Workload-Based Configuration of MEMS-Based Storage Devices for Mobile Systems

Mohammed G. Khatib
University of Twente
Enschede, NL
m.g.khatib@utwente.nl

Ethan L. Miller
University of California
Santa Cruz, CA, USA
elm@cs.ucsc.edu

Pieter H. Hartel
University of Twente
Enschede, NL
p.h.hartel@utwente.nl

ABSTRACT

Because of its small form factor, high capacity, and expected low cost, MEMS-based storage is a suitable storage technology for mobile systems. However, flash memory may outperform MEMS-based storage in terms of performance, and energy-efficiency. The problem is that MEMS-based storage devices have a large number (i.e., thousands) of heads, and to deliver peak performance, all heads must be deployed simultaneously to access each single sector. Since these devices are mechanical and thus some housekeeping information is needed for each head, this results in a huge capacity loss and increases the energy consumption of MEMS-based storage with respect to flash.

We solve this problem by proposing new techniques to lay out data in MEMS-based storage devices. Data layouts represent optimizations in a design space spanned by three parameters: the number of active heads, sector parallelism, and sector size. We explore this design space and show that by exploiting knowledge of the expected workload, MEMS-based devices can employ all heads, thus delivering peak performance, while decreasing the energy consumption and compromising only a little on the capacity. Our exploration shows that MEMS-based storage is competitive with flash in most cases, and outperforms flash in a few cases.

Categories and Subject Descriptors

D.4.2 [**Operating Systems**]: Storage Management—*Secondary storage*

General Terms

Design, Experimentation

Keywords

Data layout, MEMS, Probe-Based Storage

1. INTRODUCTION

Users of battery-powered mobile systems require increasingly large storage capacities to store large amounts of digital content. However, the storage device in a mobile system must satisfy a stringent set of requirements: (1) exhibit a short response time, (2) consume little energy, and (3) have low cost per gigabyte. Disk drives, for example, cost 0.5 $/GB but consume too much energy for a small mobile system. Therefore, disks are used mostly in laptop computers. By contrast, flash is energy efficient, but flash costs about 6.0 $/GB and is thus mainly used in small mobile systems. An ideal storage device should be as performance- and energy-efficient as flash and as cheap as disk.

One storage technology that has the potential to satisfy these three requirements is storage based on Micro Electromechanical Systems (MEMS) [1, 2]. Enabled by storage densities above 1 Tb/in^2, MEMS technology promises to deliver a large capacity, a small form-factor, and a lower cost than flash. However, using MEMS-based storage devices in the right way to live up to these promises presents a challenge. MEMS-based storage devices have a large number of heads, and to deliver peak performance, all heads must be deployed simultaneously to access each single sector. Because MEMS-based storage devices are mechanical, some housekeeping information (such as control data) must be stored for each head to access the data. Maintaining this housekeeping information may result in a significant capacity loss, as well an increase in energy consumption.

To solve the problem, we propose an approach to lay out data in MEMS-based storage devices. In this approach, we have the usual block address mapping, as in disk drives, but we also optimize over three data layout design parameters: the number of active heads, the sector parallelism, and the sector size. The objectives of the optimization are: short response time, low energy, and high capacity. We propose models to aid the designer in exploring the design space, and to choose suitable configurations of the data-layout parameters. We show that using knowledge of the expected workload to guide the optimizations leads to points in the design space where MEMS-based storage is comparable to flash in terms of performance and energy, while compromising only a little on the capacity. We also show that the data-layout design space has only Pareto optimal design points.

Simulations against real flash block traces from mobile systems show that (1) in the best case, a single-chip MEMS-based storage device exhibits 5% shorter response time than a (multi-chip) Compact Flash card at 9% more energy consumption, and (2) in the worst case a MEMS-based storage device has up to 25% longer response time and consumes up to 19% more energy. Since MEMS-based storage promises high densities, we estimate that in both cases the expected

EMSOFT'08, October 19–24, 2008, Atlanta, Georgia, USA.

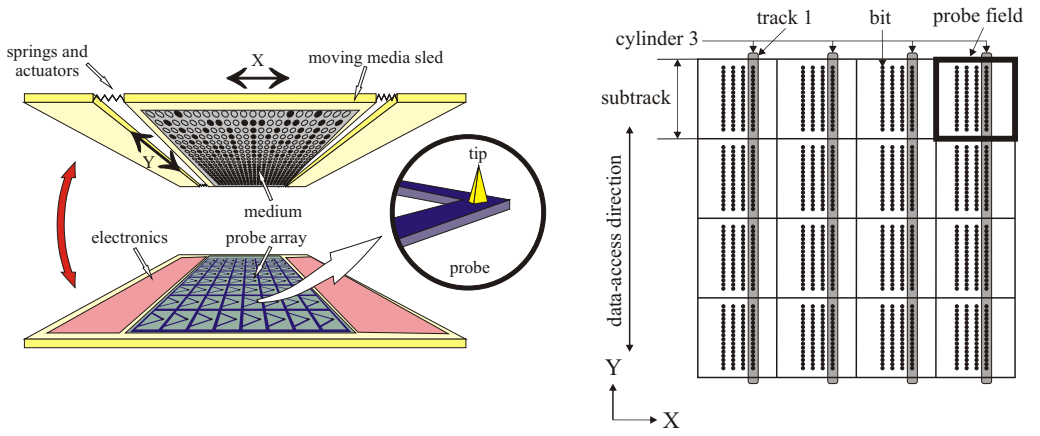


Figure 1: Three- and two-dimensional views of a MEMS-based storage device. ***(a) Two layers facing each other where the media sled is attached to springs that suspend it across the probe array. (b) The storage area of a simplified MEMS-based storage device consisting of*** 4×4 ***probes. The storage area is logically divided into 16 storage fields each accessible by a single probe.***

price per gigabyte of MEMS-based storage is significantly lower than that of flash.

The remainder of this paper is organized as follows. First, we briefly introduce MEMS-based storage and the challenge posed by its architecture on the three requirements. Section 3 details the design targets and our three data-layout parameters. Section 4 presents our experimental methodology throughout this work. Studies of the influence of each parameter on the targets and their mutual influence follow in Section 5 and Section 6, respectively. Section 7 compares MEMS-based storage devices against flash memories. Section 8 discusses crucial points of improvements for MEMS-based storage devices. Section 9 studies related work and Section 10 concludes.

2. BACKGROUND

Several design models for MEMS-based storage have been proposed [1, 3, 4, 2]. Although these models adopt different storage and actuation techniques, they have a common architecture. A MEMS-based storage device consists of two distinct physical layers, one above the other, as shown in Figure 1a. The top layer, called the *media sled*, is suspended by springs across the bottom layer, where the Z distance is maintained by nanopositioners. The bottom layer is a two-dimensional array of read/write probes or heads, called the *probe array*. For example, the IBM MEMS device [1] has a 64×64 probe array. Probes can be clustered in groups to reduce the complexity of the circuitry.

