TECHNICAL OVERVIEW

VIDEO BENCHMARK ASSUMPTIONS

TABLE OF CONTENTS

MOTIVATION

Anyone familiar with video encoding knows that different encoders employ different strategies for achieving the bitrate vs quality targets. Depending on the platform specifics, optimization strategies may differ significantly. This causes peculiarities in the behavior of the encoder output and makes it difficult to perform exact apples-to-apples comparison between two encoders.

Comparison of x264 with NVENC encoding has similar challenges. In this document, we provide details about the assumptions made in order to effectively compare x264 with NVENC.

HARDWARE PLATFORM

Following table summarizes the details of hardware used in our benchmarking setup.

Table 1 - Details of hardware platforms used

System Cfg	A10	Tesla T4	Tesla P4	x264/x265
CPU	Dual Intel(R) Xeon(R) CPU E5-2660 v3 @ 2.60GHz			
GPU	A10 (GA102)	Tesla T4 (TU104)	Tesla P4 (GP104)	N/A
RAM	128 GB			
FFmpeg version	4.2			
NVIDIA Driver	495.29	418.04	418.66	N/A

PRESET

To ensure best scalability when using FFmpeg, we incorporated NVENC encoding presets p1 (fastest) through p7 (highest quality) into FFmpeg. Details on these new presets are available in a dedicated **blog post**. Moreover, NVENC quality has been largely improved allowing users to get more than twice the number of simultaneous encode sessions per NVENC on Turing and Ampere GPUs compared to previous generations, while maintaining similar encoding quality. The table below explains how to compare GPU and CPU for similar bitrate gains.

Table 2 - Appropriate presets to compare for similar bandwidth gains

HW ARCH CODEC	Preset to compare for similar quality in Low Latency / Latency Sensitive scenarios	Preset to compare for similar quality in HQ / Latency Tolerant scenarios
CPU x264/x265	Faster	Medium
Pascal H.264/HEVC	p4	p7
Turing/Ampere H.264/HEVC	p1	p4

To maximize encoding performance, NVENC should be fed with enough encoding work or load to the GPU. Therefore, it is recommended to run multiple encoding/decoding sessions simultaneously (2 at minimum). GeForce gaming cards support up to 3 concurrent encoding sessions while this number is unlimited on professional GPUs.

NVIDIA Video Codec SDK includes sample applications named `AppEncode`, `AppDecode` and `AppTranscode` which allow you to measure performance on your GPU. You will need to use a raw video source for encode stored on SSD or RAM disk (many utility software options can be found on the web to set up a RAM disk).

Alternatively, you can use FFmpeg with NVIDIA video hardware acceleration to perform the testing (details below).

CONTENT

Our encoding benchmark uses a large variety of video content from the following types of video footage:

1. Natural video - high motion (e.g. sports) and movie-type
2. Game captures - high motion and high texture
3. Synthetic content (e.g. animated movies)
4. Amateur video content (e.g. videos shot using camcorder)
5. Video conferencing

The library consists of several thousands of videos in resolutions 720p, 1080p and 2160p (4K), with each video containing at least 600 frames.

Performance of the encoder is somewhat content-dependent. It is important to ensure that while averaging the performance, content of various types and resolutions is used to ensure that the results represent true behavior of the encoder.

LATENCY TOLERANT AND LATENCY SENSITIVE ENCODING

There are two types of use-cases for which video encoding benchmark is executed

1. **Latency-tolerant or high-quality:** Used in applications such as video archiving, streaming with high latency (> 0.5 seconds), video storage, web videos, video streaming (e.g. Netflix). This type of encoding typically has no restrictions on the encoding tools that can be used, subject to the complexity constraints. Features such as B-frames, look-ahead can be used.
2. **Latency-sensitive or low-latency:** Used in latency-sensitive applications such as cloud gaming, game-streaming, game broadcasting. These applications cannot tolerate latency more than a couple of frames. Encoding tools such as B-frames, look-ahead cannot be used in this type of encoding. This type of encoding also puts strict cap on frame-by-frame bit budget and expects strict HRD compliance at small VBV buffer size..

NVIDIA provides benchmarks in both above use-cases.

ENCODING

Each video from the library is encoded at 4 or 5 different bitrates, depending on the resolution, using libx264/libx265 and NVENC options within FFmpeg. For latency-sensitive benchmark, we compare a faster CPU preset with NVENC's p1 preset for Turing and Ampere (and p4 preset for Pascal and previous generations, as shown in Table 2). For latency-tolerant benchmark, we compare a medium CPU preset with NVENC's p4 preset for Turing and Ampere (and p7 preset for Pascal and previous generations as shown in Table 2).

Table 3 - Latency-tolerant H.264 encoding parameters

<table>
<tr><th>Operating point</th><th>Codec</th><th>FFmpeg command line parameters</th></tr>
<tr><td rowspan="3">Latency-tolerant</td><td>NVENC H.264, preset p4 (Ampere and Turing GPUs)</td><td><code>-c:v h264_nvenc</code>
<code>-preset p4</code>
<code>-tune hq</code>
<code>-b:v BITRATE</code>
<code>-profile:v high</code>
<code>-temporal-aq 1</code>
<code>-rc-lookahead 20</code>
<code>-vsync 0</code></td></tr>
<tr><td>NVENC H.264 preset p7 (Pascal and previous generations GPUs)</td><td><code>-c:v h264_nvenc</code>
<code>-preset p7</code>
<code>-tune hq</code>
<code>-b:v BITRATE</code>
<code>-profile:v high</code>
<code>-temporal-aq 1</code>
<code>-rc-lookahead 20</code>
<code>-vsync 0</code></td></tr>
<tr><td>libx264 medium</td><td><code>-c:v libx264</code>
<code>-preset medium</code>
<code>-b:v BITRATE</code>
<code>-tune psnr</code>
<code>-threads LIBXTHREADS</code>[1]
<code>-vsync 0</code></td></tr>
</table>

[1] `LIBXTHREADS` is 1 for 720p, 2 for 1080p and 4 for 2160p.

