Benchmarking FFmpeg’s Hardware Codecs

VES 101
Jan Ozer
Agenda

- Theory of testing
- H.264
  - NVIDIA
  - Quick Sync
- HEVC
  - NGCodec – Field Programmable Gate Array-based codec (FPGA)
    - Can rent on AWS
  - Intel SVT-HEVC (not really hardware but topical)
Overview

1. Cloud transcoding is the optimal workflow for many live producers
2. There are two options; software or hardware
   a. Software requires an expensive cloud computer with lots of CPUs
   b. Hardware (GPU, FPGA) requires lower CPU but may cost more
3. So, how do CPU-only and hardware systems compare?
   a. Quality-wise
   b. Cost-wise
4. The answers?
   a. Quality-wise: Hardware stacks up pretty well
   b. Cost-wise: It’s complicated; I couldn’t find a single machine that could perform all the hardware and software encodes
Theory of Testing

1. Derive most practical encoding configuration
2. Test capacity using encoding ladder
   a. Hardware - no dropped frames
   b. Software - 55 fps or higher
3. Test quality with rate distortion curves at those settings
NVIDIA H.264

- Instance
- Settings
- Capacity
- Quality
Instance - g3.4xlarge

<table>
<thead>
<tr>
<th>Name</th>
<th>GPUs</th>
<th>vCPU</th>
<th>Memory (GiB)</th>
<th>GPU Memory (GiB)</th>
<th>Price/hr* (Linux)</th>
</tr>
</thead>
<tbody>
<tr>
<td>g3s.xlarge</td>
<td>1</td>
<td>4</td>
<td>30.5</td>
<td>8</td>
<td>$0.75</td>
</tr>
<tr>
<td>g3.4xlarge</td>
<td>1</td>
<td>16</td>
<td>122</td>
<td>8</td>
<td>$1.14</td>
</tr>
<tr>
<td>g3.8xlarge</td>
<td>2</td>
<td>32</td>
<td>244</td>
<td>16</td>
<td>$2.28</td>
</tr>
<tr>
<td>g3.16xlarge</td>
<td>4</td>
<td>64</td>
<td>488</td>
<td>32</td>
<td>$4.56</td>
</tr>
</tbody>
</table>

- Instance selected and configured by engineers at Softvelum, who run the Nimble Streamer cloud transcoder. They have my undying gratitude and appreciation.
Finding the Right Settings

- **Best source - Using FFmpeg With NVIDIA GPU HW Acceleration**

- **Recommended string:**
  ```
  ffmpeg -y -vsync 0 -hwaccel cuvid -c:v h264_cuvid -i input.mp4 -c:a copy -c:v h264_nvenc -preset slow -profile high -b:v 5M -bufsize 5M -maxrate 10M -qmin 0 -g 250 -bf 2 -temporal-aq 1 -rc-lookahead 20 -i_qfactor 0.75 -b_qfactor 1.1 output.mp4
  ```

- **Concerns:**
  - Data rate fluctuations due to 2 second VBV buffer
  - Performance - Slow preset
Switch to 1 Second VBV Buffer

- 1 second buffer delivered slightly higher overall bitrate and slightly more uniform stream
- Tried Medium preset to optimize capacity
  - VMAF dropped from 82.35 to 82.19
Check for Transient Quality Issues

- VMAF plot in VQMT
- Pretty similar throughout
- Deep drop near frame 1300 is highlighted on the bottom

- Actual quality difference is negligible
## Comparisons

<table>
<thead>
<tr>
<th></th>
<th>x264 Medium</th>
<th>Original White Paper (Slow)</th>
<th>White Paper with CBR (Slow)</th>
<th>White Paper with CBR/Medium</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bitrate</td>
<td>3940</td>
<td>3716</td>
<td>3903</td>
<td>3896</td>
</tr>
<tr>
<td>Peak</td>
<td>6386</td>
<td>5384</td>
<td>5468</td>
<td>5123</td>
</tr>
<tr>
<td>VMAF</td>
<td>79.80</td>
<td>81.82</td>
<td>82.35</td>
<td>82.19</td>
</tr>
<tr>
<td>PSNR</td>
<td>33.65</td>
<td>33.65</td>
<td>33.83</td>
<td>33.74</td>
</tr>
<tr>
<td>CPU%</td>
<td>15%</td>
<td>15%</td>
<td>15%</td>
<td>15%</td>
</tr>
</tbody>
</table>

