Proceedings of ACM SIGMETRICS 2000, Santa Clara, California, June 17-21, 2000.

Modeling and Performance of MEMS-Based Storage Devices

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

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

ABSTRACT

MEMS-based storage devices are seen by many as promising alternatives to disk drives. Fabricated using conventional CMOS processes, MEMS-based storage consists of thousands of small, mechanical probe tips that access gigabytes of high-density, nonvolatile magnetic storage. This paper takes a first step towards understanding the performance characteristics of these devices by mapping them onto a disk-like metaphor. Using simulation models based on the mechanics equations governing the devices' operation, this work explores how different physical characteristics (*e.g.*, actuator forces and per-tip data rates) impact the design trade-offs and performance of MEMS-based storage. Overall results indicate that average access times for MEMS-based storage are 6.5 times faster than for a modern disk (1.5 ms vs. 9.7 ms). Results from filesystem and database benchmarks show that this improvement reduces application I/O stall times up to 70%, resulting in overall performance improvements of 3X.

1. INTRODUCTION

Magnetic disks have been the mainstream on-line secondary storage technology for over 30 years. Disks have maintained this dominant position despite the introduction of technologies such as bubble memory, holographic stores, and improved DRAM devices. A new storage technology, based on microelectromechanical systems (MEMS), has come forward promising significant performance and cost improvements relative to magnetic disks. With these promised improvements, MEMS-based storage could play a large role in future storage hierarchies, replacing disks in some systems (*e.g.*, portable and wearable computers) and complimenting them in others (*e.g.*, as caches for RAID arrays).

MEMS are very small-scale mechanical devices—on the order of 10s to 1000s of micrometers—that slide, bend and deflect in response to electrostatic, electromagnetic, and external environmental forces. MEMS devices are created using photolithographic processes similar to those used in the manufacturing of semiconductor devices, allowing MEMS devices to be directly integrated with processing elements and logic on the same silicon chip [3; 12; 22].

One promising MEMS-based storage design consists of an array of miniature read/write heads (called "probe tips") held under a rectangular sled coated with magnetic recording media. Instead of rotating, the media sled translates along the X and Y axes to seek to new locations. Data are stored and retrieved by positioning and moving the sled over the probe tip array while the tips transfer data. Small prototypes of MEMS-based storage have already been demonstrated, and large-scale devices are the goal of major efforts at many research centers, including Carnegie Mellon [1], IBM, HP, and UC-Berkeley.[1]

Although disk performance has improved dramatically over the past 30 years, there have been no fundamental changes in the mechanical operation or layout characteristics of disks; actuators still position read/write heads over concentric media tracks, and data bits are still transferred by rotating the media under individual heads. Years of research and experience have provided a healthy understanding of disk operation and performance, enabling the creation of useful and accurate models [4; 8; 15; 18; 21].

Like disks, MEMS-based storage devices have mechanical and layout characteristics that determine their performance under given workloads. Sleds suffer from mechanical positioning delays when seeking to new data locations, and data bits are stored in columns (analogous to disk tracks) with various delays involved in moving between these columns. However, because MEMS-based storage devices are not composed of rotating platters and voice-coil actuation components, their performance characteristics differ significantly from those of magnetic disks. To assist designers of both MEMS-based storage devices and the systems that use them, an understanding of these characteristics must be developed.

This paper takes a first step towards developing this understanding. We describe the mechanical and operational characteristics of MEMS-based storage devices under development at Carnegie Mellon [11]. We use classical mechanics equations to create a simulation model for these devices. Using this simulation model, we explore device performance and sensitivity to key design parameters. The results show that MEMS-based storage devices achieve access times that

[1] Although related, MEMS-based storage devices should not be confused with Magnetoresistive RAM (MRAM), another emerging nonvolatile storage technology, nor with MEMS micropositioners for disk heads, which incorporates MEMS structures into existing disk drives.

Appears in *Proceedings of ACM SIGMETRICS 2000,* June 2000.

are 6.5X faster than a modern disk. Sensitivity studies expose the important parameters in MEMS-based storage device performance. Our system-level application studies show that faster access times result in a 3X improvement of end-to-end performance for filesystem and database benchmarks.

The remainder of this paper is organized as follows. Section 2 describes MEMS-based storage devices in general and the sled-based design in detail. Section 3 describes a performance model for MEMS-based storage devices and explores its performance characteristics. Section 4 compares end-to-end performance for systems using MEMS-based storage devices to those using disk drives. Section 5 summarizes this paper's contributions and discusses ongoing research.

2. MEMS-BASED STORAGE

This section describes high-level designs for two different MEMS-based storage devices and gives a detailed description of the more promising design. The detailed description maps the device's access and layout characteristics onto a disk-like metaphor to clarify similarities and differences.

2.1 Device Designs

MEMS-based microstructures can be used to build storage devices in a variety of ways. Tradeoffs in the design process affect the robustness, manufacturability, cost, capacity, access speed and latency of these devices. As an example, Figure 1 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, quickly moving the tip to different bit positions in a small accessible area. Once positioned, the probe tip can read or write bits using standard magnetic recording techniques. Since the only moving part is the nearly massless cantilevered beam, these structures have very quick positioning times (on the order of 100s of microseconds). Unfortunately, the space efficiency of this design 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. While this design is useful for visualizing MEMS-based storage, expected capacities of only tens of megabytes per device limits its practicality in comparison to Flash Memory and other nonvolatile RAM components.

A more space-efficient design is shown in Figures 2 and 3. Here, a movable media sled is suspended by 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 could be placed. Assuming a bit cell of 0.0025 μm^2 (50 nm per side) and encoding/ECC overheads of 2 bits per byte, this yields a capacity of about 4 gigabytes per square centimeter [11]. A more aggressive goal of 0.0009 μm^2 (30 nm per side) could yield capacities of 11 GB/cm^2 or greater. While this design improves space efficiency to 30–50%, the greater sled mass increases positioning times—a necessary tradeoff to achieve disk-like capacities. Variations on this design enable minute tip deflection in X and Z to allow for skewed tracks and sled surface variations. The remainder of this paper focuses on MEMS-based storage devices based on this "moving media" model.