The top layer is the media sled on which bits are recorded. Bits can be recorded on a magnetic patterned medium as in μSPAM [4] and the CMU MEMStore [3]; a polymer medium as in the IBM MEMS device [1]; or a phase-change medium as in the Nanochip MEMS device [2]. The sled moves independently in the X, Y, and Z directions relative to the probe array. In all design models, each probe sweeps over a bounded area of the media sled, called the *probe (storage) field* as sketched in Figure 1b. Consequently, seek times shorten and a relatively high (aggregate) data rate is attainable by operating many probes simultaneously, so that each probe accesses a small part of a sector, called a *subsector*.

The media sled and the probe array in MEMS-based storage devices are separated by a distance of few nanometers, which is maintained actively by nanopositioners. Depending on the recording technique, the probes may make contact with the medium to read or write data. If the medium is magnetic, then data are read or written without contact, like in disk drives. Conversely, in the IBM MEMS device, the probes make contact with the medium only at read (or write) to create (or sense) depressions in the medium. No friction exists during seeks and during motion from one bit to another, because the probes touch the medium only on demand and repel after that (i.e., no steady contact). To access data on the medium, the media sled moves along the Y direction, along which data tracks lie as shown in Figure 1b. While accessing data, the X actuators keep the sled still along the X direction on the accessed data track, counteracting the spring restoring force. When resting, the springs hold the sled at its resting position, where every probe faces the center position of its probe field.

As of early 2008, an IBM prototype [1] can record a single bit in an area of 26 nm by 26 nm, whereas Nanochip [2] claims a 15 nm by 15 nm bit cell area with the potential to reach a scale of 2 nm by 2 nm. With such high densities, a single memory chip has a capacity of 1 TB per die. These devices have potentially low cost for three reasons. Firstly, they can be manufactured using the well-established batch MEMS fabrication technology [1]. Secondly, these devices can be manufactured using micron-scale fabrication plants, whose equipment were installed ten years ago and have passed their break-even point, avoiding the need to build dedicated fabrication plants, unlike for flash memory. Thirdly, these plants can be used to make future generations of MEMS, since MEMS poses no requirements on the lithography process when increasing the density [2].

2.1 A challenge

MEMS-based storage devices use a large number of probes

in parallel to access one sector at a time, thereby decreasing the response time and the energy consumption. Because these devices are mechanical, they must separate physical subsectors by gaps and embed a flush pad in each subsector to enable the data channel and the control mechanism to access the stored data. As a result, storing one data bit per probe per sector (in the case of striping a 4096-bit sector across 4096 probes), as in flash, decreases the capacity because storing a single data bit in a sector requires three bits of per-probe information. Thus, to reduce the capacity loss, the subsector size has to be larger than the overhead bits. To enlarge the subsector size, we (1) reduce the number of probes per sector and increase the number of simultaneously accessible sectors (*sector parallelism*), and (2) enlarge the *sector size*. Our research shows that enlarging the sector parallelism and sector size based on the expected workload maximizes the performance and minimizes the energy consumption of MEMS-based storage devices, while retaining most of the capacity.

3. DATA LAYOUT

Data layout is concerned with the way user data are organized on the storage medium of the device. Data layout influences the response time, the energy consumption and the capacity of the storage device. For example, placing related data sectors contiguously on the physical medium avoids seeks between the sectors, which results in short response time and low energy to access data.

3.1 Three data-layout parameters

The attainable data rate per probe in a MEMS-based storage device is limited by several factors including the probe resonance frequency. The per-probe data rate is 40 Kbps in the present IBM prototype [1], suggesting that systems requiring even moderate transfer rates must use many probes. As a consequence, the data-layout design space widens beyond just block mapping, posing three questions that must be answered to maximize performance and minimize energy usage without compromising performance, namely:

1. **Total number of active probes (N):** how many probes should operate (i.e., be active) simultaneously?
2. **Sector parallelism (M):** how many sectors should be simultaneously accessible from the device?
3. **Sector size (S_{sector}):** should the conventional sector size of 512 bytes stay the same in MEMS-based storage devices?

The straightforward answers to these questions would be to (1) operate all probes simultaneously to gain peak throughput, (2) access one sector at a time to utilize the bandwidth fully, and (3) keep the sector size intact to access useful data only.

While these answers are logical, our research shows that none of the three targets of MEMS-based storage devices reaches optimality with a such configuration. Before studying the influence of each parameter on the design targets, we detail the influence of the number of probes on the capacity.

3.2 Physical-subsector size

A storage device stores user data in physical sectors. In addition to the user data, a physical sector contains error-correction (ECC) data. All types of storage devices have to store ECC data to increase the reliability of the stored user data. The amount of ECC data depends on, among others, the sector size and the type of errors the device is prone to. We call the portion of user data of a physical sector, a logical sector. In disk drives, the ECC constitutes one-tenth the size of the logical sector [5]. We assume that the size of the ECC overhead is one-eighth the size of the logical sector (S_{sector}):

$$S_{\text{ECC}} = \left\lceil \frac{S_{\text{sector}}}{8} \right\rceil$$

Mechanical storage devices exhibit a physical overhead in order to address and access the user data. This physical overhead is a few bits that separate every two contiguous subsectors. The separation bits (1) allow for buffering data before writing a subsector, and (2) keep the clock of the read channel running, so that the subsector can be fully read/written. Jacob et al. [6] provide an anatomy of the physical sector in disk drives. The physical overhead in disk drives has a small influence on the capacity, because it occurs once per sector. Conversely, in MEMS-based storage devices the physical overhead has to occur every subsector, because every probe accesses a subsector due to striping. We assume that the total physical overhead per subsector is 3 bits.

