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

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

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

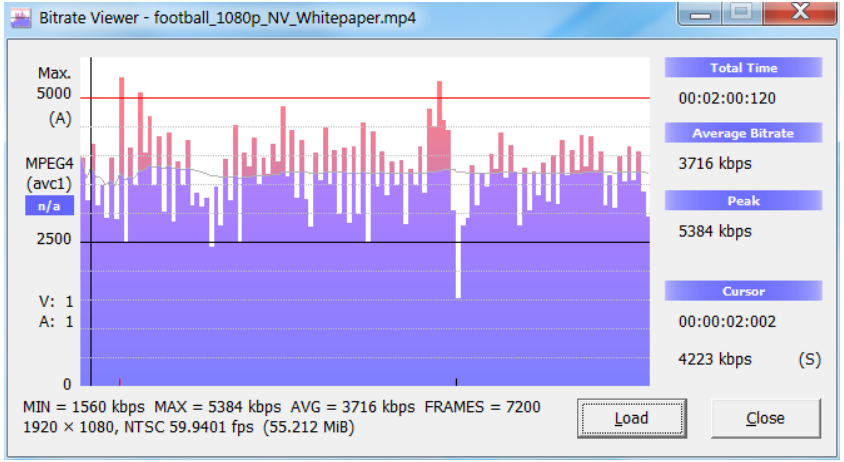
Finding the Right Settings

- Best source - Using FFmpeg With NVIDIA GPU HW Acceleration
  - https://developer.nvidia.com/designworks/dl/Using_FFmpeg_with_NVIDIA_GPU_Hardware_Acceleration-pdf (registration required)
- Recommended string:

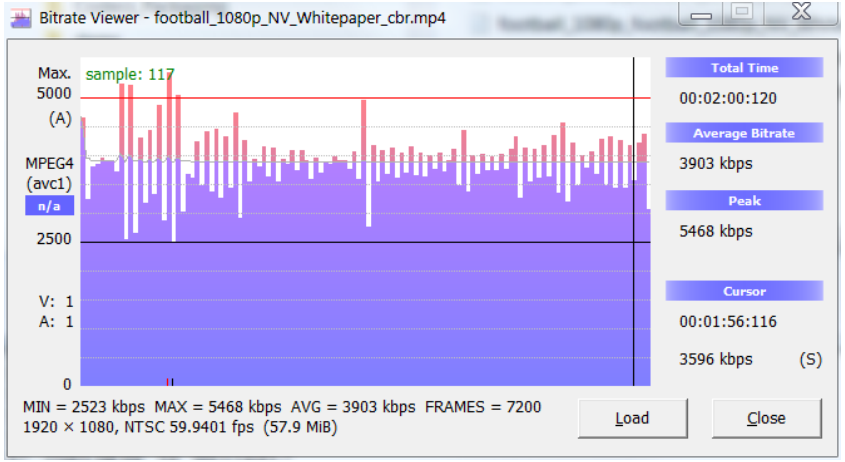
```
ffmpeg -y -vsync 0 -hwaccel cuvid -c:v h264_cuvid -i
input.mp4 -c:a copy -c:v h264_nvenc -preset slow -profile
high -b:v 5M -bufsize 5M -maxrate 10M -qmin 0 -g 250 -bf 2 -
temporal-aq 1 -rc-lookahead 20 -i_qfactor 0.75 -b_qfactor
1.1 output.mp4
```

- Concerns:
  - Data rate fluctuations due to 2 second VBV buffer
  - Performance - Slow preset

Switch to 1 Second VBV Buffer



2 second buffer

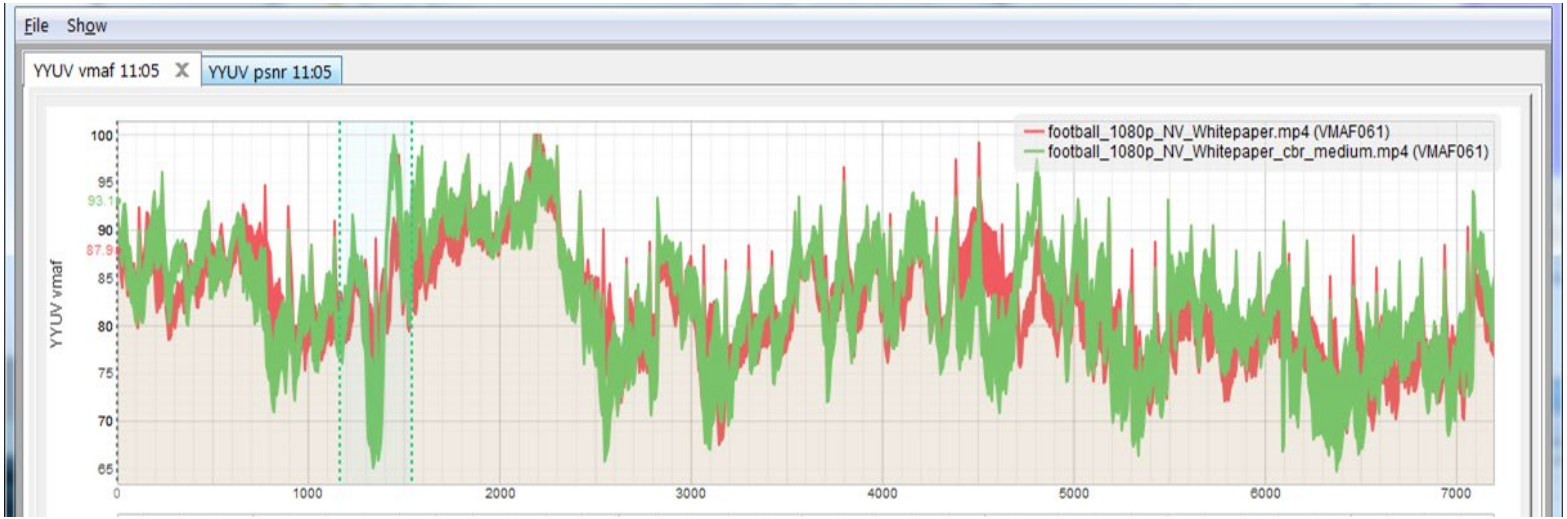


1 second buffer

- 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

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

- 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

Rez	Data rate
1080p60	6 mbps
1080p30	4 mbps
720p30	2.5 mbps
540p30	1.2 mbps
360p30	.8 mbps

NVIDIA Encodings

- Hardware decode to CUVID, then encode

```
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

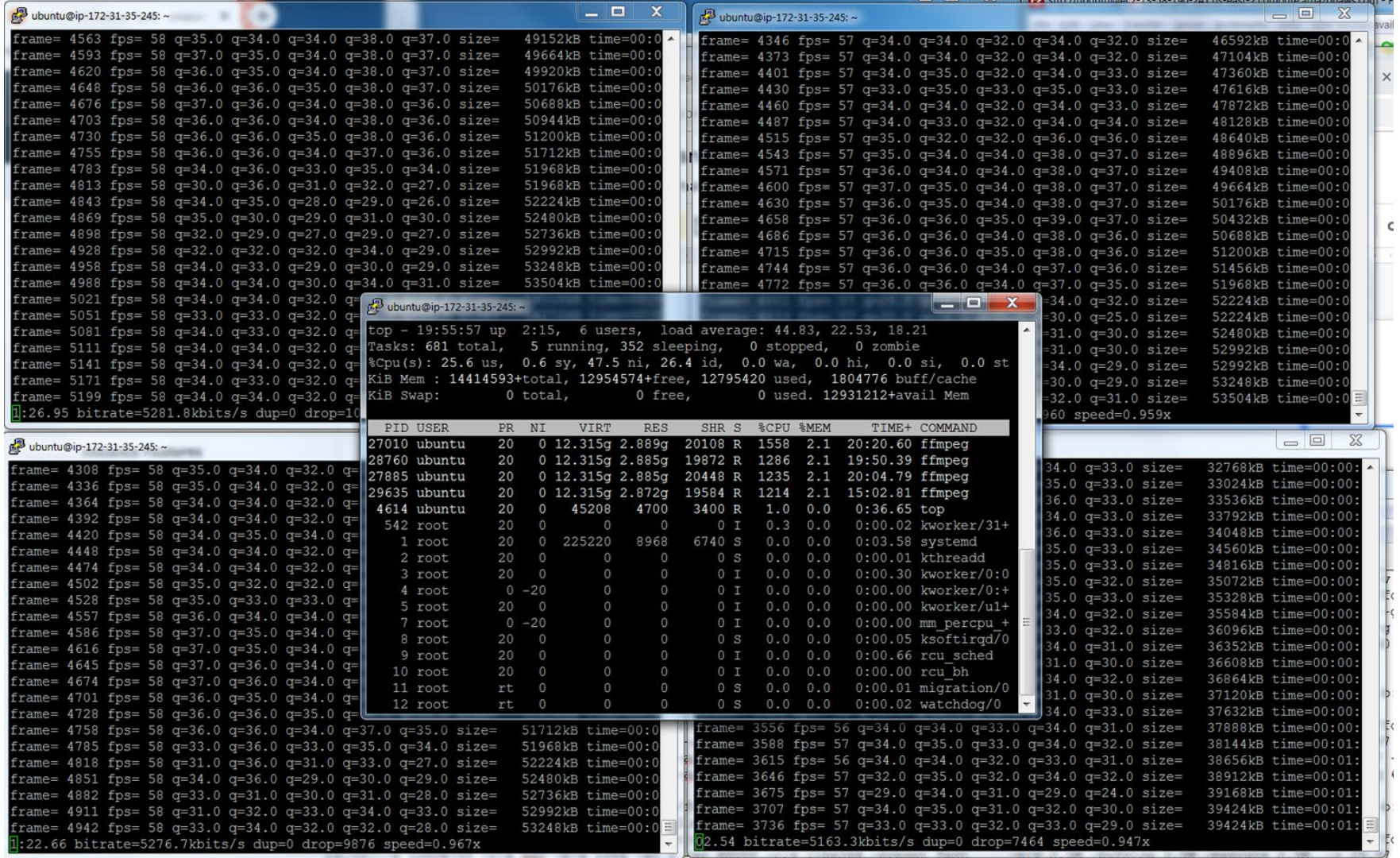
x264 Encodes

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