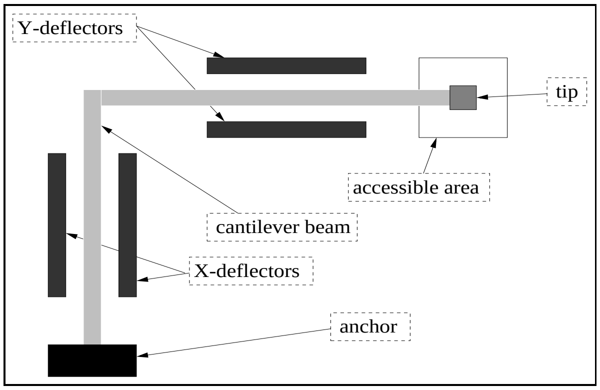


Figure 1: ***A cantilevered-beam probe tip in the "fixed media" model.*** *The X- and Y-deflectors are capable of quickly positioning the tip anywhere in the small accessible area. The overall capacity of this model is limited because only 1% of the cantilever footprint is accessible by the tip.*

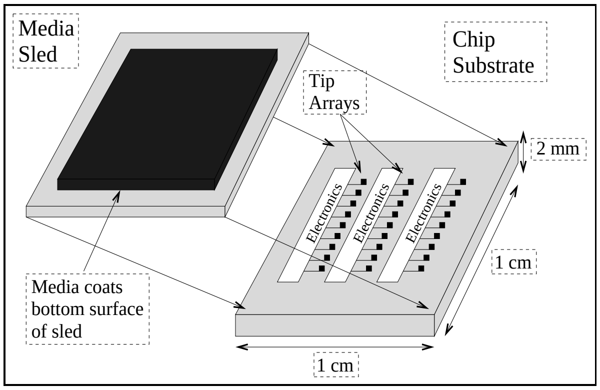


Figure 2: ***The "moving media" model.*** *The media sled is suspended above the array of fixed tips. The sled moves small distances along the X and Y axes, allowing the fixed tips to address 30–50% of the total media area. This yields capacities of gigabytes per square centimeter.*

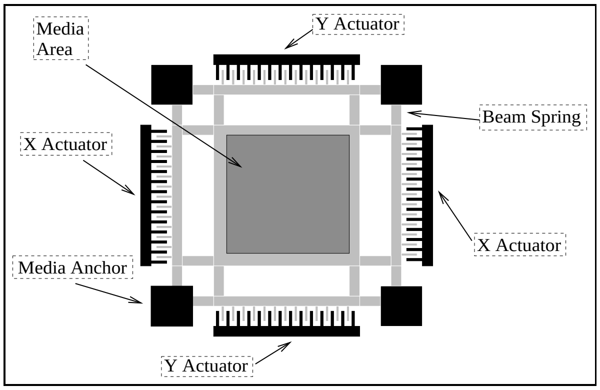


Figure 3: ***The suspended media sled in the moving media model.*** *The actuators, spring suspension, and the media sled itself are shown. Anchored regions are black and the movable structure is shaded grey.*

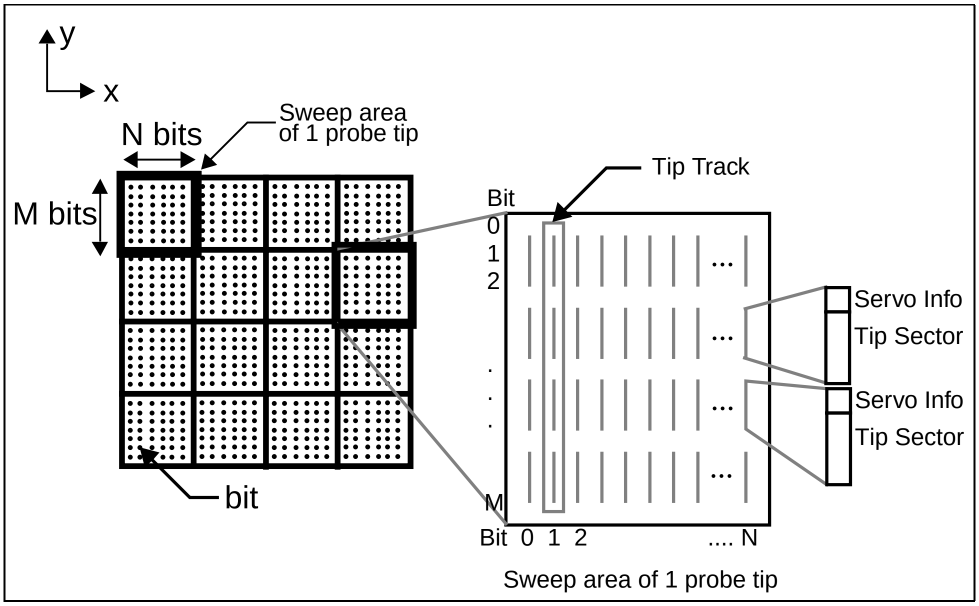


Figure 4: ***Data organization of MEMS-based storage.*** *The illustration depicts a small portion of the magnetic media sled. Each rectangle outlines the area accessible by a single probe tip, with a total of 16 tip regions shown. (A full device contains thousands of tips and tip regions.) Each region 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 ±Y direction while the tips access the media.*

2.2 Device Characteristics and Data Layout

The magnetic media on the sled is organized into rectangular regions as shown in Figure 4. Each rectangular area stores N×M bits (2000×2000 bits in our default model) and is only accessible by one probe tip. Like conventional disks, data are not byte-accessible. The smallest accessible unit of data is a "tip sector" consisting of servo information (10 bits) and encoded data/ECC (80 bits = 8 encoded data bytes). 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. Due to power and heat considerations, it is unlikely that all probe tips can be active simultaneously; rather, we currently expect groups of 200–2000 tips to be the norm.