- Very little difference in quality/CPU with Slow or Medium
- Tested with Medium to optimize performance
Testing Capacity

- Tested with this encoding ladder
- Kept opening instances and running until frame rate dropped to below 60fps

<table>
<thead>
<tr>
<th>Rez</th>
<th>Data rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1080p60</td>
<td>6 mbps</td>
</tr>
<tr>
<td>1080p30</td>
<td>4 mbps</td>
</tr>
<tr>
<td>720p30</td>
<td>2.5 mbps</td>
</tr>
<tr>
<td>540p30</td>
<td>1.2 mbps</td>
</tr>
<tr>
<td>360p30</td>
<td>0.8 mbps</td>
</tr>
</tbody>
</table>
NVIDIA Encodings

- Hardware decode to CUVID, then encode
  
  ```bash
  ffmpeg -y -vsync 0 -hwaccel cuvid -c:v h264_cuvid -i input.mp4 -c:v h264_nvenc -preset medium -b:v 5M -bufsize 5M -maxrate 5M -qmin 0 -g 120 -bf 2 -temporal-aq 1 -rc-lookahead 20 -i_qfactor 0.75 -b_qfactor 1.1 output.mp4
  ```

- Achieved two 60 fps encodes on G3.4 xlarge
x264 Encodes

- Simple x264 conversion script
  - Tested with Medium, fast, and veryfast

```bash
ffmpeg -y -re -i input.mp4 -c:v libx264 -preset medium -b:v 5M -bufsize 5M -maxrate 5M -g 120 output.mp4
```
Capacity

- On GPU optimized computer, couldn’t produce a single x264 ladder with any preset
- Compared software performance to a C5.18 xlarge, which cost about the same ($1.25/hour compared to $1.14).
- Achieved 4 simultaneous encodes
Capacity

- Four encodes compared to 2 with NVIDIA, so about 1/2 the cost, though plenty of dropped frames
- Much higher-performance NVIDIA hardware is now available, so you’ll have to perform your own cost analysis
- Look at quality after Intel QSW
Intel Quick Sync Encoding

● System:
  ○ Single socket Xeon-E3 (QSV)
  ○ Intel(R) Xeon(R) CPU E3-1585L v5 @ 3.00GHz
  ○ 4 core with Intel® Iris® Pro Graphics P580
  ○ 2x 16GB @2133MHz DDR4 memory

● Accessed:
  ○ Docker containers based upon open-source Open Visual Cloud (OVC), which enables developers to quickly deploy Intel visual-cloud based pipelines.
  ■ Links to Open Visual Cloud page: intel.ly/Vis_Cloud_com
  ■ Open Visual Cloud Dockerfiles git page: bit.ly/OpenVisGit
Which Preset? - Performance vs. Quality

<table>
<thead>
<tr>
<th>Preset</th>
<th>FPS</th>
<th>VMAF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preset 1</td>
<td>128</td>
<td>73.75</td>
</tr>
<tr>
<td>Preset 2</td>
<td>202</td>
<td>73.64</td>
</tr>
<tr>
<td>Preset 3</td>
<td>239</td>
<td>73.29</td>
</tr>
<tr>
<td>Preset 4</td>
<td>239</td>
<td>73.29</td>
</tr>
<tr>
<td>Preset 5</td>
<td>247</td>
<td>73.25</td>
</tr>
<tr>
<td>Preset 6</td>
<td>260</td>
<td>73.11</td>
</tr>
<tr>
<td>Preset 7</td>
<td>275</td>
<td>69.82</td>
</tr>
</tbody>
</table>

