



easy access

MEASURING PC HARD DRIVE AUDIO RECORDING PERFORMANCE

What's the best way to set up your PC's hard drive for digital audio recording? Useful statistics can be hard to come by, so **Martin Walker** runs his own tests.



Once you move beyond an entry-level PC (as I just have -- see the box on page 100), it becomes even more important to optimise your system if you are to achieve its maximum potential. You may have the fastest SCSI drive in the world, but this doesn't automatically ensure a blistering MIDI + Audio performance. This depends also on both operating system and audio software settings, as well as setting up the hardware correctly.

When measuring hard disk performance for audio purposes, it is the sustained data transfer rate of the drive that is important, but manufacturers rarely give this. Access Time and Spin Speed are often quoted, but frequently only a burst transfer rate is given (which of course is a lot faster than the sustained rate). Even when you do have such figures, however, it can be tricky to work out how many audio tracks your system is capable of giving you, because many other things affect this number. Any of you with a SCSI drive and a copy of Adaptec's *EZ-SCSI* will have the *SCSIbench* utility (see opposite), which allows you to select Same Sector, Sequential, and Random reads during drive speed tests. As you might expect, continually reading the same sector from any hard drive will give far higher results than when moving the read/write heads as you go (Sequential), and the slowest speed will be when leaping about all over the drive (as you would if it had a lot of fragmentation, for instance).

To give you an example, after I had set up tests for a 64K transfer size, the Ultra Wide SCSI drive in my new machine gave a reading of about 15Mb/sec for Sequential reads (this is the most useful figure). For Same Sector I/O, however, this leapt to about 31Mb/second, and for Random I/O it plummeted to about 4Mb/second, due to the large amount of read head movement in between each actual read. The random figure also varies hugely as you change the transfer size, since this determines how often the heads have to move to a different location, and therefore the proportion of the total time spent actually reading information. This, as we will see later, is the clue to the performance of hard drives in audio applications. Before that, though, let's consider three factors which contribute to hard drive performance: your choice of FAT (File Allocation Table), your PC's

"You may have the fastest SCSI drive in the world, but this doesn't ensure a blistering MIDI +

internal buss speeds, and whether or not buss mastering is used.

Audio performance."

A FAT Chance

Those of you with either Windows 98 or the most recent Windows 95B (OSR2) release have the option of formatting your drives with either the FAT16 (File Allocation Table) or FAT32 file systems. To find out which version of Windows 95 you have, open Control Panel/System. On the General page, under Microsoft Windows 95, you will either see 4.00.950 (the original release), 4.00.950a (with the service pack update), or 4.00.950B (for OSR2). PCs supplied with Windows 95B or Windows 98 will probably already have their hard drives formatted with the newer FAT32 system, which has the advantage of using the space available on a large drive in a more efficient way, by using much smaller cluster sizes (normally 4Kb). The cluster is the smallest unit of storage available, and a 1-byte file will still occupy a single cluster.

Divide And Conquer

One factor that surprises many people is that the sustained data transfer rate varies across the surface of the disk -- it is fastest at the outside edge, and slowest at the inside. This has several knock-on effects. Firstly, since drives are filled from the outside in, your transfer rate will reduce as more and more files have been written to the drive. In other words, an almost-full drive will have slower performance than an empty one. Secondly, if you create several partitions on a single drive, the first will be faster than the second, and so on. Finally, bear this in mind when running speed tests. If it's several weeks or months since the last time you ran them, don't worry if your drive seems a bit slower -- it's not wearing out, but probably a bit fuller than before!

To give you some idea of the amount of variation, I reformatted a 2.5Gb Maxtor Diamond Max EIDE drive using FAT16, measured its sustained transfer rate, and then split it into two partitions -- one of 500Mb, and the other of 2Gb -- and then measured again on each. The 500Mb measured identically (as you would expect), but the inner 2Gb one was noticeably slower. I then reformatted twice more, so that I could measure a partition starting half way across the drive (by splitting it into two equal 1.25Gb partitions), and finally near the inside (by creating a 2Gb outer partition, and a 0.5Gb inner one). The results can be seen in the graph below, and they show that on my drive, the sustained transfer rate has dropped by about 10% in the middle of the drive, and by about 30% towards the inside.

The moral of all this? If you want to split your drive into several partitions, the first tends to be for the operating system, so make this small. In my case, using a 500Mb partition ensures that the start of the main 2Gb audio partition is only

By contrast, with the worst case of a partition of 1 to 2Gb in size, a FAT16-formatted drive would use a single 32K cluster to store this 1-byte file. Over the contents of a typical drive, this wastage can result in dozens of megabytes of extra unusable space. FAT32 also overcomes the annoying 2Gb maximum size available to FAT16 partitions, which forces you to divide up drives larger than 2Gb whether you want to or not.

