Workload Based Optimization of Probe-Based Storage

[Extended Abstract] *

Miriam Sivan-Zimet
IBM Almaden
650 Harry Road
San Jose, CA 95120
mzimet@almaden.ibm.com

Tara M. Madhyastha
Department of Computer Engineering
University of California, Santa Cruz
Santa Cruz, CA 95064
tara@soe.ucsc.edu

1. INTRODUCTION

The performance gap between microprocessors and secondary storage is still a limitation in today's systems. Academia and industry are developing new technologies to overcome this gap, such as improved read-write head technology and higher storage densities. One promising new technology is probe-based storage[1]. Characteristics of probe-based storage include small size, high density, high parallelism, low power consumption, and rectilinear motion. We have created a probe-based storage simulation model, configurable to different design points, and identify its sensitivity to various parameters.

2. PROBE-BASED STORAGE

Figure 1 is a top view of a probe-based storage device. In this figure, the shaded parts move and the unshaded parts are stationary. The *mover*, or data media, is suspended above a surface on which a grid of many probe-based *tips* are embedded. Collectively, the tip array is the logical equivalent of the read/write head of a traditional disk drive. Electric forces applied to the fingers of the microactuator combs exert electrostatic forces on the mover that cause it to move in the x and y directions, overcoming the forces exerted by the anchors and beams that keep it in place. To service a read or write request, the mover first repositions itself so that the tip array can access the required data. This repositioning time is called *seek time*. The mover then accesses the data while moving at a constant velocity in the y direction, incurring *transfer time*. The time that it takes a mover to reverse direction is called *turnaround* time, and the time to move one bit column in the x direction is called an *x-move*. Finally, the time to switch between sets of active tips is *tip change* time.

A mover may be divided into one or more *clusters*. Each cluster is a media area that is accessed by many tips, only one of which can be active at a time. Using several tips in parallel, one from each cluster, compensates for the low data rate of each individual tip, which is on the order of 1Mbit/sec. The number of bits accessed simultaneously is equivalent to the number of clusters per mover times the number of movers in the device. We call this the number of *active tips*. The mover's range of movement and the bit size determine the amount of data that can be manipulated by one tip, or *tip area*. Because several tips are active at a time, different tip areas of the mover are manipulated simultaneously. Different areas of the mover are accessed by switching between sets of active tips.

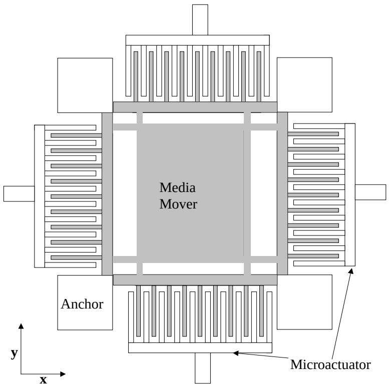

Figure 1: Mover and microactuators.

Many architectural configurations are possible for probe-based storage. For example, we might vary the number of active tips, the mover's movement range, the media density, and so on. To choose one configuration over another, we must understand how the physical configuration affects the performance. Figure 2 shows a simplified version of a dependency graph for the performance analysis of a probe-based storage device. The graph target is the service time, which consists of two parts: seek time and transfer time. Traditional disk data layout minimizes seek time and rotational latency. Analogously, we chose a layout for probe-based storage that has a similar linear ordering.

3. WORKLOAD-BASED OPTIMIZATION

To narrow the design space of our problem and make it more tractable, we divide the parameters into two groups: physical parameters and configurable parameters. We conducted a set of simulations using the Pantheon simulator [4] to study the relationship between service time and configurable parameters. The workloads are 1992 cello (4% sequential, /news partition, most requests smaller than 8KB, sector size is 256B) [2], 1992 snake (23% sequential, /usr2 partition, most requests smaller than 8KB, sector size is 512B) [2], and 1999 cello (30% sequential, large requests where more than half are for more than 8KB, sector size is 1024B) [3].

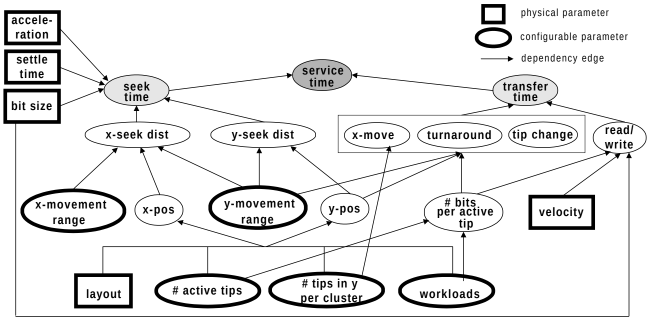

Figure 2: **Design space parameter dependency graph.**

Service time is composed of two components: transfer time and seek time. We checked the sensitivity of each of these components to each of the configurable parameters. Request sizes play the major role in transfer time calculation. Because most of the requests in both 1992 cello and snake are for 8KB, we encounter a similar behavior in both traces. 1999 cello has larger requests and so higher transfer times. Transfer time is sensitive, more than any other parameter, to the number of active tips. Higher numbers of active tips will make the transfer more parallel, so that every active tip accesses fewer bits. We determined that the number of active tips that gives relatively low transfer time at a reasonable cost is 320 (calculated from 20 movers, 16 clusters per mover). This choice gives transfer time that is on the same order of magnitude as the seek time.