Table 4 - Latency-sensitive H.264 encoding parameters

Operating point	Codec	FFmpeg command line parameters
Latency-sensitive	NVENC H.264 preset p1 (Ampere and Turing GPUs)	-c:v h264 _ nvenc -preset p1 -rc cbr -tune ll -multipass 0 -b:v BITRATE -bufsize BITRATE/FRATE -profile:v high -g 999999 -vsync 0
	NVENC H.264 preset p4 (Pascal and previous generations GPUs)	-c:v h264 _ nvenc -preset p4 -rc cbr -tune ll -multipass 0 -b:v BITRATE -bufsize BITRATE/FRAMERATE -profile:v high -g 999999 -vsync 0
	libx264 faster	-c:v libx264 -preset faster -b:v BITRATE -bufsize BITRATE/FRATE -maxrate BITRATE -minrate BITRATE -g 999999 -x264opts no-sliced-threads:no-psy=1:aq-mode=0 -tune zerolatency -threads LIBXTHREADS[1] -vsync 0

[1] LIBXTHREADS is 1 for 720p, 2 for 1080p and 4 for 2160p.

Table 5 - Latency-tolerant HEVC encoding parameters

Operating point	**Codec**	**FFmpeg command line parameters**

Latency-tolerant	NVENC HEVC preset p4 (Ampere and Turing GPUs)	`-c:v hevc_nvenc`
		`-preset p4`
		`-tune hq`
		`-b:v BITRATE`
		`-profile:v high`
		`-b_ref_mode 2`
		`-b_adapt 0`
		`-temporal-aq 1`
		`-rc-lookahead 20`
		`-vsync 0`

	NVENC HEVC preset p7 (Pascal and previous generations GPUs)	`-c:v hevc_nvenc`
		`-preset p7`
		`-tune hq`
		`-b:v BITRATE`
		`-profile:v high`
		`-b_ref_mode 2`
		`-b_adapt 0`
		`-temporal-aq 1`
		`-rc-lookahead 20`
		`-vsync 0`

	libx265 medium	`-c:v libx265`
		`-preset medium`
		`-b:v BITRATE`
		`-bf 2`
		`-tune psnr`
		`-threads LIBXTHREADS`[1]
		`-vsync 0`

1 `LIBXTHREADS` is 1 for 720p, 2 for 1080p and 4 for 2160p.

Table 6 - Latency-sensitive HEVC encoding parameters

Operating point	Codec	FFmpeg command line parameters
Latency-sensitive	NVENC HEVC preset p1 (Ampere and Turing GPUs)	-c:v hevc _ nvenc -preset p1 -rc cbr -tune ll -multipass 0 -b:v BITRATE -bufsize BITRATE/FRATE -profile:v high -g 999999 -vsync 0
	NVENC HEVC preset p4 (Pascal and previous generations GPUs)	-c:v hevc _ nvenc -preset p4 -rc cbr -tune ll -multipass 0 -b:v BITRATE -bufsize BITRATE/FRATE -profile:v high -g 999999 -vsync 0
	libx265 faster	-c:v libx265 -preset faster -b:v BITRATE -bufsize BITRATE/FRATE -maxrate BITRATE -minrate BITRATE -g 999999 -tune zerolatency -x265-params no-sliced-threads=1:psy-rd=0.0:no-psy=1:aq-mode=0 -threads LIBXTHREADS[1] -vsync 0

[1] LIBXTHREADS is 1 for 720p, 2 for 1080p and 4 for 2160p.

EVALUATION OF RESULTS

After encoding each video, metrics such as SSIM, PSNR, output bitrate and encoding performance are measured. To measure encoding performance, we measure the time taken to encode all frames at the application level. If multiple files are being encoded in parallel, then the aggregate number of frames in all parallel encoded videos are used to compute performance in frames/second. While evaluating the encoded video bitstreams, we consider the bitrate *actually* generated by the encoder (and *not* the target bitrate specified on the command line). We have observed that in certain cases, an encoder may generate bitstream at bitrate different (sometimes significantly different) from the target bitrate specified on the command line. Using the actual generated bitrate is, therefore, preferable when evaluating the encoded video bitstreams.

In addition to these, the videos are visually inspected to confirm that there are no distortions or unexpected artifacts.

The rate-distortion characteristics of each encoded video are analyzed using PSNR and SSIM and we calculate metrics such a BD-PSNR, BD-SSIM and BD-BR for each video (BD = *Bjontegaard* metric). These metrics are averaged over all videos at a given resolution to generate charts such as those shown at **http://developer.nvidia.com/nvidia-video-codec-sdk**.

Ready to get Started

Learn more about hardware accelerated video encode and decode with **NVIDIA Video Codec SDK**.