Talagala, N., S. Asami, D. Patterson, R. Futernick, and D. Hart [2000]. "The art of massive storage: A case study of a Web image archive," *Computer* (November).

Talagala, N., and D. Patterson [1999]. "An analysis of error behavior in a large storage system," Tech. Report UCB//CSD-99-1042, Computer Science Division, University of California at Berkeley (February).

Thadhani, A. J. [1981]. "Interactive user productivity," *IBM Systems J.* 20:4, 407–423.

Exercises

Solutions to the "starred" exercises appear in Appendix B.

✪ 7.1 [15] <7.14> Using the two formulas in the fallacy starting on page 767 and the coefficient definitions given in the caption of Figure 7.51 (page 768), calculate the seek time each way for moving the arm over one-third of the cylinders of the disks in Figure 7.2 (page 682). Assume that the disk access is a read. What is the error with respect to the manufacturer's reported average seek time for the two formulas? What does this say about the suitability of the coefficient definitions in Figure 7.51 for the disks in Figure 7.2? (*Hint:* Using a spreadsheet program will be helpful to find the answers for this and several other exercises in this chapter.)

7.2 [25] <7.14> Using the two formulas in the fallacy starting on page 767 and the coefficient definitions given in Figure 7.51 (page 768), write a short program to calculate two "average" seek times by estimating the time for all possible seeks using these formulas and then dividing by the number of seeks. How close are the answers for Exercise 7.1 to these answers?

7.3 [15/12] <7.14> Average seek distance depends on the actual disk accesses generated by a workload.

a. [15] <7.14> Using the statistics in the caption of Figure 7.52 (page 769) and in the displayed bar graphs, calculate the average seek distance for the two workloads. Use the midpoint of a range as the seek distance. For example, use 98 as the seek distance for the entry representing 91–105 in Figure 7.52. For the business workload, just ignore the missing 6% of the seeks. For the UNIX workload, assume the missing 15% of the seeks have an average distance of 300 cylinders.

b. [12] <7.14> If the two workloads in Figure 7.52 were each measured on the three disks in Figure 7.2, what similarities and what differences could be expected in the three sets of results? Explain.

7.4 [20] <7.14> Figure 7.2 (page 682) gives the manufacturers' average seek times. Using the two formulas in the fallacy starting on page 767, the definitions of the coefficients given in Figure 7.51 (page 768), and assuming the statistics in Figure 7.52 (page 769) and read accesses only, what are the average seek times for each workload on the disks in Figure 7.2? Make the same assumptions as in part (a) of Exercise 7.3.

7.5 [10/10/10/10/10] <7.2> In this exercise, we will run a program to evaluate the behavior of a disk drive. Disk sectors are addressed sequentially within a track, tracks sequentially within cylinders, and cylinders sequentially within the disk. Determining head switch time and cylinder switch time is difficult because of rotational effects. Even determining platter count, sectors/track, rotational delay, and minimum time to media is difficult based on observation of typical disk workloads. The key is to factor out disk rotational effects by making consecutive seeks to individual sectors with addresses that differ by a linearly increasing amount starting with 0, 1, 2, and so forth.

The Skippy algorithm, from work by Nisha Talagala and colleagues of U.C. Berkeley [2000], is

```
fd = open("raw disk device");
for (i = 0; i < measurements; i++) {
            //time the following sequence, and output <i, time>
            lseek(fd, i * SINGLE_SECTOR, SEEK_CUR);
            write(fd, buffer, SINGLE_SECTOR);
}
close(fd);
```

The basic algorithm skips through the disk, increasing the distance of the seek by one sector before every write, and outputs the distance and time for each write. The raw device interface is used to avoid file system optimizations. SINGLE_SECTOR is the size of a single sector in bytes. The SEEK_CUR argument to lseek moves the file pointer an amount relative to the current pointer. A technical report describing Skippy and two other disk drive microbenchmarks (run in seconds or minutes rather than hours or days) is at *http://sunsite.berkeley.edu/Dienst/UI/2.0/Describe/ncstrl.ucb/CSD-99-1063*.

