Research Feature

Advances in Disk Technology: Performance Issues

Although the computer industry has made regular, significant advances in magnetic recording technology for hard disk drives, some advances—such as those in head design, media, and channel technology—are primarily concerned with increasing disk density and do not necessarily improve total performance.

Spencer W. Ng
IBM Almaden Research Center

Over the past few years, we've seen magnetic recording technology enjoy rapid advancements, bringing us new technologies like thin-film discs, magneto-resistive heads, and maximum-likelihood-partial-response channels. Combined with continuous evolutionary improvements, these and other advances provide the computer industry with disk drives that are ever cheaper, smaller, and more capacious—a trend that's likely to continue.[1,2]

Some of the advances also improve disk performance. These advances include increased rotational speeds, faster seek times, and higher data transfer rates.[3] However, the impact of other disk technology advances—such as increases in disk density or total drive capacity—is less clear and not so well understood.

How, for example, does disk density affect data transfer rate? How do multiple platters affect head seek time? And how does partitioning large-capacity drives affect performance for each partition? The answers to these questions—and others like them—largely depend on particular configurations and user workload.

PERFORMANCE FACTORS

We measure disk drive performance by how fast a disk can complete a user request for reading or writing data.[4] The simplest benchmarking method is the *total job completion time* for a complex task involving a long sequence of disk I/Os. An alternative benchmark is *throughput*, which is the amount of data or number of accesses that a user can transact with a drive, per unit of time, while the drive performs a particular type of workload.

The time required by a disk drive to execute and complete a user request consists of four major components: *command overhead*, *seek time*, *rotational latency*, and *data transfer time*.

Command overhead

Command overhead—the time it takes for the disk drive's microprocessor and electronics to process and handle an I/O request—depends on the type of drive interface (IDE or SCSI), whether the command is a read or a write, and whether the command can be satisfied from the disk drive's buffer or cache memory (a buffer hit or a buffer miss). Command overhead has been declining over the years due to faster embedded controller chips and more control functions getting hardware assist. In today's drives, typical command overhead is around 0.5 ms for buffer miss and 0.1 ms for buffer hit.

Seek time

Seek time—the time to move the head from its current cylinder to the target cylinder specified by the next command—has been decreasing ever since the early IBM RAMAC days. Much improvement comes from smaller and lighter disk components. Reducing disk diameter from 14 inches some years ago to as small as 1.8 inches today, with 3.5 inches being the most common size, also means the arm has less distance to travel. As seek distance decreases, *settling time*—the time to position the head over the target track until correct track identification is confirmed—becomes a relatively more important component.[3] While the fastest drives today have average seek times of less than 8 ms, 10 ms is fairly typical.

Rotational latency

Once the head has arrived at the target cylinder, rotational latency is the time it takes for the target sector to rotate under the head. Average rotational latency is one-half the time it takes the disk to do one revolution. Therefore, it is inversely proportional to rotational speed. For many years, drives of all sizes

0018-9162/98/$10.00 © 1998 IEEE

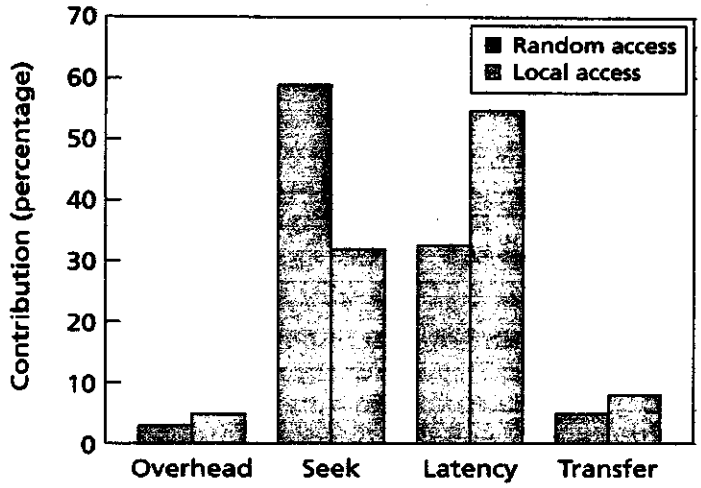

Figure 1. Relative contribution of different components to I/O time for a 4-Kbyte-block request.

rotated at 3,600 rpm. Manufacturers are now realizing the importance of speed for performance and are thus increasing rpm. The highest performance drives today spin at 10,000 rpm; however, 5,400 rpm is by far more common, representing an average latency of 5.6 ms.

