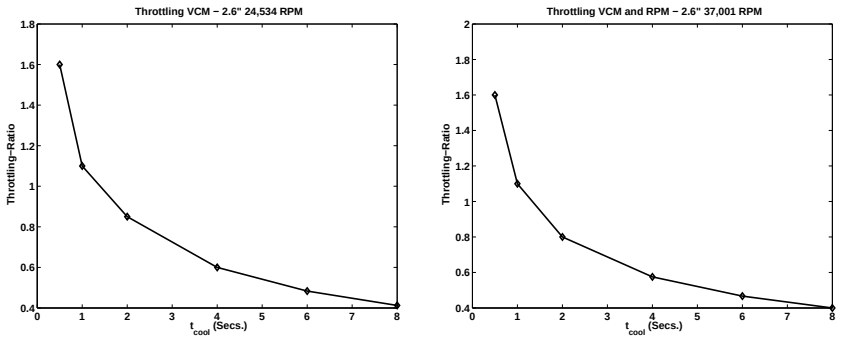
Figure 5.3(b) shows a scenario for throttling with an even more aggressive (in terms of IDR) disk. In this disk, even turning off the VCM would not allow the disk to be within the thermal envelope since the RPM is so high. However, if the RPM was a lower value, then the temperature that would be reached with the VCM off for this lower RPM (depicted by

the legend "VCM-Off+Reduced RPM") is lower than the thermal envelope. In this case, the throttling mechanism would not just stop issuing requests (to cut down VCM power) but would also pull down the RPM of this disk, when the temperature reaches close to the thermal envelope for a time t_{cool} as indicated in the Figure, and then let requests go on for time t_{heat} after bringing up the disk to full RPM. Note that in this case, even though we are performing RPM modulation, we only need a disk with 2 RPM levels, with the servicing of requests always being done only at the higher RPM. Such disks are already starting to appear in the market today [48]. A full-fledged DRPM disk, though not necessary for this throttling mechanism, can provide even finer granularity of temperature control.

The utility of both these techniques is largely dependent on the relative time given to cooling (t_{cool}) and the time it takes to get back from the lower temperature back to the thermal limits (t_{heat}). This ratio ($\frac{t_{heat}}{t_{cool}}$) is denoted as the *throttling-ratio*. In practice, we would like this ratio to be larger (greater than 1) since that allows for longer periods of operation of the disk compared to inoperation (i.e. its utilization is greater than 50%).

Let us now consider two scenario, from the roadmap (Chapter 3) perspective, where each of the throttling techniques is best-suited. First, let us consider a disk-drive that consists of a single 2.6" platter. The highest RPM that can be achieved by this disk, under the assumptions of the original roadmap, is 15020 RPM. Now let us suppose that we would like to be able to use the 2.6" size and be able to satisfy the 40% IDR CGR till the year 2005. From Table 3.3, we find that this needs an RPM of 24,534. Let us assume that we would like to build a disk which operates at this RPM even though in the worst case it would violate the thermal envelope and heat up to 48.26 C. We find that, if the VCM is turned off, the temperature of the drive is 44.07 C, which is within the design envelope and is thus a candidate for the first throttling approach. With such a disk (constant RPM of 24,534), we set the initial temperature to the thermal envelope. We then turn off the VCM for a specific period of time (t_{cool} in seconds) and then turn it back on again. We observe the time (t_{heat}) it takes for the disk temperature to reach the envelope. We repeat

this experiment for various values of t_{cool} and the corresponding throttling ratios are plotted in Figure 5.4(a).



(a) (b)

Fig. 5.4. Throttling ratios with different t_{cool} for (a) VCM-alone and (b) VCM+Lower RPM

For the second throttling scenario, let us say that we would like to stretch the 2.6" roadmap to meet the CGR expectations till the year 2007, whereby a RPM of 37,001 RPM would be required (as shown in Table 3.3). The disk temperatures with and without the VCM turned on are 57.18 C and 53.04 C respectively, both of which are above thermal limits. Let us assume that the disk drive is designed to operate at two RPMs, namely, the full-speed of 37,001 RPM and a lower-level of 22,001 RPM. Then, when there is the danger of violating the thermal envelope, we can lower the RPM in addition to performing seek-throttling to allow the disk to cool down again. The resulting throttling ratio graph for this scheme is shown in Figure 5.4(b).