Run the Skippy algorithm on a disk drive of your choosing.

a. [10] <7.2> What is the number of heads? The number of platters?

b. [10] <7.2> What is the rotational latency?

c. [10] <7.2> What is the head switch time?

d. [10] <7.2> What is the cylinder switch time?

e. [10] <7.2> What is the minimum time to media plus transfer time? The minimum time to media is the minimum time to access the disk platter surface. A disk request completes in the sum of the minimum time to media plus the transfer time if there is no rotational or seek latency.

✪ 7.6 [10/10/10/10/10] <7.2> Figure 7.53 shows the output from running the benchmark Skippy on a disk.

a. [10] <7.2> What is the number of heads? The number of platters?

b. [10] <7.2> What is the rotational latency?

c. [10] <7.2> What is the head switch time?

d. [10] <7.2> What is the cylinder switch time?

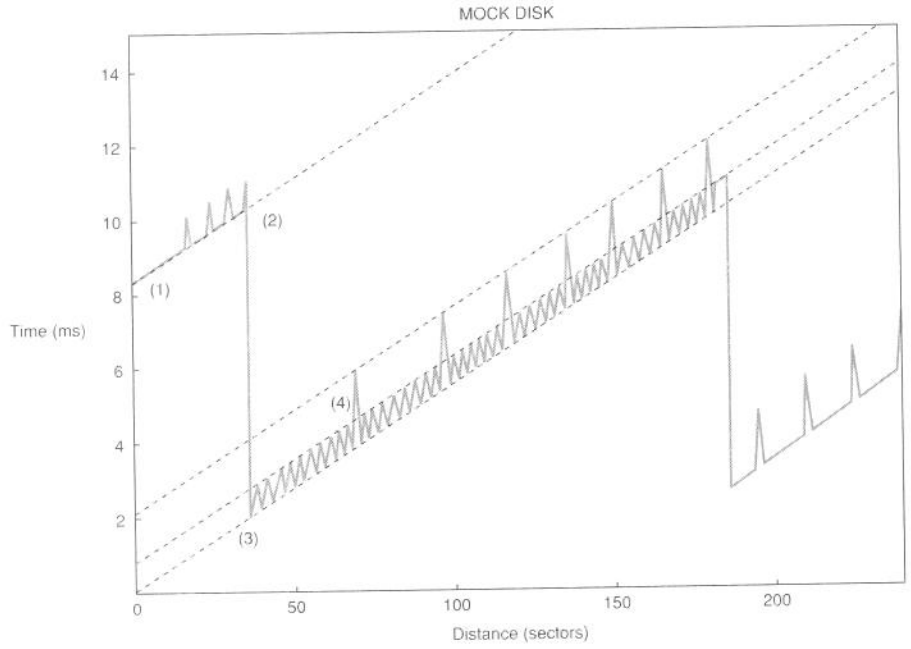

Figure 7.53 Example output of Skippy for a hypothetical disk.

e. [10] <7.2> What is the minimum time to media plus transfer time? The minimum time to media is the minimum time to access the disk platter surface. A disk request completes in the sum of the minimum time to media plus the transfer time if there is no rotational or seek latency.

7.7 [20/15/15/15/15/15] <7.7> The I/O bus and memory system of a computer are capable of sustaining 1000 MB/sec without interfering with the performance of a 2500 MIPS CPU (costing $20,000). Here are the assumptions about the software:

- Each transaction requires 4 disk reads plus 2 disk writes.
- The operating system uses 35,000 instructions for each disk read or write.
- The database software executes 50,000 instructions to process a transaction.
- The transfer size is 512 bytes.

You have a choice of two different types of disks:

- A small disk that stores 40 GB and costs $400.
- A big disk that stores 80 MB and costs $800.

Either disk in the system can support on average 100 disk reads or writes per second.

