

Towards Live 8K Video Transmissions on Commodity HW

Miloš Liška and the UG team
CESNET z.s.p.o., Prague/Brno, Czech Republic

<milos.liska@cesnet.cz>



I2 Global Summit
2017-04-25

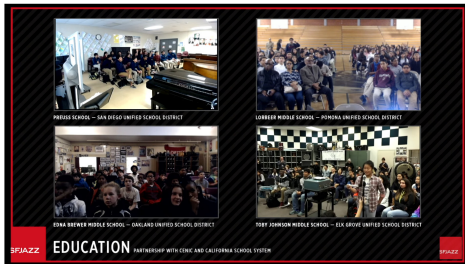
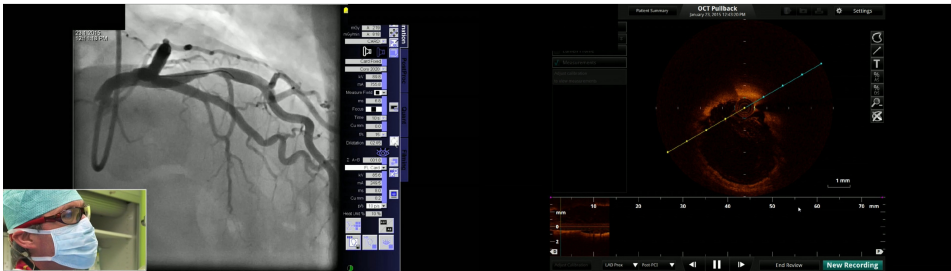


UltraGrid Platform

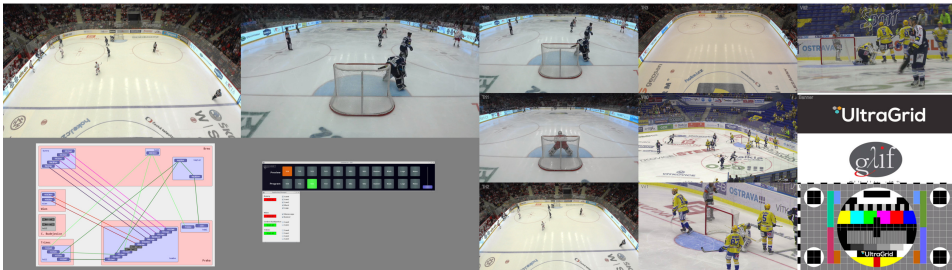
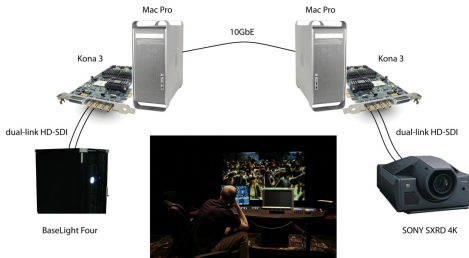
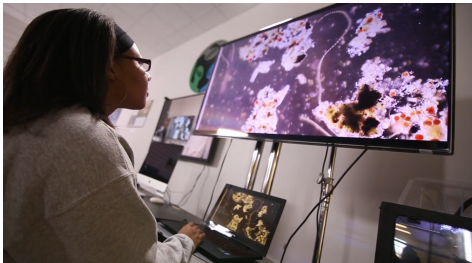
- Technology
 - an affordable platform for very high-quality interactive video (up to 8K) and audio transmissions
 - use of commodity (gaming) hardware
 - ◆ Linux and Windows PC and Mac OS platforms
 - ◆ commodity video capture cards
 - ◆ commodity GPU cards
 - ◆ commodity sound cards
 - ◆ any reasonable network
 - as low latency as possible on commodity hardware
 - open-source software, BSD (GPL) license
- Community, user support



UltraGrid Use Cases



UltraGrid Use Cases

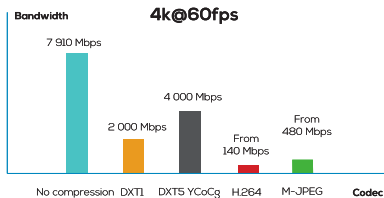
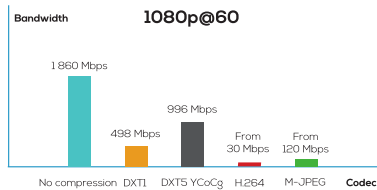
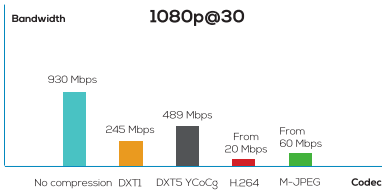