Tested at preset 4
FFmpeg Script (Intel Provided)

```bash
ffmpeg -y -init_hw_device qsv=hw -filter_hw_device hw -i football_1080p.mp4 -vf hwupload=extra_hw_frames=64,format=qsv -c:v h264_qsv -b:v 4M -maxrate 4M -b:v 4M -bufsize 4M -g 120 -idr_interval 4 -async_depth 5 -preset 4 -c:a aac -b:a 128k -ac 2 -ar 48000 football_1080p4M_p4.mp4
```
On Tested Computer

- 1 encoding ladder with Quick Sync at preset 4
  - Using preset 7 did not deliver 2 full ladders
- No ladders with x264, even using veryfast preset
- Obviously could get higher performance with other systems
- Had hope to use exclusively AWS computers to get pricing, but went with Intel supplied computers for simplicity
H.264 Quality Results

- Four videos
  - Netflix Dinner Scene
  - Harmonic football
  - GTAV
  - Netflix Meridian
  - All 1080p60
- Tested at 2-5 Mbps

- Four tested codecs
  - NVIDIA NVENC at Medium
  - Intel Quick Sync at Preset 4
  - x264 at Medium and Veryfast
Dinner Scene - Rate Distortion Curve
## Dinner Scene - BD-Rate Computations

<table>
<thead>
<tr>
<th>VMAF</th>
<th>NVIDIA</th>
<th>x264 Medium</th>
<th>QuickSync</th>
<th>x264 Very Fast</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVIDIA</td>
<td>X</td>
<td>-2.42</td>
<td>-8.94</td>
<td>-29.11</td>
</tr>
<tr>
<td>x264 Medium</td>
<td>2.48</td>
<td>X</td>
<td>-7.17</td>
<td>-25.55</td>
</tr>
<tr>
<td>QuickSync</td>
<td>9.82</td>
<td>7.72</td>
<td>X</td>
<td>-20.19</td>
</tr>
<tr>
<td>x264 Very Fast</td>
<td>41.07</td>
<td>34.31</td>
<td>25.29</td>
<td>X</td>
</tr>
</tbody>
</table>
Football - Rate Distortion Curve

FOOTBALL 1080P60 - VMAF

- NVIDIA
- x264 Medium
- Quick Sync
- x264 Very Fast

Data Rate vs VMAF Points Graph
## Football - BD-Rate Computations

<table>
<thead>
<tr>
<th>VMAF</th>
<th>NVIDIA</th>
<th>x264 Medium</th>
<th>Quick Sync</th>
<th>x264 Very Fast</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVIDIA</td>
<td>X</td>
<td>-1.68</td>
<td>-10.20</td>
<td>-21.04</td>
</tr>
<tr>
<td>x264 Medium</td>
<td>1.71</td>
<td>X</td>
<td>-8.60</td>
<td>-18.99</td>
</tr>
<tr>
<td>Quick Sync</td>
<td>11.36</td>
<td>9.41</td>
<td>X</td>
<td>-12.27</td>
</tr>
<tr>
<td>x264 Very Fast</td>
<td>26.64</td>
<td>23.44</td>
<td>13.99</td>
<td>X</td>
</tr>
</tbody>
</table>
GTAV - Rate Distortion Curve

