Intra-Disk Parallelism: An Idea Whose Time Has Come

Sriram Sankar[†] Sudhanva Gurumurthi[†] Mircea R. Stan[‡]

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

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

Technical Report CS-2008-03
February 2008

Abstract

Power is a big problem in data centers and a significant fraction of this power is consumed by the storage system. Server storage systems use a large number of disks to achieve high performance, which increases their power consumption. In this paper, we propose to significantly reduce the power consumed by the storage system via intra-disk parallelism, wherein disk drives can exploit parallelism in the I/O request stream. Intra-disk parallelism can facilitate replacing a large disk array with a smaller one, using the minimum number of disk drives needed to satisfy the capacity requirements. We show that the design space of intra-disk parallelism is large and present a taxonomy to formulate specific implementations within this space. Using a set of commercial workloads, we perform a limit study to identify the key performance bottlenecks that arise when we replace a storage array that is tuned to provide high performance with a single high-capacity disk drive. These are the bottlenecks that intra-disk parallelism would need to alleviate. We then explore a particular intra-disk parallelism approach, where a disk is equipped with multiple arm assemblies that can be independently controlled, and evaluate three disk drive designs that embody this form of parallelism. We show that it is possible to match, and even surpass, the performance of a storage array for these workloads by using a single disk drive of sufficient capacity that exploits intra-disk parallelism, while significantly reducing the power consumed by the storage system compared to the multi-disk configuration. We evaluate the performance and power consumption of disk arrays composed of intra-disk parallel drives, discuss the engineering issues involved in implementing such drives, and finally provide a preliminary cost-benefit analysis of building and deploying intra-disk parallel drives, using cost data obtained from several companies in the disk drive industry.

1 Introduction

Storage is a large power consumer in data centers. Server storage systems provide the data storage and access requirements of a variety of applications, such as, On-Line Transaction Processing (OLTP), On-Line Analytical Processing (OLAP), and Internet search engines. Given the I/O intensive nature of these workloads and the fact that there are usually several users who access the system concurrently, server storage systems need to be capable of delivering very high I/O throughput. This performance goal is achieved is by using a large number of disks and distributing the dataset of the application over the multiple drives, typically using RAID [32]. However, the result

of using multiple disk drives is that server storage systems consume a large amount of power [14, 6, 50], and disk drive power consumption constitutes over 13% of the Total Cost of Ownership of a data center [50].

The main motivation for using multiple disks for these applications is to increase I/O throughput, and not capacity, as most vendors recommend using multiple disk drives for purely performance reasons [41, 23, 4, 11]. (Multiple disks are also used to provide reliability, which we discuss in Section 4.1). Moreover, another common practice to boost performance is to use only a fraction of the space within a drive in order to leverage the higher data rates experienced at the outer tracks of a platter [2]. On the other hand, the per-disk capacity has been growing rapidly over the years, and disks with over a Terabyte of capacity are already available in the market, e.g., Hitachi Deskstar 7K1000 [20]. However, the performance of a single disk drive has been improving at a much lower rate, partly due to certain limitations in magnetic recording technology [7] and also due to thermal constraints on scaling rotational speeds [15]. As a result, server storage systems end up using a large number of disk drives to get high performance. Although industry predicts that capacity will continue to grow briskly, with 1 Terabit/inches^2 of areal density expected by the year 2013, which will allow several Terabytes of data to be stored in one disk drive, future drives are not expected to have faster rotational speeds nor significantly lower seek times [26]. Therefore, future server storage systems would still need to employ multiple disk drives to meet performance goals and the storage system will continue to be a large power consumer.

In this paper, we ask the following question: *Is it possible to design a storage system where we use the minimal set of disks, purely for satisfying capacity requirements, and still achieve the performance of a system designed for high performance?* By having fewer disks, we can reduce the total power of the storage system. However, using fewer disks can create I/O bottlenecks and lead to performance degradation. In order to bridge this performance gap, but still maintain low power consumption, we propose the use of *intra-disk parallelism*, i.e., disk drives that can exploit parallelism in the I/O request stream. Unlike traditional approaches to disk power management, where power management "knobs" are added to conventional disks [29, 14], we explore how extending the design of a disk drive to exploit parallelism can enable the storage system to be more *power efficient*. Towards this end, this paper makes the following contributions:

- We first provide a historical retrospective on intra-disk parallelism. We discuss about the multi-actuator drives that were used in mainframes back in the 1970s and 80s, why they were discontinued, and show why our intra-disk parallelism idea is different.

- We present a taxonomy for intra-disk parallelism, identifying the locations within a disk drive where parallelism can be incorporated, and discuss various design options within this space.

- We conduct a detailed limit study using a set of commercial server workloads to identify the key performance bottlenecks that intra-disk parallelism would need to alleviate, when we replace a storage array that is tuned to provide high performance with a single high-capacity disk drive. We find that rotational latency is the primary bottleneck that intra-disk parallelism needs to optimize.

- We present an intra-disk parallel design, which involves the use of multiple disk arm assemblies, and evaluate three implementations of this design. We show that even the simplest intra-disk parallel design can facilitate breaking-even with, or even surpassing the performance of a storage array, while consuming significantly less power than the multi-disk configuration.

- We explore how the average power consumption of intra-disk parallel drives can be made comparable to that of conventional hard disk drives by designing them to operate at a lower RPM. In some cases, we find that the parallel disk drive can provide higher performance than its corresponding multi-disk system, while consuming lower power than the single, conventional, higher RPM disk drive.

- We compare the performance and power characteristics of RAID arrays built using intra-disk parallel drives to those composed of only conventional disk drives that use the same recording technology and share the same architectural characteristics. We show that arrays built using intra-disk parallel drives provide the same or even better performance than those using conventional drives, while consuming 41%-60% lower power across a range of I/O intensities.

- We discuss the engineering issues that need to be addressed when building an intra-disk parallel drive and point to existing solutions to address these issues.

- We perform a preliminary cost-benefit analysis of building and deploying intra-disk parallel drives, using real data obtained from several companies in the disk drive industry. We show that intra-disk parallelism holds promise from the cost viewpoint as well.

The outline for the rest of the paper is as follows. The next section presents an overview of disk drives and introduces the intra-disk parallelism idea. Section 3 gives a historical retrospective on intra-disk parallel drives. In Section 4 we provide a taxonomy for intra-disk parallelism and Section 5 discusses the related work. Details about our workloads and evaluation infrastructure are given in Section 6 and Section 7 gives the experimental results. The engineering issues are discussed in Section 8 and the cost analysis is presented in Section 9. Section 10 concludes this paper.

2 Basics of Disk Drives and Intra-Disk Parallelism

A hard disk drive is composed of one or more platters that are stacked on top of each other and are held in place by a central spindle. Both surfaces of each platter are coated by a layer of magnetic material, which forms the recording medium. The data on the media are organized into sectors and tracks. The platter stack is rotated at a high speed at a certain Rotations Per Minute (RPM) by a *spindle motor* (*SPM*). Data is read from or written to the magnetic medium via read/write heads, which are mounted on sliders and float over the surface of the platters in a very thin cushion of air. The sliders are held in place by disk arms, which are connected to a central assembly. All the arms in the assembly are moved in unison by a single *voice-coil motor* (*VCM*). (The arm assembly is sometimes referred to as the "actuator". We shall use the terms "arm assembly" and "actuator" interchangeably in this paper). In addition to these electro-mechanical components, disks also have several electronic circuitry, such as, the disk controller, data channel, motor drivers, and an on-board cache.

At runtime, there are two structurally independent sets of electro-mechanical activities that occur within a disk drive: (i) the radial movement of the head across the surface of the disk (driven by the VCM), and (ii) the rotation of the platters under the head (driven by the SPM). These two sets of moving subsystems affect two different components of the total disk access time: (i) *seek time* - the time required to move the head to the desired track, and (ii) *rotational latency* - the time taken for the appropriate sector to rotate under the head. In addition to these two latencies, the disk access time also includes the actual time required to transfer the data between the platters and the

drive electronics. In workloads that exhibit random I/O and perform relatively small data transfers, as is the case for many server workloads [22], the latencies for the mechanical positioning activities dominate the disk access time.

Rationale Behind Intra-Disk Parallelism: In a conventional disk drive, only a single I/O request can be serviced at a time. For any given disk request that requires accessing the platters (i.e., cannot be serviced from the disk cache), the access time of the request is *serialized* through the seek, rotational latency, and data transfer phases. That is, although the arm and spindle assemblies are physically independent electro-mechanical systems, they are used in a tightly coupled manner due to the way that disk accesses are performed. Furthermore, all the resources within each electro-mechanical system of the drive are "locked up" for each I/O request. For example, all the individual arms within the arm assembly move in unison on a disk seek for an I/O request, although only one of the heads on a particular arm will actually service the request.

We propose to extend this conventional disk drive design to provide *intra-disk parallelism* by: (i) decoupling how the two electro-mechanical systems are used to service I/O requests, so that we can overlap seek time and rotational latency, either for one I/O request or across multiple requests, and (ii) decoupling the multiplicity of components *within* each of the electro-mechanical systems, e.g., the heads on an arm assembly. In order to achieve parallelism using either approach, we need additional hardware support.

3 Intra-Disk Parallelism - Historical Retrospective and Motivation

Multi-actuator disk drives used to exist in the market in the 1970s and 80s, and papers were published that explored the use of such disks in mainframes. A dual arm assembly design, where one arm was capable of motion while the other remained stationary was implemented in the IBM 3340 disk drive, which was used in the IBM System/370 mainframe [18]. A later work [42] explored the possibility of having multiple arms that are capable of moving independently, and the IBM 3380, which was a 4-actuator drive released in 1980 for the IBM System/370, embodied this feature. Spencer Ng's study [31], based on the IBM 3380 drive architecture, motivated the use of multi-actuator disks to reduce rotational latencies. Despite all these products and research, multi-actuator drives do not exist in the market anymore. Instead of using parallel disk drives, we build RAID arrays using multiple single-actuator disk drives.

