

Streaming audio in the Networked environment

Ivar Poijes

Swedish Radio

This article offers essential advice to radio broadcasters on fine-tuning a PC-based “information-infrastructure” for the editing, storage and streaming of audio files. The chosen infrastructure is based on the use of four separate entities called “clouds” – Audio workstations, Network, Audio fileserver and Audio-data storage.

The term “information-infrastructure” has been coined to describe all the components required to communicate the information that is necessary for the production and broadcasting of radio programmes. In this article, we will look only at the “long and winding road” that our streaming audio-data has to travel on its journey from the physical hard disk drive (HDD) of some distant fileserver, in a remote data-hall, to the audio workstation where the radio programme is to be edited or sent out to the transmitter network.

Most of the advice given in this article is also relevant to *video* workstations when used in a networked environment – the major differences being the higher bandwidth demands of video and the fact that errors in the audio are subjectively more disturbing than errors in the video.

Even if the diagrams and descriptions have been rather simplified, the article pinpoints some potential performance bottlenecks, and indicates the vital components that can be optimized in the typical information-infrastructure used by many radio stations.

The four clouds of the infrastructure

The four basic clouds of the infrastructure are shown in *Fig. 1*.

Obviously it’s important to realize that the performance and robustness of each cloud will influence the total performance and robustness of the information-infrastructure. It is also important to realize

that deficiencies, or failing components, in the three clouds “Audio-data storage”, “Audio-fileserver” and “Network” are likely to impact groups of, or all, workstations in the “Audio workstations” cloud to the right of the diagram. In fact, the further the deficient cloud is to the left of the diagram, the higher the chances it will affect *all* audio workstations.

Another important fact is that the techniques and components used in each cloud were never designed for the demands of streaming audio-data, except the audio-editing software. We cannot really do much about this as the computer industry in general is driven by other factors than to satisfy the needs of some poor radio stations.

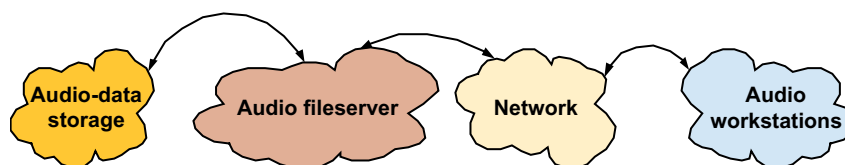


Figure 1
The four basic clouds of the information-infrastructure

What we can do, though, is to analyze the situation and try to design our four clouds as well as possible to meet our needs – and that’s what this article is really about.

The nature of streaming data

So, what’s so special about streaming data? Networks, servers and storage systems are just so much better, nicer, bigger and faster nowadays! How can these slow and tiny audio streams be a problem – when networks, servers and storage are designed today for bandwidths that are hundreds or even thousands of times faster than an audio stream?

Let’s have a brief and simplified recapitulation of the requirements of streaming audio in a networked environment.

The nature of streaming audio-data is very different from that of “typical” office applications. Even if there is a big difference between the workload and the demands of a “typical” fileserver and a “typical” application or database server, the manufacturers of servers and storage have good knowledge of these common workloads and usually know how to optimize their equipment according to these different demands.

Unfortunately few IT-companies have realized how complex the workload of streaming audio on a radio station can be, especially when the number of simultaneous users increases.

An audio file that is played from the server’s storage system and reaches the audio workstation via the network will represent a workload that can be considered as *isochronous*¹.

Audio workstations

At the very end of the chain, in the audio workstations, the speed of the data that is fed to the digital-to-analogue converter (DAC) in the audio card must be constant. If some data arrives too late, there will be drop-outs. If we have a very fast data connection at the input, we still can’t feed data to the audio card any faster than the DAC can handle it – and still be able to convert this data to analogue audio that can be listened to.

The speed of the data required by the DAC is related to the sampling frequency used: we can say that the data fed to the DAC is *synchronized* with the sampling frequency.

