Hard disks

DVRs are without any doubt a symbiosis of software and hard disk technology. If any of the two fails you would have a failed DVR and loss of important recordings. The need for better understanding of hard disks and their limitations is bigger than ever, especially for us in CCTV.

prepared by Vlado Damjanovski

Introduction

A typical "fully loaded" PC today, with many various applications, would use anything between 2 and 5 giga bytes (GB) of hard disk space. This would typically include the operating system (usually Microsoft Windows) and the various applications such as text editors, spread-sheet programs, web browsers, image editors, etc.

User data, created using the applications, can vary significantly, depending whether you are working with text files only, or text and images, or perhaps, video clips.

Digital Video Recorders (DVRs) used in CCTV are an exception to this typical scenario. They are designed and intended to use the maximum hard disk space available. With a typical large size hard disk available these days of 160GB, the internal DVR hard drive capacity can get extended to 640GB, using four such drives. Some larger systems may even include external SCSI or RAID storage drives. A typical DVR, as used in CCTV, would be working really hard, day and night, 24 hours a day, 7 days a week without (ideally) being shut down.

In our issue 12 (July/August 2001) we discussed the importance (and limitations) of the operating system. It is however, equally important to understand the hardware and their limitations, especially the hard drives.

DVRs are without any doubt a symbiosis of software and hard disk technology. If any of the two fails you would have a failed DVR and loss of important recordings. The need for better understanding of hard disks and their limitations is bigger than ever, especially for us in CCTV. With this in mind, we have prepared this article (the first in a series) which should give you a better in-depth understanding of hard drives.

Spindle speed

Expressed in **revolutions per minute** (rpm), this specification gives a very good indication of the drive performance. Desktop drives generally come in 5,400rpm and 7,200rpm varieties, with 7,200rpm drives averaging 10 percent faster (and 10 to 30 percent more expensive) than 5,400rpm models. High-end 10,000rpm and 15,000rpm hard drives offer only marginally better performance than 7,200rpm drives--and cost much more, in part because they are typically SCSI drives with added reliability features. Also, higher spin hard drives need more current, hence they get **hotter**. **Cooling is very important for all hard drives**, more so for the faster ones. So, for a typical DVR, hard disks with 5,400rpm or 7,200rpm are a good compromise between sufficient speed, reasonable cost and a relatively "cool" hard drive.

Seek time

If two drives have the same spindle speed, you may be able to determine the faster drive by checking the seek time, which measures how long it takes on average for a hard drive's read/write head to find a random track. Small differences in seek times, which range from 3.9 milliseconds (ms) for ultra fast SCSI drives to 12.1ms for slower EIDE drives, may be noticeable in database or search applications where the head scoots all over the platter, but also when doing a VMD or time/date search in a digital recorder footage.

Hard disk performance is primarily determined by the mechanical characteristics.

To read or write data, the disk head must be positioned over the correct track on the rotating media. This is usually referred to as "**seek time**". Seek times are usually quoted to include the time it takes for the head to stop vibrating after the move ("settling time"). Then a delay occurs until the correct data sector rotates under the head ("rotational latency"). Modern disks use accelerated track positioning, so that the head moves faster and faster until about the half-way point and then is decelerated to a stop at the target track. This is why the average seek is only a few times the minimum seek. The maximum seek time is usually about twice the average seek time because the head reaches its maximum speed before the middle track of the disk.

Access Time is equal to the Time to switch heads + Time to seek the data track + Time for sector to rotate under the head + Repeat for next sector.

More heads reduces the need to mechanically seek a new track.

The minimum track-seek time is the time it takes to move the heads from one track to the next adjoining track. For reading large blocks of data, such as our DVR recorded footage, this is the most significant seek performance value. The average track seek time is more important for random access of small amounts of data such as traversing a directory path.

Faster rotational speed increases the maximum data transfer rate and reduces the rotational latency. The rotational latency is the additional delay in seeking a particular data sector while waiting for that sector to come under the read head.

The following table illustrates typical differences between various rotational speed hard drives and their maximum transfer rates (discussed later in this text), which are the most important indicator of how much data we can put through the magnetic plates of the hard disks.

Rotational Speed	Rotational Latency	Maximum Transfer Rate
3600 rpm	16.7 ms	60 MBps
4500 rpm	13.3 ms	80 MBps
5400 rpm	11.1 ms	100 MBps
7200 rpm	8.3 ms	140 MBps
10,000 rpm*	6.0 ms	200 MBps
12,000 rpm*	5.0 ms	250 MBps
15,000 rpm*	4.0 ms	300 MBps

*The higher speeds require better cooling of the drive.