Data transfer time

Data transfer time depends on *data rate* and *transfer size*. There are two kinds of data rate: media and interface. *Media data rate* is how fast data can be transferred to and from the magnetic recording media. The media data rate has been increasing as a natural consequence of increasing recording density and rotational speed. For example, a disk rotating at 5,400 rpm with 111 sectors (512 bytes each) per track will have a media data rate of 5 Mbytes per second.

Interface data rate, on the other hand, is how fast data can be transferred between the host and the disk drive over its interface. SCSI drives supporting the SCSI-3 standard can do up to 20 Mbytes per second over each 8-bit-wide transfer. In contrast, IDE drives with the Ultra-ATA interface can support up to 33.3 Mbytes per second. For the purpose of discussion, 10 Mbytes per second is assumed to be the average interface data rate.

The average transfer size depends on the application and the host operating system. While the average transfer size has been creeping upward—with some video applications transferring 64 Kbytes or more per I/O—a more modest 4-Kbyte transfer size is common in many popular operating systems, such as Microsoft Windows.

Transfer time equals transfer size divided by data rate. With the above assumed typical numbers, the average media transfer time is 0.8 ms, while the average interface transfer time is 0.4 ms.

How it adds up

Therefore, the typical average time to do a random 4-Kbyte disk I/O with today's disk drives is

$$\text{overhead} + \text{seek} + \text{latency} + \text{transfer} = 0.5 \text{ ms} + 10 \text{ ms} + 5.6 \text{ ms} + 0.8 \text{ ms} = 16.9 \text{ ms}$$

For most systems, though, I/Os are not completely random but are often confined to some small portion of the disk drive during any given short window of time—a phenomenon called *locality of access*.[3,5,6] Its net effect is that the real seek component is actually smaller than the random average—often roughly one-third its size.[5,6] Hence, in this local access environment, the typical average time to do a 4-Kbyte disk I/O is given in this equation:

$$\text{overhead} + \text{seek} + \text{latency} + \text{transfer} = 0.5 \text{ ms} + 1/3 \times 10 \text{ ms} + 5.6 \text{ ms} + 0.8 \text{ ms} = 10.2 \text{ ms}$$

Figure 1 illustrates the relative contribution of the different components to disk I/O time for both random access and local access. It shows that for local access, which is the dominant user environment, latency accounts for the greatest share of I/O time.

Caching

Caching, either in the host or in the disk drive, can substantially improve I/O performance by avoiding slow mechanical access. Caching in the disk drive is particularly effective when it is used to do look-ahead prefetching, which is why all modern-day disk drives generally provide this feature. The net effect of disk cache access is that the mechanical components, namely seek and latency, are eliminated, and data transfer takes place at the interface data rate rather than the media data rate. Thus, the typical time to do a 4-Kbyte I/O access becomes

$$\text{overhead} + \text{transfer} = 0.1 \text{ ms} + 0.4 \text{ ms} = 0.5 \text{ ms}$$

This rate of access is an order of magnitude faster than retrieving data from the disk's media. Increasing the cache's hit ratio is very effective in improving a disk drive's performance.[7,8,9]

Other factors

Some of the technological advances are specifically aimed toward making the disk drive run faster, and as such they are clearly beneficial to performance. Any improvement that can reduce one or more of the disk service time components obviously falls into this category.

For instance, increasing rotational speed will reduce latency time and at the same time also reduce data transfer time, because data rate is increased as a result

(assuming the number of sectors per track stays the same). The performance impact of these improvements is self-evident, but that of other technological improvements—including those that increase recording density, increase volumetric density, make use of no-ID recording format, and provide a larger drive capacity—is less clear.

INCREASED RECORDING DENSITY

Over the past few years, magnetic recording *areal density* has increased at a rate of about 60 percent per year.[1] This increase has resulted from an increase in both bits per inch (bpi) and tracks per inch (tpi). If bpi and tpi contribute about equally to the increase, then each component is growing at approximately 27 percent per year. Increases in bpi and tpi, as well as in overall areal density, have subtle implications for a disk drive's performance.