So, opting to use FAT32 will typically result in more drive space being available, as well as the option of using partitions greater than 2Gb in size. Sadly, few things in life are free, and this is at the expense of a slightly larger overhead during file reads and writes, due to the more complicated directory structure with lots more potential entries. For this reason, many PC Musicians have carried on with FAT16. However, you can force larger cluster sizes with FAT32, either using a command line when reformatting your drive from DOS, or far more elegantly with the third-party application *Partition Magic*. The benefits for huge audio files are that larger clusters mean fewer read/write operations, and you are also likely to get less fragmentation to take care of.

There have been many mentions of the pros and cons of using FAT32 on the Net, but many of these have been anecdotal, so in the interests of proper scientific research, I carried out some tests on an empty drive with my new machine to see what I could establish. Assuming that we keep our drives well defragmented, the main FAT overhead

about 3% slower. It also suggests that if you partition your drive into three areas, they will give best audio performance when arranged as operating system C:, audio data D:, and other data E:.

is likely to be the number of read/write operations carried out during the course of a read or write. After some hours of testing, I came to the conclusion that the problems have been exaggerated. I measured absolutely identical performance when reading multi-channel audio after formatting my drive in turn as FAT16 with 32K clusters, FAT16 with 8K clusters, and then FAT32 with 4K clusters (see graph, right). This may be because I was using Windows 98, which may have further optimised file operations, or it may be that my Pentium II 300MHz machine minimises the different overheads, which might show up more on a slower machine. The write speed did vary slightly, but before you race off to reformat your drives, notice that the difference only works out to 1.6% (hardly worth bothering about).

To check that Windows 98 was not the saviour, I ran two more sets of tests with Windows 95, and Windows 95B (this time both with no buss master support, since this is not available for the original Windows 95 version -- see later). Again, there were no worthwhile differences -- although there were repeatable differences in read and write speeds, they amounted to less than one per cent. I think we can safely assume that in terms of file read and write speeds, the choice of operating system seems less important than the choice of FAT format type, which itself seems pretty minimal as long as you have a fast machine. I suspect that the FAT16/32 arguments might change if a huge number of small files were being accessed, but hard disk recording tends to open only as many files as there are audio tracks, and these are all large. It still makes sense to use larger cluster sizes if possible, as there is bound to be a slightly greater overhead on a cluttered drive, but on the basis of my results I have formatted my audio drives with FAT32 and 16K clusters.

"Don't expect disk speed utilities to give exactly the same figures every time. There are so many processes going on in any computer that there will be slight variation every time you run the test."

Buss Times

One widely misunderstood area of hard disk transfers concerns the data transfer rates relating to different busses inside the computer. A Fast SCSI buss can move data at 10Mb/S, a Wide one at 20Mb/S, and an Ultra Wide one at 40Mb/S, while the latest Ultra DMA EIDE transfers can take place at 33Mb/S. However, this doesn't make any of them inherently better or worse for data transfer, since whichever buss happens to be carrying the data, the limiting factor will normally be the sustained transfer rate of your hard drive, which will probably be somewhere between 5 and 15Mb/second.

Let's say you have a 10,000rpm Ultra Wide SCSI drive, which provides a blistering 15Mb/second capability. The fact that the Ultra Wide SCSI buss offers a maximum speed of 40Mb/second won't make your single drive go any faster, but you will normally be able to run several drives simultaneously on the same buss, which can be used by RAID arrays (see 'Spreading The Load' in the December '97 PC Musician for more details). If you have a Fast SCSI drive, upgrading to an Ultra Wide host adaptor card won't make it go any faster - again, the limiting factor is the drive itself.

Similarly, the fact that the latest Ultra DMA EIDE drives have a maximum transfer rate of 33Mb/second is largely irrelevant for our purposes. Sometimes the maximum burst speed

of a drive can momentarily reach much higher values than its sustained rate, but for hard disk recording it is always the sustained figure that is important, since we rely on this to keep a steady flow of audio data to and from the soundcard.

