How to Slash CAPEX, OPEX, and Carbon Emissions Using the NETINT T408 Video Transcoder

A How-To Guide from NETINT

Introduction

Real-time streaming experiences like live events, interactive video, cloud gaming, video communications, and virtual worlds are seeing massive consumer adoption. Meeting this demand with CPU-based codecs like x264 and x265 is expensive and inefficient, unnecessarily boosting CAPEX, OPEX, and carbon emissions generated by power hungry CPUs.

The trend for large platforms, like YouTube, is to build custom Application Specific Integrated Circuits, or ASICs, like Google's Argos Video Coding Unit (VCU), which according to one report, has replaced over 10 million Intel CPUs in YouTube alone.

While most companies can't build their own ASIC, NETINT's Codensity™ ASIC-powered T408 video transcoder can deliver the same benefits for engineers that encode and process massive quantities of live or interactive streams.

This How-To Guide compares the output quality, CAPEX, OPEX and carbon emissions for three production scenarios, as follows:

1. Encoding with FFmpeg using x264 and x265 on a 32-core AWS instance.
2. Encoding with FFmpeg using x264 and x265 on a 32-core server.
3. Encoding H.264 and HEVC with ten NETINT T408 video transcoders on a 32-core server.

The NETINT T408 video transcoder can output H.264 and HEVC encoded files and this document details the results for both codecs, first H.264, and then HEVC. Then it briefly addresses implementation details like factors to consider when buying a server to house the T408s and software options for managing transcoding activities.

Table 1 shows a three-year financial summary of the three approaches for transcoding to H.264.

Cost Summary – H.264	Three Year TCO	T408 Savings	Carbon Emissions	T408 Savings
AWS Graviton2 - CPU-only	$311,490	86%	32	50%
Dell CPU-only - Buy/Co-Locate	$193,256	78%	109	85%
Dell hosting T408's - Buy/Co-Locate	$43,420	--	15.9	NA

Table 1. Three-year cost summary for 100 H.264 live encoding ladders for a 24/7 operation.

Table 2 shows a three-year financial summary of the three approaches for transcoding to HEVC.

Cost Summary – HEVC	Three Year TCO	T408 Savings	Carbon Emissions	T408 Savings
AWS AMD EPYC 7313 - CPU-only	$713,484	95%	113	88%
Dell CPU-only - Buy/Co-Locate	$568,400	94%	325	96%
Dell hosting T408's - Buy/Co-Locate	$34,736	--	13	NA

Table 2. Three-year cost summary for 100 HEVC live encoding ladders for 24/7 operation.

About the T408

NETINT designs, develops, and sells ASIC-powered video transcoders like the Codensity T408, which is a video encoder and transcoder in a U.2 form-factor containing a single Codensity G4 ASIC (Figure 1).

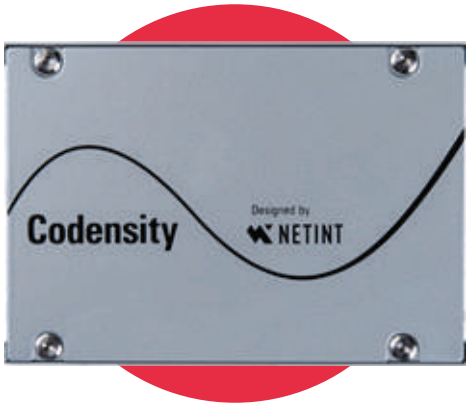


Figure 1. NETINT Codensity T408 in a U.2 form factor.

Operating in x86 and Arm-based servers, T408 video transcoders utilize our proprietary ASIC-based video processors to output H.264 or HEVC at up to 4Kp60, 4x 1080p60, or 8x 1080p30 streams per T408 module. As you'll see in the test results below, at lower resolutions, the T408 can produce even more simultaneous streams.

By offloading complex encode/decode processing to the ASIC, T408 video transcoders minimize host CPU utilization and power consumption, enabling a significant yield in stream density while reducing the cost and power requirements by orders of magnitude. The result is a significant improvement in real-time transcoding density compared to any software or even GPU-based transcoding solution, with significant energy savings.

Operationally, NETINT offers highly efficient FFmpeg and GStreamer SDKs that allow operators to apply an FFmpeg/libavcodec or GStreamer patch to complete the integration. We performed all tests for this How-To Guide using the FFmpeg integration.

In terms of power, each T408 U.2 module consumes just 7W of power at full load while delivering encoding output that equals or exceeds a 1RU server that consumes 250W of power for software-based H.264 encoding. This efficiency allows T408-equipped systems to deliver a massive reduction in CAPEX, OPEX, and carbon emissions.

H.264 Transcoding

This guide presents the H.264 results first, then HEVC.

Assumptions – H.264

To compute the costs detailed in this section, we assumed that a producer needed to encode 100 simultaneous H.264 encoding ladders for 24/7/365 operation. Here is the H.264 encoding ladder.

1080p @ 5 Mbps
1080p @ 3.5 Mbps
720p @ 2 Mbps
540p @ 1 Mbps
360p @ 600 kbps

We tested three scenarios. To assess the cost of producing on AWS, we tested using a C6g.8xlarge CPU, which was the instance recommended when we used the AWS Pricing Calculator to project AWS costs. According to AWS, this instance is driven by an AWS Graviton2 CPU.