```
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


ubuntu@ip-172-31-35-245: ~
frame= 4563 fps= 58 q=35.0 q=34.0 q=34.0 q=38.0 q=37.0 size= 49152kB time=00:0
frame= 4593 fps= 58 q=37.0 q=35.0 q=34.0 q=38.0 q=37.0 size= 49664kB time=00:0
frame= 4620 fps= 58 q=36.0 q=35.0 q=34.0 q=38.0 q=37.0 size= 49920kB time=00:0
frame= 4648 fps= 58 q=36.0 q=36.0 q=35.0 q=38.0 q=37.0 size= 50176kB time=00:0
frame= 4676 fps= 58 q=37.0 q=36.0 q=34.0 q=38.0 q=36.0 size= 50688kB time=00:0
frame= 4703 fps= 58 q=36.0 q=36.0 q=34.0 q=38.0 q=36.0 size= 50944kB time=00:0
frame= 4730 fps= 58 q=36.0 q=36.0 q=35.0 q=38.0 q=36.0 size= 51200kB time=00:0
frame= 4755 fps= 58 q=36.0 q=36.0 q=34.0 q=37.0 q=36.0 size= 51712kB time=00:0
frame= 4783 fps= 58 q=34.0 q=36.0 q=33.0 q=35.0 q=34.0 size= 51968kB time=00:0
frame= 4813 fps= 58 q=30.0 q=36.0 q=31.0 q=32.0 q=27.0 size= 51968kB time=00:0
frame= 4843 fps= 58 q=34.0 q=35.0 q=28.0 q=29.0 q=26.0 size= 52224kB time=00:0
frame= 4869 fps= 58 q=35.0 q=30.0 q=29.0 q=31.0 q=30.0 size= 52480kB time=00:0
frame= 4898 fps= 58 q=32.0 q=29.0 q=27.0 q=29.0 q=27.0 size= 52736kB time=00:0
frame= 4928 fps= 58 q=32.0 q=32.0 q=32.0 q=34.0 q=29.0 size= 52992kB time=00:0
frame= 4958 fps= 58 q=34.0 q=33.0 q=29.0 q=30.0 q=29.0 size= 53248kB time=00:0
frame= 4988 fps= 58 q=34.0 q=34.0 q=30.0 q=34.0 q=31.0 size= 53504kB time=00:0
frame= 5021 fps= 58 q=34.0 q=34.0 q=32.0 q=
frame= 5051 fps= 58 q=33.0 q=33.0 q=32.0 q=
frame= 5081 fps= 58 q=34.0 q=33.0 q=32.0 q=
frame= 5111 fps= 58 q=34.0 q=34.0 q=32.0 q=
frame= 5141 fps= 58 q=34.0 q=34.0 q=32.0 q=
frame= 5171 fps= 58 q=34.0 q=33.0 q=32.0 q=
frame= 5199 fps= 58 q=34.0 q=34.0 q=32.0 q=
]:26.95 bitrate=5281.8kbits/s dup=0 drop=10
ubuntu@ip-172-31-35-245: ~
frame= 4346 fps= 57 q=34.0 q=34.0 q=32.0 q=34.0 q=32.0 size= 46592kB time=00:0
frame= 4373 fps= 57 q=34.0 q=34.0 q=32.0 q=34.0 q=32.0 size= 47104kB time=00:0
frame= 4401 fps= 57 q=34.0 q=35.0 q=32.0 q=34.0 q=33.0 size= 47360kB time=00:0
frame= 4430 fps= 57 q=33.0 q=35.0 q=33.0 q=35.0 q=33.0 size= 47616kB time=00:0
frame= 4460 fps= 57 q=34.0 q=34.0 q=32.0 q=34.0 q=33.0 size= 47872kB time=00:0
frame= 4487 fps= 57 q=34.0 q=33.0 q=32.0 q=34.0 q=34.0 size= 48128kB time=00:0
frame= 4515 fps= 57 q=35.0 q=32.0 q=32.0 q=36.0 q=36.0 size= 48640kB time=00:0
frame= 4543 fps= 57 q=35.0 q=34.0 q=34.0 q=38.0 q=37.0 size= 48896kB time=00:0
frame= 4571 fps= 57 q=36.0 q=34.0 q=34.0 q=38.0 q=37.0 size= 49408kB time=00:0
frame= 4600 fps= 57 q=37.0 q=35.0 q=34.0 q=38.0 q=37.0 size= 49664kB time=00:0
frame= 4630 fps= 57 q=36.0 q=35.0 q=34.0 q=38.0 q=37.0 size= 50176kB time=00:0
frame= 4658 fps= 57 q=36.0 q=36.0 q=35.0 q=39.0 q=37.0 size= 50432kB time=00:0
frame= 4686 fps= 57 q=36.0 q=36.0 q=34.0 q=38.0 q=36.0 size= 50688kB time=00:0
frame= 4715 fps= 57 q=36.0 q=36.0 q=35.0 q=38.0 q=36.0 size= 51200kB time=00:0
frame= 4744 fps= 57 q=36.0 q=36.0 q=34.0 q=37.0 q=36.0 size= 51456kB time=00:0
frame= 4772 fps= 57 q=36.0 q=36.0 q=34.0 q=37.0 q=35.0 size= 51968kB time=00:0
960 speed=0.959x
ubuntu@ip-172-31-35-245: ~
top - 19:55:57 up 2:15, 6 users, load average: 44.83, 22.53, 18.21
Tasks: 681 total, 5 running, 352 sleeping, 0 stopped, 0 zombie
%Cpu(s): 25.6 us, 0.6 sy, 47.5 ni, 26.4 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 14414593+total, 12954574+free, 12795420 used, 1804776 buff/cache
KiB Swap: 0 total, 0 free, 0 used. 12931212+avail Mem
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
27010 ubuntu 20 0 12.315g 2.889g 20108 R 1558 2.1 20:20.60 ffmpeg
28760 ubuntu 20 0 12.315g 2.885g 19872 R 1286 2.1 19:50.39 ffmpeg
27885 ubuntu 20 0 12.315g 2.885g 20448 R 1235 2.1 20:04.79 ffmpeg
29635 ubuntu 20 0 12.315g 2.872g 19584 R 1214 2.1 15:02.81 ffmpeg
4614 ubuntu 20 0 45208 4700 3400 R 1.0 0.0 0:36.65 top
542 root 20 0 0 0 0 I 0.3 0.0 0:00.02 kworker/31+
1 root 20 0 225220 8968 6740 S 0.0 0.0 0:03.58 systemd
2 root 20 0 0 0 0 S 0.0 0.0 0:00.01 kthreadd
3 root 20 0 0 0 0 I 0.0 0.0 0:00.30 kworker/0:0
4 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/0:+
5 root 20 0 0 0 0 I 0.0 0.0 0:00.00 kworker/u1+
7 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 mm_percpu_+
8 root 20 0 0 0 0 S 0.0 0.0 0:00.05 ksoftirqd/0
9 root 20 0 0 0 0 I 0.0 0.0 0:00.66 rcu_sched
10 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rcu_bh
11 root rt 0 0 0 0 S 0.0 0.0 0:00.01 migration/0
12 root rt 0 0 0 0 S 0.0 0.0 0:00.02 watchdog/0
ubuntu@ip-172-31-35-245: ~
frame= 4308 fps= 58 q=35.0 q=34.0 q=32.0 q=
frame= 4336 fps= 58 q=35.0 q=34.0 q=32.0 q=
frame= 4364 fps= 58 q=34.0 q=34.0 q=32.0 q=
frame= 4392 fps= 58 q=34.0 q=34.0 q=32.0 q=
frame= 4420 fps= 58 q=34.0 q=35.0 q=34.0 q=
frame= 4448 fps= 58 q=34.0 q=34.0 q=32.0 q=
frame= 4474 fps= 58 q=34.0 q=34.0 q=32.0 q=
frame= 4502 fps= 58 q=35.0 q=32.0 q=32.0 q=
frame= 4528 fps= 58 q=35.0 q=33.0 q=33.0 q=
frame= 4557 fps= 58 q=36.0 q=34.0 q=34.0 q=
frame= 4586 fps= 58 q=37.0 q=35.0 q=34.0 q=
frame= 4616 fps= 58 q=37.0 q=35.0 q=34.0 q=
frame= 4645 fps= 58 q=37.0 q=36.0 q=34.0 q=
frame= 4674 fps= 58 q=37.0 q=36.0 q=34.0 q=
frame= 4701 fps= 58 q=36.0 q=35.0 q=34.0 q=
frame= 4728 fps= 58 q=36.0 q=36.0 q=35.0 q=
frame= 4758 fps= 58 q=36.0 q=36.0 q=34.0 q=37.0 q=35.0 size= 51712kB time=00:0
frame= 4785 fps= 58 q=33.0 q=36.0 q=33.0 q=35.0 q=34.0 size= 51968kB time=00:0
frame= 4818 fps= 58 q=31.0 q=36.0 q=31.0 q=33.0 q=27.0 size= 52224kB time=00:0
frame= 4851 fps= 58 q=34.0 q=36.0 q=29.0 q=30.0 q=29.0 size= 52480kB time=00:0
frame= 4882 fps= 58 q=33.0 q=31.0 q=30.0 q=31.0 q=28.0 size= 52736kB time=00:0
frame= 4911 fps= 58 q=31.0 q=32.0 q=33.0 q=34.0 q=33.0 size= 52992kB time=00:0
frame= 4942 fps= 58 q=33.0 q=34.0 q=32.0 q=32.0 q=28.0 size= 53248kB time=00:0
]:22.66 bitrate=5276.7kbits/s dup=0 drop=9876 speed=0.967x
frame= 3556 fps= 56 q=34.0 q=34.0 q=33.0 q=34.0 q=31.0 size= 37888kB time=00:00:
frame= 3588 fps= 57 q=34.0 q=35.0 q=33.0 q=34.0 q=32.0 size= 38144kB time=00:01:
frame= 3615 fps= 56 q=34.0 q=34.0 q=32.0 q=33.0 q=31.0 size= 38656kB time=00:01:
frame= 3646 fps= 57 q=32.0 q=35.0 q=32.0 q=34.0 q=32.0 size= 38912kB time=00:01:
frame= 3675 fps= 57 q=29.0 q=34.0 q=31.0 q=29.0 q=24.0 size= 39168kB time=00:01:
frame= 3707 fps= 57 q=34.0 q=35.0 q=31.0 q=32.0 q=30.0 size= 39424kB time=00:01:
frame= 3736 fps= 57 q=33.0 q=33.0 q=32.0 q=33.0 q=29.0 size= 39424kB time=00:01:
02.54 bitrate=5163.3kbits/s dup=0 drop=7464 speed=0.947x

