Accelerated Cloud Graphics

Franck DIARD, Ph.D.
Chief SW Architect, NVIDIA
“[GRID] ... truly appears to be prepared to change the way we play games from top to bottom.”

“... the GRID could be the long-awaited answer to cloud gaming.”

Convenience wins. Putting movies in the cloud took sales revenue from $50B (packaged DVDs) to more than $250B (on demand). Cloud gaming is next.
Scope

- System Architectures
  - Kinds of GPU “virtualization”
- HW: potential and limits
- SW: SDK, techniques
- Streaming
- New content / Apps
GRID Server Market

- All cloud gaming platforms from our partners
  - G-cluster, Playcast, Ubitus, Agawi, CyberCloud, Cloud Union
- Desktop/App remoting for VCA (Visual Computing Appliance)
- New cloud game engines
- Video GRID servers
  - vSTB: Live TV, PVR, VOD
  - Transcoding farms
GRID Cloud Gaming Ecosystem

- Partner middleware
  - They deliver full solution with content, writing their middleware
  - They buy simple “blank” gaming servers
  - They use our Monterey SDK to write their SW stack
  - They fit our HW with their goals, QOS and content with our SDK
Comparison

- **GeForce GRID**
  - Game pipeline: 50
  - Capture/Encode: 10
  - Network: 30
  - Decode: 5
  - Display: 66

- **Cloud Gen I**
  - Game pipeline: 100
  - Capture/Encode: 30
  - Network: 75
  - Decode: 15
  - Display: 66

- **Console + TV**
  - Game pipeline: 100
  - Capture/Encode: 0
  - Network: 66

- **GeForce PC**
  - Game pipeline: 50
  - Capture/Encode: 15
Cloud Gaming

CLIENT
- Render
- Decode

CLIENT
- Decode

SERVER
- Render
- Encode
- Capture

Network
- 30 ms
- 2 Frames

GRID Client
- <16 ms

GRID
- 30 ms
- 2 Frames
GRID Gaming

- What is Cloud gaming?
  - AAA games? Casual games?
  - What is the experience?
  - What is the QOS?

- That is a lot of things, the spectrum is VERY large
- What is the right architecture to do all of it?

- Well, it depends...
Some Background on Platforms
SW Architecture

Monterey SDK

NVENC
Low latency Encoder

Low Latency
Frame Buffer Capture
NvFBC

Low Latency
Render Target Capture
NvIFR

NVMOS
Platform Virtualization
Dedicated GPU

Windows 7+
Game

Windows 7+
Game

Ad-hoc API
Shimming
DirectX

Game

Game

Game

Game

GPU

GPU

GPU

GPU
Architectures: Bare Metal OS
No virtualization per se

Dx/OGL API shimming can be used
Multiple DX/OpenGL Contexts
Bare Metal OS/Shimming

Advantages
- One OS, multiple game instances
- Load balancing is dynamic depending on content
- Licencing cost (one OS licence)
- High CCU capability for casual gaming

Disadvantages
- Require game modification for input, frame output encoding
- Today limited by CPU encoding on previous middleware partners solution
The platform is virtualized, not the GPUs
XEN / Hypervisor for Gaming

- **Advantages**
  - Complete separate OSes, per direct attach GPU
  - Security
  - Deployment
  - Basically, a real desktop
    - complete high performance of DirectX, OpenGL
    - Each game can go full screen

- **Problems**
  - CPU tax, performance
  - Needs more resources
  - Not well suited for high CCUs
  - License costs
GRID HW
Kepler, first Cloud GPU

- High performance per watt
- Integrated hardware encoder
- Low-latency frame buffer reads
- 28nm
GRID Boards

- **K340**
  - 4 GK107 (GT650)
  - Only 8x Gen3 between PLX and GPU
  - 1GB per GPU
  - Low 3D perf, but more encoders

- **K520 (0.8 GTX680)**
  - 2 GK104
  - 4GB per GPU
  - Lot more 3D perf, but less encoding

- For Dx gaming, GeForce driver is king
  - GRID GPUs need best perf
  - Profiles, optimizations
Systems
HW: Systems

Requirements
- Pack many GPUs
- 2-4 slots
- VT-D Intel for direct attach/pass-through

HW requirements
- Cooling
- Rack space
- Power

SuperMicro
GRID Gaming Systems:

- SuperMicro 2027 2U
  - 3 or 4 GPU boards
  - 2 Xeons E5 Sandy Bridge
  - Full lanes of PCI 3.0
  - Supports VT-d well
  - 6-8 GK104
  - 12-16 GK107

- Some other form factors
  - VCA (TYAN, 4U)

- Balance of power, rack space, cooling
1x Chipset 2P, 16 cores
Monterey SDK: Server Side

Capture and Stream
Monterey SDK

Core components
- NvFBC: whole display grab: 1 OS = 1 user = 1 stream
- NvIFR: render buffer grab: 1 OS = n users = n streams
- Doc
- Header files, 2 dlls

To system memory
- CSC conversion
- Partners running CPU encode
- Bus contention

To H264 HW encoder
- CSC
NvFBC / NvIFR

- **Frame Buffer Capture (NvFBC)**
  - Simple onboarding, low CCU
  - A display remote display
  - 1 OS = 1 user = 1 GPU = 1 virtual display = 1 stream
  - Orthogonal to the rest of everything that puts pixels on the screen
  - Supports multiple heads

