Designing Computer Systems With MEMS-based Storage

Steven W. Schlosser, John Linwood Griffin,

David F. Nagle, Gregory R. Ganger

May 2000

CMU-CS-00-137

School of Computer Science

Carnegie Mellon University

Pittsburgh, PA 15213

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 power consumption, and enable many new classes of applications. This research explores the impact that several different MEMS-based storage designs will have on computer systems. Results from five application studies show these devices reduce application I/O stall times by 3–10X and improve overall application performance by 1.6–8.1X. Further, integrating MEMS-based storage as a disk cache achieves a 3.5X performance improvement over a standalone disk drive. Power consumption simulations show that MEMS-based storage devices use up to 10X less power than state-of-the-art low-power disk drives. Many of these improvements stem from the fact that average access times for MEMS-based storage are 10X faster than disks and that MEMS devices are able to rapidly move between active and power-down mode. Combined with the differences in the physical behavior of MEMS-based storage, these characteristics create numerous opportunities for restructuring the storage/memory hierarchy.

This research is supported by the member companies of the Parallel Data Consortium. At the time of this writing, these companies include CLARiiON Array Development, EMC Corporation, Hewlett-Packard Labs, Hitachi, Infineon Technologies, Intel Corporation, LSI Logic, MTI Technology Corporation, Novell, Inc., PANASAS, L.L.C., Procom Technology, Quantum Corporation, Seagate Technology, Sun Microsystems, Veritas Software Corporation, and 3Com Corporation. The authors also thank IBM Corporation for their support of this project. John Griffin is supported in part by a National Science Foundation Graduate Fellowship.

Keywords: microelectromechanical systems (MEMS), storage systems, memory hierarchy

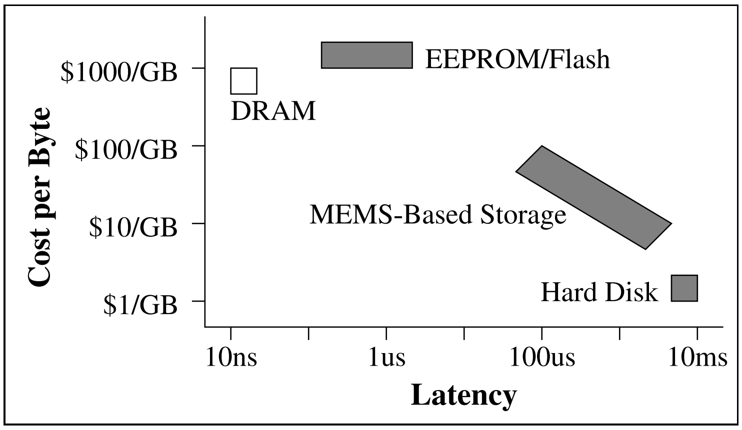


Figure 1: ***Predicted cost and latency for storage technologies in 2005.*** *MEMS-based storage fills the growing memory hierarchy gap between RAM and disk. The grey boxes represent non-volatile storage. The EEPROM box is wide because of the wide gap between read and write latencies for Flash memories. The MEMS box is wide and tall because of the many design possibilities for this new type of storage (see Section 2).*

1 Introduction

For decades, the memory hierarchy has suffered from significant access, bandwidth and cost gaps between processor, RAM, and disk [Pug71]. Fortunately, the processor/RAM gap has been mitigated by fast cache memories [PH96]. Unfortunately, the RAM/disk gap has remained unfilled, widening to 6 orders of magnitude in 1999 and continuing to widen at about 50% per year. The result is a significant performance and scalability problem across a range of applications, including databases, web servers, mail servers, program development tools, and even Microsoft Word load times [Col99].

This RAM/disk performance gap is due directly to the physical characteristics of disk drives. While disks continue to deliver capacity growth of over 60% per year, the physics of a drive's mechanical positioning system limits disk access time improvements to a modest 7% per year [PH96]. EEPROM offers a portable high-performance alternative, but its per-megabyte cost is 2 orders of magnitude higher than disk storage (see Figure 1).

MEMS-based storage is an exciting new technology that could provide significant performance gains over current disk drive technology and at costs much lower than EEPROM technology [CBF^{+}00, Bro98]. Based on microelectromechanical systems (MEMS), this non-volatile storage technology merges magnetic recording material and thousands of probe-based recording heads to provide storage capacity of 1–10 GByte of data in under 1 cm^2 of area, access times of 0.5–1.1 ms, and streaming bandwidths up to 100 MByte/s.

Further, because MEMS devices can be built using standard CMOS fabrication processes [FSR+96], integrating processing elements on the same chip with mass storage could cost significantly less than an equivalent non-volatile DRAM solution [CBF+00]. Systems could include several microprocessors or hundreds of custom computational engines (*e.g.*, MPEG encode/decode, cryptography, signal processing) integrated directly with MEMS-based storage. This integration will significantly improve performance, power consumption, and cost over multicomponent solutions. More importantly, it will lay the foundation for a single computing brick [Gra97], containing processing, non-volatile storage and volatile storage.

Although MEMS-based storage devices are still several years away from commercialization, their potential impact in reducing the memory gap makes them an important technology for systems architects' consideration. Our previous work [GSGN00a] examines the basic behavior and raw performance of a MEMS-based storage device (*e.g.*, average access time, maximum read/write bandwidth). The work presented here focuses on integrating MEMS-based storage into the memory hierarchy to improve performance and power consumption, specifically in two different roles: as a replacement for disk drives, especially in mobile applications where power is critical, and as a non-volatile cache embedded within a conventional disk drive's electronics.

The results presented below show that MEMS-based storage can reduce application I/O stall times by 3–10X for a set of five file system and database workloads. The resulting speedups for these applications range from 1.6–8.1X, depending mainly on whether the tasks are I/O- or CPU-bound. Power estimates predict that MEMS-based storage can reduce power for these workloads by a factor of up to 10X over state-of-the-art low-power disk drives. Coupling this with MEMS-based devices' better shock tolerance and higher reliability makes the technology an ideal high-capacity storage solution for mobile, low-power applications.