Cache

This is the **amount of memory built into the drive.**

Designed to reduce disk reads, the cache holds a combination of the data most recently and most frequently read from disk. Large caches tend to produce greater performance benefits when multiple users access the same drive at once. Although small differences in cache size may have little bearing on performance, a cache smaller than 2MB may be a sign of an older, slower drive.

Operating systems try to maximise performance by minimising the effect of mechanical performance. Keeping the most recently used data in memory reduces the need to go to the disk drive, move the disk heads, etc. Writing of new data may also be cached and written to disk at a later, more efficient time. Other strategies include track buffering where data sectors are read into memory while waiting for the correct sector to rotate under the head. This can eliminate the delay of rotational latency because later sectors have already been read after reading of the sought sector is complete. For modern disks, this track buffering is usually handled by a memory cache on the disk drive's built in controller.

Modern disk drives usually have cache that ranges from 512kB to 4MB of cache memory to buffer track reads and hence eliminate rotational latency. Some high-end SCSI drives have 8MB or even 16MB. However the rotational speed still limits the maximum transfer rate.

MTBF (Mean Time Between Failure)

A majority of hard disk manufacturers quote **MTBF (Mean Time Between Failure)** numbers for their hard drives. Typical hard disk MTBF numbers are anything between 300,000 and 1,000,000 hours. This is equivalent to 30 to 100 years. Although these numbers are more **theoretical** rather than a guarantee based on a statistical experience (technology doesn't allow hard drives to be used for more than a couple of years), they offer an important indicator about the hard disk life quality and life expectancy.

Practice has shown that hard drives do fail sooner than their MTBF, and some of the main reasons (apart from the quality of manufacture) are: **physical mistreatment (shocks and vibrations), temperature (insufficient cooling) and dust.**

The MTBF is based on a simple exponential distribution of failure (FailureProbability = 1 - exp(-Time/MTBF)).

So for a 500,000 hour MTBF

drive, 1% will fail in 7 months, 5% in 3 years, 10% in 6 years, and 50% in 40 years.

ATA and SCSI

The type of connection between the hard drive and the system is defined by one of a few standards.

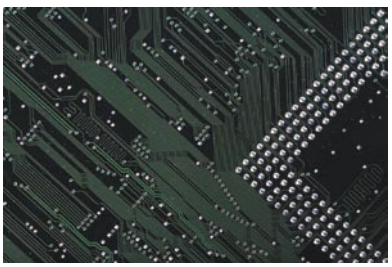
Most popular are the **EIDE (Enhanced Integrated Drive Electronics)** and the **SCSI (Small Computer System Interface).**

Drives with EIDE interfaces dominate, with every desktop PC offering built-in EIDE connectors. They are also known as **ATA** (Advanced Technology Attachment) interface.

Most modern PCs can talk to up to 4 EIDE drives without any additional hardware. This is because the EIDE controller is usually embedded in the motherboards. Although this could also be the case with the SCSI controllers, it is not so frequent, especially in the last few years when the speed of ATA drives has come closer to SCSI.

This is why most DVRs in CCTV can have up to 4 internal hard drives.

Typically, only servers or large storage machines use SCSI drives, which cost much more and require an interface card.



SCSI is designed to "talk to" more than 4 drives (usually up to 8 or 16 devices) and this is one more reason for using SCSI with larger storage capacity machines, although it is a costly option and not so common.

There are a few generations of SCSI standard, usually the latest having the fastest transfer speed.

Current EIDE drives generally conform to the ATA/100 specification (also known as Ultra ATA/100, Ultra DMA/100, and Feature ATA). The 100 in ATA/100 indicates that up to 100MB (mega bytes!) per second can transfer from the drive to the system in short bursts. This is also known as burst transfer rate.

The prevailing SCSI specifications, Ultra160 and Ultra 320, support faster, 160MBps and 320MBps (respectively) burst transfer rates.

RAID

The basic idea of **RAID (Redundant Arrays of Inexpensive Disks)** was to combine multiple small, inexpensive disk drives into an array of disk drives which yields performance exceeding that of a Single Large Expensive Drive (SLED).

Additionally, this array of drives appears to the computer as a **single logical storage unit** or drive.

The Mean Time Between Failure (MTBF) of the array will be equal to the MTBF of an individual drive, divided by the number of drives in the array. Because of this, the MTBF of an array of drives would be too low for many application requirements. However, disk arrays can be made **fault-tolerant** by redundantly storing information in various ways.