Both these graphs give insight on how long we need to let the disk cool (throttle) between successive periods of activity. In both cases, it is observed that if we want to keep the active periods at least as long as the idle periods, throttling needs to be done at a relatively finer

granularity (less than a second). The implications of these results, together with a discussion of possible techniques for throttling are given in the next section.

5.1.4 Discussion

The results presented in this section indicate that there is some amount of thermal slack between when the VCM is on and off (section 5.1.2) to temporarily ramp up the RPM. Such techniques can buy us some IDR in the near future. However, as platter sizes continue to diminish, the benefits with this technique are likely to become less significant due to the drop in VCM power. Furthermore, when the latencies to transition between the RPMs is also factored in, the benefit would be even lesser.

The more promising approach, from both effectiveness and complexity of design viewpoints, seems to be the dynamic throttling strategy. Throttling is employed to reduce/avoid sending requests to the disk for a cooling down period, before resuming again. The attractiveness of implementing throttling even with existing disks warrants a closer investigation of techniques for achieving the intended goals with little performance loss. The throttling ratio graphs indicate that keeping the disk utilization higher than 50% requires a finer granularity of throttling - possibly at the granularity of a few dozen requests. If the inter-arrival times of requests in a workload are appropriately spaced, then one could achieve such throttling at little cost (or even for free). Furthermore, another issue is how to best utilize the t_{heat} time period. For instance, one way of masking out the performance loss during the t_{cool} period, when conceivably no activity on the disk-media can be performed, data could still be served out of the disk or controller caches. This motivates investigating how aggressive prefetching during the t_{heat} phase could atleast partially mask out any performance perturbation during the throttling phases. Techniques for co-locating data items to reduce seek overheads (e.g. [95]) can reduce VCM power, and further enhance the potential of throttling. It is also possible to use mirrored disks (i.e. writes propagate to both) while reads are directed to one for a while, and then sent to another during the cool down

period. The throttling ratio graphs give an indication of how many of these disks may need to be employed for a desired cool down period. Finally, the throttling-ratio graphs also indicative of when pushing the RPM would start becoming counter-productive whereby any access to the disk would start heating it so rapidly as to render the device relatively un-usable.

There are several other techniques to enhance IDR while remaining within thermal bounds. For instance, we could use two disks, each with a different platter size. The larger disk, due to its thermal limitations, would have a lower IDR than the smaller one, although the latter, assuming the platter counts to be the same, would have a lesser capacity. Such a configuration allows the smaller disk, which itself could have capacities in the order of several Gigabytes, to serve as a cache for the larger one. This is somewhat similar in philosophy to previously proposed cache-disks [51].

Chapter 6

Conclusions

This thesis has shown how to effectively tackle the problem of high power consumption due the I/O subsystem in servers by developing a fundamental framework to reason about power-performance interactions and using it to develop architectural techniques to combat its effects. In addition, control policies are also developed to effectively utilize the mechanisms to achieve the desired power/performance behavior. First, this thesis shows, via detailed statistical analysis, that traditional approaches to power management are ineffective in server systems due to the physical nature of devices and the characteristics of the workloads that use them [39]. Then, using validated models of fundamental drive characteristics, it is shown that the heat that is generated due to the power consumption is going to severely hinder the performance growth that we have enjoyed so far in server disks [38]. After this exposition of the problems that power poses in designing and running the storage devices, the thesis focuses on the solutions. In this regard, the DRPM scheme [36] is presented, with its behavioral models, its potential benefits, and a control-policy to achieve good energy savings with little to no performance perturbation. Finally, a suite of Dynamic Thermal Management techniques, some leveraging DRPM features, are presented and evaluated to overcome the constraints imposed by the thermal envelope to continue the performance growth in disk drives.

