NVIDIA Video Technologies

Abhijit Patait, NVIDIA | GTC 2024, 3/21/2024

Agenda

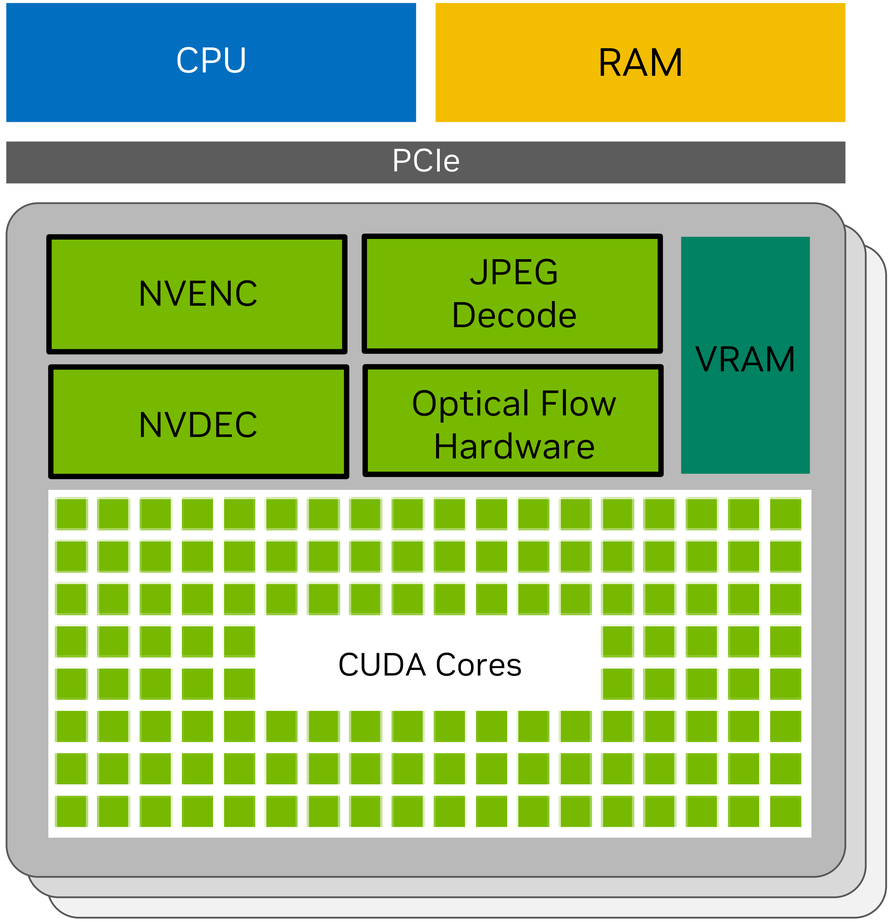
- Overview
- Software Updates
- Video Transcoding on NVIDIA L4
- Resources

Overview: Hardware-Accelerated Video/Image Processing in NV GPUs

NVIDIA Media Processing Hardware

Dedicated hardware for video/image decoding, encoding, optical flow, post-processing

- **Capabilities**
  - NVENC – **Encode** video
  - NVDEC – **Decode** video
  - JPEG decoder – **Decode JPEG** images
  - Optical flow – **Track** pixels
  - Compute/CUDA – **Post-process, train, infer**, …
- **Highly accelerated**
- **Power efficient**
- **Scalable**



Not all features are available in all GPUs. Please check NVIDIA developer zone web site for detailed information

NVIDIA Media Processing Hardware

Capabilities

- **NVDEC – Video Decode**
  - H.264
  - H.265/HEVC
  - AV1
  - VP9
  - Lossless
  - Legacy: MPEG-2, VC-1
  - 8/10-bit
  - YUV 4:2:0, 4:4:4[1]
  - Up to 8K[2]

- **NVENC – Video Encode**
  - H.264
  - H.265/HEVC
  - AV1
  - Lossless
  - 8/10-bit
  - YUV 4:2:0, 4:4:4[1]
  - Up to 8K[2]

- **NVJPEG – JPEG Decode**
  - Up to 8 JPEG parallel decoders[3]
  - Up to 16K × 16K

- **NVOFA – Optical Flow**
  - 8192 × 8192 at 1×1, 2×2, 4×4 granularity
  - Region of interest
  - 600-1800 fps for 1080p

[1] 4:4:4 supported only for HEVC in Turing+ hardware; see NVIDIA developer zone web site for details
[2] Support is codec-dependent
[3] Support is hardware-dependent
Not all features available on all hardware; see NVIDIA developer zone web site for GPU support matrix

NVIDIA Video Technologies

Software Stack

SOFTWARE

GPU video acceleration

FFMPEG Microsoft DirectX Vulkan PyNvVideoCodec

DeepStream, DALI

NPP

PyTorch, TF, Triton

VIDEO CODEC, OPTICAL FLOW SDK

Video Encode and Decode, Optical Flow APIs

CUDA TOOLKIT

APIs, libraries, tools, samples

NVIDIA Driver

HARDWARE

NVENC

Video encode

AV1 H.264 MPEG-4/AVC H.265 HEVC

NVDEC

Video decode

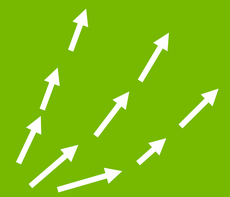
MPEG2 VC-1 H.264 MPEG-4/AVC H.265 HEVC AV1 VP8 VP9

NvJPEG

JPG

OFA

Optical Flow



Software Updates

2022 – Nov

Video Codec SDK 12.0

Ada GPUs
AV1 encoding
8K@60 encode (AV1/HEVC)

2023 – Feb

Vulkan Decode

H.264 and HEVC (GA)

2023 – Apr

Video Codec SDK 12.1

Iterative encoding
Explicit split encode

2023 – Dec

Vulkan Encode

H.264 and HEVC (Beta)

2024 – Feb

Vulkan Updates

Vulkan AV1 Decode (Beta)
Vulkan H.264/HEVC encode (GA)
Nsight profiling tools

2024 – Mar

Video Codec SDK 12.2

15% better HEVC quality

PyNvVideoCodec 1.0

Video Codec SDK Updates

- **SDK 12.1 (*Q2 2023*)**
  - Iterative encoding
  - Explicit split encode
- **SDK 12.2 (*Q1 2024*)**
  - 15% better HEVC encoding
  - Nsight profiling tools
- **PyNvVideoCodec (*Q1 2024*)**
  - Successor to VPF, supported by NVIDIA
- **Vulkan**
  - Video decode
  - Video encode