To ensure that our models accurately reflect potential implementations, we are working closely with CHI^2PS^2, the Center for Highly-Integrated Information Processing and Storage Systems at Carnegie Mellon, which is actively developing practical MEMS-based storage

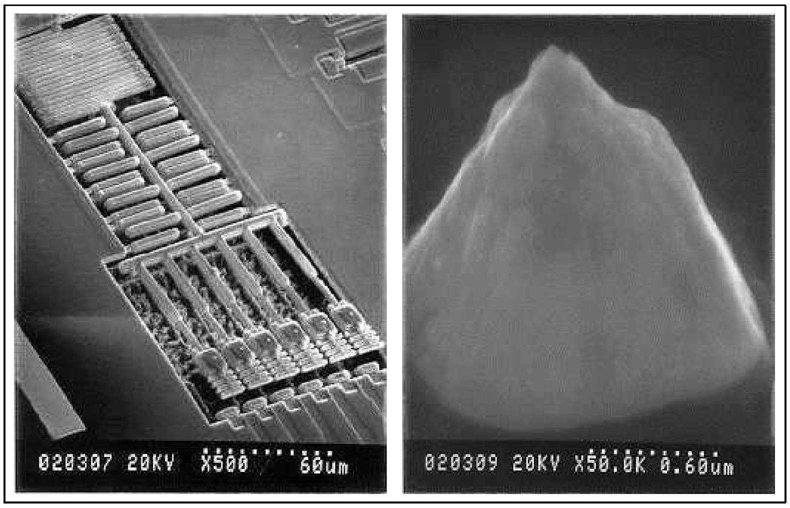


Figure 2: ***Prototype Positioning System and Probe Tip.*** *MEMS researchers have developed the prototype read/write head (probe tip) and positioning system shown above. Because the recording material is not perfectly flat, the positioning system must be able to actively adjust the height of the probe 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.*

devices. This collaboration allows us to explore the system-level impact of various types of MEMS-based storage, evaluating which physical design trade-offs are most important across a range of applications. Our results feed back to the MEMS researchers, focusing their attention on design parameters that most significantly impact system-level performance and avoiding optimizations that provide little real benefit.

The remainder of this paper is organized as follows. Section 2 describes MEMS-based storage and many of the physical trade-offs of the devices. Section 3 describes the performance model. Section 4 presents results for a number of applications. 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 devices

MEMS are very small-scale mechanical structures—on the order of tens to thousands of micrometers—fabricated on silicon chips. These microstructures are created using the same photolithographic processes used in manufacturing standard semiconductor devices. MEMS

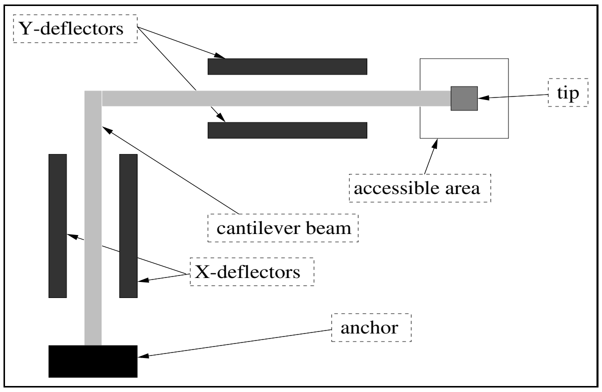
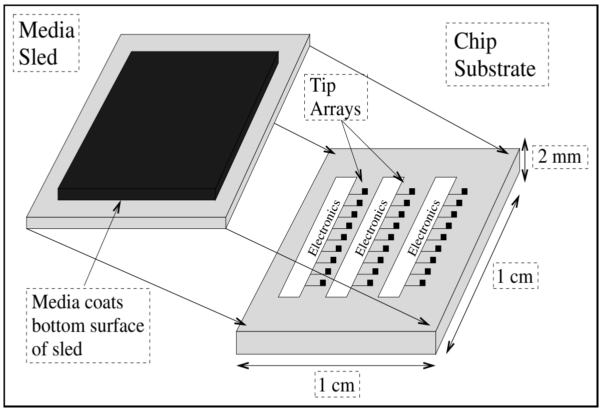


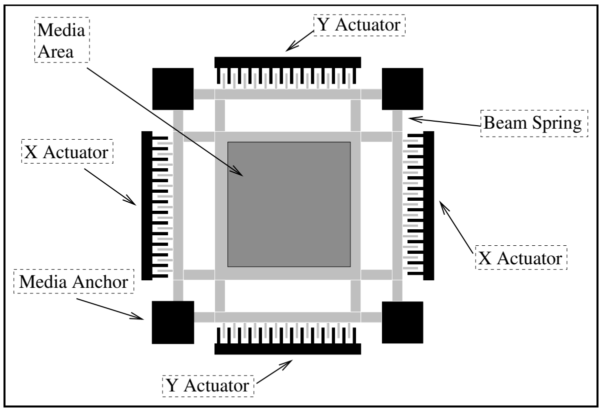
Figure 3: ***A cantilevered-beam probe tip in the "fixed media" model.*** *The X- and Y-deflectors are capable of quickly positioning the tip anywhere in the small accessible area. The overall capacity of this model is limited to tens or perhaps hundreds of megabytes because only 1% of the cantilever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

structures can be made to slide, bend, or deflect in response to an electrostatic or electromagnetic force from a nearby actuator or from external forces in the environment. MEMS machines are limited in mobility compared to standard mechanical systems. For example, it is difficult to build durable microbearings for rotating components. Previous attempts at building micromachined gear trains have shown that such devices lock up from friction within a few thousand revolutions. However, alternative designs such as spring-suspended masses which translate in the X and Y directions (instead of rotating in θ) circumvent these frictional barriers.

One class of MEMS-based storage system structures under investigation takes advantage of arrays of thousands of microscopic magnetic probes each accessing a dense substrate of magnetic material [Bro98, CBF$^+$00]. This design offers several notable advantages over disk-based storage including better cost, access time, power dissipation, mass, failure rate, and shock sensitivity. Further, there is inherent parallelism across the array of read-write tips: multiple tips may be accessed concurrently to increase throughput, accesses may be redundant to enhance reliability, or completely independent accesses may occur in parallel. In addition, the MEMS fabrication process integrates seamlessly with standard CMOS processes [FSR$^+$96]. This ease of fabrication opens the door for mass manufacturing MEMS-enhanced systems-on-a-chip—massively parallel manufacturing, small per-unit cost in high volume, a clear road map toward smaller processes, and large amounts of industry momen-



(a) Media sled and read/write tips



(b) Media sled, beam springs, and actuators

Figure 4: ***An example of the "moving media" model.*** *In (a), we see how the media sled is attached above the fixed tips. The sled can move up to 100 μm along the X and Y axes, allowing the fixed tips to address 30–50% of the total media area. In (b), we see the actuators, the spring suspension, and the media sled itself. Anchored regions are shown in black and the movable structure is in grey.*

tum.

MEMS-based microstructures can be used to build storage devices in a variety of ways, with different designs affecting the robustness, manufacturability, cost, capacity, access speed and latency of these devices. As an example, Figure 3 depicts one proposed MEMS-based storage design. In this "fixed media" model, miniature cantilevered L-shaped beams suspend a read/write head (hereafter called 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 and then using standard magnetic recording techniques to read or write the bits. The near-massless cantilevered beam enables very quick positioning times (on the order of 10-100s of microseconds), but the space efficiency is poor—only about 1% of the potential media area is used for storage. By comparison, conventional disk drives use about 50% of their platter area for data storage. While this design is useful for visualizing MEMS-based storage, expected capacities of only hundreds of megabytes per device limits its practicality in comparison to Flash memory, battery-backed RAM, or non-volatile RAM components.

