DEVELOPER STORY

Building A 300 Channel Video Encoding Server

NETINT VPU Technology with Ampere® Altra® Max Processors set new operational cost and efficiency standards

By John O'Neill

SNAPSHOT

Organization - NETINT, Supermicro, and Ampere® Computing

Problem - The demand for high-quality live video streaming has surged, putting pressure on operational costs and user expectations. Legacy x86 processors struggle to handle the intensive video processing tasks required for modern streaming needs.

Solution - NETINT reimagined the video transcoding server by combining their Quadra VPUs with Ampere's Altra Max Processor, creating a smaller, faster, and more cost-effective server. This new server architecture allows for advanced video processing capabilities, including AI inference tasks and automated subtitling using OpenAI's Whisper.

Key Features

- **High Performance:** Capable of simultaneously transcoding multiple video streams (e.g., 95× 1080i30, 195× 720i30).
- **Cost-Effective:** Reduces operational costs by 80% compared to traditional x86-based solutions.
- **Advanced Processing:** Supports deinterlacing, software decoding, and AI inference tasks.
- **Flexible Control:** Managed via FFmpeg, GStreamer, SDK, or NETINT's Bitstreams Edge application interface.

Technical Innovations

- Custom ASICs: NETINT's proprietary ASICs for high-quality, low-cost video processing.
- Ampere Altra Max Processor: Provides unprec-edented efficiency and performance, optimized for dense computing environments.
- Optimized Software: Utilizes the latest FFmpeg releases and Arm64 NEON SIMD instructions for significant performance improvements.

Impact - The collaboration between NETINT, Supermicro, and Ampere has resulted in a groundbreaking live video server that:

- Increases throughput by 20x compared to software on x86.
- Operates at a fraction of the cost.
- Expands system functionality to support video formats not natively supported by NETINT's VPU.
- Enables accurate, real-time transcription of live broadcasts through automated subtitling.

Introduction

The demand for high-quality live video streaming has grown exponentially in recent years. In both developed and emerging markets, operational costs are under pressure while user expectations are expanding. This led NETINT to reimagine the video transcoding server, resulting in a live video server that opens new video processing capabilities created in collaboration with Supermicro and Ampere Computing.

A unique aspect of this architecture is that while NETINT VPUs handle the intensive video encoding and transcoding processing, a powerful host CPU can perform additional functions like deinterlacing and software decoding that the VPU doesn't support in hardware. Additionally, a powerful host CPU can perform AI inference tasks. NETINT recently announced the industry-first automated subtitling using OpenAI's Whisper, optimized for the Ampere® Altra® Max processor, which enables accurate, real-time transcription of live broadcasts. This server performs video deinterlacing and transcoding in a dense, high-performance, and cost-effective manner not possible with legacy x86 processors.

Powered by the Ampere CPUs, the server performs video processing and transcoding tasks in a dense, high-performance, and cost-effective manner not possible with x86 processors. Video engineers control the server via FFmpeg, GStreamer, SDK, or NETINT's Bitstreams Edge application interface, making it accessible for deploying and replacing existing transcoding resources or in greenfield installations.

This case study discusses how NETINT, Supermicro, and Ampere engineers optimized the system to deliver a reimagined video server that simultaneously transcodes 95x 1080i30 streams, 195× 720i30 streams, 365× 576i30 streams, or a combined 100× 576i, 100× 720i, 10× 1080i, 40× 1080p30, 40× 720p30, and 10× 576p streams in a single Supermicro MegaDC SuperServer ARS-110M-NR 1U server. This server expands the system functionality by enabling video formats not natively supported by NETINT's VPU, such as decoding 96 incoming 1080i30 H.264 or H.265 streams via Ampere Altra Max processor and 320 incoming 1080i MPEG-2 streams.

> *"The punchline is that with an Ampere Altra Max Processor and NETINT VPU, a Supermicro 1U server unlocks a whole new world of value,"*
>
> **Alex Liu, Co-founder, NETINT.**

NETINT's Vision

Responding to customers' concerns about limited CPU processing and skyrocketing power costs, NETINT built a custom ASIC for one purpose: highest-quality, lowest-cost video processing and encoding. NETINT reinvented the live video transcoding server by combining NETINT Quadra VPUs with Ampere's Altra Max processor to create a smaller and faster server that costs 80% less to operate and increases throughput by 20x compared to software on x86.