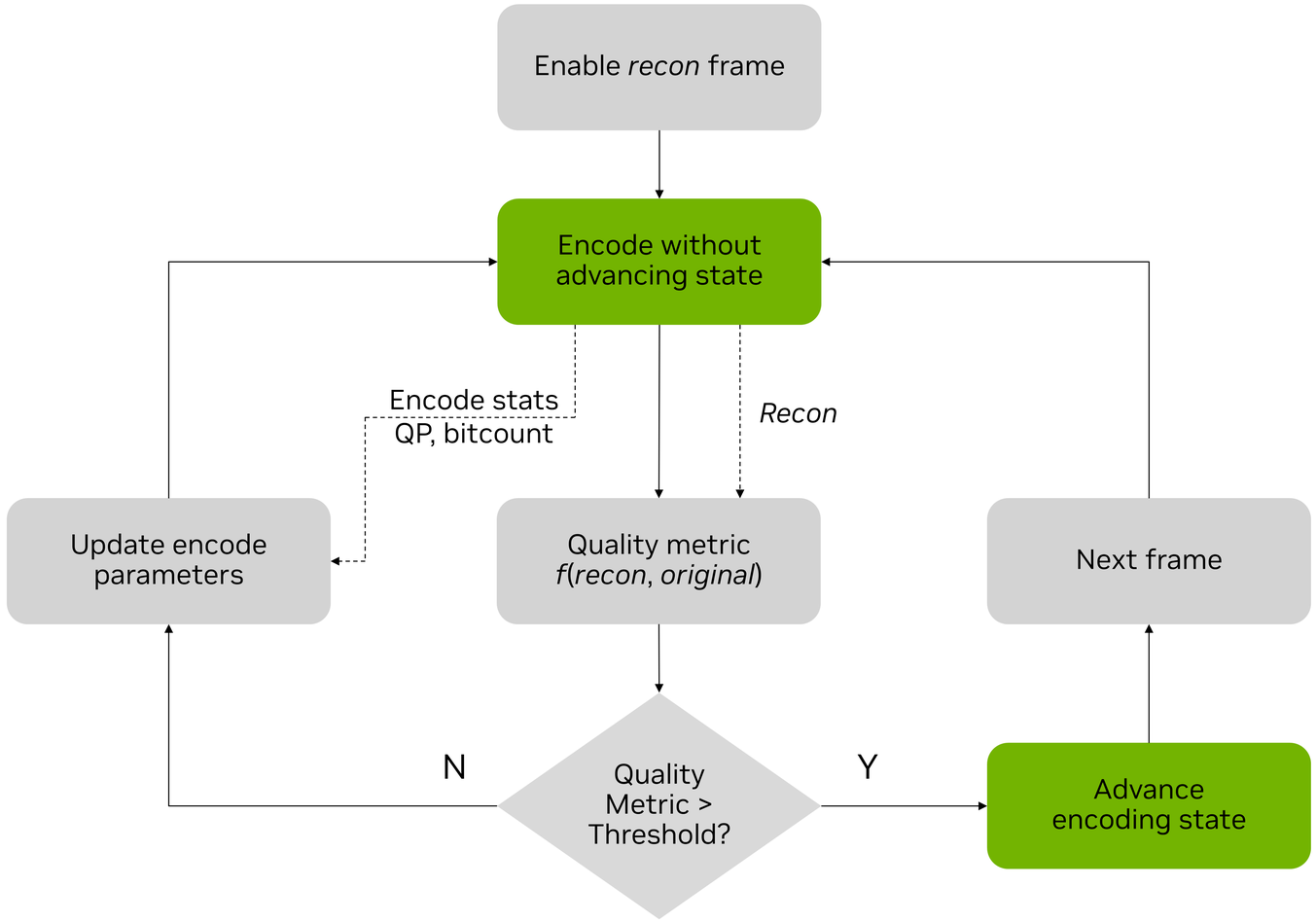
Iterative Encoding

Video encode with *application-defined* quality

Encode a frame iteratively until desired quality is reached

Iterative Encoding

Video encode with *application-defined* quality



- APIs
  - Encoder's *reconstructed* frame
  - Encode w/ or w/o advancing state
  - Key encode statistics
- Quality metric = *f*(*recon, original*)

  = PSNR, SSIM, VMAF,...

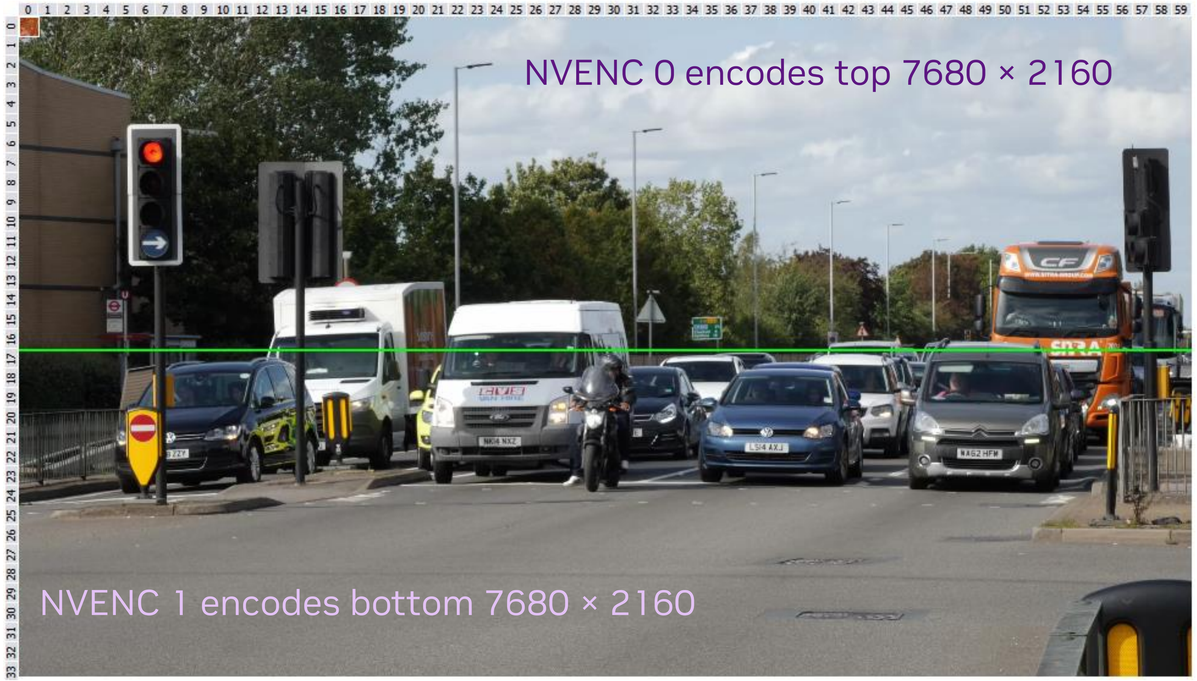
Automatic Split Encoding

Video Codec SDK 12.0

- Available for HEVC & AV1
- 8K60 encoding in real-time with multiple NVENCs
- Automatically switched ON with the following conditions:
  - Frame height:
    - > 2112 for HEVC
    - > 2048 for AV1
  - High performance presets

Preset / Tuning info	p1	p2	p3	p4	p5	p6	p7
High quality	Split	Split	Normal	Normal	Normal	Normal	Normal
Low latency	Split	Split	Split	Split	Normal	Normal	Normal
Ultra-low latency	Split	Split	Split	Split	Normal	Normal	Normal