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

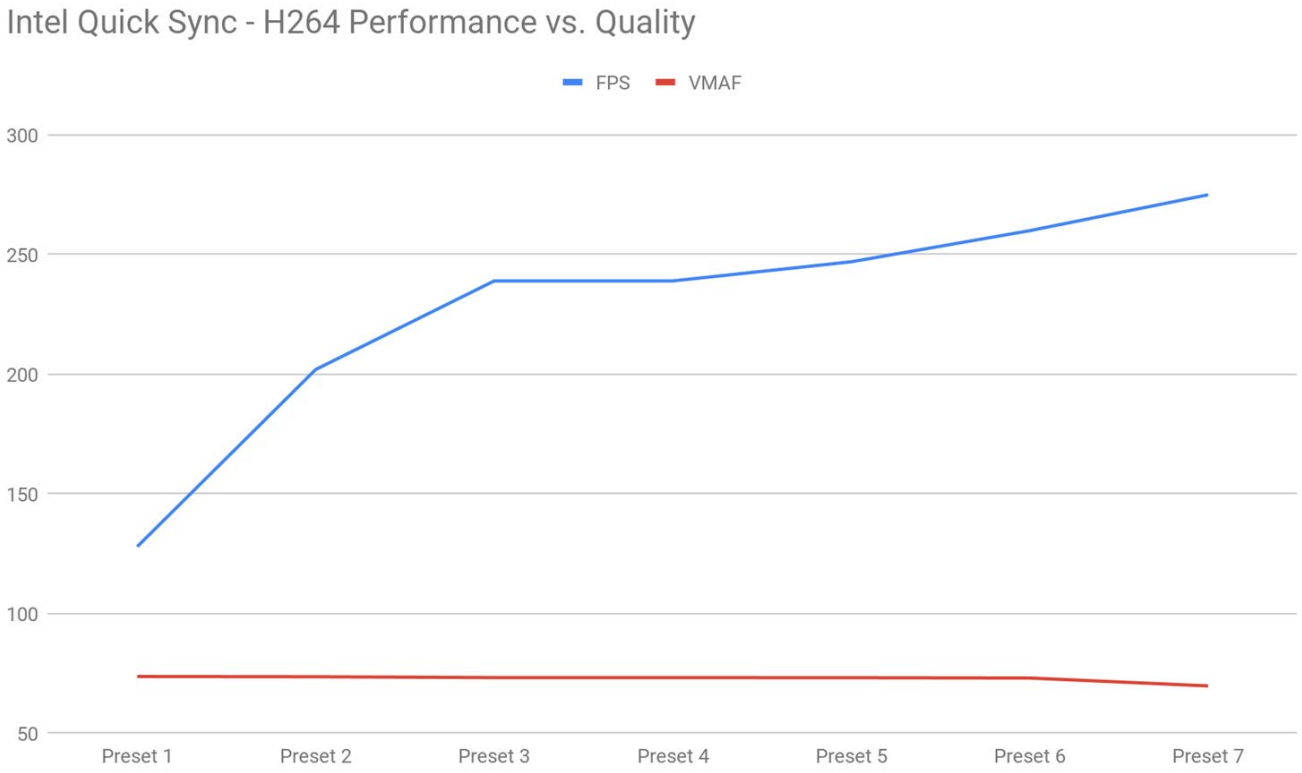
Intel Quick Sync Encoding

- 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

	FPS	VMAF
Preset 1	128	73.75
Preset 2	202	73.64
Preset 3	239	73.29
Preset 4	239	73.29
Preset 5	247	73.25
Preset 6	260	73.11
Preset 7	275	69.82