Requirements to Reinvent the Video Server

1. Engineer it smaller and faster.
2. Make it cost 80% less to operate.
3. Increase throughput by 20x.

Why NETINT Chose Ampere Processors

NETINT was already familiar with Ampere Computing's high-performance and low-power processors, which perfectly complement NETINT's Quadra VPUs. The Ampere Altra Max Cloud Native Processor is designed for a new era of computing and an energy-constrained world—delivering unprecedened efficiency and performance. From web and video service infrastructure to CDNs to demanding AI inference, Ampere products are the most efficient dense computing platforms on the market. The benefits of using a Cloud Native Processor like Ampere Altra Max include improved efficiency and scalability, which have great synergy with NETINT's high-performance and energy-efficient VPUs.

Problem

Could Ampere Altra Max simultaneously deinterlace 100 576i, 100 720i, and 10 1080i simultaneous video streams that legacy x86 processors couldn't in a cost-effective 1RU form factor?

How Ampere Responded

Engineers from NETINT, Supermicro, and Ampere unlocked the high performance available with NETINT's Quadra VPU and Ampere Altra Max 96-core processor to redefine the live stream video server. Initial results with Ampere Altra Max using FFmpeg 5.0 were encouraging compared to legacy x86 processors but didn't meet NETINT's goal to increase throughput by 20x while reducing costs by 80%.

Ampere engineers studied different deinterlacing filters available in FFmpeg and investigated recent Arm64 optimizations available in recent FFmpeg releases. An FFmpeg avfilter patch that provides optimized assembly implementation using Arm64 NEON SIMD instructions showed a significant performance increase in video deinterlacing with up to 2.9x speedup using FFmpeg 6.0 compared to FFmpeg 5.0. With all architectures, and especially true for the Arm64 architecture, using the "latest and greatest" versions of software is recommended to take advantage of performance improvements.

Figure 1

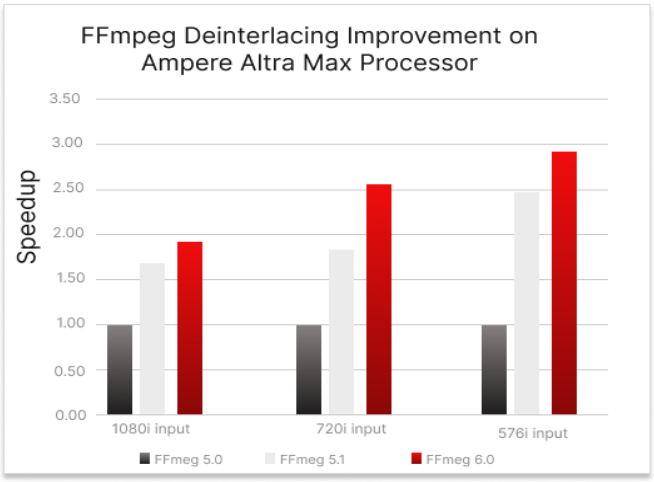

Figure 1 FFmpeg Arm64 Deinterlacing Optimizations improved performance by up to 2.9x faster on Ampere Altra Max processor going from FFmpeg 5.0 to 6.0.

Performance Challenges

NETINT, Supermicro, and Ampere engineers went to work running the full video workload, combining CPU-based video deinterlacing and transcoding using NETINT's Quadra VPUs. With outstanding results just running the deinterlacing jobs, initial results running the full video workload didn't meet the performance target. Combining their broad expertise in hardware and software optimization, the team analyzed, root caused, and were able to meet the aggressive requirements and, in the end, used just 50-60% of Ampere Altra Max Processor's CPU utilization, allowing headroom for future features.