AV1 encoding of 7680 × 4320



Explicit Split Encoding

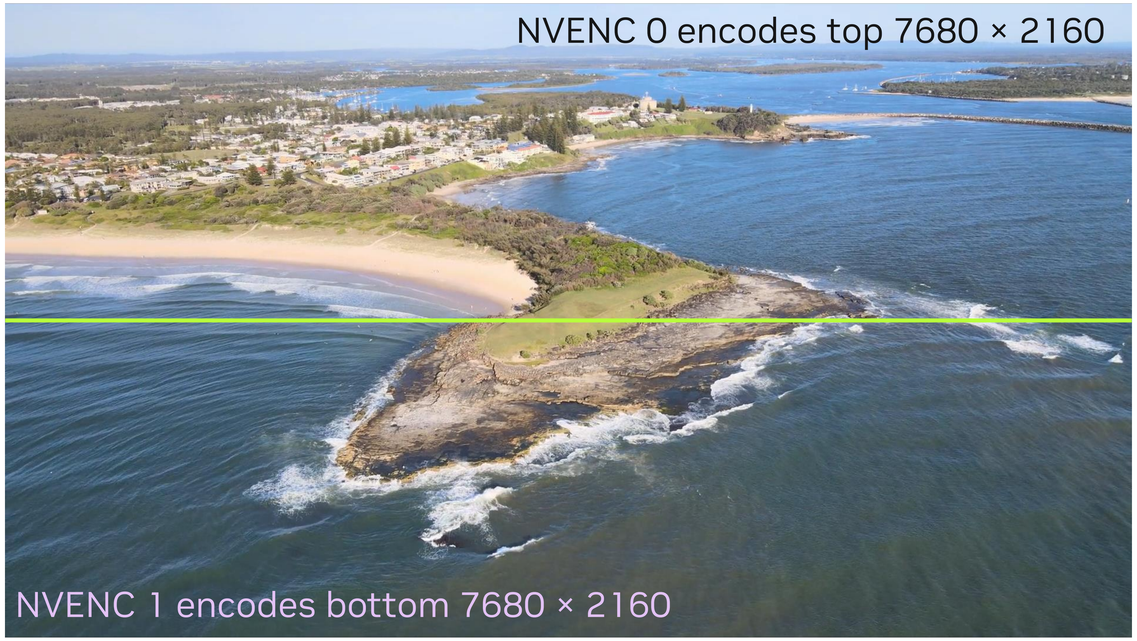
Video Codec SDK 12.1

Leverage multiple encoders to support higher resolutions via API

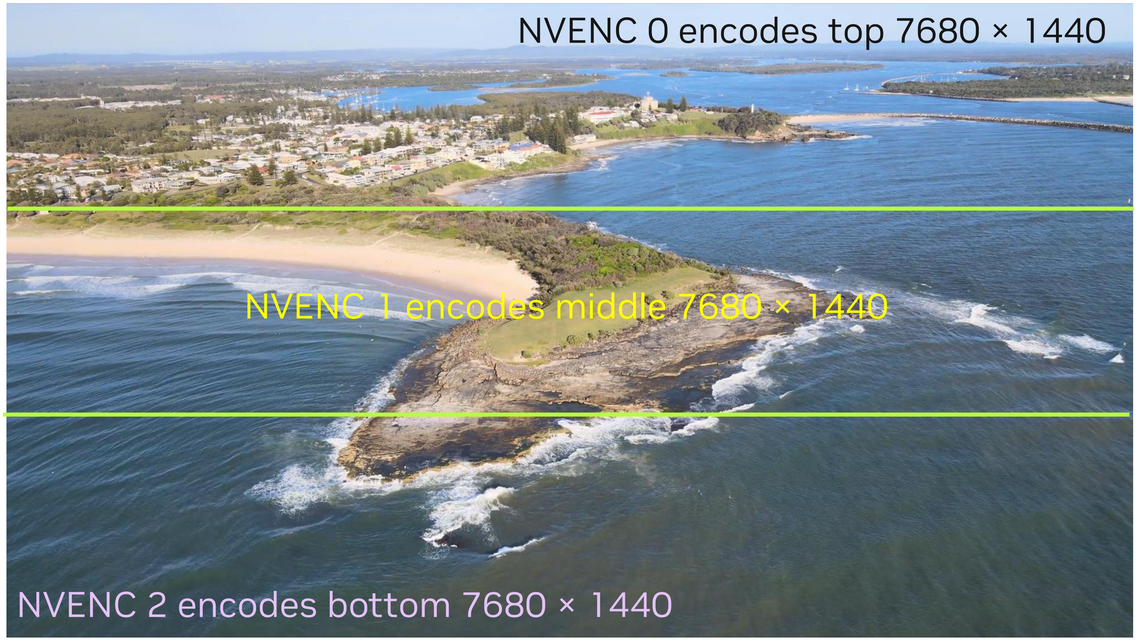
Explicit Split Encoding

Video Codec SDK 12.1

- Control automatic, 2-way, 3-way frame-split with multiple NVENCs (preset-agnostic)
- HEVC & AV1



2 NVENCs



3 NVENCs

Video Codec SDK Updates

- **SDK 12.1 (*Q2 2023*)**
  - Iterative encoding
  - Explicit split encode
- **SDK 12.2 (*Q1 2024*)**
  - 15% better HEVC encoding
  - Nsight profiling tools
- **PyNvVideoCodec (*Q1 2024*)**
  - Successor to VPF, supported by NVIDIA
- **Vulkan**
  - Video decode
  - Video encode

