In-depth technology for radio engineers

radioworld.com | February 9 2023 | **$5.00**

RADIOWORLD engineering extra

FUTURE

Hybrid models for today's studios

Dominic Giambo writes that spinning up instances from your TOC server can be your gateway to the cloud.

Gagl

Turn any Comrex IP audio codec into a hub for up to 5 remote contributors

Now available from COMREX

- ✓ High-quality Opus encoder
- ✓ Supports up to 5 participants at once
- ✓ Bidirectional audio between all guests
- ✓ Simple for guests - anyone can Gagl

$35 billed monthly

Choose an annual subscription and save $70!

www.comrex.com | +1 978 784 1776 | info@comrex.com

Writer

Dominic Giambo

Engineering Manager, Wheatstone

Hybrid models serve today's fluid studio environments

Spinning up instances from your TOC server can be your gateway to the cloud

Whatever your future facility might look like, it most certainly will involve software and a server or two.

We're moving toward a more fluid studio work environment that is a hybrid of hardware and software, with any combination of virtual or fixed, commodity or specialized hardware.

Already, in the typical AoIP studio facility today, we can cross-connect different devices and control components from anywhere in the network. More and more of those devices are becoming software applications on the network, which makes it possible to get what you need in the interface you want and where you want it.

All of this is leading to fewer racks of equipment and all the costs associated with that gear, like electrical, cooling and real estate.

We'll be able to hedge against supply chain interruptions and have more flexibility for adding on studios and sharing resources between facilities by using hardware that is available in different forms, mainly commodity servers that are already widely in use by enterprises.

If we can offload broadcast functions onto a commodity server, we can now adapt technology at the rate of Moore's Law rather than wait for the chips and development time to continually create the next generation hardware specific to each task.

Why now?

We used to rely on digital signal processors, application-specific silicon designed to do math very quickly. This made these DSPs useful for mixing and processing audio where we needed to touch thousands of samples every second and lots of math needed to be done.

Over time, Moore's Law has increased the power of the CPUs that we have. Every two years we're doubling the number of transistors that we have available in CPUs, especially server-grade CPUs. This has driven a lot of advancements, including being able to move a lot of functions into a server rather than having a DSP doing the work.

One of the things we've been able to do with all this new hardware is known as Single Instruction, Multiple Data.

SIMD is a fairly recent development, and this lets us do more math in a CPU. The basic idea is you have a set of values such as you get from an audio system, where there are thousands of samples coming in. Normally we'd need a CPU to process each sample individually and do the math on it.

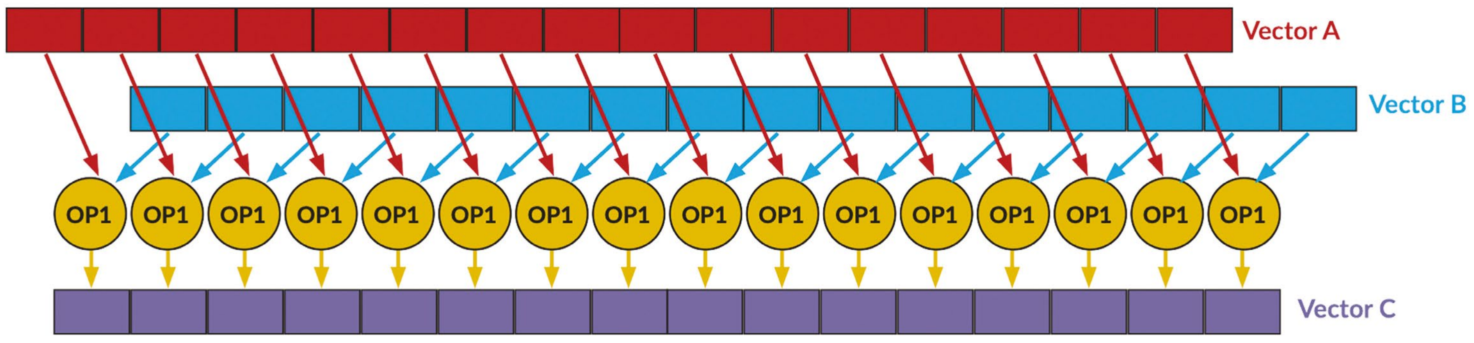
With SIMD, we're able to process entire blocks of samples. We can load up two separate vectors and do one operation on that entire block at once, which essentially allows us to do what a DSP used to do.

In Fig. 1, data from Vector A and Vector B can be added or multiplied together to create Vector C, which can be processed as a block that takes one clock cycle.