To organize the low-level media structure, we identify each bit by the triple <*x,y,tip*> 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 4) as the media sled moves along the Y axis. A tip track contains M bits, each with identical values for <*x,tip*>. Drawing on analogies from disk terminology, we refer to the set of all bits with identical values for <*x*> as a *cylinder* (shown in Figure 5). 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 5, 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 5 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. In our default model, logical sectors of 512 bytes are striped across 64 tip sectors of 8 bytes each.

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 and 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 before damping to $v_x = 0$.[2] In addition, the spring restoring force (which may

[2] 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.

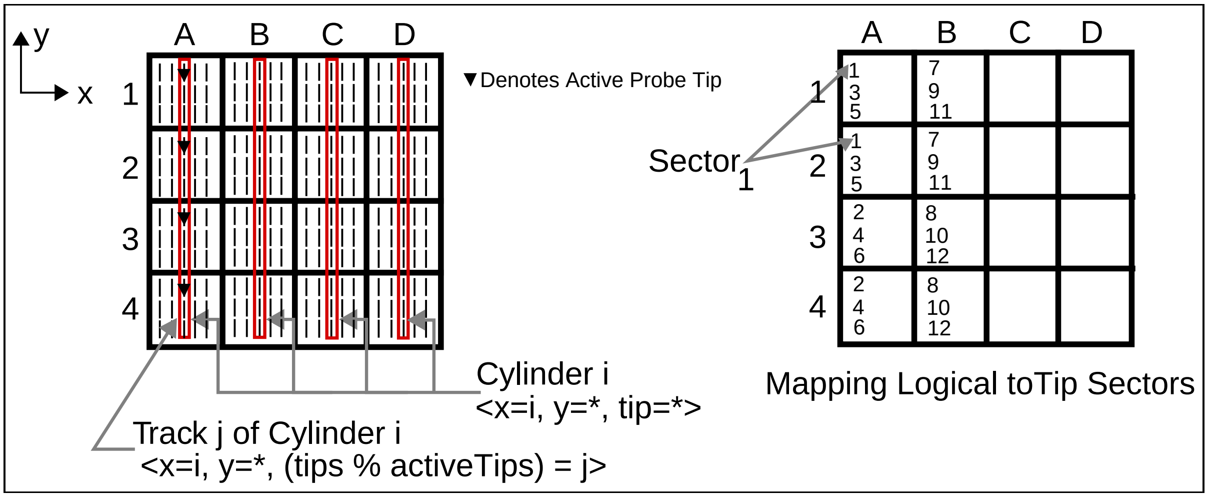


Figure 5: ***Cylinders, Tracks, and Sectors.*** *Cylinder$_i$ is defined as all of the columns of data with the same X coordinate: <x=i, y, tip>. Track$_{i,j}$ is the subset of a cylinder that is accessible by the concurrently active tips: <x=i, y, (tip % activeTips) = j>. (Note that activeTips=4 in 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.*

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.

3. DEVICE PERFORMANCE

This section describes our performance model for MEMS-based storage devices and explores the performance and sensitivity of these devices given reasonable default parameters.

3.1 Computing Device Service Times

When developing a performance model for MEMS-based storage devices, it is useful to first look at a common disk performance model. The service times for a disk access is often computed as:

$$time_{service} = time_{seek} + latency_{rotate} + time_{transfer}$$

The seek time, $time_{seek}$, is a function of the distance in cylinders that the disk arm must travel. This includes an acceleration/deceleration component, a linear component (representing the maximum velocity of the seek arm) for long seeks, and a significant disk arm settling delay (1 ms) for all non-zero seeks. The rotational latency, $latency_{rotate}$, can be computed by dividing the angular distance between the current and destination sector by the rotational velocity. Since disks rotate continuously, detailed simulation requires accounting for all advances in time, including the seek time for the access being serviced. The media transfer time, $time_{transfer}$, can be computed as the product of the number of sectors accessed divided by the number of sectors per track (in the relevant zone) and the time for a full revolution. Detailed models must also account for all track and cylinder boundaries crossed by the range of desired sectors, since each crossed boundary adds a repositioning delay equal to the corresponding skews in the logical-to-physical mapping.

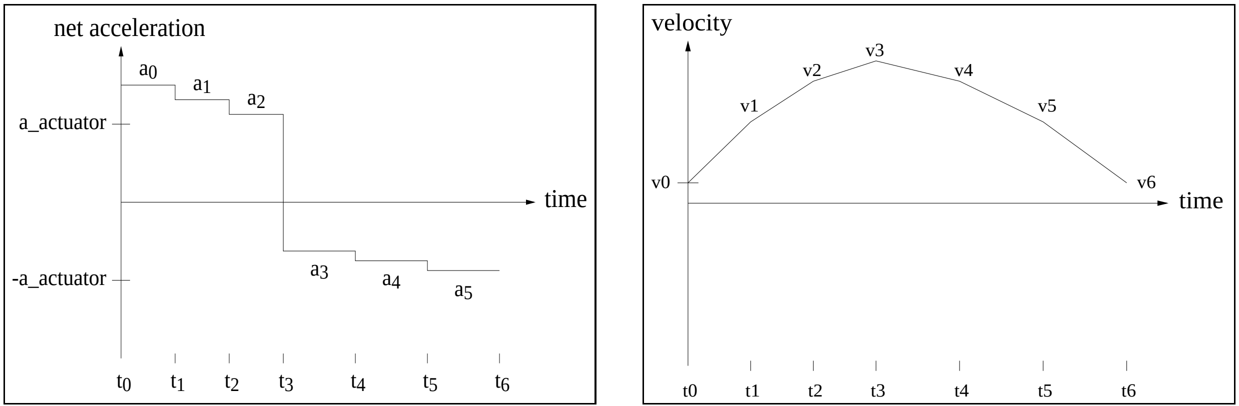
Service times for MEMS-based storage devices can be modeled with a similar equation:

$$time_{service} = time_{seek} + time_{transfer} \qquad (1)$$

The obvious difference is the absence of rotational latency. Less obvious from the equation is the much more complicated nature of the $time_{seek}$ term. Recall that the movable media sled must seek to the correct *<x,y>* position and attain the proper media access velocity in the proper Y direction. The actuation mechanisms and control loops for X and Y positioning are independent, allowing the two to proceed in parallel. Thus,

$$time_{seek} = max(time_{seek\_x}, time_{seek\_y})$$

Computing $time_{seek\_x}$ and $time_{seek\_y}$. Since the sled is a mass moving under a constant force from the actuators, equations from classical first-order mechanics (*e.g.*, $\Delta x = v_0 t + \frac{1}{2}at^2$) can be used to compute both $time_{seek\_x}$ and $time_{seek\_y}$. A seek is broken into two *phases*: acceleration and deceleration. In the acceleration phase, the actuators pull the sled toward the destination. In the deceleration phase, the actuators reverse polarity and decelerate the sled to its final destination and velocity. In addition to the actuator force, the sled springs constantly pull the sled towards