GTAV 1080p60 - VMAF

DATA RATE

DATA RATE

VMAF POINTS

VMAF POINTS

NVIDIA

x264 Medium

Quick Sync

x264 Very Fast
## GTAV - BD-Rate Computations

<table>
<thead>
<tr>
<th>VMAF</th>
<th>NVIDIA</th>
<th>x264 Medium</th>
<th>Quick Sync</th>
<th>x264 Very Fast</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVIDIA</td>
<td>X</td>
<td>-6.97</td>
<td>-4.51</td>
<td>-28.41</td>
</tr>
<tr>
<td>x264 Medium</td>
<td>7.50</td>
<td>X</td>
<td>2.36</td>
<td>-19.92</td>
</tr>
<tr>
<td>Quick Sync</td>
<td>4.72</td>
<td>-2.31</td>
<td>X</td>
<td>-23.55</td>
</tr>
<tr>
<td>x264 Very Fast</td>
<td>39.68</td>
<td>24.87</td>
<td>30.81</td>
<td>X</td>
</tr>
</tbody>
</table>
Meridian - Rate Distortion Curve
<table>
<thead>
<tr>
<th>VMAF</th>
<th>NVIDIA</th>
<th>x264 Medium</th>
<th>Quick Sync</th>
<th>x264 Very Fast</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVIDIA</td>
<td>X</td>
<td>-16.64</td>
<td>-7.47</td>
<td>-41.58</td>
</tr>
<tr>
<td>x264 Medium</td>
<td>19.96</td>
<td>X</td>
<td>10.18</td>
<td>-28.04</td>
</tr>
<tr>
<td>Quick Sync</td>
<td>8.08</td>
<td>-9.24</td>
<td>X</td>
<td>-35.31</td>
</tr>
<tr>
<td>x264 Very Fast</td>
<td>71.18</td>
<td>38.97</td>
<td>54.58</td>
<td>X</td>
</tr>
</tbody>
</table>
Overall - Rate Distortion Curve

OVERALL 1080P60 - VMAF

- NVIDIA
- x264 Medium
- Quick Sync
- x264 Very Fast

VMAF POINTS

DATA RATE
## Overall - BD Rate

<table>
<thead>
<tr>
<th>VMAF</th>
<th>NVIDIA</th>
<th>x264 Medium</th>
<th>Quick Sync</th>
<th>x264 Very Fast</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVIDIA</td>
<td>X</td>
<td>-4.49</td>
<td>-9.08</td>
<td>-27.04</td>
</tr>
<tr>
<td>x264 Medium</td>
<td>4.70</td>
<td>X</td>
<td>-4.92</td>
<td>-22.30</td>
</tr>
<tr>
<td>Quick Sync</td>
<td>9.99</td>
<td>5.17</td>
<td>X</td>
<td>-18.97</td>
</tr>
<tr>
<td>x264 Very Fast</td>
<td>37.06</td>
<td>28.69</td>
<td>23.41</td>
<td>X</td>
</tr>
</tbody>
</table>
Some major differences in ratings
Actual visual differences not that significant
Some major differences
Major Quality Differences - NVIDIA
Meridian - 4 Mbps

- Consistent problem or just football?
- With Meridian, some regions where NVIDIA exhibited transient issues
- Quick Sync had more
  - Transient issues definitely a concern
This is my first testing of hardware encoders

NVIDIA results seem good if you can make the cost-side work
  - Better than Medium quality

Intel performance was good, as were overall scores

Transient quality is a concern

<table>
<thead>
<tr>
<th>VMAF</th>
<th>NVIDIA</th>
<th>x264 Medium</th>
<th>Quick Sync</th>
<th>x264 Very Fast</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVIDIA</td>
<td>X</td>
<td>-4.49</td>
<td>-9.08</td>
<td>-27.04</td>
</tr>
<tr>
<td>x264 Medium</td>
<td>4.70</td>
<td>X</td>
<td>-4.92</td>
<td>-22.30</td>
</tr>
<tr>
<td>Quick Sync</td>
<td>9.99</td>
<td>5.17</td>
<td>X</td>
<td>-18.97</td>
</tr>
<tr>
<td>x264 Very Fast</td>
<td>37.06</td>
<td>28.69</td>
<td>23.41</td>
<td>X</td>
</tr>
</tbody>
</table>
Compared:
- NGCodec - FPGA-based encoding
- Intel SVT-HEVC - preset 10 (live)
- Intel SVT-HEVC - preset 1 (best quality)
- x265 veryfast
NGCodec

- Test spec - 16 core AMD EPYC CPU based machine with 32GB of DDR4 RAM and 1TB of SSD
- Two FPGAs
- Full PCIe 16 lanes communication speed between CPU and both FPGAs.