Answer parts (a)–(f) using the TPC-C benchmark (complex query OLTP) in Section 7.9. Assume that the requests are spread evenly to all the disks, that there is no waiting time due to busy disks or controllers, and that the account file must be large enough to handle 10,000 transactions per minute (tpm) according to the benchmark ground rules.

a. [20] <7.7> How many TPC-C transactions per second are possible with each disk choice, assuming that each uses the minimum number of disks to hold the account file?

b. [15] <7.7> What is the system cost per transaction per second of each alternative for TPC-C?

c. [15] <7.7> How fast does a CPU need to be to make the 1000 MB/sec I/O bus a bottleneck for TPS? (Assume that you can continue to add disks.)

d. [15] <7.7> As manager of MTP (Mega TP), you are deciding whether to spend your development money building a faster CPU or improving the performance of the software. The database group says they can reduce a transaction to 2 disk reads and 1 disk write and cut the database instructions per transaction to 30,000. The hardware group can build a faster CPU that sells for the same amount as the slower CPU with the same development budget. (Assume you can add as many disks as needed to get higher performance.) How much faster does the CPU have to be to match the performance gain of the software improvement?

e. [15] <7.7> The MTP I/O group was listening at the door during the software presentation. They argue that advancing technology will allow CPUs to get faster without significant investment, but that the cost of the system will be dominated by disks if they don't develop new small, faster disks. Assume the next CPU is 100% faster at the same cost and that the new disks have the same capacity as the old ones. Given the new CPU and the old software, what will be the cost of a system with enough old small disks so that they do not limit the TPS of the system?

f. [15] <7.7> Start with the same assumptions as in part (e). Now assume that you have as many new disks as you had old small disks in the original design. How fast must the new disks be (I/Os per second) to achieve the same TPS rate with the new CPU as the system in part (e)?

✪ 7.8 [20] <7.7> Assume that we have the following two magnetic disk configurations: a single disk and an array of four disks. Each disk has 12 surfaces, 27,723 tracks per surface, and 528 sectors/track. Each sector holds 512 bytes, and it revolves at 10,000 RPM. The minimum, maximum, and average seeks times are 0.5 ms, 10.5 ms, and 4.9 ms, respectively. Use the seek time formula in the fallacy starting on page 767, including the equations in Figure 7.51 (page 768). The time to switch between surfaces is the same as to move the arm one track. In the disk array all the spindles are synchronized—sector 0 in every disk rotates under the head at the exact same time—and the arms on all four disks are always over the same track. The data is "striped" across all four disks, so four consecutive sectors on a

single-disk system will be spread one sector per disk in the array. The delay of the disk controller is 0.3 ms per transaction, either for a single disk or for the array. Assume the performance of the I/O system is limited only by the disks and that there is a path to each disk in the array. Calculate the performance in both I/Os per second and megabytes per second of these two disk organizations, assuming the request pattern is random reads of 4 KB of sequential sectors. Assume the 4 KB are aligned under the same arm on each disk in the array.

7.9 [20] <7.7> Start with the same assumptions as in Exercise 7.8. Now calculate the performance in both I/Os per second and megabytes per second of these two disk organizations assuming the request pattern is reads of 4 KB of **sequential** sectors where the average seek distance is 10 tracks. Assume the 4 KB are aligned under the same arm on each disk in the array.

7.10 [20] <7.7> Start with the same assumptions as in Exercise 7.8. Now calculate the performance in both I/Os per second and megabytes per second of these two disk organizations assuming the request pattern is random reads of 1 MB of sequential sectors. (If it matters, assume the disk controller allows the sectors to arrive in any order.)

7.11 [20] <7.2> Assume that we have one disk defined as in Exercise 7.8. Assume that we read the next sector after any read and that *all* read requests are one sector in length. We store the extra sectors that were read ahead in a disk cache. Assume that the probability of receiving a request for the sector we read ahead at some time in the future (before it must be discarded because the disk cache buffer fills) is 0.1. Assume that we must still pay the controller overhead on a disk cache read hit, and the transfer time for the disk cache is 50 ns per word. Is the read-ahead strategy faster? (*Hint:* Solve the problem in the steady state by assuming that the disk cache contains the appropriate information and assuming that a request has just missed.)