6.1 Research Impact

During the course of this research, the impact of the work has gradually manifested itself. There have been research efforts in studying the applicability of TPM in the context of other server workloads [9, 84], with conclusions similar to the one presented in this thesis being reached

about its inability to tackle the power problem. The rich benefits that DRPM can offer has fueled research in applying it in conjunction with other power optimization techniques [117, 118, 66] in addition to formally analyzing the control-policy itself to select the parameters appropriately [68]. [89] also presents a formal approach to formulating a control-policy for DRPM in the context of multimedia workloads. The use of DRPM for mobile disks is explored in [69]. [90] proposes using DRPM in optical-drives targeting multimedia applications.

The notion of multi-RPM operation to provide fine-granular power management has also started appearing in actual products in the market, such as the Hitachi Desktar 7K400 disk [48]. This drive provides two different RPMs, although data-access is always performed at full-speed (whereby not requiring modifications to the servo and data-channel design). It is conceivable that more embodiments of the DRPM proposal would appear in the marketplace.

6.2 Future Research Directions

Power management in enterprise storage systems, which was hitherto considered a challenge, has been shown to be tractable with the use of DRPM. This fine-granular speed-control opens up the possibility of performing other optimizations such as caching and prefetching, data-clustering and migration etc. to increase burstiness and also increase idleness to facilitate more energy savings. Application-specific knowledge can assist in the effective formulation of such strategies. Indeed, there has been some recent work on investigating such avenues of research [84, 117]. Mapping the specifications of a SLA to DRPM control-policy parameters, especially UT and LT, is an open-problem as well and can be combined with existing techniques that attempt to optimize energy in servers that have to adhere to SLAs [14]. The reliability impact of performing DRPM operations also warrants investigation, since they induce additional duty-cycles.

Another avenue of research involves devising and evaluating throttling strategies for dynamic thermal management in the context of real server workloads. This work would first require

the development of an evaluation framework that can provide detailed insight into both the performance and thermal behavior of the storage system when running a workload. In addition to throttling, it would also be interesting to investigate complementary drive design methodologies such as replacing a single monolithic disk with smaller drives and studying the tradeoffs of the increased throughput of multiple heads against the (potentially) higher latency of the smaller and lower-power disks. This would be somewhat analogous to using Simultaneous Multi-Threading (SMT) to mask long-latency events (like a cache-miss). Indeed, there has been some research in this direction by attempting to replace a server-disk with multiple laptop disks [79]. However, such an approach, though could demonstrate the potential of this technique, is not attractive for use in a real server system due to the significantly different reliability characteristics of laptop disks as compared to enterprise drives. There are also scheduling, interface-design, and disk-buffer management issues with such multi-spindle drives in addition to the thermal interactions between them (if they are to housed in close proximity such as within a drive enclosure). Another important research question is when a point of diminishing returns might set in with IDR scaling. In this thesis, a simple version of such an analysis was conducted to show that if a set of workloads were provided with faster disks, there would have been over 20%-50% boost in their I/O response times justifying continued progress in IDR scaling. However, such an "Amdahl's Law" type analysis would need to be conducted along the different years of the roadmap to reason about design effort (and consequent throttling strategies) from an application perspective. Even from a drive-design perspective, throttling-ratio analyses can provide an indicator for when pushing RPM would start becoming counter-productive such that, even with the use of all the throttling techniques available, the drive cannot be used to perform any meaningful operation due to the volume of heat generated. Finding these correct design points would be invaluable to directing storage-system research efforts in the future.

Finally, there are other disk-drive designs, especially in the domain of Micro Electro-Mechanical Systems (MEMS), that are actively under investigation. One such instance is the

"Millipede" project at IBM Zurich [26]. Such devices are expected to be the future of storage in the time-period beyond that given in this thesis. Even in such devices, where reading and writing is performed by heating the read/write heads, managing energy and temperature are again critical issues. Writes in the Millipede are performed by creating indentations on the polymer recording medium. The energy consumption for creating such indentations is a concern for such devices and may require innovative power-management methodologies. In order to perform a read operation, thermal sensing is performed by heating a resistive heater that is co-located with the head to over 200 C. Such an intense amount of heat would require careful design of the cooling system to localize the heating to the intended area, especially if such a device might be used in portable computing devices.