Before the data reaches the DAC, it is buffered and delayed many times. The last buffer might be on the audio card itself but, before that, the digital data may have been buffered multiple times in the audio workstation, in the network equipment, in the audio fileserver and in the storage system. Each time that the data is buffered, a small delay is added to the signal. Some buffering is necessary to let the electronics handle the data in a correct way, other buffering is necessary to let the software process the data correctly – and buffering is also done to increase the performance. This last type of buffering is called *caching* and is used mainly to speed up disk access. The delay introduced in the signal chain is called *latency*. If the latency was constant all the time, we would have a so called “real-time system” by definition – a system well adapted for streaming audio.

In real life, few components in our chain, if any, are designed for constant latency – so we get continuing variations in the latency. The normal data traffic consists of short bursts of data. And typically, we will experience greater latency fluctuations when the workload is increased. The last buffers in our chain, i.e. in the Audio workstations, are supposed to compensate for the latency fluctuations in the whole chain to prevent dropouts and interruptions in the audio.

So what we are actually aiming for is just to keep the latency fluctuations within the limits that the audio workstations can handle – at any time!

1. An isochronous transmission is time-dependant in the sense that the data must be delivered within certain time constraints.

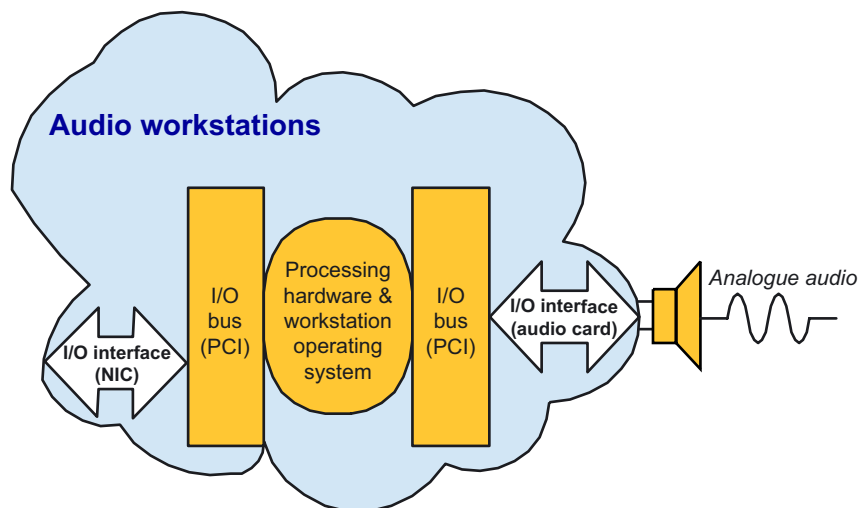


Figure 2
Main elements of the Audio workstations cloud

As long as the number of workstations is limited and other applications are kept away from the audio network, this is readily possible without too much hard work.

However, when you have many workstations used simultaneously – for editing, recording and broadcasting the audio, and they are competing for bandwidth with all other applications in the corporate network – then things can get a lot more complicated. On top of that, throw in a bunch of linear audio applications ... some multichannel workstations ... a couple of fast workstations for ripping CDs ... and combine these with an overall

demand for robustness and continuous operation, even when a disk crashes in the storage system – and then you really need to tune your information-infrastructure as well as possible.

To do this, let's have a closer look at all the clouds in our chain (*Fig. 1*), starting on the right-hand side with the Audio workstations (*Fig. 2*). The workstations described in this article are ordinary "Intel PCs", not the so-called PC workstations with high-end graphics and support for more than one CPU. The operating system is Microsoft Windows, e.g. Windows 9x, NT4, 2000 or XP. The reason for this is simply the fact that they are widely used and they are the operating systems that the author has had most experience in using.

Thanks to the constantly increasing computing demands of PC games and applications such as Microsoft Office, we are now able to buy powerful PCs at quite low prices.

PCs that are less than 2 years old are, in general, significantly better at handling I/O (Input/Output) to or from the PC than earlier models. This is vital when it comes to audio workstations that need to compensate for latency variations caused by networks, servers or storage.

To make the most of an audio workstation, we must consider the following points:

Workstation hardware

- Disable the power-saving functions in the BIOS.
- Don't buy less than 256 MB of RAM if using Windows 2000 or XP.
- Use small low-cost servers, with better redundancy, rather than ordinary PCs for mission-critical workstations (such as on-air broadcasting workstations). Remember, though, that servers are NOISY so you need to have them placed where they don't disturb anybody.
- Consider using a "disaster operations" PC in the continuity areas. This is an audio workstation containing the next few hours of output, which can operate on its own if a serious network or server failure occurs.