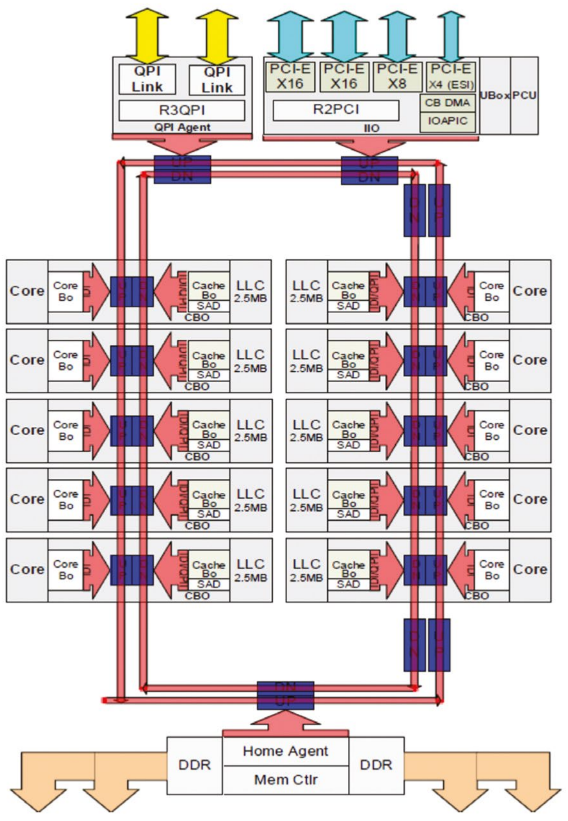
Another thing we're able to do with these additional transistors is add more CPU cores into each CPU die. Maybe we'll need only one core or two cores, or maybe we'll need 16 or 128 — we can get whatever computer power we need for a particular application. This is scaling at the rate of Moore's Law, and we're getting more and more cores every year that we can use to do more math.

Below Fig. 1: SIMD lets developers increase the computing power of CPUs. This is a typical SIMD application showing data from Vector A and Vector B that can be added or multiplied together to give an output (Vector C) that takes one clock cycle.

Single Instruction, Multiple Data Units (SIMD)



Low Core Count Layout



Left
Fig. 2: More CPU cores let developers scale to the application.

With SIMD efficiency and the extra cores, we can now build a suite of broadcast-specific software that runs on a regular CPU. Any machine running a modern Linux OS will work. Typically, though, it's a Dell or Hewlett Packard server, as servers tend to run more efficiently and have higher-quality hardware that generally has more longevity than a regular PC.

To contrast the difference between what we are doing today and this new server-based operation, let's look at the backend mixing that takes place in any AoIP networked studio today.

Fig. 3 shows a typical studio today with a hardware-specific mix engine on the top right and console surfaces on the left, with the consoles sending control inputs to the mix engine. The mix engine is then feeding the mix engine CPU and a DSP array with the audio. As coefficients change, the mix engine is receiving those coefficients and updating those signal processors and that gives you your mixing and processing.

Contrast today's DSP-based, 1RU mix engine with a software-based mix engine that is running in a server, as shown in Fig. 4. Here, the consoles and tablets still connect in the same way and connect to the mix engine as before, except the mix engine is now in a server. Instead of the mix engine using DSP to do all the mixing and processing, it's using the CPU inside a commodity server with SIMD and multiple cores.

WORLD'S BEST ANTENNAS ENGINEERED AND BUILT RIGHT HERE FOR 80 YEARS!

As we mark our latest milestone, we celebrate the **innovative engineers** who design the world's finest TV, FM and RF systems, and the **expert production teams** who build each component to the very highest spec, right here at Dielectric in Maine.