Tested at preset 4



FFmpeg Script (Intel Provided)

```
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

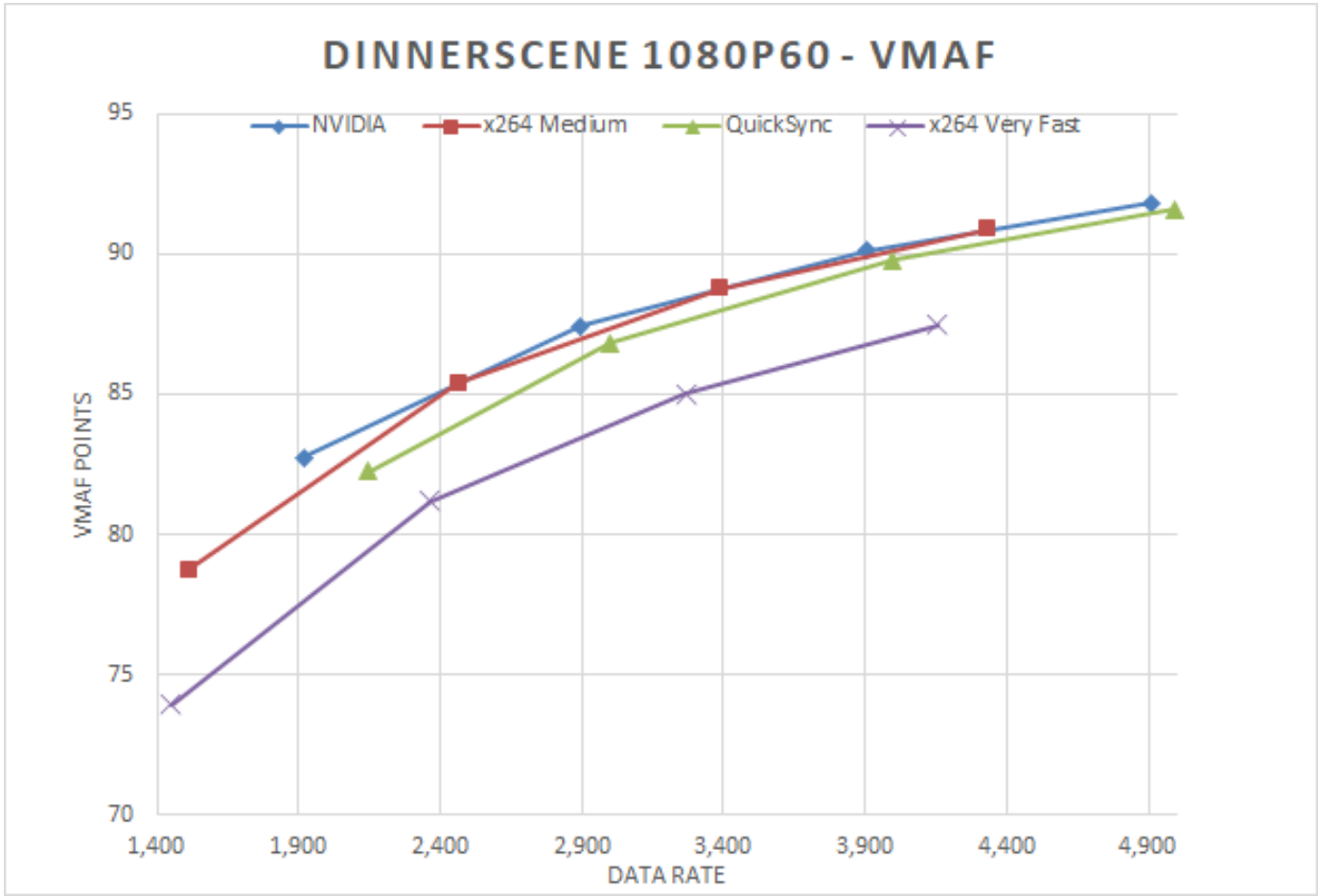
- 1 encoding ladder with Quick Sync at preset 4
  - Using preset 7 did not deliver 2 full ladders
- No ladders with x264, even using veryfast preset
- Obviously could get higher performance with other systems
- Had hope to use exclusively AWS computers to get pricing, but went with Intel supplied computers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

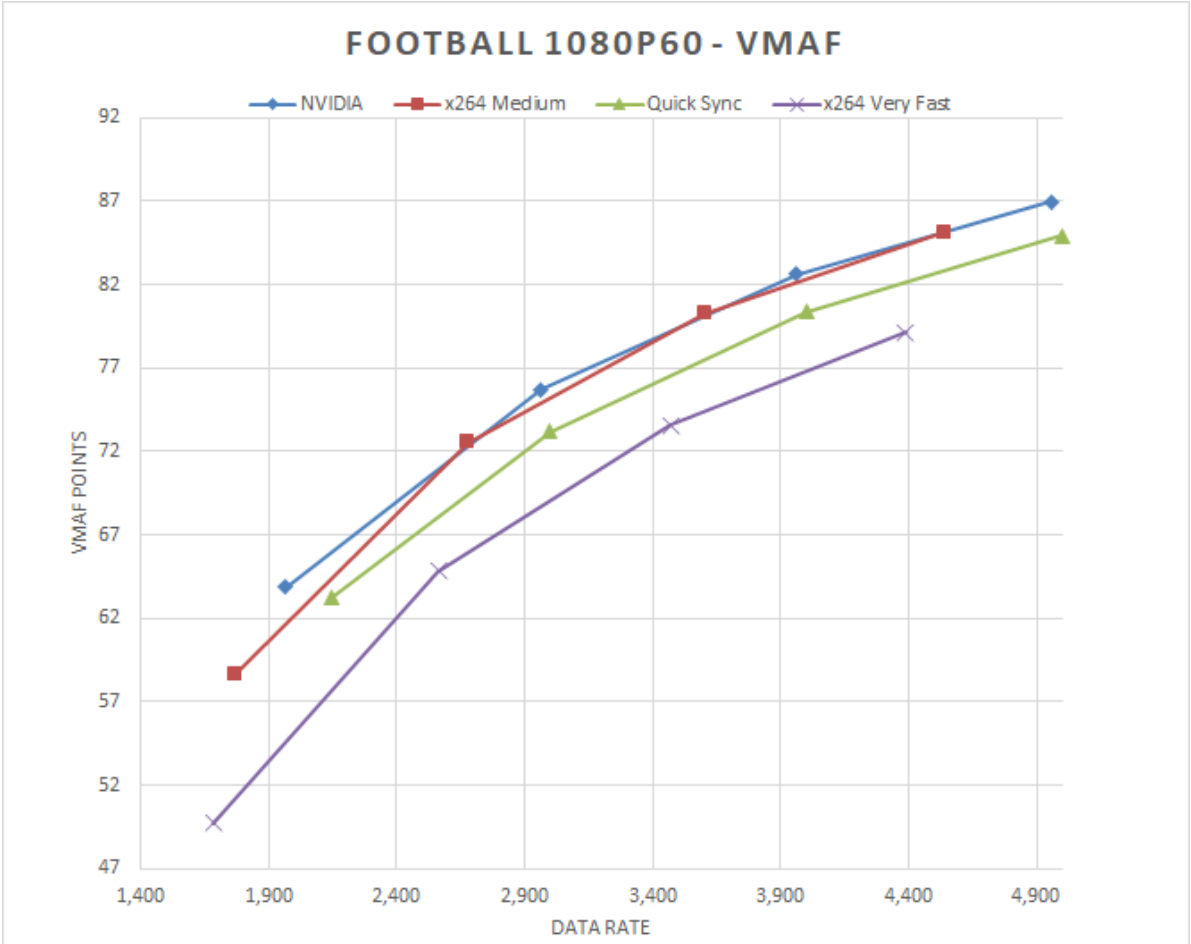
Dinner Scene - Rate Distortion Curve



Dinner Scene - BD-Rate Computations

VMAF	NVIDIA	x264 Medium	QuickSync	x264 Very Fast
NVIDIA	X	-2.42	-8.94	-29.11
x264 Medium	2.48	X	-7.17	-25.55
QuickSync	9.82	7.72	X	-20.19
x264 Very Fast	41.07	34.31	25.29	X

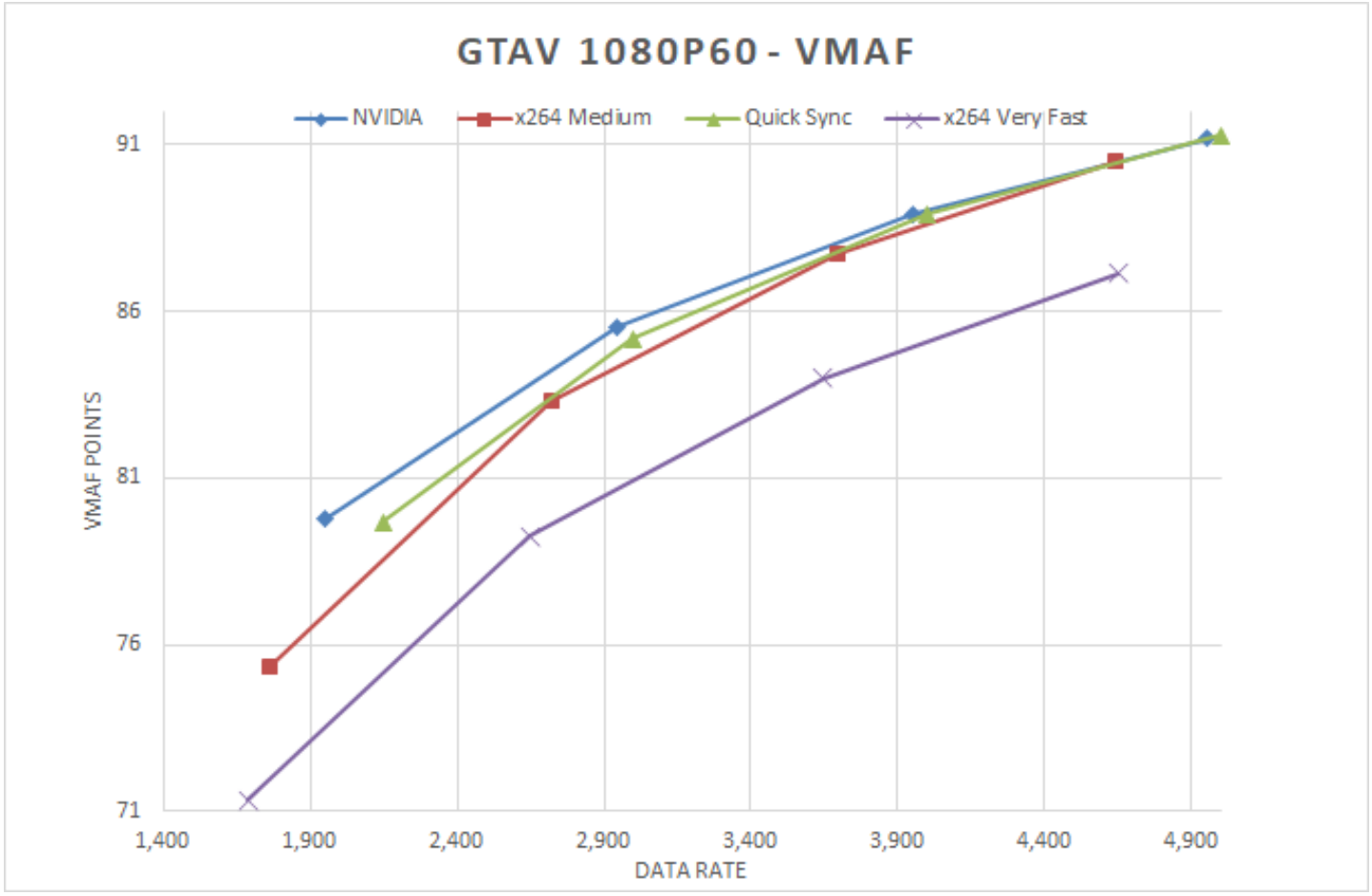
Football - Rate Distortion Curve



Football - BD-Rate Computations

VMAF	NVIDIA	x264 Medium	Quick Sync	x264 Very Fast
NVIDIA	X	-1.68	-10.20	-21.04
x264 Medium	1.71	X	-8.60	-18.99
Quick Sync	11.36	9.41	X	-12.27
x264 Very Fast	26.64	23.44	13.99	X

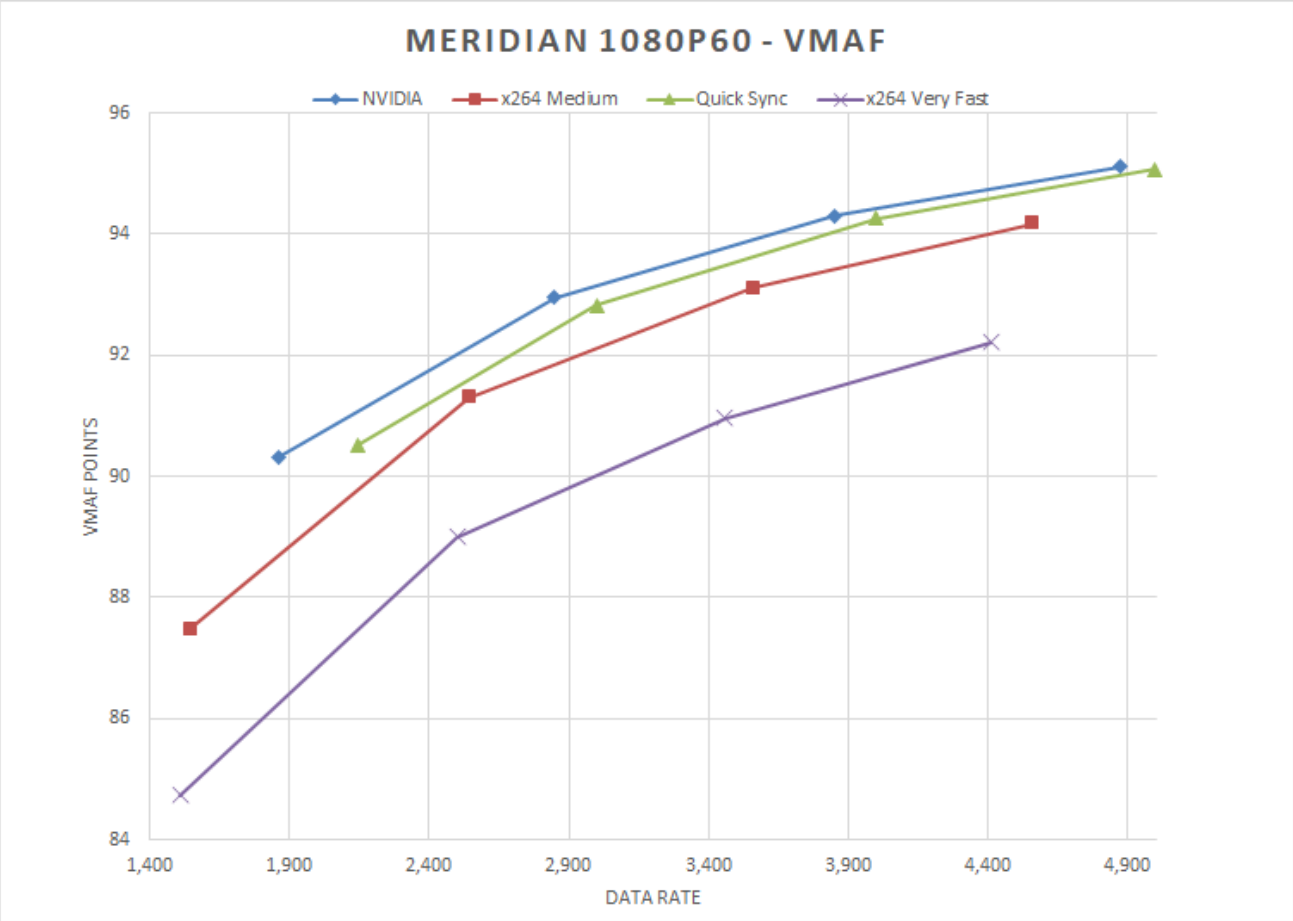
GTAV - Rate Distortion Curve



GTAV - BD-Rate Computations

VMAF	NVIDIA	x264 Medium	Quick Sync	x264 Very Fast
NVIDIA	X	-6.97	-4.51	-28.41
x264 Medium	7.50	X	2.36	-19.92
Quick Sync	4.72	-2.31	X	-23.55
x264 Very Fast	39.68	24.87	30.81	X

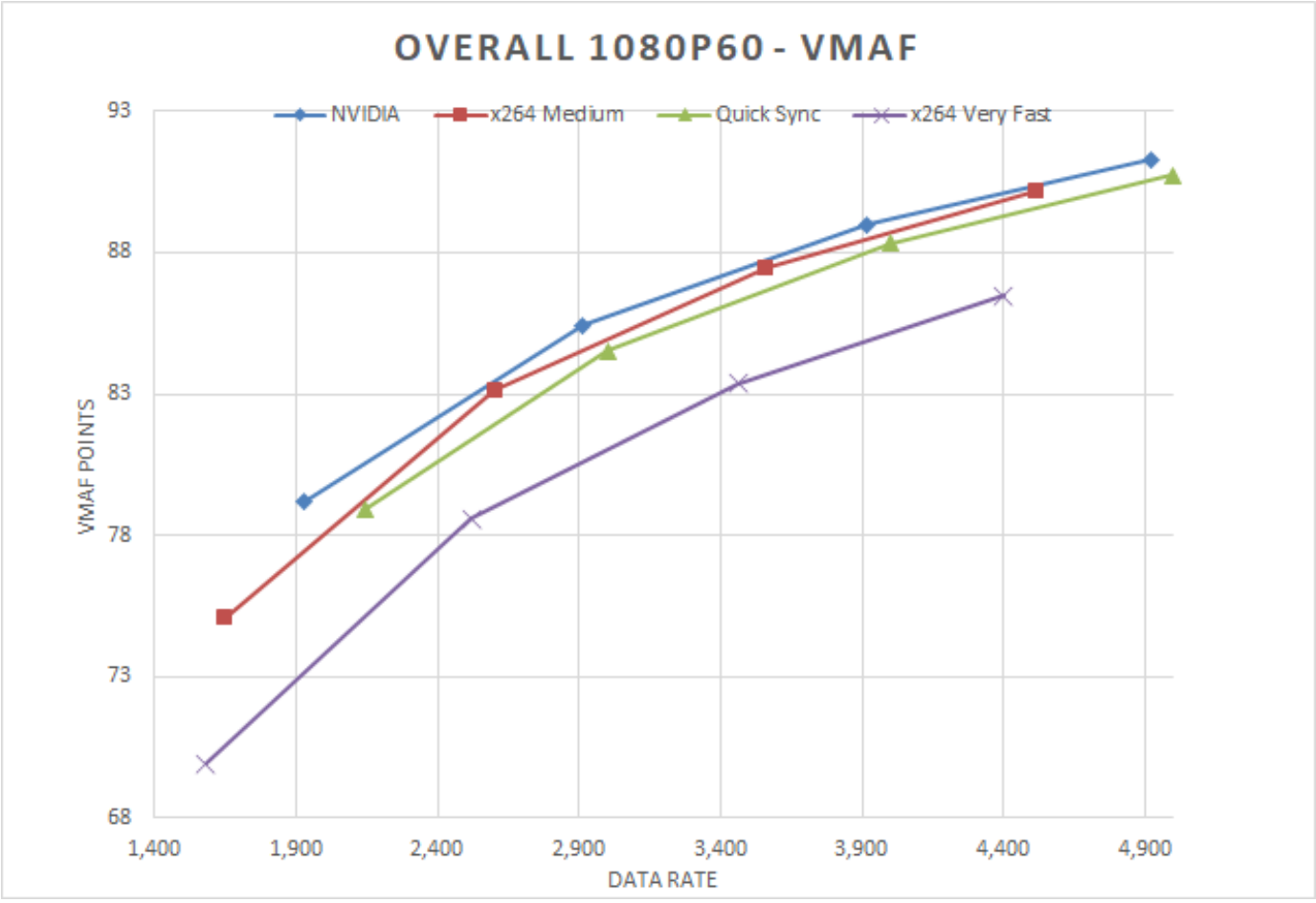
Meridian - Rate Distortion Curve



Meridian - BD Rate

VMAF	NVIDIA	x264 Medium	Quick Sync	x264 Very Fast
NVIDIA	X	-16.64	-7.47	-41.58
x264 Medium	19.96	X	10.18	-28.04
Quick Sync	8.08	-9.24	X	-35.31
x264 Very Fast	71.18	38.97	54.58	X

Overall - Rate Distortion Curve



Overall - BD Rate

VMAF	NVIDIA	x264 Medium	Quick Sync	x264 Very Fast
NVIDIA	X	-4.49	-9.08	-27.04
x264 Medium	4.70	X	-4.92	-22.30
Quick Sync	9.99	5.17	X	-18.97
x264 Very Fast	37.06	28.69	23.41	X

Football - VMAF - Plot NVIDIA vs. VeryFast

- Some major differences in ratings
- Actual visual differences not that signficant

Football - VMAF - Plot NVIDIA vs. Quick Sync

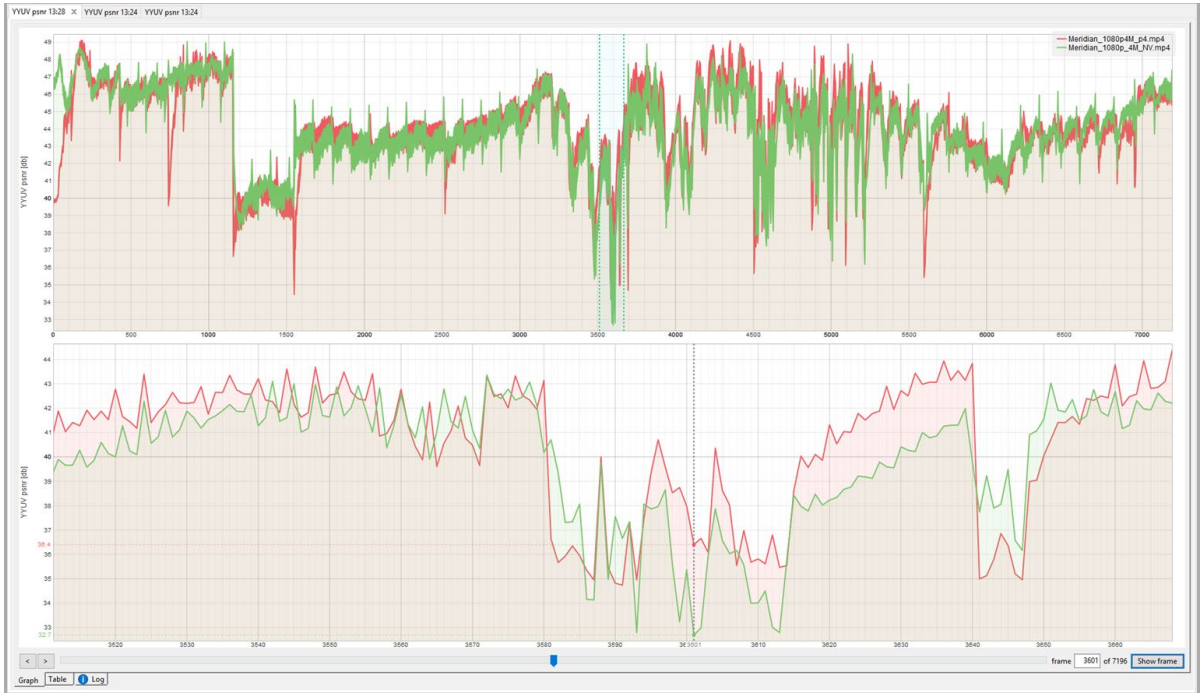
- Some major differences

Major Quality Differences - Original

Major Quality Differences - NVIDIA

Major Quality Differences - Quick Sync

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

H.264 Summary

VMAF	NVIDIA	x264 Medium	Quick Sync	x264 Very Fast
NVIDIA	X	-4.49	-9.08	-27.04
x264 Medium	4.70	X	-4.92	-22.30
Quick Sync	9.99	5.17	X	-18.97
x264 Very Fast	37.06	28.69	23.41	X

- 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

HEVC

- 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