References

[1] A. Acharya, M. Uysal, and J.H. Saltz. Active Disks: Programming Model, Algorithms and Evaluation. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pages 81–91, October 1998.

[2] Adaptive Power Management for Mobile Hard Drives. Adaptive Power Management for Mobile Hard Drives. Technical report, IBM Corporation, Storage Systems Division, April 1999.

[3] G. Alvarez, K. Keeton, E. Riedel, and M. Uysal. Characterizing Data-Intensive Workloads on Modern Disk Arrays. In *Proceedings of the Workshop on Computer Architecture Evaluation Using Commercial Workloads*, January 2001.

[4] D. Anderson, J. Dykes, and E. Riedel. More Than An Interface - SCSI vs. ATA. In *Proceedings of the Annual Conference on File and Storage Technology (FAST)*, March 2003.

[5] K.G. Ashar. *Magnetic Disk Drive Technology: Heads, Media, Channel, Interfaces, and Integration*. IEEE Press, 1997.

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

[7] P. Bohrer, E.N. 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.

[8] G.E. Box and G.M. Jenkins. *Time Series Analysis Forecasting and Control*. Holden-Day, 2nd edition, 1976.

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

[10] S.H. Charrap, P.L. Lu, and Y. He. Thermal Stability of Recorded Information at High Densities. *IEEE Transactions on Magnetics*, 33(1):978–983, January 1997.

[11] 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.

[12] J.S. Chase, D.C. Anderson, P.N. Thakur, A.M. Vahdat, and R.P. 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.

[13] J. Chen and J. Moon. Detection Signal-to-Noise Ratio versus Bit Cell Aspect Ratio at High Areal Densities. *IEEE Transactions on Magnetics*, 37(3):1157–1167, May 2001.

[14] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam. Managing Server Energy and Operational Costs in Hosting Centers. In *Proceedings of the ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems (To Appear)*, June 2005.

[15] E. Christen and K. Bakalar. VHDL-AMS - A Hardware Description Language for Analog and Mixed-Signal Applications. *IEEE Transactions on Circuits and Systems-II: Analog and Digital Sig nal Processing*, 46(10):1263–1272, October 1999.

[16] N.S. Clauss. A Computational Model of the Thermal Expansion Within a Fixed Disk Drive Storage System. Master's thesis, University of California, Berkeley, 1988.

[17] D. Colarelli and D. Grunwald. Massive Arrays of Idle Disks for Storage Archives. In *Proceedings of the Conference on Supercomputing*, pages 1–11, November 2002.

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

[19] D. Dammers, P. Binet, G. Pelz, and L.M. Voßkämper. Motor Modeling Based on Physical Effect Models. In *Proceedings of the IEEE/ACM International Workshop on Behavioral Modeling and Simulation (BMAS)*, pages 78–83, October 2001.

[20] D. Damon and E. Christen. Introduction to VHDL-AMS - Part 1: Structural and Discrete Time Concep ts. In *Proceedings of the International Symposium on Computer-Aided Contr ol System Design*, pages 264–269, September 1996.

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

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

[23] P.A. Eibeck and D.J. Cohen. Modeling Thermal Characteristics of a Fixed Disk Drive. *IEEE Transactions on Components, Hybrids, and Manufacturing Technology*, 11(4):566–570, December 1988.

[24] E.N. Elnozahy, M. Kistler, and R. Rajamony. Energy-Efficient Server Clusters. In *Proceedings of the Workshop on Power-Aware Computer Systems (PACS'02)*, pages 124–133, February 2002.

[25] Enterprise Storage: A Look into the Future (TNM Seminar Series), September 2002.

[26] E. Eleftheriou et al. Millipede - A MEMS-Based Scanning-Probe Data-Storage System. *IEEE Transactions on Magnetics*, 39(2):938–945, March 2003.

[27] K. Flautner, S. Reinhardt, and T. Mudge. Automatic performance setting for dynamic voltage scaling. In *Proceedings of the International Conference on Mobile Computing and Networking (MOBICOM)*, pages 260–271, July 2001.