Therefore, before we discuss intra-disk parallelism, it is first important to understand why multi-actuator drives were discontinued and why intra-disk parallelism, in the context of modern disk drives, is different.

Disk Drive Characteristics	**Disks From SIGMOD'88 RAID Paper [32]**			**Modern Disk Drive Technology**	
	IBM 3380 AK4	Fujitsu M2361A	Conners CP3100	Seagate Barracuda ES	**Projection for 4-Actuator Intra-Disk Parallel Drive**
Areal Density (Mb/in^2)	12			128000	
Disk Diameter (inches)	14	10.5	3.5	3.7	3.7
Formatted Data Capacity (MB)	7,500	600	100	750,000	750,000
No. Actuators	4	1	1	1	**4**
Power/box (Watts)	6,600	640	10	13	**34**
Transfer Rate (MB/s)	3	2.5	1	72	***Explored Section 7***
Price/MB (including controller)	$18-$10	$20-$17	$10-$7	$0.00042-$0.00034	***Explored in Section 9***

Table 1: Comparison of disk drive technologies over time. The Seagate Barracuda ES disk drive is a state-of-the-art SATA disk drive. The rightmost column presents the projected characteristics of a 4-actuator intra-disk parallel drive that extends the Barracuda design. The performance and costs aspects are explored in this paper.

Table 1 gives the characteristics of five disk drives along several axes. The first four disk drives are actual

products that have appeared in the market and the fifth is a hypothetical intra-disk parallel drive. The disks listed in the first three columns of the table and their characteristics are from the 1988 SIGMOD paper by Patterson, Gibson, and Katz that introduced RAID [32]. The IBM 3380 AK4 (described earlier), the Fujitsu M2361A, and the Conners CP3100 are mainframe, minicomputer, and personal computer drives respectively and were state-of-the-art products of their time. The areal density information about disk drives during this time period was obtained from [44]. The fourth disk drive - the Seagate Barracuda ES - is a state-of-the-art SATA disk drive that is representative of disk drives available in the market today. The technical specifications of this disk drive (including the areal density information) were obtained from the manufacturer datasheets [40]. The price per Megabyte was calculated based on data that we obtained about the Barracuda from retail websites, such as `buy.com` and `pricegrabber.com`. The specifications in the last column of this table are for a hypothetical intra-disk parallel drive that extends the Barracuda architecture to include four independent actuators. The power consumption for this drive is calculated assuming that all four VCMs are active and all the arm assemblies are moving, which represents the peak power consumption scenario for this design. The power consumption is calculated using detailed disk power models equivalent to those given in [49]. As a simple validation test, we calculated the difference between the seek and idle power for this drive (thereby factoring out the SPM power), which we obtained from the manufacturer datasheet [40], and compared it to the VCM power obtained from the power models. We found the VCM power values calculated using these two methods to be very close. (*NOTE:* This power number for the intra-disk parallel drive is an approximation and is merely meant to facilitate the high-level discussion in this section. We perform more detailed power modeling and analysis of intra-disk parallel drives later in this paper).

Let us first look at the three disk drives that are discussed in the RAID paper [32]. The IBM 3380 used 14-inch platters. Since the platter size has a fifth-power impact on the power consumption of a disk drive [24], the spindle assembly of this drive consumed a very large amount of power. Moreover, larger platters require more powerful VCMs, and this disk had 4 actuators. As a result, the IBM 3380 consumed a massive 6,600 Watts of power. Even the Fujitsu M2361A drive, which had only one actuator, but a large 10.5-inch platter consumed 640 Watts of power. On the other hand, the Conner CP3100 had a much smaller platter size (3.5 inches) and therefore consumed only 10 Watts. Although the high-end drives provided higher capacity than a single personal computer drive, their price per Megabyte was in the $10-$20 range, compared to $7-$10 for the CP3100. Therefore, the high-end drives were much more expensive than the smaller drive, their power consumption was one to two orders of magnitude higher, and provided only moderately faster transfer rates than the CP3100. Therefore, as the RAID paper pointed out, using multiple CP3100 drives allowed one to surpass the performance of the IBM 3380 while consuming an order of magnitude less power than the mainframe drive. RAID was a clear winner and the high-end multi-actuator drives soon disappeared from the market.

When we fast-forward to the modern era, the first thing that we observe is that the areal density has improved over *four orders of magnitude*, largely due to Giant Magneto-Resistive head technology. This technological breakthrough has lead to a huge drop in the price per Megabyte of storage. Although higher densities have boosted disk transfer rates as well, by close to two orders of magnitude, disk performance is still limited by delays in the electro-mechanical system. Compared to performance improvements in microprocessors over the same time period, disk drives have woefully lagged behind and the speed gap between processors and disks has widened significantly. This speed gap has been one of the main reasons why RAID-based storage systems are used in servers that run I/O intensive applications.

When we examine the internal organization of the CP3100 and Barracuda drives, we can see that both have 4

platters and that their platter sizes are approximately the same. However, the CP3100 was a 3575 RPM drive [9] whereas the Barracuda operates at 7200 RPM. Since the power consumption of a disk drive is proportional to the fifth-power of the platter size, is cubic with the RPM, and is linear with the number of platters [24, 15], the power consumption of the CP3100 and the Barracuda are close, but the CP3100 consumes slightly less power than the Barracuda. However, when compared to the IBM 3380, the Seagate Barracuda provides *two orders of magnitude higher capacity*, consumes *two orders of magnitude less power*, and *costs three orders of magnitude less* than the old mainframe drive.

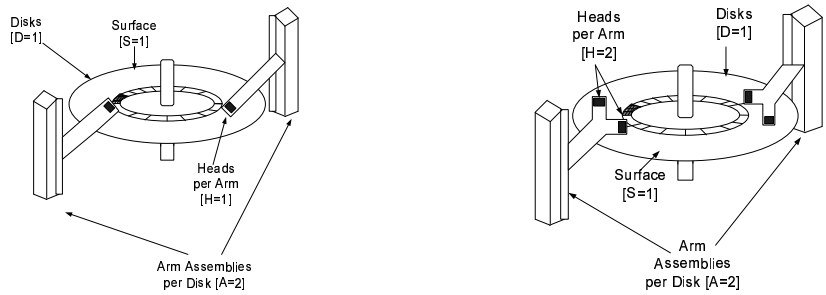
Now consider the hypothetical 4-actuator intra-disk parallel drive given in the last column of the table, which extends the Barracuda's architecture. Since this parallel drive has 4 actuators, all of which could be in motion simultaneously, its worst-case power consumption will be higher than the Barracuda. Using the power models described previously, we find the power consumption of the intra-disk parallel drive to be 34 Watts. Although 34 Watts is still significant and it is desirable to reduce the power consumption, the key insight here is that since this 4-actuator drive is an extension of a *modern* disk drive, which uses relatively small platter sizes, arms, etc., its power consumption is much lower than the large IBM 3380 disk drive - *two orders of magnitude lower* - and the power consumption is within 3X that of the conventional drive. Given this reversal in the power consumption trends from the past, and with all the other advancements in the disk drive design and manufacturing processes and the importance of the storage power problem in servers and data centers [6, 14, 50], there is a strong incentive to re-examine whether parallel disk drive architectures are beneficial in building high-performance, energy-efficient storage systems.

4 The $DASH$ Parallel Disk Taxonomy

Multi-actuator drives are a single design point within the space of intra-disk parallelism. Since the design space of intra-disk parallelism is large, it is desirable to have a taxonomy for systematically formulating specific designs within this space. We have developed one such taxonomy. In this taxonomy, a specific disk configuration is expressed hierarchically as a 4-tuple: $\mathbf{D_k A_l S_m H_n}$, where, k, l, m, and n indicate the degree of parallelism in four of the possible electro-mechanical components in which parallelism can be incorporated, starting from the most coarse-grained to the most fine-grained component - the **D**isk stack, **A**rm assembly, **S**urface, and **H**ead. For example, a conventional disk has the configuration $D_1A_1S_1H_1$, which indicates that there is a single disk stack that is accessed by one set of arms, and data is accessed one surface at a time using a single head per surface. This design provides a single data transfer path between the disk drive and the rest of the system. Figure 1(a) shows the physical design of a $D_1A_2S_1H_1$ configuration, which is a 2-actuator drive that can provide a maximum of two data transfer paths to/from the drive. Figure 1(b) shows a $D_1A_2S_1H_2$ configuration, which consists of two arm assemblies and with two heads on each arm that can access a single surface, thereby providing a maximum of four possible data transfer paths to/from the disk drive. We now discuss each of these parallelism dimensions in more detail.

- **Level 1: Disk Stacks [D]**

  We can have multiple disk stacks, each with its own spindle, which is precisely the form of parallelism that RAID provides. However, this form of parallelism can be incorporated even within a single disk drive, by shrinking the platter size. Since the power dissipated by the spindle assembly is strongly influenced by the platter size (approximately 4.6^{th} power of the platter size [24]), shrinking the platters can facilitate incorporating multiple disk stacks within the power envelope of a single disk drive. In fact, there has been previous work



(a) A $D_1A_2S_1H_1$ disk drive. (b) A $D_1A_2S_1H_2$ disk drive.

Figure 1: Example design points within the $DASH$ intra-disk parallelism taxonomy.

that explores the possibility of replacing a laptop disk drive with a small RAID array composed of smaller diameter disks [48].

- **Level 2: Arm Assemblies [A]**

  The number of actuators could be varied for each disk to provide parallelism. Providing parallelism along this dimension can be used to minimize seek time and rotational latency. The variables in this dimension are: the number of arm assemblies and the placement of these assemblies within the drive.