7.12 [20/10/20/20] <7.7–7.10> Assume the following information about a MIPS machine:

- Loads take 2 cycles.
- Stores take 2 cycles.
- All other instructions are 1 cycle.

Use the summary instruction mix information on MIPS for gcc from Figure 2.32.

Here are the cache statistics for a write-through cache:

- Each cache block is 4 words, and the whole block is read on any miss.
- A cache miss takes 23 cycles.
- Write through takes 16 cycles to complete, and there is no write buffer.

Here are the cache statistics for a write-back cache:

- Each cache block is 4 words, and the whole block is read on any miss.
- A cache miss takes 23 cycles for a clean block and 31 cycles for a dirty block.
- Assume that on a miss, 30% of the time the block is dirty.

Assume that the bus

- is only busy during transfers
- transfers on average 1 word/clock cycle
- must read or write a single word at a time (it is not faster to access two at once)

a. [20] <7.7–7.10> Assume that DMA I/O can take place simultaneously with CPU cache hits. Also assume that the operating system can guarantee that there will be no stale-data problem in the cache due to I/O. The sector size is 1 KB. Assume the cache miss rate is 5%. On average, what percentage of the bus is used for each cache write policy? (This measure is called the *traffic ratio* in cache studies.)

b. [10] <7.7–7.10> Start with the same assumptions as in part (a). If the bus can be loaded up to 80% of capacity without suffering severe performance penalties, how much memory bandwidth is available for I/O for each cache write policy? The cache miss rate is still 5%.

c. [20] <7.7–7.10> Start with the same assumptions as in part (a). Assume that a disk sector read takes 1000 clock cycles to initiate a read, 100,000 clock cycles to find the data on the disk, and 1000 clock cycles for the DMA to transfer the data to memory. How many disk reads can occur per million instructions executed for each write policy? How does this change if the cache miss rate is cut in half?

d. [20] <7.7–7.10> Start with the same assumptions as in part (c). Now you can have any number of disks. Assuming ideal scheduling of disk accesses, what is the maximum number of sector reads that can occur per million instructions executed?

7.13 [50] < 7.7> Take your favorite computer and write a program that achieves maximum bandwidth to and from disks. What is the percentage of the bandwidth that you achieve compared with what the I/O device manufacturer claims?

7.14 [20] <7.2, 7.4> Search the World Wide Web to find descriptions of recent magnetic disks of different diameters. Be sure to include at least the information in Figure 7.2 on page 682.

7.15 [20] <7.14> Using data collected in Exercise 7.14, plot the two projections of seek time as used in Figure 7.51 (page 768). What seek distance has the largest percentage of difference between these two predictions? If you have the real seek distance data from Exercise 7.14, add that data to the plot and see on average how close each projection is to the real seek times.

7.16 [15] <7.2, 7.4> Using the answer to Exercise 7.15, which disk would be a good building block to build a 2 TB storage subsystem using mirroring (RAID 1)? Why?

7.17 [15] <7.2, 7.4> Using the answer to Exercise 7.15, which disk would be a good building block to build a 20 TB storage subsystem using distributed parity (RAID 5)? Why?

7.18 [15] <7.8> Starting with the example on page 728, calculate the average length of the queue and the average length of the system for that example and the following two examples.

7.19 [15] <7.8> Redo the example that starts on page 728, but this time assume the distribution of disk service times has a squared coefficient of variance of 2.0 (C = 2.0), versus 1.0 in the example. How does this change affect the answers?

7.20 [20] <7.11> The I/O utilization rules of thumb on page 748 are just guidelines and are subject to debate. Redo the example starting on page 749, but increase the limit of SCSI utilization to 50%, 60%, . . . , until it is never the bottleneck. How does this change affect the answers? What is the new bottleneck? (*Hint:* Use a spreadsheet program to find answers.)