A more space-efficient design is shown in Figure 4. Here, a movable media sled is suspended with springs above an array of several thousand fixed probe tips. The media area on the sled is about 1 cm^2, under which perhaps 10,000 probe tips can be placed. Assuming a bit cell of 0.0025 μm^2 (50 nm per side) and encoding/ECC overheads of 2 bits per byte, the device's data storage capacity is about 4 GByte/cm^2 [CBF^+00]. A more aggressive goal of 0.0009 μm^2 (30 nm per side) would yield capacities of 11 GByte/cm^2 or greater. While this design improves space efficiency to 30–50%, the sled mass increases positioning times over the fixed media design—a necessary tradeoff to achieve disk-like capacities. For a complete description of the device characteristics see [CBF^+00, GSGN00a].

There are many other probe-based storage research projects and designs. IBM's initial efforts employed cantilevered probe tips that melted pits into a rotating polymer disk [BS97]. IBM's more recent Millipede project [D^+99, V^+99], continues to read and write thermo-mechanically, but operates thousands of probe-tips in parallel as they move over a static substrate. Two startup companies, Kionix [Dav99] and Nanochip [Nan99], are also developing probe-based magnetic storage architectures. The Kionix device uses a moving media design, similar to Figure 4, while the Nanochip design attaches the heads to an actuated platform, with fixed media. Finally, researchers at Carnegie Mellon University have explored write-once architectures with mechanisms similar to those described in Figure 4, but with storage capacities 100X greater than the write-many probe-based storage and 10X denser than current high-capacity tape [KBK97].

While many 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 direction. Therefore, this study uses the CMU MEMS group's moving-media model as a basis for quantitative analysis of future MEMS-based storage systems.

2.1 Device and Data Layout

The magnetic media on the sled is organized into rectangular regions as shown in Figure 5. Each rectangular area stores N×M bits, and is only accessible by one probe tip. Multiple tip sectors are grouped into *logical sectors*, similar to logical blocks in SCSI disks. Unlike most conventional disks, multiple probe tips can access the media in parallel—thus many tip sectors can be read or written simultaneously.

To organize the low-level media structure, each bit is identified by the triple $<x,y,tip>$

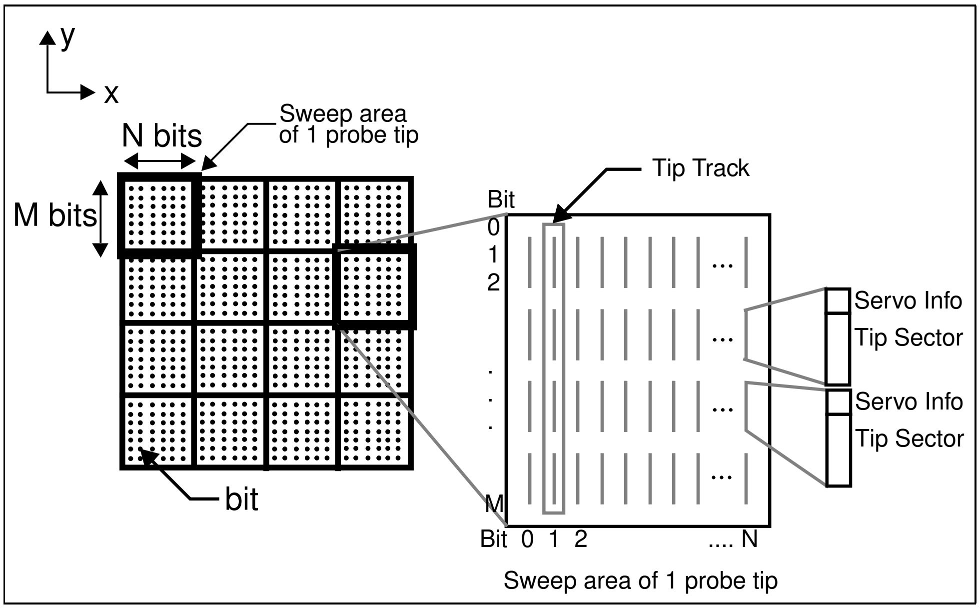


Figure 5: ***Data organization of MEMS-based storage.*** *The illustration depicts a small portion of the magnetic media sled. Each rectangle outlines the area accessible by a single probe tip, with a total of 16 tip regions shown. (A full device contains thousands of tips and tip regions.) Each region stores N×M 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 10% of the device capacity. To read or write data, the media passes over the active tip(s) in the* $\pm Y$ *direction while the tips access the media.*

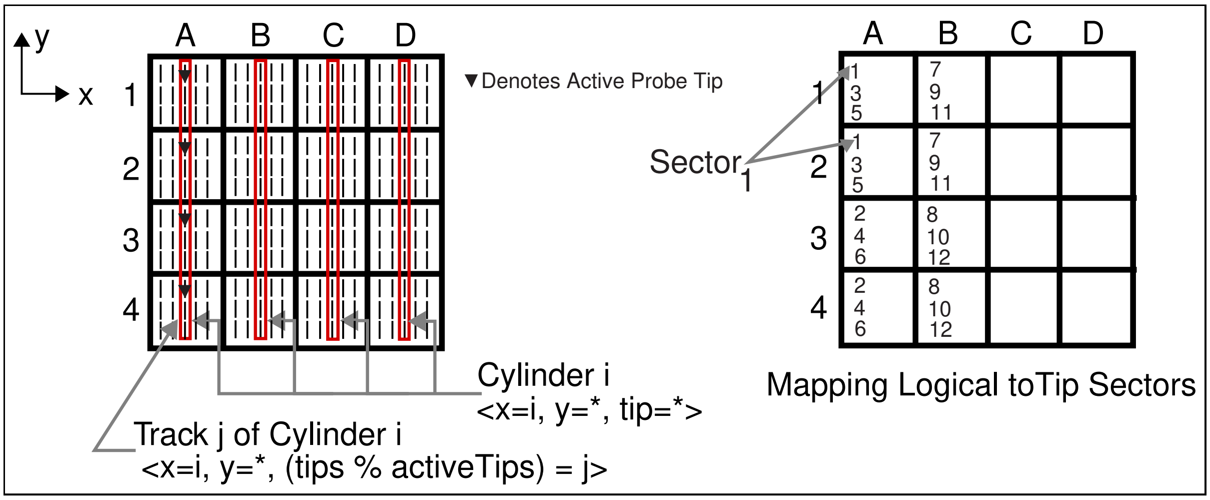


Figure 6: ***Cylinders, Tracks, and Sectors.*** *$Cylinder_i$ is defined as all of the columns of data with the same X coordinate: <x=i, y, tip>. $Track_{i,j}$ is the subset of a cylinder that is accessible by the concurrently active tips: <x=i, y, (tip % activeTips) = j>. (Note that activeTips=4 in this figure and that the tips are linearly numbered such that A1=0, A2=1, etc.) Each logical sector in the figure to the right consists of two tip sectors. For example, $Sector_1$ consists of the first tip sectors of the two upper tip regions, A1 and A2.*

where <*x,y*> represent bit coordinates within the region addressable by <*tip*>. Each active tip reads or writes data within a column of bits (called a *tip track*; see Figure 5) as the media sled moves along the Y axis. A tip track contains M bits, each with identical values for <*x,tip*>. Drawing an analogy to disk terminology, the set of all bits with identical values for <*x*> is called a *cylinder* (shown in Figure 6). In other words, a cylinder consists of all bits that are accessible by any tip without moving the sled along the X axis; there are N cylinders per device. Because only a subset of probe tips can be active at once (recall the power and heat considerations above), cylinders are divided into *tracks*. A track consists of all bits within a cylinder that can be read or written by concurrently active tips. In Figure 6, tips A1, A2, A3 and A4 are active and the corresponding track is indicated. As with conventional disks, reading or writing a complete cylinder requires multiple passes with track switches (*i.e.*, switching which tips are active) in between.