```
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

Intel SVT-HEVC

- 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

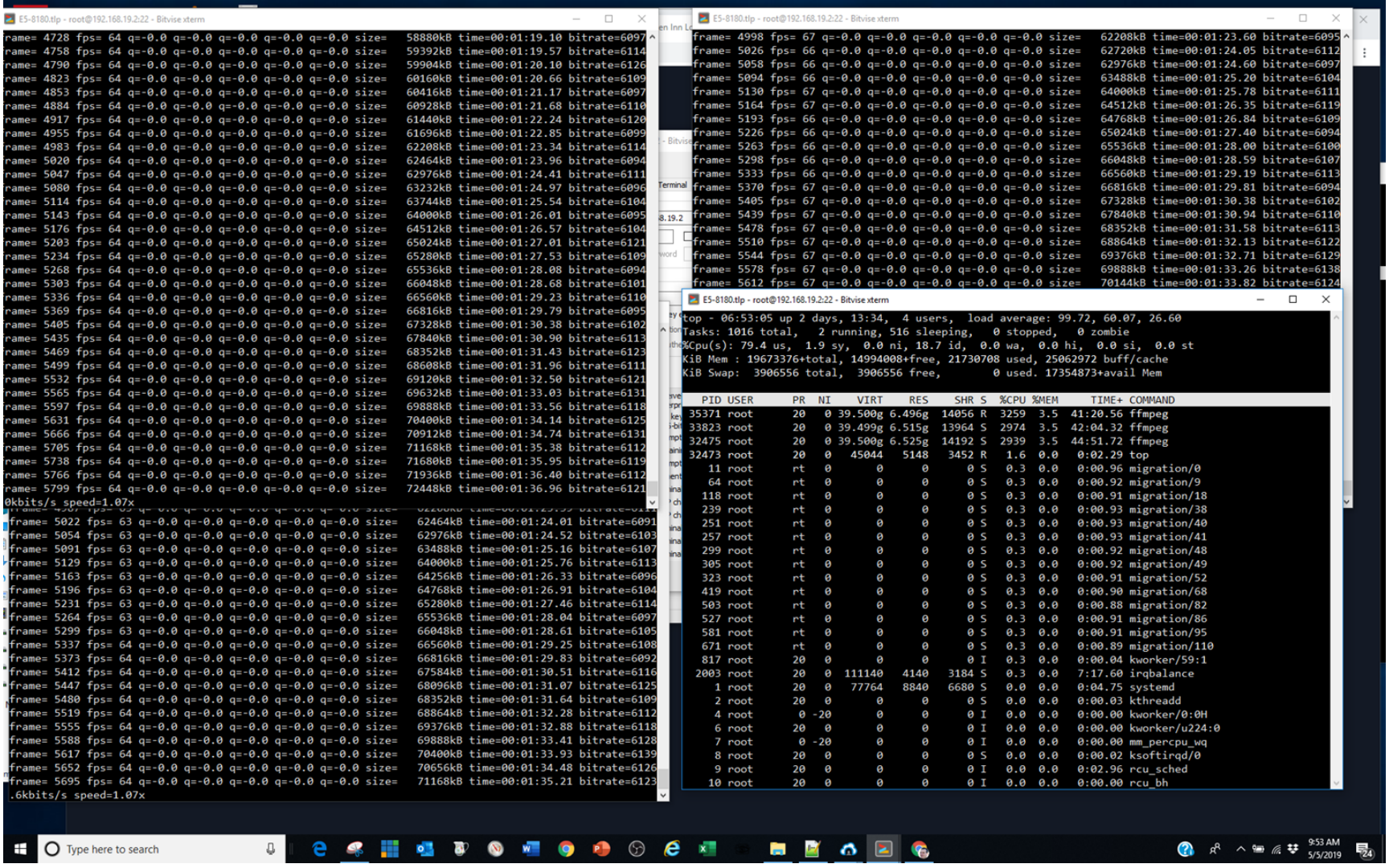
Intel Script

```
./ffmpeg -y -i football_1080p.mp4 -c:v libsvt_hevc -rc 1 -tune 1 -
preset 1  -b:v 4M -maxrate 4M -bufsize 8M -c:a aac -b:a 128k -ac 2 -
ar 48000 football_SVT_HEVC_4M_p1.mp4
```

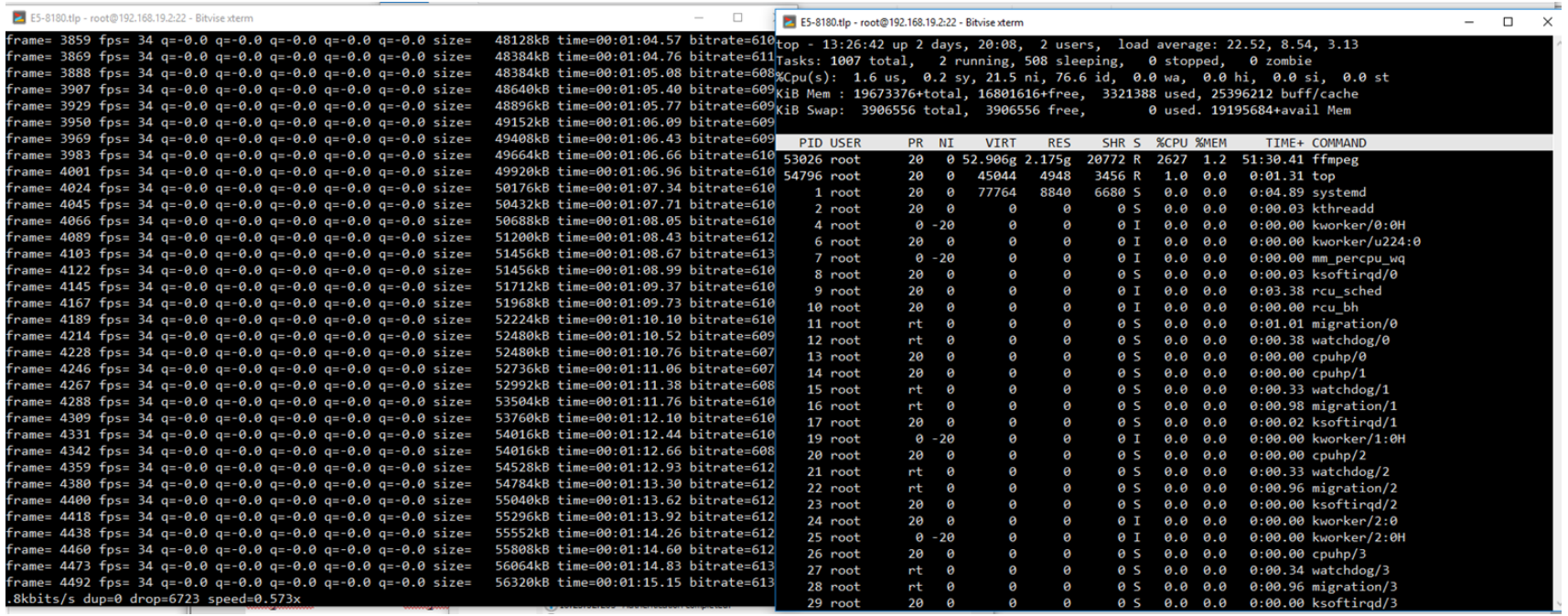
- Intel supplied
- Doubled buffer size wherever possible on HEVC encodes