To understand more specifically the relationship of the number of active tips to service time, we compared the service times for cello 1992 using three different values for the number of active tips (80, 320, and 1280). Results are shown in Figure 3, and they are similar to 1992 snake [3]. The prominent difference between the three graphs is the service time values. For 1992 cello, 80 active tips result in service time of about 1.7ms. With 320 active tips, which is four times higher, this value decreases by almost half to about 0.9ms. With 1280 active tips, which is again four times 320, the improvement does not follow the same ratio, and is about 0.7ms. The graphs show that the transfer time is very sensitive to the number of active tips, in contrast to the seek time, which does not change dramatically as we quadruple the number of active tips. This comparison supports the choice of 320 active tips, under the guidelines that higher values will not be chosen if the improvement gained is not significant.

Seek time results show that seek time is mainly sensitive to the movement range. The movement range in x and y set the dimensions of one tip area. As we increase the movement range in y, the tip accesses more sectors before turnaround or x-move occurs. Therefore seek time is more sensitive to the movement in y than in x.

It is misleading to compare only the service times, because the capacity changes with the movement range. The movement range should be the one that optimizes the ratio of service time to capacity. For example, for a 2.4GB device the best configuration is 40×40μm. Increasing the movement range is a tradeoff between seek time and capacity. Examination of the ratio between seek time and capacity for different movement ranges indicates that our default values act as a lower bound for the movement range necessary to a good ratio of seek time to capacity.

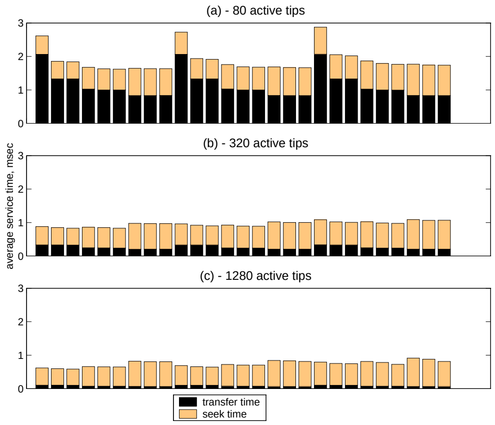

Figure 3: **Service time for 1992 cello with different values for movement range in x and y, and number of tips in y per cluster. Movement in x is: 20μm, 40μm, 80μm changing every nine bars. Movement in y is: 20μm, 40μm, 80μm changing every three bars. Number of tips in y per cluster: 2,10,25, changing every bar.**

4. CONCLUSIONS

We have outlined a design space for probe-based storage devices and studied their behavior under different workloads and array configurations. We identified the dependencies between service time and a set of configurable parameters. We simulated the device model with traced workloads, checking the performance with different set of parameter values. Choosing values for the configurable parameters is a tradeoff between capacity, cost, and performance. We found an optimized lower bound for these values. Our results show that the same set of values is suitable for workloads that differ in their sequentiality level and request sizes.

We conclude that it is possible to construct a probe-based storage device with a seek time to transfer ratio similar to traditional disks that will have a very low service time (e.g., less than 1ms for 1992 cello) for a variety of workloads from traditional file systems.

5. NOTES

This research was supported in part by the National Science Foundation under grant NSF CCR-0073509. A full version of this paper is available as *UCSC-CRL-02-19* at www.soe.ucsc.edu/ tara/stargroup/ Publications/publications.html

6. REFERENCES

[1] GRIFFIN, J. L., W.SCHLOSSER, S., GANGER, G. R., AND NAGLE, D. F. Modeling and performance of MEMS-based storage devices. In *Proceedings of ACM SIGMETRICS 2000* (Santa Clara, California, USA, June 2000), pp. 56–65.

[2] RUEMMLER, C., AND WILKES, J. Unix disk access patterns. In *Proc. of Winter'93 USENIX Conference* (Jan 1993), pp. 405–420.

[3] SIVAN-ZIMET, M. Workload based optimization of probe-based storage, M.S thesis. Tech. Rep. 01-06, University of California, Santa Cruz, June 2001.

[4] WILKES, J. The Pantheon storage-system simulator. Tech. Rep. HPL-SSP-95-14, Storage Systems Program, Computer Systems Laboratory, Hewlett-Packard Laboratories, Palo Alto, CA, May 1996.