From above, striping a physical sector across K probes results in a physical subsector of size ($S_{\text{p-subsector}}$):

$$S_{\text{p-subsector}} = \left\lceil \frac{S_{\text{sector}} + S_{\text{ECC}}}{K} \right\rceil + 3\,. \qquad (1)$$

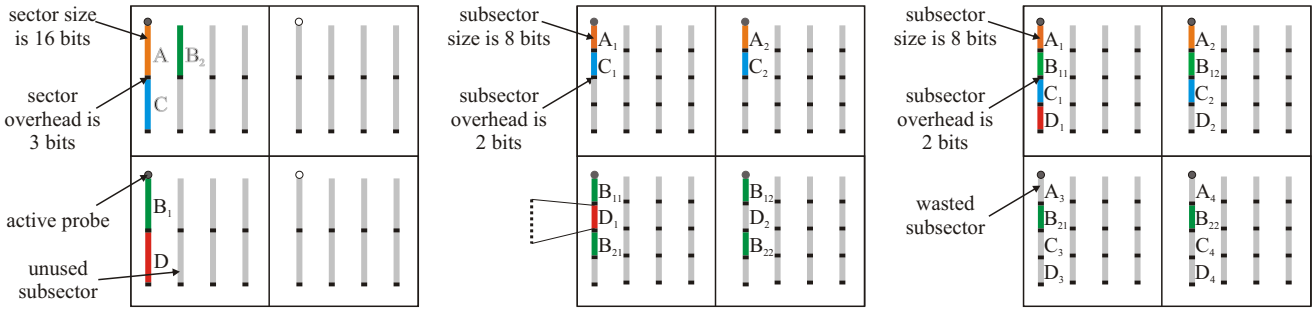
To avoid very small capacities, we assume that the minimum physical-subsector size is 8 bits. To avoid seeks within an access to a subsector, the maximum physical-subsector size is smaller than the subtrack size $\frac{\text{field length}}{\text{bit length}} = \frac{100000}{40} = 2500$ bits (Figure 1b).

Table 1: Settings of the model of the simulated MEMS-based storage device

total # of probes	64×64	probes
bit/track pitch	40	nm
probe field area	100×100	μm^2
per-probe data rate	40	Kbps
seek power	120	mW
bit access power	0.25	mW
max actuation power	120	mW
inactive power	5	mW
shutdown time-out	1	ms
# of active probes	64,128,256,512 1024,2048,4096	probes
sector parallelism	1,2,4,8,16	sector
(logical) sector size	0.5,1,2,4,8	KB

4. EXPERIMENTAL METHODOLOGY

IBM demonstrated a MEMS-based storage device of 64×64 probes. Although their prototype is not available for experiments, sufficient specification data are available in the literature [1]. We use trace-driven simulations to study the data-layout design space of MEMS-based storage devices. We use the DiskSim simulator [7]; a validated modular simulator for simulating various types and architectures of storage subsystems. We refine the energy model of the seek operation in the CMU MEMS model [8] to account for non-constant power dissipation across the medium. Also, we



(a) 2 probes, 2 sector, 16 bits (b) 4 probes, 2 sectors, 16 bits (c) 4 probes, 1 sector, 32 bits

Figure 2: Three possible configurations of the three parameters (# active probes, sector parallelism, sector size) of a simplified MEMS-based storage device. ***The first configuration (a) deploys 2 out of 4 probes simultaneously, each accessing a 16-bit sector at a time. By deploying twice as many active probes, (b) a probe accesses only half a 16-bit sector, so that 4 probes access two sectors at a time, or (c) a probe accesses a quarter of a 32-bit sector, so that 4 probes access one sector at a time. Increasing the sector parallelism (as in b) causes external fragmentation and thus seeks like from $B_{11}||B_{12}$ ("||" means in parallel) to $B_{21}||B_{22}$, whereas increasing sector size (as in c) causes internal fragmentation, wasting capacity such as A_3 and D_3.***

modify the CMU model to include time and energy models for the shutdown operation [9]. The models use the bang-bang optimal control model, which captures the dynamics of the system and factors in all forces during the sled motion [10, 11]. Further, all parameters including, the bit dimensions and the per-probe data rate of the model are set to those of the IBM MEMS device [1]. To reduce the idle energy, we deploy a fixed-timeout power management policy that shuts down the sled, if no requests arrives within 1 ms after the completion of the previous request. Table 1 summarizes the key settings.

Since our design is targeted at mobile applications, we gathered traces on an HP iPAQ H2215 PDA. An embedded version of Linux (kernel version 2.6.17) has been ported onto this PDA. Jens Axboe's block trace utility [12] was used to log I/O events, which are forwarded to a host machine, so that the gathered traces were not contaminated by the operations needed to store trace records locally. The CF card functioned as the main storage device on which the root file system was located. As a result, all I/O activities went to and from the CF card. We logged different system and application activities. System activities included booting and starting the Graphical User Interface, whereas application activities include: firing applications, such as the text editor and web browser; taking photos; streaming audio and video from/to the storage device; and creating, copying and deleting files. We also measured the energy consumption of the CF card and recorded it on the host machine for energy comparison.

MEMS-based storage devices are expected to serve as storage devices in future computer systems. They will communicate with the file system layer as flash memories do at present. As a result, the performance and energy consumption of MEMS-based storage devices is influenced by the type of the file system and its block size. To strengthen our simulation study, we traced and simulated with different settings of the I/O subsystem. We captured the aforementioned scenarios on `ext3`, the default Linux file system. We also captured on `ext2`, a non-journaling version of `ext3`. In addition, we formatted each file system with the default maximum block size, 4 KB, and a smaller one, 1 KB.

5. INFLUENCE OF EACH PARAMETER

This section studies the influence of each parameter individually on the three design targets; i.e., response time, energy and capacity. We simulate with the trace captured on the CF card when formatted with the ext3 file system and a block size of 4 KB (the ext3-4K trace).

5.1 Number of active probes

Performance.

A MEMS-based storage device has a large number of read/write probes to enhance performance. Increasing the number of probes a sector is striped across shortens the read/write time, because the subsector size decreases as Equation(1) shows. Figures 2a and 2b show that doubling the number of active probes from 2 to 4 results in smaller subsectors a probe has to access per sector; compare A to $A_1||A_2$ ("||" denotes parallel access). Thus, the time to read/write a striped sector boils down to the time a probe takes to read/write one subsector:

$$t_{\text{RW}} = \frac{S_{\text{p-subsector}}}{r_{\text{probe}}}, \qquad (2)$$

where $S_{\text{p-subsector}}$ is the size of the physical subsector size calculated in Equation (1), and r_{probe} is the data rate per probe.

Simulating against ext3–4K, Figure 3 plots the response time as a function of the number of probes per sector of size 512 bytes. Because the minimum subsector size is 8 bits, the maximum number of probes per sector is 512. The response times are normalized to the response time when deploying 64 probes (83 ms). Figure 3 confirms the significant influence of the number of probes on the response time. As the number of probes doubles, the response time approximately halves.

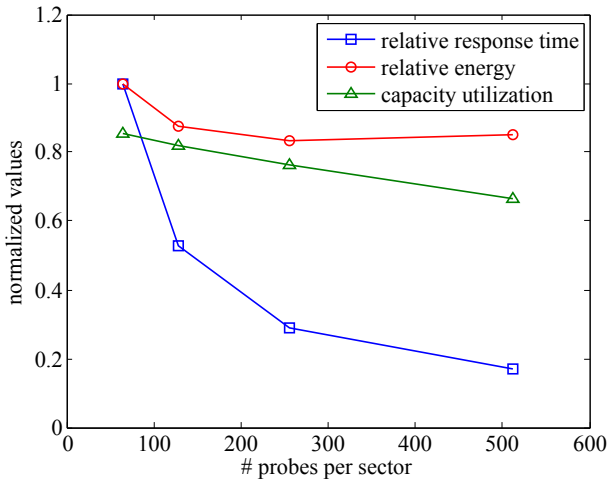


Figure 3: Relative average response time, relative total energy consumption, and capacity utilization of the IBM MEMS device as a function of the number of active probes deployed per sector

Energy.