Workstation operating system

- Never use file compression on the internal hard drive.
- Disable all screensavers.
- Limit the number of services and applications that are running by default on the audio workstation. There are many "tuning sites" on the Internet that describe how to tune Microsoft's client operating systems. Which services that can be turned off will depend on your configuration and environment.

Network interface card

- Configure your Network Interface Cards (NICs) for the speed and duplex mode that match the network switch it is connected to. Never use any "auto" settings for speed or duplex mode when using Ethernet or

Fast Ethernet. The most common settings are 10 Mbit/s & half duplex for Ethernet, and 100 Mbit/s & full duplex for Fast Ethernet. For Gigabit Ethernet, the only available setting is auto. Many NICs on new PCs can support all three standards – Ethernet, Fast Ethernet and Gigabit Ethernet.

- If possible, adjust the network buffering to maximum. This can be done by some NIC drivers and the Novell Client software, if this is used.
- If the NIC parameters can be set both by the OS and an installed NIC-specific management software (such as 3 Com NIC doctor), you should confirm that the settings correspond.
- The technique for prioritizing traffic in an Ethernet network is very complicated to manage and therefore almost unusable – unless your network guys are really skilled, extremely nice and have a lot of spare time! The easiest way to give certain audio workstations a higher priority in the network is simply to give them higher speed on their network connections. The most critical audio workstations are often the on-air broadcasting workstations and the least critical might be the CD-ripping workstations.
- The strategy for prioritizing the tasks could look like this:
 - Non-audio workstations and CD-ripping workstations should get 10 Mbit/s half-duplex Ethernet connections.
 - Ordinary audio workstations should get 100 Mbit/s full-duplex Fast Ethernet connections.
 - On-air broadcasting workstations should get 1 Gbit/s Gigabit Ethernet connections. Please remember that if “by accident” you use an on-air broadcasting workstation with a Gigabit Ethernet connection for large file copying, then the load on the network and the actual servers and disks can be considerable.

Internal disk controller / host bus adapter

- Use internal SCSI hard drives, rather than IDE/ATA hard drives, if files are to be copied to the local hard drive during editing or on-air broadcasting. (See the section called *I/O interface between the HDD and its controller* on page 7.)

Network cloud

As mentioned earlier, the network topology in this cloud is Ethernet / Fast Ethernet / Gigabit Ethernet (*Fig. 3*). This is the most common and least expensive network solution to implement today. For streaming data however, the access method of Ethernet (CSMA/CD) is very ill suited – especially when the network traffic gets high. By using fast network switches with high headroom, we can compensate for most of the limitations of CSMA/CD – if the total network traffic is kept at low to moderate.

Small simple networks with few components are neither costly nor difficult to set up. Larger networks are more complicated and require a lot of competent planning, configuration and testing before they can be successfully implemented.

The following advice is general and most of the points should be well known to skilled network implementers:

- Only use switched networks, no hubs!, for streaming data.