- **Level 3: Surfaces [S]**

  The two surfaces on each platter could be independently accessed. Parallelism across surfaces can be implemented by having heads on multiple arms within a single assembly accessing data on various surfaces, or by having heads on arms mounted on different assemblies (this design requires parallelism along the A-dimension as well). Given the high track-density on modern disks, achieving deterministic alignment of heads on multiple arms that are on a single assembly is very challenging from the engineering perspective. This makes the first approach to surface-level parallelism difficult to implement, although having fewer arm assemblies could provide power benefits.

- **Level 4: Heads [H]**

  Conventional disk drives have only a single head per surface on each arm, but this assumption could be relaxed. There are two possibilities for such a design, based on where we place the heads on the arm: (a) on a radial line on the arm, from the axis of actuation, or (b) equidistant from the axis of actuation (which is illustrated in Figure 1(b)). There are two design variables in this level of this taxonomy: the distance between each head and the number of heads per arm.

There are two issues about this taxonomy that are worth noting:

- For a given point in the taxonomy, a variety of physical implementations are possible. For example, in a disk that has two arm assemblies (i.e., $A = 2$), we may have one arm that is capable of motion while the other is stationary, or both that are capable of motion at the same time. The actual choice depends on tradeoffs between design and manufacturing costs, power/thermal constraints, and the expected benefits for the applications for which the product is intended.

- The taxonomy deals only with parallelism in the electro-mechanical subsystem of the disk drive and not the electronic data channel. If a disk drive provides multiple data transfer paths (for example, a drive with $A = 2$ might allow both arms to transfer data), then the data channel of the drive must have sufficient bandwidth to transport this data to gain maximum performance benefit. In general, we assume that the data channel provides sufficient bandwidth to transport the bits between the platters and the on-board electronics for all the disk configurations that we consider. We plan to study data channel issues in more depth in our future work.

4.1 Intra-Disk Parallelism and RAID

Intra-disk parallelism is *not* a replacement for RAID. RAID is used for boosting I/O throughput and also for reliability. Although intra-disk parallelism addresses the former issue, multiple parallel disk drives may still be needed for certain I/O intensive workloads to achieve high performance. We evaluate RAID arrays that are built using intra-disk parallel drives in Section 7.3. RAID would also be needed from a reliability viewpoint, since the failure of an intra-disk parallel drive can have adverse consequences, and the system designer would have to provision as many parallel drives as necessary to meet her storage system reliability requirements.

5 Related Work

Disk Power Management: In order to boost I/O performance, server storage systems use a combination of faster disks to reduce latency and a large number of disks to improve bandwidth. However, this approach leads to significant increases in data center power and cooling costs [27] and has motivated research into power management of server storage. To manage power in high-throughput server storage systems, the use of multi-RPM disk drives has been proposed [6, 14] and such disks are now commercially available [46, 21]. Researchers have also explored how multi-RPM disks can be used in conjunction with data clustering techniques [33] and storage cache management strategies [51]. A number of other techniques have been proposed for building energy efficient server storage systems, such as, MAID [8], which uses cache disks for concentrated access to a specific set of disks while keeping others in the spun down state, and diverted accesses techniques [34].

Solid-State Disks: Another interesting approach to building low power storage systems is to use solid-state disks. Flash memory is already used in a variety of consumer electronic products and has become popular for mobile storage. Another possibility is to use MEMS based storage [5], which holds great promise for providing faster response times and significantly lower power consumption than conventional disk drives. However, from an economic perspective, the cost per megabyte for flash and MEMS remain orders of magnitude higher than hard disk drives [39]. According to a recent study by the IDC [38], hard disk drives will remain the dominant storage technology for at least another decade, and therefore it is important to develop extensions to conventional disk drive architectures to meet performance goals and reduce power. However, we believe that there are opportunities for using solid-state disks in conjunction with techniques that we discuss in this paper, and we plan to investigate these possibilities in future work.

Freeblock Scheduling: Finally, an alternative approach to overlapping multiple I/O requests inside a conventional disk drive is to use freeblock scheduling [30]. In freeblock scheduling, the rotational latency periods of foreground I/O requests are used to service I/O requests of background tasks. Intra-disk parallelism can provide the same functionality as freeblock scheduling by utilizing independent hardware components for servicing foreground and

background I/O requests. However, freeblock scheduling in a conventional drive is restricted by the fact that the I/O accesses for the background process(es) need to be serviced within a tight deadline i.e., before the rotational latency period of a foreground request completes. This places restrictions on the type of tasks for which freeblock scheduling can be applied, and number of I/O requests that can be serviced before the deadline.

6 Experimental Setup and Workloads

Workload	Number of Requests	Number of Disks	Disk Capacity (GB)	RPM	Number of Platters
Financial	5,334,945	24	19.07	10000	4
Websearch	4,579,809	6	19.07	10000	4
TPC-C	6,155,547	4	37.17	10000	4
TPC-H	4,228,725	15	35.96	7200	6

Table 2: Workloads and the configuration of the original storage systems on which the traces were collected.

Our experiments are carried out using the Disksim simulator [12], which models the performance of disks, caches, storage interconnects, and multi-disk organizations in detail, and has been validated against several real disk drives. We augmented Disksim with power models for the spindle and arm assemblies that we developed in our prior work [49]. These power models are based on the fundamental physical and electrical characteristics of the two electro-mechanical systems of the disk drive.

We use a set of commercial server I/O traces as our workload suite. Information about these traces and the original storage systems on which they were collected are given in Table 2. Financial and Websearch are I/O traces collected at a large financial institution and at a popular Internet search-engine respectively [45]. The TPC-C trace was collected on a 2-way SMP machine running the IBM DB2 EEE database engine. The TPC-C benchmark was run for a 20-warehouse configuration with 8 clients. The TPC-H trace was collected on an 8-way IBM Netfinity SMP machine with 15 disks and running the IBM DB2 EE edition. The TPC-H benchmark was run in the power test mode, in which the 22 queries of the benchmark are executed consecutively.

6.1 Metrics

In our evaluations we use two main metrics: *response time* and *average power*. These metrics are defined as follows:

- **Response Time:** The response time is the average time between the submission and the completion of an I/O request presented to the storage system and is expressed in milliseconds. The response time has a direct impact on the throughput of the storage system and is our primary performance metric. In most of our results, we present the response time characteristics of the storage system using Cumulative Distribution Functions (CDF) rather than as averages. A CDF graph expresses the fraction of I/O requests whose response times are less than or equal to a given value on the x-axis. A CDF allows us to visualize the scenario where a large number of I/O requests may be experiencing relatively short response times whereas a few other requests may have very long response times. A storage system that is experiencing heavy bottlenecks will have a CDF curve that is skewed towards numerically higher response time buckets, which indicates that the storage system is unable to service I/O requests fast enough. Although a real system would handle such an overload condition at a higher level, for example, by dropping connections to the server, we do not attempt to modulate the arrival rate of the I/O requests to the storage system in this study. Instead, our goal is to design the storage system so that it can efficiently service I/O requests as they arrive.

- **Average Power Consumption:** The average power consumption is the total energy consumed from the beginning to the end of the simulation period divided by the duration of that period.

7 Results

We conduct three sets of experiments. The first is a limit study to determine the performance and power ramifications of replacing a multi-disk storage array with a single high-capacity disk drive. The objective of this experiment is to determine the power benefits of such a system migration and the performance gap between the performance-optimized storage array and the single disk drive configuration, and the bottlenecks that lead to this gap. Based on these results, we formulate three intra-disk parallel designs, which progressively extend the conventional disk drive architecture. In the second set of experiments, we evaluate the performance and power of these intra-disk parallel designs. The third set of experiments use synthetic workloads to evaluate the performance and power characteristics of RAID arrays that are built using intra-disk parallel drives and compare them to arrays that are composed of conventional drives that use the same underlying recording technology and share common architectural characteristics, such as, platter sizes, RPM, and disk cache capacity, with the parallel drives.

7.1 Performance and Power Limit Study

The main reason that server storage systems use multiple disks is to boost performance [2, 4, 23]. On the other hand, disk capacity has been growing steadily over the years and it is now common to find commercial hard drives that have several hundreds of Gigabytes of storage capacity. With the availability of high-capacity disk drives, the workload data could be housed in fewer disks, thereby saving power. However, the reduction in I/O bandwidth by using fewer disks could lead to serious performance loss.

In order to quantify the performance loss and power benefits of such a storage system migration, we conduct a limit study. In this study, we analyze the extreme case of migrating the entire dataset of a workload onto a *single* state-of-the-art disk drive that has sufficient capacity to store that dataset. We model this high-capacity disk drive to be similar to the 750 GB Seagate Barracuda ES drive [40]. This is a four-platter, 7200 RPM drive, and has an 8 MB on-board cache. We denote this disk as the High Capacity Single Drive (*HC-SD*) configuration, and the corresponding multi-disk storage system whose data it stores as *MD*. We make the following assumption about how the data from *MD* is laid out on *HC-SD*: we assume that *HC-SD* is sequentially populated with data from each of the drives in *MD*. For example, if there are two disks, D1 and D2 in *MD*, we assume that *HC-SD* is populated with all the data from D1, followed by all the data in D2. (We resort to this approach because there is insufficient information available in the I/O traces about the specific strategy that was used to distribute the application data in *MD* in order for us to perform a more workload conscious data layout). Using this data layout, we compare the performance and power of *MD* and *HC-SD* for each of the workloads.

The performance of the workloads on the two system configurations are given in Figure 2. The graphs present performance as a Cumulative Distribution Function (CDF) of the response time. The corresponding power consumption results are given in Figure 3. Each stacked bar in Figure 3 gives the average power of the entire storage system, broken down into the four main operating modes of a disk: (i) idle, (ii) seeking, (iii) rotational latency periods, and (iv) data transfer between the platters and the electronics. Each pair of bars for a workload give the power consumption of the *MD* and *HC-SD* systems respectively.