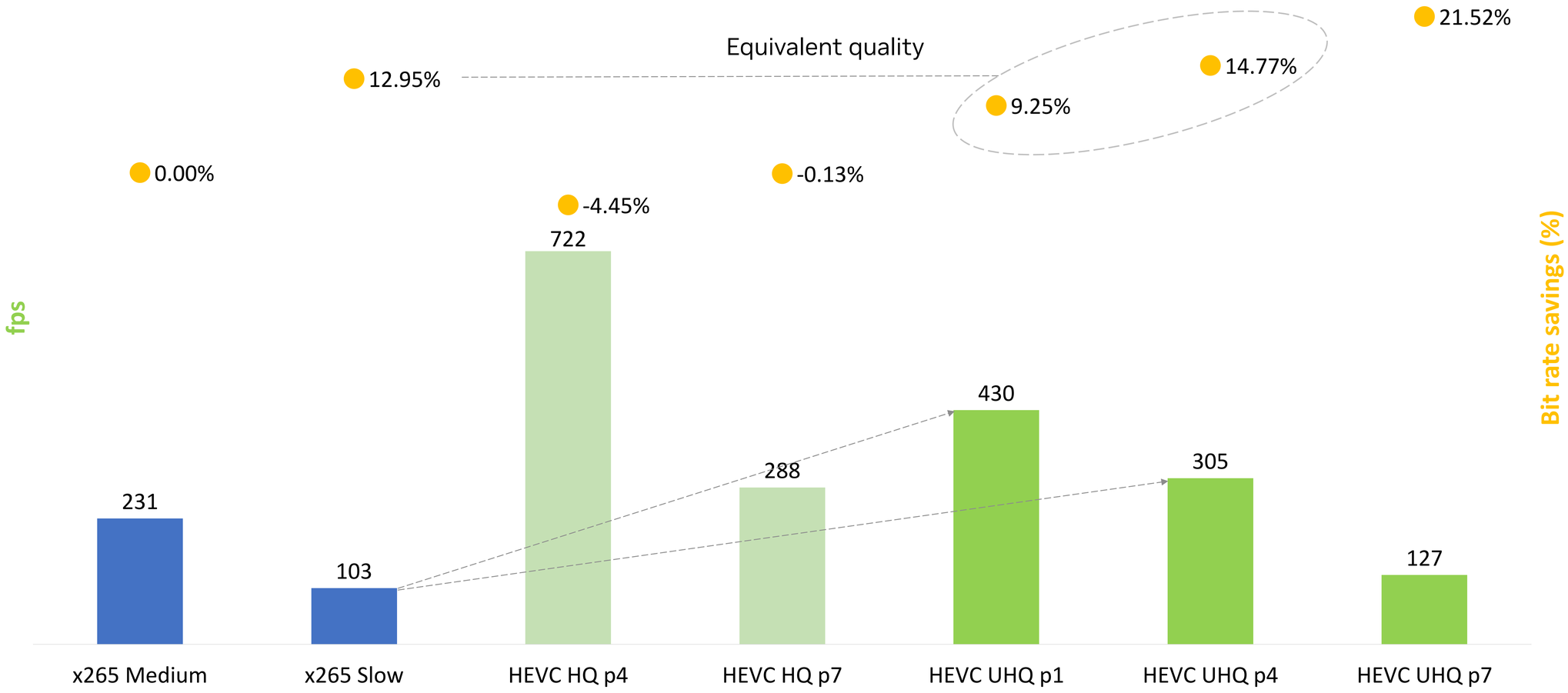
Enhanced HEVC Encode Quality

Quality of *x265 slow* at 3x-4x performance

Courtesy: Pexels.com

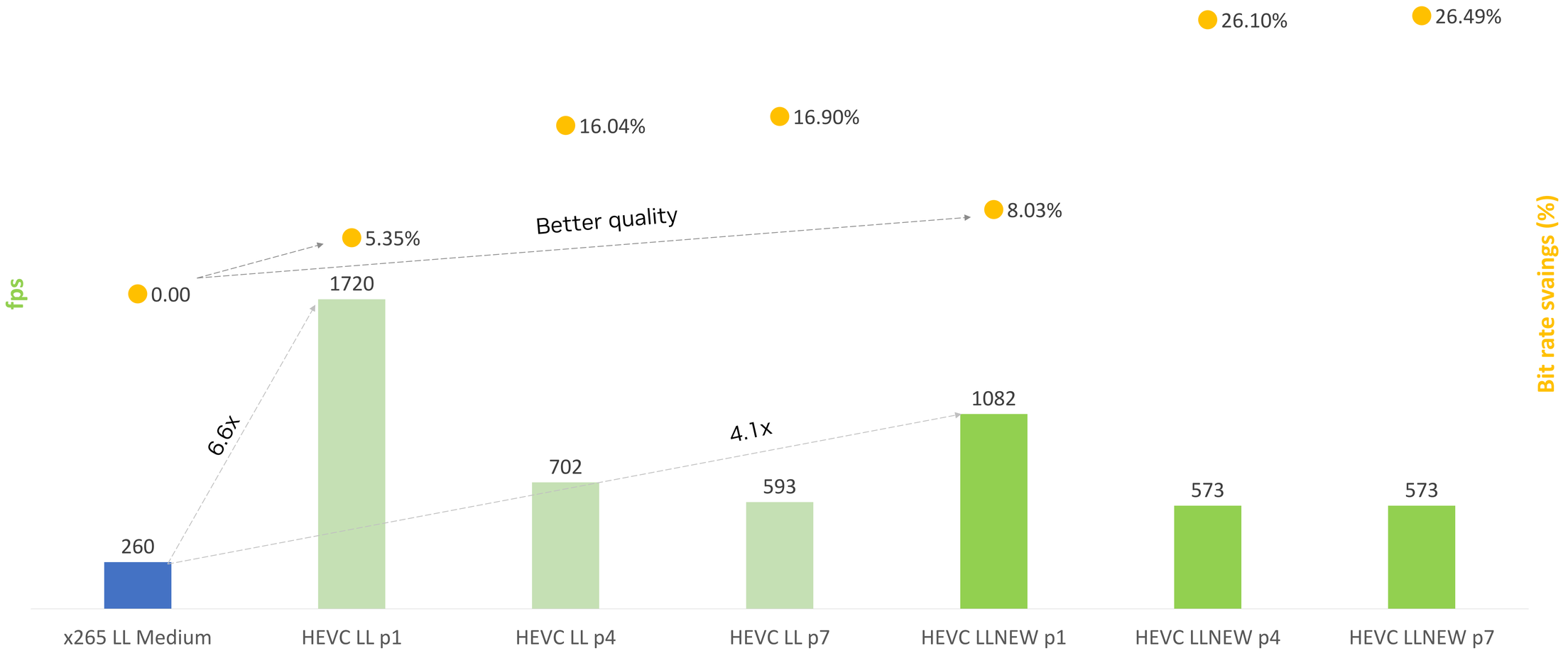
Latency-tolerant HEVC Encoding: NVIDIA L4 *vs* x265 on Intel 8480

Higher is better



Latency-sensitive HEVC Encoding: NVIDIA L4 vs x265 on Intel 8480

Higher is better



LL = Low latency tuning info (SDK 12.1)
LLNEW = Low latency tuning info + Uni-B + High-bit-depth encoding

Enhanced HEVC Encode Quality

Details

Coding Unit Tree

Temporal Filtering

Unidirectional B-frames

High Bit-depth Encoding

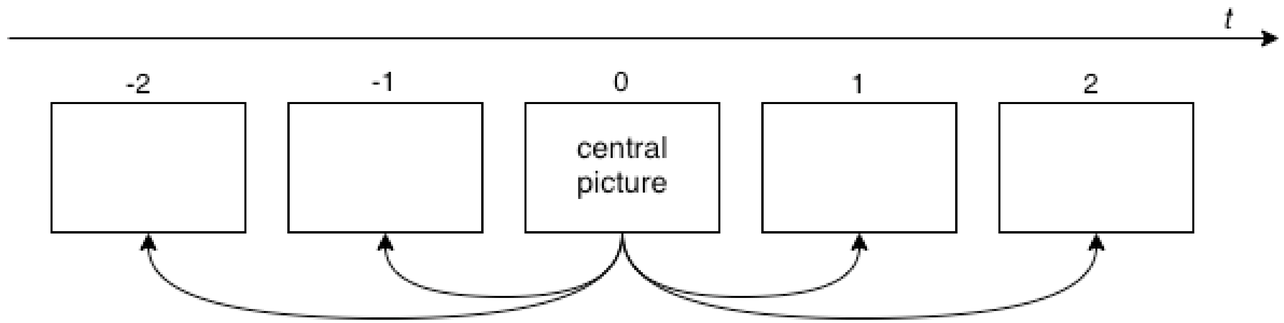
Coding Unit Tree

- **Lookahead statistics**
  - Track block propagation using MVs to adjust QP
  - Better rate control
  - Statistics propagated backwards
- **Multiple Lookahead Levels**
  - Quality/performance tradeoff
- **Improved Benefits with Temporal Filter**

Temporal Filtering

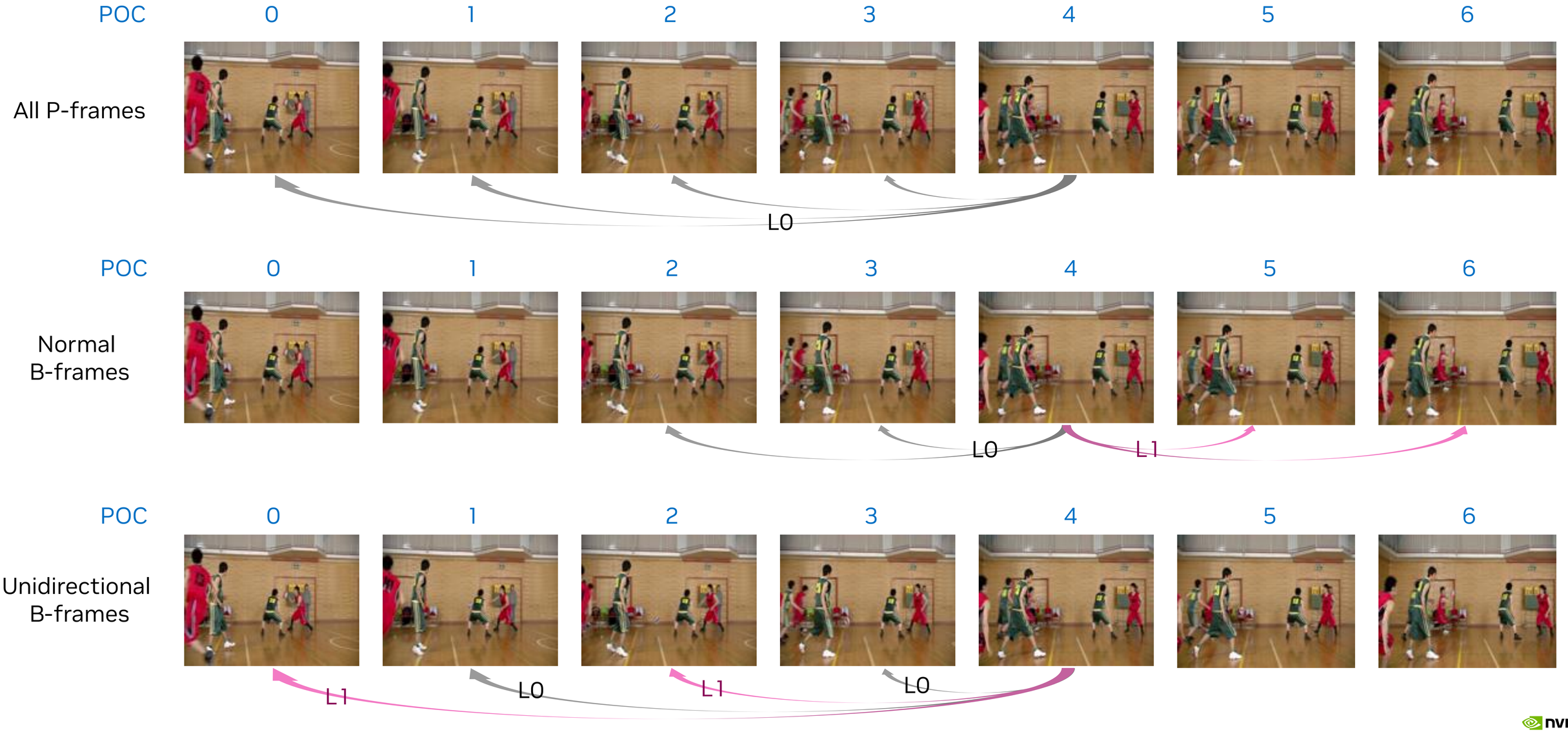
Noise reduction for improved coding efficiency

- Camera introduces noise
- Noise reduces correlation ➔ reduces compression efficiency
- Temporal filtering of adjacent frames reduces camera noise with minimal impact to the content.