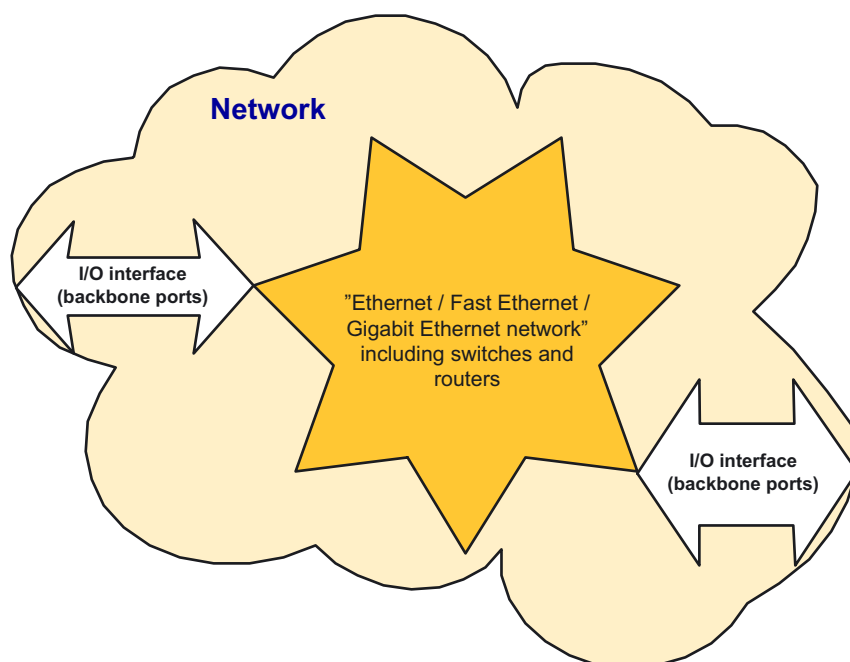


Figure 3
Main elements of the Network cloud

- When upgrading your network to higher speed, verify that any older cabling, including patch cords, are certified for the new speed. This is most important with the new line of “10/100/1000 switches” that support copper-based Ethernet, Fast Ethernet and Gigabit Ethernet on the same physical ports.
- There are some “lower cost” 10/100/1000 switches available on the market. They are designed with an internal bandwidth that is too low to support Gigabit speed on more than a few ports at the same time, even though all ports can be put to Gigabit speed. An overbooked switch can be overloaded by the connected nodes. This can sometimes be accepted for common office implementations but will be really disastrous for streaming audio-data. If network switches that can be overbooked are used, you need to have full control of the connected congested bandwidth and throughput.
- Define the speed and duplex mode on every port. **The only time a port should be set to “auto sensing” is when a gigabit node is connected.** The long story about “auto sensing speed and duplex mode for Ethernet and Fast Ethernet” is a real nightmare, especially for streaming data – and will not be covered here. Just remember to set all ports and all NICs to the corresponding speeds and duplex modes – 100 Mbit/s full-duplex connected to 100 Mbit/s full-duplex ... 10 Mbit/s half-duplex connected to 10 Mbit/s half-duplex, etc.
- Avoid using “Spanning Tree Algorithm” on the server ports. Put the ports in “Port Fast” mode (Cisco) or a corresponding setting.

Audio fileserver

Next along the chain of clouds we reach the Audio fileserver (*Fig. 4*).

The audio fileserver described in this article are Intel servers, since they are very common and the ones that the author has most experience in using.

The simplified block diagram of the Audio fileserver is very similar to that of the Audio workstation. The most important difference is that the fileserver can handle a lot more I/O, both internally and externally, to the network and the storage system.

Many small and mid-range servers use the same processors as the PC workstations in this infrastructure.

Typically, the bottleneck at the fileserver would not be caused by the computing power but rather by the internal I/O architecture.

Depending on the server operating system and the file system used, the system requirements of the server will vary. But for smaller audio installations, any modern mid-range server should be sufficient.

The number of parameters that are tuneable will depend on the server OS that is used.

The following advice is fairly general and is not addressed to any specific server OS (such as Linux, Microsoft Windows XX Server or Novell NetWare).

Server hardware

- Do not buy “no name” servers for your production environment.
- Check that the servers are certified and, if applicable, optimized for the server OS you are using. Most

