Using Intradisk Parallelism to Build Energy-Efficient Storage Systems

Server storage systems use numerous disks to achieve high performance, thereby consuming a significant amount of power. Intradisk parallelism can significantly reduce such systems' power consumption by letting disk drives exploit parallelism in the I/O request stream. By doing so, it's possible to match, and even surpass, a storage array's performance for these workloads using a single, high-capacity disk drive.

Sudhanva Gurumurthi
Mircea R. Stan
University of Virginia

Sriram Sankar
Microsoft

Many applications today, used by millions of people around the clock, store and process massive data sets. These data-centric computing applications include transaction processing, search engines, and e-mail services as well as newer types of applications, such as social networking and photo and video sharing. The amount of data these applications must handle is growing tremendously, and experts estimate that the amount of data generated and stored worldwide will reach nearly 1,000 exabytes in a few years.[1] Data centers will need to store and centrally manage this data, and applications will use it to process and deliver content to users. In addition to storage capacity, storage systems within data centers must provide these data-intensive applications with high I/O performance.

The conventional approach to building high-performance storage systems is to aggregate several disk drives and form disk arrays. However, because disk access times are relatively large, high-performance storage systems use a large number of disks, and performance (rather than capacity) is the primary driver for selecting the number of disks. Moreover, many high-performance storage systems also tend to underutilize disk capacity to leverage the higher data rates on the outer zones of the platters and reduce the impact of disk-arm positioning delays.[2,3] This approach to storage system design has resulted in a significant increase in data centers' storage power consumption. Moreover, researchers expect disk drive performance to improve relatively slowly due to limitations in the magnetic recording technology and thermal constraints associated with making the platters spin faster.[4]

Although modern disk drives offer some parallelism in the form of prefetching and tagged command queuing, which helps aggregate multiple I/O requests within the drive to improve disk arm scheduling, achieving good parallel I/O performance requires multiple

Published by the IEEE Computer Society

disk drives. Because future data-center storage systems will need to handle massive data sets and deliver high performance, the status quo in high-performance storage system design won't scale well from an energy viewpoint. However, storage density has been growing briskly over the years, and some drives available today can store terabytes of data.

Given these trends, it's natural to ask the question: Can we design storage systems that use the minimal set of disks, purely for satisfying capacity requirements, and still achieve performance comparable to systems that are designed for high performance? With fewer disks, we can reduce a storage system's total power consumption, but this can create I/O bottlenecks and lead to diminished performance. To bridge this performance gap, but still maintain low power consumption, we propose extending disk drive architecture to support intradisk parallelism. Intradisk parallelism provides performance, power, and cost benefits that make it a promising technology for building high-performance, low-power server storage systems.

Disk drives and intradisk parallelism

A hard disk drive consists of one or more platters stacked on top of each other and held in place by a central spindle. The surfaces of each platter have a magnetic material coating, 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 that are mounted on sliders and float over the platters' surface in a thin cushion of air. Disk arms connected to a central assembly hold the sliders in place. A single voice coil motor (VCM) moves all the arms in the assembly in unison. (The arm assembly is also known as the *actuator*; this article uses the terms interchangeably.) In addition to these electromechanical components, disks have several electronic circuitries such as the disk controller, data channel, motor drivers, and on-board cache.

At runtime, two structurally independent sets of electromechanical activities occur within a disk drive:

- the head (driven by the VCM) moves radially across the disk's surface and
- the platters (driven by the SPM) rotate under the head.

These two moving subsystems affect two different components of the total disk access time:

- *Seek time* is the time required to move the head to the desired track.
- *Rotational latency* is the time taken for the appropriate sector to rotate under the head.

In addition to these two latencies, the disk access time 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, the latencies for the mechanical positioning activities dominate the disk access time.

A conventional disk drive can only service a single I/O request at a time. For any given disk request that requires accessing the platters (that is, it can't be serviced from the disk cache), the request's access time is serialized through the seek, rotational-latency, and data-transfer phases. That is, although the arm and spindle assemblies are physically independent of the electromechanical systems, they are used in a tightly coupled manner due to the way that disk accesses are performed. Furthermore, all the resources within each of the drive's electromechanical system 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.

Intradisk parallelism enhances this design by

- decoupling how the two electromechanical systems 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
- decoupling the multiplicity of components within each electromechanical system, such as the heads on an arm assembly.

Achieving parallelism using either approach requires additional hardware support.