Because multiple tips are active simultaneously, logical sectors can be striped across tip sectors (in multiple tip tracks) to reduce access time. Figure 6 illustrates a layout where each logical sector is striped across two tip sectors. In order to read logical sectors 1 and

2, tips A1 through A4 are activated while the sled seeks to the top of cylinder 2 and moves down (in −Y) across the first tip sector. Tip A1 reads half of logical sector 1, tip A2 reads the other half, and tips A3 and A4 read logical sector 2.

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 under the first bit of the pre-sector servo information, and the sled must be moving in the correct direction at the right velocity ($v_x = 0$, $v_y = \pm v_{access}$). Managing this can be tricky: whenever the sled moves in X (*i.e.*, the destination cylinder differs from the starting cylinder), extra *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[1] before damping to $v_x = 0$. In addition, the spring restoring force (which may be as large as ±75% of the sled actuating force) makes the sled acceleration a function of instantaneous sled position.

The media access requires constant velocity in the Y dimension. This *access velocity* is a design parameter and is determined by the maximum per-tip read and write rates, the bit width, and the sled actuator force. Large transfers may require that data from multiple tracks and/or cylinders be accessed. To switch tracks during large transfers the sled performs a *turnaround* (reversing direction such that $<x,y>_{final} = <x,y>_{initial}$ and $v_{final} = -v_{initial}$) and may switch the set of active tips. Because of the spring restoring force mentioned above, turnaround time is a function of both instantaneous sled position and direction of motion. The turnaround time is expected to dominate any additional activity, such as the time to switch tips, during both track and cylinder switches.

2.2 MEMS device characteristics

MEMS-based storage devices have a rich set of physical characteristics (e.g., acceleration, read/write velocity) and architectural characteristics (e.g., layout of data, number of sleds)

[1] Actually, $time_{settle}$ is the time before the amplitude of oscillation in X damps to become smaller than a percentage of the bit cell width. The sled also oscillates in Y; the magnetic sensing logic is expected to compensate for this motion. If such circuitry were not available, the sled could instead seek to a position some distance before the first servo bit to allow time for damping.

	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 force	+/–	–	+/-	+			+
increasing # of sleds	–	–	–	+	–	??	+
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.*

that directly impact the capacity, bandwidth, latency, reliability, and power consumption of this new technology.

Of course, physical characteristics often constrain architectural designs. For example, packaging and power dissipation constraints limit the number of tips that can be simultaneously active. A recent analysis [CBF$^+$00] estimates power consumption about 1 mW per active tip and 100 mW of overhead for the media positioning system. In a design with 10,000 tips/cm^2, using all of the tips simultaneously would consume 10.1–30.1 W/cm^2, which is an order of magnitude more power than low-cost plastic packaging can tolerate (*e.g.*, about 1 W). For this reason we limit the number of concurrent tips in our models to only 640–3200 tips[2].

Clearly, there are a number of different physical parameters one must consider when exploring how to construct MEMS-based storage devices. Those that are the most important from an architectural point of view are listed in Table 1. To help understand these parameters and their relationship to performance, the following section provides some basic intuition about these physical parameters, technology trends that enable physical improvements (e.g., decreasing bit size), and their relationship to device performance characteristics.

[2]In the past, disk drives have used multiple heads for accessing data in parallel; however, cost makes this prohibitively expensive.

2.3 Physical Characteristics and Trends

Table 1 lists the most important physical characteristics of MEMS-based storage devices and their trends. The bit size is determined by two things: areal density of the storage media and the resolution of the probe tip. The media sled's magnetic recording materials are similar to current disk drive media coatings, which have achieved a bit density of over 50GByte/in^2 [Pre99], with projected annual growth rates of 60-100% [MD96]. Current models of MEMS-based storage devices use perpendicular recording techniques, which allow for square bit spots. Future disk drives can also utilize this technique, leading to higher media densities. The finer positioning resolution of MEMS actuators, however, will allow the MEMS-based devices to access smaller spots, leading to higher capacities using the same media.

The next two physical characteristics, sled velocity and acceleration, are related. Velocity itself is bounded by the spring and actuator force available to turn the sled around and the available distance between the actuator fingers[3]. As the sled velocity increases, it takes longer (time and distance) to stop the sled and then reverse direction. If the sled travels too fast, the actuator fingers will touch, destroying the device. Therefore, for a fixed force, there is a maximum sled velocity. Maximum velocity can be increased in 4 ways: (1) by making the distance between actuator fingers larger, which is difficult; (2) by increasing the actuator force (i.e., voltage); (3) by alternative actuator designs, such as IBM's micromagnetic actuator; (4) by decreasing the mass of the sled. Decreased sled mass will become possible as manufacturing technology evolves to allow full-strength hollowed-out mechanical structures. Therefore, because $F = ma$, a decrease in the sled mass will effectively yield an increase in actuator force, allowing sled velocity to increase.

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, each with their own actuators.

Finally, increasing error rates is a property of the manufacturing process and magnetic materials used. Often in disk drive design, raw media error rates are increased by usage of

[3]Sled velocity is also bounded by the probe tip's maximum data rate. However, current sled velocities set data rates 2 orders of magnitude below expected probe tip data rates.

higher areal-density materials. Of course, increased media error rates are compensated for by more powerful error-correcting codes. MEMS-based storage can benefit in the same way. Further, it can tolerate probe-tip failures simply by remapping the failed probe tip's data to a functional tip. As the next section will show, the massive number of probe tips on these devices creates an even more powerful opportunity: RAID-like error recovery directly within a single MEMS-based storage device.

2.4 Performance Characteristics and Trends

The physical parameters' impact on overall performance creates an interesting set of relations and trade-offs. Table 1's second column shows the impact of these parameters on sled seek time[4]. Increasing the sled's access velocity increases the seek time in the Y-dimension because it takes longer for the sled to "ramp up" to the access velocity whenever the sled performs a turnaround. (X-dimension time does not change because the initial and final $V_x = 0$.) Of course, seek time decreases 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.

Many of the physical improvements also affect settle time. If the sled vibrations during

[4]Seek time is the time to move the sled from some point A with initial velocity $V_y = V_{access}$ and $V_x = 0$, to some point B, with final velocity $V_y = V_{access}$ and $V_x = 0$.

settle are actively damped by the system, stronger actuator forces[5] 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 (while the sled settles over a smaller area).

Turnaround time can be decreased by increasing the effective actuator force. The increased 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 spring stiffness 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 decrease 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 turnaround near the device center, the spring force is close to zero and has little impact on turnaround time. However, turning around near the device's edge can increase turnaround time by as much as 4X.