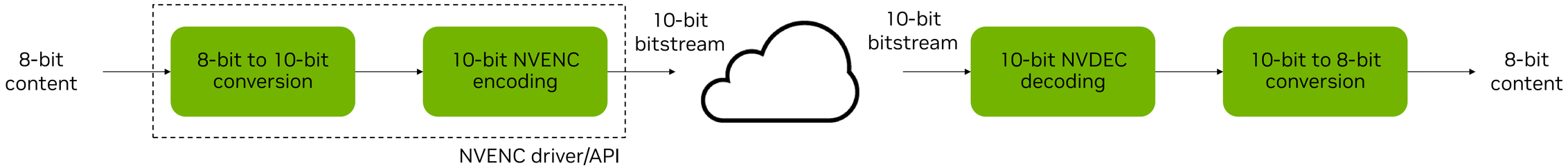
Unidirectional B-Frames

Same latency as P-frames but higher efficiency



Increased Bit Depth Encoding

Encode 8-bit content as 10-bit



- Encoding 8-bit content as 10-bit results into better decorrelation
- Requires end-end control of pipeline

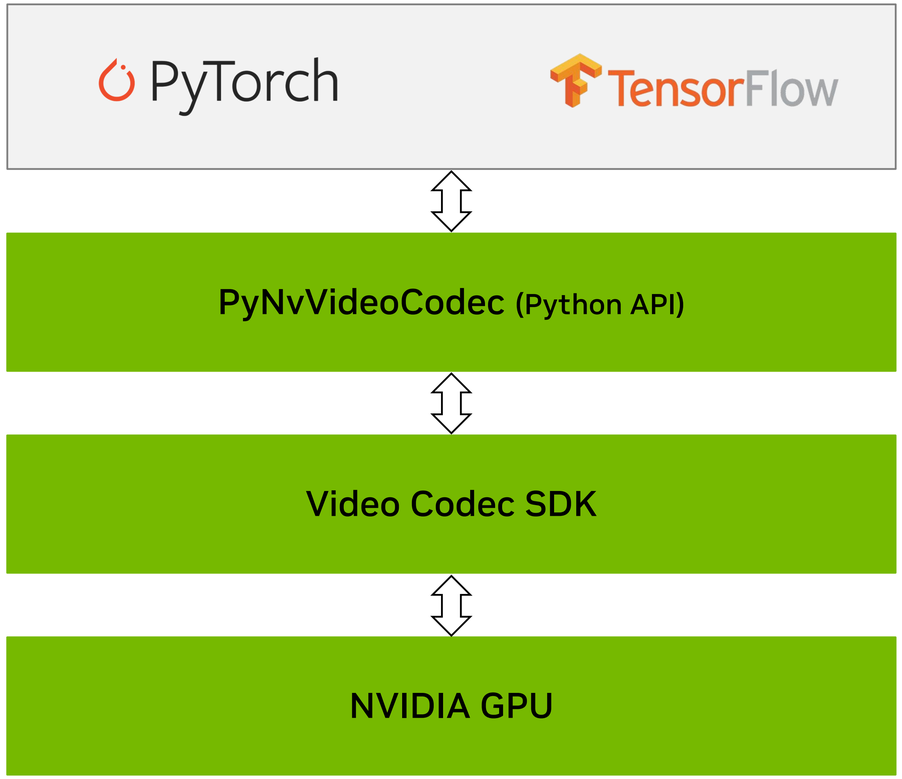
Video Codec SDK Updates

- **SDK 12.1 (*Q2 2023*)**
  - Iterative encoding
  - Explicit split encode
- **SDK 12.2 (*Q1 2024*)**
  - 15% better HEVC encoding
  - Nsight profiling tools
- **PyNvVideoCodec (*Q1 2024*)**
  - NVIDIA-supported Python Video Encode/Decode Package
- **Vulkan**
  - Video decode
  - Video encode

PyNvVideoCodec

Hardware-accelerated video encode and decode for Python applications

- Easy-to-use Python APIs for hardware-accelerated video encode and decode
- Successor of VPF
- Simplified installation using pip install
- Compatible with popular DL frameworks
- CUDA stream support for optimizing throughput



Installing PyNvVideoCodec

Pip install from PyPi

- Ready to use Python WHL.
- Supports popular OS configurations.
- Recommended way.

Steps

1. Open the shell prompt and run following command.

```
pip install pynvvideocodec
```

Download source on NGC and build

- Source code with dependencies.
- Sample applications demonstrating API usage.
- Source code distribution to enable customizations.
- MIT license, built and maintained by NVIDIA.

Steps

1. Download PyNvVideoCodec zip file.
2. Open the shell prompt, go to the folder containing PyNvVideoCodec.zip and run following command.

```
pip install PyNvVideoCodec.zip
```

Video Decode


```python
# Python sample code for decoding video

import PyNvVideoCodec as nvc

demuxer = nvc.CreateDemuxer("input_video.mp4")
decoder = nvc.CreateDecoder(gpuid, demuxer.GetCodecID())

# Iterate through packets and extract decoded frame
for packet in demuxer:
    for decoded_frame in decoder.Decode(packet):
        # Get tensor from decoded frame
        decoded_tensor = torch.from_dlpack(decoded_frame)
```

- Decoded surface as a tensor (*zero copy*).
- CUDA stream and stream-ordered memory allocation support enables overlapping of decoding and inferencing workloads for better throughput.

Video Encode


```
# Python sample code for encoding video

import PyNvVideoCodec as nvc

encoder = nvc.CreateEncoder(1920, 1080, “NV12”)

# Encode raw videoframes from video source one by one
bitstream = encoder.Encode(video_frame)

# At the end, get remaining bitstream from the encoder queue
bitstream = encoder.EndEncode()
```

- `CreateEncoder` API can takes optional encode parameters in the form of key-value pairs for fine-grain control.
- Encode API can take tensor from DL frameworks like PyTorch (CPU or GPU buffer).

Video Transcode


```
# Python sample that decodes video sequence, runs a clamping kernel
# on surfaces(min 0, max 127 and encodes them

import PyNvVideoCodec as nvc
import torch

demuxer = nvc.CreateDemuxer(“input_video.mp4”)
decoder = nvc.CreateDecoder(gpuid, demuxer.GetCodecID())
encoder = nvc.CreateEncoder(1920, 1080, “NV12”)

with open(“output_video.mp4”, "wb") as output_file:
    for packet in demuxer:
        for decoded_frame in decoder.Decode(packet):
            # Post-process decoded_frame as needed such as
            # scale, crop, denoise, etc.
            ...
            # Encode the frame using NVENC
            bitstream = nvenc.Encode(post_processed_frame)
            output_file.write(bytearray(bitstream))
    bitstream = nvenc.EndEncode()
    bitstream = outputfile.write(bytearray(bitstream))
```

- Decoded surface can be accessed as a tensor, processed and passed to encoder.

PyNvVideoCodec Highlights

- Video codecs
  - H.264 (8-bit only)
  - HEVC (8/10-bit)
  - AV1 (8/10-bit)
- Video surface formats
  - NV12
  - YUV 4:2:0
  - YUV 4:4:4
- Data exchange formats - DLPack, CUDA Array Interface
- Platforms
  - Windows 10 and above
  - Linux – Ubuntu 18.04 and above
- Available for download on PyPi and NGC in Q1 2024.