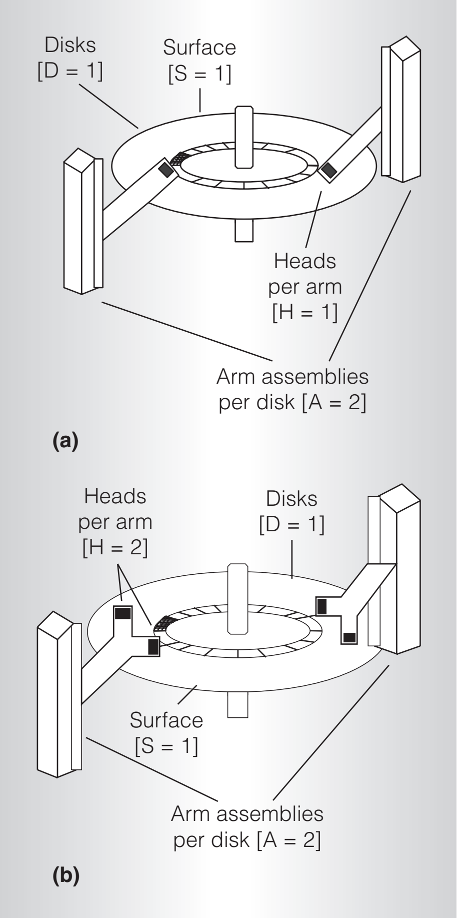

Figure 1. Example design points within the DASH intradisk parallelism taxonomy: a $D_1A_2S_1H_1$ disk drive (a) and a $D_1A_2S_1H_2$ disk drive (b).

Multiactuator drives—that is, disk drives with two or more arm assemblies capable of independent motion—were widely used in the 1970s and 1980s but are no longer available. Instead of using parallel disk drives, we build RAID arrays using multiple single-actuator disk drives. (Although it's important to understand why the industry discontinued multiactuator drives and why intradisk parallelism, in the context of modern disk drives, is different, such a discussion is out of this article's scope. An analysis of this trend is available elsewhere.[5])

DASH parallel disk taxonomy

Multiactuator drives are a single design point within the space of intradisk parallelism. Because the intradisk parallelism design space is large, it needs a taxonomy for systematically formulating specific designs, which we've developed. Specifically, we express a specific disk configuration hierarchically as a four-tuple, $D_kA_lS_mH_n$, where k, l, m, and n indicate the degree of parallelism in four of the possible electromechanical components in which we can incorporate parallelism, from the most coarse- to the most fine-grained component: the disk stack (D), arm assembly (A), surface (S), and head (H). For example, a conventional disk has the configuration $D_1A_1S_1H_1$, indicating a single disk stack that is accessed by one set of arms, and data that 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 1a shows the physical design of a $D_1A_2S_1H_1$ configuration, which is a two-actuator drive that can provide a maximum of two data transfer paths to and from the drive. Figure 1b shows a $D_1A_2S_1H_2$ configuration, which consists of two arm assemblies and two heads on each arm that can access a single surface, thereby providing a maximum of four possible data transfer paths to and from the disk drive.

This approach gives us four levels of parallelism dimensions: D, A, S, and H. Level one (D) involves disk stacks. We can have multiple disk stacks, each with its own spindle, which is precisely the form of parallelism that RAID provides. However, we can incorporate this form of parallelism within a single disk drive by shrinking the platter size and adding more disk stacks. Because the platter size strongly influences the power dissipated by the spindle assembly (approximately 4.6th power of the platter size[6]), shrinking the platters can facilitate incorporating multiple disk stacks within a single disk drive's power envelope and area. In fact, previous work has explored the possibility of replacing a laptop disk drive with a small RAID array consisting of smaller-diameter disks.[7]

At level two (A), we could vary the number of actuators for each disk to provide parallelism. Providing parallelism along this dimension can help 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.

At level three (S), the two surfaces on each platter could be accessed independently. We can implement parallelism across surfaces 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 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.

Lastly, for level four (H), conventional disk drives only have 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:

- on a radial line on the arm, from the axis of actuation, or
- equidistant from the axis of actuation (which Figure 1b illustrates).

This level involves two design variables: the distance between each head and the number of heads per arm.

This taxonomy deals only with parallelism in the disk drive's electromechanical subsystem and not the electronic data channel. 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 designs that we evaluate.

Experimental setup and workloads

To evaluate intradisk parallelism, we carried out experiments using the Disksim simulator,[8] which models the performance of storage systems in detail. We augmented Disksim with power models for the electromechanical components that we developed in prior work.[9] We used a set of commercial server I/O traces as our workload suite. Table 1 provides information about these traces and the original storage systems on which they were collected. Financial and Websearch are I/O traces collected at a large financial institution and a popular Internet search engine (see the UMass Trace Repository, http://traces.cs.umass.edu), respectively. We collected the TPC-C trace data on a two-way SMP machine running the IBM DB2 EEE database engine and ran the TPC-C benchmark for a 20-warehouse configuration with eight clients. We collected the TPC-H trace on an eight-way IBM Netfinity SMP machine with 15 disks and running the IBM DB2 EE edition. We ran the TPC-H benchmark in the power test mode, in which we executed the benchmark's 22 queries consecutively.

Table 1. The original storage systems' workloads and configuration.

Workload	Requests	Disks	Capacity (Gbytes)	RPM	Platters
Financial	5,334,945	24	19.07	10,000	4
Websearch	4,579,809	6	19.07	10,000	4
TPC-C	6,155,547	4	37.17	10,000	4
TPC-H	4,228,725	15	35.96	7,200	6

Results

We conduct three sets of experiments. The first aimed to determine the power benefits of replacing a multidisk storage array with a single high-capacity disk drive, the performance gap between the performance-optimized storage array and the single disk drive configuration, and the bottlenecks that lead to this gap. In the second set of experiments, we evaluated the performance and power characteristics of an intradisk parallel design that would alleviate the bottlenecks we identified in the first experiment. The third set of experiments used synthetic workloads to evaluate the performance and power characteristics of RAID arrays built using intradisk 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 with the parallel drives (such as platter sizes, RPM, and disk cache capacity).