Five types of array architectures, RAID-1 through RAID-5, were defined by the Berkeley paper, each providing disk fault-tolerance and each offering different trade-offs in features and performance. In addition to these five redundant array architectures, it has become popular to refer to a non-redundant array of disk drives as a RAID-0 array.

The following is a summary of the five different RAID versions:

* **RAID-0** is the fastest and most efficient array type but offers no fault-tolerance.

* **RAID-1** is the array of choice for performance-critical, fault-tolerant environments. In addition, RAID-1 is the only choice for fault-tolerance if no more than two drives are desired.

* **RAID-2** is seldom used today since ECC is embedded in almost all modern disk drives.

* **RAID-3** can be used in data intensive or single-user environments which access long sequential records to speed up data transfer. However, RAID-3 does not allow multiple I/O operations to be overlapped and requires synchronised-spindle drives in order to avoid performance degradation with short records.

* **RAID-4** offers no advantages over RAID-5 and does not support multiple simultaneous write operations.

* **RAID-5** is the best choice in multi-user environments which are not write performance sensitive. However, at least three, and more typically five drives are required for RAID-5 arrays.

Data transfer rates

The **burst transfer rate** is a relatively meaningless number. It refers to the top speed at which data can be transferred between the hard drive's cache memory and the system (interface specifications such as ATA/100 indicate the external transfer rate).

The internal transfer rate, also termed the **sustained transfer rate**, tells more about the speed of the drive. Generally ranging from

Various data transfer standards

Standard	Also Known As	Maximum Transfer (MB/s)	Maximum Devices	Status
ATA	IDE	6.0	2	Obsolete
Fast ATA	Fast IDE	11.1	2	Obsolete
ATA-2	**Enhanced IDE**	**16.6**	**2**	**Legacy**
ATA-3	**Enhanced IDE**	**16.6**	**2**	**Legacy**
ATA-4	**Ultra DMA/33**	**33.0**	**4**	**Current**
ATA-4	**Ultra ATA/66**	**66.0**	**4**	**Current**
ATA-4	**Ultra ATA/100**	**100.0**	**4**	**Current**
SCSI-1	SCSI-1	5	8	Obsolete
SCSI-2	Fast SCSI	10	8	Obsolete
SCSI-2	Fast and Wide	20	16	Obsolete
SCSI-3	**Ultra SCSI**	**20**	**8**	**Legacy**
SCSI-3	**Wide Ultra**	**40**	**15**	**Legacy**
SCSI-3	**Ultra-2**	**40**	**8**	**Current**
SCSI-3	**Wide Ultra-2**	**80**	**15**	**Current**
SCSI-3	**Ultra-3**	**80**	**8**	**Current**
SCSI-3	**Wide Ultra-3**	**160**	**15**	**Current**
SCSI-3	**Ultra 160**	**160**	**15**	**Current**
SCSI-4	**Ultra 320**	**320**	**15**	**Current**
FC-AL	**Fiber Channel**	**100.0+**	**126**	**Current**
USB		**1**	**127**	**Current**
USB-2		**60**	**127**	**Current**
IEEE1394	**Fire-Wire**	**25.0**	**63**	**Current**
IEEE1394b	***Fire-Wire B***	***400.0***	***?***	***Future***

14MBps to 62MBps (**mega bytes per second!**), it indicates how fast data can be read from the outermost track of a hard drive's platter into the cache. A difference of a few megabytes on the high end of that range won't be perceptible, but specs on the low end may indicate a slow drive, particularly for such demanding applications as video editing or continuous recording of multiple cameras such as in video surveillance.

The sustained transfer rate is an important parameter of the DVRs' hard drives, which ultimately defines the upper limit of how many pictures per second your system can record and play-back.

Certainly, this performance also depends on the operating system, the processor and the compression speed, file sizes etc., but ultimately, if the hard drive can not cope with such a through-output the DVR can not achieve what it is (theoretically) capable of.

Very often, if the DVR processor is fast and the compression is done by hardware compression chips, the "**bottle-neck**" in a DVR's performance **could be the hard disk sustained transfer rate.**