From Figure 2, we can see that naively replacing a multi-disk system with a single disk drive can lead to severe

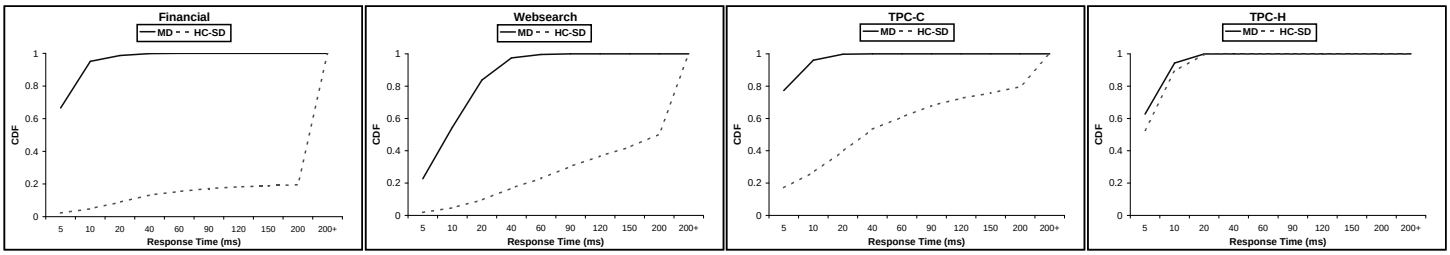


Figure 2: The performance gap between *MD* and *HC-SD*.

performance loss. Most of these workloads are I/O intensive and therefore reducing the I/O bandwidth creates significant performance bottlenecks. The only exception is the TPC-H workload. TPC-H has a fairly large inter-arrival time (8.76 ms, on average), which is less than the average response time of both *MD* and *HC-SD* for this workload (3.99 ms and 4.86 ms respectively) and hence experiences very little performance loss. Therefore, in either case, the storage system of TPC-H is able to service I/O requests faster than they arrive.

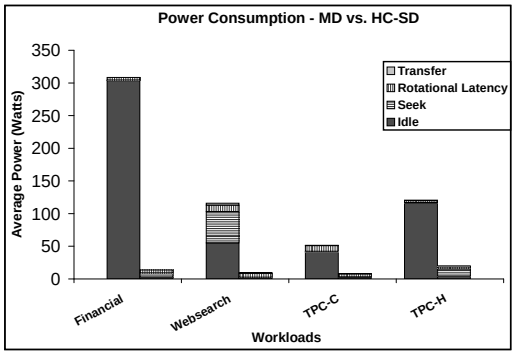


Figure 3: The power gap between *MD* and *HC-SD*. For each workload, the bar on the left corresponds to *MD* and the one on the right to *HC-SD*.

When we look at Figure 3, we see that migrating from a multi-disk system to a single-disk drive provides an *order of magnitude* reduction in the power consumption of the storage system. This result strongly motivates us to develop techniques to bridge the performance gap between *MD* and *HC-SD* while keeping the power consumption close to that of *HC-SD*. One interesting trend that we can observe in Figure 3 is that, despite all the workloads being I/O intensive and with no long period of inactivity, a large fraction of the power in the *MD* configuration is consumed when the disks are idle, which concurs with previous studies on server disk power management [16, 6].

In order to bridge the performance gap between *MD* and *HC-SD*, it is important to know what the key bottlenecks are. The performance of a disk drive is influenced by variety of factors, including, disk seeks, rotational latencies, transfer times, and disk cache locality. To determine the root cause of the performance loss in *HC-SD*, we need to isolate the effect of each factor on the disk response time. We find that disk transfer times are much smaller than the mechanical positioning delays across all the workloads, and therefore do not consider it further in the bottleneck analysis. To isolate the effect of disk cache size, we reran all the *HC-SD* experiments with a 64 MB cache. We find that using the larger disk cache has negligible impact on performance.

To determine empirically whether disk seeks are a bottleneck, we artificially modified the seek times calculated by the simulator so that they are one-half and one-fourth respectively of the actual seek time of each request. We also consider the ideal case where all disk seeks incur zero latency, thereby eliminating the effect of this factor on

performance. The results for the one-half, one-fourth, and zero seek time cases are shown by the CDF curves labeled *(1/2)S*, *(1/4)S*, and *S=0* respectively in the first row of graphs in Figure 4. We conduct a similar experiment for the rotational latencies, where we evaluate the performance if the rotational latencies are one-half and one-fourth of the original values respectively, and the case where this latency is eliminated completely. These rotational latency results are labeled as *(1/2)R*, *(1/4)R*, and *R=0* respectively in the second row of graphs in Figure 4.

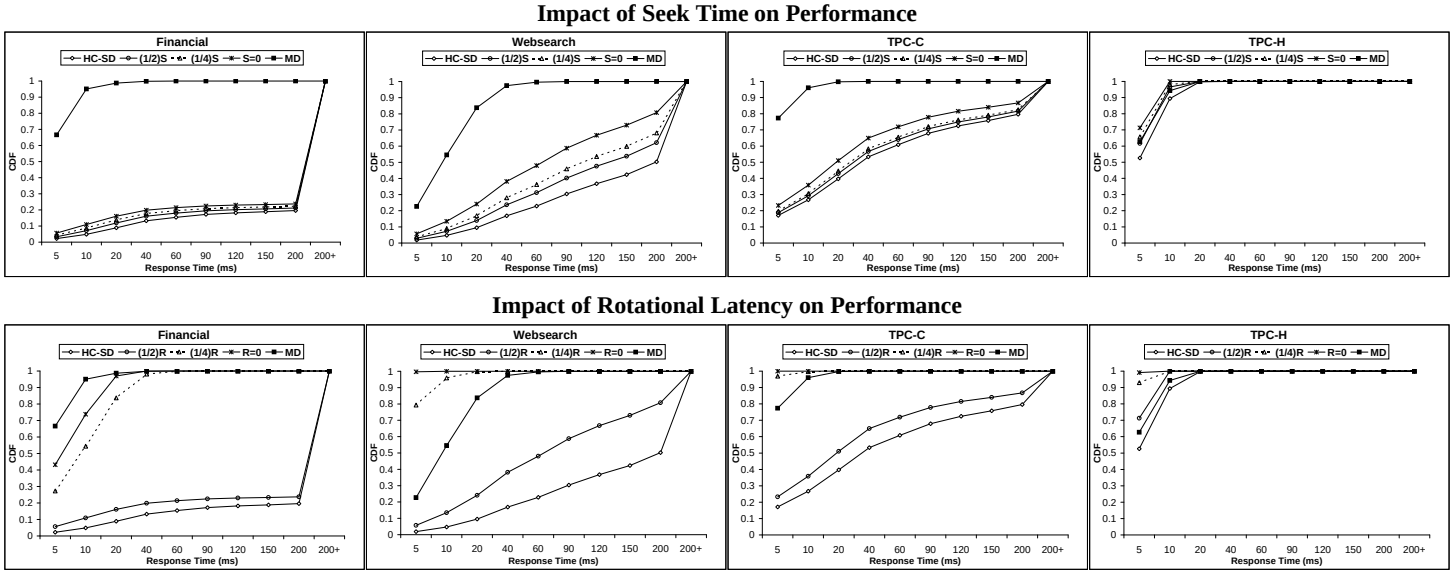


Figure 4: Bottleneck analysis of *HC-SD* performance. The graphs on the top row show the effect of seek time and the ones in the bottom row show the impact of rotational latency.

In Figure 4, we can clearly see that rotational latency is the primary performance bottleneck. In the case of Financial and TPC-C, even completely eliminating seek time does not boost performance significantly, whereas similar optimizations to the rotational latencies show large benefits. For Websearch and TPC-C, halving the rotational latencies lead to a significant boost in performance, which is evident by the extent to which the *(1/2)R* curves shift upwards from their corresponding *HC-SD* curves. In fact, for Websearch, TPC-C, and TPC-H, we see that a further reduction in the rotational latencies to one-fourth their original values (the *(1/4)R* curves) would allow us to surpass the performance of even the *MD* system. Although boosting seek time can also help *HC-SD* match the performance of *MD* for TPC-H, we can observe a slightly higher sensitivity to rotational latency than to seek time.

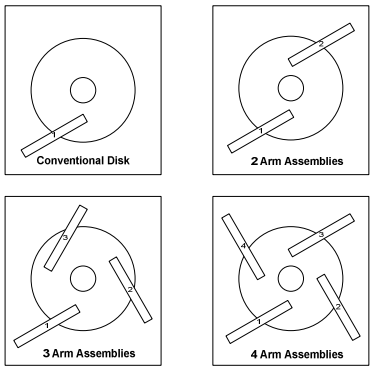
To summarize, we find that the primary bottleneck to performance when replacing *MD* by *HC-SD* is *rotational latency*. One straightforward approach to mitigating this bottleneck would be to increase the RPM of the drive. However, increasing the RPM can cause excessive heat dissipation within the disk drive [15], which can lead to reliability problems [19]. Indeed, commercial product roadmaps show that disk drive RPMs are not going to increase in the future [26], and therefore we need to explore alternative approaches to boost performance.

7.2 Design and Evaluation of Intra-Disk Parallelism

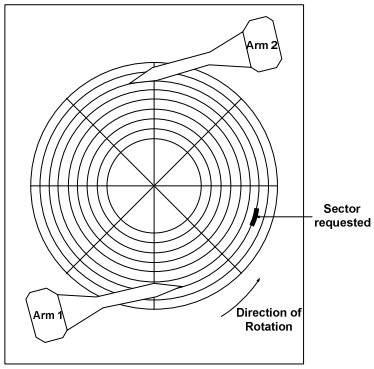
Having seen that rotational latency is the primary reason for the performance gap between *HC-SD* and *MD*, we now explore how intra-disk parallelism designs can help bridge this gap. Rotational latency could be minimized by incorporating parallelism along any of the four dimensions (D, A, S, or H) discussed in Section 4. For example, we could go in for a coarse-grained RAID-style design that provides parallelism along the D-dimension, by having

multiple spindle assemblies that can mask the rotational latency of one I/O request with the service time of others. At the other end of the spectrum, we could optimize along the fine-grained H-dimension, allowing multiple heads on an arm perform data accesses simultaneously. Such a design does not require the use of multiple spindles and is therefore easier to operate at a lower power. However, the effectiveness of such fine-grained parallelism depends on whether the data that is accessed by the heads on a single arm can satisfy the I/O requests presented to the storage system within a given window of time. Such data access restrictions can limit the ability of the disk to choose multiple pending I/O requests to be scheduled in parallel, especially if the workloads perform random I/O.