Performance and power-limit study

To quantify the performance loss and power benefits of a storage system migration, we analyzed the extreme case of migrating a workload's entire data set onto a single state-of-the-art disk drive with sufficient capacity. We modeled this high-capacity disk drive to be similar to the 750-Gbyte Seagate Barracuda

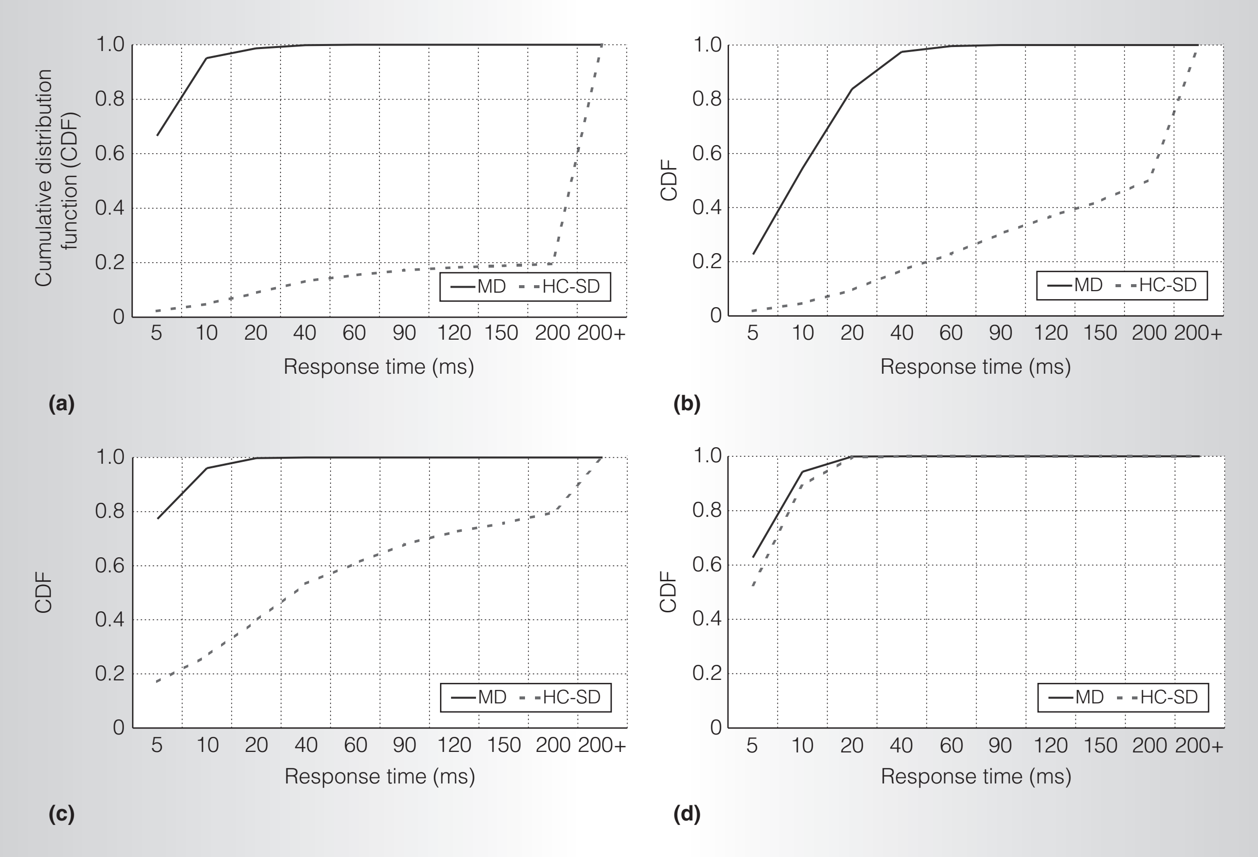

Figure 2. The performance gap between the multidisk (MD) and high-capacity (HC-SD) configurations for Financial (a), Websearch (b), TPC-C (c), and TPC-H (d).

ES drive, with a four-platter, 7,200 RPM drive, and an 8-Mbyte on-board cache. We call this the high-capacity single drive (HC-SD) configuration, and its corresponding multidisk storage system is MD. We assumed 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 resorted to this approach because there was insufficient information available in the I/O traces about the specific strategy that was used to distribute the application data in MD for us to perform a more workload-conscious data layout.) Using this data layout, we compared the performance and power of MD and HC-SD for each of the workloads.

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

From Figure 2, we can see that naively replacing a multidisk system with a single disk drive can lead to severe performance loss. Most of these workloads are I/O intensive, so reducing the I/O bandwidth creates significant performance bottlenecks. The only exception is the TPC-H workload. It has a fairly large interarrival 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 and 4.86 ms, respectively), so it experiences little performance loss. Therefore, in either case, TPC-H's storage system can service I/O requests faster than they arrive.