Then we produced the files on a Dell server driven by an AMD EPYC 7351P 16-Core/32-thread CPU running Ubuntu with 64 GB of RAM. First, we tested CPU-only encoding with FFmpeg and the designated presets, then using ten installed T408s. We ran all tests with a 3-minute excerpt from the Netflix test clip Meridian which was converted to 1080p30 @ 90 Mbps.

Performance Test Description

We drove all systems remotely via SSH, retrieving the same test file from a RAM drive and writing the output files to RAM, removing disk I/O from the equation, and simulating live operation. We tested three encoding schemes:

1. x264 using the medium preset (on the AWS and the Dell systems).
2. x264 using the veryfast preset (on the AWS and the Dell systems).
3. NETINT T408 H.264 encoder using the default settings decoding the incoming H.264 stream with the onboard T408 decoder (hosted on the Dell system).

We produced all streams with a two-second GOP size using CBR bitrate control. All command strings are shown in Appendix I. FFmpeg 4.3.1 was used for all tests.

The goal of each test was to determine how many live 30 fps encoding ladders (not individual streams) each encoding schema could produce. To test, we opened multiple instances and produced encoding ladders until one or more encoding ladders dropped below 30 fps.

As an example, Figure 2 shows three ladders encoding at 30 fps using x264 and the veryfast preset on the Dell system. When we started encoding an additional ladder in the fourth window, all ladders dropped below 30 fps and never recovered.

