TECHNICAL OVERVIEW

VIDEO BENCHMARK ASSUMPTIONS

TABLE OF CONTENTS

- 1. Motivation
- 2. Hardware Platform
- 3. Presets
- 4. Content
- 5. Latency Tolerant and Latency Sensitive Encoding
- 6. Encoding
- 7. Evaluation of Results

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 to effectively compare x264 with NVENC.

Hardware Platform

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

System Config	L4	libAOM- AV1	A10	Τ4	x264, x265
CPU	Dual Intel(R) Xeon(R) Gold- 6140 @ 2.30GHz		Dual Intel(R) Xeon(R) CPU E5-2660 v3 @ 2.60GHz		
GPU	L4 (AD104)	N/A	A10 (GA102)	T4 (TU104)	N/A
RAM	385 GiB		128GiB		
FFmpeg version	6		4.4		
NVIDIA Driver	520.65 (or latest)	N/A	495.29	495.29	N/A

Presets

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 <u>blog post</u>. 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.

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
Turing/Ampere/Ada H.264/HEVC/AV1	p1	p4

Table 2 - Appropriate presets to compare for similar bandwidth gains

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 (8 at minimum). GeForce gaming cards support up to 5 concurrent encoding sessions while this number is unlimited on professional GPUs.

<u>NVIDIA Video Codec SDK</u> includes sample applications named <u>AppEncode</u>, <u>AppDecode</u> and <u>AppTranscode1toN</u> 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

Latency Tolerant and Latency Sensitive Encoding

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

- 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.
- Latency-sensitive or low-latency: Used in latency-sensitive applications such as cloud gaming, gamestreaming, 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 faster CPU preset with NVENC's p1 preset for Turing, Ampere and Ada. For latency-tolerant benchmark, we compare medium CPU preset with NVENC's p4 preset for Turing, Ampere and Ada.

Operating point	Codec	FFmpeg command line parameters
	NVENC H.264, HEVC preset p4 (Ampere, Turing GPUs)	-c:v \$NVENC -preset p4 -tune hq -b:v BITRATE -profile:v CODEC_PROFILE -temporal-aq 1 -rc-lookahead 20 -vsync 0
Latency-tolerant	NVENC H.264, HEVC preset p4 (Ada GPUs)	-c:v \$NVENC -no-scenecut 1 -g 250 -preset \$PRESET -tune hq - b:v BITRATE -profile:v CODEC_PROFILE -b_ref_mode 2 - b_adapt 0 -temporal-aq 1 -rc- lookahead 20 FRAME_LIMIT -vsync passthrough
	libx264 medium	-c:v libx264 -preset medium -b:v BITRATE -tune psnr -threads LIBXTHREADS ¹ -vsync 0
	libx265 medium	-c:v libx265 -preset medium -b:v BITRATE -bf 2 -tune psnr -threads LIBXTHREADS ¹ -vsync 0

Table 3 - Latency-tolerant H.264, HEVC encoding parameters

 $^{^{1}}$ LIBXTHREADS is 1 for 720p, 2 for 1080p and 4 for 2160p.

Operating point	Codec	FFmpeg command line parameters
	NVENC H.264, HEVC preset p1 (Ampere and Turing GPUs)	-c:v \$NVENC -preset p1 -rc cbr -tune ll -multipass 0 -b:v BITRATE -bufsize BITRATE/FRATE -profile:v CODEC_PROFILE -g 999999 -vsync 0
Latency-sensitive	NVENC H.264, HEVC preset p1 (Ada GPUs)	-c:v \$NVENC -preset \$PRESET -rc cbr -tune ll -multipass 0 -b:v BITRATE -bufsize BITRATE/FRATE - profile:v CODEC_PROFILE -g 999999 FRAME_LIMIT -vsync passthrough
	libx264 faster	-c:v libx264 -preset faster -b:v BITRATE -bufsize BITRATE/FRATE -maxrate BITRATE -g 999999 -x264opts no-sliced-threads:no- psy=1:aq-mode=0 -tune zerolatency -threads LIBXTHREADS ¹ -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 -vsync passthrough

Table 4 - Latency-sensitive H.264, HEVC encoding parameters

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

Operating point	Codec	FFmpeg command line parameters
	NVENC AV1, preset p4	-c:v \$NVENC -no-scenecut 1 -g 250
	(Ada GPUs)	-preset \$PRESET -tune hq -
		b:v BITRATE -profile:v
		CODEC_PROFILE -b_ref_mode 2 -
		b_adapt 0 -temporal-aq 1 -rc-
		lookahead 20 FRAME_LIMIT -vsync
		passthrough
Latency-tolerant	libaom – AV1	-c:v libaom-av1 -cpu-used 8 -b:v
		BITRATE -threads 8 -g 250 -
		keyint_min 250

Table 5 - Latency-tolerant AV1 encoding parameters

Table 6 - Latency-sensitive AV1 encoding parameters

Operating point	Codec	FFmpeg command line parameters
Latency-sensitive	NVENC AV1, preset p1 (Ada GPUs)	-c:v \$NVENC -preset \$PRESET -rc cbr -tune ll -multipass 0 -b:v BITRATE -bufsize BITRATE/FRATE - profile:v CODEC_PROFILE -g 9999999 FRAME_LIMIT -vsync passthrough
	libaom – AV1	-c:v libaom-av1 -cpu-used 8 -b:v BITRATE -threads LIBXTHREADS - bufsize BITRATE/FRATE -g 999999 - usage 1

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

Evaluation of Results

After encoding each video, metrics such as PSNR, SSIM, 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 BD-BR %(PSNR) and BD-BR%(SSIM) for each video (BD = *Bjontegaard* metric). These metrics are used to generate charts such as those shown at <u>http://developer.nvidia.com/nvidia-video-codec-sdk</u>.

^{© 2020} NVIDIA Corporation. All rights reserved. NVIDIA, the NVIDIA logo, DGX POD, DGX SuperPOD, NGC, NGC-Ready, EGX, Clara, CUDA, and Mellanox are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. All other trademarks and copyrights are the property of their respective owners. Aug20