As Figure 3 shows, migrating from a multidisk system to a single-disk drive reduces the storage system's power consumption by an order of magnitude. 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.

To bridge the performance gap between MD and HC-SD, it's important to identify the key bottlenecks. Various factors influence a disk drive's performance, including disk seeks, rotational latencies, transfer times, and disk cache locality. To determine the root cause of the HC-SD performance loss, we need to isolate the effect of each factor on the disk response time. We can see that disk transfer times are much smaller than the mechanical positioning delays across all the workloads, so we don't need to consider that further. To isolate the effect of disk cache size, we reran all the HC-SD experiments with a 64-Mbyte cache. We found that using the larger disk cache has negligible impact on performance.

To empirically determine whether rotational latencies are a bottleneck, we artificially modified the rotational latencies the simulator calculated so that they are one-half and one-fourth of the actual rotational latency of each request, respectively. We also considered the ideal case in which all rotational latencies are zero, thereby eliminating the effect of this factor on performance. We conducted a similar experiment for the seek times.

This analysis revealed that the primary performance bottleneck when replacing MD by HC-SD is rotational latency. One straightforward approach to mitigating this bottleneck would be to increase the drive's RPM. However, this could cause excessive heat dissipation within the disk drive,[4] which can lead to reliability problems. Commercial product roadmaps show that disk drive RPMs aren't going to increase in the future,[10] so we need to explore alternative approaches to boost performance.

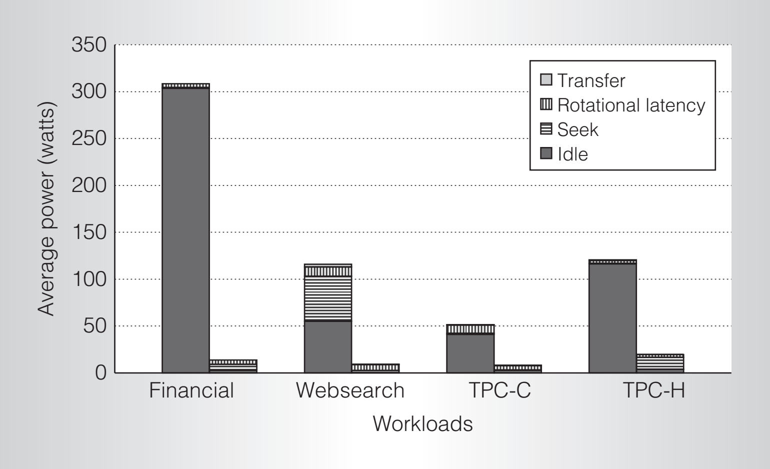

Figure 3. The power gap between MD and HC-SD. For each workload, the left bar corresponds to MD and the right bar to HC-SD.

Evaluating intradisk parallelism

Having seen that rotational latency is the primary reason for the performance gap between HC-SD and MD, we next explored how intradisk parallelism designs can help bridge this gap. We could minimize rotational latency by incorporating parallelism along any of the four dimensions (D, A, S, or H). For example, we could try 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, letting multiple heads on an arm perform data accesses simultaneously. Such a design wouldn't 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 disk's ability to choose multiple pending I/O requests to be scheduled in parallel, especially if the workloads perform random I/O.

Because rotational latency is the primary performance bottleneck, we focus on intradisk parallelism along the A dimension, which provides a reasonable trade-off between power consumption and I/O scheduling flexibility. Incorporating parallelism along this dimension requires replicating the VCM and the arms, but not the spindle assembly. Because the VCM's average power is typically much lower than the SPM power,[9] we can boost performance by incorporating additional arm assemblies without significantly increasing the power consumption. Because our goal is to minimize rotational latency, we used the shortest-positioning time first (SPTF) scheduling policy at the disk. With multiple actuators, the SPTF-based disk arm scheduler has the flexibility to choose the arm assembly that minimizes the overall positioning time for a particular I/O request.

We evaluated the behavior of a multiactuator disk drive design HC-SD-SA(n), which is an instance of $D_1A_nS_1H_1$ in our taxonomy. 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 in time, only a single arm (SA) assembly can be in motion and 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 that disk request's positioning time.

We also evaluated two extensions to this design, in which we relaxed the two restrictions imposed by this design. Our first extension allowed multiple arms to be in motion simultaneously and the second extension allowed multiple channels to transfer data simultaneously. We found that these two extensions provide little benefit over the HC-SD-SA(n) design. In our evaluation, we vary the number of arm assemblies n from 1 to 4.

Performance behavior. Figure 4a gives the CDFs of the HC-SD-SA(n) design's response time, along with those of the corresponding MD systems. We compared the performance of the HC-SD-SA(n) design points for each workload to its corresponding MD system. To quantify the designs' impact on rotational latency, we plot the probability density function (PDF) of the I/O requests' rotational latencies (see Figure 4b).