(a) Sled acceleration versus time

(b) Sled velocity versus time

Figure 6: ***Piecewise-constant approximation of acceleration and velocity during a Y-dimension seek.*** *The graph in (a) is the derivative of (b) with respect to time. $a_{actuator}$ is the sled acceleration caused by the actuator force; the net accelerations during each "chunk" are different because of the effects of the spring restoring force. $v_o = v_6 = v_{access}$; in other words, at the end of a seek the sled is traveling at the correct access velocity. In the case of an X seek (not shown), $v_o = 0$. In this example, each phase of the seek is divided into 3 chunks per phase; our model divides each seek into 8 chunks per phase.*

its centermost position. The spring force in each dimension is linear with respect to the sled's displacement (from center) in that dimension, which means that spring force varies as the sled moves.

We use piecewise-constant approximation to determine the spring force's contribution to net acceleration. Each phase of the seek is broken into a set of smaller *chunks*, with the net acceleration in each chunk being the sum of the acceleration due to the actuators and the average acceleration due to the springs. As an example, the acceleration curve for a sled seeking from the outermost position to the centermost position is shown in Figure 6(a). This acceleration curve leads to the velocity curve shown in Figure 6(b). In this example, the springs help during the acceleration phase ($t_0 ... t_3$), but hurt during the deceleration phase ($t_3 ... t_6$). Also, because this example seek moves toward the centermost position, the spring's impact decreases in each chunk as the sled approaches its rest position.

To parameterize the model, the spring force at full displacement is set to a percentage (called $spring\_factor$) of the actuator force. Generally speaking, the spring factor should be a large percentage of the actuator forces since for manufacturability reasons the springs should be as stiff as possible. So, when the sled is at its full displacement, the springs should push back against the actuators with an almost equal force, yielding a high $spring\_factor$.

An expression for the net acceleration at any point x is:

$$a(x) = a_{actuator} \pm \left[(a_{actuator} * spring\_factor) * \frac{offset(x)}{max\_offset} \right]$$

When the actuator is pulling against the springs, the second term will be negative. For each chunk, the constant net acceleration is taken to be the average of the net accelerations at its endpoints:

$$a_i = \frac{a(x_i) + a(x_{i+1})}{2}.$$

Given these constant accelerations, we can compute the velocity of the sled at the end of each chunk:

$$v_i = v_{i-1} + a_{i-1}(t_i - t_{i-1}). \qquad (2)$$

Since the initial position x_0, the initial velocity v_0, and the acceleration during each chunk are all known, the times at the end of each chunk can be computed. To do this, we integrate the velocity curve v_i to find an expression for position x_i:

$$x_i = x_{i-1} + v_{i-1}(t_i - t_{i-1}) + \frac{1}{2}(v_i - v_{i-1})(t_i - t_{i-1}). \qquad (3)$$

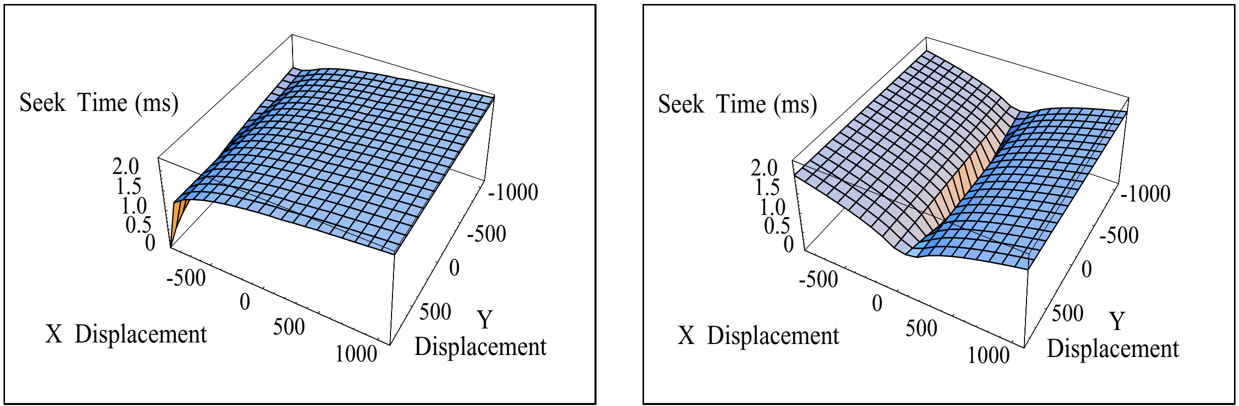
Plugging Equation 2 into Equation 3 yields a quadratic that can be solved for t_i, the time that the sled arrives at the end of chunk i:

$$t_i = \frac{-(v_{i-1} - a_i t_{i-1}) + \sqrt{v_{i-1}^2 + 2a_i(x_i - x_{i-1})}}{a_i} \qquad (4)$$

Extra settling time for $time_{seek\_x}$. Equation 4 describes the base seek time for both the X and Y dimensions. In the X dimension, the sled starts and ends each seek at rest ($v_0 = 0$). Extra settling time, t_{settle}, must be added onto X-dimension seeks to model the time required for the oscillations of the sled-spring system to damp out. t_{settle} is dependent on the resonant frequency of the system, f, which depends on the construction of the sled and the stiffness of the springs.

$$time_{settle} = \frac{1}{2\pi f} * number_{timeconstants} \qquad (5)$$

where $number_{timeconstants}$ is a measure of how much damping is needed before the probe tips can begin to robustly access the media. This oscillation could be damped by the sled-spring system itself or by the atmosphere. More likely, the system will have a closed-loop control system that actively damps the oscillations using the actuators. Active



(a) Seek times from a corner of the media.