- Performance
  - One full encoding ladder for each FPGA
NGCodec Script

```bash
ffmpeg -y -re -i football_1080p.mp4 -c:a aac -b:a 128k -ac 2 -ar 48000 -c:v NGC265 -b:v 3M -g 0 -idr-period 120 football_1080p_3M_ngc265.mp4
```

- NGCodec provided
- No real preset to toggle quality vs. encoding speed
  - Either live and full quality or not live
  - Buffer setting is fixed
What is SVT-HEVC?

- "The Scalable Video Technology for HEVC Encoder (SVT-HEVC Encoder) is an HEVC-compliant encoder library core that achieves excellent density-quality tradeoffs, and is highly optimized for Intel® Xeon Scalable Processor and Xeon D processors"
- bit.ly/GY-SVT-HEVC
- Basically, a highly efficient codec for multi-threaded operation
Which Preset

- Tested 2
- Preset 10 for real time
- Preset 1 for VOD
Intel supplied

Doubled buffer size wherever possible on HEVC encodes
Hardware Testing

- Test system:
  - Dual socket Xeon-E5
  - Xeon Skylake Platinum 8180
    2.5GHz 38.5MB 205W 28 cores
  - 12x 16GB @2666MHz

- Performance
  - 3 full ladders in software at preset 1 (see next slide)
  - x265 veryfast was under 30 fps
x265 Very Fast - Not One Full Encoding Ladder
X265 Script

```bash
ffmpeg -y -re -i football_1080p.mp4 -c:v libx265 -preset veryfast -x265-params keyint=120:min-keyint=120:scenecut=0:bitrate=4000k:vbr-maxrate=4000k:vbr-bufsize=8000k -pix_fmt yuv420p Football_1080p_6MB.mp4
```

- Simple as possible
HEVC Quality Results

- Four videos
  - Netflix Dinner Scene
  - Harmonic football
  - GTAV
  - Netflix Meridian
  - All 1080p60
- Tested at 1-4 Mbps

- Four tested codecs
  - NGCodec
  - SVT-HEVC @ 1 and 10
  - X265 at veryfast
HEVC - Dinner Scene - Rate Distortion Curve
### HEVC - Dinner Scene - BD-Rate Computations

<table>
<thead>
<tr>
<th>VMAF</th>
<th>NGCODEC</th>
<th>x265 Very Fast</th>
<th>SVT-HEVC-L10</th>
<th>SVT-HEVC-L1</th>
</tr>
</thead>
<tbody>
<tr>
<td>NGCODEC</td>
<td>X</td>
<td>4.46</td>
<td>-17.83</td>
<td>NA</td>
</tr>
<tr>
<td>x265 Very Fast</td>
<td>-4.27</td>
<td>X</td>
<td>-21.45</td>
<td>NA</td>
</tr>
<tr>
<td>SVT-HEVC-L10</td>
<td>21.70</td>
<td>27.31</td>
<td>X</td>
<td>NA</td>
</tr>
<tr>
<td>SVT-HEVC-L1</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>X</td>
</tr>
</tbody>
</table>
HEVC - Football - Rate Distortion Curve

![Rate Distortion Curve for Football 1080P60 - VMAF](image)
## HEVC - Football - BD-Rate Computations