Since rotational latency is the primary performance bottleneck, we choose to focus on intra-disk parallelism along the A-dimension, which we believe provides a reasonable tradeoff between power consumption and I/O scheduling flexibility. Incorporating parallelism along this dimension requires replication of the VCM and the arms, but not the spindle assembly. Since the average power of the VCM is typically much lower than the SPM power [49], there are opportunities to boost performance by incorporating additional arm assemblies without significantly increasing the power consumption. Since our goal is to minimize rotational latency, we use the Shortest-Positioning Time First (SPTF) [47] scheduling policy at the disk. With multiple actuators, the SPTF-based disk arm scheduler has flexibility in choosing that arm assembly which minimizes the overall positioning time for a particular I/O request.



(a) Disk drive floorplans



(b) Minimizing rotational latency using two actuators. A conventional disk drive has only the arm labeled "Arm 1".

Figure 5: Intra-disk parallelism along the A-dimension.

We evaluate the behavior of three disk drive designs, all of which are instances of $D_1A_nS_1H_1$ and progressively extend the conventional disk drive architecture to provide intra-disk parallelism along the A-dimension:

- **HC-SD-SA(*n*):** This design extends the conventional *HC-SD* architecture by incorporating $n-1$ additional arm assemblies. (*HC-SD-SA(1)* is the same as *HC-SD*). However, this design retains two key characteristics of conventional disk drives in that, at any given point of time: (i) only a single arm (*SA*) assembly can be in motion, and (ii) only a single head can transfer data over the channel. However, for any given I/O request, the disk arm scheduler can choose between any of the idle arm assemblies based on whichever would minimize the positioning time of that disk request.

- **HC-SD-MA(*n*):** This design relaxes the first restriction in *HC-SD-SA(n)* by allowing Multiple Arm (*MA*) assemblies to be in motion simultaneously. However, as in the previous design, the single data channel design is assumed to be capable of transferring data to/from a single head. This design allows overlapping the service

time of one I/O request with the positioning phases of other requests that are waiting for disk access.

- **HC-SD-MC(*n*):** Here, we relax the assumption about the data channel from the previous *HC-SD-MA(n)* design and assume the existence of Multiple Channels (*MC*) where the data from heads on multiple arm assemblies can be transferred simultaneously, thereby providing even higher peak disk throughput.

In our experiments, we vary the number of arm assemblies (n) from 1 to 4. The placement of the arm assemblies within the disk drive for each of these four design points are given in the floorplan diagrams in Figure 5(a).

7.2.1 Performance Behavior

HC-SD-SA(*n*): The CDFs of the response time of the *HC-SD-SA(n)* design, along with those of the corresponding *MD* systems, are given in first row of graphs in Figure 6. We compare the performance of the *HC-SD-SA(n)* design points for each workload to the corresponding *MD* system of that workload. In order to quantify the impact that these designs have on rotational latency, we plot the Probability Density Function (PDF) of the rotational latencies of the I/O requests, given in the second row of graphs in Figure 6.

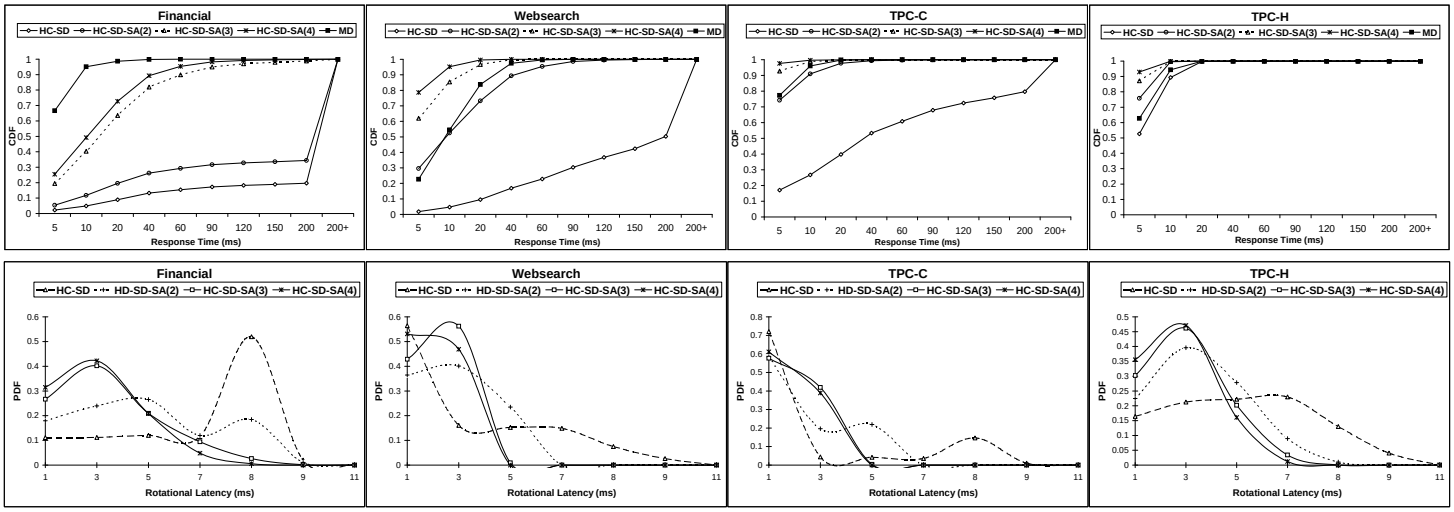


Figure 6: Performance impact of the *HC-SD-SA(n)* design.

When we look at the response time CDFs, we can see that the *HC-SD-SA(n)* design can provide substantial performance benefits compared to *HC-SD*. The rotational latency benefits of this design stem from the fact that, since there are multiple arms that are located at different points within the disk drive, the closest idle arm can be dispatched to service a given I/O request. In the case of Websearch and TPC-C, going from one to two arm assemblies provides a large boost in response times. The performance of these two workloads on *HC-SD-SA(2)* nearly match that of their *MD* counterpart. TPC-H also gets a slight improvement in response time, which allows it to perform better than *MD*. With three sets of disk arms, the Financial workload overcomes a substantial portion of the rotational latency bottleneck and gets a large performance boost. Websearch and TPC-C outperform *MD* with the use of three arm assemblies. As we can see from the PDF graphs for Websearch, TPC-C, and TPC-H, increasing the number of arms from one to two substantially shortens the tail of distributions from a higher to a lower range of rotational latencies. Going in for a third disk arm creates a similar shift in the rotational latency distribution for

Financial. However, increasing the number of arms beyond three provides diminishing performance returns, which can be seen from the closeness of the *HC-SD-SA(3)* and *HC-SD-SA(4)* curves in both the CDF and PDF graphs.

The high-level performance characteristics of these workloads can be explained from the bottleneck analysis in Section 7.1. When we look at the second row of graphs in Figure 4, we can see that significant reduction in the rotational latency of I/O requests on *HC-SD* can make its response times match or even exceed *MD* for Websearch, TPC-C, and TPC-H. Indeed, in Figure 6, we can observe that the *HC-SD-SA(n)* design provides these performance benefits for Websearch, TPC-C, and TPC-H. This result indicates that an intra-disk parallel design as simple as *HC-SD-SA(n)* can effectively mitigate rotational latency bottlenecks for these workloads. In the case of TPC-H, as noted previously, the load on the *HC-SD* system is relatively light and therefore going in for intra-disk parallelism does not result in significant performance improvements.

When comparing the response time CDFs of Websearch and TPC-C in Figure 6 to the rotational latency graphs in Figure 4, we can observe an interesting trend. When going from a *HC-SD* to a *HC-SD-SA(2)* configuration, the CDF curves for these two workloads shift up by a large amount, indicating a significant improvement in performance. On the other hand, the *HC-SD* and *(1/2)R* curves for these two workloads in Figure 4 show a smaller performance improvement. Intuitively it may appear that the *HC-SD-SA(2)* design, by virtue of having two arm assemblies, should, on average, halve the rotational latency of the I/O requests. However, the behavior of *HC-SD-SA(2)* depends on a variety of factors, such as, the stream of I/O block references, and how the disk arms are assigned to service the requests. These factors can cause the performance of *HC-SD-SA(2)* to diverge significantly from *(1/2)R*. Indeed, when we plot the PDF of the rotational latencies for *(1/2)R* and *HC-SD-SA(2)*, we find that the tail of the distribution is at 11 ms and 7 ms respectively for the two configurations for Websearch, and at 9.5 ms and 7 ms for TPC-C. (The PDF graphs are not shown here due to space limitations).

HC-SD-MA(*n*) and HC-SD-MC(*n*): On evaluating these two intra-disk parallel designs, we found that they provide very little performance improvements over *HC-SD-SA(n)*. We now explain why this happens. (We do not show the graphs from this experiment due to space limitations).

Both *HC-SD-MA(n)* and *HC-SD-MC(n)* attempt to exploit parallelism across I/O requests at the disk level. The former design attempts to overlap the seek time of one or more requests (based on the number of available arms) with the service time of another request, while the latter design goes one step further and facilitates the multiple in-flight I/O requests to transfer their data in parallel over the data channel. Therefore, in order to exploit the parallelism offered by these two disk drive designs, we need a sufficient "window" of requests from which we can choose requests to schedule to the multiple hardware resources.