Unlike the response time, which decreases as the number of probes increases, energy to access data does not decrease because more probes are switched on at the same time, dissipating larger amount of power. Actuation energy, however, decreases. Actuators are powered on to hold the media sled still along the X direction and to move it along Y. Increasing the number of probes decreases the subsector size and thus shortens the time the sled is held on X and the distance it travels along Y (compare Figure 2a to Figure 2b). Consequently, the actuation energy decreases. The total read/write energy per physical sector can be written as follows:

$$\begin{aligned} E_{\text{RW}} &= E_{\text{probes}} + E_{\text{actuation}} \\ &= t_{\text{RW}} \times P_{\text{probe}} \times K + t_{\text{RW}} \times P_{\text{actuation}} \,, \end{aligned} \tag{3}$$

where K is the number of probes per sector, P_{probe} is the power a probe dissipates to read or write one single bit, and $P_{\text{actuation}}$ is the power dissipated by both X and Y actuators. Note that increasing the number of probes has no influence on the actuation energy, since the probes touch the medium during the actual read and write operations. From Equation (3) we can observe that the reduction in read/write energy is bounded by the read/write energy (Amdahl's law). Figure 3 confirms this bound and shows the energy figures normalized to the energy when deploying 64 probes (13.6 J). As the number of probes increases, the actuation energy decreases, as does the total energy. The energy difference between every two successive points decreases as the number of probes increases, since the actuation energy becomes less prominent (Amdahl's law). A minimum point exists at 256 probes, after which energy starts increasing slowly. This increase is due to the additional overhead bits that need to be accessed (Equation (1)), which becomes more noticeable (compare three bits of overhead per sector in Figure 2a to four bits in Figure 2b). Figure 3 also shows that the number of probes has a larger influence on the response time than the energy, since it influences the read/write time more than the read/write energy.

Capacity.

As explained in Section 3.2, several physical bits are written along each subsector to enable its accessibility. As the number of probes increases, the subsector size decreases and the relative overhead per sector increases. As a result, the (effective) capacity of the device decreases. Figure 3 shows the utilization of the physical capacity of the device (about 3 GB). Unlike response time and energy, the values of the capacity are normalized to the raw (physical) capacity of the device. Figure 3 shows a loss of 35% (about 1 GB) in capacity when deploying 512 probes due to the overhead.

Further, Figure 3 shows that the three design targets compete when designing a MEMS-based storage device: a gain in performance results in a loss in capacity. Also, performance gain can compete with energy reduction.

5.2 Sector parallelism

Sector parallelism represents the number of sectors that are simultaneously accessible from the storage medium. It deals with the number of probes a sector is striped across. If a MEMS device has N total active probes that access M sectors simultaneously, the number of probes per sector (K) is:

$$K = \frac{N}{M} \,. \tag{4}$$

Performance.

Increasing the sector parallelism (i.e., M) results in fewer probes per sector (i.e., K). As a consequence, the subsector size increases (Equation (1)). Increasing the sector parallelism has one positive influence and two negative influences on the performance of MEMS-based storage devices. The positive influence is that increasing the subsector size reduces the overhead (Figure 2b versus Figure 2a) and thus decreases the overhead read/write time. On the other hand, one negative influence is that increasing the subsector size results in more data bits that a probe has to access, thus increasing the data read/write time. The second negative influence is under-utilizing the sector parallelism if the size of the requests are not a multiple of the number of simultaneously accessible sectors. If a request demands L sectors from a MEMS device, which is capable of accessing M sectors simultaneously, the response time for the request (t_{request}) is:

$$t_{\text{request}} = \left\lceil \frac{L}{M} \right\rceil \times t_{\text{RW}} + t_{\text{seek}} \,. \tag{5}$$

For example accessing file D in the MEMS device shown in Figure 2b incurs under-utilization of those probes associated with D_2, because it has no useful data. Khatib et al. [13] give a detailed study of the mutual influence between the request size, request address, and the sector parallelism. The study shows that sector parallelism can be tuned based on the characteristics of the expected workload, such as the majority request size, to enhance the performance, and to diminish the two negative influences. Note that in addition to the read/write time, a seek time exists. The seek model is rather more complicated than the read/write time, as detailed by Hong et al. [11].

Figure 4 shows the response times normalized to the response time when sector parallelism is 1 (24 ms). It shows that sector parallelism of 8 exhibits the shortest response time when deploying 256 probes. Setting the sector paral-

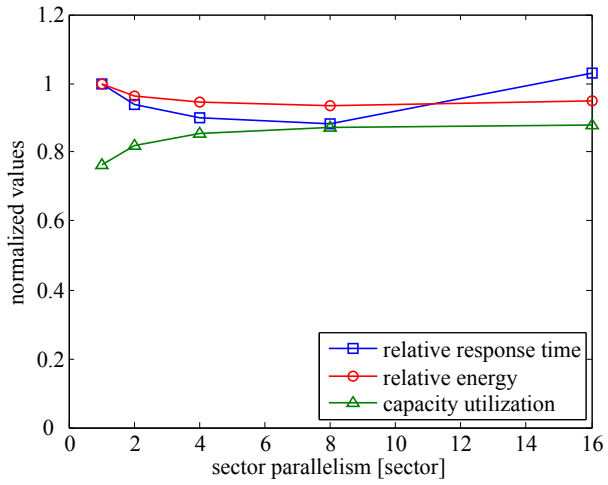


Figure 4: Relative average response time, relative total energy consumption, and capacity utilization of the IBM MEMS device as a function of the sector parallelism

lelism to larger than 8 under-utilizes the active probes and results in large response times.

Energy.

A discussion similar to the performance applies to the energy of MEMS-based storage devices as a function of sector parallelism. The total energy consumed to satisfy a request of L sectors is:

$$E_{\text{request}} = t_{\text{seek}} \times P_{\text{seek}} + \left\lceil \frac{L}{M} \right\rceil \times t_{\text{RW}} \times P_{\text{probe}} \times N + \left\lceil \frac{L}{M} \right\rceil \times t_{\text{RW}} \times P_{\text{actuation}} \,. \quad (6)$$

In addition to the two negative influences on performance, a third negative influence on energy exists. As the parallelism increases, the subsector size increases, which extends the time the medium should be held still along the X direction, and increases the traveled distance along Y. As a consequence, the actuation energy increases. Nonetheless, tuning the sector parallelism as done for the performance (in the face of the first two negative influences) and employing a larger number of probes simultaneously (in the face of the third influence) reduce the energy. Figure 4 shows that, indeed, the energy consumption is minimal for sector parallelism of 8. We deploy 256 probes simultaneously to minimize the third influence since it is the minimum in Figure 3. The values are normalized to the energy when sector parallelism is 1 (11.2 J).

