Appears in Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems, 2000

Designing Computer Systems with MEMS-based Storage

Steven W. Schlosser, John Linwood Griffin, David F. Nagle, Gregory R. Ganger

Carnegie Mellon University

{schlos, griffin2}@ece.cmu.edu, bassoon@cs.cmu.edu, ganger@ece.cmu.edu

ABSTRACT

For decades the RAM-to-disk memory hierarchy gap has plagued computer architects. An exciting new storage technology based on microelectromechanical systems (MEMS) is poised to fill a large portion of this performance gap, significantly reduce system power consumption, and enable many new applications. This paper explores the system-level implications of integrating MEMS-based storage into the memory hierarchy. Results show that standalone MEMS-based storage reduces I/O stall times by 4–74X over disks and improves overall application runtimes by 1.9–4.4X. When used as on-board caches for disks, MEMS-based storage improves I/O response time by up to 3.5X. Further, the energy consumption of MEMS-based storage is 10–54X less than that of state-of-the-art low-power disk drives. The combination of the high-level physical characteristics of MEMS-based storage (small footprints, high shock tolerance) and the ability to directly integrate MEMS-based storage with processing leads to such new applications as portable gigabit storage systems and ubiquitous active storage nodes.

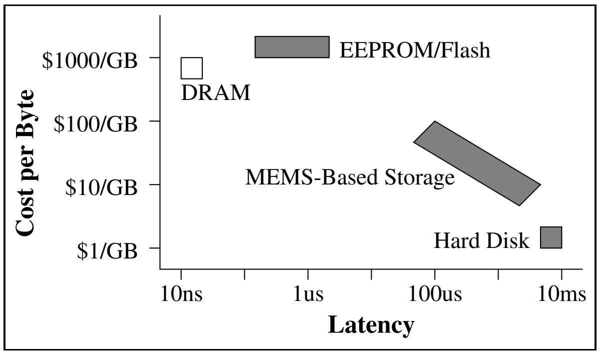

Figure 1: ***Predicted cost and latency for storage technologies in 2005.*** *MEMS-based storage fills the growing memory hierarchy gap between RAM and disk (grey boxes represent nonvolatile storage). EEPROM's wide box spans the gap between its read and write latencies. The wide and tall MEMS-based storage box represents the many design possibilities for this new technology (discussed in Section 2).*

1. INTRODUCTION

For decades, the memory hierarchy has suffered from significant latency, bandwidth and cost gaps among the processor, RAM, and disk [29]. Although the processor-to-RAM performance gap has been mitigated by fast cache memories [28], the RAM-to-disk gap has remained unfilled, widening to 6 orders of magnitude in 2000 and continuing to widen by about 50% per year. The result is a significant performance and scalability problem across a range of applications: databases, web servers, mail servers, software development tools, even Microsoft Word load times [6].

This RAM-to-disk performance gap is due to the physical characteristics of disk drives—although disks continue to deliver capacity growth of over 60% per year, their mechanical positioning systems limit access time improvements to only 7% per year [28]. EEPROM technologies (such as Flash RAM or Memory Sticks [1]) offer a high-performance nonvolatile secondary storage alternative to disks, but their current and future cost per megabyte remains 2 orders of magnitude higher than disk storage (Figure 1).

Microelectromechanical systems (MEMS)-based storage is an exciting new technology that could provide significant performance gains over current disk drive technology at costs much lower than EEPROM [3,4]. MEMS-based storage is a nonvolatile storage technology that merges magnetic recording material with thousands of probe-based recording heads to provide on-line storage capacity of 1–10 GB of data in under 1 cm^2 of area. Simulation shows these devices have access times of 0.5–1.1 ms and streaming bandwidths up to 320 MB/s.

Further, integrating MEMS-based storage with processing elements lays the foundation for a single computing "brick" containing processing, volatile primary storage and nonvolatile secondary storage [12]. As MEMS-based storage devices are built with traditional low-cost VLSI-style parallel lithographic manufacturing processes [10], the cost of integrating processing elements with MEMS-based mass storage on the same chip could prove significantly less than an equivalent nonvolatile RAM solution [4]. Several microprocessors or hundreds of custom computational engines (*e.g.*, MPEG encode/decode, cryptography, signal processing) fabricated directly with MEMS-based storage could significantly improve performance, power consumption, and cost over traditional multicomponent solutions.

Although MEMS-based storage is still several years away from commercialization, its potential impact on the memory gap makes this technology both important and interesting for systems architects' consideration. Following on our previous work [13] of developing a performance model for MEMS-based storage and examining its basic behavior and raw performance (average access time, maximum read/write bandwidth, *etc.*), this work examines the integration of MEMS-based storage into the memory hierarchy. We examine the impact on performance and power consumption for two different uses of MEMS-based storage: as a replacement for disk drives and as a nonvolatile cache embedded within a conventional disk drive's electronics.

Our results show that replacing disks with MEMS-based storage reduces application I/O stall times by 4–74X for a set of five file system and database workloads. Application speedups range from 1.9–4.4X. Power simulations predict energy reduction by a factor of up to 54X over state-of-the-art low-power disk drives. Combined with the expected better shock tolerance and higher reliability, this makes MEMS-based storage technology an excellent high-capacity storage solution for mobile, low-power applications.

To ensure that our models accurately reflect potential implementations, we are working closely with researchers at the Center for Highly Integrated Information Processing and Storage Systems at Carnegie Mellon [5] who are actively developing practical MEMS-based storage devices. This collaboration allows us to explore the system-level impact of various MEMS-based storage designs by determining which physical design trade-offs are most important to application performance. Our feedback allows these researchers to focus their attention on design parameters that significantly impact system-level performance, while avoiding optimizations that provide little practical benefit.

The remainder of this paper is organized as follows. Section 2 describes MEMS-based storage and many of the physical design trade-offs of the devices. Section 3 describes our experimental setup. Section 4 presents results from a number of application studies. Section 5 discusses more general system-level issues and explores a wide range of applications for MEMS-based storage. Section 6 draws conclusions and discusses continuing work.

2. MEMS-BASED STORAGE

2.1 High-level Device Design

MEMS are very small-scale mechanical structures—on the order of tens to thousands of microns—fabricated on silicon chips using photolithographic processes much like those employed in manufacturing standard semiconductor devices. MEMS structures can be made to slide, bend, or deflect in response to an actuator's electrostatic or electromagnetic force or external forces. MEMS machines have interesting strengths and limitations compared to standard mechanical systems. For example, large-aspect-ratio cantilever designs that would fail under load when built at the macroscopic scale can be built reliably on the microscopic scale. As a counterexample, it is difficult to build durable microbearings for rotating components—prototypes of micromachined gear trains have locked up from friction within several thousand revolutions. Because of this limitation it is difficult to replicate disk-based storage designs on the microscopic scale. Alternative designs, such as rectangular spring-suspended masses (*media sleds*) that translate two-dimensionally (instead of rotating about an axis), circumvent this frictional barrier and are proving to be mechanically robust.