[28] G.R. Ganger. *System-Oriented Evaluation of I/O Subsystem Performance*. PhD thesis, The University of Michigan, June 1995.

[29] G.R. Ganger. *System-Oriented Evaluation of I/O Subsystem Performance*. PhD thesis, The University of Michigan, June 1995.

[30] G.R. Ganger, B.L. Worthington, and Y.N. Patt. *The DiskSim Simulation Environment Version 2.0 Reference Manual*, December 1999. http://www.ece.cmu.edu/ ganger/disksim/.

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

[32] J.L. Griffin, J. Schindler, S.W. Schlosser, J.S. Bucy, and G.R. Ganger. Timing-Accurate Storage Emulation. In *Proceedings of the Annual Conference on File and Storage Technology (FAST)*, January 2002.

[33] E. Grochowski. Hitachi GST, San Jose Research Center, 2004. Private Correspondence.

[34] E. Grochowski and R.D. Halem. Technological Impact of Magnetic Hard Disk Drives on Storage Systems. *IBM Systems Journal*, 42(2):338–346, 2003.

[35] D. Grunwald, P. Levis, K. Farkas, C. Morrey III, , and M. Neufeld. Policies for dynamic clock scheduling. In *Proceedings of the Symposium on Operating Systems Design and Implementation (OSDI)*, pages 73–86, October 2000.

[36] 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.

[37] S. Gurumurthi, A. Sivasubramaniam, and V.K. Natarajan. Disk Drive Roadmap from the Thermal Perspective: A Case for Dynamic Thermal Management. Technical Report CSE-05-001, The Pennsylvania State University, February 2005.

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

[39] S. Gurumurthi, J. Zhang, A. Sivasubramaniam, M. Kandemir, H. Franke, N. Vijaykrishnan, and M.J. 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.

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

[41] Heat Density Trends in Data Processing, Computer Systems, and Telecommunications Equipment White Paper, 2000.

[42] T. Heath, E. Pinheiro, and R. Bianchini. Application-Supported Device Management for Energy and Performance. In *Proceedings of the Workshop on Power-Aware Computer Systems (PACS'02)*, pages 114–123, February 2002.

[43] 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.

[44] G. Herbst. IBM's Drive Temperature Indicator Processor (Drive-TIP) Helps Ensure High Drive Reliability. In *IBM Whitepaper*, October 1997.

[45] S. Hetzler. IBM Almaden Research Center, 2004. Private Correspondence.

[46] Hitachi Global Storage Technologies - HDD Technology Overview Charts. http://www.hitachigst.com/hdd/technolo/overview/storagetechchart.html.

[47] Hitachi Global Storage Technologies - Research and Technology Overview. http://www.hitachigst.com/hdd/research/storage/pm/.

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

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

[50] HP News Release. HP Announces "Smart" Cooling Solution for Data Centers, March 4 2003. http://www.hp.com/hpinfo/newsroom/press/2003/030304b.html.

[51] Y. Hu and Q. Yang. DCD Disk Caching Disk: A New Approach for Boosting I/O Performance. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*, pages 169–178, May 1996.

[52] R.F. Huang and D.L. Chung. Thermal Design of a Disk-Array System. In *Proceedings of the InterSociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, pages 106–112, May 2002.

[53] G.F. Hughes. Center for Magnetic Recording Research, University of California, San Diego, 2004. Private Correspondence.

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

[55] IBM DB2. http://www-3.ibm.com/software/data/db2/.

[56] IBM Hard Disk Drive - Travelstar 40GNX. http://www.storage.ibm.com/ hdd/ travel/ tr40gnx.htm.

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

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

[59] IBM Predictive Failure Analysis. http://www.storage.ibm.com/hdd/ipl/oem/tech/pfa.htm.

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

[61] R. Jain. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. John Wiley and Sons Inc., 1991.

[62] H.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.

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

[64] 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.

[65] H. Levy and F. Lessman. *Finite Difference Equations*. Dover Publications, 1992.

[66] D. Li and J. Wang. EERAID: Energy-efficient Redundant And Inexpensive Disk Array. In *Proceedings of the SIGOPS European Workshop*, September 2004.