Peak (streaming) bandwidth is achieved by having the sled sweep the entire chip 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) provides more time when the probe tips can access data; (3) increasing the number of independent sleds, which decreases each sled's mass and creates parallelism. 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. Therefore, as the time spent reading an entire track decreases and the turnaround

[5] Actuator force is increased by: (1) increasing the voltage applied to the actuators or (2) reducing the sled mass (by improving the sled design or splitting one large sled into N small sleds).

time increases, the device eventually spends more time turning around than reading. At this point, peak bandwidth decreases. Therefore, for a given actuator force, sled mass and spring force, there is a maximum velocity after which peak bandwidth declines.

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 (a 10,000 probe tip device with 100 failed probes would lose 1% capacity).

Power requirements also 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 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 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 devices could easily use 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 tolerate one or multiple tip or sector failures.

	1st gen.	2nd gen.	3rd gen.
bit width (nm)	50	40	30
sled acceleration (g)	70	82	105
access speed (kbit/s)	400	700	1000
access speed (mm/s)	20	35	50
resonant frequency (Hz)	729	729	1008
settle constants	2.0	1.0	1.0
X settling time (ms)	0.431	0.215	0.158
active tips	640	1280	3200
maximum throughput (MByte/s)	25.6	89.6	320
number of sleds	1	1	1
per-sled capacity (GByte)	2.56	4.00	7.11
bidirectional access	no	yes	yes

Table 2: ***MEMS device parameters used in our experiments.***

3 Modeling of MEMS-Based Storage Devices

This section describes the MEMS-based storage models and the simulation techniques used in the experiments described below. Because these devices are in their infancy, timing models are derived from extensive discussions with members of the CMU MEMS project who are actively developing this technology. In return, these results help researchers refine their designs by identifying which device characteristics are most important for system-level performance. A detailed description of the performance model and a detailed exploration of MEMS sensitivity to various design parameters is presented in [GSGN00a].

3.1 3 Generations of Devices

Given the wide range of parameters, exploring the entire MEMS-based storage design space would take a considerable amount of time. To reduce this effort, three models of MEMS-based storage are used, based on anticipated technology advances over the first three generations (Table 2).

The "1st generation (G1)" model represents a conservative initial MEMS storage device, which could be fabricated within the next three years [CBF$^+$00]. Each sled has a full range of motion of 100 μm along the X and Y axes, and the actuators will 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 kbits/s. This design only offers unidirectional accesses, with reads and writes only occurring 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 6,400 tips underneath each sled. The sled travels at 20 mm/s during media access but is not restricted to that speed during "seeks". This yields a raw capacity of 2.56 GB per sled. However, media errors require a 10-bit-per-byte encoding. Also, sled tracking and synchronization information requires 10 bits for every 80 data bits.

The "2nd Generation (G2)" model. Several fundamental improvements enhance G2 over G1. First, media access occurs in both the + and - Y direction. Second, per-tip data rate increases to 700 kbits/s based on trends in probe-tip technology. An 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 number of settling time constants that are included 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 will increase G2's bit density by (at least) 20%.

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

The Reference disk. A validated DiskSim module [SG99] for the Atlas 10K TM09100W [Qua99] 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/second. Its seek time drops to a 3 ms average and rotates at 20,000 RPM. The Atlas 10K and SuperDisk parameters are compared in Table 3.

	Atlas 10K	**"SuperDisk"**
RPM	10,025	20,000
Max Bandwidth (MB/s)	25	125
data surfaces	6	12
average rotational latency	2.21 ms	1.36 ms
average seek (read/write)	5.7 ms/6.19 ms	3.12 ms/3.58 ms
max full stroke	10.83 ms/11.32 ms	8.50 ms/8.96 ms

Table 3: ***A comparison of the Quantum Atlas 10K TM09100W disk drive and the extrapolated SuperDisk model.*** *Specifications for the Atlas10K are from [Qua99, SG99].*

3.2 Simulation Environments

Using the model described above and in [GSGN00a] and using the device parameters in Table 2, we developed simulation models for each MEMS device and integrated those models into DiskSim, a freely-available disk simulator that very accurately models disk drives [GWP98], including the Atlas 10K. DiskSim was used for the microbenchmark and trace-based experiments described below. For the application experiments, DiskSim was integrated with SimOS [RHWG95]. SimOS models 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 [IBMa] was used to compare against our MEMS 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 technology must offer a significant improvement in I/O and overall application performance. For mobile applications, power dissipation is also crucial. Using microbenchmarks and six different applications, this section compares the performance and power usage of our MEMS-based storage device models (G1, G2, and G3) against a 1999 Atlas 10K disk drive and the hypothetical SuperDisk described above.

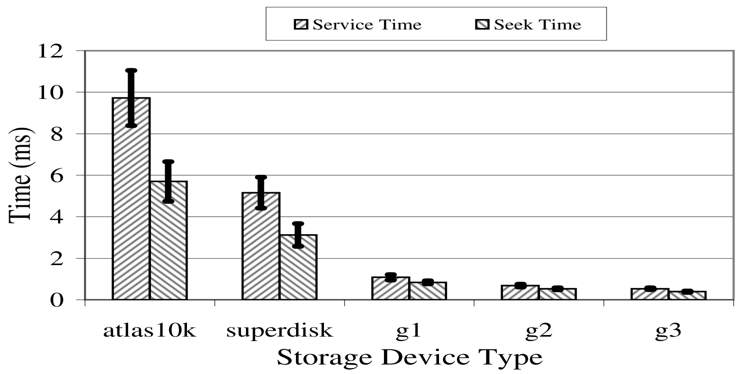


Figure 7: ***Average total response times of each model under the microbenchmark.*** *Interesting features to note are the overall better performance from the MEMS devices and their smaller variances.*

4.1 Microbenchmark Results

The first workload of interest is a simple microbenchmark of 10,000 randomly-distributed requests. Two thirds of the requests were reads, and the arrival rate was 20 requests per second. Figure 7 shows that all three MEMS models outperform the Atlas 10K and SuperDisk disks by almost 10X and 5X, respectively.

Figure 7 also shows that the MEMS 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 quite a bit, causing the wide variation in access time. In this experiment, the coefficients of variation ($\frac{\sigma}{\mu}$) for the Atlas and SuperDisk access times are 0.76 and 0.79, respectively. In contrast, the MEMS-based storage devices have coefficients of variation between 0.18 and 0.20. The small variation is due to spring effects, the absence of rotational latency, and the fact that a full throw of the media is on the order of 100 microns as compared to several centimeters in a disk drive. Therefore, seeks times are tightly constrained. The lower variances, and thus greater potential predictability, has intriguing consequences for the design of embedded systems with both storage and 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-based storage 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 provided 4 times the throughput. Early results indicate that inter-sled stripe unit trade-offs conform to the expectations given by earlier disk striping work [RB89, CP90]. In addition, similar benefits can be gained by aggregating multiple single-sled devices together, as in a RAID system. Given the significantly lower volume of MEMS-based storage devices, many independent sleds could be fit into a standard drive enclosure, increasing both the performance and the capacity per volume relative to conventional disks.