Capacity.

Increasing the sector parallelism has a positive influence on the capacity, because the subsector size increases and thus the overhead per sector decreases. Figure 4 shows that the loss in capacity of about 0.3 GB made when employing 256 probes (see Figure 3) can be earned back by formatting with sector parallelism of 8 sectors, yet at a further reduction in energy and enhancement in performance.

5.3 Sector size

Equations (1) and (4) show that increasing the logical-sector size increases the physical subsector size as the sector parallelism does. As a consequence, MEMS devices exhibit the same influences as when increasing the sector parallelism. The main difference between increasing the sector parallelism and increasing the sector size is that the former can under-utilize probes if sectors are not requested, whereas the latter can under-utilize probes if the sector does not fully contain useful user data. Our analysis shows the same trends to those in Figure 4.

5.4 Sector parallelism versus sector size

Sector parallelism and sector size are two seemingly similar solutions in the face of small subsectors when increasing the number of probes. However, they treat the storage space differently, which in turn influences the performance and thus energy. Increasing the sector parallelism increases external fragmentation, since related sectors are not necessarily spatially co-located. For example, accessing sectors B_{11}, B_{12}, B_{21}, and B_{22} shown in Figure 2b can not be done entirely in parallel, causing one more seek and read/write to access $B_{21}||B_{22}$ after $B_{11}||B_{12}$. On the other hand, increasing the sector size increases internal fragmentation, because sectors are not fully utilized, if the file system lacks intelligent placement techniques. For example, A_{21} and A_{22} in Figure 2c are wasted storage space.

External fragmentation increases seek and read/write operations, whereas internal fragmentation increases storage-space underutilization. Nonetheless, sector parallelism and sector size can be tuned based on the workload to enhance performance at yet large capacity.

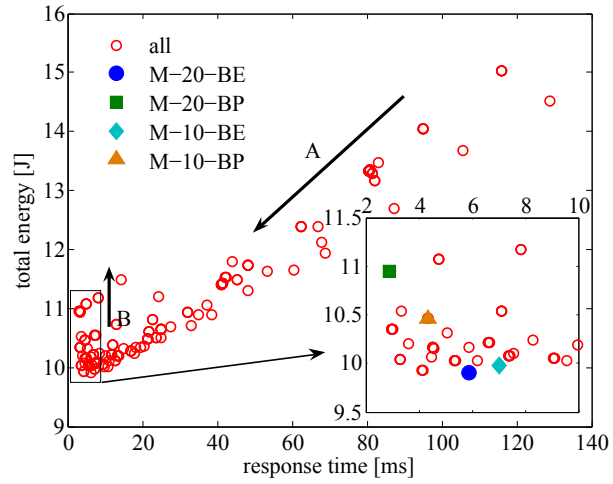
6. DESIGN SPACE

This section studies the design space of the data layout of MEMS-based storage devices composed of all feasible configurations of the three parameters discussed in Section 3.1. As Table 1 shows, we consider seven different settings of the number of probes, five settings of the sector parallelism, and also five settings of the sector size. All settings are a power of two, since the maximum number of probes and the conventional sector size are power of two.

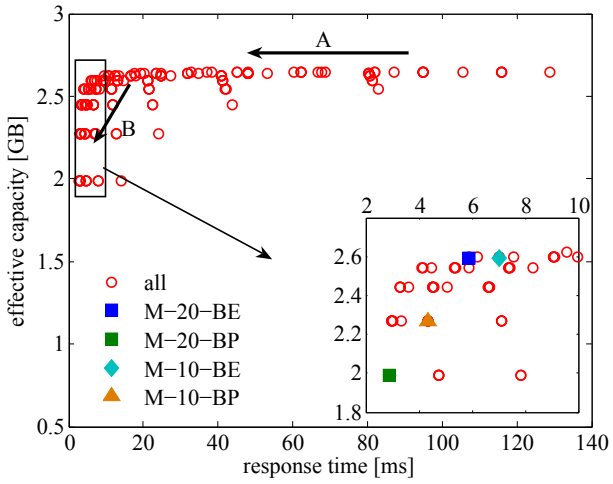
Figure 5 gives three different views of the three dimensional design space, where every configuration of the parameters (i.e., circle in the figures) exhibits a certain response time, energy consumption, and capacity when simulating with the ext3–4K trace. In total there are 175 configurations out of which 20 configurations are infeasible, because they either exhibit a subsector size smaller than 8 bits (the minimum) or larger than 2500 bits (the sub-track size, i.e., the maximum).

Figure 5a plots the response time versus the energy consumption. We can identify two trends: trend A and trend B. Trend A shows that as the number of probes increases, the response time and energy consumption decrease. However, trend B shows that at a certain point the energy consumption increases as the number of probes increases, because the energy to access the overhead bits becomes noticeable.

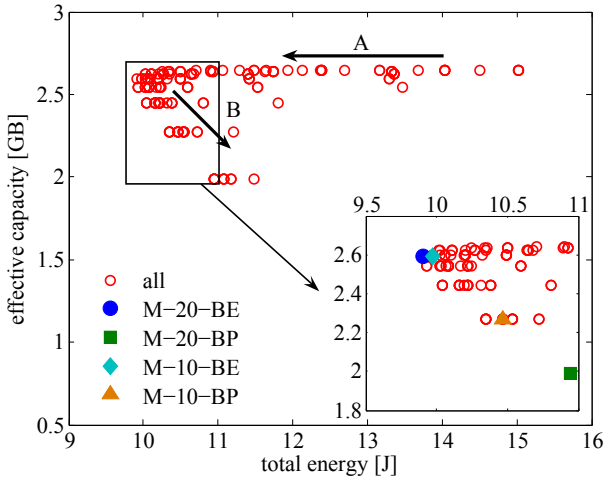
Figure 5b shows the response time versus the effective capacity. Trend A shows that increasing the number of probes reduces the response time while retaining most of the device physical capacity. This trend corresponds to sector parallelism larger than 1 and/or sector size larger than 512 bytes



(a)



(b)



(c)

Figure 5: Trade-offs between the three targets for all 155 feasible configurations when simulating with the ext3–4K trace

as shown in Figure 4. Unlike trend A, trend B shows that a loss in capacity occurs, if the sector parallelism is 1 and/or sector size is 512 bytes as shown in Figure 3. By deploying large sector parallelism and/or sector size, we can retain a large part of the physical capacity at a negligible loss in response time as shown by the points around 2.5 GB.