Higher bits per inch

Bits per inch, also called *linear density*, dictates how many bits can be stored on a track, which in turn determines the number of sectors on a track. Generally speaking, then, higher bpi means more sectors per track. Today's disk drives almost invariably use zoned recording to maximize a disk's storage capacity. Within each zone, the number of sectors per track is constant. This means the bpi toward the outer diameter of a zone is somewhat lower than the bpi toward the inner diameter of the same zone. Ideally, the bpi at the inner diameter of all the zones should be about the same and should be at the maximum value that the technology used in the disk drive allows.

Increasing bpi has several effects on performance, including a higher media data rate, a constraint on rpm, fewer head switches, and a bigger cylinder.

Higher media data rate. Because

$$\text{media data rate} = 2\pi \times \text{radius} \times \text{bpi} \times \text{rotational speed}$$

for a given rpm, a higher bpi implies a higher media data rate. This is obviously good for performance. Since bpi is increasing every year, increasing applications' block size of data transfer should allow them to take greater advantage of the I/O time reduction due to an improved data rate. This is illustrated in Figure 2, which assumes a 30 percent improvement in data rate.

Constraint on rpm. While increasing the media data rate is good, increasing the bpi too much can push the data rate beyond what the drive's data channel can handle (because of cost or technology limitations). For example, today's disk drive electronics can handle only up to about 25 Mbytes per second. Disk drive designers are then faced with a dilemma: They could forfeit some of the increase in bpi, in which case the drive would not gain as much capacity as it otherwise would, or they could reduce the rotational speed. Even when channel technology is improving to support higher speed, designers may as well be able to increase the bpi commensurately, but they would not be able to increase the drive's rpm. Reducing the rpm—or not being able to increase it—tends to limit the ability to enhance a disk drive's performance.

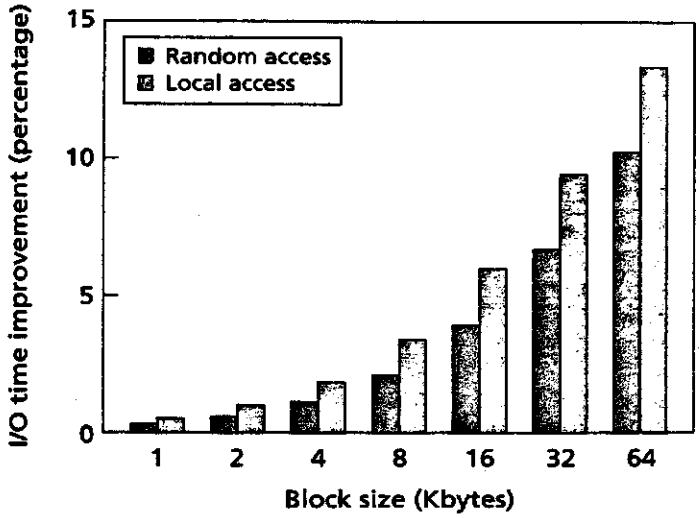

Figure 2. Net effect on I/O time with a 30 percent improvement in data rate.

Fewer head switches. Whenever the end of a track is reached, it takes the drive a finite amount of time to switch to the next track. Switching to the next track on the same cylinder is commonly known as a *track switch*, and switching to the next track on the next cylinder is commonly known as a *cylinder switch*. The track or cylinder switch time, typically in the order of milliseconds, adds to total I/O time if the requested piece of data spans multiple tracks. In fact, for a given request size, the average value of this additional time component of an I/O time is

$$\text{average switch time} = (\text{request size} - 1/\text{track size}) \times \text{head switch time}$$

where request size and track size are both in number of sectors. When there are more sectors on a track—due to increased bpi—a small piece of data is less likely to span two tracks, while larger data will span fewer tracks. In either case, avoiding having to do a head switch or reducing the number of head switches is good for performance.