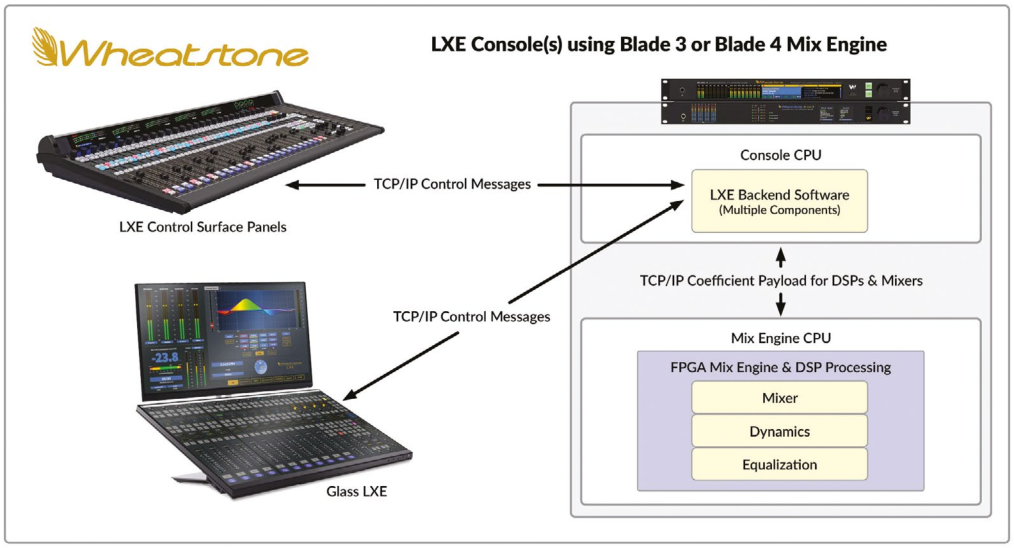
Dielectric®
Trusted for Decades. 80 YEARS Ready for Tomorrow.
1942 2022

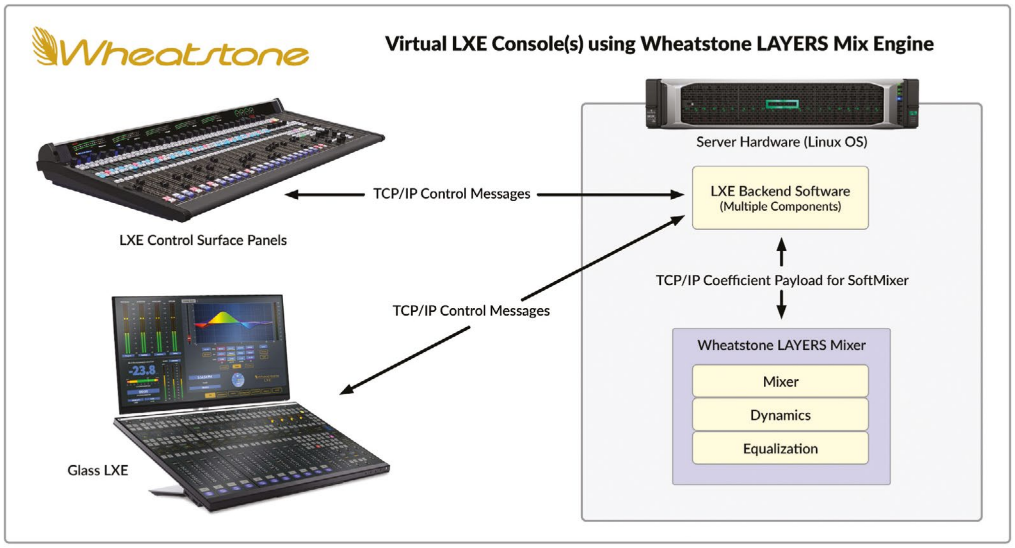
But there's more to broadcasting than mixing. We can do the same with audio processing and streaming functions. In much the same way, we can spin up instances off the server for mixing with all the routing, logic and EQ/dynamics, we can run instances of FM/HD processing with AGC/limiting, RDS, stereo generator and MPX directly from the server into the transmitter. We can also run instances for stream provisioning, metadata and processing specifically for internet streams that are going to content distribution networks. See Fig. 5.

All this can run on one server. That's a single server running multiple mix engines serving multiple surfaces at once with very little utilization due to the use of SIMD and multipole cores available. You can choose your CPU based on what kind of processing and the number of mixers you need.

By integrating this suite of software into a new or existing installation, you can choose the audio endpoints appropriate for your application and leverage the server resources for mixing and processing.

By having a server at the ready, you have the ultimate in redundancy. You have software that can run on nearly any X86 CPU and with built-in automatic failover found in most studios today, you can seamlessly switch to redundant hardware in the event of a failure. As cloud becomes more prevalent and backup interconnect connections become more reliable, as we're seeing daily, you have the option of moving all these resources offsite to be managed by a cloud provider.


Wheatstone
LXE Console(s) using Blade 3 or Blade 4 Mix Engine
LXE Control Surface Panels
TCP/IP Control Messages
Console CPU
LXE Backend Software (Multiple Components)
TCP/IP Coefficient Payload for DSPs & Mixers
TCP/IP Control Messages
Mix Engine CPU
FPGA Mix Engine & DSP Processing
Mixer
Dynamics
Equalization
Glass LXE


Wheatstone
Virtual LXE Console(s) using Wheatstone LAYERS Mix Engine
Server Hardware (Linux OS)
LXE Control Surface Panels
TCP/IP Control Messages
LXE Backend Software (Multiple Components)
TCP/IP Coefficient Payload for SoftMixer
TCP/IP Control Messages
Wheatstone LAYERS Mixer
Mixer
Dynamics
Equalization
Glass LXE

Top
Fig. 3: A typical AoIP networked studio today with purpose-built mix engines and console surfaces.