Let's analyse this with one **practical example:**

Let's be conservative and assume that we have a typical and "not-so-good" hard disk with a sustained transfer rate of only 14 MBps. If we translate this rate into mega bits per second we should multiply 14 by 8, and we get 112Mbps transfer rate. Let's now assume that we are recording on a DVR with JPG compression that records good quality images of, lets say, 40kB. If we don't do anything else while the recorder records (i.e. not playing back), to find out what the maximum (theoretical) recording performance of such a machine is we need to divide 14MBps by 40kB. This gives a number of 350. If we have 16 cameras connected to the DVR, then the theoretical maximum recording rate would be 350/16=21 pictures per second per camera. This is a theoretical maximum of a DVR where no other processes are active. In a real situation, the DVR has to "spend time" doing time base correction, i.e. synchronising the un-synchronised cameras. This will reduce the theoretical rate by at least 50% to 10 pictures per second per camera. If we decide to play-back at the same time, or archive, this would further reduce the recording performance by another factor of at least 50%, obtaining 5 pictures per second per camera as a theoretical maximum with such a hard drive. In addition to this, the intelligence of the operating system of handling files has to be considered as well. And this is all true if we assume that our example DVR does hardware JPEG compression at a faster rate, hence not "wasting" the operating system and main processor's time. In many practical DVRs you will find that the compression might be done by the "proprietary software encoding scheme", which practically means that there will be some additional "bottlenecks" for our theoretical recording performance, which most likely would drop down from the above calculated 5 pictures per second to maybe 1 or 2 pictures per second per camera.

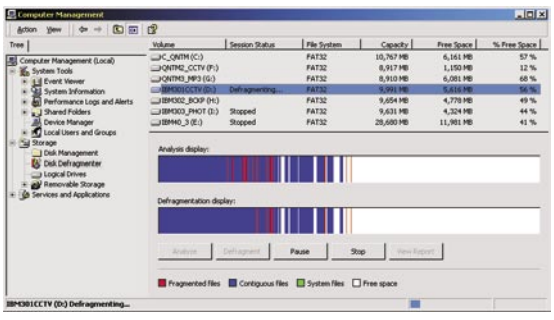
But there is one more important, and almost invisible factor that we need to consider in this example.

This is the point where we have to acknowledge that fragmenting of files while performing continuous recording (24 hours a day, 7 days a week), is handled by the operating system (read - "Windows") and can affect the recording performance, especially after a longer period of recording of a few days or weeks. No number can be attached to this performance reduction factor as it depends on the DVR software design but it is certainly going to reduce the performance further although the actual hard disk sustained transfer rate might be unaffected (the hard disk "wastes time" searching for the free fragments as dictated by the operating system).

As can be concluded from the above example, there are many factors influencing the performance of a digital video recorder, not just the "DVR front end", but other underlying, and invisible processes. The hard drives are the starting and the ending point in such a chain of operation.

File fragmentation

The Operating Systems **organise physical disk sectors into files** using data structures that may combine contiguous sectors into clusters which are the minimum amount of disk space that the system can allocate to a file.



Usually the system will attempt to allocate contiguous clusters to each file.

This significantly improves performance by minimising the amount of track positioning that the disk must do to read the file. Over time however, **writing and deleting of files, fragments the allocation of clusters**.

Since this slows down the system over time, periodic use of a de-fragmentation utility is recommended (after a complete backup!).

Before you use your brand new hard drive you need to format it. Formatting prepares the hard drive for use with your operating system. It is similar to having a blank sheet of writing paper, without lines. We have an agreed method of writing and reading – horizontal, from the top left hand corner to the right and then down one line, and so on – this would be in computer terms the filing system.

In order to use the space on the sheets of paper efficiently we decide to draw horizontal lines.

The alphabet, the language and the sentences with their meaning when writing on such a sheet of paper would be equivalent to the operating system in a PC.

The multiple sheets of paper would make a notebook which is in computer terms equivalent to a hard disk with its magnetic platters.

Your disk drive is formatted into sectors, clusters (a group of sectors) and tracks according to the operating system you are using.

Formatting is a method of organising what is saved to the disk.

A **File Allocation Table (FAT)** on your hard drive keeps track of the addresses of your files within the clusters, sectors, and tracks. It is like an Index in your book. Without the File Allocation Table (FAT) your computer would not be able to retrieve any information from the hard drive. A file is stored on a disk drive (and other media) in one or more clusters. Clusters are the atomic unit of data allocation, made up of

one or more sectors. Sectors, in turn, are physical storage units.

As a file is written to the disk, the file may not be written in contiguous clusters.

Non-contiguous clusters slow down the process of reading and writing the file. The further apart on the disk the non-contiguous clusters are, the worse the problem because of the time it takes to move the hard drive's read/write head. A file with non-contiguous clusters is said to be fragmented. To optimise files for fast access, a volume may be defragmented.

Defragmentation is the process of moving portions of files around on the disk in order to defragment files; that is, the process of moving a file's clusters on the disk to make them contiguous.

When you save a file, the system begins in the first cluster of the first sector on the first track and writes the file in contiguous addresses until the file has been completely saved.