As the response time CDFs show, the HC-SD-SA(n) design provides substantial performance benefits compared to HC-SD. Because multiple arms 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 significantly boosts response times. The performance of these two workloads on HC-SD-SA(2) nearly matches that of their MD counterpart. TPC-H also slightly improves response time, allowing 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. We can see from the PDF graphs for Websearch, TPC-C, and TPC-H that increasing the number of arms from one to two substantially shortens the tail of the distributions from a higher to a lower range of rotational latencies. Using a third disk arm creates a similar shift in Financial's rotational-latency distribution. However, increasing the number of arms beyond that diminishes performance returns, which the closeness of the HC-SD-SA(3) and HC-SD-SA(4) curves in both the CDF and PDF graphs demonstrates.

Our bottleneck analysis revealed that a 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. Figure 4 shows that the HC-SD-SA(n) design provides these performance benefits for Websearch, TPC-C, and TPC-H. This result indicates that an intradisk 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 we noted previously, the load on the HC-SD system is relatively light, so using intradisk parallelism doesn't significantly improve performance.

Power behavior. Although HC-SD-SA(n) drives use multiple actuators, because only one VCM is active at any given time, these drives' peak power consumption will be comparable to conventional disk drives. Peak power consumption is important for the disk drive designer, who must design the drive to operate within a certain power and

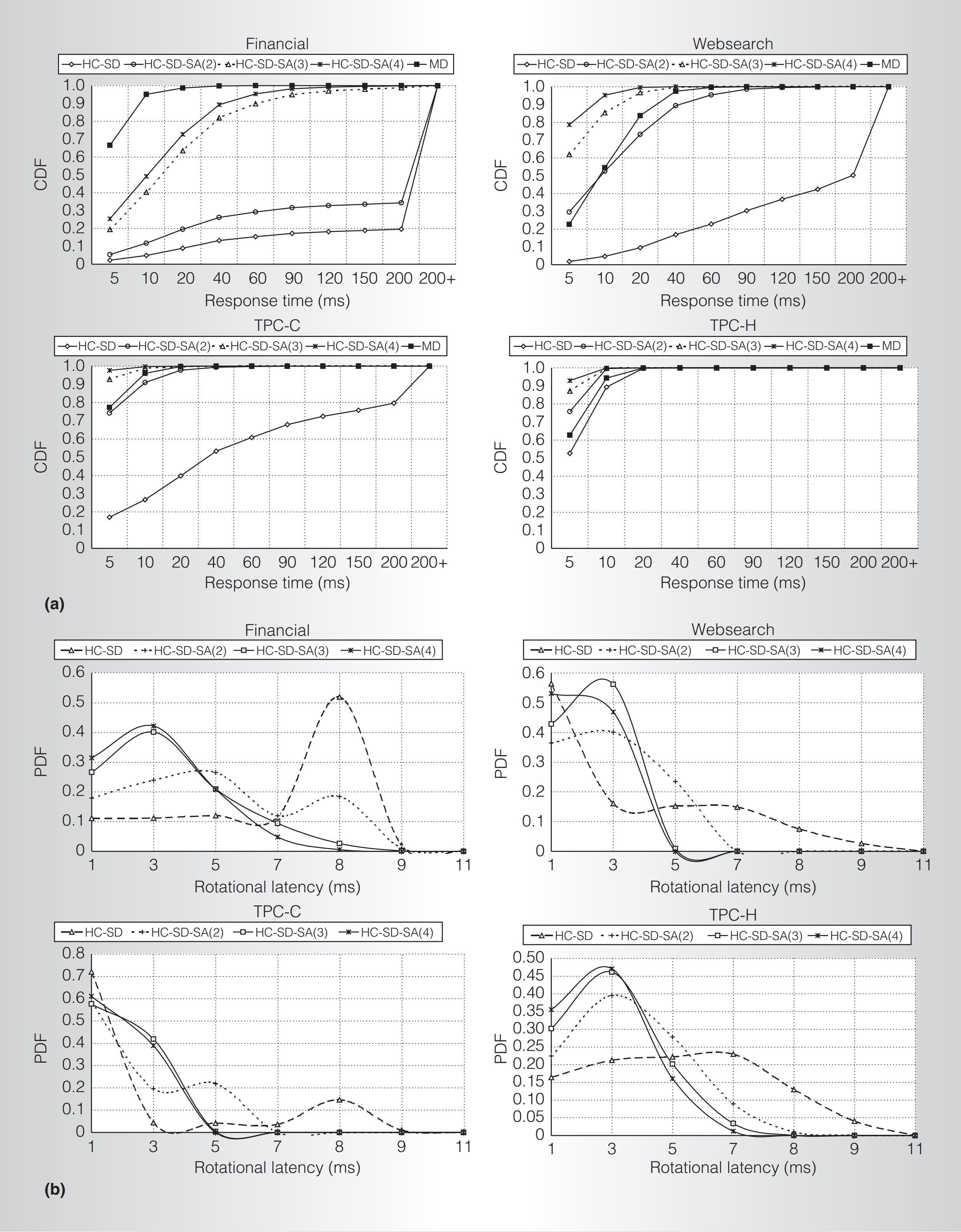

Figure 4. Performance impact of the high-capacity single drive with one or more additional arms (HC-SD-SA(*n*)) design for each workload's cumulative distribution function (CDF) (a) and probability density function (PDF) (b).

thermal envelope for reliability purposes.[4] However, from an operating cost perspective, the average power of intradisk parallel disks should ideally be comparable to conventional drives as well.