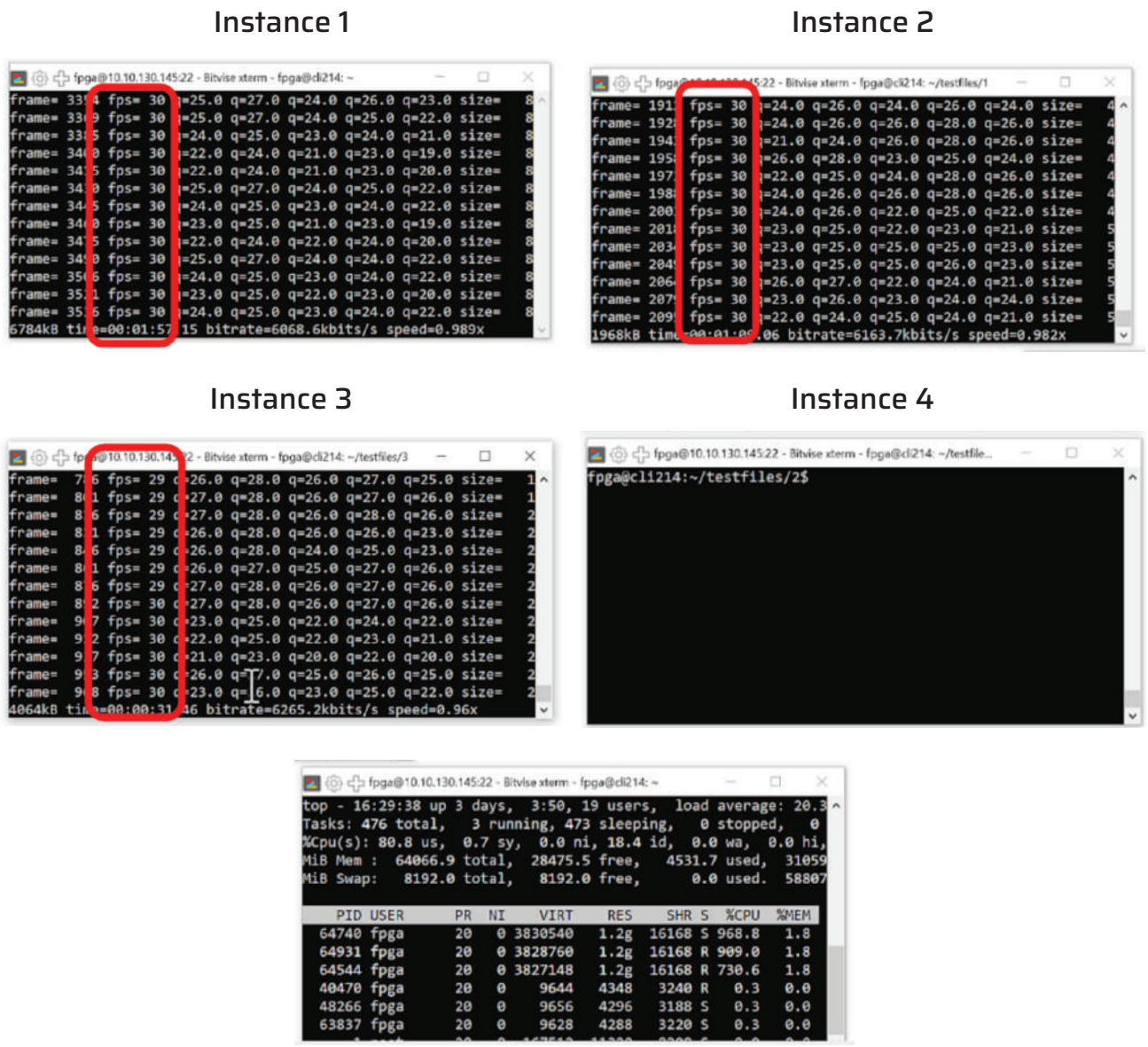


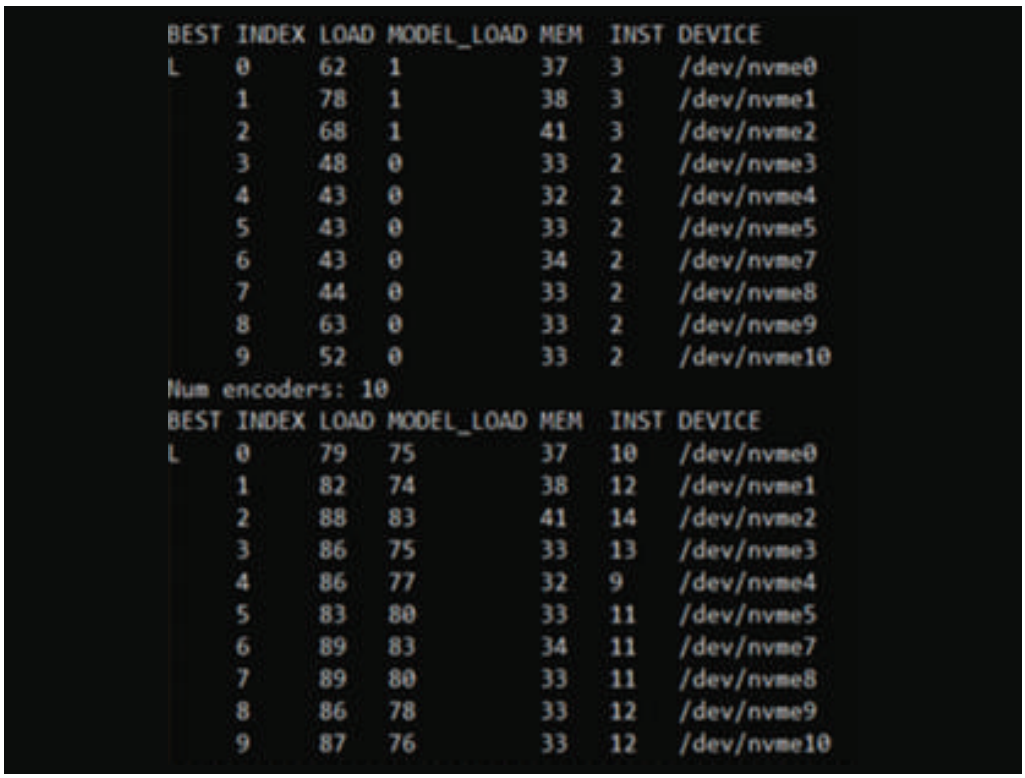
Figure 2. Three ladders encoding at 30 fps using x264 and the veryfast preset.

A look at the CPU utilization shown in the Top utility on the bottom of Figure 2 reveals why. With 32 total cores, the system has an available 3200% of CPU resources. The combined utilization of the three instances shown in the figure exceeded 2600% or just under nine cores each to produce the three 30 fps streams. With only about six cores left, the system lacked the resources to produce another encoding ladder at the full 30 fps.

Test Results

Using this procedure, the Dell system encoding with the CPU only produced a single ladder with x264 and the medium preset, and the three ladders shown in Figure 2 with x264 and the veryfast preset. The AWS instance produced one ladder with the medium preset and four ladders with the veryfast preset. In contrast, when utilizing the ten T408s, the Dell system produced 23 simultaneous ladders.

Figure 3 shows decode (top) and encoder (bottom) utilization of the T408s when producing the 23 ladders. During this trial, CPU utilization per FFmpeg instance averaged about 42% for each encoding ladder for a total load of under 1000% (23*42%=966%). Since 32-cores/3200% were available on this system, this left sufficient resources for file I/O and other encoding-related operations. We discuss more about the operational aspects of the T408-based encoding farm and how to configure the server that houses it in the implementation section below.



BEST	INDEX	LOAD	MODEL_LOAD	MEM	INST	DEVICE
L	0	62	1	37	3	/dev/nvme0
	1	78	1	38	3	/dev/nvme1
	2	68	1	41	3	/dev/nvme2
	3	48	0	33	2	/dev/nvme3
	4	43	0	32	2	/dev/nvme4
	5	43	0	33	2	/dev/nvme5
	6	43	0	34	2	/dev/nvme7
	7	44	0	33	2	/dev/nvme8
	8	63	0	33	2	/dev/nvme9
	9	52	0	33	2	/dev/nvme10

Num encoders: 10

BEST	INDEX	LOAD	MODEL_LOAD	MEM	INST	DEVICE
L	0	79	75	37	10	/dev/nvme0
	1	82	74	38	12	/dev/nvme1
	2	88	83	41	14	/dev/nvme2
	3	86	75	33	13	/dev/nvme3
	4	86	77	32	9	/dev/nvme4
	5	83	80	33	11	/dev/nvme5
	6	89	83	34	11	/dev/nvme7
	7	89	80	33	11	/dev/nvme8
	8	86	78	33	12	/dev/nvme9
	9	87	76	33	12	/dev/nvme10

Figure 3. T408 utilization when producing 23 simultaneous ladders.

Once we ascertained the number of simultaneous streams each encoding scheme produced, we divided this into 100 to compute the number of servers required to produce the 100 simultaneous streams. This yielded the data shown in Table 3.