In fact, an occasional cache-boosted surge for a fraction of a second can cause other problems. Since the SCSI host adaptor card is plugged into the PCI buss, if transfer rates attempt to shoot up to the maximum 40Mb/second, then the SCSI host adaptor can block the PCI buss (the maximum rate of which is only 33Mb/sec). If you are using a PCI buss mastering soundcard (such as those from Event, or the Korg 1212), you may experience occasional clicks and glitches, due to there temporarily being no PCI bandwidth left for the soundcard to operate. For this reason, Steinberg recommend reducing the maximum transfer rate of the SCSI buss to a lower setting like 10Mb/second (from the SCSI BIOS). Even 10Mb/second is sufficient to achieve the maximum 32 simultaneous tracks offered by VST. This is no doubt also the origin of the Buss Throttling tweak for S3 graphics cards, since PCI graphics cards can also grab the entire bandwidth of the PCI buss unless told otherwise.

Buss Mastering

There is little point in repeating the EIDE-versus-SCSI argument here, except to say that although the latest EIDE drives are extremely fast, the fastest (and the most expensive) drives still tend to be the SCSI models, and in particular those running at 10,000rpm. However, a fundamental factor with both types is that drives will take a large chunk of your available CPU power unless buss mastering is being used. SCSI host adaptor cards are available with and without buss mastering -- in the Adaptec range, for instance, this is the reason why the 2940 model (in its various incarnations) is the most popular.

However, for EIDE drives, most modern PCs have this facility built in. The first motherboard chipset to offer this facility appeared in late 1995, after Windows 95 first appeared, and so the original Windows 95 release did not have buss master drivers. Windows 95B (OSR2) arrived with Microsoft buss master drivers, but by then the newer TX chipset offered a new and improved Ultra DMA Mode, and motherboard manufacturers are still supplying special Intel-written drivers for best performance. These will normally be already installed if you buy a new PC that benefits from them, and most new motherboards will arrive with a floppy disk containing the relevant drivers.

As you might expect, Windows 98 includes the latest buss master drivers, and automatically installs them. Mind you, buss master operation is not enabled by default, and it is vital that you check that your drives are running in buss master mode if they support it (most modern ones do). Buss master support needs to be enabled for each EIDE drive (and any non-SCSI CD-ROM drives), and you do this from the System section of the Control Panel. Under 'Disk drives', click on the Properties button for each EIDE drive, select the Settings tab, and then make sure that the DMA box is ticked. If you don't see this box, it may be because this facility is not available on your PC, or that you have already installed specific hard disk drivers from your hard drive manufacturer. You will need to restart your PC for the changes to take effect, but the results will be well worth the wait.

If you buy a complete system, buss mastering should already be set up on your machine. Since I decided to personally install all the software on my new machine, however, the first time I ran the *Dskbench* utility (see later) it showed reasonable EIDE hard drive speeds, but colossal CPU overheads. With my Fujitsu MPB3021AT drive, Sustained Transfer Rate measured 6.3Mb/second and took 98.8% CPU time before activating buss master DMA; directly afterwards it measured 8.8Mb/second, and took 1.4% CPU time! By comparison, my SCSI drive measured about 2% using an Adaptec 2940 host adaptor card. Although its

read speed was a sustained 15Mb/second, however, write speed initially measured slightly under 6Mb/second. This turned out to be because SCSI drives are normally shipped with their read cache enabled, but not the write cache. Using *EZ-SCSI's SCSI Interrogator* utility, I enabled the write cache, and the write speed immediately jumped up to 14.5Mb/second.

If you have the original release of Windows 95, the lack of buss mastering won't stop you running any real-time plug-ins on the EIDE drives due to lack of processor time, but it will limit the maximum number of audio tracks that you can run alongside plug-ins. Also, whatever the number of audio tracks you are running, there will be significantly less processor time left over to run plug-ins.

Having got everything hunky-dory with my Windows 98 installation, I temporarily rebooted under Windows 95B, and re-ran my tests. As expected, buss mastering was not enabled, so I ran the *Bmide_95.exe* file supplied on a floppy disk with my new motherboard, and then rebooted. As expected, CPU overhead plummeted, but still seemed unusually high at about 30%, and although read speeds improved, the write speed had dropped significantly compared with its pre-buss master value. I double-checked by running *Echo Reporter* (of which more later) on the same drive using both Windows 98 and Windows 95B, and this still showed read speeds roughly the same, but write speeds lower with Windows 95B. This may be an anomaly with my system, and I will report back if I resolve it.