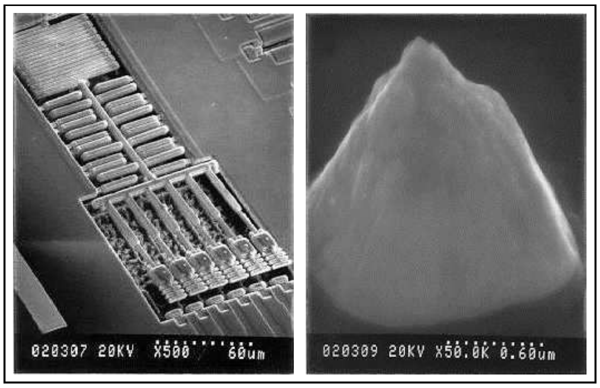

Figure 2: ***Prototype positioning system and probe tip.*** *Because the recording material is not perfectly flat, the positioning system must be able to actively adjust the height of the probe tips. The tips could use one of several recording schemes, from simple "typewriting" with permanent magnets, to more complex magnetoresistive sensing techniques found in normal disk drives.*

One class of MEMS-based storage device under investigation employs an array of thousands of cantilevered magnetic read/write heads (*probe tips*, shown in Figure 2), each accessing a dense substrate of magnetic material in much the same way disk heads access magnetic platters [3,4]. This design offers notable advantages over disk-based storage along several axes, including access time, device size and mass, energy consumption, cost, failure modes, and sensitivity to shock. Multiple probe tips can concurrently access the media to achieve one of several forms of parallelism: all tips can be used to access data (to increase throughput); some tips can be used for error detection and correction (to enhance reliability); or completely independent accesses can proceed in parallel. In addition, the MEMS fabrication process can be integrated with standard CMOS processes [10], opening the door to combine processing and nonvolatile storage for large-scale manufacturing of system-on-a-chip architectures.

MEMS microstructures can be used to build storage devices in a variety of ways—design decisions affect the manufacturability, robustness, cost, capacity, access speed and latency of these devices. Figure 3 depicts one proposed MEMS-based storage design. In this "fixed media" model, miniature cantilevered L-shaped beams suspend a probe tip over a fixed magnetic substrate. Voltages applied to deflectors generate electrostatic forces in the X and Y directions, rapidly moving the tip to different bit positions. Standard magnetic recording techniques are used to read or write the bits, with the same unlimited number of read and write cycles as found in disk drives. The nearly-massless cantilevered beam enables very quick positioning times (on the order of tens to hundreds of microseconds) but the space efficiency

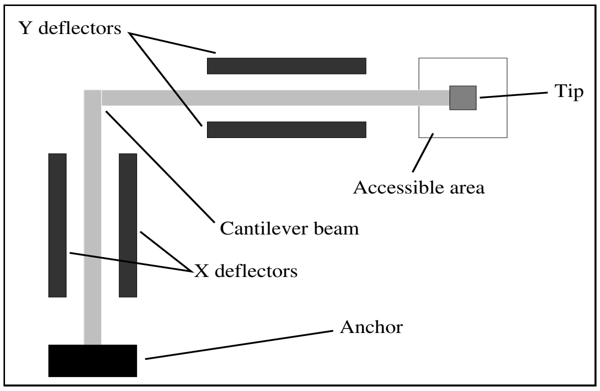

Figure 3: ***A cantilevered-beam probe tip in the "fixed media" model.*** *The X- and Y-deflectors are capable of quickly positioning the tip anywhere in the small accessible area. The overall capacity of this model is limited to tens or perhaps hundreds of megabytes because only 1% of the media area is accessible by the tip.*

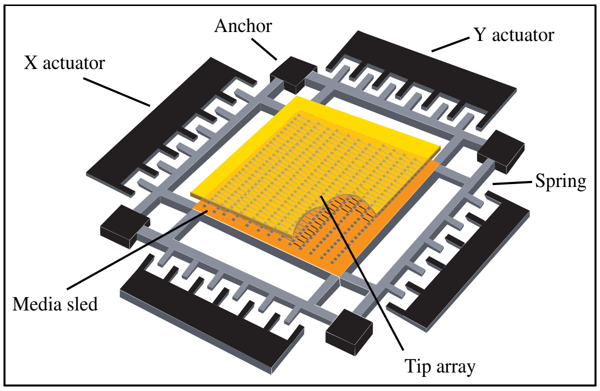

Figure 4: ***The "moving media" model.*** *The media sled is attached below the fixed tips. The sled can move along the X and Y axes, allowing the fixed tips to address 30–50% of the total media area.*

is poor—only about 1% of the potential media area can be used for storage. In comparison, conventional disk drives use about 50% of their platter area for data storage. This design is useful for visualizing MEMS-based storage, but its expected capacity of only tens to hundreds of megabytes per device limits its practicality in comparison to Flash RAM, battery-backed RAM, and other nonvolatile primary storage components.

Researchers at Carnegie Mellon are investigating a more media efficient device design (Figure 4). In this "moving media" model, a rectangular media sled is suspended by springs above an array of several thousand fixed probe tips. A device's footprint is about 14×14 mm, with a usable area on the media sled of about 8×8 mm. Up to 10,000 tips can be fabricated over this 8×8 mm area. Assuming a bit cell of 0.0025 μm^2 (50 nm per side) and encoding/ECC overheads of 2 bits per byte, a device's data storage capacity is about 4 GB [4]. A more aggressive goal of 0.0009 μm^2 (30 nm per side) yields capacities of 11 GB or greater. While this device design improves space efficiency to 30–50%, the greater sled mass increases positioning times relative to the fixed media design above—a necessary tradeoff to achieve disk-like capacities. For a more thorough description of the characteristics of this design see References [4] and [13].

2.2 Data Layout and Access Characteristics

The sled's magnetic media is organized into rectangular regions as shown in Figure 5. Each region stores M×N bits (*e.g.,* 2000×2000). There is a one-to-one mapping between regions and tips; each tip accesses its exclusive region of the media. Bits within a region are grouped into vertical 90-bit columns called *tip sectors*; each tip sector contains 10 bits of sled positioning information and 80 encoded data bits providing 8 data bytes. The 8-byte tip sector is the smallest accessible unit of data in MEMS-based storage. Groups of 64 tip sectors from separate regions may be combined into 512-byte *logical sectors*, analogous to logical blocks in SCSI disks. This striping is both possible and practical because, unlike most conventional disks, large numbers (200–2000) of probe tips can simultaneously access the media. Striping logical blocks across tip sectors in multiple regions reduces access time and increases bandwidth, reliability, and fault tolerance.

To access data, electrostatic actuators (capacitive comb fingers) pull the sled to a certain *x,y* offset—positioning the tips above an exact location on the media by moving the media—then drag the sled such that each active tip reads or writes an entire tip sector (*i.e.,* such that groups of tips access whole logical sectors). As in the earlier design, the probe tips read or write data using standard magnetic recording techniques.

Positioning the sled for read or write involves several mechanical and electrical actions. To seek to a desired sector, the appropriate probe tips must be activated, the sled must be positioned so the tips are above the first bit of the presector servo information, and the sled must be moving in the correct direction at the correct velocity for access. Whenever the sled moves in X, an extra constant *settling time* must be taken into account—the rapid acceleration and deceleration of the sled causes the spring-sled system to momentarily oscillate in X before damping to zero X motion. (The sled also oscillates in Y; however, the magnetic sensing logic is expected to compensate for this motion.) In addition, the springs apply a restoring force toward the "sled-at-rest" position, increasing or decreasing the effective sled actuating force by as much as 75%.

Media access requires that the sled move at constant velocity in the Y direction. This *access velocity* is a design parameter and is determined by the maximum per-tip read and write rates, the bit width, and the maximum sled acceleration. Large transfers could span multiple columns of bits, requiring the sled to perform a *turnaround* (reversing direction such that the sled ends up in the same position at reverse velocity) and switch the set of active tips. The turnaround time is expected to dominate any additional activity, such as the time to switch which tips are active.