Ladders Per Server - H.264	Server Cost	T408 Cost	System Cost	Ladders	Servers Needed	CAPEX / Ladder
x264 Medium - Dell Server	$3,200	-	$3,200	1	100	$3,200
x264 Veryfast - Dell Server	$3,200	-	$3,200	3	34	$1,067
x264 Medium - AWS Graviton2	NA		NA	1	100	
x264 Veryfast - AWS Graviton2	NA		NA	4	25	
T408 (10 modules) - Dell Server	$3,200	$3,000	$6,200	23	5	$270

Table 3. The number of servers needed to produce 100 simultaneous streams using the three encoding techniques.

Note that the actual number of ladders that each production scheme can produce at 30fps will vary based upon the encoding ladder and the content, including differences in resolution, the number of rungs, frame rate, and the encoding complexity of the video itself. However, encoding even three fewer simultaneous ladders with the T-408-based solution would still enable five servers to produce the 100 target ladders, so there would be no change in the CAPEX, OPEX, and carbon emissions reported below if producing with the T408-based schema.

In general, as compared to hardware, software encoding experiences more significant performance variations when dealing with different content, so the industry best practice is to maintain a 25% capacity buffer as the minimum, and 50% is also very often seen. Due to the more consistent performance of ASICs, the T408-based system only needs a 5% buffer, which means even greater real-world savings.

Comparative Quality – H.264

Let's take a quick look at output quality. By way of background, when hardware-based H.264 encoding first debuted, output quality trailed that produced by software encoding techniques, often by a significant margin. Today, at least as it relates to the T408, quality won't be an issue. You see this in Table 4, which shows that the T408-encoded stream rated best over both x264 medium and x264 veryfast with the 1080p Meridian test clip used for this study.

Quality – 1080p @ 5 Mbps – H.264	VMAF	PSNR
x264 Medium	95.80	44.40
x264 VeryFast	94.36	44.14
T408	95.95	44.88

Table 4. VMAF and PSNR values for the top-rung of the encoding ladder (1080p@5 Mbps).

As with throughput, comparative quality will vary based upon multiple factors. A more detailed comparison of NETINT's ASIC-based video transcoders showing equivalent or better quality to software-based codecs/presets used for live encoding and transcoding is available upon request.

CAPEX and OPEX Comparisons

Looking at Table 3, it's clear that few engineers will attempt to produce live streams with software only encoding using x264 and the medium preset - it's simply too expensive. So, we focused our economic comparison on the difference between producing the 100 streams with x264 and the veryfast preset and the T408 video transcoder.

For the x264/veryfast comparison we priced two alternatives, encoding via AWS, and buying servers and running them in a colocation facility. For the T408 production, we priced buying the necessary servers and installing them in a colocation facility. In all three instances we computed the three-year cost total for CAPEX and OPEX.

x264/veryfast - AWS

Again, the C6g.8xlarge instance tested produced four simultaneous encoding ladders encoding with FFmpeg using the x264 codec and the veryfast preset. Accordingly, you would need 25 instances to produce the 100 simultaneous target streams.

AWS offers an estimator, to estimate the cost, you enter the number of cores (32), the number of servers (25), estimated utilization (100%) and the commitment period (three-years), and AWS provides a monthly estimate. As shown in Table 5, this was $8,653, which we multiplied by 12 for the yearly cost, and multiplied the yearly cost by three for the three-year total of $311,490.

Cost - x264/veryfast - AWS Graviton2 – H.264	Cores	Servers Needed	Per Month	Per Year	Three Years
Servers	32	25	$8,653	$103,830	$311,490
Total Cost - 3 years	$311,490				

Table 5. Three-year cost for producing 100 simultaneous encoding ladders with the x264 codec/veryfast preset using AWS.

x264/veryfast - Buy Servers and Co-Locate

The next option required buying 34 servers, which we assumed would cost $3,200 each (the actual cost of the Dell server) and running them from a colocation facility that would charge $69/month per server. We priced this option using a colocation facility rather than on-premise because colocation costs are widely available and reasonably consistent while internal housing costs vary by company, location, and accounting method and practices.

As shown in Table 6, these price and cost estimates produced a CAPEX charge of $108,800, and a three-year OPEX charge of $84,456, totaling $193,256 for the three-year period.

Cost x264/veryfast - Buy Servers and Colocate – H.264		Servers Needed	Cost Per Month	Rental Per Month	Per Year (x12)	Three Years (x3)
Cores	32	34	$69	$2,346	$28,152	$84,456
Computer cost	$3,200					
Quantity	34					
Total CAPEX	$108,800					
Total OPEX	$84,456					
Total Cost - 3 years	$193,256					

Table 6. Three-year cost for producing 100 simultaneous encoding ladders with the x264 codec/veryfast preset by buying and colocating the servers.

T408 - Buy Servers and Co-Locate – H.264

Table 7 shows the three-year cost of purchasing five 32-core servers with ten T408 cards each and running them from a colocation facility. This produced CAPEX of $31,000, OPEX of $12,420, and a three-year total of $43,420.