Speed Utilities

There are several types of disk speed utility available, and many people have one included as part of a utility suite such as *Norton Utilities* or *Nuts & Bolts*. However, while these give a fairly repeatable figure for general purposes, they are not suitable for testing out the effects of hard disk audio tweaks, since their test file sizes are unlikely to be large enough to defeat any caching systems in place. What we need is a utility specifically designed to measure a hard drive when being used like a typical hard disk recording system. To do this you need multi-file reads to simulate the way a multitrack audio application has to open and read various large files (one for each running track). The most accurate results will be obtained from these utilities if you make sure that no other

Putting My Money Where My Mouth Is

As I reported in last month's PC Notes, I have just bought a new, faster and more powerful PC, complete with a 300MHz Pentium II processor, and a separate SCSI hard drive specifically for hard disk audio recording specifically chosen for its low acoustic noise. When buying a system, everybody's needs will be different, but it is worth briefly explaining my rationale. I chose a 300MHz Pentium II as the best value for money at the time of purchase, but although this works with a 66MHz front-side buss (see September's PC Notes), I specified a motherboard with one of the new 440BX chipsets which supports the new 100MHz buss speed. I also requested 64Mb of 100MHz-capable SDRAM, with the result that as and when future requirements and funds permit, I can simply upgrade to a 450MHz Pentium II processor, without having to change any other components.

My hard drive requirements are fairly modest (I'm ruthless about purging unwanted software, and don't indulge in games), and so chose a 2Gb EIDE drive for installing my operating system and applications. I chose Windows 98 to research last month's PC Musician feature, but also transferred across the 2.5Gb EIDE hard drive from my previous PC, and installed Windows 95 on this. Rather than use dual-booting software, I find the easiest way to swap between them is to enter the BIOS during bootup, and change the entry for my Windows 98 C: drive to 'None'. Then my second drive automatically becomes the C: drive, and boots up into Windows 95 instead. I intend using this for compatibility testing, as well as for reviewing software and hardware that is only on my machine a short time, so that it can be purged regularly. This may seem a bit of a waste of a 2.5Gb drive, until I explain that I've partitioned it as 0.5Gb (500Mb) for the Windows 95 operating system, leaving me with another 2Gb partition for more general storage.

My main audio drive is a Fujitsu MAC3045, an Ultra Wide 4.5Gb SCSI device that spins at 10,000rpm but still remains blissfully quiet compared to many others. This may not seem very large for audio, but still provides enough space for 106 minutes of continuous 8-track recording, or 53 minutes of 16-track -- fine for my

applications are being run at the same time. purposes.

Probably the most easily available disk test program that fits this description is the *Echo Reporter* from Event, which is freely downloadable from their web site (www.event1.com). I have downloaded this a couple of times in the last year, and it has changed slightly during that time. The latest version, 2.01, has a more thorough system analysis (the previous one couldn't examine the IRQs on some machines), and the transfer file size has increased from 32Mb to the current 128Mb. This does mean that the latest version takes four times as long to run the disk speed test, but it should make the results more accurate and repeatable.

Another one that I've found recently is *DskBench*, which can be found in the software section of www.prorec.com. It's basic and unpretentious, and needs to be run from the drive you wish to test (for multiple drives, simply copy it across to each one -- it only takes up 41K). Both this and *Echo Reporter* set up eight files, long enough to defeat any caching in place (128Mb in the case of the *Echo* utility, and 16Mb for *DskBench*). However, *DskBench* has an added feature in measuring sustained transfer rate for a single file, using a huge 256Mb file read sequentially. It also carries out its multi-file tests using various block sizes varying between 128K and 4K (see later), and measures percentage CPU overhead. The downside of this is that the tests take an age to run (the first time I ran it I thought my PC had crashed), but I found its results very repeatable, which made it easy to see the result of any adjustments. If you want to terminate the tests before they get right down to the 4K block size, you can safely use the 'three-fingered salute' (Ctrl-Alt-Delete keys) and then End Task.

Repeatability

The first few times you run any utilities like this, you will probably be disappointed at the low transfer rates measured. The reason for this is that we are not measuring maximum burst rates, and not even the maximum sustained transfer rate, but more realistic figures based on reading and writing multiple audio tracks. However, any form of sustained transfer rate can only be measured accurately if file sizes are large, so that whatever cache system is currently being used, it will soon be emptied, and the remainder of the data transfer will be directly from the drive itself, rather than from some high-speed memory buffer. The only disadvantage of this is that due to the large file sizes, tests will take longer to run (at least a few minutes), but this is the only way that you will be able to see whether the operating system tweak you have made has had the desired effect.

However, when you are using disk speed utilities, don't expect them to give exactly the same figures every time. There are so many processes going on in any computer that there will be slight variation (probably of the order of a few percent) every time you run the test. Of course, where hard disks are concerned, a heavily fragmented drive will be significantly slower than a freshly defragmented one, so you can ensure more consistent results by running a 'defragger' utility before carrying any tests, to keep the playing field as level as possible. In fact, if you are planning to try out a selection of suggested tweaks, it makes sense to do them all in the space of a few hours. If you start by fully defragmenting your drive, and then carry out the adjustments one after the other, you will minimise the chances of anything else changing in the meantime, and then you should see the results of your tweaks more easily.