In the *HC-SD* configuration, the rotational latency bottleneck results in long disk response times relative to the inter-arrival times. The *HC-SD-SA(n)* design mitigates the rotational latency bottleneck effectively for most of the workloads, thereby lowering the response time. However, since the arrival rate of I/O requests does not change across the designs, fewer requests get queued at the disk waiting to be serviced. This behavior has the effect of shrinking the scheduling window, thereby diminishing the effectiveness of the *HC-SD-MA(n)* and *HC-SD-MC(n)* designs over *HC-SD-SA(n)*. For example, for the two-arm configuration, we find that the inter-arrival times of I/O requests for the Websearch and TPC-C workloads are within 50 ms for 99% of the requests. On the other hand, 75% and 92% of the I/O requests in Websearch and TPC-C have response times below 40 ms for the HC-SD-SA(2) configuration. For those configurations where the response time is greater than the inter-arrival time (e.g., Financial, where 73% of the requests have an inter-arrival time less than 10 ms, whereas only 26% of the requests have response times lower

than 10 ms for *HC-SD-SA(2)*), we find that the providing additional arm assemblies to reduce the rotational latency has the first order impact on performance, rather than masking seek time or providing parallel data transfers using fewer sets of arms.

We note that one possible reason that *HC-SD-MA(n)* and *HC-SD-MC(n)* appear less effective can be attributed to the use of trace-driven simulation. In a real system, improvements in disk performance would translate to better system responsiveness at the higher level, which can increase the arrival rate of I/O requests. This increase would enlarge the window of requests for *HC-SD-MA(n)* and *HC-SD-MC(n)*. Since we do not modulate the arrival rate of I/O requests in this study, it is possible that the benefits of these two intra-disk parallelism designs are being masked. We focus solely on the *HC-SD-SA(n)* designin the remainder of this paper, but plan to re-visit these two other designs in our future work

7.2.2 Power Behavior and Optimization

Although *HC-SD-SA(n)* drives use multiple actuators, since only one VCM is active at any given time, the *peak power* consumption of these drives will be comparable to conventional disk drives. Peak power consumption is important for the disk drive designer, who has to design the drive to operate within a certain power/thermal envelope for reliability purposes [15]. However, it would be desirable, from an operating cost perspective, for the *average power* of intra-disk parallel disks be comparable to conventional drives as well. The average power consumption of the *HC-SD-SA(n)* designs and that of *HC-SD* are given in Figure 7. Each graph shows the power consumption, broken down into the four operating modes of the disk. The leftmost bar in each graph shows the power consumption of the *HC-SD* configuration. We omit the intermediate *HC-SD-SA(3)* design point from the graphs for space and clarity purposes.

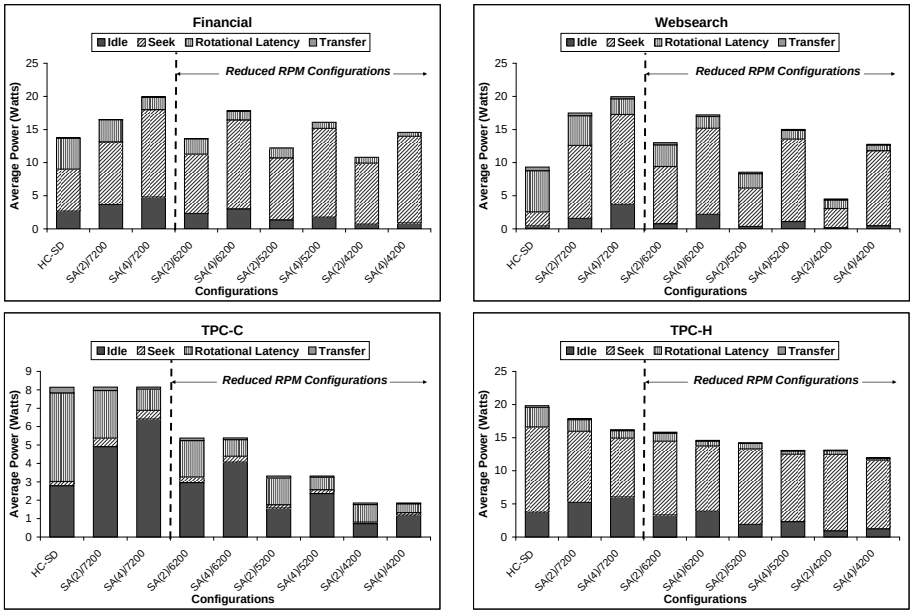


Figure 7: Average power consumption of the disk drive configurations. Each bar corresponds to a particular disk drive configuration and the x-axis labels are in the format: <*HC-SD-SA(n) configuration*>/<*RPM Value*>.

First, let us look at the 3 leftmost bars in each graph, which gives the average power consumption for the 7200 RPM disk drive configurations. We can see that the power consumed by the intra-disk parallel configurations are

comparable to *HC-SD* for TPC-C and TPC-H. The power consumption is about 2 Watts higher for the *HC-SD-SA(2)* configuration for Financial, but 6 Watts higher for the 4-arm design. For Websearch, the power consumed by the intra-disk parallel designs are significantly higher than *HC-SD*. Although the peak power consumption of a *HC-SD-SA(n)* drive will be close to that of a conventional disk drive, the average power can vary significantly based on the disk seeking characteristics of the workload. Indeed, when we look at the distribution of the seek times of the I/O requests in Websearch, we find that the percentage of requests that have a non-zero seek time for the *HC-SD*, *HC-SD-SA(2)*, and *HC-SD-SA(4)* configurations are 55%, 83%, and 90% respectively. The increased seek activity leads to more power being consumed by the arm assembly. This trend is clearly visible in the Websearch graph, where the power consumed during the seeking phases of the disk are higher for the intra-disk parallel designs. A similar trend is seen for the Financial workload as well, although the increase in seek power is less pronounced than in Websearch. However, as we saw earlier, the use of multiple arms and the SPTF scheduling algorithm leads to a significant decrease in the rotational latency, which results in a large performance boost for Websearch, allowing the intra-disk parallel design to surpass the performance of *MD*, while consuming roughly an *order of magnitude less power* than *MD*. On the other hand, the sharp reduction in the rotational latencies provided by the *HC-SD-SA(n)* designs for TPC-C leads to a large reduction in the power consumption. Among the four workloads, the absolute power consumption of the disks in TPC-C is the lowest and is close to the idle power of the disk drive. The reason for this is because the bulk of the power consumed by the *HC-SD* disk in TPC-C is due to rotational latency, during which time the arms are stationary and therefore the VCM does not consume any power. The intra-disk parallel drives reduce the rotational latencies (as shown in Figure 6) and therefore the power consumed in the rotational latency phase decreases. In TPC-H, both the seek and rotational latency components are optimized when going in for intra-disk parallelism and therefore the overall power consumption of the drives are reduced by going in for the *HC-SD-SA(n)* designs. However, the absolute reduction in power is small since TPC-H is not as heavily bottlenecked as the other three workloads and therefore its sensitivity to intra-disk parallelism is lower.

Reducing Average Power Consumption Through Lower RPM Design: Since RPM has nearly a cubic impact on the power consumption of a disk drive [24], one way to reduce the power consumption of an intra-disk parallel drive is to design it for a lower RPM. Lowering the RPM, on the other hand, would tend to increase the rotational latency. However, the extent to which I/O response time is impacted by the reduction in RPM can be offset by the use of multiple actuators. In order to determine how these factors interact, we analyze the power and performance of three lower RPM design points for *HC-SD-SA(n)*: 6200 RPM, 5200 RPM, and 4200 RPM respectively. The power consumption for these lower RPM design points are shown in Figure 7, and the response time CDFs are given in Figure 8. We plot the CDFs for only those workloads and design points where we can break-even with or achieve better performance than *MD*.

As we we can see from Figures 7 and 8, there are several design points where, for the three workloads, we can: (i) *match or surpass* the performance of the multi-disk system, (ii) consume an *order of magnitude* less power than *MD*, and (iii) consume power that is close to or *less than* that of a single conventional disk drive (for TPC-C and TPC-H).

7.3 Using Intra-Disk Parallel Drives to Build RAID Arrays for High Performance

For workloads that are very I/O intensive, a single intra-disk parallel drive might not be sufficient to meet performance goals. This naturally raises the question whether one should go in for a RAID array made up of conventional

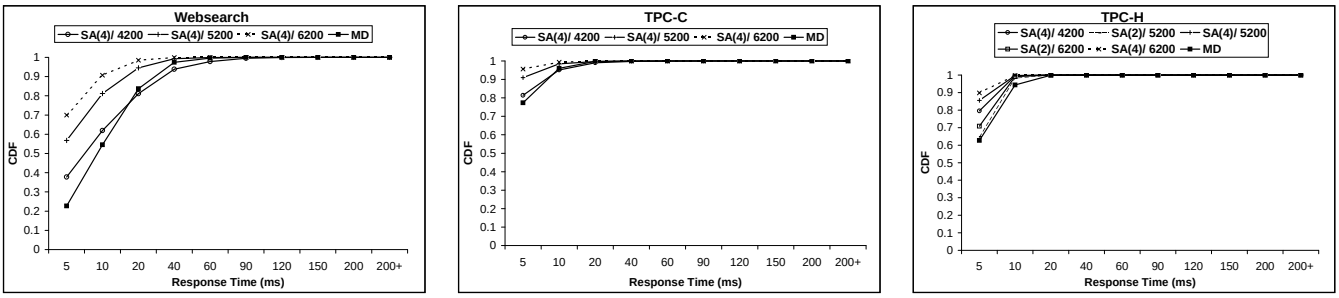


Figure 8: Performance of reduced RPM *HC-SD-SA(n)* designs whose response times match or exceed *MD*. Each legend entry is in the format: *<HC-SD-SA(n) configuration>/<RPM Value>*.

disk drives or an array that is composed of intra-disk parallel drives. We now explore this issue and compare the performance and power characteristics of these two types of RAID arrays. We consider conventional and intra-disk parallel drives that use the same underlying recording technology and have the same architectural characteristics, in terms of platter sizes, number of platters, RPM, and disk cache capacity.