Cost - T408 - Buy Servers and Colocate – H.264		Servers Needed	Cost Per Month	Rental Per Month	Per Year (x12)	Three Years (x3)
System cost (32-core)	$6,200	5	$69	$345	$4,140	$12,420
Number required	5					
Total CAPEX	$31,000					
Total OPEX	$12,420					
Total cost - 3 years	$43,420					

Table 7. Three-year cost for producing 100 simultaneous encoding ladders by purchasing and collocating fiveT408-equipped servers.

Table 8 compares the three-year cost for all three schemes. The T408-based option represents a 86% cost reduction over encoding with AWS and a 78% savings over purchasing and collocating servers.

Cost Summary – H.264	Three Year Cost	T408 Savings
AWS, Graviton2 - CPU-only	$311,490	86%
Dell server, CPU-only - Buy/Co-Locate	$193,256	78%
Dell Server, T408 - Buy/Co-Locate	$43,420	

Table 8. Three-year cost comparison for H.264.

Carbon Emissions

Now we turn our attention to carbon emissions which are presented in Table 9. In the table, the Dell Server - CPU only and Dell Server - T408 watts/hour are actual measurements on the test system during operation. To estimate the AWS server watts/hour, we reduced the CPU-only Dell Server number by 60%, which is the savings that Amazon claims that Graviton2 CPUs provide over other CPUs. In all three cases, we multiplied this by the number of servers, then hours, days, and years, to compute the three-year power consumption total.

Carbon Emissions – H.264	AWS Graviton2	Dell Server - CPU Only	Dell Server - T408
Server Watts/Hour	116	289	287
Number of servers	25	34	5
Total Watts/hour - server	2,890	9,826	1,435
T408 (10x 7 watts)			70
Total Watts/hour	2,890	9,826	1,435
Total Kilowatts/hour	3	10	1
Per day (24x)	69	236	34
Per year (365x)	24,692	83,953	12,261
Three year total	74,076	251,860	36,782
Metric tons of CO2 - per EPA	32	109	15.9

Table 9. Three-year carbon emission comparisons.

To compute metric tons of CO2, we used the EPA Greenhouse Gas Equivalencies Calculator available here. To estimate the CO2 emissions in the calculator, you enter in the total kilowatt hours used and the calculator displays the metric tons equivalent of greenhouse gas emissions. The T408-based option represents a 50% savings over AWS (assuming Amazon's 60% savings estimate is accurate) and an 85% reduction as compared to purchasing and running 34 servers.

HEVC Transcoding

The next section performs a similar analysis with HEVC transcoding. As you'll see, with CPU-only transcoding, the complexity of HEVC significantly reduces throughput, increasing the cost per ladder and power consumption along with carbon emissions considerably over H.264.

In contrast, because of the operational efficiency of the T408 video transcoder, throughput actually increases over H.264 since the encoding ladder utilizes fewer rungs. This translates to dramatically lower CAPEX, OPEX, and carbon emissions as compared to CPU-only encoding in AWS or using the Dell server in a colocation scenario.

As with H.264, in this section, we present the costs and carbon emissions incurred to produce 100 simultaneous live HEVC encoding ladders under the following three scenarios:

1. Encoding with FFmpeg using x265 on a 32-core AWS instance.
2. Encoding with FFmpeg using x265 on the Dell server.
3. Encoding with ten NETINT T408 video transcoders on the same Dell server.

Note that the c6g.8xlarge Graviton2-based system used for H.264 proved surprisingly inefficient with x265, failing to produce even a single x265 ladder using the ultrafast preset. AWS offers two similarly configured systems (16-core/32-thread), one using Intel processors (c6i.8xlarge) and the other AMD (c6a.8xlarge), which both produced two full x265 ladders using the ultrafast preset. The AMD system was slightly cheaper, so we used that in our analysis.

AWS doesn't specifically identify the processors used in the c6a.8xlarge, but states here that they are "3rd generation AMD EPYC processors." The 16-core configuration appears to correspond to the AMD EPYC 7313 CPU, which is a 16-core third generation model AMD EPYC CPU and draws 155 watts, the figure we used for power consumption. Otherwise, though system pricing remained the same for the Dell system, we updated the power consumption which did change slightly for HEVC processing for both CPU-only and T408-based transcoding.

Assumptions - HEVC

Here is the HEVC encoding ladder used for our tests, with command strings presented at the end of this document.

- 1080p @ 3.5 Mbps
- 1080p @ 1.8 Mbps
- 720p @ 1 Mbps
- 360p @ 500 kbps

Table 10 shows a three-year summary of the three approaches (also presented in Table 2). As you can see, the efficiency of ASIC-based transcoding delivers even more impressive cost savings and environmental benefits with advanced codecs like HEVC.

Encoding Scheme - HEVC	Three Year Cost	T408 Savings	Carbon Emissions	T408 Savings
AWS, AMD EPYC 7313 - CPU-only	$713,484	95%	113	88%
Dell Server, CPU-only - Buy/Co-Locate	$568,400	94%	325	96%
Dell Server, T408 - Buy/Co-Locate	$34,736	---	13	NA

Table 10. Three-year cost summary for producing 100 HEVC live encoding ladders for 24/7 operation.

Test Description – HEVC

As with H.264, we tested to determine how many full HEVC encoding ladders each system could produce, and measured output quality. We also tracked energy consumption and translated this to carbon emissions using the same procedures as for H.264.

As shown in Table 11, the Dell server running FFmpeg produced only a single x265/ultrafast ladder, which means 100 systems to produce the 100 target live broadcasts and a huge cost per ladder. The AWS system running the newer AMD CPU eked out only two ladders, requiring 50 instances to meet the target.