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


E5-8180.tlp - root@192.168.19.2:22 - Bitvise xterm
frame= 4728 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 58880kB time=00:01:19.10 bitrate=6097
frame= 4758 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 59392kB time=00:01:19.57 bitrate=6114
frame= 4790 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 59904kB time=00:01:20.10 bitrate=6126
frame= 4823 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 60160kB time=00:01:20.66 bitrate=6109
frame= 4853 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 60416kB time=00:01:21.17 bitrate=6097
frame= 4884 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 60928kB time=00:01:21.68 bitrate=6110
frame= 4917 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 61440kB time=00:01:22.24 bitrate=6120
frame= 4955 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 61696kB time=00:01:22.85 bitrate=6099
frame= 4983 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 62208kB time=00:01:23.34 bitrate=6114
frame= 5020 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 62464kB time=00:01:23.96 bitrate=6094
frame= 5047 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 62976kB time=00:01:24.41 bitrate=6111
frame= 5080 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 63232kB time=00:01:24.97 bitrate=6096
frame= 5114 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 63744kB time=00:01:25.54 bitrate=6104
frame= 5143 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 64000kB time=00:01:26.01 bitrate=6095
frame= 5176 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 64512kB time=00:01:26.57 bitrate=6104
frame= 5203 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 65024kB time=00:01:27.01 bitrate=6121
frame= 5234 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 65280kB time=00:01:27.53 bitrate=6109
frame= 5268 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 65536kB time=00:01:28.08 bitrate=6094
frame= 5303 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 66048kB time=00:01:28.68 bitrate=6101
frame= 5336 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 66560kB time=00:01:29.23 bitrate=6110
frame= 5369 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 66816kB time=00:01:29.79 bitrate=6095
frame= 5405 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 67328kB time=00:01:30.38 bitrate=6102
frame= 5435 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 67840kB time=00:01:30.90 bitrate=6113
frame= 5469 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 68352kB time=00:01:31.43 bitrate=6123
frame= 5499 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 68608kB time=00:01:31.96 bitrate=6111
frame= 5532 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 69120kB time=00:01:32.50 bitrate=6121
frame= 5565 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 69632kB time=00:01:33.03 bitrate=6131
frame= 5597 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 69888kB time=00:01:33.56 bitrate=6118
frame= 5631 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 70400kB time=00:01:34.14 bitrate=6125
frame= 5666 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 70912kB time=00:01:34.74 bitrate=6131
frame= 5705 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 71168kB time=00:01:35.38 bitrate=6112
frame= 5738 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 71680kB time=00:01:35.95 bitrate=6119
frame= 5766 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 71936kB time=00:01:36.40 bitrate=6112
frame= 5799 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 72448kB time=00:01:36.96 bitrate=6121
0kbits/s speed=1.07x
frame= 5022 fps= 63 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 62464kB time=00:01:24.01 bitrate=6091
frame= 5054 fps= 63 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 62976kB time=00:01:24.52 bitrate=6103
frame= 5091 fps= 63 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 63488kB time=00:01:25.16 bitrate=6107
frame= 5129 fps= 63 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 64000kB time=00:01:25.76 bitrate=6113
frame= 5163 fps= 63 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 64256kB time=00:01:26.33 bitrate=6096
frame= 5196 fps= 63 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 64768kB time=00:01:26.91 bitrate=6104
frame= 5231 fps= 63 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 65280kB time=00:01:27.46 bitrate=6114
frame= 5264 fps= 63 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 65536kB time=00:01:28.04 bitrate=6097
frame= 5299 fps= 63 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 66048kB time=00:01:28.61 bitrate=6105
frame= 5337 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 66560kB time=00:01:29.25 bitrate=6108
frame= 5373 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 66816kB time=00:01:29.83 bitrate=6092
frame= 5412 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 67584kB time=00:01:30.51 bitrate=6116
frame= 5447 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 68096kB time=00:01:31.07 bitrate=6125
frame= 5480 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 68352kB time=00:01:31.64 bitrate=6109
frame= 5519 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 68864kB time=00:01:32.28 bitrate=6112
frame= 5555 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 69376kB time=00:01:32.88 bitrate=6118
frame= 5588 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 69888kB time=00:01:33.41 bitrate=6128
frame= 5617 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 70400kB time=00:01:33.93 bitrate=6139
frame= 5652 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 70656kB time=00:01:34.48 bitrate=6126
frame= 5695 fps= 64 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 71168kB time=00:01:35.21 bitrate=6123
.6kbits/s speed=1.07x
E5-8180.tlp - root@192.168.19.2:22 - Bitvise xterm
frame= 4998 fps= 67 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 62208kB time=00:01:23.60 bitrate=6095
frame= 5026 fps= 66 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 62720kB time=00:01:24.05 bitrate=6112
frame= 5058 fps= 66 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 62976kB time=00:01:24.60 bitrate=6097
frame= 5094 fps= 66 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 63488kB time=00:01:25.20 bitrate=6104
frame= 5130 fps= 67 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 64000kB time=00:01:25.78 bitrate=6111
frame= 5164 fps= 67 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 64512kB time=00:01:26.35 bitrate=6119
frame= 5193 fps= 66 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 64768kB time=00:01:26.84 bitrate=6109
frame= 5226 fps= 66 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 65024kB time=00:01:27.40 bitrate=6094
frame= 5263 fps= 66 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 65536kB time=00:01:28.00 bitrate=6100
frame= 5298 fps= 66 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 66048kB time=00:01:28.59 bitrate=6107
frame= 5333 fps= 66 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 66560kB time=00:01:29.19 bitrate=6113
frame= 5370 fps= 67 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 66816kB time=00:01:29.81 bitrate=6094
frame= 5405 fps= 67 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 67328kB time=00:01:30.38 bitrate=6102
frame= 5439 fps= 67 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 67840kB time=00:01:30.94 bitrate=6110
frame= 5478 fps= 67 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 68352kB time=00:01:31.58 bitrate=6113
frame= 5510 fps= 67 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 68864kB time=00:01:32.13 bitrate=6122
frame= 5544 fps= 67 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 69376kB time=00:01:32.71 bitrate=6129
frame= 5578 fps= 67 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 69888kB time=00:01:33.26 bitrate=6138
frame= 5612 fps= 67 q=-0.0 q=-0.0 q=-0.0 q=-0.0 q=-0.0 size= 70144kB time=00:01:33.82 bitrate=6124
E5-8180.tlp - root@192.168.19.2:22 - Bitvise xterm
top - 06:53:05 up 2 days, 13:34, 4 users, load average: 99.72, 60.07, 26.60
Tasks: 1016 total, 2 running, 516 sleeping, 0 stopped, 0 zombie
%Cpu(s): 79.4 us, 1.9 sy, 0.0 ni, 18.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 19673376+total, 14994008+free, 21730708 used, 25062972 buff/cache
KiB Swap: 3906556 total, 3906556 free, 0 used. 17354873+avail Mem
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
35371 root 20 0 39.500g 6.496g 14056 R 3259 3.5 41:20.56 ffmpeg
33823 root 20 0 39.499g 6.515g 13964 S 2974 3.5 42:04.32 ffmpeg
32475 root 20 0 39.500g 6.525g 14192 S 2939 3.5 44:51.72 ffmpeg
32473 root 20 0 45044 5148 3452 R 1.6 0.0 0:02.29 top
11 root rt 0 0 0 0 S 0.3 0.0 0:00.96 migration/0
64 root rt 0 0 0 0 S 0.3 0.0 0:00.92 migration/9
118 root rt 0 0 0 0 S 0.3 0.0 0:00.91 migration/18
239 root rt 0 0 0 0 S 0.3 0.0 0:00.93 migration/38
251 root rt 0 0 0 0 S 0.3 0.0 0:00.93 migration/40
257 root rt 0 0 0 0 S 0.3 0.0 0:00.93 migration/41
299 root rt 0 0 0 0 S 0.3 0.0 0:00.92 migration/48
305 root rt 0 0 0 0 S 0.3 0.0 0:00.92 migration/49
323 root rt 0 0 0 0 S 0.3 0.0 0:00.91 migration/52
419 root rt 0 0 0 0 S 0.3 0.0 0:00.90 migration/68
503 root rt 0 0 0 0 S 0.3 0.0 0:00.88 migration/82
527 root rt 0 0 0 0 S 0.3 0.0 0:00.91 migration/86
581 root rt 0 0 0 0 S 0.3 0.0 0:00.91 migration/95
671 root rt 0 0 0 0 S 0.3 0.0 0:00.89 migration/110
817 root 20 0 0 0 0 I 0.3 0.0 0:00.04 kworker/59:1
2003 root 20 0 111140 4140 3184 S 0.3 0.0 7:17.60 irqbalance
1 root 20 0 77764 8840 6680 S 0.0 0.0 0:04.75 systemd
2 root 20 0 0 0 0 S 0.0 0.0 0:00.03 kthreadd
4 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/0:0H
6 root 20 0 0 0 0 I 0.0 0.0 0:00.00 kworker/u224:0
7 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 mm_percpu_wq
8 root 20 0 0 0 0 S 0.0 0.0 0:00.02 ksoftirqd/0
9 root 20 0 0 0 0 I 0.0 0.0 0:02.96 rcu_sched
10 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rcu_bh
Type here to search
9:53 AM
5/5/2019