Generally, the HC-SD-SA(*n*) drives have higher average power consumption than HC-SD due to increased seek activity (that is, due to the VCM), even though seeks aren't the primary performance bottleneck. One way to reduce an intradisk parallel drive's power consumption is to design it to operate at a lower RPM. Because RPM has nearly a cubic impact on a disk drive's power consumption,[6] a lower RPM design would consume less power. On the other hand, lowering the RPM would tend to increase the rotational latency. However, using multiple actuators can offset the extent to which the reduction in RPM impacts I/O response time. To determine how these factors interact, we analyzed the power and performance of three lower RPM design points for HC-SD-SA(*n*) with 6,200, 5,200, and 4,200 RPM, respectively. We found several design points where we can match or surpass the multidisk system's performance, consume an order of magnitude less power than MD, and consume power that is close to or less than that of a single conventional disk drive.

Using intradisk parallel drives to build RAID arrays

For I/O-intensive workloads, a single intradisk parallel drive might not be sufficient to meet performance goals. This naturally raises the question whether we should opt for a RAID array made up of conventional disk drives or an array consisting of intradisk parallel drives. We explore this issue by comparing the performance and power characteristics of these two types of RAID arrays. We consider conventional and intradisk 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.

Because we wanted to study the trade-offs between the two types of storage systems for a range of I/O intensities, we used the synthetic workload generator in Disksim to create workloads with one million I/O requests for this experiment. For all the synthetic workloads, 60 percent of the requests were reads and 20 percent were sequential. We varied the interarrival 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's a steady stream of requests arriving at the storage system. We varied the distribution mean and considered 8-, 4-, and 1-ms interarrival time values, which represent light, moderate, and heavy I/O loads, respectively. We evaluated 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 intradisk parallel drives (the HC-SC-SA(2) and HC-SD-SA(4) configurations).

Figure 5 gives the experiment's results. Figures 5a through 5c give the performance characteristics under each interarrival time scenario for disk arrays 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—that is, maximum response times incurred by 90 percent of the requests in the workload. Figure 5d shows the HC-SD-based disk array's average power consumption 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 HC-SD array's steady-state performance.

Figure 5 shows a clear performance advantage for intradisk parallelism. For the relatively light 8-ms interarrival 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 it takes four HC-SD drives to get performance that is comparable to the two-disk HC-SD-SA(2) array. A single four-actuator drive breaks even with the performance of the four-disk HC-SD and two-disk HC-SD-SA(2) arrays, respectively.

From the power perspective, the array of conventional disks consumes 61.4 W, whereas the HC-SD-SA(2) and HC-SD-SA(4) arrays consume 37.1 and 26.2 W of power, respectively. Under moderate and heavy I/O loads (see Figures 5b and 5c, respectively), the intradisk parallel drives can mitigate the I/O