Bigger cylinder. If the number of heads does not change as bpi is increased, then more sectors per track means more sectors per cylinder. Since many systems, especially single-user desktop machines, only run one or a few applications at a time, only a small fraction of the disk

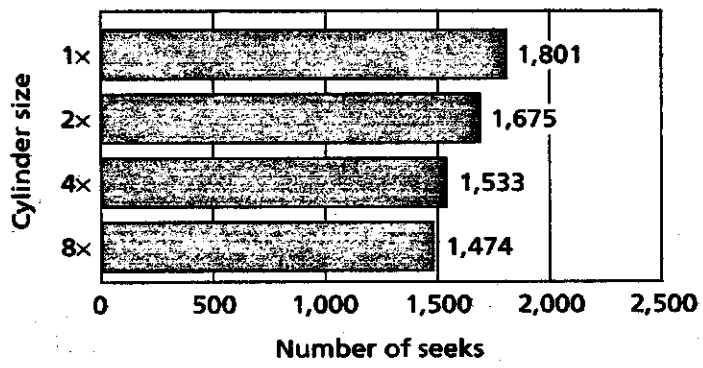

Figure 3. Total number of seeks in simulation of PC workloads.

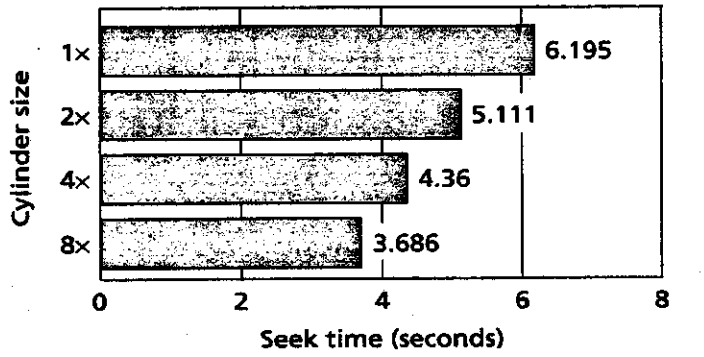

Figure 4. Total seek time in simulation of PC workloads.

drive's data is being accessed during any window of time. When operating within a small range of data, having more data sectors in a cylinder has two effects:

1. *The seek distance is reduced.* If the size of each cylinder is increased by, say, 30 percent because bpi is increased by that percentage, then an equal amount of data would occupy only 77 percent (100/130) as many cylinders as before. As a result, seek distance would be reduced by 23 percent. Shorter seek distance means shorter seek time.
2. *The number of seeks is reduced.* When dealing only with a small amount of data, having a bigger cylinder increases the likelihood that the next piece of data required by the user will be found in the current cylinder, thus avoiding a seek completely. As an example, consider an application program involving 16 Mbytes of data. For a hypothetical drive with six tracks per cylinder and 80 Kbytes (160 sectors) per track, there is a 3 percent probability that the next piece of nonsequential data that this program accesses is located in its current cylinder and therefore does not require a seek. Increasing the size of the cylinder will raise this probability proportionally.

These two effects of higher bpi result in either reducing or eliminating seek time, therefore improving performance. We verified this improvement using an in-house event-driven simulator, which tracks the total number of seeks and total seek time for each I/O trace input. Starting with a hypothetical disk drive with the average parameters given earlier, we increased the number of sectors per track two, four, and eight times, keeping all remaining parameters the same.

Using a workload of several popular PC applications as the input trace, the simulation produced the number of seeks shown in Figure 3 and the total seek times shown in Figure 4. As these figures show, doubling the size of a cylinder reduces the number of seeks by 7 percent and the total time spent on seeking by 17 percent.

Higher tracks per inch

Seek time actually consists of two parts: (1) travel time for the actuator to move from its current position to a point near the target cylinder and (2) settling time. While travel time is relatively simple, settling time is quite complex and depends on many factors.[10] When every other variable is fixed, a given servo should take longer to settle if the tracks are narrower and closer together.[3] A simplistic first-order-of-approximation model of the seek time of a drive that uses maximum acceleration—one that accelerates at its maximum speed to the halfway point, then decelerates at the same rate—is

$$\text{seek time} = A + B \times \sqrt{\text{<seek distance>}} + C \times \log(\text{tpi})$$

where A, B, and C are some constants specific to the disk drive. In this equation, tpi has two opposing effects on the seek time of an I/O. First, assuming everything else is equal—the same number of sectors per track and the same number of tracks per cylinder—higher tpi means a shorter physical seek distance from one given logical block to another given logical block. This means the travel time is smaller, which helps to reduce the seek time, as the above equation shows.