(b) Seek times from the center of the media.

Figure 7: ***Seek time profiles for the MEMS-based storage device.*** *These graphs were generated directly from the seek time equations.*

sled mobility in X and Y	100 μm
bit cell width (area)	50 nm (0.0025 $\mu\mathrm{m}^2$)
number of tips	6400
simultaneously active tips	1280
tip sector length	80 bits (8 data bytes)
servo overhead	10 bits per tip sector
device capacity (per sled)	2.1 GByte
sled acceleration	114.8 $\mathrm{m/s}^2$
per-tip data rate	400 kbit/s
settling time constants	1
sled resonant frequency	220 Hz (see note)
spring_factor	75%

Table 1: ***Device parameters used in our experiments.*** *Although MEMS-based storage devices have yet to be completely fabricated and tested, we believe these are reasonable values for initial analyses of these devices. Note: While finishing this paper, we learned of a modified spring design that increases the sled-spring resonant frequency to 739 Hz. This new design would increase performance by reducing the sled settling time by 3X. Section 3.3 discusses the performance of such an improvement.*

Average service time	1.49 ms (0.25)
Maximum service time	4.51 ms
Average seek time	1.27 ms (0.19)
Maximum seek time	1.66 ms
Average X seek time	1.24 ms (0.21)
Maximum X seek time	1.66 ms
Average Y seek time	0.90 ms (0.31)
Maximum Y seek time	1.62 ms
Settling time	0.72 ms
Average per-request turnaround time	0.20 ms (0.20)
Maximum per-request turnaround time	1.34 ms

Table 2: ***Performance characteristics of the MEMS-based storage device model.*** *These numbers are based on a random workload of 10,000 requests; the random workload is described in Section 3.2. Standard deviations are provided in parentheses.*

damping has the effect of reducing $number_{timeconstants}$ and therefore $time_{settle}$.

Extra turnaround times for $time_{seek\_y}$. Y-dimension seeks, for which the final velocity is the access velocity rather than zero, are not expected to require extra settling time. However, since the media sled may be moving in the wrong direction before the seek and/or after the seek, it may be necessary to reverse the sled's direction once or twice. For each such turnaround:

$$time_{turnaround} = 2 * \frac{v_{access}}{a(x)} \qquad (6)$$

Computing $time_{transfer}$. The $time_{transfer}$ component of the MEMS-based storage device service time differs from that of conventional disks in two ways. First, the time to transfer a single sector is the product of the number of tips over which each sector is striped, the rate at which bits are read ($v_{access} * width_{bit}$), and the percentage of bits read that are actual data (*e.g.*, rather than servo and ECC). Second, the time to transfer a range of sectors must take into account the fact that multiple sectors can be accessed in parallel; the number of sectors accessed in parallel is the number of concurrently active tips divided by the number of tips per sector. As with conventional disks, when a range of sectors to be transfered crosses a track or cylinder boundary, a track or cylinder switch is required. The sequential track switch time is equal to the minimum turnaround time, since switching the active tips is expected to take less than this time. The sequential cylinder switch time can be computed as a single cylinder seek, but optimizations of the control loop can be expected to reduce this time to the minimum turnaround time by taking advantage of the tips' ability to deflect small distances in the X dimension.

3.2 Performance of the Default Model

We built this performance model into the DiskSim device simulator [4], allowing us to evaluate its performance under different workloads and parameter settings. This section explores MEMS-based storage device performance given the parameters listed in Table 1. Based on detailed discussions with engineers designing and building MEMS-based storage

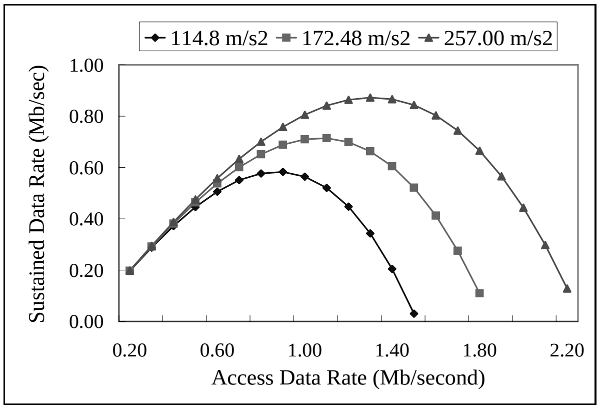


Figure 8: ***Sensitivity of MEMS-based storage device performance to the access velocity.*** *Three actuator acceleration values are shown. The maximal point for each acceleration value represents a balance between the benefit of higher data rates and the increased time required to turn around for track and cylinder switches.*

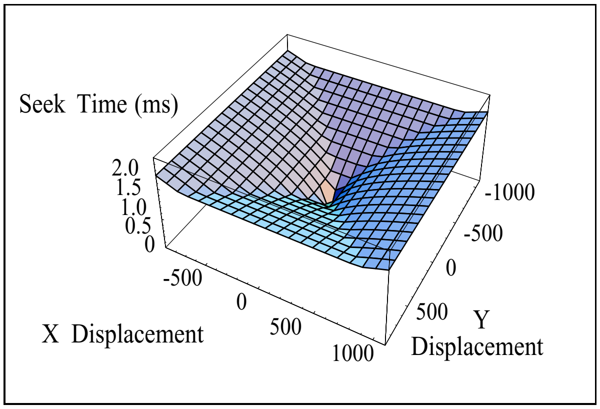


Figure 9: ***Seek times for MEMS-based storage devices when no settling time is required for X-dimension seeks.*** *Without settling time delays, Y-dimension seeks become more a more significant component of overall seek times.*

devices, we believe that these values and equations are reasonable starting points for evaluating these devices.