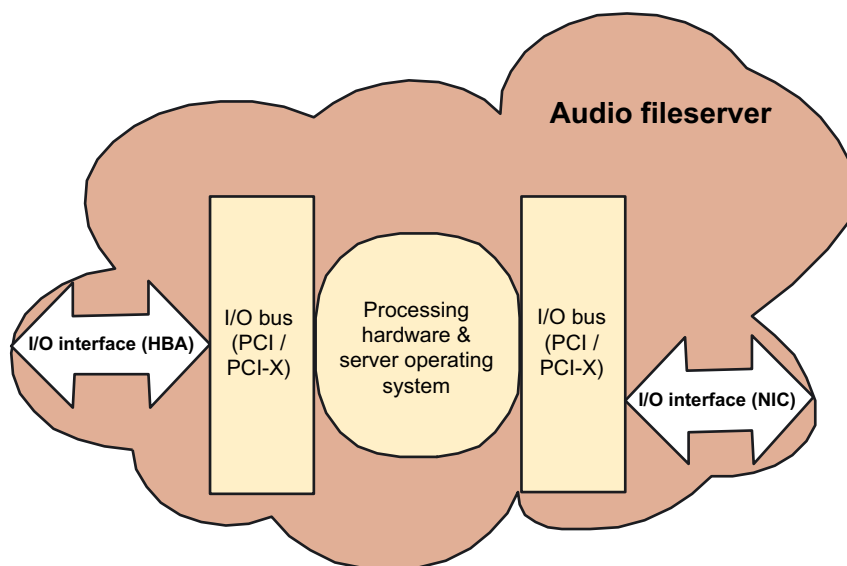


Figure 4
Main elements of the Audio fileserver cloud

servers have a setting in the BIOS where the appropriate OS can be chosen before the installation.

- Different operating systems handle hardware interrupts (IRQ)s differently (which is one of the reasons for choosing the OS in the BIOS). The server BIOS and the PCI bus are supposed to handle the allocation in an optimal way. Additionally, you may need to prevent conflicting IRQs (or the phenomena when high utilization on the NICs can “block” the disk controller) by prioritizing the hardware interrupts before the operating system loads.
- The following IRQs are often possible to assign manually in the server BIOS (listed from highest priority to lowest): IRQ 9/2, 10, 11, 12, 14, 15, 3, 4, 5, 7. Disabling the unused ports or PCI devices can release IRQs. Remember to put disk controllers and Host Bus Adapters (HBAs) on higher priority than NICs!
- Avoid sharing IRQs between different types of PCI devices sharing the same PCI bus. If you have two NICs and two HBAs on the same PCI bus and need to have some shared IRQs, then let the NICs share one interrupt (such as 11 or 5) and let the HBAs share another interrupt with higher priority (e.g. 10).
- For larger installations, you can benefit from a server with multiple PCI buses and balance the HBA and NIC loading between the PCI buses. If you have two PCI buses, let the NICs share one bus and the HBAs the other.

Server operating system

- Install and test all OS patches and new OS releases in a test environment before installing them on your production servers. If Microsoft Windows is used, keep track of the security patches needed to minimize the risk from new viruses and hacker attacks.
- Decide on new OS releases or versions only once or twice a year, not more often.
- Activate “Read Ahead/Pre Fetch” functions in the server OS, if applicable.
- Use the largest possible block size for the file system.
- Do not activate file compression, not even on the system volume/disk on the server.
- If “salvage” or “undelete” functions are supported by the file system, disable this feature (or if you decide to keep this feature, you need to keep track of the amount of files that can be “salvaged” or “undeleted” and manually “purge” these files before the OS starts reclaiming the space as an automatic (performance degrading) background process.
- Never use screensavers on Windows servers!

Network interface cards (NICs)

- Better performance can often be achieved if servers are equipped with dedicated so-called “Server Network Adapters” with integrated processors.
- Connect the servers to the network with “fastest possible” connections; e.g. 100 Mbit/s Fast Ethernet, 1000 Mbit/s Gigabit Ethernet or Fast Ether channel (a Cisco standard to bundle two or more Fast Ethernet connections to increase the bandwidth).

Host bus adapters (HBAs)

- For larger installations with external storage systems, use Fibre Channel HBAs instead of parallel SCSI HBAs. Fibre Channel uses full duplex and handles streaming I/O much better than parallel SCSI.
- Set HBA parameters such as “Que Depth” and “Frame Size” (the Frame Size parameter is only applicable for Fibre Channel) according to the recommendations for your storage system and the current storage configuration.
- Verify the HBA driver performance for your OS before implementing a new HBA type. There is often more performance difference between the HBA drivers for different OSs from different HBA vendors, than between different HBAs.

Audio-data storage

The last cloud, and in many ways the most important one, is the cloud containing the audio-data storage (Fig. 5).

If we get performance problems here, we will most likely suffer a lot of audio dropouts or interruptions on all the audio workstations.

How does storage work?