4.2 Application Results

This section presents the results of running some real-world benchmarks and applications on systems with simulated MEMS-based storage devices in two different configurations: first, as a simple replacement for disks and second as a non-volatile disk cache.

Comparing MEMS-based storage devices to disks. The first two applications, the Andrew Benchmark Suite [HKM+88] and PostMark [Kat97] 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 web commerce environments. Figures 8 and 9 show that MEMS-based storage devices can significantly reduce the I/O time for these workloads. For Andrew, the G2 MEMS-based storage device provides a modest additional reduction in I/O wait time beyond G1. The improvement is due to G2's ability to access data as the sled moves in either Y-direction (i.e., up or down). The percentage change looks about the same from G1 to G2 as from G2 to G3.

The data for PostMark (Figure 9) shows a dramatic benefit for MEMS-based storage devices even when compared to the SuperDisk. This impressive improvement comes from a fundamental physical difference in how MEMS-based storage accesses data. Specifically, the frequent create and delete operations in PostMark cause repeated synchronous writes to file system metadata, forcing the storage devices to make same sector, back-to-back updates. For a conventional disk, such back-to-back same-sector accesses require a full rotation (typically 6–8 ms on today's disks) between updates. This explains why PostMark spends much of its I/O time waiting for full disk rotations. MEMS-based storage does not involve rotating platters, and so the MEMS models do not suffer from these full rotation latencies for back-to-back rewrites. Specifically, MEMS-based storage can write a sector, immediately reverse direction and be in position to access the sector again in 0.063 ms. This physical difference gives MEMS-based storage a fundamental performance advantage over rotating media for this access pattern. While this specific problem could be significantly reduced with a small amount of write-back caching (either in the file system or at the disk), similar behavior is exhibited by many read-modify-write activities, such as transaction processing and RAID parity updates.

The next set of benchmarks, GNULD (a simple benchmark in which a large set of object files are linked using the gnu linker) and the TPC-D [Cou98] queries, also show significant performance improvements for MEMS-based storage. However, Figure 10 shows that the SuperDisk provides almost the same performance as the G1 MEMS-device for TPC-D query 4. This is because SuperDisk's higher streaming bandwidth more than compensates for the higher access times for this data mining query. However, a disk drive's streaming bandwidth varies by ∼40%, depending on the location of the data (i.e., outer *vs.* inner tracks). For these experiments, all of the data is located on the disk's outer tracks, making the performance best-case. In contrast, MEMS devices do not have any variation in streaming bandwidth (for contiguous data). Therefore, if the data had resided on SuperDisk's inner (i.e., slower) tracks, SuperDisk's performance would have been much lower. With their increased bandwidth and lower access times, the G2 and G3 MEMS device's outperform the SuperDisk.

The results for TPC-D query 6, shown in Figure 11, show the expected result for workloads that are CPU-bound rather than I/O-bound — eliminating the I/O stall time provides only a modest 8% decrease in overall runtime. As CPU speeds continue to increase relative to disk speeds, of course, the importance of I/O increases.

For several of the benchmarks, CPU time decreases slightly with the better-performing MEMS devices. All of these decreases are in the system time charged to the application. The reason for the decrease is that shorter run times reduce the amount of time an application can be charged for general system overhead, such as I/O interrupt handling. Therefore, system time will generally decrease by a modest amount when applications complete in less time.

MEMS-based storage devices as caches for disks. MEMS-based storage can also be used as a non-volatile addition to the storage hierarchy. With their low-cost entry point, MEMS-based storage devices could be incorporated into future disk drives as a very large (1-10 GByte) non-volatile MEMS cache. With their superior performance, the MEMS cache could 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 non-volatile storage to absorb synchronous disk writes both at a client and at a file server increases performance from 20% to 90%[BAD+92].

To explore MEMS-based storage as a non-volatile cache for disk, DiskSim was augmented to allow a MEMS device to serve as a cache for a disk. The MEMS cache was 2.5 GB, the disk was 9.2 GB, and the workload was the 1-day cello trace from [RW93]. This trace actually includes eight separate devices so the experiments use a single cache per device. The results show that the average I/O response time is: 14.66 ms for an Atlas10K disk drive without any MEMS cache; 4.03 ms for a disk with a G2 type MEMS-cache and 2.76 ms using just a large G2 MEMS device (instead of disk). Since most of the read requests are serviced from the client-side DRAM cache, the MEMS cache 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 non-volatile, providing the same data integrity guarantees as disk drives. In addition, an experiment in which all eight devices were re-mapped to a larger version of the Atlas10K disk with a single MEMS cache had a slightly higher average access time of 4.66 ms. This longer service time stems from an increase in queueing since the large device is doing the work of eight. It shows, however, that caching absorbs enough of the device's activity to give 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 deliberately allocate specific data onto it.

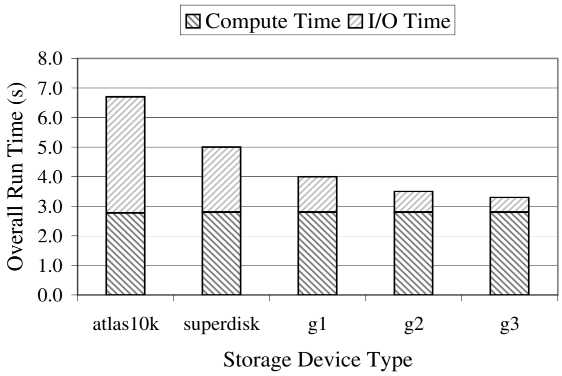


Figure 8: ***Runtime for the Andrew Benchmark.***

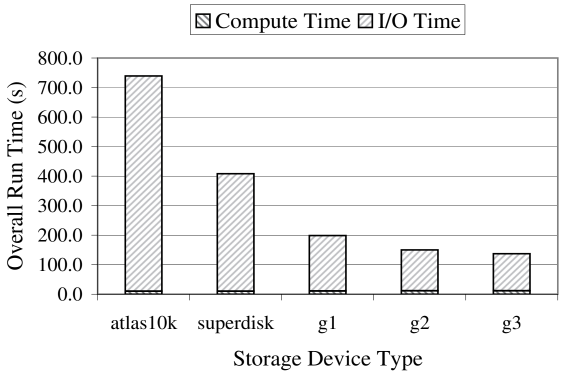


Figure 9: ***Runtime for the PostMark Benchmark.***

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 [WGSS95]. Further, because RAID arrays are less cost-sensitive than individual disks, arrays of MEMS devices could be incorporated more cost-effectively.