x265 Very Fast - Not One Full Encoding Ladder



X265 Script

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

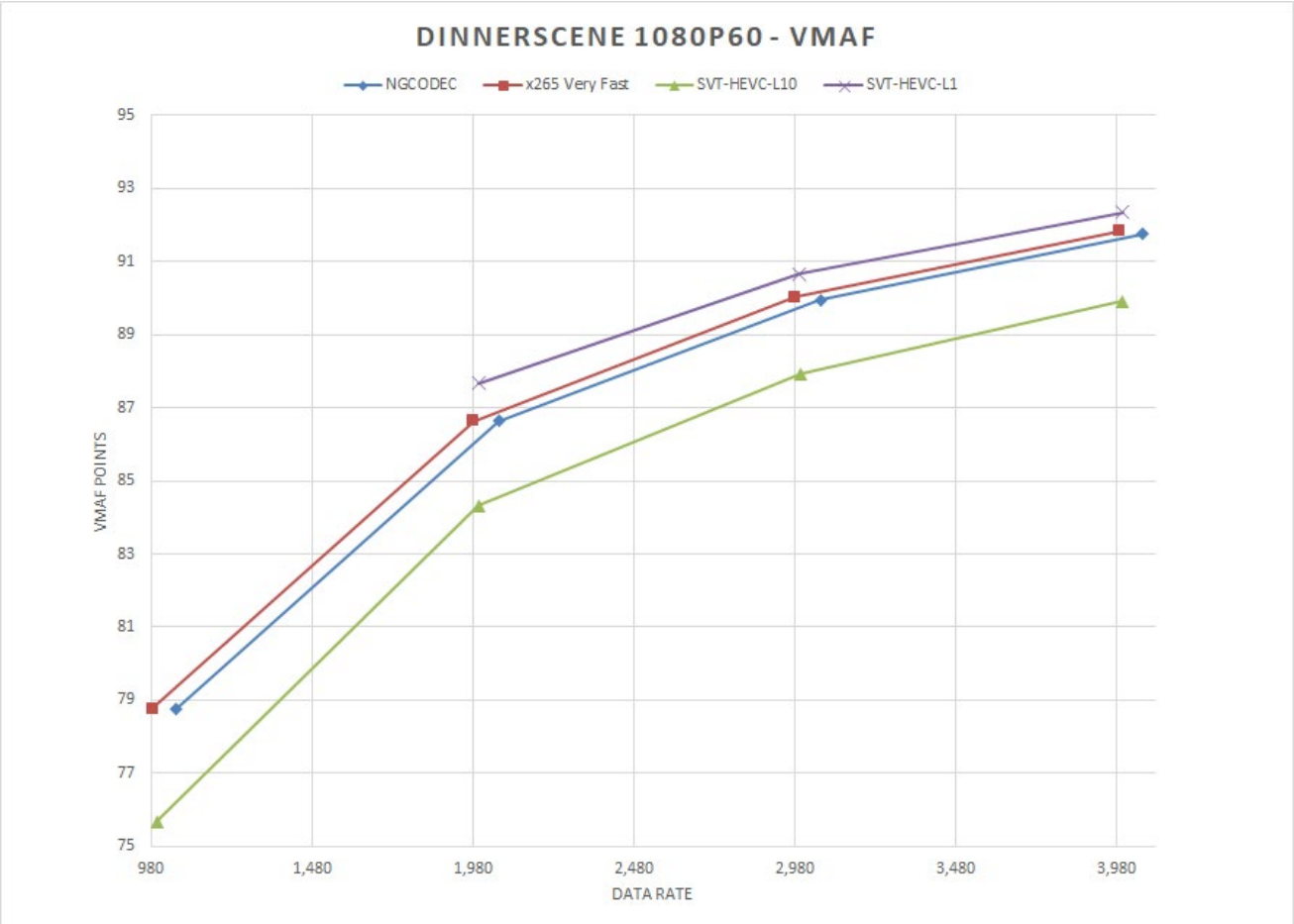
- Simple as possible

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

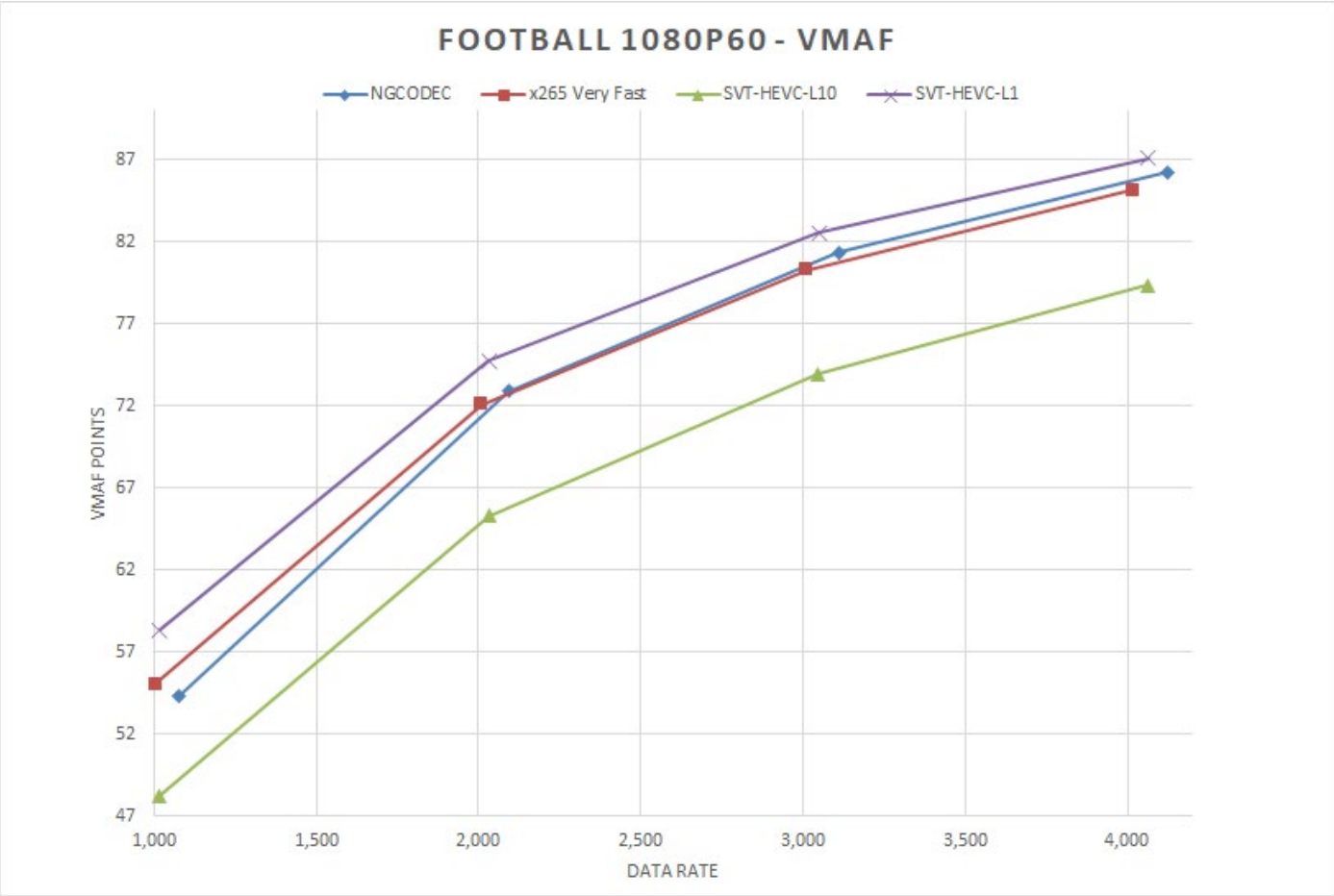
HEVC - Dinner Scene - Rate Distortion Curve



HEVC - Dinner Scene - BD-Rate Computations

VMAF	NGCODEC	x265 Very Fast	SVT-HEVC-L10	SVT-HEVC-L1
NGCODEC	X	4.46	-17.83	NA
x265 Very Fast	-4.27	X	-21.45	NA
SVT-HEVC-L10	21.70	27.31	X	NA
SVT-HEVC-L1	NA	NA	NA	X

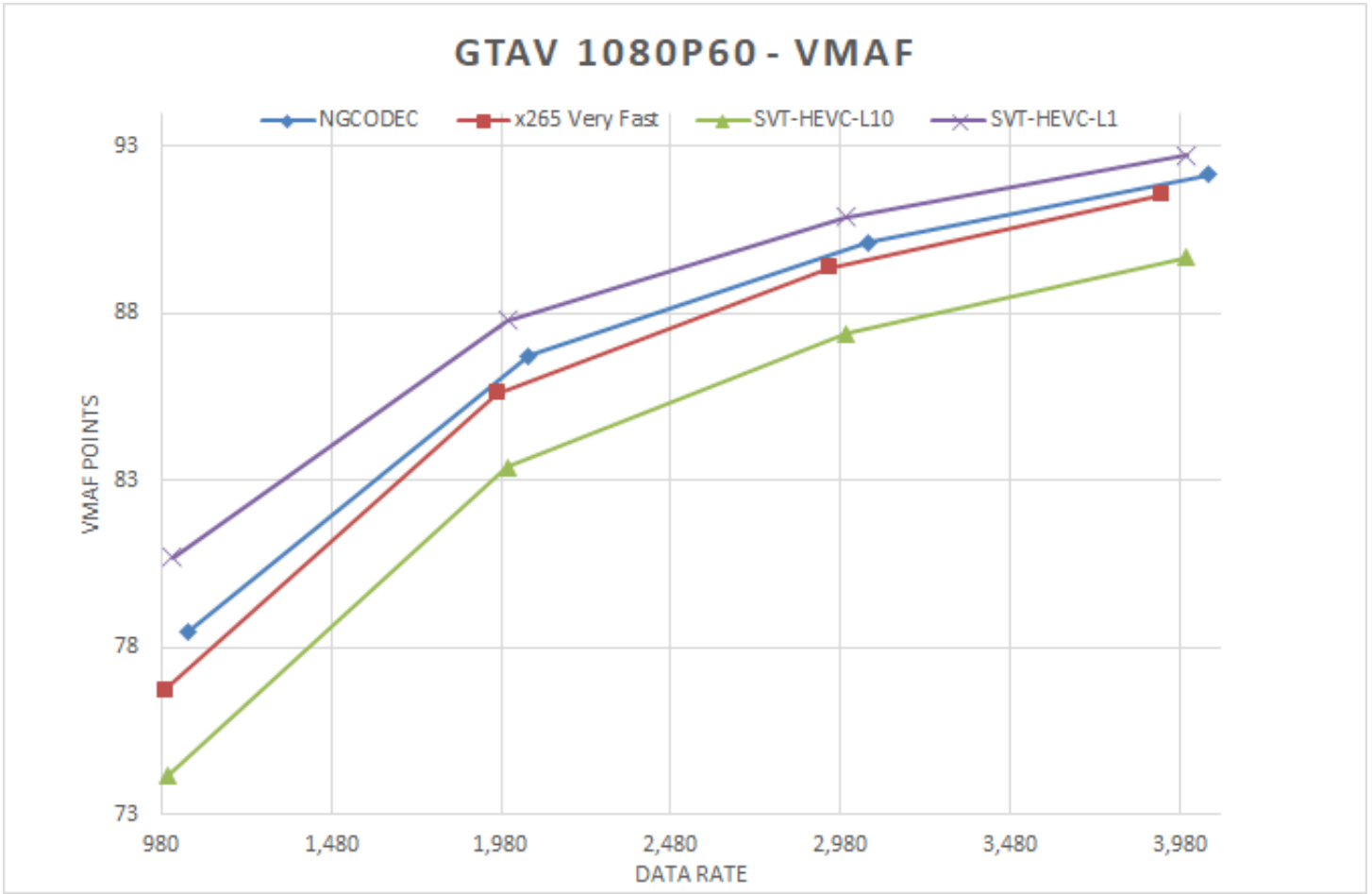
HEVC - Football - Rate Distortion Curve



HEVC - Football - BD-Rate Computations

VMAF	NGCODEC	x265 Very Fast	SVT-HEVC-L10	SVT-HEVC-L1
NGCODEC	X	1.85	-24.02	12.80
x265 Very Fast	-1.82	X	-26.45	11.83
SVT-HEVC-L10	31.62	35.97	X	52.48
SVT-HEVC-L1	-11.35	-10.58	-34.42	X