- **Inband Frame Readback (NvIFR)**
  - Complex onboarding, high CCU for small games
  - Injected into apps API calls or modified apps
  - 1 OS = n users = n apps = n streams
  - Can use several GPUs mapped into OS with Dx/OpenGL affinity
NvFBC

- Overlay, mouse, GDI, Dx, OGL, video
- 0 CPU cycles

Compressor process comes async to grab content

Negative latency:
- NvFBC grabs the display back buffer
- If Window Manager runs VSYNC’d, NvFBC grabs bits before they make it to the local screen (if any)
NvFBC Usage case

- VCA, VDI solutions, high-end gaming
- One VM on NMOS has a full OS and will run only one game/desktop per VM
- So it can go full screen and take ownership of the single display attached to VM
- Game’s behavior is not modified, at all
- Only one game can run

- A second background process uses NvFBC to capture what is on the screen, gets back the H264 bits and streams it out
- Complete asynchronous, pure GPU/DMA
NvIFR

- SDK to use with API shimming
- Render target read back: Dx9, Dx10, Dx11 (OGL in testing)
- Format conversion, scaling
- In band with GFX API: Present() call
- Page locked sysmem
- H.264 interface
- Asynchronous Event Signaling, CPU friendly, interrupt driven
  - Not blocking main render loop
- Texture sharing extension for DX9
- Linux Support
NvIFR Usage case

- Game is injected/hooked
- DX device creation is forced offscreen, window hidden
- n smaller games can run at once, generating n streams
### NvENC: Fast pathed from NvIFR/NvFBC

- Completely separate GPU unit: <2 watts
- PSNR comparable to x264
- up to 16 encoding contexts, per GPU
- High Profile
- Single pass or dual pass
  - 720p: 4 ms
  - 1080p: 8 ms
- 4 720p streams per GPU at 30 Hz
- Constrained VBV buffer size
- network packet framing for real time delivery
- CBR, VBR, Min QP
- CUDA Interoperability
- I-frame on-demand
- Reference picture invalidation logic API for packet loss
- 4Kx4K support
- Stereo MVC Encoding
Baremetal Shimming Techniques
Windows DirectX Gaming Limitations...

- If Windows is great...
  - DirectX and OpenGL games run great on our GPUs

- It can’t do more than one game fullscreen at once

- So to run several games on the same baremetal OS

- So games they must be injected/hooked to prevent them from taking ownership of the display and preventing other games from launching/rendering
Shim layer DX affinity

- $N$ GPUs attached to Windows desktop
  - $N$ D3D adapters
- Apps create D3D device on default adapter (DX ordinal 0)
- Apps have to be injected and hooked so
  - The `CreateDevice()` ordinal parameter is overridden to DX ordinal $n$
- DXVA follows the Dx device
- NvIFR uses the same device so the HW encoder will be on the same GPU
Monterey SDK: Client Side

Decode and Display
Decoders

- Low latency
- Power efficient
- H264 packets, fastest path to display
- Windows
- Android, Surface RT
- Mac
- LGTV
HW decoding with nvCUVID/DX

- Popping off h264 frames from network socket
- Feeding elementary stream to nvCUVID -> DXVA
- YUV comes out in a CUDA buffer
- Run CUDA program to convert YUV to ARGB CUDA buffer
- Map cuda ARGB buffer into ARGB texture
- Draw quad with ARGB texture
Android Decode

- Low latency decode/display
- Bypass of OS traditional stack, this is not streaming
- Java frame work calls jni with native window handle
- Set output of decoder directly into native window

SDK:
- HoneyComb/ICS and up, native JNI
- Decoder: 8ms 720p
- Tear free display
- 720p / 60 FPS
- 1080p / 30 FPS
New Type of Content
Multi-View game engine
Cloud Game Engine

- Cloudgine, EA
- In house prototype
- New type of games
- One single app running, one OS
- Using all GPUs
- Rendering different point of views from n different users
Higher CCUs

- Having one complete VM per game is too heavy to reach 80 CCUs per servers
- Extreme CCUs can run baremetal OS with DX shimming to run loads of casual small games
- Looking at some very specific and very limited applications
Video workloads

- **XBMC Multimedia Center**
  - Mostly HW DXVA decoding
  - Compositing, menus, PIP
  - Reencoding with NvIFR
  - Injected with Detours
  - Dx9Ex Present path hooked
  - NvIFR running in side encoding thread
Virtual Set Top Box (vSTB)

- Windows
- GPU
- GPU
- GPU
- GPU

64 transcoded streams

Storage Content Server

64 streams Multicast

GTC TALKS SERIES
Demos

- GRID server (3 x K340)
- Streaming n sessions of XBMC
  - 7 Mbps per stream
- HW decoding, encoding
GRID Support

- For partners
- SDK with collection of samples, tools, tips and techniques
- Very responsive and knowledgeable team
- Solution engineers
- Release planning
Thanks

franck.diard@nvidia.com
Phil Eisler, Jon Barad, Thomas Ruge