2.3 Other MEMS-based Storage Examples

There are several other research laboratories pursuing probe-based storage research projects, including IBM [8, 34, 38], Hewlett-Packard [36], Kionix [7], and Nanochip [27]. IBM's

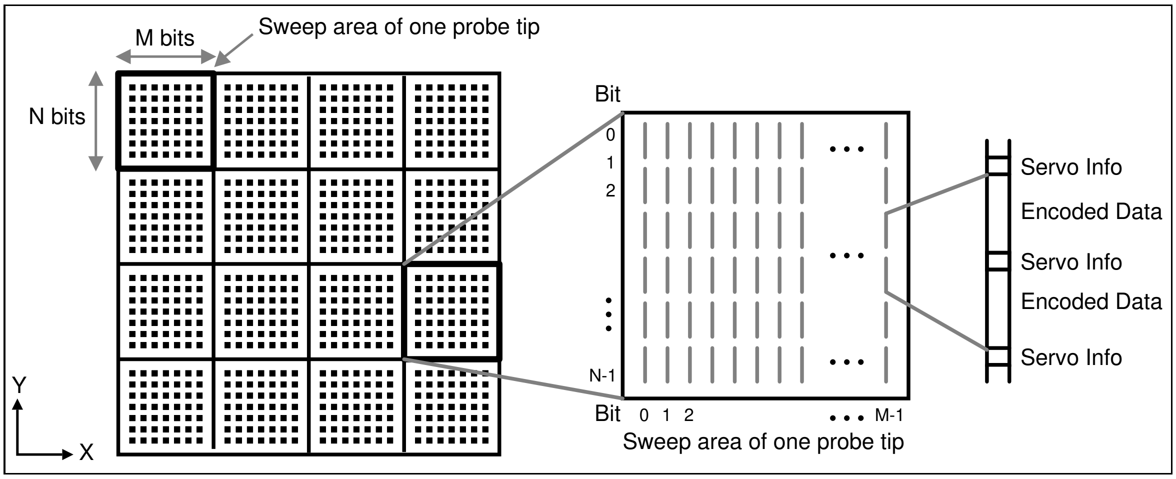

Figure 5: ***Data organization of MEMS-based storage.*** *The illustration depicts a small portion of the magnetic media sled. Each rectangle outlines the region accessible by a single probe tip, with a total of 16 regions shown. (A full device contains thousands of tips and regions.) Each region stores $M \times N$ bits, organized into vertical "tip sectors" containing encoded data and ECC bits. These tip sectors are demarcated by "servo information" strings that identify the sector and track information encoded on a disk. This servo information is expected to require about 11% of the device capacity. To read or write data, the media passes under the active tip(s) in the $\pm Y$ direction while the tips access the media.*

initial efforts employed cantilevered probe tips that melted pits into a rotating polymer disk. IBM's more recent Millipede project has built a working prototype, combining a moving media sled with a 32×32 atomic force microscope-based probe tip array. Two startup companies, Kionix and Nanochip, are also developing probe-based magnetic storage architectures with moving media. Kionix uses atomic force microscope-based tips while Nanochip employs small scanning tunneling microscope-based tips.

While these designs differ significantly in their read/write mechanisms, most use a similar media sled design[1]. Further, IBM's and Nanochip's design parameters and performance data closely match our own results. All achieve about the same bit density of approximately 250 Gbit/in^2, sweep the sled about ± 50 μm, and can seek in under 0.5 μs. However, aggregate data rate varies significantly, with Nanochip supporting 3.1 MB/s across 400 probe tips while IBM supports 250 MB/s across 1000 probe tips. Both Nanochip and IBM currently require a separate erase cycle before writing new data. While differences exist between the various project designs, most employ a similar storage architecture, with either a media sled or a large group of probe tips moving in the X and Y directions. Therefore, this study uses the Carnegie Mellon MEMS Laboratory's moving media model as a basis for quantitative analysis of future MEMS-based storage systems.

2.4 Physical Characteristics and Trends of MEMS-based Storage

MEMS-based storage devices have a rich set of physical characteristics (*e.g.,* acceleration, access velocity) and architectural characteristics (*e.g.,* layout of data, number of sleds) that directly impact the capacity, bandwidth, latency, reliability, and power consumption of this new technology.

As would be expected, the physical characteristics of MEMS-based storage often constrain architectural designs. For example, packaging and power dissipation constraints limit the number of tips that can be simultaneously active. A recent analysis [4] estimates power consumption at 1 mW per active tip plus 100 mW of overhead for the media positioning system. In a design with 10,000 probe tips, using all of the tips simultaneously consumes 10.1 W—about 10X more power than low-cost plastic packaging can tolerate.

Table 1 outlines MEMS-based storage's different physical parameters that are most important from an architectural point of view. The following paragraphs describe each parameter, the technology trends that enable improvements (for example, decreasing bit sizes), and their relationship to device performance characteristics.

Bit size. The bit size is determined by the areal density of the storage media and the resolution of the probe tips. The media sled's magnetic recording film is similar to that of current disk drive media, where bit densities of over 50 Gbit/in^2 [20] and annual growth rates of 100% per year [26] have been observed. Unlike disk drives, which use longitudinal recording techniques to write rectangular bits with a 16:1 aspect ratio, MEMS-based storage uses perpendicular recording techniques that create square bit spots. Future disk drives might also utilize this technique to achieve higher media densities. The finer positioning resolution of the MEMS actuators, however, will allow MEMS-based storage devices to access smaller spots than disks, leading to even greater bit densities.

Access Velocity. The access velocity of the media sled is bounded by the effective actuator force. The effective actuator force is the sum of the force from the actuators and the restoring force of the springs. The limit manifests itself

[1] We believe that Hewlett-Packard is pursuing a similar device design; unfortunately, no project details are available at the time of this publication.

	seek time	settle time	turnaround time	peak bandwidth	capacity	power	reliability
decreasing bit size		+		+	+	+	–
increasing sled access velocity	–		+	+/–	–	+	
increasing sled acceleration							
increasing actuator force	–	–	–	+	+	+	+
decreasing sled mass	–	–	–			–	+
increasing spring stiffness	+/–	–	+/–	+			+
increasing # of sleds	–	–	–	+	–		+
increasing # of active tips				+		+	
increasing error rate				–	–	+	–

Table 1: *MEMS-based storage devices' physical characteristics, their projected trends, and projected impact on device performance.* *Decreases in performance are denoted with a "–" while increases are denoted by "+". For example, decreasing bit size, which is made possible by technology advances in magnetic materials, could increase the settle time because it will take longer to position the tip over a smaller bit.*

in three ways. First, a higher access velocity will require that the sled accelerate for a longer period of time to ramp up. If too much time is spent accelerating to the access velocity, regions of the media will become inaccessible. Second, as the access velocity increases, the time to turn the sled around at the end of the tracks increases. In this way, higher access velocities are not useful because the sled will lose too much time turning around. This is explored more thoroughly in [13]. Lastly, while the sled is turning around it may pass too far beyond the end of a track and crash into the actuators, causing damage to the device.

Sled Acceleration. The maximum sled acceleration could be increased in three ways: (1) increasing the effective actuator force by increasing the voltage supplied to the positioning system or increasing the spring stiffness, (2) decreasing the sled mass (which will become possible as manufacturing technology evolves to allow full-strength hollowed-out mechanical structures), (3) employing alternate actuator designs, such as IBM's micromagnetic actuator [8, 34].

Spring stiffness. Spring stiffness is determined by the thickness of the spring beams. A certain amount of spring stiffness is necessary to support the media sled and to avoid damage from the various forces encountered during manufacturing (chip assembly) and common external shocks (shipping, normal use in mobile environments). The maximum spring restoring force increases as the spring stiffness increases, and cannot exceed the available actuator force.

Number of sleds. Increasing the number of sleds is an architectural design choice. Instead of manufacturing one large sled across all of the probe tips, it should be straightforward to create four independent sleds, for example, each with their own actuators. Although this might decrease per-device capacity (because of the space overhead of the extra actuators) it will likely improve request service time by improving internal parallelism. As an alternative, the multiple sleds could be used in RAID-style redundancy schemes, improving fault tolerance and reliability of the MEMS-based storage device.