Bottom
Fig. 4: Wheatstone's Layers mix engine software running in a server. One server can supply multiple mix engine instances for all consoles in a facility, replacing racks of mix engine hardware designed specifically for that purpose.

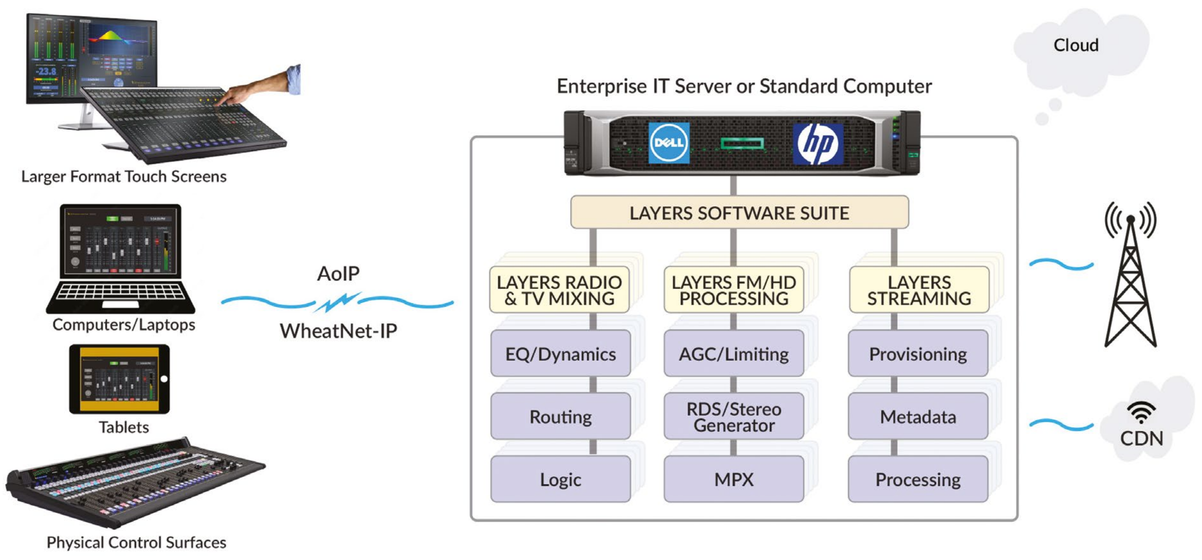
Onsite cloud

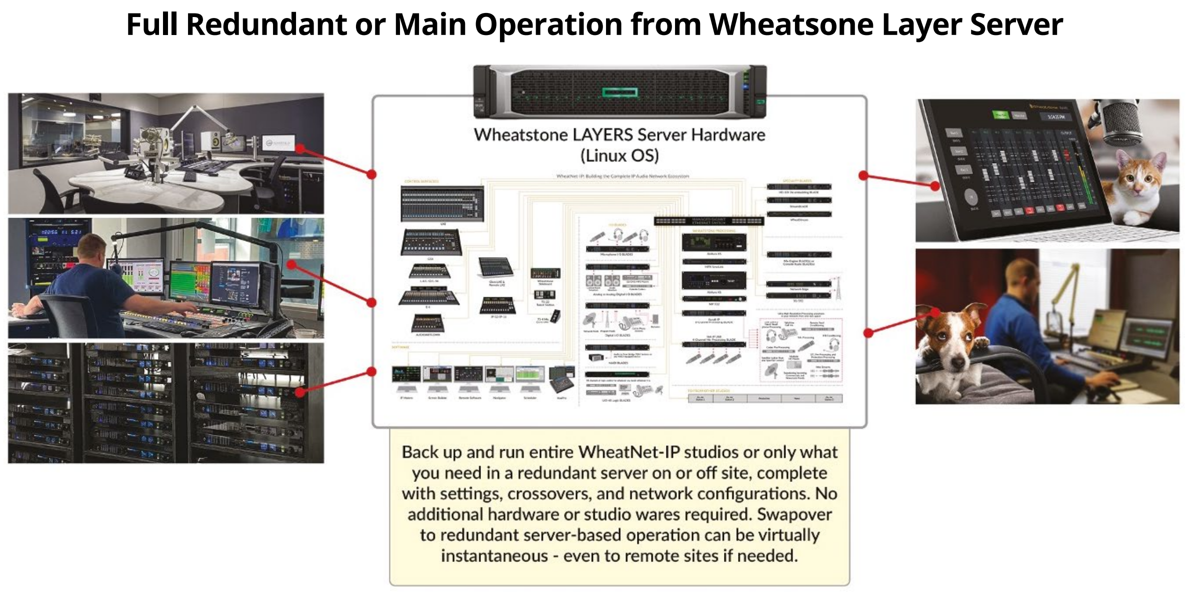
One advantage to an onsite versus offsite server is that you eliminate the long transmit delays between studio and a cloud provider. Audio requires constant, real-time processing, and loss of that for even a second is not tolerable. It could be some time before we have confidence in the reliability of cloud technology, and as that reliability grows, we can move further in the cloud.