Splitting The Load

I've already mentioned that to simulate using a hard drive for multitrack audio, utility

programs need to open multiple files and read them using streaming. This simply means that a small chunk of each file is read in turn and stuck in a set of RAM buffers, which then hold enough audio data to keep the soundcard going until the next batch of reads occur. As long as the buffers are big enough to ensure that the file reads stay ahead, audio playback should never stutter or glitch.

Now we finally come to the big difference between those impressive sustained transfer rates, and the reality of multitrack audio. Each time the drive read heads move from one track chunk to the next it takes some extra seek time, and this appreciably increases the total time taken. The amount of extra time taken depends on how often the next set of chunks is required, and this is directly related to the size of the chunks we use. Anyone who uses *Cubase* may already get the feeling that they know what's coming next -- yes, this is the Disk Block Buffer Size that you can set in the Audio System window. Not all audio software provides user adjustments for this setting (*Cubase Audio 6* apparently fixes it at 12K, but there are ways to increase this), but if you understand the reasoning behind this value, you should be able to investigate and optimise whatever parameters are available in your software Preferences or Setup windows.


Echo Reporter uses a fixed 32K block size, but *DskBench* carries out its 8-track playback test with a set of different block sizes: 128K, 64K, 32K, 16K, 8K, and 4K (see the top graph on the right). Once you have studied these figures, you will realise why being able to change the block size can make a huge difference to the maximum possible number of playback channels (the associated values for this are shown in the lower graph to the right). It also explains why so many people tend to be disappointed with the figures that *Echo Reporter* reports for their drive. They are not sustained transfer rates, but simulated real-world figures for eight-track audio recording -- which can then be extrapolated for different numbers of channels after the test. Hopefully this finally explains the huge variations in results using different drive speed checks, and why the only really valid ones for multitrack audio are those that use large file sizes (to overcome the effect of any caching and measure sustained transfer rates) and multiple streamed file reads, to include the read/write head seek time.

Summary

Of course, nothing in life is free, and the larger the block size used for each audio channel, the more RAM will be used for buffering. For a chosen number of audio channels you may find that beyond a certain point, increasing the block size uses more RAM than is sensible. For instance, *Cubase VST* allows block sizes of 32, 48, 64, 96, 128, and 256K, but with minimum memory per channel of 96, 144, 192, 288, 384, and 768K. If you foolishly set a 256K block size, and try for 32 channels, you will use 24Mb of RAM for the buffers alone. With a more sensible 64K (the default size), 32 channels will take only 6Mb for RAM buffers.

As you can see from the lower graph on this page, even with a 32K block size, my old Maxtor Diamond Max drive should manage the maximum of 32 mono 44.1kHz 16-bit tracks available in *Cubase VST*. In fact, 32 channels will take 88200 (44.1kHz 16-bits) times 32, which is 2.69Mb/second. So why are we all buying such fast drives for hard disk audio? Well, *Cubase* offers up to 32 mono or stereo tracks, and stereo instantly doubles the requirement to 5.38Mb/second. Even my new Ultra Wide SCSI drive only just scrapes through this requirement with a 64K block size, although its sustained transfer rate is about 15Mb/second. In addition, given the huge drop in drive transfer rates when reading multiple tracks, there will be inevitable extra overheads when you stop measuring with a neat single long file for each track, and enter the real world with lots of smaller track sections dropping in and out, and the inevitable few bits of file fragmentation, as well as little hiccups caused

by the operating system. As soon as you attempt to record several tracks at the same time, the figures will drop even further.

The important thing is to measure the performance of your own drives, and find out the current figure for block size used by your MIDI + Audio sequencer -- this should finally give you a realistic number of achievable tracks. Happy testing! 

Sound On Sound

Media House, Trafalgar Way, Bar Hill,
Cambridge CB3 8SQ, UK.

Telephone: +44 (0)1954 789888

Fax: +44 (0)1954 789895

Email: info@sospubs.co.uk

Website: sospubs.co.uk

© 1998 Sound On Sound Limited. The contents of this article are subject to worldwide copyright protection and reproduction in whole or part, whether mechanical or electronic, is expressly forbidden without the prior written consent of the Publishers. Great care has been taken to ensure accuracy in the preparation of this article but neither Sound On Sound Limited nor the Editor can be held responsible for its contents. The views expressed are those of the contributors and not necessarily those of the Publishers or Editor.