Since we wish to study the tradeoffs between the two types of storage systems for a range of I/O intensities, we use synthetic workloads for this experiment. We use the synthetic workload generator in Disksim to create workloads that are composed of one million I/O requests. For all the synthetic workloads, 60% of the requests are reads and 20% of all requests are sequential. These parameters are based on the application I/O characteristics described in [36]. We vary the inter-arrival time of the I/O requests to the storage system using an exponential distribution. An exponential distribution models a purely random Poisson process and depicts a scenario where there is a steady stream of requests arriving at the storage system. We vary the mean of the distribution and consider three different inter-arrival time values: 8 ms, 4 ms, and 1 ms, which represent light, moderate, and heavy I/O loads respectively. We evaluate the performance and power for a range of disk counts in the storage system, from a single-drive configuration to a 16-disk system using both conventional disk drives (the *HC-SD* configuration) and intra-disk parallel drives (the *HC-SC-SA(2)* and *HC-SD-SA(4)* configurations). The results from this experiment are given in Figure 9. The first three graphs give the performance characteristics under each inter-arrival time scenario for disk arrays that are composed of *HC-SD*, *HC-SD-SA(2)* and *HC-SD-SA(4)* drives. We express performance in terms of the 90th percentile of the response time in the CDFs (i.e., maximum response times incurred by 90% of the requests in the workload). The power graph shows the the average power consumption of the *HC-SD*-based disk array when it reaches its steady-state performance and that of the *HC-SD-SA(2)* and *HC-SD-SA(4)* arrays when their performance breaks even with the steady-state performance of the *HC-SD* array.

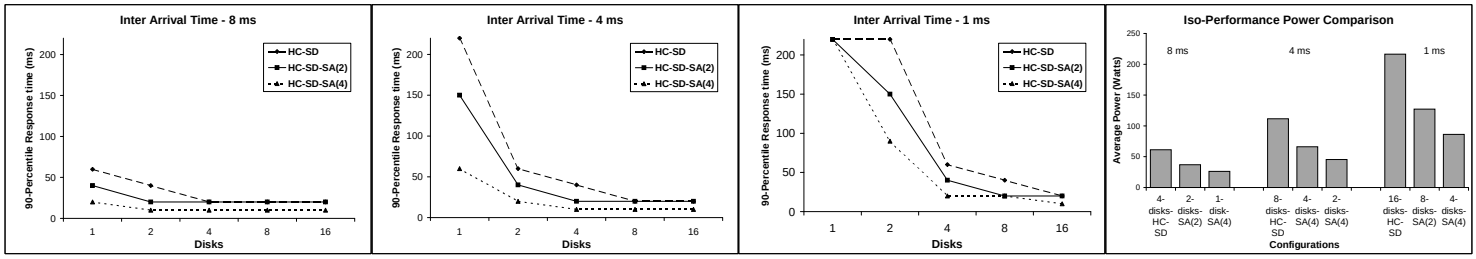


Figure 9: Performance and power characteristics of RAID arrays using intra-disk parallel drives

The graphs in Figure 9 show a clear performance advantage for intra-disk parallelism. For the relatively light 8

ms inter-arrival time workload, the performance of *HC-SD-SA(2)* and *HC-SD-SA(4)* reach their steady-state values with just two disks in the array, whereas 4 *HC-SD* drives are required to get performance that is comparable to the 2-disk *HC-SD-SA(2)* array. We can see that a single 4-actuator drive is able to break-even with the performance of the 4-disk *HC-SD* and 2-disk *HC-SD-SA(2)* arrays respectively. From the power perspective, the array of conventional disks consumes 61.4 Watts, whereas the *HC-SD-SA(2)* and *HC-SD-SA(4)* arrays consume 37.1 Watts and 26.2 Watts of power respectively. Under moderate and heavy I/O loads (4 ms and 1 ms inter-arrival times respectively), we can see that the intra-disk parallel drives are able to mitigate the I/O bottlenecks with fewer disks than arrays composed of conventional disk drives. For the 1 ms inter-arrival time workload, we find that the ratio of the number of intra-disk parallel drives to conventional drives needed to break-even in performance is the same as under lighter loads. However, since we need 16 conventional disks to break-even with the performance of an 8-disk *HC-SD-SA(2)* and 4-disk *HC-SD-SA(4)* array respectively, the average power consumption of the intra-disk parallel drive based arrays are lower. We find that the *HC-SD-SA(2)* and *HC-SD-SA(4)* arrays consume 41% and 60% less power than the *HC-SD*-based array respectively.

These results indicate that using intra-disk parallel drives is more attractive, performance and power-wise, than using conventional disks to build RAID arrays for I/O intensive workloads.

8 Issues in Implementing Intra-Disk Parallel Drives

Our discussions so far have focused on the performance and power aspects of intra-disk parallelism. We now discuss three key engineering issues that need to be addressed when building intra-disk parallel drives.

- **Vibration Tolerance:** One problem that can arise with with having multiple actuators within a single disk drive enclosure is vibration. When more than one set of arms are in motion, the physical movement of one arm assembly can induce off-track errors in the other. These vibration induced off-track errors, if left unchecked, can lead to the inability to reliably perform disk seeks or data transfers between the platters and the head, thereby negating the benefits of intra-disk parallelism. Although vibration problems are expected to be less severe for *HC-SD-SA(n)* drives, since only one actuator is active at any given time, it is still important to address this issue for intra-disk parallelism in general.

  Modern server drives are already built to handle significant amounts of vibration, since these disks are usually housed with several other drives within a single rack or cabinet [1, 37]. At runtime, a single disk drive can experience a large amount of external vibration induced by the other drives that are operating in close proximity. To operate reliably and efficiently under such heavy vibration conditions, the servo processing system of server drives are designed to use data from vibration sensors embedded within the drive to adjust to varying degrees of vibration [1, 13]. Although the source of the vibrations are different in an intra-disk parallel drive (internal arm assembly vs. external disk drive), the vibration compensation technologies that exist in modern server drives can be leveraged for intra-disk parallel drives.

- **Air Turbulence:** Another reason for vibration related problems inside an intra-disk parallel disk drive is air turbulence due to the presence of multiple arm assemblies. Here, there are two turbulence-related issues that need to be tackled: (i) vibration of the platters, and (ii) vibration of the heads. Studies on the air flow pattern within disk drives [28, 10] show that there is turbulence in a region surrounding the head, but the gap flow reverts to laminar beyond that region. By placing the arm assemblies diagonally from each other (as shown in

Figure 1), the vibration of the platter due to the second arm will be at most additive (i.e., the effects of the two heads will be independent of each other, and the total is at most twice larger), and the heads on the respective arm assemblies will not affect each other either. These platter vibrations can be reduced to acceptable levels via engineering methods [25], and the impact of the turbulence can be mitigated using the servo mechanisms discussed earlier.

- **Disk Drive Reliability:** Intra-disk parallel drives make use of extra hardware components. If the failure of any one component were to render the drive unusable, then the Mean Time to Failure (MTTF) of an intra-disk parallel drive would be worse than a conventional disk drive. In order to mitigate this problem, intra-disk parallel drives need to be designed to allow *graceful degradation* so that a failure (or an impending failure) in a head or arm assembly can be handled by deconfiguring the failing component. Almost all modern disk drives are equipped with sensors, based on the Self-Monitoring Analysis and Reporting Technology (SMART) [43], which can predict impending failures. A recent study of failure data collected from a large number of disks has shown that the data from SMART sensors correlate highly with disk failures and motivate the need to enhance the SMART architecture [35]. The firmware of the intra-disk parallel drives need to be modified to allow deconfiguration of hardware components based on data from these sensors at runtime.

9 Preliminary Cost-Benefit Analysis of Intra-Disk Parallel Drives

Our results thus far have highlighted how intra-disk parallel drives, built using modern disk drive technology, offer a fundamentally different set of tradeoffs, performance and power wise, than the multi-actuator (e.g., IBM 3380) and conventional drives of the past. In Section 7.3 we saw that a single 4-actuator intra-disk parallel drive delivers performance that is comparable to two 2-actuator drives and to a disk array of four conventional drives. Since these performance and power benefits are obtained by extending conventional disk drive architectures with additional hardware components, we are faced with an important question: *Would it be worth spending more money on a single intra-disk parallel drive than on multiple conventional drives?* We now provide a preliminary estimate of the cost of manufacturing intra-disk parallel drives, using *real cost data* obtained from several companies within the disk drive industry. Our analysis reveals that intra-disk parallelism is promising from the cost viewpoint as well.

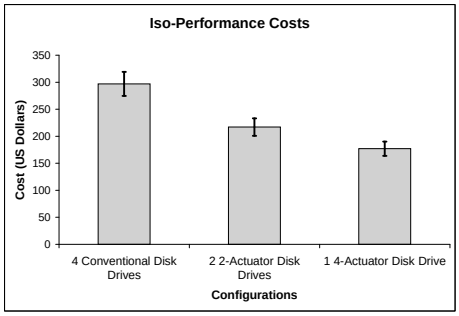
Building a disk drive involves material costs, for all the hardware components, such as the heads, motors, and the electronics, and also labor costs and other overheads. Studies about the disk drive industry have shown that the bulk of the manufacturing costs of a disk go into the materials [17, 3] and, therefore, we focus on quantifying these costs. Many of the components that go into a disk drive are manufactured by different companies, each of whom specialize in making a particular component, such as the head or a pivot bearing, and supply their components to disk drive companies on a volume basis. In order to estimate the cost of each of these components, we contacted several major component manufacturers to obtain data about the price at which they supply these components to disk drive companies, on a volume basis, for their server hard drives. (Note: A few of the large disk drive companies manufacture several of these components in-house. However, given the relatively low market price differentiation between disk drive products of the same class from different companies, we assume that the component manufacturing costs are comparable across the industry). A component-wise breakdown of costs of several key disk drive components are given in Table 10(a). The companies from whom we obtained this data are listed in the figure caption. (**Caveats:** (i) The costs listed in Table 10(a) are *estimates* provided to us by the companies through personal correspondence. Sometimes we were provided a single value and sometimes we were given a price range. The exact price of a

component would depend on the precise low-level specifications of the disk drive to be built and other purchasing issues that are too early to finalize at the current stage of this research project. (ii) We identified the 9 components listed in Table 10(a) as the key cost contributors based on discussions that we had with a disk drive company about manufacturing cost issues. (iii) We assume that the material costs for building a disk drive and the final cost of the product are related and that a rise or fall in the manufacturing costs will translate to similar effects on the price at which the drive is marketed).