UltraGrid 1.4



- New stable release
- 501 files changed, 59595 insertions(+), 35797 deletions(-)
- Development moved to GitHub <https://github.com/CESNET/UltraGrid>
- AJA and Magewell capture cards support, (GPU accelerated) H.264 and H.265 support, huge GPUJPEG performance improvements (up to 8K 60fps), Syphon support, videoconferencing mode and audio mixer, hugely increased Windows performance, Reed-Solomon FEC scheme for compressed streams, bumps to recent versions of supported drivers, APIs, etc.

Bandwidth Requirements



Compression Performance

- Compression throughputs (Intel[®] Core[™] i7-4960X CPU @ 3.6GHz (6 cores), NVIDIA GeForce GTX 960 GPU (Maxwell))

<i>Compression Type</i>	Highest Achievable Resolution/Framerate
JPEG:90 (GPUJPEG)	4K 120 fps or 8K 60 fps
H.264 (ffmpeg)	4K 30 fps
H.265 (ffmpeg)	4K 24 fps (almost)
H.264 a H.265 (GPU NVENC)	2× 4K 30 fps

Table : Compression throughputs.

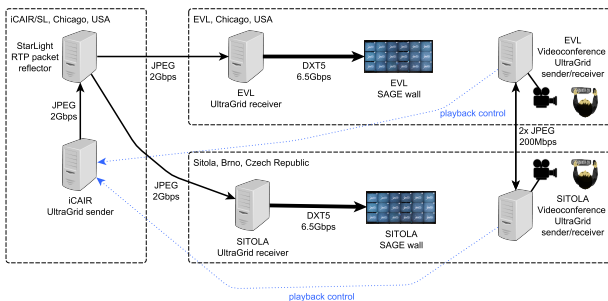
Latency

- As low end-to-end video latency as **40 ms**
 - Uncompressed 60p video
 - Blackmagic DeckLink 4K Extreme, Desktop Video 1.8.2, Linux
- Audio
 - video embedded: synchronous
 - independent source/playback: lowest reliable **15 ms**
 - ◆ low-latency mode `--param low-latency-audio` vs. reliability
 - ◆ can be manually tweaked to better match the video latency with `--audio-delay <delay_ms>`

Live 8K 60fps Video

- 8K 60 fps video = $7680 \times 4320 \times 60 \times 16$ (4:2:2 subsampling with 8b) = **31.85 Gbps**
- The challenge
 - Video acquisition using existing 8K cameras with optical fiber transmitters
 - Commodity video interfaces
 - ◆ 12G-SDI only 12 Gbps
 - ◆ 24G-SDI with 24 Gbps in development
 - The network
 - PC architecture with PCI-E limitations (both performance and physical)

2012 – GPU-JPEG Transatlantic Multi-Point 8K

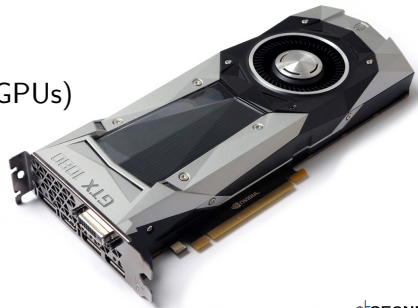


- from pre-rendered sources
- JPEG → DXT5-YCoCg on a single machine



GPU-Accelerated JPEG Compression

- A video compression workhorse in UltraGrid
- Available also as BSD-licensed open-source library:
<https://github.com/CESNET/GPUJPEG>
- Fine-grained parallelization of JPEG (over multiple GPUs)
- With recent optimizations **up to 8K, 80fps**
 - 2x Nvidia Titan X, 12.1 TFLOPS
 - Nvidia M60, up to 9.6 TFLOPS
 - **Nvidia GTX 1080**, 8.9 TFLOPS, MSRP 600 USD