Hard disk drive

The basic function of the storage system is that data is stored on the spinning physical disk as magnetic patterns corresponding to the digital data. The magnetic write/read head, like the combined record/playback head on a tape recorder, can either write/record data to the disk or read/playback data from the disk. Each hard disk drive (HDD) – what we normally just call a “hard drive” or a “hard disk” – consists of multiple physical disks, covered with a magnetic surface, stacked over each other on a single spindle. Each side of each disk has its own write/read head,

mounted on a small swing arm. The internal controller of the HDD will handle the signal flow to and from all the heads on the swing arms, as well as the position of the swing arms and the speed of the spinning spindle. The mechanics of the HDD are very precise and delicate, and are very sensitive to shocks, dust and wear.

The long-term quality of an HDD is very much decided by the quality of the HDD mechanics – and since moving mechanical parts always wear out eventually, so too do HDDs if they are kept spinning.

I/O interface between the HDD and its controller

The HDD and the connected disk controller communicate with each other using a parallel or serial interface. For servers, the most common parallel interface is *parallel SCSI*.

SCSI (Small Computer Systems Interface) is an old (early 1980s) standardized interface specification and a set of commands for communication between some peripheral devices (e.g. HDDs) and computers. The SCSI standard has developed over the years from a peak transfer rate of 5 MByte/s to 160 MByte/s or more, and is still evolving. Confusion is being created by manufacturers inventing new cool “unofficial names” for different SCSI features.

The latest SCSI standard is called SCSI-3 and is more a collection of different standards, some of them quite different from each other.

Another parallel interface is IDE (Integrated Drive Electronics or Intelligent Drive Electronics), also known as ATA (Advanced Technology Attachment), which is found in most PCs. The later versions of IDE are called Enhanced IDE (EIDE) or Fast ATA (often called ATA-2).

The typical IDE/EIDE HDD is very cheap and seems to provide high capacity and good performance at a low price, which is the reason why it is so popular in PCs. Unfortunately there are some major limitations that make IDE/EIDE ill-suited for use in server storage.

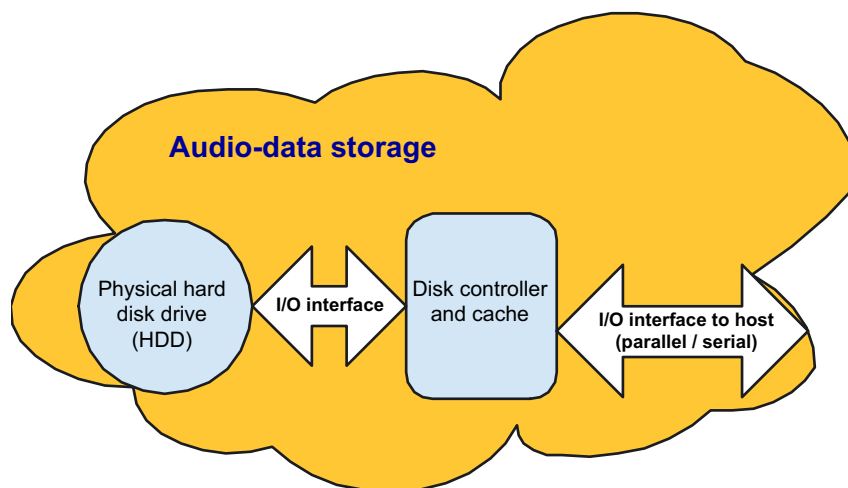


Figure 5
Main elements of the Audio-data storage cloud

The most important differences between SCSI and IDE/EIDE are:

- IDE/EIDE will load the CPU of the computer a lot more than SCSI, as the SCSI-controller handles most of the processing that needs to be done.
- IDE/EIDE can only execute one command at a time while SCSI can queue up to 256 commands per logical unit/disk. This is called “command tagged queuing” and allows a device to accept multiple concurrent commands. Since multiple commands can be accepted by the device before earlier commands are completed, the device can optimize its operation for improved performance.
- SCSI can connect more devices per bus, and the physical bus/cabling can be much longer.
- The price of a SCSI HDD is generally considerably higher than the price of an IDE/EIDE hard drive – but, generally, the quality is also higher.