We give the per-component cost estimates provided to us by the manufacturers and calculate the material costs for a conventional disk drive, a 2-actuator intra-disk parallel drive and also a 4-actuator drive. To be consistent with our previous discussions, we calculated the cost for a four-platter drive. In Figure 10(b), we show the costs of the three storage system configurations that deliver equivalent performance, based on the the results in Section 7.3. Each of the bars in the Figure are based on the average of the low and high costs of each disk drive configuration listed in Table 10(a). The low-to-high cost range is depicted using error bars.

Component	Component Cost	Conventional Disk Drive	2-Actuator Disk Drive	4-actuator Disk Drive
Media	6-7	24-28	24-28	24-28
Spindle Motor	5-10	5-10	5-10	5-10
Voice-Coil Motor	1-2	1-2	2-4	4-8
Head Suspension	0.50-0.90	2-3.6	4-7.2	8-14.4
Head	3	24	48	96
Pivot Bearing	3	3	6	12
Disk Controller	4-5	4-5	4-5	4-5
Motor Driver	3.5-4	3.5-4	5-6	8-10
Preamplifier	1.2	1.2	2.4	4.8
Total Estimated Cost		67.7-80.8	100.4-116.6	165.8-188.2

(a) Estimated component and disk drive costs (in US Dollars).



(b) Iso-performance cost comparison between conventional and intra-disk parallel drives. The error-bars give the cost range based on the values in the table on the left.

Figure 10: Preliminary cost-benefit analysis of intra-disk parallel drives. Personal communication from: US Fuji Electric Inc., Nidec Corporation, H2W Technologies Inc., Hutchinson Technology Inc., Hitachi Metals America Ltd., NMB Technologies Corporation, STMicroelectronics. The cost data was collected in November 2007.

As Table 10(a) indicates, the bulk of the cost increase for building intra-disk parallel drives is expected to be in the heads. Other components, such as, the VCMs and their motor drivers, head suspensions, pivot bearings, and head preamplifiers are expected to constitute only a small part of the overall cost of an intra-disk parallel drive. However, the overarching question is whether this increased cost (and its corresponding higher selling price) would be worth the investment for the eventual customer of the product. As Figure 10(b) indicates, the use of 2 *HC-SD-SA(n)* intra-disk parallel drives delivers equivalent performance as 4 conventional disk drives, but at *27% lower cost*. One 4-actuator drive delivers the same performance, but at *40% lower cost* than the 4-disk array of conventional drives. These results are encouraging and motivate us to explore intra-disk parallelism further.

10 Conclusions

Server storage systems consume a large amount of power. These systems are built using a large number of disk drives to meet the I/O performance demands of server workloads. In this paper, we show that we can build server storage systems using far fewer disks, thereby providing huge power savings, but provide intra-disk parallelism to maintain high performance. We present a taxonomy for the intra-disk parallelism design space, discuss implementation issues, and provide a preliminary cost-benefit analysis of building and deploying intra-disk parallel drives using real cost

data obtained from the disk drive industry. Given the performance, power, and cost benefits of intra-disk parallelism, which is a complete trend-reversal from the multi-actuator drives of decades past, we strongly believe that intra-disk parallelism holds great promise for building high-performance, low power server storage systems.

11 Acknowledgements

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

References

[1] 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.

[2] M. Ault. Tuning Disk Architectures for Databases. *DBAzine.com*, June 2005. http://www.dbazine.com/oracle/or-articles/ault1.

[3] R.E. Bohn and C. Terwiesch. The Economics of Yield-Driven Processes. *Elsevier Journal of Operations Management*, 18(1):41–59, December 1999.

[4] D.K. Burleson. Disk Management for Oracle. *DBAzine.com*, June 2005. http://www.dbazine.com/oracle/or-articles/burleson19.

[5] L.R. Carley and et al. Single Chip Computers with Microelectromechanical Systems-based Magnetic Memory. *Journal of Applied Physics*, 87(9):6680–6685, May 2000.

[6] 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.

[7] 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.

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

[9] Conner: CP-3100 105MB 3.5”/HH SCSI1 SE Specification. http://stason.org/TULARC/pc/hard-drives-hdd/conner/CP-3100-105MB-3-5-HH-SCSI1-SE.html.

[10] N. Tsuda et al. Unsteady Analysis and Experimental Verification of the Aerodynamic Vibration Mechanism of HDD Arms. *IEEE Transactions of Magnetics*, 39(2), March 2003.

[11] J. Freitas and W. Wilcke. Technology and Application Trends in High-End Data Centers. *IBM Journal of Research and Development*, July 2008.

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

[13] G. Guo and J. Zhang. Feedforward Control for Reducing Disk-Flutter-Induced Track Misregistration. *IEEE Transactions on Magnetics*, 39(4):2103–2108, July 2003.

[14] 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.

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

[16] 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.

[17] S.M. Hampton. Process Cost Analysis for Hard Disk Manufacturing. Technical Report 96-02, Information Storage Industry Center, University of California, San Diego, September 1996.

[18] K. Haughton. Design Considerations in the IBM 3340 Disk File. In *Proceedings of the IEEE Computer Society Conference*, February 1974.

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

[20] Hitachi Deskstar 7K1000. http://www.hitachigst.com.

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

[22] W.W. Hsu and A.J. Smith. Characteristics of I/O Traffic in Personal Computer and Server Workloads. *IBM Systems Journal*, 42(2):347–372, 2003.

[23] IBM Tivoli Access Manager for e-Business Version 6.0 - Performance Tuning Guide.

[24] I.Sato, K. Otani, M. Mizukami, S. Oguchi, K. Hoshiya, and K-I. Shimokura. Characteristics of Heat Transfer in Small Disk Enclosures at High Rotation Speeds. *IEEE Transactions on Components, Packaging, and Manufacturing Technology*, 13(4):1006–1011, December 1990.

[25] M. Kazemi. Numerical Analysis of Off-Track Vibration of Head Gimbal Assembly in Hard Disk Drive Caused by the Airflow. *Microsystems Technology Journal*, 13(8-10), April 2007.

[26] M.H. Kryder. Future Storage Technologies: A Look Beyond the Horizon. In *Computerworld Storage Networking World Conference*, April 2006. http://www.snwusa.com/documents/presentations-s06/MarkKryder.pdf.

[27] C. Larabie. Power and Storage: The Hidden Cost of Ownership - Storage Management. *Computer Technology Review*, October 2003.

[28] E. Lennemann. Aerodynamic Aspects of Disk Files. *IBM Journal of Research and Development*, 18(6):480–488, 1974.

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

[30] C. Lumb, J. Schindler, and G.R. Ganger. Freeblock Scheduling Outside of Disk Firmware. In *Proceedings of the USENIX Conference on File and Storage Technonologies (FAST)*, January 2002.

[31] S.W. Ng. Improving Disk Performance Via Latency Reduction. *IEEE Transactions on Computers*, 40(1):22–30, January 1991.

[32] 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.

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

[34] E. Pinheiro, R. Bianchini, and Cezary Dubnicki. Exploiting redundancy to conserve energy in storage systems. *SIGMETRICS Performance Evaluation Review*, 34(1):15–26, 2006.

[35] E. Pinheiro, W-D. Weber, and L.A. Barroso. Failure Trends in a Large Disk Drive Population. In *Proceedings of the USENIX Conference on File and Storage Technonologies (FAST)*, February 2007.

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

[37] T.M. Ruwart and Y. Lu. Performance Impact of External Vibration on Consumer-Grade and Enterprise-Class Disk Drives. In *Proceedings of the IEEE/NASA Goddard Conference on Mass Storage Systems and Technologies (MSST)*, pages 11–14, April 2005.

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

[39] S. W. Schlosser, J. L. Griffin, D. Nagle, and G. R. Ganger. Designing computer systems with MEMS-based storage. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pages 1–12, 2000.

[40] Seagate Barracuda ES SATA 3.0Gb/s 750-GB Hard Drive Product Manual. http://www.seagate.com/.

[41] D.E. Shasha and P. Bonnet. *Database Tuning: Principles, Experiments, and Troubleshooting Techniques*. Morgan Kauffman Publishers, 2003.

[42] A. J. Smith. On the effectiveness of buffered and multiple arm disks. In *Proceedings of the International Symposium Computer architecture (ISCA)*, pages 242–248, 1978.

[43] Storagereview/The PC Guide - Self Monitoring Analysis and Reporting Technology(SMART). http://www.storagereview.com/guide2000/ref/hdd/perf/qual/featuresSMART.html.

[44] The IBM 350 RAMAC Disk File, February 1984. http://www.magneticdiskheritagecenter.org/MDHC/RAMACBrochure.pdf.

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

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

[47] B.L. Worthington, G.R. Ganger, and Y.N. Patt. Scheduling Algorithms for Modern Disk Drives. In *Proceedings of the ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, pages 241–251, May 1994.

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

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

[50] Q. Zhu, Z. Chen, L. Tan, Y. Zhou, K. Keeton, and J. Wilkes. Hibernator: Helping Disk Arrays Sleep Through The Winter. In *Proceedings of the Symposium on Operating Systems Principles (SOSP)*, pages 177–190, October 2005.

[51] 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.