Number of active tips. Increasing the number of concurrently active tips is an architectural and cost design choice. As mentioned above, 10,000 simultaneously active tips would consume 10.1 W—an order of magnitude more power than low-cost plastic packaging can tolerate, requiring more expensive packaging technologies capable of dissipating more heat.

Error rates. Error rates are a property of the manufacturing process and the magnetic materials used. In disk drive design, raw media error rates increase with higher areal densities and are compensated for by using more powerful error-correction codes. MEMS-based storage can benefit in the same way. Further, it can use RAID-like error detection and recovery across probe tips—even as a compliment to the multiple-sled (or multiple-device) RAID schemes mentioned above.

2.5 Performance Characteristics and Trends

Table 1 also highlights how the interaction between physical parameters and overall device performance creates an interesting set of relations and trade-offs. This section outlines a basic model of how several design choices impact the various performance parameters.

Seek time. The second column of Table 1 shows the impact of the physical parameters on seek time (sled positioning time in X and Y). Increasing the sled's access velocity increases the Y-direction seek time by increasing the time required to "ramp up" to the access velocity (after the sled performs a turnaround). X-direction time does not change because the initial and final velocity in X is zero. Seek time will decrease as acceleration increases, due to either increasing actuator force or decreasing sled mass.

With increasing spring force, the impact on seek time is dependent on the initial and final sled locations. For example, if the sled is near the edge of the media (*i.e.*, close to full displacement), the spring force is near its maximum, pulling the sled toward the center while the actuator force is pulling the sled towards the edge. Since the spring force at maximum displacement is predicted to be up to 75% of the actuator force, the effective actuator force when moving away from the center is only 25% at full displacement. Likewise, the effective force when moving towards the center can be 175%. This means that a short seek towards the center will be able to accelerate quickly (with 1.75X the actuator force), but will have only 1/4 the force available to decelerate. Note that if the seek is longer, the spring forces help decrease seek time. For example, if the seek is from one end of the device to the other, the sled will effectively accelerate and decelerate with 175% of the actuator force. In this case, seek time decreases with increasing spring stiffness.

Settle time. If the sled employs an active damping system to damp sled vibrations, stronger actuator forces will dampen the spring-sled system more quickly, directly decreasing settling time. However, decreasing the bit size requires longer damping times, in turn increasing settle time.

Turnaround time. Turnaround time decreases with increasing effective actuator force. The extra force increases the rate of deceleration and acceleration (*i.e.,* allowing the sled to stop and then start moving in the opposite direction more quickly). In contrast, increasing the sled's velocity directly increases the turnaround time. Increasing the stiffness of the springs improves turnaround time whenever the sled is initially moving in opposition to the spring force. The best case is when the sled is moving towards the device edge and then turns around. Here, the spring force pulls the sled toward the center, benefiting both stopping and restarting the sled. Even if the sled is not at the edge, but closer to the center, turnaround time decreases as long as the sled is initially moving against the spring force (*i.e.* moving away from the center of the device). However, when the sled is initially moving with the spring force (*i.e.,* moving towards the center of the device), the sled must turn around against the spring force. For turnarounds near the device center, the spring force is close to zero and has little impact. However, turning around near the device's edge can increase turnaround time by as much as 4X.

Peak bandwidth. Peak (streaming) bandwidth is achieved by having the sled sweep its full distance in the Y direction while data is accessed, turning around while seeking one bit in the X direction, and then repeating the process in the −Y direction. Most physical trends improve peak bandwidth, including: (1) decreasing bit size, which increases the number of bits per second passing under a tip; (2) increasing sled acceleration or spring force, which (by decreasing turnaround time) reduces the time when the probe tips cannot access data; (3) increasing the number of independent sleds, which decreases each sled's mass; (4) increasing the number of concurrently active tips. Even increasing sled velocity will initially increase streaming bandwidth by decreasing the time it takes to read an entire track. However, increasing velocity also increases turnaround time. As the time spent reading an entire track decreases and the turnaround time increases, the device eventually spends more time turning around than reading. At this point, peak bandwidth decreases. For a given actuator force, sled mass and spring force, there is a maximum velocity after which peak bandwidth declines.

Capacity. MEMS-based storage capacity is directly increased by either decreasing the bit size (*i.e.,* increasing areal density) or by increasing the actuator force. This latter can improve density by decreasing the distance required during turnaround (at the device edge). With greater force, the distance decreases, creating more useful area where bits can be stored and accessed. In contrast, increasing the sled velocity increases the turnaround time (and distance), which decreases the effective media area. Increasing the number of sleds also decreases capacity because more of the die area must be used for actuators. Like disk drives, capacity also decreases with increasing error rates because: (1) more powerful error-correcting codes must be used, decreasing the ratio of data bits to ECC bits; (2) entire bad sectors are not used; and (3) probe tip failures render regions of the media inaccessible.

	G1	G2	G3
bit width (nm)	50	40	30
sled acceleration (g)	70	82	105
access speed (kbit/s)	400	700	1000
X settling time (ms)	0.431	0.215	0.144
total tips	6400	6400	6400
active tips	640	1280	3200
max throughput (MB/s)	25.6	89.6	320
number of sleds	1	1	1
per-sled capacity (GB)	2.56	4.00	7.11
bidirectional access	no	yes	yes

Table 2: ***MEMS-based storage parameters used in our experiments.***

Power. Power requirements increase with several physical trends, including: (1) decreasing bit size, which requires more signal processing power to resolve each bit; (2) increasing sled velocity, which requires more force to achieve higher speeds; and (3) increasing error rate, which requires more error-correction bits to be read or written during each access.

Reliability. Reliability improves with many physical trends, including increasing actuator force, decreasing sled mass, and increasing spring force. These all directly increase the shock tolerance of MEMS-based storage devices, allowing them to sustain greater drops and bounces in portable devices. Increasing the number of sleds can also increase reliability, by allowing a device to tolerate entire sled failures. In the simple case, where each sled independently holds information (*i.e.,* no redundancy), a single sled failure would lose that sled's data. However, multi-sled MEMS-based storage devices could easily implement RAID configurations, allowing the entire device to tolerate a sled failure without any loss of data. Even a single sled can employ RAID among different probe tip storage locations. Depending on the configuration (*e.g.,* mirroring, RAID level 5), a device could also tolerate one or multiple tip or sector failures.

3. PERFORMANCE MODELS

This section describes the MEMS-based storage models and the simulation techniques used in the experiments described below. A detailed description of the performance model and an exploration of its sensitivity to various design parameters is presented in Reference [13].

3.1 Three Generations of Devices

Given the wide range of parameters, exploring the entire MEMS-based storage design space is not feasible. Instead, three models of MEMS-based storage are used, based on anticipated technology advances over the first three generations (Table 2).

	Atlas 10K	"SuperDisk"
RPM	10,025	20,000
Max bandwidth (MB/s)	25	170
Avg. seek (ms) (rd/wr)	5.7/6.19	3.12/3.58
Max full stroke (ms)	10.83/11.32	8.50/8.96

Table 3: ***Performance characteristics of the Quantum Atlas 10K disk drive and the extrapolated SuperDisk model.***

The "1st generation (G1)" model represents a conservative initial MEMS-based storage device, which could be fabricated within the next three years [4]. The sled has a full range of motion of 100 μm along the X and Y axes, and the actuators accelerate the sled at $70g$. To access data, the device uses a relatively primitive recording scheme, leading to a per-tip data rate of 400 kbit/s. This design only supports unidirectional accesses, where reads and writes only occur when the sled moves in the positive Y direction.

G1's media, tip resolution, and sled positioning system provide a square bit cell of 50 nm such that each tip addresses a 2000×2000 array of bits. The sled footprint is 0.64 cm^2 allowing 6400 tips for each sled. This yields a raw capacity of 2.56 GB per sled. However, media error management requires a 10-bit-per-byte encoding. Also, sled tracking and synchronization information requires 10 tracking bits for every 80 data bits. During media access, the sled is restricted to the fixed access velocity. However, the sled speed is not limited during seeks.

