

Elements of Disk Access

This is a brief description of the mechanics that go on behind the scenes whenever you read or write to a disk. It will also give you an insight into why bigger disks are not always better.

Disks are essentially slow devices. This is due to their mechanical nature. The time it takes for a CPU to access a disk is somewhere between 1mSec and 100mSec, depending on the performance of the disk and how lucky you are! Compare this with the sub-microsecond access to main memory and you can see that there is a great incentive to minimize the number of disk accesses your application makes.

There are three basic ways to do this: good software design, using many disks simultaneously and data caching. In practice the first option is in direct conflict with marketing pressures to reduce the development time of a product and

Disk Caches

This involves keeping copies of recently read or written data in memory in the hope that it will soon be requested again. If the data can be kept in an intermediate buffer, the next time it is requested it can be copied rather than retrieved from the disk. This greatly reduces the number of disk access and therefore increases the speed of applications. A modern operating system will be able to get "hit" rates of greater than 90%, so for every 100 logical accesses only 10 physical access will have to be made

generally only gets addressed in "release 2" and then only if the product gets poor reviews. The method of caching frequently accessed data in memory is usually left to the operating system, with a small amount of caching, mainly of entire tracks, being done by the drive itself.

In some cases, such as database products, the application can make use of privileged knowledge of the structure of it's data to do a better job of caching data than the operating system. In which case it makes sense to extend the size of the database cache at the expense of the number of general purpose operating systems buffers.

The final method of speeding up data accesses is to have many disks all making accesses simultaneously – rather than waiting for a single drive to fetch many pieces of data in turn. This is the favoured approach for commercial-sized databases, where data tables, indexes and other specialised data are routinely spread across several devices.

Although the operation of a drive has not changed since it's invention, the performance has improved – although nowhere near as much as every other aspect of computer hardware: disks are definitely being left behind in the performance stakes and are now the primary bottleneck in most data-centric commercial applications.



The diagram above shows the main parts of a disk drive. The two main moving parts are the rotating spindle that carries the platters and the heads that access tracks of data on the platters.

Each disk drive will spin the platters at a constant speed. This speed regulation is one of the reasons why it takes several seconds for a disk to "spin up" after it has been switched on. The second moving part, the head assembly, is used to position the read/write heads over the correct track of the platter to access the required data. The width and therefore the number of tracks (and so, ultimately, the capacity of the disk) is determined by the magnetic properties of the platter and the characteristics of the read/write head. Most of the advances in disk technology have been in improving these areas – which is why disk capacities have risen so much in the past few years.

A step-by-step account of a disk transfer.

When the operating system (whichever one you use, they all do basically the same thing) requests a disk drive to retrieve some data it

will ask for it in multiples of a disk block. This is usually a few KB, 4-16KB being typical. The request is passed to the disk drive which has some in-built intelligence. The disk will decode the request, work out which track and sector (position) within the track contains the data and then set about reading or writing it.

These actions require the heads to be placed over the target track and then for the disk electronics to wait until the sector containing the data (or the sector to be written) to pass under the head. These two operations, which are collectively known as the disks access time, are the most time consuming parts of an I-O operation.

Access Time

An normal model disk drive will spin its platters at 7200 rpm, that's 120 revolutions per second, or one every 8.3 milliseconds. Just from the law of averages, half the time the wanted sector will be in the semi-circle coming up to the disk heads and half the time it will be in the semicircle just passed.

Therefore we can say that waiting for the right sector to appear under the heads will take on average 4.1 milliseconds. The other element of access time is the time taken to move the heads over the track containing the sector. It's much more difficult to work out how long this will take as it depends on the location of the required track relative to where the disk heads are positioned beforehand. If the track is adjacent then the additional time will be (typically) some hundreds of microseconds. After that it gets more interesting.

Although the head assembly is small, it does have some inertia. This means that it has to accelerate and decelerate rather than jump instantaneously to where it's supposed to be. The farther it has to move the longer it takes to get there. However because it accelerates and decelerates, the relationship between the number of tracks crossed and time taken is not linear. So for example, if it took 1mS. to move 10 tracks, it might only take 6 milliseconds to move across 1000 tracks. It can take up to 20mSec for the heads to move from across the whole radius of the disk – say from the innermost track to the outermost.

After the data has been read off the platter, the drive electronics will transfer it back to the processor. This is done across either an IDE, SATA or SCSI bus. The data transfer operation will take place at some 10's of MB per second so will typically take much less than 1 millisecond to complete.

To summarize

CPU sends request to disk drive	10 uSec
Drive decodes request	50 uSec
Drive positions heads over track	(say) 10 mSec
Platter rotates data under the head	4 mSec
Drive reads sector	20 uSec
Transfer back to CPU	50 uSec
Total time for a read	14 mSec

From this you can see that there is a practical limit of about 60-70 random (i.e. needing disk-head movement) operations per second.

The following section will describe how disk manufacturers are improving the speed of their disk drives.

Faster Drives

From the summary table, the two biggest areas for improvement are the time taken to position the disk heads over the correct track and the time for a platter to rotate so the wanted data is under the head. Dealing with the last item first, this can be done by increasing the rotational speed of the drive. (There are other ways, like adding a second set of heads - but this isn't done anymore). A high performance disk drive nowadays will have a rotational speed of maybe 15,000 RPM. This will reduce the average latency for reading a sector from 4 milliseconds to 2 msec. However, the improvement comes with a price. The speed control is more complex, the motor needed to spin the disk (and its power consumption) is larger and even the centrifugal force on the outside of the platter becomes a limiting factor. For these reasons, high-speed disks are more costly and generally only used in enterprise-class products.

The issue of the time take to position the heads is more difficult. Here the weight of the head and its supporting arm means there is a lot of inertia to overcome when starting or stopping the head positioning. In addition to this the thinner width of the tracks, means the positioning has to be more precise. Together these factors add up to a problem. Again this can be overcome but at a price

- this time in the control electronics. Moving the heads around faster means larger actuators (essentially electro-magnets with some control electronics) that consume more power and have to be switched faster and more precisely. This too, adds to the cost of high-performance drives.

In practice, faster head-movements and faster spinning disks mean that a high performance device can execute up to about 130 random I-O operations per second. If your application needs more than that, your options are to spread the data across more physical disks – so 2

are performing operations simultaneously, or to organise your data so that it is not spread across the entire disk.. We saw earlier that the average head-positioning time is the largest component of the whole access time. If we can limit the distance the head has to travel, for example to the outer 25% of the platter, the heads will take less time to get there and so the drive will take less time to access the requested data.. This does reduce the amount of data you can store on a drive, but their low cost and high capacity means that the trade-off is frequently worthwhile.