[67] Kester Li, Roger Kumpf, Paul Horton, and Thomas E. Anderson. Quantitative Analysis of Disk Drive Power Management in Portable Computers. In *Proceedings of the USENIX Winter Conference*, pages 279–291, 1994.

[68] 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*, pages 271–283, October 2004.

[69] X. Liu, P. Shenoy, and W. Gong. A Time Series-based Approach for Power Management in Mobile Processors and Disks. In *Proceedings of the Workshop on Network and Operating System Support for Audio and Video (NOSSDAV)*, pages 74–81, June 2004.

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

[71] Y-H. Lu and G.D. Micheli. Adaptive Hard Disk Power Management on Personal Computers. In *Proceedings of the IEEE Great Lakes Symposium*, March 1999.

[72] M. Mallary, A. Torabi, and M. Benakli. One Terabit Per Square Inch Perpendicular Recording Conceptual Design. *IEEE Transactions on Magnetics*, 38(4):1719–1724, July 2002.

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

[74] A. Moshovos, G. Memik, B. Falsafi, and A.N. Choudhary. JETTY: Filtering Snoops for Reduced Energy Consumption in SMP Servers. In *Proceedings of the 7th International Symposium on High-Performance Computer Architecture (HPCA)*, January 2001.

[75] V. Natarajan, S. Gurumurthi, and A. Sivasubramaniam. Is Traditional Power Management+Prefetching == DRPM for Server Disks. In *Proceedings of the Workshop on Computer Architecture Evaluation Using Commercial Workloads (CAECW)*, February 2005.

[76] 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.

[77] H.H. Ottesen and G.J. Smith. Servo Format for Disk Drive Data Storage Devices. In *United States Patent 6,775,081*, August 2001.

[78] A. E. Papathanasiou and M. L. Scott. Energy Efficient Prefetching and Caching. In *Proceedings of the USENIX Annual Technical Conference*, June 2004.

[79] A.E. Papathanasiou and M.L. Scott. Power-Efficient Server-Class Performance from Arrays of Laptop Disks. In *USENIX Annual Technical Conference (Work In Progress)*, June 2004.

[80] 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.

[81] 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, June 1988.

[82] R.H. Patterson, G.A. Gibson, E. Ginting, D. Stodolsky, and J. Zelenka. Informed Prefetching and Caching. In *Proceedings of the ACM Symposium on Operating Systems Principles (SOSP)*, pages 79–95, December 1995.

[83] 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.

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

[85] E. Pinheiro, R. Bianchini, E. V. Carrera, and T. Heath. Load Balancing and Unbalancing for Power and Performance in Cluster-Based Systems. In *Proceedings of the Workshop on Compilers and Operating Systems for Low Power*, September 2001.

[86] A. Popescu. Traffic Self-Similarity. In *IEEE International Conference on Telecommunications (ICT) Tutorial*, June 2001.

[87] A.L. Poplawski and D.M. Nicol. An Investigation of Out-of-Core Parallel Discrete-Event Simulation. In *Proceedings of the Winter Simulation Conference*, pages 524–530, December 1999.

[88] Power, Heat, and Sledgehammer, 2002. http://www.max-t.com/downloads/whitepapers/SledgehammerPowerHeat20411.pdf.

[89] R. Rao and S. Vrudhula. Energy Optimization for a Two-Device Data Flow Chain. In *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*, November 2004.

[90] R. Rao, S. Vrudhula, and M. Krishnan. Disk Drive Energy Optimization for Audio-Video Applications. In *Proceedings of the Conference on Compilers, Architectures and Synthesis for Embedded Systems (CASES)*, pages 93–103, September 2004.

[91] I.S. Reed and G. Solomon. Polynomial Codes Over Certain Finite Fields. *Journal of the Society for Industrial and Applied Mathematics*, 8:300–304, June 1960.

[92] E. Riedel. Device Trends - Where Disk Drives are Headed. In *Information Storage Industy Consortium (INSIC) Workshop on the Future of Data Storage Devices and Systems (DS2)*, April 2004.

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

[94] M. Rosenblum and J.K. Ousterhout. The Design and Implementation of a Log-Structured File System. *ACM Transactions on Computer Systems*, 10(1):26–52, 1992.