Video Codec SDK Updates

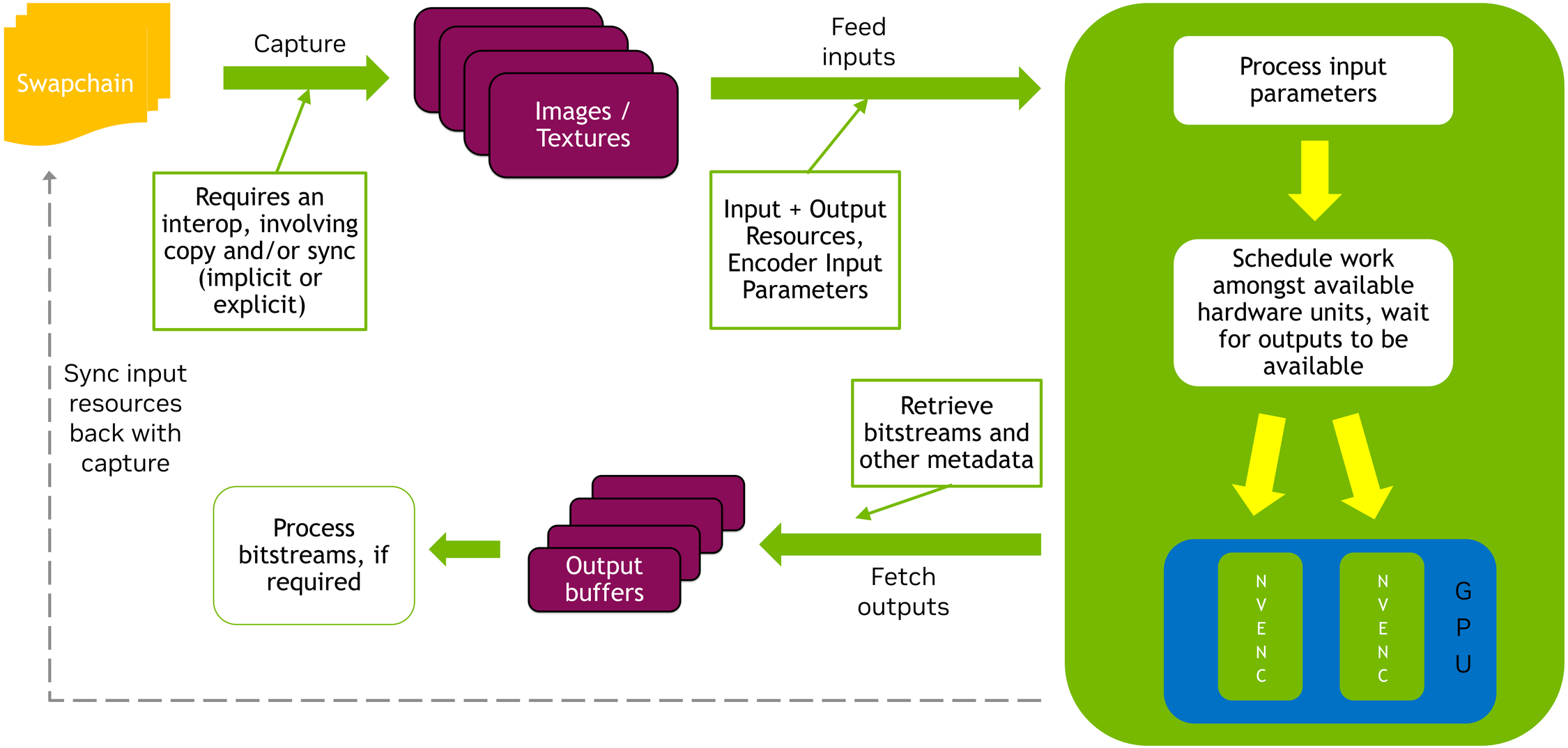
- **SDK 12.1 (*Q2 2023*)**
  - Iterative encoding
  - Explicit split encode
- **SDK 12.2 (*Q1 2024*)**
  - 15% better HEVC encoding
  - Nsight profiling tools
- **PyNvVideoCodec (*Q1 2024*)**
  - Successor to VPF, supported by NVIDIA
- **Vulkan**
  - Decode
  - Encode

Vulkan Video Extensions

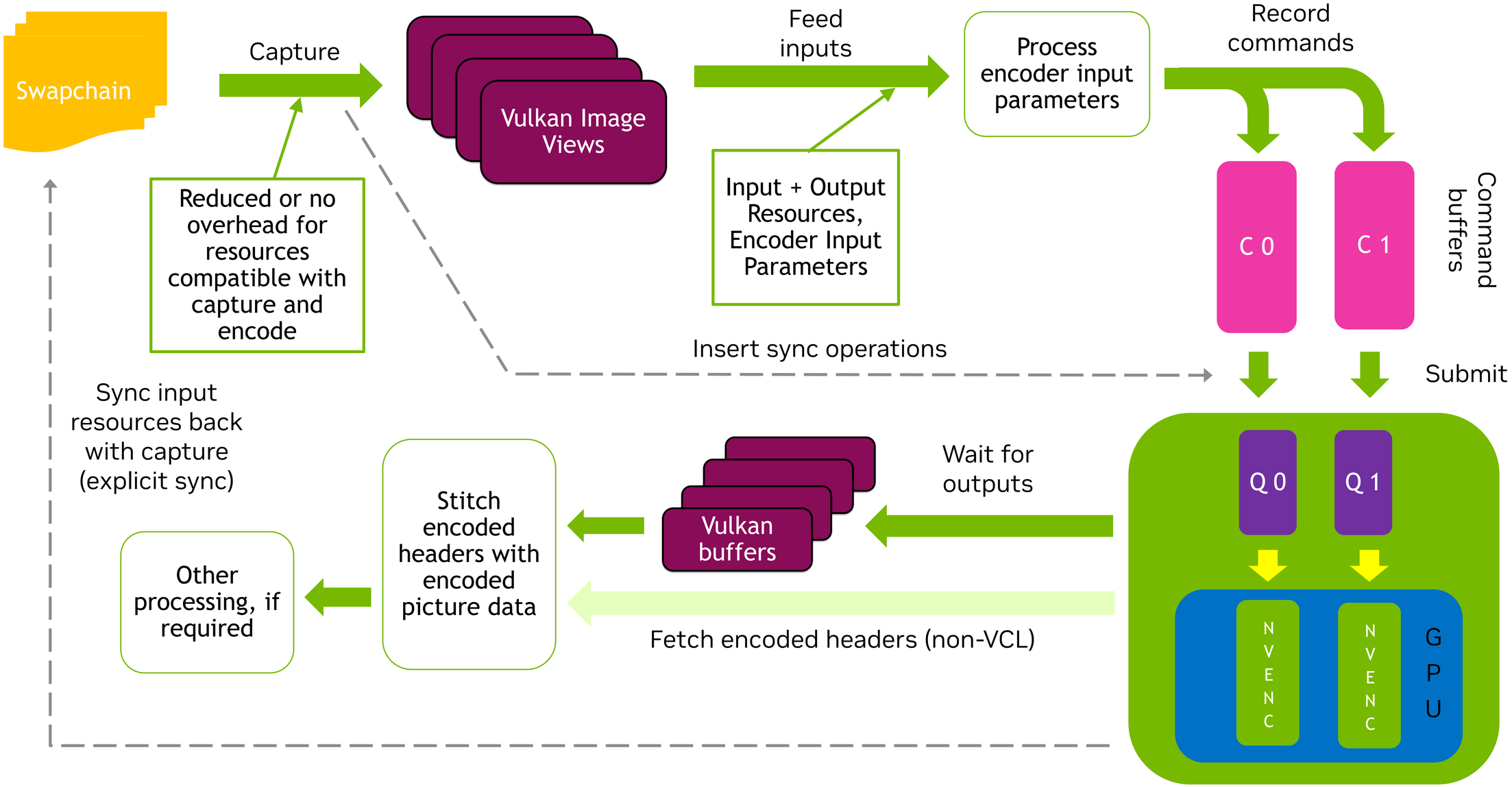
Motivation

- **Cross-platform, standardized, multi-vendor**
- **Better interoperability with the rest of the GPU**
  e.g., Graphics and video can share resources
- **Integration with existing Vulkan ecosystem (same tooling/debugging aids)**
- **Fine-grained control over work submission, memory management and synchronization**

Capture + Encode: High-Level API



Capture + Encode: Vulkan Video



Vulkan Video Roadmap

	**Specification/ Extension**	H.264 & HEVC Decode				H.264 & HEVC Encode	AV1 Decode	AV1 Encode Enc. Quality VP9 Decode
	**Vulkan SDK Support**		H.264 & HEVC Decode					AV1 Decode & Encode
	**NVIDIA Driver Support**	Day-0 Beta	Production (R530)			Day-0 Beta	Day-0 Beta Production (R550)	...