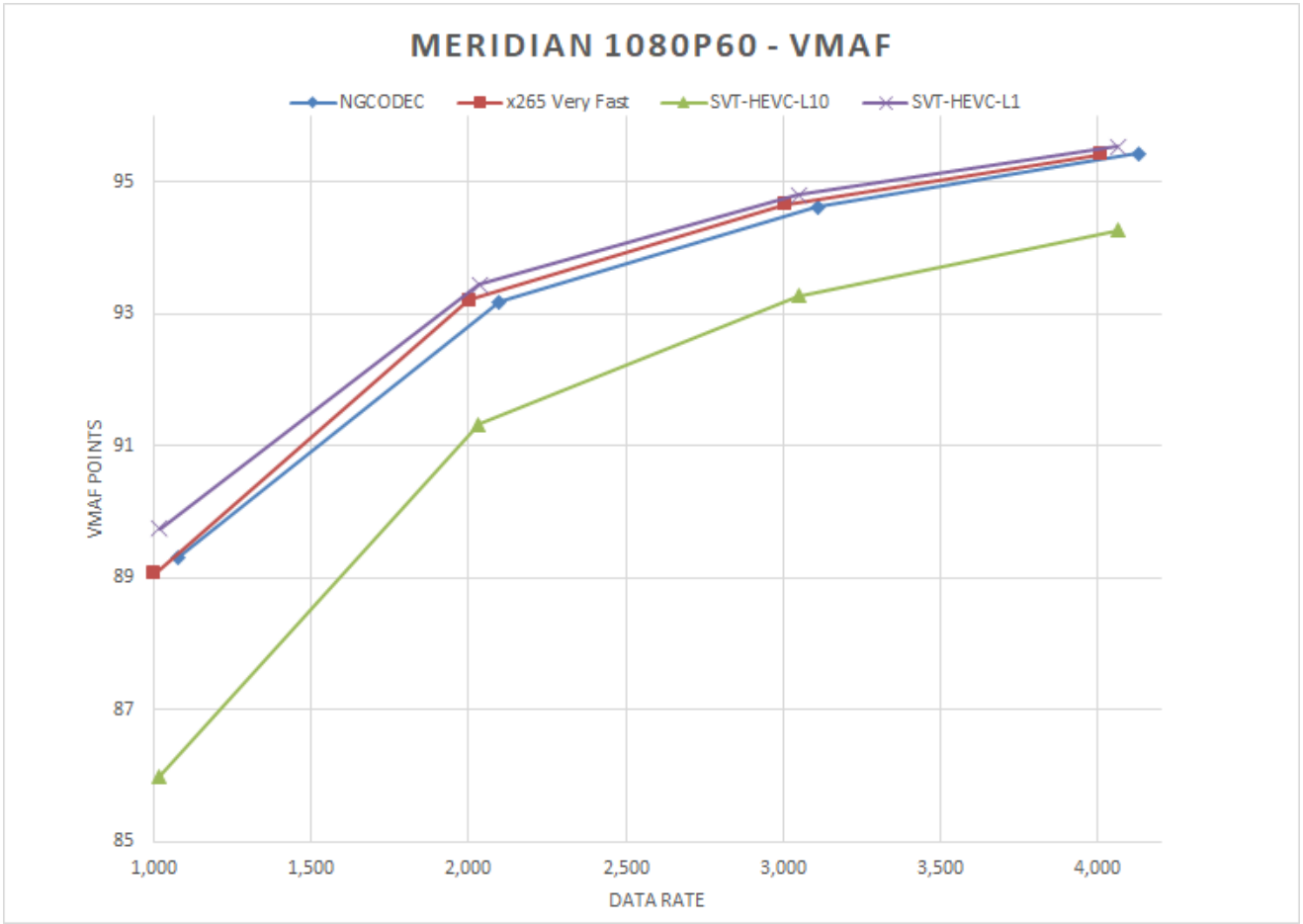
HEVC - GTAV - Rate Distortion Curve



HEVC - GTAV - BD-Rate Computations

VMAF	NGCODEC	x265 Very Fast	SVT-HEVC-L10	SVT-HEVC-L1
NGCODEC	X	-6.08	-24.64	16.46
x265 Very Fast	6.48	X	-19.32	24.37
SVT-HEVC-L10	32.69	23.95	X	57.10
SVT-HEVC-L1	-14.13	-19.60	-36.35	X

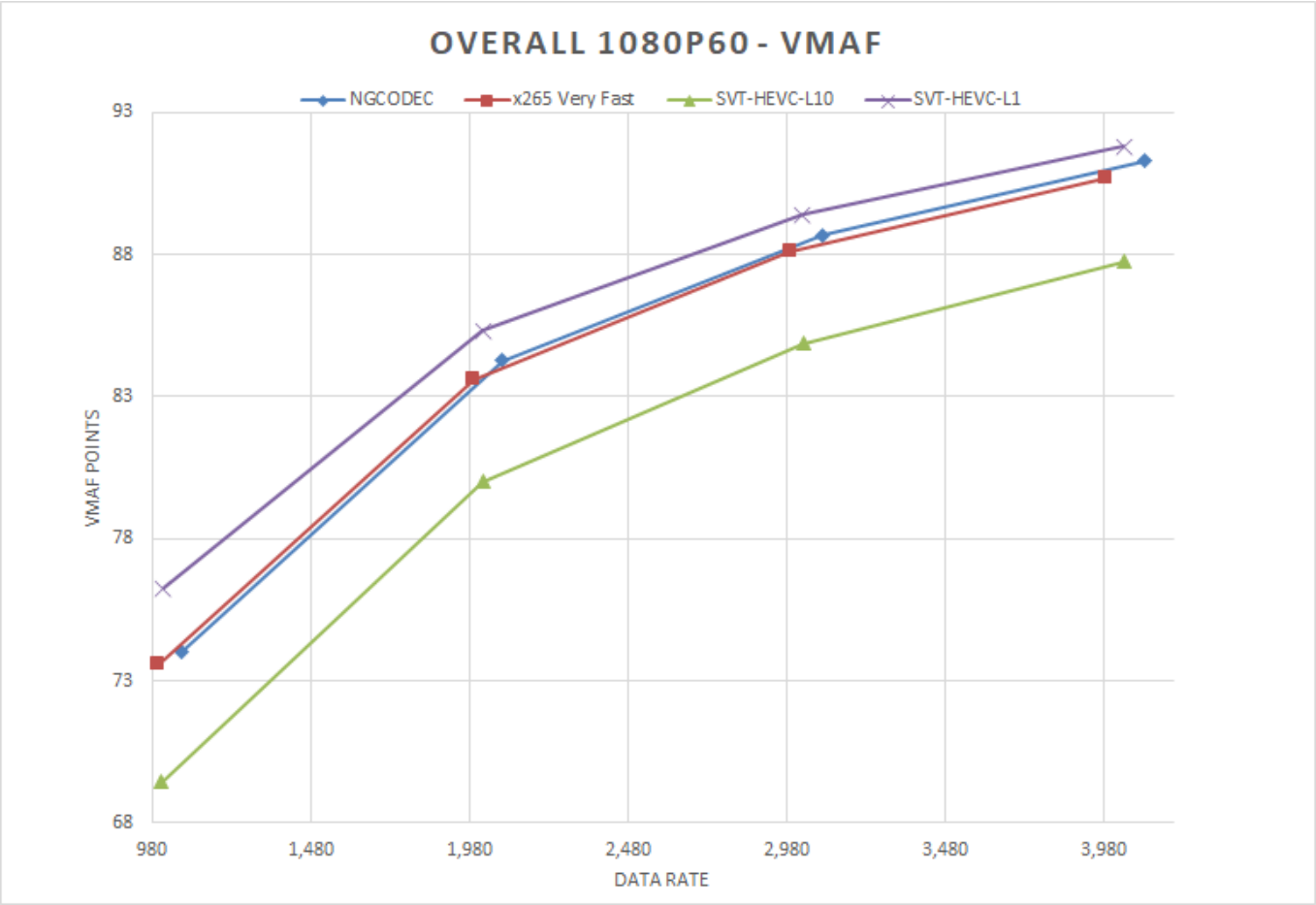
HEVC - Meridian - Rate Distortion Curve



HEVC - Meridian - BD Rate

VMAF	NGCODEC	x265 Very Fast	SVT-HEVC-L10	SVT-HEVC-L1
NGCODEC	X	3.83	-29.14	9.79
x265 Very Fast	-3.69	X	-31.73	5.67
L10	41.13	46.47	X	55.64
SVT-HEVC-L1	-8.91	-5.37	-35.75	X

HEVC - Overall - Rate Distortion Curve (less Dinner Scene)



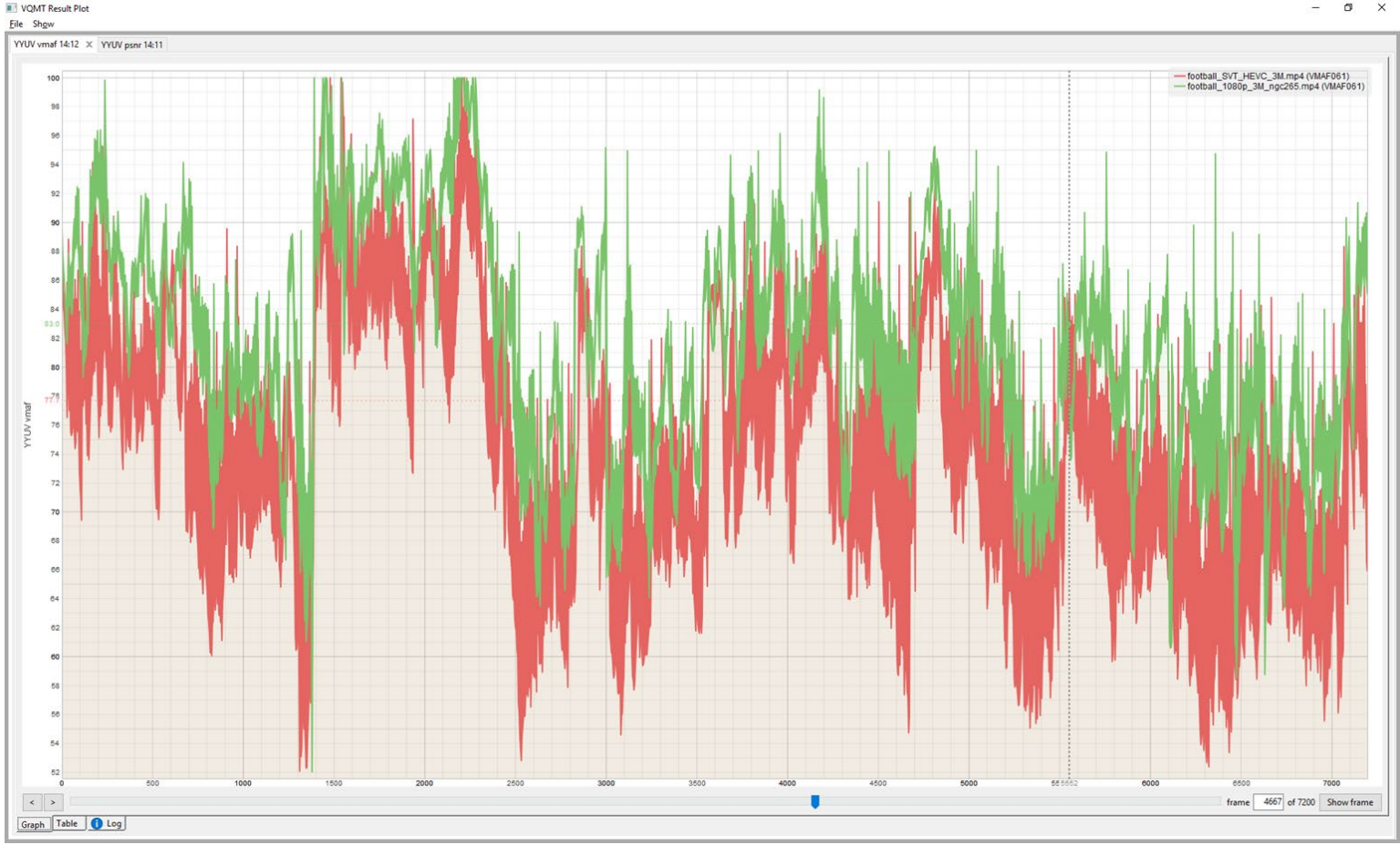
HEVC - Overall - BD Rate

VMAF	NGCODEC	x265 Very Fast	SVT-HEVC-L10	SVT-HEVC-L1
NGCODEC	X	0.05	-24.87	13.41
x265 Very Fast	-0.05	X	-25.33	14.15
L10	33.11	33.92	X	54.04
SVT-HEVC-L1	-11.83	-12.39	-35.08	X

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 isnew)
    - 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