As you write, edit and delete information, the computer searches the FAT for unused clusters, sectors and tracks. If your file is larger than the number of bytes available in a cluster, the system will continue to look to the FAT table for available clusters. **The clusters do not need to be in contiguous blocks to store your file.** The FAT efficiently records the addresses of the clusters where the file has been stored.

In a simple single-tasking operating system, defragmentation is straightforward: the defragmentation software is the sole task, and there are no other processes to read from or write to the disk. However, in a multitasking operating system, some processes may be reading from and writing to the hard drive while another process is trying to defragment that hard drive. The trick is to avoid writing to the file being defragmented without stopping the writing process for very long. Solving this problem is not trivial, but it is possible.

Some file systems are publicly documented, such as the FAT16 and FAT32 file systems (used by Microsoft operating systems). This allows programmers to manipulate on-disk data structures (such as file allocation tables, or FAT) directly. However, NTFS (used by Windows NT/2000/XP) is deliberately opaque.

Non contiguous placement of blocks in a file is bad for performance, since files are often accessed in a sequential manner. It forces the operating system to split a disk access and the disk to move the head. This is called "external fragmentation" or simply "fragmentation" and is a common problem with Microsoft's file systems. Windows users are accustomed to defragging their disks every few weeks and some have even developed some ritualistic beliefs regarding defragmentation.

Disk fragmentation in the Windows file systems, whether it be the traditional FAT file system of Windows 3/95A, the FAT32 file system of Windows 95B/98, or the NTFS file system of Windows NT/2000/XP, causes significant performance degradation.

Executive Software is a Microsoft partner and the company that wrote the defragmentation code that's included in Windows. Here's what they have to say about Windows fragmentation.

Disk fragmentation cuts directly across the integrity of your system. Files fragmented into 200 pieces take 200 times longer to access. Files shattered into 200,000 pieces will take 200,000 times longer, and so on... And, that's just one computer and one file! The mathematics of a day's work on an entire network is staggering.

This is exactly the scenario when using Digital Video Recorders in CCTV.

DVRs in CCTV are continuously writing and reading data.

Many DVR manufacturers have even included a scandisk and a defragmenting option in their DVR applications, making the machine inoperable for unpredictable amount of time.

Depending on the size of your drive, the defragmenting process may take several hours. This is even more critical in CCTV where hard drives used in the DVRs are usually of a larger capacity. The size of your drive, the amount of activity on your computer, and the length of time since it was last defragmented will all determine how long the process will take.

The Unix/Linux concept of handling files is different.

Linux native file systems (**ext2** and as of recently **ext3**) **do not need defragmentation** under normal use.

Linux's Ext2 does not force you to choose large blocks for large file systems, except for very large file systems in the 0.5 TB range (that's terabytes with 1 TB equalling 1024 GB) and above, where small block sizes become inefficient. So unlike Windows there is no need to split up large disks into multiple partitions to keep block size down in Linux.

The Second Extended File system (its abbreviated form is Ext2FS or simply ext2) has been GNU/Linux's default file system for many years. This file system fragments very little because of the way the file system is designed.

Fragmentation on a typical ext2 disk is **usually between zero and three percent no matter how much file system activity occurs.**

By default, the Linux swap area on disk is on its own disk partition and does not affect normal files.

Ext2FS respects the usual standards for Unix-type file systems. Since its conception, it was destined to evolve, while still offering a great robustness and good performances.

The Ext3 is the Third Extended File System, and is the Ext2FS' successor. It is compatible with the latter but it is enhanced by a very interesting feature: journaling.

One of the major flaws of Windows and "traditional" Unix/Linux file systems like Ext2FS is their **low tolerance to abrupt system breakdowns** (power failure or crashing software). Generally speaking, such events involve a very long exam of the file system's structure, attempts to correct errors, sometimes resulting in an extended corruption. Hence, a partial or total lost of saved data.

Journaling answers this problem. To simplify, let's say that the object is to save actions (such as the saving of a file) before really doing it. We could compare its functioning to the one of a boat captain who notes, in his log book, daily events. The result: an **always coherent file system**. And if problems occur, the verification is very rapid and the eventual repairs very limited.

The time spent to verify a file system is thus proportional to its actual use and not to its size.

Hence, Ext3FS offers the journal file system technology, while keeping Ext2FS' structure ensuring an excellent compatibility.

With the release of the new kernel last year, Linux changed the default file system from the venerable ext2 format to the journaling ext3 file system.

The ext3 file system is essentially an enhanced version of ext2 file system, offering total protection from loss of data even when power failure occurs.

These improvements provide further advantages of Linux over Windows.