The "2nd Generation (G2)" model. Several fundamental improvements enhance G2 over G1. First, media access occurs in both the +Y and −Y directions. Second, per-tip data rate increases to 700 Kbit/s based on trends in probe tip technology. A decrease in the sled mass and an increase in the actuator voltage leads to an increase in sled acceleration to $82g$. Also, improvement in the servo system reduces the settling time for each X seek. Decreases in per-tip power utilization can lead to a larger number of tips that can be active simultaneously, vastly improving the maximum throughput. Finally, media material improvements increase G2's bit density by 20%.

The "3rd Generation (G3)" model. G3 approaches the high-end of many MEMS-based storage parameters and characteristics. Here the bit density scales down to 30 nm per bit, and a decrease in the sled mass leads to higher sled acceleration. In this case a change in the suspension and sled design leads to a higher resonant frequency, resulting in a shorter X settling time. Throughput is increased, largely because of the addition of more active tips.

The reference disk. A validated DiskSim module [35] for the Quantum Atlas 10K [30] enabled a comparison of a modern disk's performance to MEMS-based storage device performance.

The SuperDisk model was created to compare MEMS-based storage to an aggressive disk drive projection to the year 2005. Extrapolating on the current performance trends in disk drive technology, the SuperDisk achieves streaming bandwidth of up to 125 MB/s. Its seek time drops to a 3 ms average and it rotates at 20,000 RPM. The Atlas 10K and SuperDisk parameters are compared in Table 3.

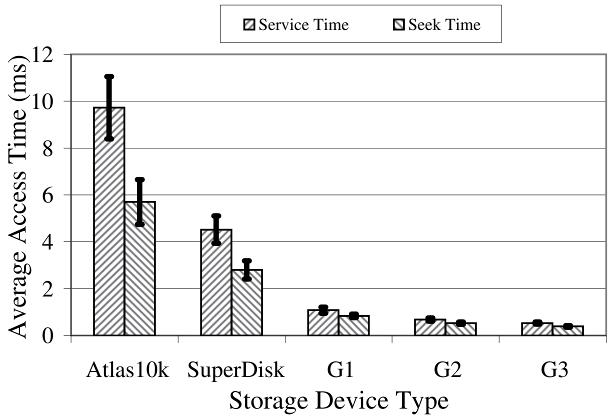

Figure 6: ***Average access times of each model under the microbenchmark.*** *MEMS-based storage devices provide both better performance and smaller variance. Bars on each column represent one standard deviation. The workload was 10,000 randomly-distributed requests, two thirds reads with an arrival rate was 20 requests per second.*

3.2 Simulation Environments

Using the model described in Reference [13] and the device parameters in Table 2, we created simulation models for each MEMS-based storage device and integrated those models into DiskSim, a freely-available disk simulator that accurately models disk drives [11], including the Atlas 10K. DiskSim was used for the microbenchmark and trace-based experiments described below. For the application experiments, DiskSim was integrated with the SimOS machine simulator [32]. SimOS was configured to model a 1 GHz Alpha 21164-based system with 128 MB of RAM running Digital UNIX version 4. The OS runs atop the virtual machine, using special device drivers to interact with simulated I/O devices. Finally, a model of IBM's low-power disk drive [17] was used to compare against our MEMS-based storage power models. These power models were driven using timing-accurate traces of SCSI block requests gathered from Linux's SCSI device driver.

4. PERFORMANCE RESULTS

To successfully fill the memory/storage gap, MEMS-based storage technology must offer a significant improvement in I/O and overall application performance. For mobile applications, power dissipation is crucial. Using microbenchmarks and six different workloads, this section compares the performance and power utilization of our MEMS-based storage device models (G1, G2, and G3) against a 1999 Quantum Atlas 10K disk drive and the hypothetical SuperDisk described above.

4.1 Microbenchmark Results

The first workload is a microbenchmark of 10,000 randomly-distributed requests. Two thirds of the requests were reads, and the arrival rate was 20 requests per second. Figure 6 shows that all three MEMS models outperform the Atlas 10K and SuperDisk disks by approximately 10X and 5X, respectively.

Figure 6 also shows that MEMS-based storage devices have much less access time variation than disk drives. In a disk drive, the distances over which the heads and media must travel to reach an individual block vary significantly, causing a wide variation in access time. Standard deviations of average service time for the random benchmark on the Atlas and SuperDisk are 2.66 and 1.40, respectively. In contrast, the MEMS-based storage devices have standard deviations between 0.26 and 0.09. This small variation is due to spring effects, the absence of rotational latency, and the much shorter full-throw distance of 100 microns (vs. several centimeters in a disk drive). Therefore, seek times are tightly constrained. The lower variances, and thus greater potential predictability, has intriguing consequences for the design of embedded systems with real-time requirements.

Another characteristic, which does not appear in this graph, is the benefit of parallelism. A MEMS-based storage device may include multiple fully-independent sleds over which data are striped. A conventional disk queues incoming requests when the device is already servicing a previous request, because most modern disks include only one mechanism for accessing the media. However, a multi-sled MEMS device can simultaneously service multiple requests if their data falls on separate sleds, much like disk arrays. Under the same microbenchmark with an increased inter-arrival rate, a 4-sled device has the potential to provide 4 times the throughput. Similar benefits can be gained by aggregating multiple single-sled devices together, creating a MEMS-based RAID system. Given their significantly lower volume, many MEMS-based storage devices could be fit into a standard drive enclosure, increasing both performance and capacity per volume relative to conventional disks[2].

4.2 Application Results

This section presents the results from real-world benchmarks, measured on systems with simulated MEMS-based storage devices in two different configurations: first, as a simple replacement for disks; and second, as a nonvolatile disk cache.

Comparing MEMS-based storage devices to disks. The first two applications, the Andrew Benchmark Suite [16] and PostMark [21] were designed for file system and I/O performance analysis. The Andrew Benchmark consists of a set of file and directory operations followed by a long compile. The PostMark benchmark performs many small file operations (*e.g.*, create, delete, read, write) and was designed to be representative of the file system workloads seen in e-mail, news, and electronic commerce environments. Table 4 shows that MEMS-based storage devices can significantly reduce the I/O time for these workloads. Both Andrew and Postmark show an improvement in I/O service time between 4X and 6X, with an overall application performance improvement between 2X and 4X.

The GNU Linker benchmark, Gnuld, is a test in which a large set of object files are linked using the GNU linker. All of the MEMS-based storage devices improve performance over the Atlas10k, with the G3 device decreasing I/O time by 7X. However, SuperDisk's higher bandwidth greatly enhances its performance over the G1 device.

The TPC-D [37] benchmarks also see a large reduction in I/O time from the MEMS-based storage devices. The higher bandwidth of the SuperDisk, however, greatly enhances its performance for the TPC-D queries. In both cases, the SuperDisk out-performs the G1 MEMS device. The performance of the MEMS-based storage devices is also hampered by very high disk cache hit rates for the TPC-D queries, which are between 83% and 90%, respectively, for the disks. Our MEMS-based storage device does not include a prefetching cache, and so cannot benefit from the high sequentiality and data reuse of these benchmarks. However, even without a RAM cache, the MEMS-based storage devices outperform the baseline disk by a wide margin.

MEMS-based storage devices as caches for disks. MEMS-based storage can also be used as an augmentation of the existing storage hierarchy. For example, with their low entry cost, MEMS-based storage devices could be incorporated into future disk drives as very large (1-10 GB) nonvolatile caches. The superior performance of MEMS-based storage devices would allow the cache to absorb latency-critical synchronous writes to metadata and cache small files to improve small read performance. For example, Baker *et al.* show that using fast nonvolatile storage to absorb synchronous disk writes both at a client and at a file server increases performance from 20% to 90% [2].