8K Video Capture

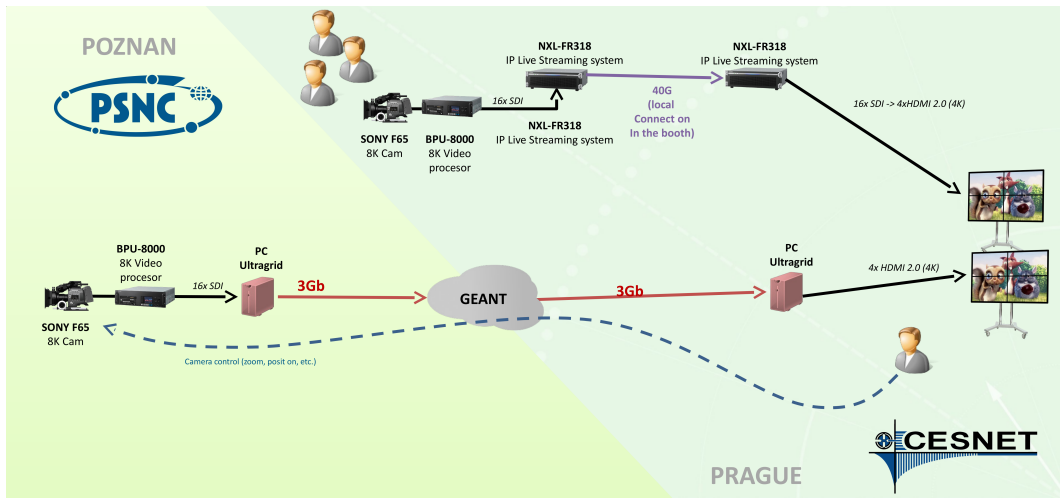
- PSNC 8K Laboratory – **Sony F65 + BPU-8000**
- **2× AJA CorVid88** ≈ 2×2800 USD



- Many thanks to PSNC for being partners in the project.



TNC'16 Demo



TNC'16 Demo



TNC'16 Demo Implementation

- Sender implementation
 - 2× AJA CorVid 88 – 16× 3G-SDI
 - NVidia Tesla M60 accelerator for video compression
 - 10GE NIC
 - Supermicro X10SRA-F MB + 2× Intel E5-1620 CPU + 64 GB RAM
 - 8K video transmission implemented as 4 separate 4K 60 fps streams
- Receiver implementation
 - 10GE NIC
 - 2× NVidia Titan X for video decompression/playback
 - 4 commodity Samsung TVs connected using HDMI 2.0
 - Asus X99-a MB + i7-5960X CPU + 24 GB RAM



Performance Notes

- Jumbo frames!
- Scalable video displaying
 - Frame data stay in the GPU memory (CUDA context) after GPUJPEG decompression and are reused in the OpenGL context
 - CUDA and OpenGL contexts do not run in parallel on NVidia Maxwell architecture (works on Pascal)
- End-to-end transmission latency

$$\underbrace{27.6}_{\text{cam+bps}} + \underbrace{50}_{\text{aja}} + \underbrace{11.4}_{\text{enc}} + \underbrace{2 + 11.6 + 4}_{\text{net}} + \underbrace{6.2}_{\text{dec}} + \underbrace{27.3}_{\text{disp}} = \mathbf{140.1 \text{ ms}}$$

Closing Remarks

- Current sender implementation
 - 2× AJA CorVid 88 – 16× 3G-SDI
 - NVidia GTX 1080
 - 10GE NIC (PCI-E 4× Mellanox ConnectX-3)
 - Asus X99-a MB + i7-5960X CPU + 16 GB RAM
- Current receiver implementation
 - 10GE NIC
 - NVidia GTX 1080 (or NVidia Quadro M6000)
 - 4 commodity TVs connected using DP1.2/HDMI 2.0
 - Asus X99-a MB + i7-5960X CPU + 16 GB RAM
- Astrodesign and Ikegami camera workflows with CCU and 16× 3G-SDI output
- Dell UltraSharp 32 8K Monitor: UP3218K, MSRP 5000 USD
- **Let's talk about the applications!**

Thank you for your attention!

<milos.liska@cesnet.cz>
<ultragrid-dev@cesnet.cz>

<http://www.ultragrid.cz/>

This work is supported by the CESNET E-Infrastructure (LM2015042) project.