High Performance GPU Video Encoding
Agenda

- Why GPU Video Encoding
- NVIDIA H.264 Video Encoding Solutions
- Hardware Architecture
- Software Architecture
- Performance
Why GPU Video Encoding?
Benefits of Encoding on GPU

- Low power
  - Fixed function hardware
  - Reduced memory transfers
- Low latency
- High performance
- Higher density
  - 2x channel density @ ~50% power consumption
- Scalability
Cloud Streaming - Encode on CPU

- App -> Textures & vertices in sys-mem
- Textures & vertices in vid-mem -> Render & Present
- Captured image in vid-mem -> Transfer image to sys-mem
- Encode -> Packetize & transmit

**Large memory transfers**
- Power, Latency

**CPU-intensive tasks**
- Power

**CPU-intensive tasks**
- Cost/seat ↑, channel density ↓
Cloud Streaming - Encode on CPU

- **Large memory transfers**
- **CPU-intensive tasks**
- **Fixed number of CPUs**

- **Power, Latency**
- **Power**
- **Cost/seat ↑, channel density ↓**
- **Limited scalability**
Cloud Streaming - Encode on GPU

- No large memory transfers
- Low-power, low-latency
- Use CPU only where needed
- Low power
- Use CPU only where needed
- Cost/seat ↓, channel density ↑
Cloud Streaming - Encode on GPU

- No large memory transfers
- Low-power, low-latency
- Use CPU only where needed
- Low power
- Use CPU only where needed
- Cost/seat ↓, channel density ↑
- Add GPUs as needed
- Excellent scalability
NVIDIA H.264 Video Encoding Solutions
NVIDIA H.264 Video Encoding Solutions

CUDA
- Hybrid processing (CPU + CUDA)
- ME, intra-prediction, mode decision in CUDA
- VLE on CPU
- Performance scales with CUDA cores
- Works on all GPUs (Tesla, Fermi, Kepler, ...)

NVENC
- Fully hardware accelerated
- ME, intra-prediction, mode decision, VLE
- High performance, low power
- Kepler+ GPUs
NVIDIA H.264 Video Encoding Solutions

- Distributed with CUDA SDK libraries
- No low-latency streaming
- All Platforms - GeForce, Quadro, Tesla, GRID
- Windows only

- Proprietary software API
- Optimized for low-latency streaming
- Better visual quality
- Quadro, GRID and Tesla
- Windows & Linux
Power vs. Performance

- NVENC (HP)
- CUDA (GK104)
- CUDA (GF110)
- NVENC (HQ)
- CUDA (GK107)
- CUDA (GF104)
## NVENC Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>What it enables</th>
</tr>
</thead>
<tbody>
<tr>
<td>H.264 base, main, high profiles</td>
<td>Wide range of use-cases</td>
</tr>
<tr>
<td>Up to 8x HD encode (1080p @ 240 fps)</td>
<td>Faster than real-time encoding</td>
</tr>
<tr>
<td>Flexible ME, QP maps</td>
<td>Customizable quality, region of interest encoding</td>
</tr>
<tr>
<td>YUV 4:2:0 and planar 4:4:4 support</td>
<td>High quality encoding without chroma subsampling</td>
</tr>
<tr>
<td>MVC</td>
<td>Full resolution stereo encode</td>
</tr>
<tr>
<td>Up to 4096 x 4096 in HW</td>
<td>High resolution encode</td>
</tr>
<tr>
<td>API</td>
<td>NVENC SDK (Flexible API, Win/Linux, x86)</td>
</tr>
<tr>
<td></td>
<td>GRID SDK (Capture+ Encode, Win - now, Linux - future)</td>
</tr>
<tr>
<td>NVENC and CUDA parallelism</td>
<td>Simultaneous and parallel HW and CUDA encoding for increased performance</td>
</tr>
</tbody>
</table>
NVENC Hardware Architecture
NVENC Arch: Microcontroller

- NVIDIA proprietary microcontroller
- Runs firmware
- Programs encoder blocks
- Rate control

Diagram details:
- Host
- Microcontroller
- DMA Controller
- Motion Estimation
- Mode Decision
- Intra search & recon loop
- Entropy Coding
- Video memory (FB) interface
- Memory
NVENC Arch: Motion Estimation