When deploying the T408s, the Dell system produced 27 simultaneous ladders allowing four systems to meet the target with 8 streams to spare. This produced a cost per stream that was 93% lower than the Dell server/CPU-only approach.

With the T408s, note that CPU utilization for each FFmpeg instance averaged around 32%, or about 1% of the 3200% available on the 32-core Dell system. Together, all 27 instances consumed about 900%, about 28% of available resources. You shouldn't need a particularly powerful system to house the T408s, even when producing HEVC, and power consumption of that system should be modest, as our tests below demonstrate.

Ladders Per Server - HEVC	Server Cost	T408 Cost	Total Cost	Ladders	Servers Needed	CAPEX per Ladder
Dell Server, CPU-only - Buy/Co-Locate	$3,200	-	$3,200	1	100	$3,200
AWS, AMD EPYC 7313 - CPU-only	NA		NA	2	50	NA
Dell Server, T408 - Buy/Co-Locate	$3,200	$3,000	$6,200	27	4	$230

Table 11. The number of servers needed to produce 100 simultaneous streams using the three encoding techniques.

Comparative Quality

Table 12 reports comparative quality for the Meridian test clip used in these trials. We include x265/Medium quality as a reference even though no CPU-only alternative could deliver even a single encoding ladder using that preset. Though the scores are all relatively close, the T408 scored best in PSNR comparisons and ahead of ultrafast in VMAF.

Quality – 1080p @ 3.5 Mbps - HEVC	VMAF	PSNR
x265 Medium (32-core)	96.03	44.15
x265 Ultrafast (32-core)	95.36	43.21
T408 (32-core)	95.52	44.19

Table 12. VMAF and PSNR values for the top-rung of the encoding ladder (1080p@3.5 Mbps).

While comparative encoding quality will vary from clip to clip, with HEVC transcoding, the T408 will almost always deliver better than x265/ultrafast quality that will approach or even exceed x265/medium quality.

CAPEX and OPEX Comparisons

Here are the CAPEX and OPEX computations.

x265/ultrafast - AWS

We used the AWS estimator to compute a cost per month for the 50 c6a.8xlarge systems of $19,819 per month, which includes a three-year commitment. This sets up the rest of the calculations shown in Table 13 and a three-year total of $713,484.

Cost - x265/Ultrafast via AWS - HEVC	Cores	Servers Needed	Per Month	Per Year (12x)	Three Years
Servers	32	50	$19,819	$237,828	$713,484
Total Cost - 3 years	$713,484				

Table 13. Three-year cost for producing 100 simultaneous encoding ladders with the x265 codec/ultrafast preset using AWS.

x265/ultrafast - Buy Servers and Co-Locate

The next option required buying 100 servers, which we assumed would cost $3,200 each, and running them from a colocation facility that would charge $69/month per server. The combined CAPEX and OPEX charge here, as shown in Table 14, was $568,400.

Cost - x265/Ultrafast - Buy Servers and Colocate - HEVC		Servers Needed	Cost Per Month	Rental Per Month	Per Year (x12)	Three Years (x12)
Cores	32	100	$69	$6,900	$82,800	$248,400
Computer cost	$3,200					
Quantity	100					
Total CAPEX	$320,000					
Total OPEX	$248,400					
Total Cost - 3 years	$568,400					

Table 14. Three-year cost for producing 100 simultaneous encoding ladders with the x265 codec/ultrafast preset by buying and colocating the servers.

T408 - Buy Servers and Co-Locate

Table 15 shows the three-year cost of purchasing four 32-core servers with ten T408 cards each and running them from a colocation facility. This produced CAPEX of $24,800, OPEX of $9,936, and a three-year total of $34,736.

T408 - Buy Servers and Co-Locate - HEVC		Servers Needed	Cost Per Month	Rental Per Month	Per Year (x12)	Three Years (x3)
System cost (32-core)	$6,200	4	$69	$276	$3,312	$9,936
Number required	4					
Total CAPEX	$24,800					
Total OPEX	$9,936					
Total Cost - 3 years	$34,736					

Table 15. Three-year cost for producing 100 simultaneous encoding ladders by purchasing and collocating four T408-equipped servers.

Table 16 shows the carbon emissions comparisons. As mentioned above, the c6a.8xlarge AWS instance used in our tests appears to be an AMD EPYC 7313 CPU, which draws 155 watts by itself. Drawing data from this source, we added 25 watts for the motherboard, and 24 watts for the 64 GB of RAM (see here). Ignoring other components, this came to 204 watts for the complete system. The Dell Server results were actual numbers produced in our labs. As before, to compute metric tons of CO2, we used the EPA Greenhouse Gas Equivalencies Calculator available here.

Carbon Emissions – HEVC	AWS	Dell Server - CPU Only	Dell Server – with T408s
Server Watts/Hour	204	293	293
Number of servers	50	100	4
Total Watts/hour - server	10,200	29,300	1,172
Total Watts/hour	10,200	29,300	1,172
Total Kilowatts/hour	10	29	1
Per day (24x)	245	703	28
Per year (365x)	87,149	250,339	10,014
Three-year total	261,446	751,018	30,041
Metric tons of CO2 - per EPA	113	325	13

Table 16. Three-year carbon emission comparisons.