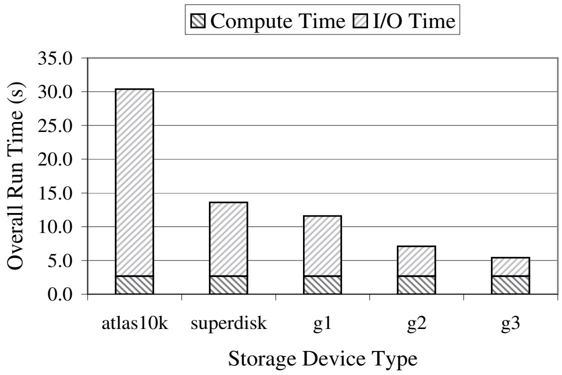


Figure 10: ***Runtime for TPC-D Query #4.***

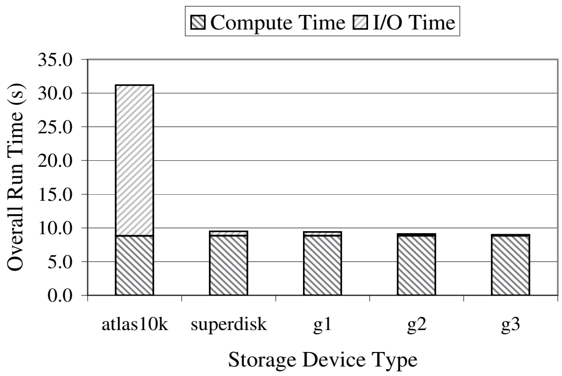


Figure 11: ***Runtime for TPC-D Query #6.***

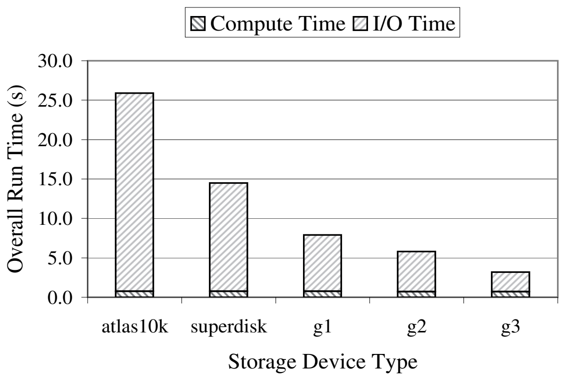


Figure 12: ***Runtime for Gnuld.***

4.3 Power Results

The physical characteristics of MEMS-based storage devices make them potentially much less power hungry than disk drives—even low-power drives such as the IBM Travelstar [IBMb, IBMc]. This power advantage comes from several sources. The bulk of power usage in disk drives is in keeping the disk spinning. 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.

Of course, considerable power can be saved by turning off the drive spindle during long idle periods. Low-power drives call this standby mode. Numerous studies have demonstrated the power saving of standby mode [LSM99, HIR94, DKM94, LKHA94], and current low-power drives do incorporate this feature. MEMS-based storage can also use a standby mode, stopping sled movement during idle times. Further, the sled's low mass allows MEMS to quickly move between active and standby mode (0.5 ms), where a low-power drive requires on the order of 2 seconds to 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.

Another power saving comes from the electronics of a MEMS device. 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. 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 watts [IBMb, IBMc] 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 milliseconds).

However, single-chip, self-contained MEMS-based storage devices have much smaller parts and fewer manufacturing variations. This allows MEMS to spend much less power in the positioning system ([CBF+00] estimates total MEMS electronics power to be 1 Watt). Also, MEMS-based storage can adjust its power consumption during data accesses by reading or writing at a smaller granularity than, say, 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[6]. The device only needs to activate as many tips as are necessary to satisfy a request, which could result in a substantial reduction of unnecessary power drain.

The model for disk drive power is based on IBM's low-power Travelstar disk using IBM's low-power drive management techniques described in [IBMb, IBMc]. The device has 5 power modes: 1) active mode (data is being accessed) consumes 2.5 watts for reads and 2.7 for writes; 2) performance idle (some electronics are powered down) consumes 2.0 watts; 3) fast idle (head is parked and servo control is powered down) consumes 1.3 watts; 4) low-power idle (heads are unloaded from the disk) consumes 0.85 watts; 5) standby (spindle motor is stopped) consumes 0.2 watts. From [IBMa], 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 device, power per access is computed using the physical parameters in [CBF+00]. The results show each probe tip consumes 1 milliwatt. Given the power budget of 1 watt, the simulated MEMS device is limited to no more than 1,000 probe tips simultaneously active. Given the physical characteristics of the 2nd generation device (see Table 2), the maximum bandwidth was 89 MBytes per second. Further, given the sled design, the power consumed to keep the sled in motion is 0.1 watt. Therefore, the maximum

[6] 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.

	Andrew		Postmark		Netscape	
Category	Disk	MEMS	Disk	MEMS	Disk	MEMS
active	14.026	1.400	1378.162	188.859	321.211	32.121
perfIdle	13.025	0.000	1003.969	0.000	1924.100	0.000
goToActive	0.880	0.000	0.000	0.000	513.480	0.000
fastIdle	2.400	0.000	0.000	0.000	1799.885	0.000
lowPowerIdle	2.722	0.000	0.000	0.000	1000.467	0.000
spinup	0.000	0.615	0.000	86.933	228.800	1.288
standby	0.000	0.744	0.000	32.517	308.885	1318.763
Total (Joules)	33.053	2.759	2382.131	308.309	6096.828	1352.172

Table 4: ***Comparison of energy required to execute two workloads using disks and MEMS-based storage devices.*** *All numbers are given in Joules.*

power for this MEMS device is 1.1 watts. Finally, standby power consumption is 0.05 watts.

Table 4 shows that the total energy consumed for the MEMS device is between a factor of 4X and 10X lower, depending on the application. For highly active workloads like Andrew and Postmark, most of saving comes directly from MEMS's lower energy consumption during data accesses (active mode). For example, for the Andrew Benchmark, the disk drive uses 14 Joules to access data while MEMS-based storage uses only 1.4 Joules. However, for more interactive workloads such as Netscape, most of the power saving comes from MEMS-based storage's ability to aggressively use its low-power standby mode. In contrast, the disk drive spends much of its power moving between the various modes.

5 Potential of MEMS-based Storage and Computation

The results from Section 4 show that MEMS-based storage devices have significant advantages over disk drives. I/O performance can increase by an order of magnitude over disk drives and the physical characteristics of MEMS provide some fundamental performance advantages that rotating media cannot compete against. Further, unlike conventional disk caches, which often use volatile RAM, a MEMS-based storage disk cache creates significant performance improvements without risking possible loss of data integrity. 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 many portable applications such as notebook PCs, PDAs, and video camcorders, MEMS-based storage provides a more robust and lower power solution. Many of these devices cannot cope with device rotation (e.g., rapidly turning a PDA) and are very sensitive to shock (e.g., dropping a device). MEMS-based storage does not suffer any gyroscopic effects and can absorb much greater external forces.

Further, 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 devices can ride the linear decline in IC manufacturing process costs. However, large capacity drives enjoy a 10X price advantage for high-capacity storage (*e.g.,* 50 GB in 1999) 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 MBytes of compressed data per patient; high-resolution MRI's 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, the tape systems and storage management costs are tremendous. Further, the time to access data from a tape is on the order of an hour, including the time to retrieve the tape, insert it into a tape drive, and seek to the data (which could be at the end of the tape). This is often far too much time during a medical emergency and has lead many to believe that disk may soon replace tape for many high-capacity applications.