To explore MEMS-based storage as a nonvolatile cache for disk, DiskSim was augmented to allow a MEMS-based storage device to serve as a cache for a disk. The cache was 2.5 GB, the disk was 9.2 GB, and the workload was the 1-day cello trace from [33]. This trace actually includes eight separate devices so the experiments use a cache per disk. The results show that the average I/O response time is 14.66 ms for an Atlas10K disk drive without any MEMS cache vs. 4.03 ms for a disk with a G2 type MEMS-cache (and 2.76 ms for a single large G2 MEMS device that replaced the disk). Since most of the read requests are serviced from the client-side DRAM cache, the 3.5X performance improvement, over just a disk drive, is achieved mainly by quickly servicing writes. However, unlike DRAM-based write caching (which absorbs writes but risks losing data), the MEMS cache is nonvolatile, providing the same data integrity guarantees as disk drives. An alternate experiment in which all eight devices in the cello trace were re-mapped to a larger version of the Atlas10K disk with a single MEMS cache only suffered a slight increase in average access time to 4.66 ms. This longer service time stems from an increase in queueing since the large single device is doing the work of eight. It shows, however, that caching absorbs enough of the device's activity to provide a good performance boost.

Instead of using the MEMS-based storage device as a cache, it is also possible to expose the device to the OS so that file systems can allocate specific data onto it. Depending on their access patterns and performance needs, file systems could place small structures (*e.g.*, file system metadata) on MEMS-based storage, while using the disk for streamed or infrequently-accessed data. This could be done on individual disks or within RAID arrays, creating the potential for AutoRAID-like systems [39]. Further, because RAID ar-

[2]As measured in bits/cm^3, MEMS-based storage devices have a much higher density than disk drives. This is because drives must dedicate significant volumetric area to the spindle, platter separations, and the actuator. However, packing many MEMS-based storage devices into a small area will significantly increase heat dissipation requirements for the MEMS devices.

	Andrew		Postmark		Gnuld		TPC-D #4		TPC-D #6	
Device	Compute	I/O	Compute	I/O	Compute	I/O	Compute	I/O	Compute	I/O
Atlas 10k	2.8	3.9	9.8	730.4	0.8	25.1	2.7	27.7	8.9	22.3
Superdisk	2.8	1.7	10.0	397.0	0.7	8.8	2.7	3.3	8.8	0.3
G1 MEMS	2.8	1.8	10.3	257.4	0.8	11.3	2.7	14.8	8.9	5.5
G2 MEMS	2.8	1.0	10.9	171.0	0.8	4.6	2.7	5.2	8.9	0.2
G3 MEMS	2.8	0.7	11.0	170.9	0.8	3.6	2.7	4.2	8.8	0.3

Table 4: ***Comparison of five applications on disks and MEMS-based storage devices.*** *All numbers are in seconds.*

rays are less cost-sensitive than individual disks, arrays of MEMS-based storage devices could be incorporated more cost-effectively into RAID arrays, providing significant performance improvements for RAID's costly write operations.

4.3 Power Utilization

The physical characteristics of MEMS-based storage devices may make them less power hungry than even low-power disk drives [18, 19]. This power advantage comes from several sources: lower overall power requirements for moving the media and operating the read/write tips, and faster transitions between active and standby modes.

While the media sled in a MEMS-based storage device does move continuously in the X and Y directions during data access, the sled has much less mass than a disk platter and therefore takes far less power to keep in motion. Specifically, it takes less than 100 mW to continuously move a MEMS sled, while it takes over 600 mW to continuously spin a disk drive.

Another power savings comes from the electronics of MEMS-based storage devices. In disk drives, the electronics span multiple chips and great distance from the magnetic head at the end of the arm to the drive interface. Therefore, high-speed signals must cross several chip boundaries, increasing power dissipation. Further, disks' large physical platters, heads, arms and actuators require sophisticated, power-hungry signal processing algorithms to compensate for imperfect manufacturing, thermal changes, environmental changes, and general wear. Current low-power drives consume almost 1.5 W [18, 19] in drive electronics, much of it spent on accurately positioning the recording head. Of course, not all drive electronics must be active during short idle periods; some electronics, such as the servo control, can be powered down. This technique reduces total drive power by up to 60%, adding a small additional time penalty to return to active mode (from 40–400 ms).

Drive power can also be saved by turning off the spindle motor during long idle periods. Numerous studies have demonstrated the power savings of this standby mode [9, 15, 24, 25], and current low-power drives do incorporate this feature. MEMS-based storage can also employ a standby mode, stopping sled movement during periods of inactivity. Further, the sled's low mass allows MEMS to quickly switch between active and standby mode (0.5 ms), where a low-power drive requires up to 2 seconds to spin up and return to active mode. This long delay significantly increases access time for the first request after an idle period. Therefore, drive power-management algorithms usually wait at least 10 seconds before going into standby mode. During this 10 second delay, and during the 2 second spin-up time, considerable power is wasted. In contrast, MEMS-based devices can transition from standby-to-active in 0.5 ms, allowing these devices to be much more aggressive in using standby mode.

MEMS-based storage also has the ability to adjust its power consumption during data accesses by reading or writing at a smaller granularity than standard 512 byte blocks. Since most power is dissipated by the probe tips, and not by positioning or moving the media sled, reading or writing only the necessary data could save considerable power. The device only needs to activate as many tips as are necessary to satisfy a request, which could result in a substantial power savings. In contrast, the power required to move a disk drive's arm and spindle, and to servo control the head over the appropriate sector is much greater than the power necessary to actually read or write the 512 byte sector.

To understand how much power a MEMS-based storage device could save over a low-power drive, we simulated both and measured their power consumption across six workloads. The disk drive power model is based on IBM's low-power Travelstar disk and power management techniques described in [18, 19]. The device has 5 power modes: (1) active mode (data is being accessed) consumes 2.5 W for reads and 2.7 W for writes; (2) performance idle (some electronics are powered down) consumes 2.0 W; (3) fast idle (head is parked and servo control is powered down) consumes 1.3 W; (4) low-power idle (heads are unloaded from the disk) consumes 0.85 W; (5) standby (spindle motor is stopped) consumes 0.2 W. From Reference [17], the maximum time spent in the intermediate modes is: 1 second for performance idle, 3 seconds for fast idle, and 8 seconds for low-power idle.

For the MEMS-based storage device, power for a benchmark is computed during simulation by using the physical parameters in Reference [4]; each probe tip and its signal processing electronics consume 1 mW. To minimize packaging costs, we set our power budget to about 1 W. This limits the MEMS-based storage device to no more than about 1,000 simultaneously active probe tips. Further, given the sled design, the power consumed to keep the sled in motion is 0.1 W. Therefore, the maximum power for this MEMS-based storage device is 1.1 W. Standby power consumption is estimated to be 0.05 W.