A serial HDD interface found on some, mainly high-end, storage systems is Fibre Channel, which also offers full duplex SCSI communication to the disk.

For file servers or audio workstations aimed at handling multiple streams of digital audio-data, SCSI hard drives are superior to IDE/EIDE. Don’t even consider installing a server using IDE/EIDE storage – if performance is important.

Controller (SCSI) and cache

The controller organizes and executes the SCSI commands to the disks. The commands are buffered in the command queue and the execution is optimized for improved performance.

The controller will also handle all the extra processing that is needed when the disks are organized in redundant arrays, RAID.

The data that is read from the physical disks are buffered in the “read cache” and reorganized before being forwarded to the I/O interface.

If you start reading the first data blocks of a file, it is most likely that you will also want to read the following blocks of the same file. The function called “read ahead” or “pre fetch” (or something similar) will try to figure out what data are most likely to be needed from the disks and send it to the “read cache” in advance.

If a request comes for data that is already in the cache, the request can be served faster and the performance is improved. This task is quite easy if the number of requests for different data is low but, when the load is increased, it gets harder to predict what data is going to be requested next.

In theory, the larger the cache, the greater the amount of data that can be *read* ahead, the greater the number of requests that can be served directly *from* the cache, and the greater the improvement in performance.

If, however, the requested data is not found in the “read cache”, the performance will be decreased by first searching the cache and then having to address the physical disks instead.

When data is to be written to the disks, it is buffered in the “write cache” and reorganized so it can be written efficiently to the disks. The connected host (server) can get the message that data is written to disk as soon as data is written to the “write cache”. This will improve writing performance since it is much faster to write to cache than to disks. As a background process, the data in the “write cache” must be de-staged (written) to the disks as efficiently as possible. If data in the “write cache” is changed before it is written to disk, the change can be done just in the cache – which will improve performance.

The larger the cache, the greater the amount of data that can be *written* to the cache, the greater the number of requests that can be served directly from the cache, and the greater the performance will be improved – again, that’s the simple theory.

If, however, the de-staging is consistently slower than data is being written to the “write cache”, the cache will be filled up and will stop accepting more data until data is “flushed to disk”. This state is quite unusual and can occasionally be accepted in the normal office environment. But for an audio fileserver under heavy load, this state can more easily occur – and the consequences can be severe.

When this happens, the typical behaviour is that everything works fine for several minutes up to a few hours, while the “write cache” slowly fills up – until the “magic” threshold is reached. At this point, the controller must prioritize de-staging instead of accepting write or read requests from the host. During this period, you are often forced to end most of the sessions to the server, before the storage system will be able to accept new I/O.

In this case, the best advice may be to reduce dramatically the size of the “write cache” and, if possible, lower the threshold when data starts de-staging to disk.

The “streaming data efficiency” of the caching algorithms, in conjunction with the cache size, has a major impact on the controller performance – and the performance of the whole storage system.

[During the last couple of years, the author has been involved – together with some storage manufacturers – in the very time-consuming process of optimizing RAID controller microcode for streaming data. The focus has been on storage aimed at clustered audio file servers and the results have been really successful for the Hitachi Freedom Storage Lightning and Hitachi Freedom Storage Thunder series from Hitachi Data Systems. The “optimized” systems were Lightning 7700e, 9910 & 9960 and Thunder 9200 & 9500 which all had different performance problems that were only apparent when they had a workload of multiple isochronous read and write data streams. Many problems would only appear when a disk crashed or was replaced – which is very likely to happen sooner or later. The solution in most cases involved the caching algorithms.]

The I/O interface to the host / server

All data to and from the storage system will pass through this interface. As mentioned earlier, we use SCSI commands for communication between the storage system and the HBA in the server. The most common SCSI interface is a parallel bus. The fastest parallel SCSI techniques today are called Ultra 160 SCSI and Ultra 320 SCSI and are reputed to give you up to 160 MByte/s and 320 MByte/s peak transfer rate, respectively.

These figures might be reachable in an optimized lab environment but never in the typical production environment.