On the other hand, tracks that are narrower and closer together require a longer settling time, as the last component of the equation indicates. Thus, whether increasing tpi is good for performance depends on which of these two opposing effects is more dominant. Because many single-user systems access only a small portion of the disk during any window of time, seek distance is short and the settling time dominates, making increased tpi bad for performance.

Fewer heads

Drive manufacturers commonly take advantage of the increased density that advanced technology provides to reduce the cost of a disk drive by achieving a given capacity using fewer disk platters and heads. For example, if a certain recording technology allows 800 Mbytes

of data on one disk platter, a 4-Gbyte disk drive would require five platters. If the disk density is increased by 67 percent, only three platters would be required for a 4-Gbyte drive. While cost is reduced, this design option has some performance implications, including a lower sustained data rate and a smaller cylinder.

Lower sustained data rate. The *sustained data rate* is the actual data rate a user gets from a disk drive when transferring large amounts of data spanning many tracks. It is different from the media data rate because every time the drive reaches the end of a track it takes a finite amount of time to switch to the next track. The various parameters affect the sustained data rate according to the following equation:

$$\text{sustained data rate} = \text{media data rate} \times \frac{(\text{no. heads} \times \text{rot. time})}{(\text{no. heads} \times \text{rot. time} + (\text{no. heads} - 1) \times \text{trk. switch time} + \text{cyl. switch time})}$$

Since cylinder switch time is typically larger than track switch time,[3] the equation above shows that the sustained data rate is lowered if the number of heads is reduced. This is also shown in Figure 5, which illustrates a 10-ms rotation time, 2-ms track switch time, and 4-ms cylinder switch time.

Therefore, having fewer heads by itself is not good for performance. However, in this case where fewer heads results from higher density, the decrease in sustained data rate is offset—or perhaps even more than compensated for—by any increase in media data rate resulting from higher bpi.

Smaller cylinder. Since an increase in areal density results from increases in bpi and tpi, the total number of sectors in a cylinder actually decreases if the number of heads is reduced to maintain constant capacity. A disk drive manufacturer will maintain capacity by using newer technology to produce a lower cost version of an existing product. Some systems whose performance is limited by the disk arm may not be able to use more capacity in a drive. When bpi is increased by a factor of x and tpi by a factor of y, then the areal density is increased by a factor of xy. Because

$$\text{drive capacity} = \text{no. tracks per cylinder} \times \text{no. cylinders per surface} \times \text{no. sectors per track}$$

the number of tracks per cylinder must be correspondingly reduced by a factor of xy if capacity is to stay unchanged. Since the number of sectors per track is increased by only x, the number of sectors in a cylinder is actually reduced by a factor of y. One of the benefits of a bigger cylinder is that the number of seeks is reduced. A smaller cylinder, then, would have the opposite effect, increasing the number of seeks and reducing performance. Note, however, that there is no change in the average physical seek distance. This is because even though the seek distance in number of cylinders is increased by a factor of y, the track density is also increased by that same factor.

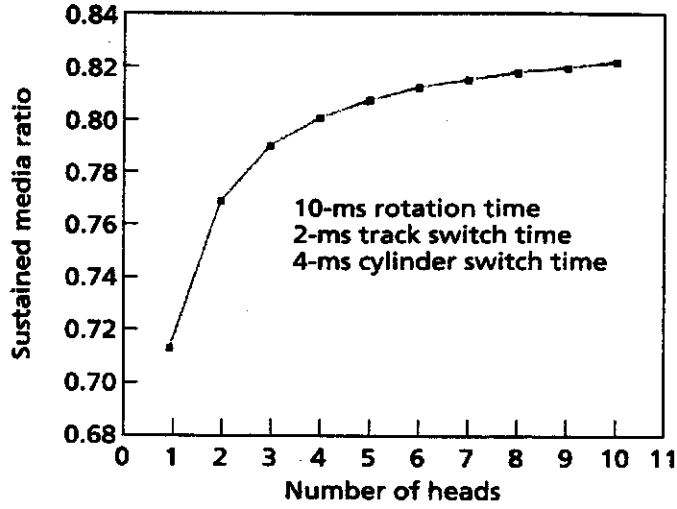

Figure 5. Ratio of sustained data rates to media data rates.