The initial results didn't meet the target of simultaneously transcoding 100× 576i, 100× 720i, 10× 1080i, 40× 1080p30, 40× 720p30, and 10× 576p input videos. Investigating the performance showed performance initially was close to the goal yet unexpectedly slowed down over time. Following the performance methodology outlined in Ampere's tutorial, "Performance Analysis Methodology for Optimizing Altra Family CPUs," by first characterizing platform-level performance metrics. Figure 2 shows the mpstat utility data: initially, the system was running within ~4% of the performance target yet was only running at ~71% overall CPU utilization, with ~36% in user space (mpstat %usr), and ~35% in system-related tasks – kernel time (mpstat %sys), waiting for IO (mpstat's %iowait), and soft interrupts (mpstat %soft). The fact that the system was idle ~29% of the time indicated that something was blocking performance.

Figure 2

Average:	CPU	%usr	%nice	%sys	%iowait	%irq	%soft	%steal	%guest	%gnice	%idle	CPU Utilization
Average:	all	36.29	0	15.29	10.48	0	9.3	0	0	0	28.64	71.36

Figure 2 *mpstat utility output showing the system is idle 100.0 - 71.4 = 28.6% of the time during initial performance analysis when the system wasn't meeting the performance target. This showed us what we needed to determine what was limiting system performance.*

With the large percentage in software interrupts and IO wait time, we initially investigated interrupts using the softirq tool in BCC, which provides BPF-based Linux IO analysis, networking, monitoring, and more. The softirq tool traces the Linux kernel calls to measure the latency for all the different software interrupts on the system, outputting a histogram graph showing the latency distribution. The BCC tools are very powerful and easy to run. It showed ~20 microsecond average latency in the driver used by NETINT's VPU while handling ~40K interrupts/s. As our performance problem was of the order of milliseconds, the BCC softirq tool showed that software interrupts weren't limiting performance, so we continued to investigate what was limiting performance.

Figure 3

```
softirq = block
     usecs               : count     distribution
         0 -> 1          : 222       |                                        |
         2 -> 3          : 118363    |*****************                       |
         4 -> 7          : 276249    |****************************************|
         8 -> 15         : 164502    |***********************                 |
        16 -> 31         : 49870     |*******                                 |
        32 -> 63         : 69452     |**********                              |
        64 -> 127        : 34325     |****                                    |
       128 -> 255        : 9134      |*                                       |
       256 -> 511        : 730       |                                        |
       512 -> 1023       : 284       |                                        |
      1024 -> 2047       : 714       |                                        |
      2048 -> 4095       : 5646      |                                        |
      4096 -> 8191       : 2171      |                                        |
      8192 -> 16383      : 7         |                                        |
```

Figure 3 *BCC softirq tool measures software interrupt latency. softirq block device output showing block IRQ average latency of ~12 usecs and thus not critical for the overall performance when running at 30 FPS or 33 milliseconds per frame.*

Next, we used the perf record/perf report utilities to measure various Performance Measurement Unit (PMU) counters to characterize the low-level details of how the application was running on the CPU, looking to pinpoint performance bottleneck(s). As we initially didn't know what was limiting performance, we collected PMU counter data to measure CPU utilization (CPU cycles, CPU instructions, Instructions per Clock, frontend, and backend stalls), cache and memory access, memory bandwidth, and TLB access. As the system after reboot reached ~96% of the performance target and degraded to ~60% after running many jobs, we collected perf data after reboot and when the performance was poor. Analyzing the PMU data to look for the biggest differences in the good and poor performance cases, the kernel function alloc_and_insert_iova_range stood out by taking 40x more CPU cycles in the poor performance case. Searching Linux kernel source code via the very powerful live grep website showed this function is related to IOMMU. Rebooting the kernel with the iommu.passthrough=1 option resolved the performance degradation over time issue by reducing TLB miss rate. We were at **~96%** of the performance target, so we were close but needed extra performance to meet our goals!

NETINT engineers made the final performance speedup. They saw additional Arm64 deinterlacing optimizations available in FFmpeg mainline, which met our performance goals while reducing the overall CPU utilization to 50-60%, down from 70%.