Q4	Q1	Q2	Q3	Q4	Q1	Q2 & beyond
2022	2023				2024	

Video Profiling Using Nsight

Nsight Systems vs Others

For Video Accelerator Trace

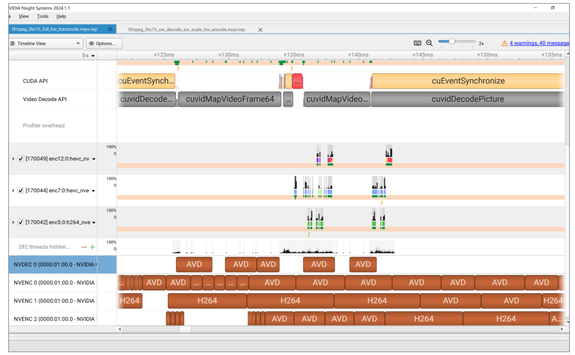
Tool	Function	Visualization	Cross-platform	Timeline View	API Trace	HW Workload Trace	SW Dev Friendly
nvidia-smi	GPU device monitoring	Text only	Yes	Manual log inspection	No	Engine Utilization Only	No
GPUView	GPU system profiling	Graphic UI	No	Yes	Yes	As viewed by Kernel	Yes
NVIDIA Nsight Systems	GPU System profiling	Graphic UI	Yes	Yes	Yes	Yes	Yes

```
CUDA Version                                : 12.5

Attached GPUs                               : 1
GPU 00000000:01:00.0
    Utilization
        Gpu                                 : 7 %
        Memory                              : 7 %
        Encoder                             : 70 %
        Decoder                             : 12 %
        JPEG                                : 0 %
        OFA                                 : 0 %
    GPU Utilization Samples
        Duration                            : 14.00 sec
        Number of Samples                   : 71
        Max                                 : 8 %
        Min                                 : 0 %
        Avg                                 : 4 %
```