INCREASED VOLUMETRIC DENSITY

Another evolutionary trend seen in magnetic disk drives is an increase in volumetric density. While the growth in areal density contributes to much of this increase, improvement in the third dimension is also a contributing factor: Miniaturization of drive components now makes possible reduced disk-to-disk spacing, allowing more disk platters to be packaged in a drive of a given height. Increasing the number of heads means proportionally increasing the number of sectors in a cylinder. Having more sectors in each cylinder is beneficial to performance—namely in reducing both seek distance and number of seeks.

NO-ID RECORDING FORMAT

The technological advances described above have reduced the physical space required to store information, increasing a drive's raw capacity. To a user, however, a drive's formatted capacity is more important because it is the amount of usable space remaining when formatting overhead is subtracted from raw capacity. In traditional formatting, data bytes are grouped into sectors, each preceded by an *ID field*, or *header*, containing the sector's physical address (cylinder–head–sector). The servo system that controls the head uses the ID field to access the correct data sector. The ID fields—and gaps between them and the sectors—reduce the disk's available space, as shown in Figure 6a.

No-ID recording format increases a disk drive's capacity by making formatting more efficient. As shown in Figure 6b, no-ID recording, or *headerless recording*, eliminates the ID field and its associated gap, thus allowing more data sectors on each track. The drive is able to find the sectors by keeping a table that shows the relationships between sectors and embedded

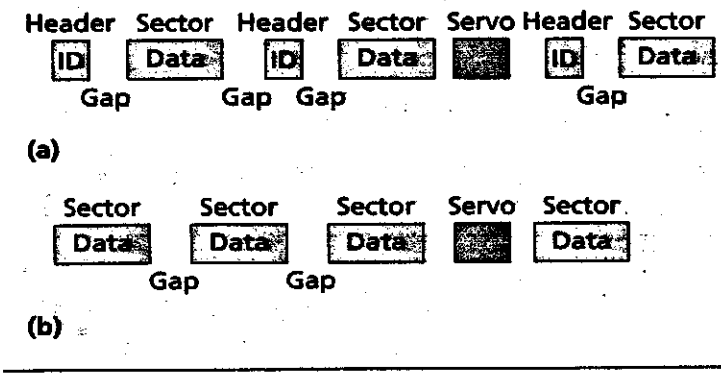

Figure 6. Recording formats: (a) traditional, (b) no-ID.

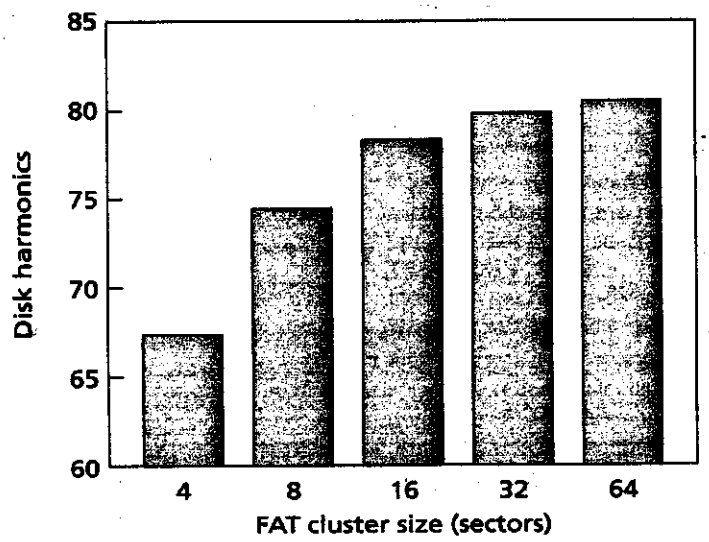

Figure 7. PC Bench 7.01 throughput test results, showing better performance when cluster sizes are larger.

servos. From a performance standpoint, this is about equivalent to increasing the bpi. This new technique, therefore, enjoys these positive performance effects:

- The user sees a higher data rate because of more user data per track.
- More sectors per track means fewer head switches.
- More sectors per track means bigger cylinders for a given number of heads, resulting in fewer and shorter seeks.