Table 2 summarizes performance metrics for the default MEMS-based storage device under a synthetically-generated random workload of 67% read requests, exponential request size distribution with 4 kByte mean, and request locations uniformly distributed across the device capacity. The average service time is dominated by the average seek time, which in turn is often dominated by the X-dimension settling time.

Figure 7 shows the seek time to every point on the sled from a corner (a) and from the center (b). These are calculated directly using the method described in section 3.1, independent of the DiskSim simulator. It is interesting to note that for most seeks shown in 7(a), there is no dependence on the Y-dimension movements, except for the shortest X-dimension seeks. This is due to the fact that seek time in the X-dimension almost always dominates seek time in the Y-dimension because of the extra settling time. In 7(b), we again see the dominance of the X-dimension seek time, resulting in an independence of Y-position. This effect is discussed further below.

As with conventional disks, seek delays for MEMS-based storage devices depend on the relative locations and motions of the movable components and the desired data. Therefore, appropriate request scheduling [2] can be expected to reduce positioning delays. Having cast MEMS-based storage devices into a disk-like model, our early results [7] indicate that most of the algorithms and insights from previous disk scheduling research (*e.g.*, [2; 5; 6; 9; 19; 23]) will also be relevant to systems with MEMS-based storage devices.

3.3 Sensitivity to Model Parameters

To understand which device characteristics are important to performance, we have explored the model's performance sensitivity to the different parameters. This section describes the most interesting results.

Sensitivity to per-tip data rate. Overall bandwidth to and from the media is determined by the number of simultaneously active tips and the per-tip data rate. Like conventional disks, MEMS-based storage devices must switch tracks (or cylinders) when media transfers cross track boundaries. Unlike conventional disks, for which rotation speed is independent of seek arm positioning, the time required for MEMS-based storage devices to switch tracks depends directly upon the access velocity (Equation 6). Specifically, because of their Cartesian nature, MEMS-based storage devices turn around each time a media transfer crosses a track boundary. Reversing direction requires decelerating, changing direction, and re-accelerating to the access velocity. As the access velocity increases, this turnaround time increases. Therefore, one should expect diminishing returns from increasing per-tip data rate while keeping other parameters constant. Figure 8 shows the sustained bandwidth of a single tip given increasing per-tip data rates and three different values of actuator acceleration. For each actuator acceleration, there is a maximum data rate after which turnaround times dominate transfer rates. This is an important result, because it indicates that the recording head and channel need not handle ever-higher data rates, making them simpler to manufacture and less power-hungry. Further, this result suggests that efforts may be better spent on improvement of other design characteristics; in fact, showing this result to the Carnegie Mellon MEMS-based storage researchers has triggered exactly this change in their plans.

Sensitivity to settling time. Whenever the sled moves in the X-dimension, some time is required to damp the sled's oscillations, as described above. This settling time is based on the system's resonant frequency and the ability of the control system to damp out the motion. We model this by computing a settling time constant (Equation 5). The number of settling time constants added can be varied to allow for improved control systems. The default model described in Table 2 adds one time constant of 0.72 ms. In order to see the effect of the settling constant, we ran the same experiment as shown in Figure 7(b) without the settling time in X. Rather than uniformly decreasing seek times by 0.72 ms, as one might expect, the result is as shown in Figure 9. Without settling time delays for X-dimension seeks, overall

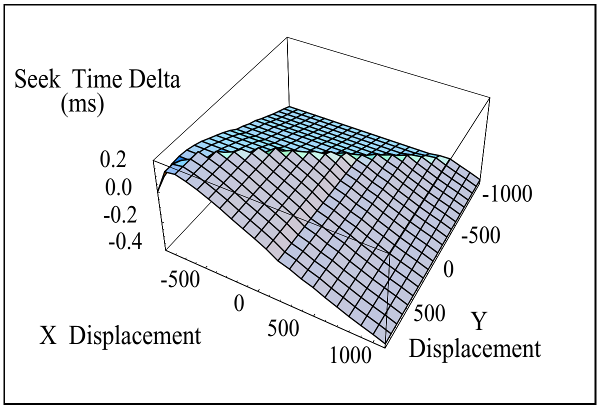


Figure 10: ***Delta in seek times from <-1000,1000> given a spring factor of 75% (compared to 0%).*** *Short seeks are made slightly longer and long seeks are shorter.*

seek times are much more dependent on Y-dimension seeks, making the seek profile match better with expectations for a two-dimensional movement.

Sensitivity to spring forces. The effect of springs on seek time is shown in Figure 10. This graph shows the same set of seeks as Figure 7(a), but in this case we only see the differences in seek times caused by the spring forces. The net effect of adding the spring forces is to lengthen the time for short seeks and to shorten the time for long seeks. The intuition behind this result is fairly straightforward. Consider a *spring_factor* value of 50%, meaning that the springs push back with 50% of the actuator force when the sled is at full displacement. If the actuators are pulling the sled towards the center, then the net force on the sled is 150% of the actuator force. If the actuators are pulling against the springs, then the net force is only 50% of the actuator force. Thus, at a given displacement, the impact of the springs is greater when they hurt than when they help. During a short seek, the displacement remains relatively constant throughout the seek, and so the springs will hurt one phase of the seek more than it helps the other. During long seeks, the displacement changes significantly. As a result, the springs tend to help noticeably in one of the two phases and be either less significant or also helpful in the other. Therefore, long seeks are generally helped by the springs.

The springs' effects on turnaround times are similar to those for short seeks. Figure 11 shows turnaround times with and without springs for each displacement, assuming that the sled is moving at the constant access velocity in the positive direction. Superimposed on the graph is the constant turnaround time that results from a spring factor of 0%. In the left half of the graph, the springs act against the actuators during the turnaround. In the right half, they help. As with short seeks, the impact of the springs is more significant when they hurt than when they help.

Spring forces could have some interesting effects on both the layout of data and the scheduling of requests. While the springs make seek times more uniform, reducing the importance of these techniques, their effect on turnaround times can be much more detrimental. Therefore, avoiding the most costly turnarounds could be important to performance.