Write-once MEMS devices provide an attractive alternative to tape. With areal densities 100X greater than write-many MEMS devices and 10X greater than high-capacity tape [KBK97], it should be cost-effective to build storage "bricks" that hold thousands of MEMS devices. Each brick would hold Terabytes or Petabytes of data that could be accessed in seconds. Power dissipation would not be a problem for these devices because, while the devices are very densely packed, data accesses should occur very infrequently, and use a small number of the chips within the brick. Therefore, each brick would only need to dissipate a few watts of power.

Further, by incorporating logic into the MEMS-based storage device, it would be possible to process or translate the data directly within the storage brick. 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. These interface functions could be implemented directly within the device. Finally, with massive numbers of storage bricks there is massive computational parallelism available, creating the ultimate active disk [Rie99].

Another application domain for MEMS-based storage is bulk non-volatile 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.*, bridges, walls, roadways), weather or seismic tracking, and medical applications. For example, 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, non-volatile storage device to buffer data could be used. 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.

Another compelling opportunity presented by MEMS-based storage is near-absolute 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 shows there is a diverse set of potential device designs, which 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 3–10X, with overall performance improvements ranging from 1.6–8.1X. 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 10X 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 [GSGN00b]. Further demonstrations in the mobile and archival storage domains should also show the utility of MEMS-based storage in systems.

References

[BAD+92] M. Baker, S. Asami, E. Deprit, J. Ousterhout, and M. Seltzer. Non-volatile memory for fast, reliable file systems. In *ASPLOS*, pages 10–22, October 1992.

[Bro98] C. Brown. Microprobes promise a new memory option. *EE Times*, pages 6,41,44, January 12 1998.

[BS97] M. Ross B. Schechter. Leading the way in storage. *IBM Research Magazine*, 35(2), 1997.

[CBF+00] L. Richard Carley, James A. Bain, Gary K. Fedder, David W. Greve, David F. Guillou, Michael S.-C. Lu, Tamal Mukherjee, Suresh Santhanam, Leon Abelmann, and Seungook Min. Single chip computers with MEMS-based magnetic memory. *Journal of Applied Physics*, 87(to appear), 2000.

[Col99] Rick Colsen. Sorting Disk Blocks to Reduce Load Times. Personal Communication, Intel Corporation, 1999.

[Cou98] Transaction Processing Performance Council. TPC benchmark d (decision support) standard specification. http://www.tpc.org/benchmark_specifications/TPC_D/210.pdf, February 1998.

[CP90] P. Chen and D. Patterson. Maximizing throughput in a striped disk array. In *International Symposium on Computer Architecture*, pages 322–331, June 1990.

[D+99] M. Despont et al. In *Proceedings of MEMS 1999*, pages 564–569, January 1999.

[Dav99] T. Davis. Realizing a completely micromechanical data storage system (kionix, inc). In *Diskcon 99 International Technical Conference*, September 1999.

[DKM94] F. Douglis, P. Krishnan, and B. Marsh. Thwarting the power-hungry disk. In *Winter USENIX*, pages 292–306, January 1994.

[FSR+96] 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, February 1996.

[Gra97] J. Gray. What Happens When Processing, Storage, and Bandwidth are Free and Infinite. In *IOPADS Keynote*, November 1997.

[GSGN00a] John Linwood Griffin, Steven W. Schlosser, Gregory R. Ganger, and David F. Nagle. Modeling and Performance of MEMS-Based Storage Devices. In *ACM SIGMETRICS 2000*, page to appear, June 2000.

[GSGN00b] John Linwood Griffin, Steven W. Schlosser, Gregory R. Ganger, and David F. Nagle. Operating Systems Management of MEMS-based Storage Devices. Technical Report CMU-CS-00-137, Carnegie Mellon University School of Computer Science, May 2000.

[GWP98] G. Ganger, B. Worthington, and Y. Patt. The DiskSim Simulation Environment Version 1.0 Reference Manual. Technical Report CSE-TR-358-98, The University of Michigan, Ann Arbor, February 1998.

[HIR94] 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, October 1994.

[HKM+88] J. Howard, M. Kazar, S. Menees, D. Nichols, M. Satyanarayanan, R. Sidebotham, and M. West. Scale And Performance Of a Distributed File System. *ACM TOCS*, 6(1):51–81, February 1988.

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

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

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

[Kat97] J. Katcher. PostMark: A New File System Benchmark. Technical Report TR3022, Network Appliance, October 1997.

[KBK97] S. Khizroev, J. Bain, and M. Kryder. Considerations in the Design of Probe Heads for 100 Gbit/in2 Recording. *IEEE Trans. Magnet.*, 33(5):2893–2895, 1997.

[LKHA94] 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, January 1994.

[LSM99] Y-H. Lu, T. Simunic, and G. De Micheli. Software controlled power management. In *7th International Workshop on Hardware/Software Codesign*, pages 157–161, May 1999.

[MD96] C. Denis Mee and E. D. Daniel. *Magnetic Storage Handbook - Second Edition*. McGraw-Hill, 1996.

[Nan99] Nanochip Inc. Nanochip, Inc Product Overview. In *Diskcon 99 International Technical Conference*, September 1999.

[PH96] David A. Patterson and John L. Hennessy. *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann Publishers, Palo Alto, California, 2nd edition, 1996.

[Pre99] IBM Pressroom. IBM Sets Another Disk-Drive World Record. http://www.ibm.com/press/prnews.nsf/oct, October 1999.

[Pug71] E. Pugh. Storage Hierarchies: Gaps, Cliffs, and Trends. *IEEE Transactions on Magnetics*, pages 810–814, December 1971.

[Qua99] Quantum Corporation. *Quantum Atlas 10K 9.1/18.2/36.4 GB Ultra 160/m S Product Manual III SCSI Hard Disk Drives: Ultra SE SCSI-3 Version*, August 1999.

[RB89] A. Reddy and P. Banerjee. An Evaluation of Multiple-Disk I/O Systems. *IEEE Transactions on Computers*, 38(12):1680–1690, December 1989.

[RHWG95] M. Rosenblum, S. Herrod, E. Witchel, and A. Gupta. Complete Computer System Simulation: The SimOS Approach. *IEEE Parallel & Distributed Technology*, 3(4), Winter 1995.

[Rie99] Erik Riedel. *Active Disks - Remote Execution for Network-Attached Storage*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, November 1999.

[RW93] C. Ruemmler and J. Wilkes. Unix disk access patterns. In *Winter USENIX Conference*, pages 405–420, jan 1993.

[SG99] J. Schindler and G. Ganger. Automated Disk Drive Characterization. Technical Report CMU-CS-99-176, Carnegie Mellon University School of Computer Science, November 1999.

[V$^+$99] P. Vettiger et al. In *Prceedings of the 10th International Conference on Scanning Tunneling Microscopy (STM-99)*, page 4, July 1999.

[WGSS95] J. Wilkes, R. Golding, C. Staelin, and T. Sullivan. The HP AutoRAID Hierarchical Storage System. In *15th ACM SOSP*, pages 96–108, December 1995.