Figure 4

Function	STALL_FRONTEND (%)			CPU Cycles			CPU Instructions		
	Slow Config	Fast Config	% Diff	Slow Config	Fast Config	% Diff	Slow Config	Fast Config	% Diff
ffmpeg__ffmpeg__filter_line_c	6.0%	7.0%	86.2%	49.4%	49.0%	100.8%	69.0%	69.7%	98.9%
[kernel.kallsyms]__swapper__cpuidle_enter_state	24.0%	18.6%	129.0%	2.6%	1.9%	131.4%	0.2%	0.2%	104.8%
[kernel.kallsyms]__ffmpeg__arch_local_irq_enable	6.7%	7.0%	96.7%	2.5%	2.0%	120.6%	1.0%	0.6%	154.0%
[kernel.kallsyms]__swapper__arch_local_irq_enable	2.3%	1.5%	159.2%	1.5%	0.5%	288.7%	0.8%	0.2%	405.0%
libc.so.6__ffmpeg__0x0000000000097ecc	0.1%	0.1%	75.0%	1.1%	1.5%	76.9%	0.2%	0.2%	110.0%
[kernel.kallsyms]__ffmpeg____alloc_and_insert_iova_range	1.9%	0.0%	**18900.0%**	0.8%	0.0%	**4000.0%**	0.2%	#N/A	#N/A

Figure 4 *perf utility output showing performance critical functions when the system was running slow and fast. The function __alloc_and_insert_iova_range shows a very large increase in the CPU cycles and Stall Frontend. This led us solving the performance degradation over time by using the Linux kernel boot option iommu.passthrough=1.*

The Results

The result is the NETINT 300 Channel Live Stream Video Server Ampere Edition based on a collaboration of NETINT, Supermicro, and Ampere, which can simultaneously transcode 95× 1080i30 streams, 195× 720i30 streams, 365× 576i30 streams, or a combined 100× 576i, 100× 720i, 10× 1080i, 40x 1080p30, 40× 720p30, and 10× 576p streams in a Supermicro MegaDC SuperServer ARS-110M-NR 1U server. This server expands the system functionality to enable running video workloads that require high-performance CPU performance in a dense, power, and cost-effective 1U server.

Call to Action

NETINT's vision to reimagine the live video server based on customer demands resulted in the NETINT Quadra Video Server Ampere Edition in a Supermicro 1U server chassis, unlocking a whole new world of value for customers who need to run video workloads that require high-performance CPU processing in addition to video transcoding with NETINT's VPUs.

Alex Liu and Mark Donningan from NETINT, Sean Varley from Ampere Computing, and Ben Lee from Supermicro have a webinar available to watch on NETINT's YouTube channel, "How to Build a Live Streaming Server that delivers 300 HD interlaced channels," which provides additional information.

Other video workloads that are excellent to run on this server include AI inference processing, which NETINT recently announced and demonstrated at NAB 2024 - NETINT unveiled the Industry-First Automated Subtitling Feature With OpenAI Whisper running on Ampere.

About the Companies

NETINT
Founded in 2015, NETINT's big dream of combining the benefits of silicon with the quality and flexibility of software for video encoding using proprietary ASICs is now a reality. As the first commercial vendor for video processing-specific silicon, NETINT pioneered the development of the video processing unit (VPU). Nearly 100,000 NETINT VPUs are deployed globally, processing over 300 billion minutes of video.

Supermicro
Supermicro is a global technology leader committed to delivering first-to-market innovation for Enterprise, Cloud, AI, Metaverse, and 5G Telco/Edge IT Infrastructure, with a focus on environmentally friendly and energy-saving products. Supermicro uses a building blocks approach to allow for combinations of different form factors, making it flexible and adaptable to various customer needs. Their expertise includes system engineering, focused on the importance of validation, and ensuring that all components work together seamlessly to meet expected performance levels. Additionally, they optimize costs through different configurations, including choices in memory, hard drives, and CPUs, which together make a significant difference in the overall solutions that Supermicro provides.

Ampere Computing
Ampere is a modern semiconductor company designing the future of cloud computing with the world's first Cloud Native Processors. Built for the sustainable Cloud with the highest performance and best performance per watt, Ampere processors accelerate the delivery of all cloud computing applications. Ampere Cloud Native Processors provide industry-leading cloud performance, power efficiency and scalability. For more information visit https://amperecomputing.com.