<table>
<thead>
<tr>
<th>VMAF</th>
<th>NGCODEC</th>
<th>x265 Very Fast</th>
<th>SVT-HEVC-L10</th>
<th>SVT-HEVC-L1</th>
</tr>
</thead>
<tbody>
<tr>
<td>NGCODEC</td>
<td>X</td>
<td>1.85</td>
<td>-24.02</td>
<td>12.80</td>
</tr>
<tr>
<td>x265 Very Fast</td>
<td>-1.82</td>
<td>X</td>
<td>-26.45</td>
<td>11.83</td>
</tr>
<tr>
<td>SVT-HEVC-L10</td>
<td>31.62</td>
<td>35.97</td>
<td>X</td>
<td>52.48</td>
</tr>
<tr>
<td>SVT-HEVC-L1</td>
<td>-11.35</td>
<td>-10.58</td>
<td>-34.42</td>
<td>X</td>
</tr>
</tbody>
</table>
HEVC - GTAV - Rate Distortion Curve
<table>
<thead>
<tr>
<th>VMAF</th>
<th>NGCODEC</th>
<th>x265 Very Fast</th>
<th>SVT-HEVC-L10</th>
<th>SVT-HEVC-L1</th>
</tr>
</thead>
<tbody>
<tr>
<td>NGCODEC</td>
<td>X</td>
<td>-6.08</td>
<td>-24.64</td>
<td>16.46</td>
</tr>
<tr>
<td>x265 Very Fast</td>
<td>6.48</td>
<td>X</td>
<td>-19.32</td>
<td>24.37</td>
</tr>
<tr>
<td>SVT-HEVC-L10</td>
<td>32.69</td>
<td>23.95</td>
<td>X</td>
<td>57.10</td>
</tr>
</tbody>
</table>
HEVC - Meridian - Rate Distortion Curve

MERIDIAN 1080P60 - VMAF

DATA RATE

VMAF POINTS

NGCODEC, x265 Very Fast, SVT+HEVC-L10, SVT+HEVC-L1
<table>
<thead>
<tr>
<th>VMAF</th>
<th>NGCODEC</th>
<th>x265 Very Fast</th>
<th>SVT-HEVC-L10</th>
<th>SVT-HEVC-L1</th>
</tr>
</thead>
<tbody>
<tr>
<td>NGCODEC</td>
<td>X</td>
<td>3.83</td>
<td>-29.14</td>
<td>9.79</td>
</tr>
<tr>
<td>x265 Very Fast</td>
<td>-3.69</td>
<td>X</td>
<td>-31.73</td>
<td>5.67</td>
</tr>
<tr>
<td>L10</td>
<td>41.13</td>
<td>46.47</td>
<td>X</td>
<td>55.64</td>
</tr>
<tr>
<td>SVT-HEVC-L1</td>
<td>-8.91</td>
<td>-5.37</td>
<td>-35.75</td>
<td>X</td>
</tr>
</tbody>
</table>
HEVC - Overall - Rate Distortion Curve (less Dinner Scene)
<table>
<thead>
<tr>
<th>VMAF</th>
<th>NGCODEC</th>
<th>x265 Very Fast</th>
<th>SVT-HEVC-L10</th>
<th>SVT-HEVC-L1</th>
</tr>
</thead>
<tbody>
<tr>
<td>NGCODEC</td>
<td>X</td>
<td>0.05</td>
<td>-24.87</td>
<td>13.41</td>
</tr>
<tr>
<td>x265 Very Fast</td>
<td>-0.05</td>
<td>X</td>
<td>-25.33</td>
<td>14.15</td>
</tr>
<tr>
<td>L10</td>
<td>33.11</td>
<td>33.92</td>
<td>X</td>
<td>54.04</td>
</tr>
</tbody>
</table>
What About Transient Quality?

NGCodec vs. Intel SVT @ P10

- Some spikes
- Quality delta in frames are not significant
What’s the Bottom Line?

- Hardware encoding showed great promise
  - H.264 - NVIDIA was worth exploring
    - Intel not so much - lower quality and transient issues
  - HEVC - NGCodec - best for live encoding
    - SVT - Real time quality needs improvement (but codec is new)
    - Best quality looks competitive with x265 (but need to compare at x.265 Medium to Slow for true comparison)
    - Will run these tests for upcoming article in Streaming Media
Suggested Procedure

- Test capacity using current encoding ladder
- Test quality as shown
  - Performance/quality graphs should provide a good starting point