Table 5 shows that the total energy consumed for the MEMS-based storage device is between approximately 10X and 50X lower, depending on the application. The five workloads already discussed are highly active and so most of the savings comes directly from lower energy consumption during data accesses (active mode). To test a more interactive workload, we traced the disk accesses generated by a user

	Andrew		Gnuld		Postmark		TPC-D #4		TPC-D #6		Netscape	
Category	Disk	MEMS	Disk	MEMS	Disk	MEMS	Disk	MEMS	Disk	MEMS	Disk	MEMS
active	19.5	0.7	84.6	3.6	1930.6	42.0	115.6	8.5	59.0	8.4	321.2	1.4
perfIdle	13.3	0.3	39.8	0.0	1181.1	7.7	45.4	0.1	43.6	0.3	1924.1	0.01
goToActive	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	513.5	0.0
fastIdle	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1799.9	0.0
lowPowerIdle	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1000.5	0.0
spinup	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	228.8	20.0
standby	0.0	0.2	0.0	0.0	0.0	8.0	0.0	1.1	0.0	1.9	308.9	327.9
Total (Joules)	32.8	1.2	124.4	3.6	3111.7	57.7	161.0	9.7	102.6	10.6	6096.9	349.3

Table 5: ***Comparison of energy required to execute six different workloads using disks and MEMS-based storage devices.*** *All numbers are given in Joules.*

browsing with Netscape on a Linux workstation for ten minutes. In this case, much of the power savings comes from MEMS-based storage's ability to aggressively use its low-power standby mode. In contrast, the disk drive spends 90% of its power transitioning between active and standby modes.

5. POTENTIAL OF MEMS-BASED STORAGE AND COMPUTATION

Section 4 shows that MEMS-based storage devices have significant advantages over disk drives. For example, I/O performance can increase by an order of magnitude. Further, unlike conventional disk caches, which often consist of volatile RAM, a MEMS-based disk cache creates significant performance improvements without risking data loss. Other advantages, such as physical size, portability, and the potential to integrate processing within the same substrate, create many exciting possibilities for system architects.

For portable applications such as notebook PCs, PDAs, video camcorders, and biomedical monitoring, MEMS-based storage provides a more robust and lower-power solution. Many of these applications involve rapid device rotation (*e.g.,* rapidly turning a PDA) and are prone to inducing shock (*e.g.,* dropping a device). MEMS-based storage does not suffer gyroscopic effects and can absorb much greater external forces. MEMS-based storage can also be integrated with biomedical sensors, allowing long-term medical monitoring devices to be implanted directly into the human body.

MEMS-based storage creates a new low-cost entry point for modest-capacity applications of 1–10 GB. This is because the baseline costs of a disk's mechanical components keep manufacturing prices from falling below a certain point, while MEMS-based storage devices can ride the linear decline in IC manufacturing process costs. However, large capacity drives may continue to enjoy a 10X price advantage for high-capacity storage (*e.g.,* 75 GB in 2000) because the drive assembly costs are subsumed by the media cost. Therefore, MEMS-based storage is not necessarily intended as a replacement for high-capacity disk drives, but as a supplement in the storage hierarchy.

With new applications aggressively creating massive amounts of data, MEMS-based storage can help solve data archival problems, including capacity, time to access data, and long-term data retrieval. For example, low-resolution medical biopsies generate over 600 MB of compressed data per patient; high-resolution MRIs generate 10X to 100X more data. Maintaining this data on-line is a costly problem, usually requiring that the data be migrated from disk to tape within a relatively short period of time. While tape is 100X cheaper than disk, tape system and storage management costs are tremendous, and the time to first access of data from a tape can be on the order of an hour.

Write-once versions of MEMS-based storage devices provide an attractive alternative. With potential areal densities up to 100X greater than write-many MEMS-based storage devices and 10X greater than high-capacity tape [22], it should be possible to build cost-effective storage "bricks" that hold thousands of MEMS devices. Each storage brick would contain an aggregate capacity of terabytes or petabytes while providing initial access time of under 1 second. Further, incorporating processing and interface logic with MEMS creates an *active* MEMS-based storage device capable of data processing or translation within the storage device. For archival storage, this avoids the common problem of not having a tape drive that can read the tape, or not having the application/hardware/OS capable of running the old program to process the data. Instead, the application is stored with the data and can be executed within the active device.

Active MEMS-based storage devices provide massive computational parallelism, creating the ultimate active storage device. For example, a single G3 device with 10,000 active probe tips, and the appropriate packaging for heat dissipation, could access data at 1 GB/s. Modern processors can easily consume data at that rate, but moving 1 GB/s between the storage system and host CPU requires costly interconnects [23]. Processing data within the storage device, where on-chip interconnects support GB/s bandwidths, avoids this cost. Moreover, it creates a very scalable architecture where adding storage automatically adds local bandwidth and computation [31].

As an illustration, consider the cost of a *select* database operation from Postgres. Measured on the Compaq Alpha architecture [31], select requires 3.8 instructions to process each byte of data. Therefore, processing the 1 GB/s of data a MEMS-based storage device could deliver would require 3,800 MIPS worth of processing power. Because the select operation allows parallel processing of the data, the 3,800 MIPS could be embedded in numerous ways: as a single 3,800 MIPS processor, 38 simple 100-MIPS processors, a custom ASIC, or reconfigurable logic. The resulting active storage device could complete a 5 GB select operation in just 5 seconds. In contrast, a modern disk drive streaming data at 50 MB/s would take 100 s just to read the data.

A different application domain for MEMS-based storage is bulk nonvolatile storage for embedded computers. Single-chip "throw-away" devices that store very large datasets can be built for such applications as civil infrastructure monitoring (*e.g.*, for bridges, walls, and roadways), weather and seismic tracking, and medical applications. One forthcoming application is temporary storage for microsatellites in low Earth orbit. Given that a satellite in a very low orbit passes over a single point very quickly, communications may only be possible in very short bursts. Therefore, a low-volume, high-capacity, nonvolatile storage device could be used to buffer data between transmission bursts. MEMS-based storage devices could also add huge databases to single-chip continuous speech recognition systems and be integrated into low-cost consumer or mobile devices. Such chips could be completely self-contained, with hundreds of megabytes of speech data, custom recognition hardware, and only minimal connections for power and I/O.

MEMS-based storage offers enhanced data security. With true systems-on-a-chip, sensitive data never has to move beyond the processor and the on-chip data store without being properly encrypted via on-chip circuitry. Such a design would provide no opportunity for traffic snooping devices, even if on the storage network, to capture a cleartext copy of sensitive information. Further, the self-contained nature of these components allow for the construction of inexpensive, high-capacity, tamper-proof smart cards.

6. CONCLUSIONS

This work demonstrates that MEMS-based storage has the potential to fill the ever-growing gap between RAM and disk access times and is an attractive alternative to disk drives for portable, low-power applications. Further, the range of device parameters and their impact on overall performance results in a diverse set of potential device designs that can be optimized for different application requirements (improved latency, bandwidth, capacity, or power).

The application results show that MEMS-based storage reduces application I/O stall times by 4–74X, with overall performance improvements ranging from 1.9–4.4X. Using MEMS as a cache for disk also achieves a significant performance improvement of 3.5X. Further, MEMS low-power requirements deliver up to a 54X power win over low-power disk drives. Most of these improvements result from the fact that average access times for MEMS-based storage are 10 times faster than disks (*e.g.,* 0.5-1.08 ms) and that MEMS is able to rapidly move between active and power-down modes.

Future work in this area includes exploring how to restructure storage systems (hardware and software) to best exploit MEMS-based storage devices. A first step is to develop an optimized file system which takes advantage of the physical characteristics of the device to improve performance, which is discussed further in [14]. Further, demonstrations in the mobile and archival storage domains should illustrate the utility of MEMS-based storage in systems. Finally, there are interesting research problems for active MEMS-based storage devices and the distributed algorithms necessary to use and manage them.

ACKNOWLEDGMENTS

We thank Rick Carley, the CMU MEMS Laboratory, and the anonymous reviewers for helping us refine this paper. We thank the members and companies of the Parallel Data Consortium (including CLARiiON, EMC, HP, Hitachi, Infineon, Intel, LSI Logic, MTI, Novell, PANASAS, Procom, Quantum, Seagate, Sun, Veritas, and 3Com) for their interest, insights, and support. We also thank IBM Corporation and Intel Corporation for supporting our research efforts. John Griffin is supported in part by a National Science Foundation Graduate Fellowship.