Figure 5c shows the energy consumption versus the effective capacity. One similar trend to the previous figure can be observed, namely trend A. Trend B shows that a loss in capacity is accompanied by a loss in energy for configurations with large number of probes. The reason is that employing large number of probes simultaneously increases the overhead per sector, causing a loss in energy as well as capacity, unlike trend B in Figure 5b. The reason is that although increasing the overhead increases the response time, a larger decrease in response time occurs by decreasing the number of data bits per probe (see Figure 3), which results in an overall decrease in response time.

Zooming in on the parts where the optima can be found in Figure 5, we find that no optimal solution exists but a set of Pareto optimal points; thus trade-offs are inevitable. We plot the best-energy (M-20-BE and M-10-BE) and best-performance (M-20-BP and M-10-BP) configurations when deploying 4096 and 2048 probes, which we discuss in Section 7.2.

7. COMPARISON AGAINST FLASH

Flash memory is widely used in mobile systems because of its high performance and energy efficiency. Because MEMS devices are expected to be employed in mobile systems, we compare MEMS devices against flash memory.

In this section, we compare several MEMS devices against a SanDisk Standard CompactFlash card [14] from performance and energy perspectives. We choose the CompactFlash form, because it has a superior performance to smaller forms like MMC (Multimedia Card) and SD (Secure Digital) cards. Further, we do not choose very-high–performance cards like CF Extreme-III, because these cards pack more chips at a higher cost, and we use just a single-chip MEMS. In other words, we try to be as fair as possible to MEMS-based storage devices in terms of performance and cost.

7.1 Assumptions

In this work, we enlarge the bit dimensions in MEMS model to 40 nm × 40 nm (compared to 26 nm × 26 nm), so that the formatted MEMS device has a capacity that is approximately equal to that of the flash card: about 2 GB. Doing so, we maintain a fair comparison, since seeks in the MEMS device span the whole address space, thus reporting on its worst-case seek time and energy. Simplifying the power network, we assume that unused active probes cannot be switched off, thus reporting on the maximum read/write energy. That is, if we have 4096 active probes and a request demands data that are accessible by just 2048 probes, the device still consumes an amount of energy corresponding to 4096 active probes.

7.2 MEMS devices

As shown in Section 6, designing a MEMS-based storage device is a multi-objective optimization problem. Because of the (expected) low fabrication cost of MEMS devices, this work assumes that the designer is willing to compromise on the capacity to make MEMS-based storage devices

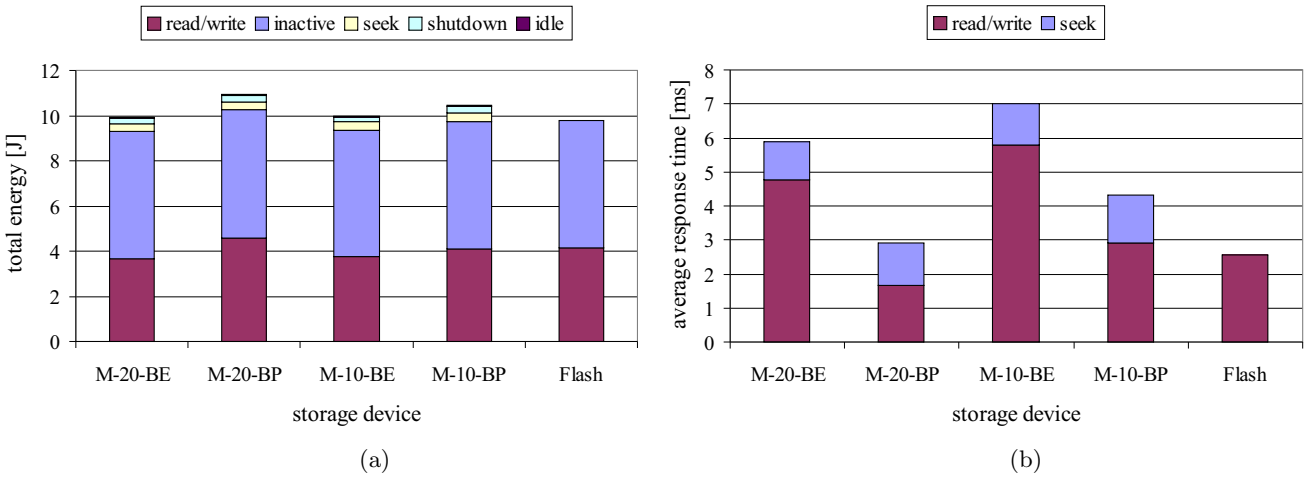


Figure 6: (a) Total energy consumption and (b) average response time of several MEMS devices and the flash card for the ext3–4K trace

as performance- and energy-efficient as flash memory. We, therefore, choose the two overall-best devices performance- and energy-wise. We also show the best-capacity device to evaluate the loss in capacity by the other two devices. That is, we have three MEMS-based storage devices configured with the most energy-, performance-, and capacity-efficient configuration. The configurations are (number of probes: 4096 probes, sector parallelism: 16 sectors, sector size: 4 KB), (4096 probes, 1 sector, 4 KB), and (64 probes, 4 sectors, 8 KB), respectively (see Figure 5).

The best energy and performance devices have a nominal throughput of 20 MB/s, whereas the SanDisk Standard CF card has a minimum read/write throughput of 10 MB/s. This is an advantage for MEMS, since by deploying just one chip a high throughput is achievable, whereas several flash chips in addition to a high-end controller are needed to achieve such throughput. Nevertheless, to enrich our comparison, we additionally choose other energy and performance devices out of the configurations that employ just 2048 probes, which have a nominal throughput of 10 MB/s. Their configurations are (2048 probes, 16 sectors, 2 KB), (2048 probes, 1 sector, 4 KB), respectively.

Next, we discuss the comparison in detail for the ext3–4K trace and then briefly discuss the results for the other traces.

7.3 Results for the ext3–4K trace

Figure 6 shows the energy and response time of the MEMS devices. We exclude the best-capacity device (2.65 GB) since it exhibits a response time of approximately 116 ms, rendering it impractical. The first two devices in Figure 6a correspond to the best configurations that result in the minimum energy consumption and shortest response time, called M-20-BE (best-energy device) and M-20-BP (best-performance device), respectively. The letter M is for MEMS and 20 denotes the nominal throughput $4096 \times 40\,\text{Kb/s} = 20\,\text{MB/s}$. The other two devices are the best energy and performance devices when employing 2048 active probes, called M-10-BE and M-10-BP, respectively. The capacity of the devices in Figure 6a is 2.60 GB, 1.99 GB, 2.60 GB, and 2.27 GB, respectively.