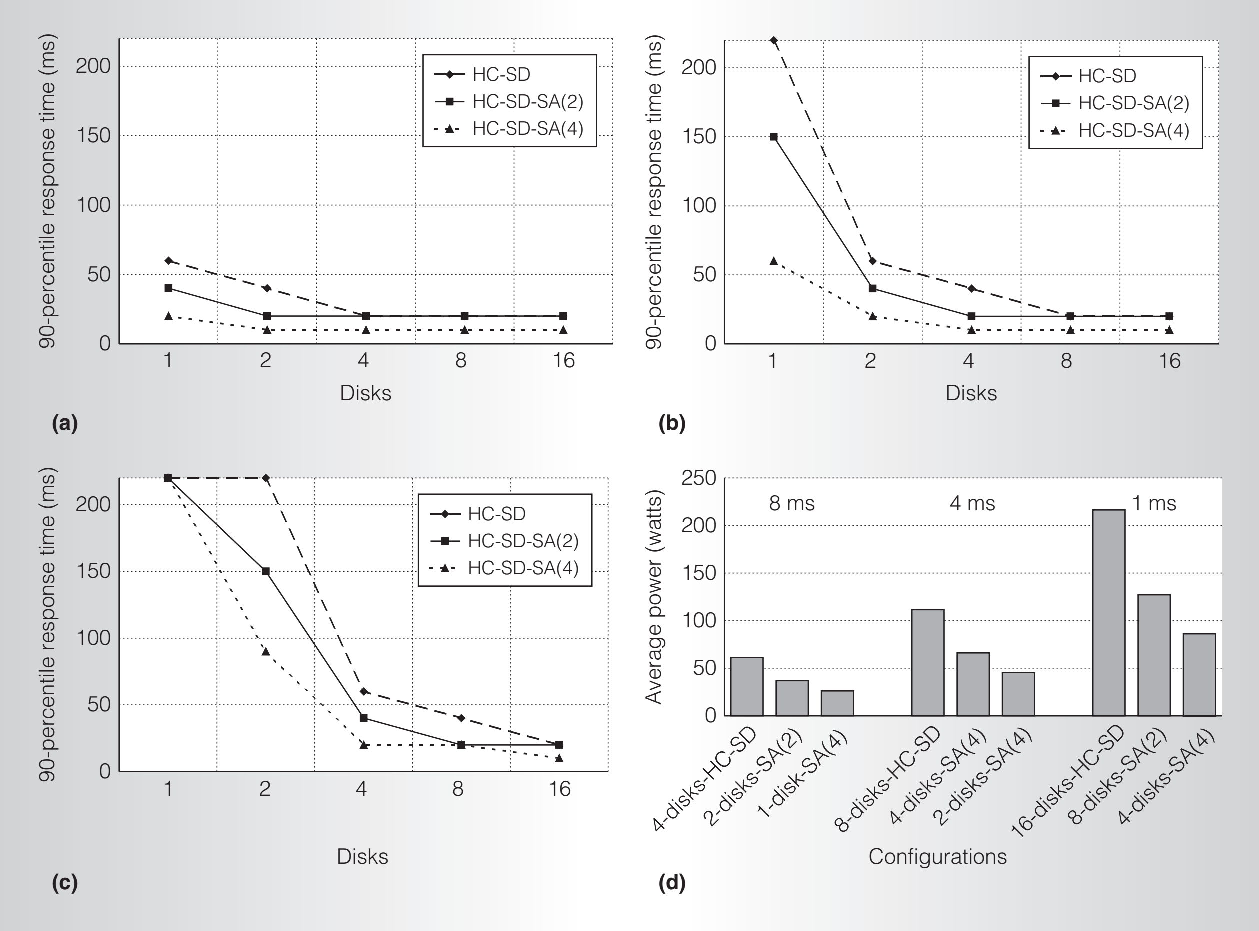

Figure 5. Performance of RAID arrays using intradisk parallel drives with an interarrival time of 8 ms (a), 4 ms (b), and 1 ms (c) as well as power characteristics for an iso-performance power comparison (d).

bottlenecks with fewer disks than arrays composed of conventional disk drives. For the 1-ms interarrival time workload, the ratio of the number of intradisk parallel drives to conventional drives needed to break even in performance is the same as under lighter loads. However, because we need 16 conventional disks to break even with the performance of an eight-disk HC-SD-SA(2) and four-disk HC-SD-SA(4) array, the average power consumption of the intradisk-parallel-drive-based arrays are lower. The HC-SD-SA(2) and HC-SD-SA(4) arrays consume 41 and 60 percent less power than the HC-SD-based array, respectively.

These results clearly indicate that using intradisk parallel drives is more attractive, in terms of both performance and power, than using conventional disks to build RAID arrays for I/O-intensive workloads.

Cost-benefit analysis

We can obtain the performance and power benefits of intradisk parallel drives by extending conventional disk drive architectures with additional hardware. The next step is to ask whether it's worth spending more money on a single intradisk parallel drive than on multiple conventional drives. A preliminary cost estimate of manufacturing intradisk parallel drives, using cost data we obtained from several companies within the disk drive industry, sheds some light on this question.

Building a disk drive involves material and labor costs as well as other overhead. Studies about the disk drive industry have shown that materials account for the bulk of a disk's manufacturing costs,[11,12] so we focus on quantifying these costs. Many of the components that go into a disk drive

Table 2. Estimated component and disk-drive costs (in US dollars).

Component name	Component	Conventional disk drive	Two-actuator disk drive	Four-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

are manufactured by different companies, each of which specializes in making a particular component, such as a head or a pivot bearing, and supply their components to disk-drive companies on a volume basis. To estimate each component's cost, we contacted several major component manufacturers to obtain prices for components supplied to disk-drive companies for their server hard drives. Sometimes manufacturers gave us a single value, and other times a price range. Table 2 lists cost estimates of several key disk-drive components. A component's exact price depends 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.

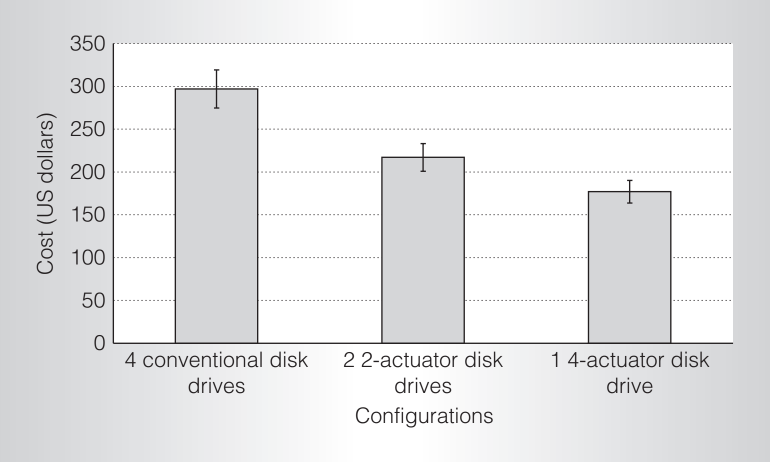

Figure 6. Iso-performance cost comparison between conventional and intradisk parallel drives. The error bars give the cost range based on the values in Table 2.

Also, we assumed that the material costs for building a disk drive and the product's final cost are related and that a rise or fall in the manufacturing costs will translate to similar effects on the drive's market price.