- Exhaustive full-pel search (L0, L1, Bi)
  - Temporal
  - Spatial
  - Coloc
  - Constant
  - External

- Half-pel and quarter-pel refinement

- Motion compensation

- Motion Estimation
- Mode Decision
- Intra search & recon loop
- Entropy Coding

- Video memory (FB) interface

- Host

- Microcontroller

- DMA Controller
NVENC Arch: Mode Decision

- Calculates inter-MB cost
- Compares to intra-MB cost and decides final winner
NVENC Arch: Microcontroller

- H.264 intra search
- Forward DCT & quantization
- Recon-loop (IDCT, IQT, deblocking)

Data flow:
- DMA Controller
- Motion Estimation
- Mode Decision
- Intra search & recon loop
- Entropy Coding

Host

Video memory (FB) interface

Memory
NVENC Arch: Microcontroller

- DMA Controller
- Motion Estimation
- Mode Decision
- Intra search & recon loop
- Entropy Coding

Video memory (FB) interface

Memory

• CAVLC & CABAC
NVENC Software Architecture
Using NVENC

**NVENC SDK**
- Direct Encode
  - No capture
  - Transcoding
  - Archiving
  - Video editing
  - CUDA + encoding
  - D3D, CUDA interop
  - Exhaustive encoder settings

**GRID SDK**
- Capture + Encode
  - Capture + encode
  - Optimized for low-latency apps
  - Limited encoder settings
Direct NVENC Encode (NVENC SDK)

- Client application
  - Initialize, Configure, Encode
  - NVENC API
    - Configure HW
      - CUDA Driver
      - NVENC Driver
      - DirectX Driver
  - HW Encode
- NVENC firmware + hardware
- Encoded bitstream
Capture and Encode (GRID SDK)

1. **Client application**
   - DX/OGL Present
   - GRID SDK
   - Capture
   - YUV
   - DirectX Driver
   - GPU 3D Engine

2. **NVENC Hardware**
   - Encode
   - NVENC Driver

3. **Encoded Bitstream**
NVENC SDK

- Available on NVIDIA developer zone
- .DLL/.so, interface header, documentation, sample apps
- Unified API for Windows and Linux
- Works on x86/x64
- Various presets and API’s for
  - Transcoding
  - Video conferencing
  - Remote graphics (Cloud gaming, remote desktop, capture & stream)
- Supports CBR, VBR rate control
NVENC SDK (Contd.)

- Advanced features
  - Dynamic resolution change
  - Dynamic bitrate change
  - Reference picture invalidation
  - Temporal SVC
  - Intra-refresh
  - Two-pass rate control for constant quality
GRID SDK Encode

- Licensed from NVIDIA
- .DLL/.so, interface header, documentation, sample apps
- Windows (now) and Linux (future)
- Works on x86/x64
- Various presets and API’s for
  - Remote graphics (Cloud gaming, remote desktop, capture & stream)
- Optimized for low latency and high quality
**NVENC Performance**

<table>
<thead>
<tr>
<th>Preset</th>
<th>Video 1080p</th>
<th>Gaming 1080p</th>
<th>Gaming 720p</th>
<th>Gaming 720p</th>
</tr>
</thead>
<tbody>
<tr>
<td>HP</td>
<td>223.21 fps</td>
<td>216.45 fps</td>
<td>483.09 fps</td>
<td>485.44 fps</td>
</tr>
<tr>
<td>HQ (With 1-B frame)</td>
<td>116.69 fps</td>
<td>122.55 fps</td>
<td>263.85 fps</td>
<td>290.70 fps</td>
</tr>
<tr>
<td>HQ (No B-frames)</td>
<td>144.30 fps</td>
<td>130.72 fps</td>
<td>311.53 fps</td>
<td>336.70 fps</td>
</tr>
</tbody>
</table>
Performance \((n\) simultaneous HD encodes)\)

- Encode bit rate = 30 Mbps
- Performance shown with more than 1 context is the sum of fps obtained from each context
Gaming Sequence (720p) - 5 Mbps
Gaming Sequence (720p) - 10 Mbps
Questions?