REFERENCES

[1] S. Araki. The memory stick. *IEEE Micro*, 20(4):40–46, July/Aug. 2000.

[2] M. Baker, S. Asami, E. Deprit, J. Ousterhout, and M. Seltzer. Non-volatile memory for fast, reliable file systems. In *5th Conference on Architectural Support for Programming Languages and Systems*, pages 10–22, Oct. 1992.

[3] C. Brown. Microprobes Promise a New Memory Option. *EE Times*, pages 6,41,44, 12 Jan. 1998.

[4] L. R. Carley, J. A. Bain, G. K. Fedder, D. W. Greve, D. F. Guillou, M. S. C. Lu, T. Mukherjee, S. Santhanam, L. Abelmann, and S. Min. Single Chip Computers with Microelectromechanical Systems-based Magnetic Memory. *Journal of Applied Physics*, 87(9):6680–6685, 1 May 2000.

[5] Center for Highly Integrated Information Processing and Storage Systems, Carnegie Mellon University. http://www.ece.cmu.edu/research/chips/.

[6] R. Colsen. Sorting Disk Blocks To Reduce Load Times. Personal Communication, Intel Corporation, 1999.

[7] T. Davis. Realizing a Completely Micromechanical Data Storage System (Kionix, Inc.). In *Diskcon 99 International Technical Conference*, Sept. 1999.

[8] M. Despont, J. Brugger, U. Drechsler, U. Dürig, W. Häberle, M. Lutwyche, H. Rothuizen, R. Stutz, R. Widmer, H. Rohrer, G. Binnig, and P. Vettiger. VLSI-NEMS Chip for AFM Data Storage. In *Proceedings 12th International Workshop on Micro Electro Mechanical Systems*, pages 564–569, Orlando, FL, 17–21 Jan. 1999.

[9] F. Douglis, P. Krishnan, and B. Marsh. Thwarting the Power-Hungry Disk. In *Winter USENIX*, pages 292–306, Jan. 1994.

[10] G. K. Fedder, S. Santhanam, M. L. Reed, S. C. Eagle, D. F. Guillou, M. S.-C. Lu, and L. R. Carley. Laminated High-Aspect-Ratio Microstructures In a Conventional CMOS Process. In *Proceedings of the IEEE Micro Electro Mechanical Systems Workshop*, pages 13–18, San Diego, CA, Feb. 1996.

[11] G. Ganger, B. Worthington, and Y. Patt. The DiskSim Simulation Environment Version 1.0 Reference Manual. Technical Report CSE-TR-358-98, Department of Computer Science and Engineering, University of Michigan, Feb. 1998.

[12] J. Gray. What Happens When Processing, Storage, and Bandwidth are Free and Infinite. In *IOPADS Keynote*, Nov. 1997.

[13] J. L. Griffin, S. W. Schlosser, G. R. Ganger, and D. F. Nagle. Modeling and Performance of MEMS-Based Storage Devices. In *ACM SIGMETRICS 2000*, Santa Clara, CA, 17–21 June 2000. Published as *Performance Evaluation Review*, **28**(1):56–65, June 2000.

[14] J. L. Griffin, S. W. Schlosser, G. R. Ganger, and D. F. Nagle. Operating Systems Management of MEMS-based Storage Devices. In *Proceedings of the 4th Symposium on Operating Systems Design & Implementation*, San Diego,

CA, 23–25 Oct. 2000.

[15] M. Horowitz, T. Intermaur, and R.Gonzalez. Low-Power Digital Design. In *Proceedings of the 1994 IEEE Symposium on Low Power Electronics*, pages 10–12, Oct. 1994.

[16] J. Howard, M. Kazar, S. Menees, D. Nichols, M. Satyanarayanan, R. Sidebotham, and M. West. Scale And Performance Of a Distributed File System. *ACM Trans. on Computer Systems*, 6(1):51–81, Feb. 1988.

[17] IBM. Adaptive Power Management for Mobile Hard Drives. http://www.almaden.ibm.com/almaden/pbwhitepaper.pdf.

[18] IBM. IBM family of microdrives. http://www.storage.ibm.com/hardsoft/diskdrdl/micro/-datasheet.pdf.

[19] IBM. IBM Travelstar 8GS. http://www.storage.ibm.com/-hardsoft/diskdrdl/travel/32ghdata.pdf.

[20] IBM Press Room. IBM Sets Another Disk-Drive World Record. http://www.ibm.com/press/prnews.nsf/, Oct. 1999.

[21] J. Katcher. PostMark: A New File System Benchmark. Technical report TR3022, Network Appliance, Oct. 1997.

[22] S. Khizroev, J. Bain, and M. Kryder. Considerations in the Design of Probe Heads for 100 Gbit/in^2 Recording. *IEEE Trans. Magnet.*, 33(5):2893–2895, 1997.

[23] J. Laudon and D. Lenoski. The SGI Origin: A ccNUMA Highly Scalable Server. *24th International Symposium on Computer Architecture*, pages 241–251, 2–4 June 1997.

[24] K. Li, R. Kumpf, P. Horton, and T. Anderson. A Quantitative Analysis of Disk Drive Power Management in Portable Computers. In *Winter USENIX*, pages 279–292, Jan. 1994.

[25] Y.-H. Lu, T. Šimunić, and G. D. Micheli. Software Controlled Power Management. In *7th International Workshop on Hardware/Software Codesign*, pages 157–161, May 1999.

[26] C. D. Mee and E. D. Daniel. *Magnetic Storage Handbook, Second Edition*. McGraw-Hill, 1996.

[27] Nanochip Inc. Nanochip, Inc. Product Overview. In *Diskcon 99 International Technical Conference*, Sept. 1999.

[28] D. A. Patterson and J. L. Hennessy. *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann Publishers, Palo Alto, CA, 2nd edition, 1996.

[29] E. Pugh. Storage Hierarchies: Gaps, Cliffs, and Trends. *IEEE Transactions on Magnetics*, pages 810–814, Dec. 1971.

[30] Quantum Corporation. *Quantum Atlas 10K 9.1/18.2/36.4 GB Ultra 160/m SCSI Hard Disk Drive Product Manual*, 6 Aug. 1999. Publication number 81-119313-05.

[31] E. Riedel. *Active Disks—Remote Execution for Network-Attached Storage*. PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, Nov. 1999. Technical report CMU–CS–99–177.

[32] M. Rosenblum, S. Herrod, E. Witchel, and A. Gupta. Complete Computer System Simulation: The SimOS Approach. *IEEE Parallel & Distributed Technology*, 3(4), Winter 1995.

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

[34] B. Schechter and M. Ross. Leading The Way In Storage. *IBM Research Magazine*, 35(2), 1997.

[35] J. Schindler and G. Ganger. Automated Disk Drive Characterization. Technical Report CMU-CS-99-176, Carnegie Mellon University School of Computer Science, Nov. 1999.

[36] J. W. Toigo. Avoiding a Data Crunch—A Decade Away: Atomic Resolution Storage. *Scientific American*, May 2000. http://www.sciam.com/2000/0500issue/0500toigbox6.html.

[37] Transaction Processing Performance Council. TPC Benchmark D (Decision Support) Standard Specification. http://www.tpc.org/benchmark_specifications/TPC_D/-210.pdf.

[38] P. Vettiger, M. Despont, U. Drechsler, U. Dürig, W. Häberle, M. I. Lutwyche, E. Rothuizen, R. Stutz, R. Widmer, and G. K. Binnig. The "Millipede"—More Than One Thousand Tips for Future AFM Data Storage. *IBM Journal of Research and Development*, 44(3):323–340, 2000.

[39] J. Wilkes, R. Golding, C. Staelin, and T. Sullivan. The HP AutoRAID Hierarchical Storage System. In *15th ACM SOSP*, pages 96–108, Dec. 1995.