But right now, using servers in your TOC gives you many of the benefits of cloud without risking everything on a cloud provider that you have no control over. Even once you're fully in the cloud, you may still want to run an on-premises server so that if the cloud goes down, you can always failover to that server. A backup server doesn't necessarily have to be powered up and online all the time, but having it there at the ready lets you keep listeners and advertisers happy — plus it satisfies the regulatory requirements for EAS, should that cloud go down.

There are cloud-like technologies that you can use in your facility to manage server software. Hypervisor software like VMWare can enable you to use part of the same server for automation and another part for another task without requiring changes to the architecture. The nice thing about virtualization is it decouples the software even more from the hardware, so you can copy the VM around or move it to a new computer without doing an installation; just copy the VM and run it.

Or, you might want to go with a container model where you have a number of services running on a container engine, as shown in Fig. 7. With Docker and other containerization software, you don't have the overhead of the guest OS running in every single instance as you would with virtualization. Instead, you have one container engine running the containers specific to each service or application.



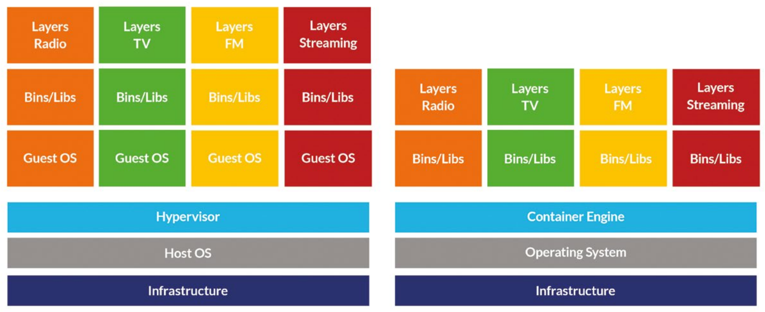

Full Redundant or Main Operation from Wheatsone Layer Server
Wheatstone LAYERS Server Hardware (Linux OS)

Back up and run entire WheatNet-IP studios or only what you need in a redundant server on or off site, complete with settings, crossovers, and network configurations. No additional hardware or studio wares required. Swapover to redundant server-based operation can be virtually instantaneous - even to remote sites if needed.

Top
Fig. 5: Wheatstone Layers Server Software running on a commodity server. Included are multiple instances for mix engine routing, logic and EQ; FM/HD AGC/limiting, RDS, stereo generator and MPX out; and stream provisioning, processing and metadata support out to a CDN. Instances of any of these functions can be spun up at any time and multiple instances of each or all can run at one time from a single commodity server.

Middle
Fig. 6: Layers Server Software running on a server used as a backup. A redundant server located on or offsite can replace a fixed studio backup with all studio settings, crossovers, and network configurations included for instantaneous failover in an emergency.

Bottom
Fig. 7: A container model. Unlike virtualization software like VMWare, containerization software like Docker doesn't have the additional overhead of a guest OS running every instance. One container engine manages containers specific to each service or application.



Going to a cloud provider

At some point, you might want to use a third-party cloud provider. One example might be one-off mixing for a remote event. There are microservices on AWS, Azure and other clouds that go up instantly and come down instantly, depending on request. You might want to spin up a mixing instance on AWS for a short period of time and then shut it down, in which case you would pay Amazon for only that time period.

Instances can be spun up on demand in seconds, especially if you're using a container.

Containers generally come up quickly and configure themselves as part of their startup. The container typically grabs the license for whatever software you're using (that license is baked into the container) and then it would be online.

As broadcasting continues to move into a more fluid studio environment, we'll see a more hybrid model of hardware and software, server and cloud. RW