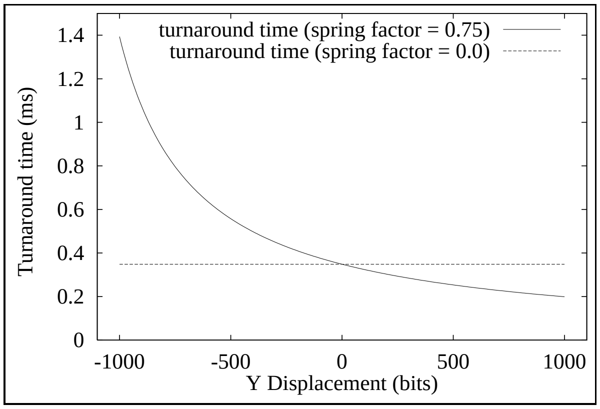


Figure 11: ***The effect of springs on turnaround time.*** *This figure shows the turnaround time at each displacement from center given that the sled is moving at the access velocity in the positive direction. Therefore, the springs hurt the turnaround time for the negative displacements and help in the positive.*

4. APPLICATION PERFORMANCE

This section presents a brief end-to-end performance comparison of systems using conventional disk drives with systems using MEMS-based storage devices. These comparisons make use of application benchmarks running over simulated hardware. A more extensive comparison of these models is available in [17].

4.1 Simulation Environment

In order to study the end-to-end performance effects of integrating MEMS-based storage with modern computer systems, we combine our DiskSim-based device simulator with the SimOS machine simulator. SimOS is a complete hardware simulator capable of booting operating systems and providing statistical analyses of real-world applications running in the SimOS environment [14]. We chose the *SimOS-Alpha* port developed at Compaq Western Research Laboratory; this version simulates an Alpha 21164-based system with 128 MB of primary storage running the Digital UNIX 4.0 operating system. To better approximate systems in which MEMS-based storage will be integrated, we scale the SimOS-Alpha processor clock to 1.0 GHz.

The DiskSim simulator allows us to directly compare the performance of the MEMS-based storage device model with existing and future magnetic disks. For comparisons with existing disks we use a validated model [16] of the Quantum Atlas 10K TM09100W [13]. To compare MEMS-based storage devices with future disks, we simulate a "Superdisk" model based on an aggressive extrapolation of current disk trends to the year 2005. The superdisk streams data at 125 MB/s, has an average seek time of 3 ms, and rotates at 20,000 RPM. Figure 12 compares the average access times of these three devices.

4.2 Application Performance

We present the results of two application benchmarks, PostMark and TPC-D running over Postgres. PostMark [10] is a filesystem benchmark consisting of a series of file operations (create, delete, read, write) on small files. PostMark is meant to be representative of file activity in the Internet

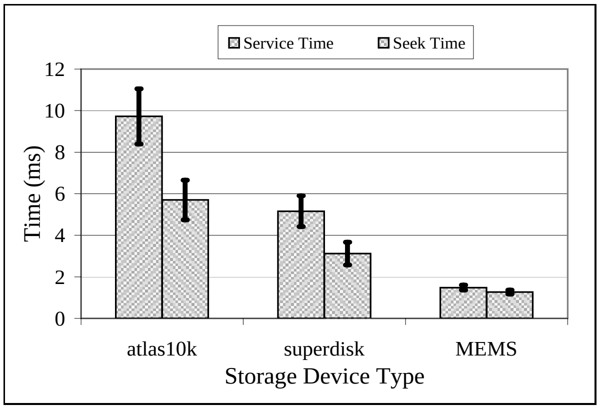


Figure 12: ***Average access and seek times for the random workload.*** *The error bars show the standard deviations.*

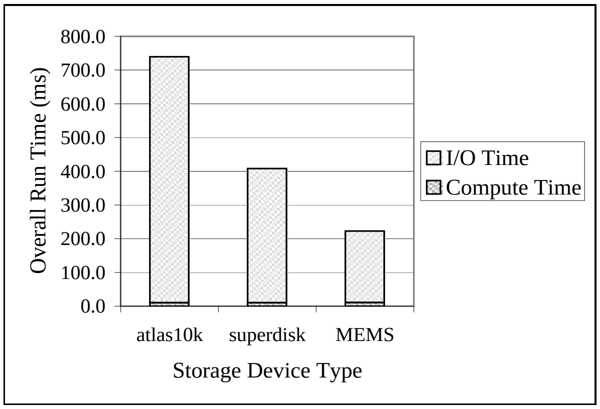


Figure 13: ***Runtime of the PostMark benchmark.*** *Each overall runtime is broken into I/O stall time and compute time. As expected, faster devices reduce I/O stall times.*

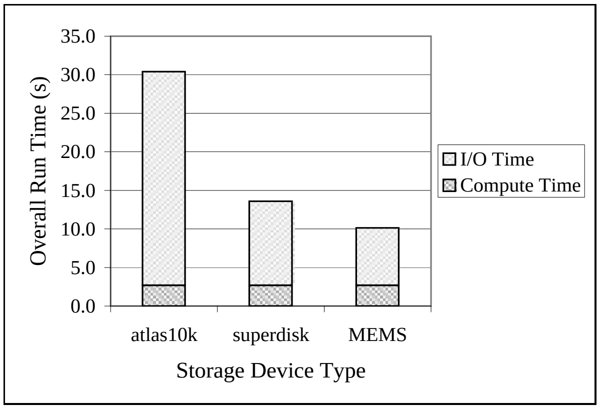


Figure 14: ***Runtime of query #4 from the TPC-D benchmark suite.*** *Each overall runtime is broken into I/O stall time and compute time. As expected, faster devices reduce I/O stall times.*

environment—*e.g.*, electronic mail servers, newsgroup access and storage, and web-based commerce applications. TPC-D [20] is a large-scale database benchmark. TPC-D exercises complex, long-running decision support queries against large complex data structures. We report the performance results of a subset (query #4) of the full TPC-D benchmark.