As you can see, with both H.264 and particularly HEVC, a T408-based transcoding workflow delivers substantial economic and environmental benefits at the same or better quality than software-only encoding using presets typically used for live transcoding.

Choosing a CPU/Server for the T408s

We'll conclude with a look at choosing a server for the T408s and how to interface with and operate the transcoders. In terms of operating system, the T408 transcoding software supports the following:

- OS: Ubuntu 16.04.3 LTS; kernel: 4.10.0-28-generic
- OS: Ubuntu 16.04.3 LTS; kernel: 4.15.0-64-generic
- OS: Ubuntu 18.04.2 LTS; kernel: 4.15.0-45-generic
- OS: CentOS 7.2.1511; kernel: 3.10.0-327.el7.x86_64
- OS: CentOS 7.5.1804; kernel: 3.10.0-862.11.6.el7.x86_64
- OS: CentOS 7.6.1810; kernel: 3.10.0-957.el7.x86_64

The minimum requirements for the system housing the T408s is an Intel i5 CPU or equivalent with 4GB DDR3 or DDR4. However, you may need a more powerful CPU depending upon your specific transcoding application.

To explain, the T408 can decode incoming H.264/HEVC streams via onboard decoders and performs all encoding onboard. However, scaling the source video to lower resolutions for different rungs on the encoding ladder is performed by the host CPU.

For this reason, your selection of the host CPU(s) for the server should consider the specific tasks the system will perform. In a gaming or other interactive environment, where you are inputting a single stream and outputting a single stream at the same resolution, host CPU requirements should be limited, and a modest CPU should perform well.

In contrast, if your application involves creating full encoding ladders from HD or particularly 4K source videos, a more powerful CPU will increase overall system throughput. For example, we tested the same H.264 encoding ladder on a system with 64-cores and ten T408s and the system produced 30 simultaneous ladders.

All that said, the 32-core Dell system used in our testing cost $3,200 and produced 23 H.264 five-rung ladders and 27 4-rung HEVC ladders with over 50% of CPU capacity to spare. In most cases, you won't need a particularly powerful system to host the T408s.

The bottom line is that it's hard to predict which CPU configuration will perform best in your T408 host. During your pre-deployment testing, you should plan to test different CPUs to find the optimal configuration.

Operating the T408 System

As mentioned above, NETINT offers highly efficient FFmpeg and GStreamer SDKs that allow operators to apply an FFmpeg/libavcodec or GStreamer patch to complete the integration.

In the FFmpeg implementation, the libavcodec patch on the host server functions between the T408 NVMe interface and the FFmpeg software layer, allowing existing FFmpeg-based video transcoding applications to control T408 operation with minimal changes.

The T408 device driver software includes a resource management module that tracks T408 capacity and usage load to present inventory and status on available resources and enable resource distribution. User applications can build their own resource management schemes on top of this resource pool or let the NETINT server automatically distribute the decoding and encoding tasks.

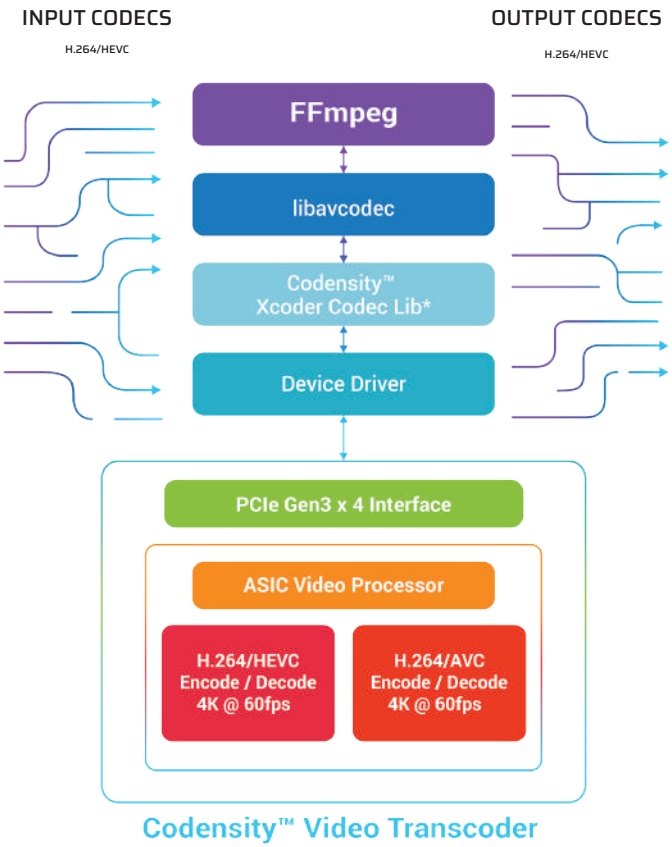


Figure 4. The T408 transcoding architecture.

In this mode, users simply launch multiple transcoding jobs, and the device driver will automatically distribute the decode/encode tasking among the available resources. We used this mode of automatic distribution for all T408 testing reported in this document.

Or, users can assign different decoding and encoding tasks to different T408 devices, and even control which streams are decoded by the host CPU or a T408. With these and similar controls, users can most efficiently balance the overall transcoding load between the T408s and host CPU and maximize throughput. Note that these resource management functions have been integrated into FFmpeg to simplify operation.