Figure 6a shows that the flash card consumes less energy than the four MEMS devices and outperforms all of them. Yet, the difference in energy consumption between MEMS devices and flash memory is small and lies in the range of 1–11%. The figure shows also the energy breakdown of the four MEMS-based storage devices and the flash card. Like in the flash card, the prominent energy components in all MEMS devices are the read/write and inactive energy. Taking the assumptions into consideration (Section 7.1), MEMS-based storage devices can well be as energy-efficient as flash memories.

Unlike energy, the response time of MEMS devices varies greatly between configurations. The prominent component is the read/write time, which varies from 3 ms to 7 ms. On the other hand, the seek time is in the range of 1.0–1.5 ms. Figure 6b shows that the M-20-BP device exhibits smaller read/write time than the flash card. However, with the seek time added, the total response time becomes longer than the flash memory. The MEMS devices have relatively 13% to 173% longer response time than the flash card.

7.4 Results for the other traces

We also explore the design space and then compare for the other three traces, namely ext3–1K, ext2–4K, and ext2–1K. We choose the overall best-performance MEMS-based storage device for each trace. The configurations of these MEMS devices are (4096 probes, 4 sectors, 1 KB), (4096 probes, 1 sector, 4 KB), and (4096 probes, 4 sectors, 1 KB), respectively. Figure 7a compares the energy consumption of MEMS against flash for all traces. The MEMS devices consume 2% to 19% more energy than the flash card.

Figure 7b confirms our observation for the ext3–4K trace that MEMS-based storage devices exhibit shorter read/write time than flash. However, with the seek time added, the response time of MEMS-based storage devices becomes longer. Unlike for the other traces, for the ext2–4K trace, the corresponding best-performance MEMS device exhibits a 5% shorter response time than the flash card. For the other traces, however, MEMS devices exhibit up to 25% longer response time. Unlike seek energy, seek time influences the response time noticeably, so that the zone-based scheduling technique proposed by Hong et al. [15] can prove worthwhile.

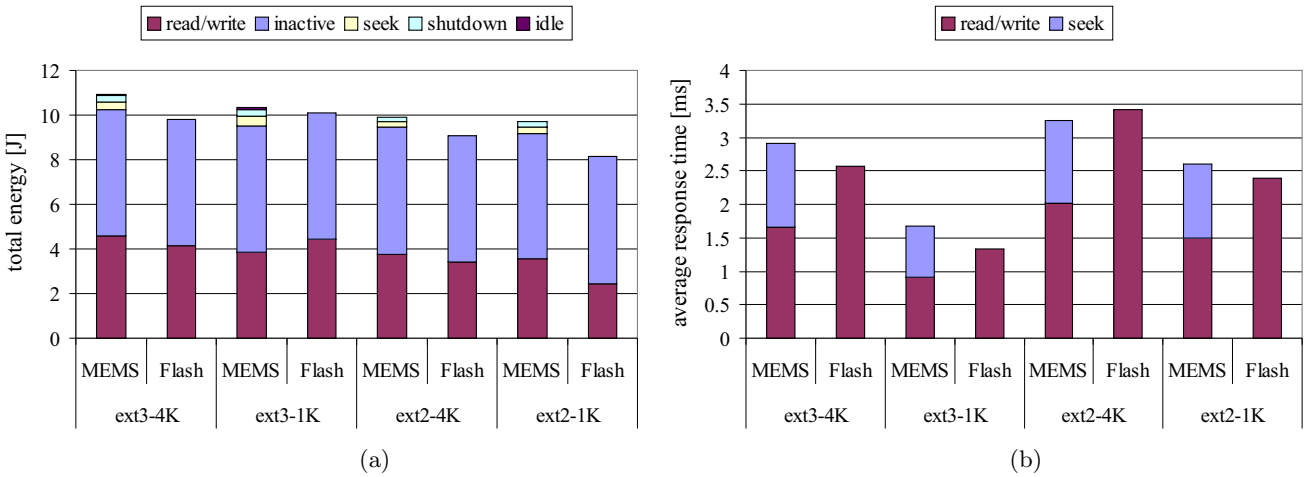


Figure 7: (a) Total energy consumption and (b) average response time of the best-performance MEMS devices and the flash card for four different traces

8. ENHANCING MEMS

This work emphasizes the importance of configuring the data layout of MEMS devices based on the expected workload, so that MEMS devices become competitive with flash memories. Our discussions also reveal the necessity for enhancing MEMS devices on the device level. We have identified the following issues:

per-probe data rate Increasing the attainable data rate per probe shortens the read/write time and energy. Recall that read/write time and energy are the first and second prominent components, respectively. While the read/write time shortens when using more probes per sector (see Figure 3), the read/write energy does not. Therefore, enhancing the inherent performance of the probe is necessary for energy efficiency.

configurable power net Implementing a (coarse) configurable power net certainly further reduces the read/write energy (the second energy component). This is particulary important, because, as our results reveal, MEMS devices can be configured with large sector parallelism for better performance, less energy, and larger capacity.

actuators Section 5 shows that deploying a large number of probes reduces the actuation energy. Targeting flash packages, that have less power budget than CompactFlash like SD (Secure Digital) and MMC (Multimedia Card), limits the number of probes that can be deployed at a time. As a consequence, the actuation energy increases. For such small packages, energy-efficient actuators should be used.

low-power electronics MEMS devices can be shut down aggressively and thus spend a large fraction of time in inactivity. Consequently, as Figure 7 shows, the inactive energy is the most prominent energy component. MEMS devices as well as flash memories can reduce the inactive energy by using lower supply voltage, by applying voltage and frequency scaling techniques, or even by switching off most of the electronics.

9. RELATED WORK

Most of the work on MEMS-based storage devices in the literature focuses on the deployment of MEMS devices as a cache or a replacement for disk drives in server systems [16, 15, 17, 15, 18]. Two earlier papers configure (but do not investigate) the data layout of MEMS-based storage devices [19, 20]. Both keep the sector at the conventional size of 512 bytes. Sivan-Zimet [19] configures the data layout, so that the sector parallelism is 1 sector, where just 320 probes are active at a time. Sivan-Zimet deploys all probes per one sector in order to enhance the throughput, since in her model the per-probe data rate is 1 Mbps. Schlosser [20] configures the data layout of the CMU G2 MEMStore, so that the sector parallelism is 10. In his data layout, Schlosser stripes a sector across 64 probes, where 640 probes are active at a time. CMU G2 MEMStore has a per-probe data rate of 700 Kbps and an 8-byte (ECC and physical) overhead per subsector. Sivan-Zimet and Schlosser simulate against server traces, and target replacing disks with MEMS devices. Although the data rate in CMU MEMStore is lower than that of the MEMS model of Sivan-Zimet, Schlosser stripes a sector across only 64 probes, whereas Sivan-Zimet stripes across 320 probes.