7.21 [15/15] <7.2> Tape libraries were invented as archival storage, and hence have relatively few readers per tape.

a. [15] <7.2> Calculate how long it would take to read all the data from a StorageTek PowderHorn 9310 assuming a single silo with 6000 tapes, 60 GB uncompressed capacity per tape, 16 tape drives that read at 11 MB/sec, and a maximum of 450 tape changes per hour per drive (includes tape load/unload time in the drive plus robot arm time to/from the tape storage slot in the silo).

b. [15] <7.2> Assume the 16 tape drives each have a helical scan read/write head with a 2000-hour rated lifetime. How many complete scans of the storage silo in part (a) can be done before exceeding the head lifetime?

7.22 [25] <7.2> Extend Figure 7.3 on page 685 and Figure 7.4 on page 686 to the present time, showing price per disk and price per gigabyte by collecting data from advertisements in the January and July issues of *PC* magazine. How fast are prices changing now?

7.23 [Discussion] <7.2> Recording density for disk drives has increased exponentially for decades. The superparamagnetic limit is a physical characteristic of recording media that may soon thwart density improvements. What is the superparamagnetic limit? Do people believe it is a real limit? What would be the impact if it were? Search the literature on magnetic recording for information to support your discussion. One place to start your investigation is *http://www.research.ibm.com/journal/rd/443/thompson.html*.

7.24 [Discussion] <7.2> With the cost in 2001 of a 40 GB IDE disk about the same as a 40 GB tape, the economics of magnetic tape are far different from earlier days when tape performed software distribution, mass storage, backup, and disaster insurance (an easy form in which to send data to a remote site) functions. What technologies or combinations of technologies are the competition faced by tape for each of these functions? What advantages are offered by the competing technologies? For which functions does tape face the strongest competition? What advantages can tape offer compared to the competing technologies?

7.25 [Discussion] <7.2> Figure 7.4 on page 686 shows that the price per gigabyte of personal computer disks decreased by a factor of 10,000 over the span of 18 years. Change of such magnitude is difficult to fully appreciate, especially so

without a direct, personal context. Imagine your own life 18 years from today as it would be with 10,000-fold improvement in aspects of daily life that matter to you. Some examples might be your salary or the speed of long-distance travel. Can you find examples from outside the field of computing of comparable growth rates (about 60% per year) over a similar period?

7.26 [50] <7.5, 7.11> A more sophisticated analysis of RAID failures relies on Markov models of faults; see Gibson [1992]. Learn about Markov models and redo the simplified failure analysis of the disk array on page 746.

7.27 [Discussion] <7> Text, audio, photo, and video works of popular interest have significant economic value. Today, each of these formats when represented digitally can be stored cheaply, copied easily and exactly, and edited readily by inexpensive desktop computer systems. Recordable, removable media storage and high-speed networking provide distribution channels of improving cost and performance. Authoring and communicating digital works is within the means of more organizations and individuals than ever before, but at the same time the obstacles to the unauthorized copying and dissemination of the digital works of others are shrinking. One strategy to prevent or limit copyright infringement is to control storage technology.

What involvement have the recording and movie industries had in the computer storage marketplace for the digital audiotape, digital minidisk, CD-R, CD-RW, and DVD formats? How have the availability of standard computer bus interfaces for these devices and their hardware unit and recording media pricing been affected?

What are some of the existing technologies and proposed methods for storage devices to prevent unauthorized copying of intellectual property and to support data access models other than ownership, such as viewing restrictions, limited number of viewings, and pay-per-view (e.g., Copy Protection for Recordable Media (CPRM), CSS and region codes for DVD, and encoding to prevent media interoperability between consumer audio CD players and CD-ROM readers)? How are these techniques faring in the consumer marketplace and other marketplaces? What are or would be the effects of these access controls on an individual wishing to work with data files of his/her own authorship?