Furthermore, since the increase in track size is achieved not by actually changing the underlying bpi, it does not put a constraint on rpm.

LARGER DRIVE CAPACITY

Because of the way certain file systems use the disk drive, a drive's capacity can itself have an effect on observable performance. Most file systems assign disk space to a file in chunks of sectors called *allocation units*. In order to keep the table for managing these allocation units reasonably small, some file systems will limit their number by increasing their size as disk or partition size increases.

For example, in the file allocation table (FAT) of DOS, Windows, and OS/2, where an allocation unit is called a *cluster*, the cluster size is 16 sectors for a drive with capacity between 256 and 511 Mbytes and 32 sectors for a drive with capacity between 512 and 1,023 Mbytes. Just how allocation unit size affects performance depends on the size of the user's files.

Large files

With the FAT file system, each FAT sector contains 256 entries, each representing a cluster. When the cluster size is large, a FAT sector will cover more data sectors, so it takes fewer FAT sectors to describe a large file—one of hundreds or thousands of sectors. For a user, this means fewer FAT accesses to look up the location of all data associated with the file. Fewer FAT accesses means fewer disk accesses and therefore faster performance.

This can be shown using the throughput test of Ziff-Davis's PC Bench version 7.01. This benchmark test performs sequential and random disk accesses of reads and writes for files ranging from 256 Kbytes to 32 Mbytes and reports a disk harmonic score as the result. Using an IBM DPEA 1,080-Mbyte disk drive on an IBM ValuePoint 466DX2 PC with 8 Mbytes of DRAM, we experimented with partitions of various sizes, simulating a single-partition disk drive of various capacities—and thus cluster sizes. Because we used a blank disk, the partition we created for each throughput test always started at about the same location, meaning that the only difference from run to run was the capacity and cluster size. As Figure 7 illustrates, better performance (higher disk harmonics) is achieved for this benchmark if the cluster size—that is, drive capacity—is bigger.

Small files

While applications using large files are helped by drives with a greater capacity, those using small files (relative to the cluster size) may see worse performance. Here, a different dynamic is at work: Because a cluster is the smallest number of sectors that can be allocated to a file, a small file of a few kilobytes will occupy only a small fraction of a large cluster. This has two negative effects on performance:

- As shown in Figure 8, the user's data are more spread out with larger cluster sizes due to the unused sectors, meaning that the files occupy a wider portion of the disk drive. Thus, to move from one file to another requires a longer seek.
- Most disk drives today do look-ahead prefetch into a buffer, allowing quick servicing of sequential data requests. When a file occupies only a small portion of a cluster, prefetch fills the look-

ahead buffer with mostly unusable data, making prefetching less effective.

To test this, we repeated the previous experiment, this time using PC Bench version 9.0. We used the DOS disk mix test, which is a completely different test from the throughput test in version 7.01. This test was run with host system caches of 1 and 2 Mbytes. Unlike the version 7.01 test, file sizes used for this benchmark are not known; however, because this benchmark executes common PC applications, it can be assumed that file sizes are small. As shown in Figure 9, disk scores drop—indicating decreasing performance—as drive capacity and cluster size increase.

Unfortunately, not all advances in recording technology are beneficial to disk drive performance. Sometimes an advance may be beneficial or harmful to performance depending on how or where it is applied. As discussed earlier, increasing areal density without reducing heads can improve performance, but increasing it while reducing heads will lower performance. Sometimes, it also depends on characteristics of the user's workload, or the size of file transfers.

Since cost is always the number one concern, drive makers will continue to find ways to increase areal density—regardless of its impact on performance. They will also aim to increase speed to beyond 10,000 rpm and to reduce seek time and command overhead. It is important for both the user and the disk drive designer to be aware that while some advances may bring benefits in other areas, they don't necessarily improve performance. This fact should be considered when weighing technology options. ❖