[95] C. Ruemmler and J. Wilkes. Disk Shuffling. Technical Report HPL-91-156, HP Laboratories, October 1991.

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

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

[98] Seagate Cheetah 15K.3 SCSI Disc Drive: ST3734553LW/LC Product Manual, Volume 1. http://www.seagate.com/support/disc/manuals/scsi/100148123b.pdf.

[99] K. Skadron, M.R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. Temperature-Aware Microarchitecture. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*, pages 1–13, June 2003.

[100] M. Sri-Jayantha. Trends in Mobile Storage Design. In *Proceedings of the International Symposium on Low Power Electronics*, pages 54–57, October 1995.

[101] Storage Review - Reference Guide. http://storagereview.com/guide2000/ref/hdd/op/formIn25.html.

[102] G.J. Tarnopolsky. Hard Disk Drive Capacity at High Magnetic Areal Density. *IEEE Transactions on Magnetics*, 40(1):301–306, January 2004.

[103] TPC-C Benchmark V5. http://www.tpc.org/tpcc/.

[104] TPC-H Benchmark. http://www.tpc.org/tpch/.

[105] N.N. Tran. *Automatic ARIMA Time Series Modeling and Forecasting for Adaptive Input/Output Prefetching*. PhD thesis, University of Illinois at Urbana-Champaign, 2002.

[106] Transaction Processing Performance Council. http://www.tpc.org/.

[107] W. Tschudi. NY Data Center No. 2 - Energy Benchmarking and Case Study. Technical report, Ernest Orlando Lawrence Berkeley National Laboratory, July 2003.

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

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

[110] R. Viswanath, V. Wakharkar, A. Watwe, and V. Lebonheur. Thermal Performance Challenges from Silicon to Systems. *Intel Technology Journal*, Q3 2000.

[111] L.M. Voßkämper, R. Schmid, and G. Pelz. Combining Models of Physical Effects for Describing Complex Electromechanical Devices. In *Proceedings of the IEEE/ACM International Workshop on Behavioral Modeling and Simulation (BMAS)*, pages 42–45, October 2000.

[112] R. Wood. The Feasibility of Magnetic Recording at 1 Terabit per Square Inch. *IEEE Transactions on Magnetics*, 36(1):36–42, January 2000.

[113] B. Worthington, G. Ganger, Y. Patt, and J. Wilkes. On-Line Extraction of SCSI Disk Drive Parameters. In *Proceedings of the ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, pages 146–156, May 1995.

[114] 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.

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

[116] J. Zedlewski, S. Sobti, N. Garg, F. Zheng, A. Krishnamurthy, and R. Wang. Modeling Hard-Disk Power Consumption. In *Proceedings of the Annual Conference on File and Storage Technology (FAST)*, March 2003.

[117] Q. Zhu, F.M. David, C. Devraj, Z. Li, Y. Zhou, and P. Cao. Reducing Energy Consumption of Disk Storage Using Power-Aware Cache Management. In *Proceedings of the International Symposium on High-Performance Computer Architecture (HPCA)*, Febuary 2004.

[118] Q. Zhu, A. Shankar, and Y. Zhou. Pb-lru: A self-tuning power aware storage cache replacement algorithm for conserving disk energy. In *Proceedings of the International Conference on Supercomputing (ICS)*, pages 79–88, 2004.

Vita

Sudhanva was born and raised in the coastal city of Madras (now Chennai) in the south-east of India. After his fifth grade, he moved to the United States and lived in Falls Church, VA and stayed there until the ninth grade. He registered for the Bachelors program in Computer Science and Engineering at the College of Engineering Guindy in Anna University, Chennai. He then went to Penn State for graduate school in the Fall of 2000, for which he has a special fondness, since his father himself is a PSU alumnus. Sudhanva spent three months as an intern in the Power Aware Systems group at IBM Research at Austin, TX. and six months in Hudson, MA, in the Fault Aware Computing Technology group at Intel.

Sudhanva's research interests include computer architecture, storage systems, and fault-tolerance. He is a member of the ACM.