This leads us to investigate the data layout of MEMS devices. Our research makes the case for exploiting the knowledge of the expected workload to configure the data layout of MEMS devices, so that they become competitive with flash performance- and energy-wise. To the best of our knowledge, this is the first work that tailors MEMS devices to mobile systems and compares MEMS devices with flash memory. Setting our MEMS model with recent figures from the IBM prototype [1], we compare MEMS devices with flash memories. To achieve this, (1) we refine CMU MEMS model and update its settings with figures from the IBM MEMS device. (2) We show that the data-layout parameters increase from address mapping, as in disk drives, to encompass three additional parameters: number of probes, sector parallelism, and sector size. (3) We study the individual influence of each parameter as well as their mutual influence on three targets: response time, energy consumption and, capacity of MEMS-based storage devices.

10. CONCLUSIONS

This work enhances the energy efficiency and performance efficiency of MEMS-based storage devices, tailoring them to mobile systems. MEMS-based storage devices have a large number of heads and to deliver peak performance, these devices should deploy all probes to access a single sector at a time. This, however, results in a huge capacity loss, because each head must maintain some housekeeping information.

We propose techniques to configure the data layout in MEMS-based storage devices. The configuration parameters are: number of active probes, sector parallelism, and sector size. We make the case for configuring these parameters based on the expected workload the device will experience when deployed. Simulations against PDA block traces show that, indeed, a workload-based configuration of these parameters makes MEMS-based storage devices competitive with flash performance- and energy-wise.

Simulation results show that MEMS-based storage devices consume up to 19% more energy than flash memory and exhibit up to 25% longer response time at lower price due to the lower fabrication costs. We summarize our study with suggestions to enhance MEMS-based storage devices on the device level, demonstrating the big chance for these devices to become more performance- and energy-efficient.

11. ACKNOWLEDGEMENTS

We wish to thank the Parallel Data Lab at Carnegie Mellon University for providing us with DiskSim. Our gratitude goes also to Johan B.C. Engelen and Leon Abelmann at the University of Twente for providing us with more insight into MEMS-based storage devices. This research is supported by the Technology Foundation STW, applied science division of NWO and the technology programme of the Ministry of Economic Affairs under project number TES.06369 and by the industrial sponsors of the SSRC, including Agami Systems, Data Domain, Hitachi, LSI Logic, NetApp, Seagate Technology, and Symantec.

12. REFERENCES

[1] M. A. Lantz, H. E. Rothuizen, U. Drechsler, W. Häberle, and M. Despont, "A Vibration Resistant Nonopositioner for Mobile Parallel-Probe Storage Applications," *Journal of Microelectromechanical Systems*, vol. 16, pp. 130–139, February 2007.

[2] "Nanochip Inc.." `http://nanochipinc.com/tech.htm`. Accessed in November 2007.

[3] L. R. Carley, J. A. Bain, G. K. Fedder, D. W. Greve, D. F. Guillou, M. S. C. Lu, T. Mukherjee, S. Santhanam, L. Abelmann, and S. Min, "Single-chip computers with microelectromechanical systems-based magnetic memory (invited)," *Journal of Applied Physics*, vol. 87, no. 9 III, pp. 6680–6685, 2000.

[4] L. Abelmann, T. Bolhuis, A. M. Hoexum, G. J. M. Krijnen, and J. C. Lodder, "Large capacity probe recording using storage robots," *IEE Proceedings: Science, Measurement and Technology*, vol. 150, no. 5, pp. 218–221, 2003.

[5] S. McCarthy, M. Leis, and S. Byan, "Larger Disk Blocks or Not?," tech. rep., 2002.

[6] B. Jacob, S. W. Ng, and D. T. Wang, *Memory Systems (Cache, DRAM, Disk)*, ch. 18, pp. 650–652. Morgan Kaufmann, 2008.

[7] H. S. Bucy, G. R. Ganger, and Contributors, "The DiskSim simulation environment version 3.0," reference manual, School of Computer Science, Carnegie Mellon University, January 2003.

[8] J. L. Griffin, S. W. Schlosser, G. R. Ganger, and D. F. Nagle, "Modeling and performance of MEMS-based storage devices," in *Proceedings of ACM SIGMETRICS 2000*, (Santa Clara, California, 17-21 June), pp. 56–65, 2000.

[9] M. G. Khatib, J. B. Engelen, and P. H. Hartel, "Shutdown Policies for MEMS-Based Storage Devices – Analytical Models," Tech. Rep. TR-CTIT-08-03, Jan. 2008.

[10] T. Madhyastha and K. P. Yang, "Physical modeling of probe-based storage," in *Proceedings of the 18th IEEE Symposium on Mass Storage Systems and Technologies*, pp. 207–224, Apr. 2001.

[11] B. Hong and S. A. Brandt, "An analytical solution to a MEMS seek time model," Tech. Rep. UCSC-CRL-02-31, Storage Systems Research Center, University of California, Santa Cruz, Sept. 2002.

[12] "Kernel Trace Systems." `http://elinux.org/Kernel_Trace_Systems`. Accessed in November 2007.

[13] M. G. Khatib, B.-J. van der Zwaag, F. C. van Viegen, and G. J. M. Smit, "Striping policy as a design parameter for MEMS-based storage systems," in *The 2nd International Workshop on Software Support for Portable Storage*, (Seoul, Korea), pp. 25–32, Oct. 2006.

[14] "SanDisk CompactFlash Standard card." `http://www.sandisk.com/OEM/ProductCatalog(1337)-CompactFlash_Memory_Card.aspx`. Accessed in April 2007.

[15] B. Hong, S. A. Brandt, D. D. E. Long, E. L. Miller, and Y. Lin, "Using mems-based storage in computer systems—device modeling and management," *Transactions on Storage*, vol. 2, no. 2, pp. 139–160, 2006.

[16] S. W. Schlosser, J. L. Griffin, D. F. Nagle, and G. R. Ganger, "Designing computer systems with MEMS-based storage," in *Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems*, pp. 1–12, 2000.

[17] B. Hong, F. Wang, S. A. Brandt, D. D. E. Long, and S. J. Thomas J. E. Schwarz, "Using mems-based storage in computer systems—mems storage architectures," *Transactions on Storage*, vol. 2, no. 1, pp. 1–21, 2006.

[18] M. Uysal, A. Merchant, and G. A. Alvarez, "Using mems-based storage in disk arrays," in *FAST '03: Proceedings of the 2nd USENIX Conference on File and Storage Technologies*, (Berkeley, CA, USA), pp. 89–101, 2003.

[19] M. Sivan-Zimet, "Workload based optimization of probe-based storage," Master's thesis, University of California, Santa Cruz, California, USA, Sept. 2001.

[20] S. W. Schlosser, *Using MEMS-Based Storage Devices in Computer Systems*. PhD thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania, May 2004. Report Nr. CMU-PDL-04-104.