Figure 13 shows the relative performance of the PostMark benchmark running on the three storage devices. With the MEMS-based storage device, the benchmark completes three times faster than with the baseline disk and almost twice as fast as with the Superdisk. This can be attributed to the much faster positioning times. In the case of the Atlas disk, the average access time for this benchmark was 5.49 ms and for the MEMS-based storage device it was 1.04 ms. PostMark is largely characterized by many small accesses, mostly to filesystem metadata. Specifically, there were almost 150,000 requests averaging 15 sectors. The shorter seek times of the MEMS device are especially beneficial for this type of access pattern. Furthermore, when the workload repeatedly writes the same blocks, as is often the case with metadata updates, disks suffer large rotational penalties whereas MEMS-based storage devices can simply turn around.

The results for the TPC-D query, shown in Figure 14, show a similar speedup for the MEMS-based storage device. These results are particularly impressive when considering that the MEMS model lacks an on-board prefetching cache (the on-board cache hit rate is almost 84% for both the Atlas and the Superdisk).

5. CONCLUSIONS

This paper develops a performance model for MEMS-based storage devices and uses it to evaluate their performance. The results of this study provide both MEMS researchers and computer system researchers with a significant glimpse into the potential performance wins and design tradeoffs of MEMS-based storage. Overall, these devices provide much lower average service times (*e.g.*, 1–2 ms) than conventional disks for locality-free workloads; this results in a 3X overall performance improvement for I/O-intensive applications in our experiments.

Continuing this work, we are exploring: (1) how to best structure MEMS-based storage devices given the complex interactions between physical parameters; (2) how to appropriately configure file system and OS structures to manage such devices; and (3) how to use MEMS-based storage in a wide range of current and future applications, such as data mining, speech recognition, and portable computing.

We are also exploring the characteristics of other emerging nonvolatile storage technologies such as Magnetoresistive RAM (MRAM) and Ferroelectric RAM (FeRAM). Like MEMS-based storage, these technologies should fit into the memory hierarchy between DRAM/SRAM and disk drives. Unlike MEMS-based storage, these technologies involve no mechanical components and so are expected to have lower access times. Also unlike MEMS-based storage, however, their storage densities are constrained by the limits of photolithography rather than the limits of magnetic recording. Thus, MRAM and FeRAM technologies are unlikely to approach the storage capacities of MEMS-based storage devices.

ACKNOWLEDGEMENTS

We thank Rick Carley and the other members of CMU's Center for Highly Integrated Information Processing and Storage Systems [1] for the numerous discussions that helped to refine the performance models. We also thank the members of the Parallel Data Consortium for their interest, insights, and support.

This material is based in part upon work supported under a National Science Foundation Graduate Fellowship.

REFERENCES

[1] Center for Highly Integrated Information Processing and Storage Systems (CHI^{2}PS^2) home page. http://www.ece.cmu.edu/research/chips/.

[2] P. Denning. Effects Of Scheduling On File Memory Operations. In *IFIPS Spring Joint Computer Conference*, pages 9–21, Apr. 1967.

[3] 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.

[4] 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, Feb. 1998.

[5] R. Geist and S. Daniel. A Continuum of Disk Scheduling Algorithms. *ACM Transactions on Computer Systems*, pages 77–92, Feb. 1987.

[6] R. Geist, R. Reynolds, and E. Pittard. Disk Scheduling in System V. In *ACM SIGMETRICS Conference*, pages 59–68, May 1987.

[7] J. L. Griffin, S. W. Schlosser, G. R. Ganger, and D. F. Nagle. Modeling and Scheduling of MEMS-Based Storage Devices. Technical Report CMU-CS-00-100, Carnegie Mellon Univeristy School of Computer Science, Nov. 1999.

[8] D. Hunter. Modeling Real DASD Configurations. Technical Report Research Report RC8608, IBM, Sept. 1997.

[9] D. Jacobson and J. Wilkes. Disk Scheduling Algorithms Based on Rotational Position. Technical Report HPL-CSP-91-7, Hewlett-Packard Laboratories, Feb. 1991.

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

[11] L. R. Carley, J. A. Bain, G. K. Fedder, et al. Single Chip Computers With MEMS-Based Magnetic Memory. In *44th Annual Conference on Magnetism and Magnetic Materials*, November 1999.

[12] M. Madou. *Fundamentals of Microfabrication*. CRC Press, Boca Raton, Fla., 1997. ISBN 0-8493-9451-1.

[13] 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.

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

[15] C. Ruemmler and J. Wilkes. An Introduction to Disk Drive Modeling. *IEEE Computer*, 27(3):17–28, Mar. 1994.

[16] J. Schindler and G. Ganger. Automated Disk Drive Characterization. Technical Report CMU-CS-99-176, Carnegie Mellon University School of Computer Science, Nov. 1999. An extended abstract appears in *Proceedings of ACM SIGMETRICS 2000*.

[17] S. W. Schlosser, J. L. Griffin, D. F. Nagle, and G. R. Ganger. Filling the Memory Access Gap: A Case for On-Chip Magnetic Storage. Technical Report CMU-CS-99-174, Carnegie Mellon Univeristy School of Computer Science, Nov. 1999.

[18] P. Seaman, R. Lind, and T. Wilson. On Teleprocessing System Design, Part IV: An Analysis Of Auxilliary Storage Activity. *IBM Systems Journal*, 5(3):158–170, 1966.

[19] T. Teorey and T. Pinkerton. A Comparative Analysis of Disk Scheduling Policies. *Communications of the ACM*, 15(3):177–184, Mar. 1972.

[20] Transaction Processing Performance Council, San Jose, California. *TPC Benchmark D (Decision Support) Standard Specification*, 2.1 edition, Feb. 1998. http://www.tpc.org/dspec.html.

[21] N. Wilhelm. A General Model for the Performance of Disk Systems. *Journal of the ACM*, 24(1):14–31, Jan. 1977.

[22] K. Wise. Special Issue on Integrated Sensors, Microactuators and Microsystems (MEMS). *Proceedings of the IEEE*, 86(8):1531–1787, Aug. 1998.

[23] B. Worthington, G. Ganger, and Y. Patt. Scheduling Algorithms for Modern Disk Drives. In *ACM SIGMETRICS Conference*, pages 241–251, May 1994.