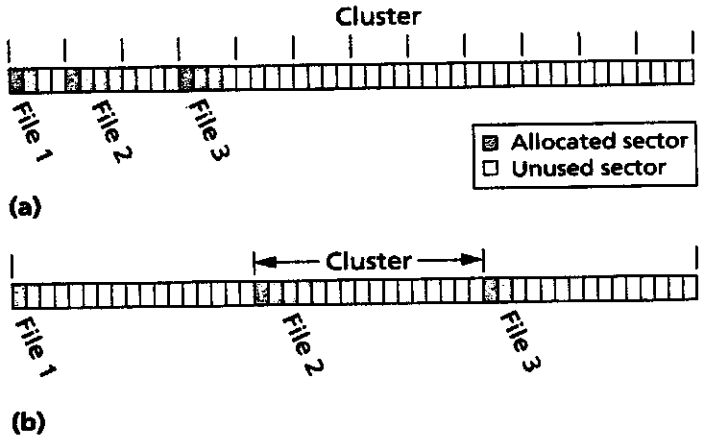

Figure 8. Effect of (a) four-sector cluster size and (b) 16-sector cluster size on small files.

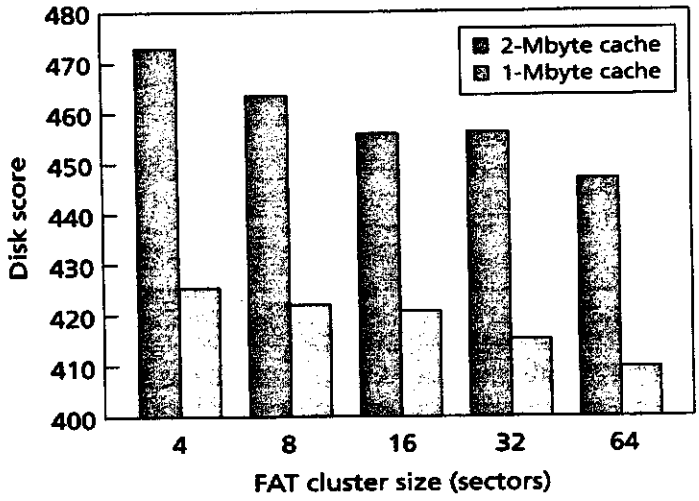

Figure 9. PC Bench 9.0 DOS disk mix test results. Higher disk scores indicate degraded performance.

References

1. E. Grochowski and D.A. Thompson, "Outlook for Maintaining Areal Density Growth in Magnetic Recording," *IEEE Trans. Magnetics*, Nov. 1994, pp. 3,797-3,800.
2. E. Grochowski and R.F. Hoyt, "Future Trends in Hard Disk Drives," *IEEE Trans. Magnetics*, May 1996, pp. 1,850-1,854.
3. C. Ruemmler and J. Wilkes, "An Introduction to Disk Drive Modeling," *Computer*, Mar. 1994, pp. 17-28.
4. J.E. Smith, "Characterizing Computer Performance with a Single Number," *Comm. ACM*, Oct. 1988, pp. 1,202-1,206.
5. W.C. Lynch, "Do Disk Arms Move?" *Performance Evaluation Rev.*, Dec. 1972, pp. 3-16.
6. R.A. Scranton, D.A. Thompson, and D.W. Hunter, "The Access Time Myth," IBM Research Report RC 10197, IBM, Yorktown Heights, N.Y., 1983.
7. K.S. Grimsrud, J.K. Archibald, and B.E. Nelson, "Multiple Prefetch Adaptive Disk Caching," *IEEE Trans. Knowledge and Data Eng.*, Feb. 1993, pp. 88-103.
8. R. Karedla, J.S. Love, and B. Wherry, "Caching Strategies to Improve Disk System Performance," *Computer*, Mar. 1994, pp. 38-46.
9. A.J. Smith, "Disk Cache—Miss Ratio Analysis and Design Considerations," *ACM Trans. Computer Systems*, Aug. 1985, pp. 161-203.
10. L.S. Fan et al., "Magnetic Recording Head Positioning at Very High Track Densities Using a Microactuator-Based, Two-Stage Servo System," *IEEE Trans. Industrial Electronics*, June 1995, pp. 222-233.

Spencer W. Ng is a research staff member at the IBM Almaden Research Center. His research interests include storage architecture and performance, especially for disk drives. He holds a number of patents and has published a dozen papers on storage design. He received a BS in electrical engineering from Washington State University and an MS and a PhD in electrical engineering from the University of Illinois at Urbana.

Contact Ng at IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120; spencer@almaden.ibm.com.