Summary and Conclusion

Overall, with H.264 and particularly HEVC, the T408 delivers dramatic reductions in CAPEX and OPEX and significantly cuts carbon emissions as compared to other production alternatives, all while producing quality similar to CPU-only transcoding. With a highly functional resource management schema and FFmpeg and GStreamer integrations, implementing a T408-based solution should be fast and simple for most streaming producers.

Appendix I. Command Line Arguments – H.264

x264 Medium

```
ffmpeg  -y -re -i /ramdisk/Meridian.mp4  \
-y -an -c:v libx264 -preset medium -threads 8  -b:v 5M -bufsize 5M -maxrate 5M  -g 60  -f null -  \
-y -an -c:v libx264 -preset medium -threads 8  -b:v 3.5M -bufsize 3.5M -maxrate 3.5M  -g 60  -f null -  \
-y -s 1280x720 -an -c:v libx264 -preset medium -threads 8 -b:v 2M -bufsize 2M -maxrate 2M  -g 60  -f null -  \
-y -s 960x540  -an -c:v libx264 -preset medium -threads 8 -b:v 1M -bufsize 1M -maxrate 1M  -g 60  -f null -  \
-y -s 640x360  -an -c:v libx264 -preset medium -threads 8 -b:v .6M  -bufsize .6M  -maxrate .6M  -g 60  -f null -
```

x264 Very Fast

```
ffmpeg  -y -re -i /ramdisk/Meridian.mp4  \
-y -an -c:v libx264 -preset veryfast -threads 8  -b:v 5M -bufsize 5M -maxrate 5M  -g 60  -f null -  \
-y -an -c:v libx264 -preset veryfast -threads 8  -b:v 3.5M -bufsize 3.5M -maxrate 3.5M  -g 60  -f null -  \
-y -s 1280x720 -an -c:v libx264 -preset veryfast -threads 8 -b:v 2M -bufsize 2M -maxrate 2M  -g 60  -f null -  \
-y -s 960x540  -an -c:v libx264 -preset veryfast -threads 8 -b:v 1M -bufsize 1M -maxrate 1M  -g 60  -f null -  \
-y -s 640x360  -an -c:v libx264 -preset veryfast -threads 8 -b:v .6M  -bufsize .6M  -maxrate .6M  -g 60  -f null -
```

NETINT – H.264

```
ffmpeg -re -c:v h264_ni_dec -i /ramdisk/Meridian.mp4 \
-y  -an  -c:v h264_ni_enc -xcoder-params RcEnable=1:bitrate=5000000:cbr=1:intraPeriod=60  -f null -  \
-y  -an  -c:v h264_ni_enc -xcoder-params RcEnable=1:bitrate=3500000:cbr=1:intraPeriod=60  -f null -  \
-y -s 1280x720  -an  -c:v h264_ni_enc -xcoder-params RcEnable=1:bitrate=2000000:cbr=1:intraPeriod=60  -f null -  \
-y -s 960x540  -an  -c:v h264_ni_enc -xcoder-params RcEnable=1:bitrate=1000000:cbr=1:intraPeriod=60  -f null -  \
-y -s 640x360  -an  -c:v h264_ni_enc -xcoder-params RcEnable=1:bitrate=600000:cbr=1:intraPeriod=60  -f null -
```

Appendix II. Command Line Arguments – HEVC

x265 Ultrafast

```
ffmpeg -y -re -i  /ramdisk/Meridian.mp4 \
-c:v libx265 -an -preset ultrafast -x265-params keyint=60:min-keyint=60:scenecut=0:bitrate=3500:vbv-max-
rate=3500:vbv-buf-size=3500:open-gop=1 -f null -  \
-c:v libx265 -an -preset ultrafast -x265-params keyint=60:min-keyint=60:scenecut=0:bitrate=1800:vbv-max-
rate=1800:vbv-buf-size=1800:open-gop=1 -f null -  \
-c:v libx265 -an -s 1280x720 -preset ultrafast -x265-params keyint=60:min-keyint=60:scenecut=0:bitrate=1000:vbv-max-
rate=1000:vbv-buf-size=1000:open-gop=1 -f null -  \
-c:v libx265 -an -s 640x360 -preset ultrafast -x265-params keyint=60:min-keyint=60:scenecut=0:bitrate=500:vbv-max-
rate=500:vbv-buf-size=800:open-gop=1 -f null -
```

NETINT – HEVC

```
ffmpeg -y -re -c:v h264_ni_logan_dec  -i  /ramdisk/Meridian.mp4 \
-y -c:v h265_ni_logan_enc  -an  -xcoder-params
RcEnable=1:bitrate=3500000:intraPeriod=60:decodingRefreshType=1  -f null -  \
-y -c:v h265_ni_logan_enc  -an  -xcoder-params
RcEnable=1:bitrate=1800000:intraPeriod=60:decodingRefreshType=1   -f null -  \
-y -s 1280x720 -c:v h265_ni_logan_enc  -an  -xcoder-params
RcEnable=1:bitrate=1000000:intraPeriod=60:decodingRefreshType=1   -f null -  \
-y -s 640x360 -c:v h265_ni_logan_enc  -an  -xcoder-params
RcEnable=1:bitrate=500000:intraPeriod=60:decodingRefreshType=1   -f null -
```