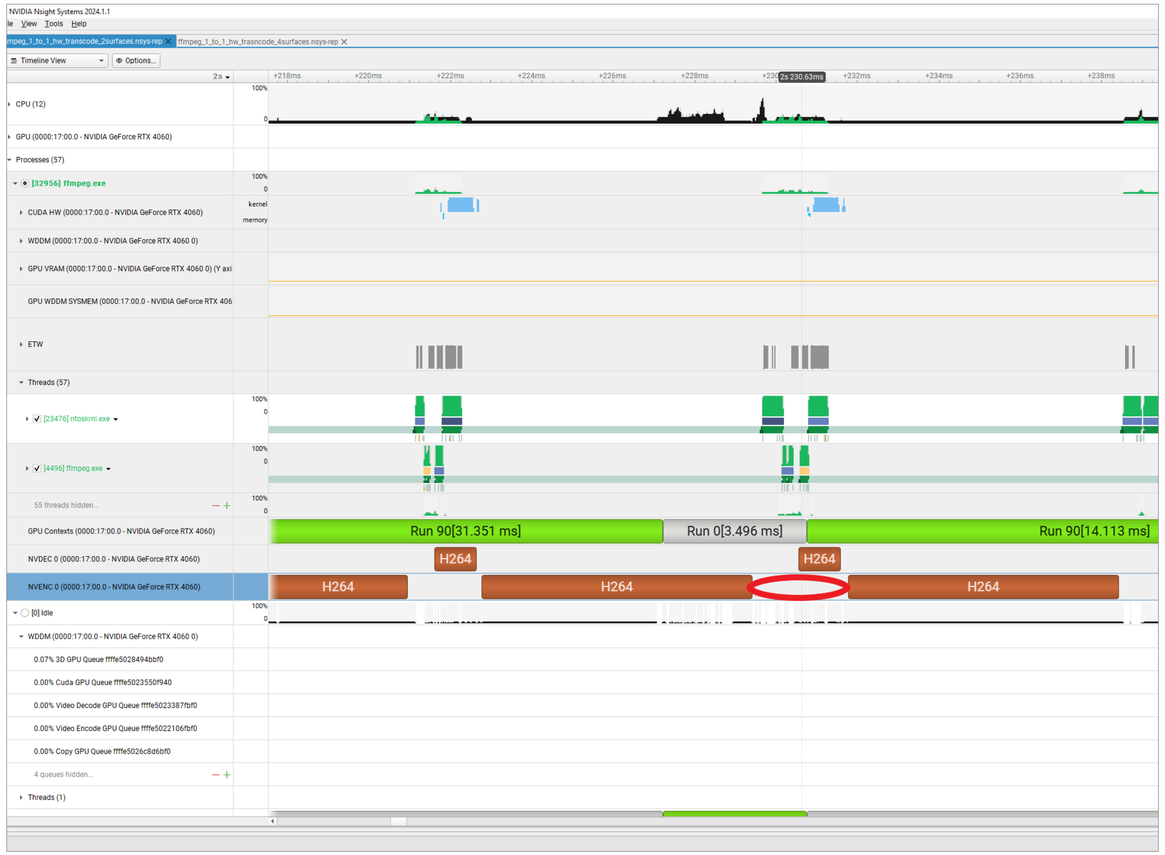
nvidia-smi

GPUView



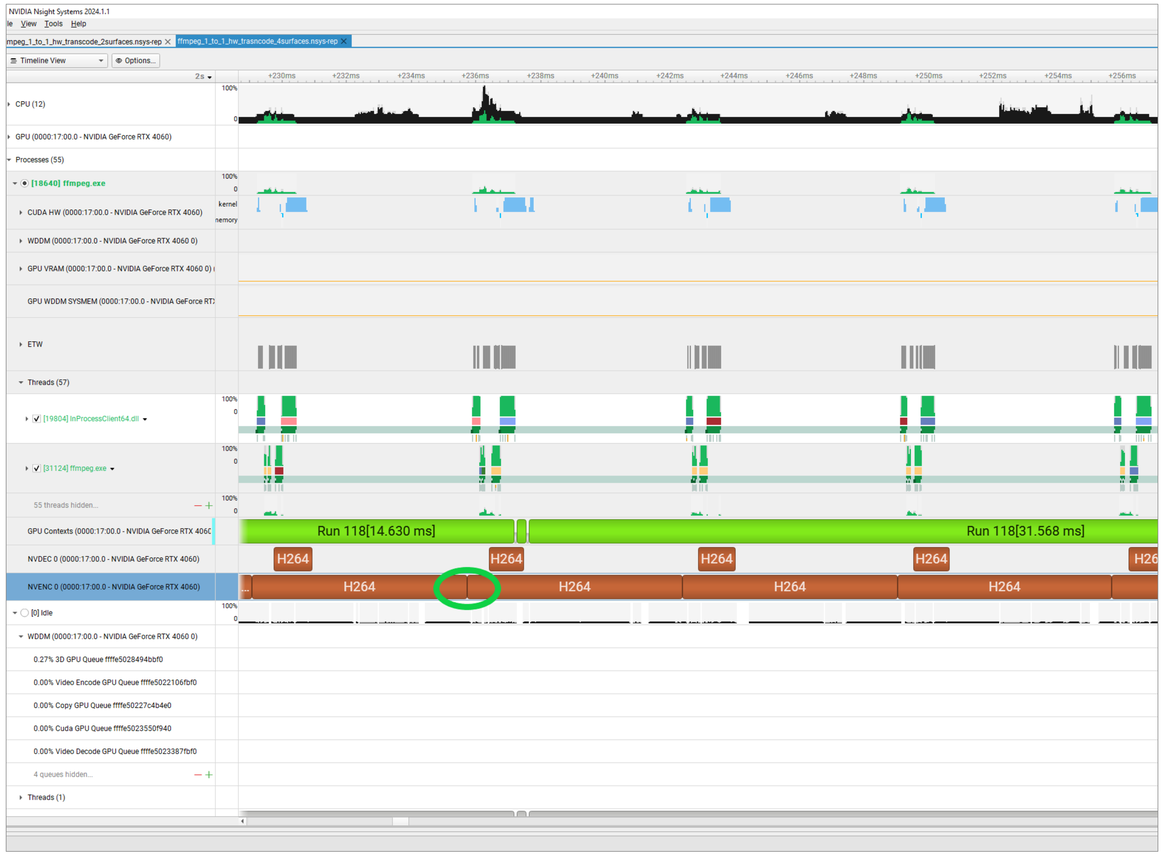
Nsight Systems

Video Profiling Using Nsight: An Example



Unoptimized

FFMPEG 1:1 transcode



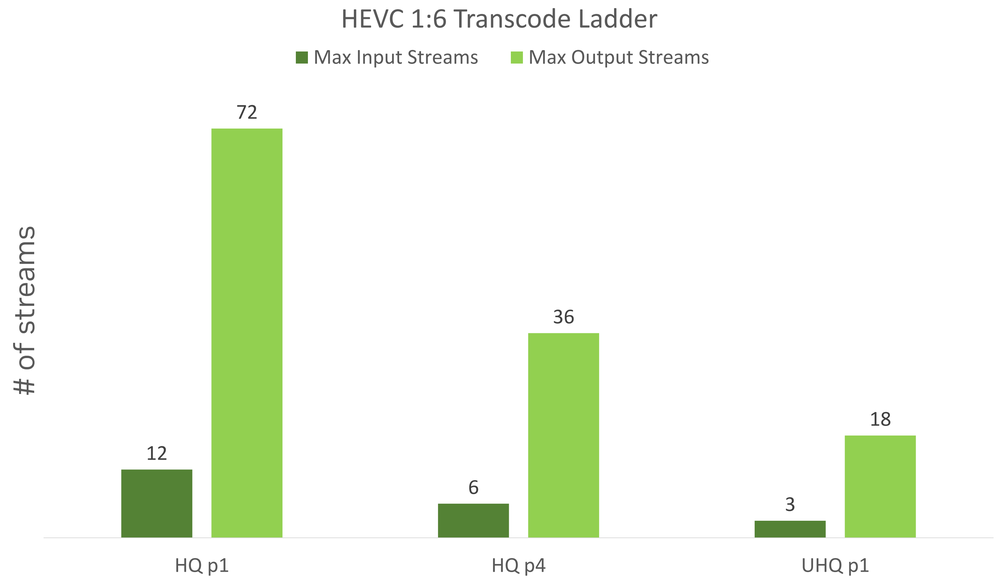
Optimized

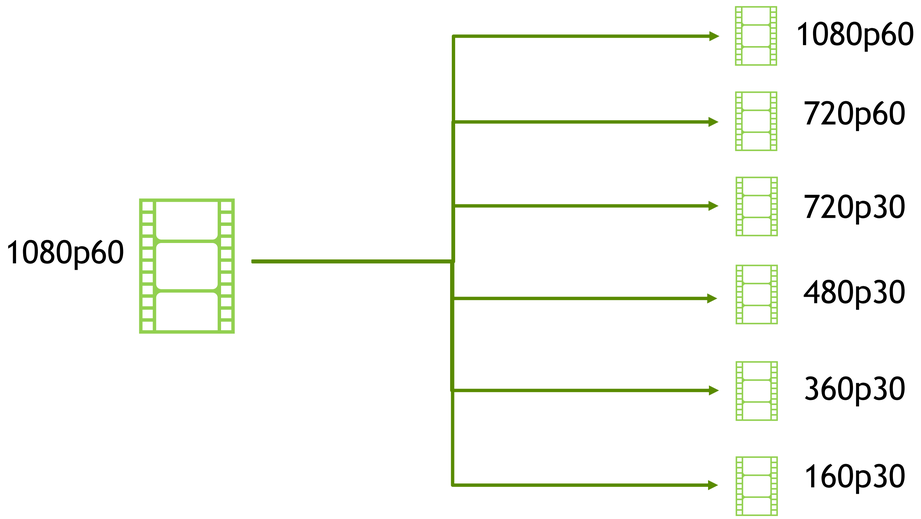
FFMPEG 1:1 transcode

Video Transcode on NVIDIA L4

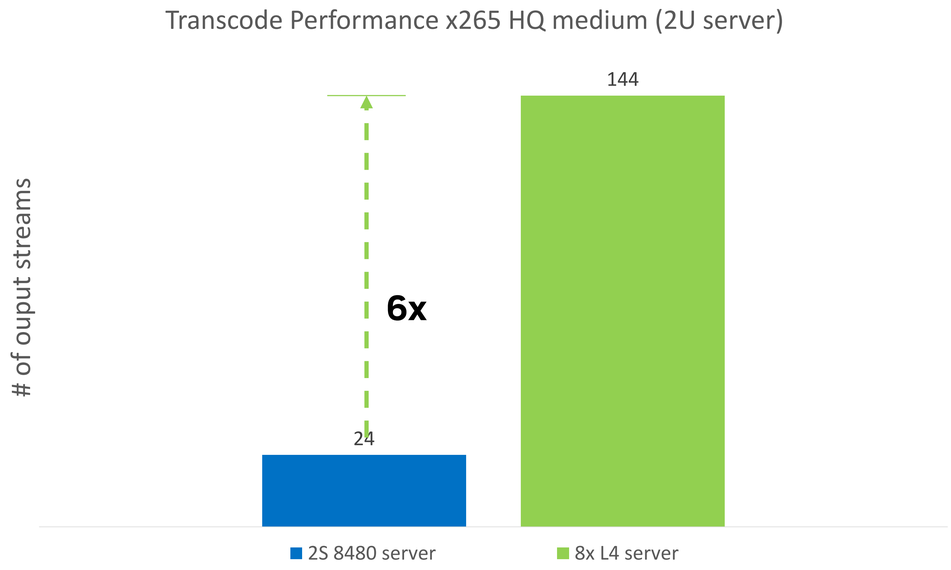
Live Transcode (HEVC) on NVIDIA L4

L4 - 72W
Single Slot, Low Profile
2 NVENC | 4 NVDEC





High ROI for Live Transcode with NVIDIA L4



VIDEO
Video Transcode Pipeline
(1K 1080p60 input streams)

CPU server (2x Platinum 8480+ per server)	L4 server (8x L4 per server)
$5M TCO (3yrs)	$1.2M TCO (3yrs)
250 servers 275kW total power	41 servers 54kW total power
$833/stream Lifetime deployment cost	$205/stream Lifetime deployment cost

80% Energy Savings
$4M Cost Savings

Dual 8480 server (2U chassis, Dual Intel Platinum 8480, 512GB RAM, 1TB SSD, NIC)
L4 server (2U chassis, 2S 32C CPU, 128GB RAM, 1TB SSD, CX6 NIC)

Resources

- **Video Codec SDK**
  - Main page: https://developer.nvidia.com/video-codec-sdk
  - Online documentation: https://docs.nvidia.com/video-technologies/index.html
  - PyNvVideoCodec: https://docs.nvidia.com/video-technologies/pynvvideocodec/index.html

- **Vulkan Video**
  - Sample applications: https://github.com/nvpro-samples/vk_video_samples
  - Vulkan video drivers: https://developer.nvidia.com/vulkan-driver

- **NVIDIA Video Support**
  - Forums: https://forums.developer.nvidia.com/c/gaming-and-visualization-technologies/video-processing-optical-flow/189
    - NVIDIA Developer Forums ➔ Gaming and Visualization Technologies ➔ Video Processing and Optical Flow
  - Email: video-devtech-support@nvidia.com