Using the provided per-component cost estimates, we calculated the material costs for a conventional disk drive, a two-actuator intradisk parallel drive, and a four-actuator drive. For consistency, we calculated the cost for a four-platter drive. Figure 6 shows the costs of the three storage system configurations that deliver equivalent performance, based on our earlier results. We depict the low-to-high cost range using error bars.

Table 2 indicates that the bulk of the cost increase for building intradisk parallel drives will likely be in the heads. Other components, such as the VCMs and their motor drivers, head suspensions, pivot bearings, and head preamplifiers, constitute only a small part of the overall cost of an intradisk parallel drive. However, the overarching question is whether this increased cost (and its corresponding higher selling price) would be worth the investment for the product's eventual customer. As Figure 6 indicates, two HC-SD-SA(*n*) intradisk parallel drives deliver equivalent performance to four conventional disk drives, but at 7 percent lower cost. One four-actuator drive delivers the same performance, but at 40 percent lower cost than the four-disk array of conventional drives.

The results of our performance, power, and cost-benefit analysis for the multi-actuator intradisk parallel drive are encouraging and should motivate researchers to explore intradisk parallelism further. There are several other points in the DASH taxonomy with performance, power, and cost characteristics that require further study. There are also opportunities to explore how we can use intradisk parallelism in conjunction with solid-state disks to build large high-performance, energy-efficient storage systems for data centers. MICRO

Acknowledgments

This research was supported in part by the US National Science Foundation Career award CCF-0643925, NSF grant CNS-0551630, a Marco IFC grant, and gifts from HP and Google.

References

1. J.F. Gantz et al., ''The Expanding Digital Universe: A Forecast of Worldwide Information Growth through 2010,'' white paper IDC, 2007.
2. D.E. Shasha and P. Bonnet, *Database Tuning: Principles, Experiments, and Troubleshooting Techniques,* Morgan Kauffman, 2003.
3. M. Ault, ''Tuning Disk Architectures for Databases,'' *DBAzine.com,* Jun. 2005; www.dbazine.com/oracle/or-articles/ault1.
4. S. Gurumurthi, A. Sivasubramaniam, and V. Natarajan, ''Disk Drive Roadmap from the Thermal Perspective: A Case for Dynamic Thermal Management,'' *Proc. Int'l Symp. Computer Architecture* (ISCA 05), IEEE CS Press, 2005, pp. 38-49.
5. S. Sankar, S. Gurumurthi, and M.R. Stan, ''Intra-Disk Parallelism: An Idea Whose Time Has Come,'' *Proc. Int'l Symp. Computer Architecture* (ISCA 08), IEEE CS Press, 2008, pp. 303-314.
6. I. Sato et al., ''Characteristics of Heat Transfer in Small Disk Enclosures at High Rotation Speeds,'' *IEEE Trans. Components, Packaging, and Manufacturing Technology,* vol. 13, no. 4, 1990, pp. 1006-1011.
7. R. Youssef, ''RAID for Mobile Computers,'' master's thesis, Information Networking Inst., Carnegie Mellon Univ., 1995.
8. G.R. Ganger, B.L. Worthington, and Y.N. Patt, ''The DiskSim Simulation Environment Version 2.0 Reference Manual,'' www.ece.cmu.edu/ganger/disksim.
9. Y. Zhang, S. Gurumurthi, and M.R. Stan, ''SODA: Sensitivity Based Optimization of Disk Architecture,'' *Proc. Design Automation Conf.* (DAC), ACM Press, 2007, pp. 865-870.
10. M.H. Kryder, ''Future Storage Technologies: A Look beyond the Horizon,'' *Proc. Computerworld Storage Networking World Conf.* (SNW 06), 2006; http://www.snwusa.com/documents/presentations-s06/MarkKryder.pdf.
11. S.M. Hampton, *Process Cost Analysis for Hard Disk Manufacturing,* tech. report 96-02, Information Storage Industry Center, Univ. of California, San Diego, 1996.
12. R.E. Bohn and C. Terwiesch, ''The Economics of Yield-Driven Processes,'' *Elsevier J. Operations Management,* vol. 18, no.1, 1999, pp. 41-59.

Sudhanva Gurumurthi is an assistant professor in the Computer Science Department at the University of Virginia. His research interests include energy-efficient data center architectures and silicon reliability. Gurumurthi has a PhD in computer science and engineering from Penn State University. He is a member of the ACM and the IEEE.

Sriram Sankar is a member of the Hardware Performance and Standards team at Microsoft. His research interests include computer architecture and storage systems. Sankar has an MS in computer science from the University of Virginia. He is a member of the ACM.

Mircea R. Stan is a professor in the Charles L. Brown Department of Electrical and Computer Engineering at the University of Virginia. His research interests include high-performance and low-power VLSI, temperature-aware circuits and architecture, and embedded systems. Stan has a PhD in electrical and computer engineering from the University of Massachusetts at Amherst. He is a member of the ACM and the Institution of Engineering and Technology and a senior member of the IEEE.

Direct questions and comments about this article to Sudhanva Gurumurthi, Dept. of Computer Science, Univ. of Virginia, Charlottesville, VA 22904; gurumurthi@cs.virginia.edu.