There are two main reasons why you can’t count on these impressive figures:

- 1) You have a considerable signal overhead caused by the SCSI signalling (even if Ultra 320 SCSI claims to use “Packetized SCSI” to minimize the command overhead).
- 2) Parallel SCSI uses parallel electrical wires for the data transfer and you can only send data in one direction at a time – half duplex. So the best performance is achieved if you only read or write data, while the combination of simultaneous reading and writing will decrease the usable throughput considerably.

Since simultaneous reading and writing of streaming data will be the normal workload for the interface between the audio-data storage and the audio file server, we can benefit a lot by using an interface that can transfer data in both directions simultaneously – full duplex. Fibre Channel offers full duplex SCSI communication at high speed and is well suited for the interface between the audio storage and audio file server.

Abbreviations

ATA	Advanced Technology Attachment	IDE	Integrated / Intelligent Drive Electronics
BIOS	Basic Input / Output System	IRQ	Interrupt ReQuest
CPU	Central Processing Unit	NIJ	National Information Infrastructure (USA)
CSMA/CD	Carrier Sense Multiple Access / Collision Detect	OS	Operating System
DAC	Digital-to-Analogue Converter	RAID	Redundant Array of Independent (or Inexpensive) Disks
HDD	Hard Disk Drive	RAM	Random-Access Memory
I/O	Input/Output	SCSI	Small Computer Systems Interface

Choosing your audio storage systems

Consider the following when choosing audio storage systems (RAID, internal, external):

Hardware

- Don't buy "no name" equipment.
- Check that the storage system is certified and, if applicable, optimized for the server OS you are using.
- For smaller installations, an internal RAID controller (in a PCI slot in the server) usually gets best performance per €/£/\$.
- For larger installations with external storage, use Fibre Channel instead of parallel SCSI since Fibre Channel uses full duplex.

Tuning (default RAID tuning is rarely the best for streaming data)

- Choose RAID 0+1 or RAID 5 in a multi-user environment.
- Choose the largest possible stripe/chunk size.
- Set the disk rebuild priority to as low as possible.
- Verify that background processes such as "internal health checks" are not impacting the performance – if so, turn off all suspect processes if possible. Or change to another type or brand of RAID controller.
- Don't use "too large" a cache on the RAID controller. (Some high-end storage systems can successfully use a very large cache, typically 2 – 8 GByte, in combination with adaptive caching algorithms and fast backend bandwidth to the disks).
- Enable "read ahead" or "pre fetch" on the RAID controller.

The bottom line

The running of streaming data applications in a network environment is a challenge that most of us have already accepted without too many second thoughts.

Streaming data is a typical isochronous workload, which is dependant on data being delivered on time. All the components – hardware as well as software – in our typical information-infrastructure are more or less unpredictable when it comes to guaranteed response times. All these unpredictable responses are added and the result is an information-infrastructure that is almost non-deterministic when it comes to timing.

Usually the fluctuations are not completely unpredictable, since every component has certain timing limits and will stop functioning if these are exceeded. So a better description than "non-deterministic" might be *deterministic-chaotic*.



Ivar Poijes works as an IT developer at the IT Strategy and Development Department of Swedish Radio (former named Swedish Broadcasting Corporation). Starting as a music-loving maintenance engineer at Swedish Radio in the late seventies, he had gained a great experience in testing and troubleshooting both analogue and digital audio equipment when the first implementations of computer-based digital audio editing started in 1994. He soon became involved in tuning and troubleshooting activities relating to streaming media in network environments. Over the years, he has worked on several projects concerning the tuning and optimization of different high-availability solutions for streaming media in corporate networks.

Mr Poijes has been a certified Novell Engineer since 1998 and has designed the streaming media server and storage concept for the new corporate digital-audio-software implementation at Swedish Radio. He can be contacted at ivar.pojjes@sr.se.

During normal operation, we can compensate a lot for the deterministic-chaotic behaviour of each component by following the tuning recommendations given in this article and, hence, give our audio workstations better working conditions.

To address non-normal operations, e.g. when components suddenly stop functioning, we need to write a new chapter. This next chapter could have a title such as: *Robustness in the deterministic-chaotic information-infrastructure used for isochronous data streaming* or, perhaps, *Help! My disks, servers, switches and workstations are crashing – how do I prevent a disaster?*

This chapter has not yet been written ... but who knows when it might be needed?
