CLOUD GAMING WITH NVIDIA GRID TECHNOLOGIES
Franck DIARD, Ph.D., SW Chief Software Architect | GDC 2014
Introduction
Cloud’ification

< 2013

Music, Movies, Books

Spotify
hulu
NETFLIX

2014+

Games

Agawi
NETZYN
SCALABLE GRAPHICS
Playcast
ubitus
G-cluster Global
GPUs vs. Consoles

Graph showing the comparison between PC GPUs and consoles over time, with key models and milestones such as GeForce 3, Xbox, Xbox 360, PS3, GeForce 8800, PS4, and GeForce Titan. Key highlights include a 2x performance increase in 5 years, an 8-year gap between generations, and a 3x performance increase in 3 years.
Cloud Gaming Advantages

For Publishers

- No Piracy
- Precise Accounting
- Game Updates

For Gamers

- Mobility
- Ubiquity
- Instant Play
Cloud Gaming Content Sources

Native
- Windows DirectX / OpenGL
- Linux OpenGL

Players
- Android Games
  - GeniMotion: 3D OGL
  - VirtualBox + Android x86
  - BlueStacks
  - Flash player

New Engines
- Shiva 3D
GRID SDK Components

Server Side on GRID HW
- NvFBC
  - Low Latency
  - Desktop Capture
- NvENC
  - Low latency
  - Hardware Encoder
- NvIFR
  - Low Latency
  - Render Target Capture

Client Side on NV
- Low Latency Decode on GeForce & Tegra

Sample Code (NvIFR & NvFBC)
GRID Programming Guide
GRID Server Setup Guide
Amazon G2 Getting Started Guide
Full VM Environment

One stream per VM, one VM per GPU

XEN

Windows

GPU

VM

VM

Full VM Environment

One stream per VM, one VM per GPU
Increasing CCU with light Sandboxing and Shimming

1 Single instance of Windows

With \( m > n \)
NVIDIA Client

- Decode API is designed for Low latency (1 frame)
- Power efficient using the H.264 hardware decoder
- GRID SDK Examples:
  - Windows PC
    - Based on DX9CUDAGPUDecode
  - NVIDIA Shield (Android)
    - Based on TegraH264HWDecode
- Decoder Client Binary available
  - Handles Audio, Video, and Input
  - For development and testing

Client Side
Low Latency Decode & Display on GeForce & Tegra
## GRID Gaming Products

<table>
<thead>
<tr>
<th>Product Name</th>
<th>GRID K340</th>
<th>GRID K520</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPUs</td>
<td>4 x GK107</td>
<td>2 x GK104</td>
</tr>
<tr>
<td>Total shader (SMs)</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>Core Clocks</td>
<td>900 MHz</td>
<td>800 MHz</td>
</tr>
<tr>
<td>Memory Size</td>
<td>4GB</td>
<td>8GB</td>
</tr>
<tr>
<td>720p30 Encodes</td>
<td>~40</td>
<td>~20</td>
</tr>
<tr>
<td>1080p30 Encodes</td>
<td>~16</td>
<td>~8</td>
</tr>
<tr>
<td>Driver Support</td>
<td>GeForce Drivers and Game Profiles</td>
<td>GeForce Drivers and Game Profiles</td>
</tr>
<tr>
<td>Target Apps</td>
<td>High Density Gaming</td>
<td>High Performance Gaming</td>
</tr>
</tbody>
</table>

Next gen GRID will enable 2x encoding and 2x 3D performance.
Where does it run?

- Dedicated capacity
  - In telco data centers
  - Flood the map, cutting down latency

- Running on top of Amazon Web Services
  - Flexible
  - Less initial investment - rent what you need
  - Big BW from servers to Internet backbone
Building GRID Servers

- **Dual Xeon E5**
  - 2x10 core 2.5 GHz CPUs
  - SuperMicro, ASUS
  - Up to 5 boards = 20 GPUs

- **Xeon E3 systems**
  - 4x 4-core 3.5 GHz CPUs
  - 2 boards = 8 GPUs
  - Cirrascale, CARRI/GIGABYTE
Resources


GRID-devtech-support@nvidia.com

Thanks
NVIDIA GRID Server

- Amazon Instance = NMOS - A VM with a GPU connected through a hypervisor.
- A background process NvFBC, captures screen contents and encodes to H.264.
- GRID SDK Sample: NvFBCH264 captures full-screen graphics and encodes to H.264.
- Full Streaming Server binary available
  - Capture, Encode, and Stream Audio and Video
  - Receives game input with a network connected client
  - Games requires minimal changes.
GAME STREAMING ON AMAZON G2 AND GRID

Eric Young
Overview

- Game Streaming
- Amazon G2 and GRID
- Onboarding Games for GRID
Streaming Games from the Cloud

Client
User Client

Internet
Client Input Controls
Video + Audio Stream

Server
Application or Game
Overview

- Game Streaming
- Amazon G2 and GRID
- Onboarding Games for GRID
Stream on Amazon G2

What is required?
- Middleware based on GRID SDK.
- Server captures frames, encodes to H.264, and streams.
- Rent Amazon G2 instances for $0.78-$0.82 per hour. Spin up/down instances based on traffic demand.

Amazon G2 instance: **g2.2xlarge**
- 1 NVIDIA GRID GK104 (GeForce GTX 670)
- Quad-Core CPUs
- 15GB Memory
- 60GB Storage
GRID Streaming Rapid Prototyping

Challenges

- Game developer want an easy way to prototype a streaming solution without developing a full solution
- Software to handle Gamepad Input, Audio and Video streaming
- Use Amazon G2 rent a GRID hardware/hour and not require purchasing hardware.

NVIDIA Streaming Tools

- Streaming Tool not part of the GRID SDK.
- Useful for rapid prototyping to help onboard games running in cloud
- Testing the game streaming experience running over the cloud
Onboarding Games for GRID

How do I get started?

- Use Private Amazon AMI images with GRID Streaming components.
- Contact NVIDIA (GRID-devtech-support@nvidia.com) for the software components.
- GRID Streaming Prototype is a tool to help Game Developer onboard your games.
GRID Streaming Prototype

GRID Streaming Server
- NvFBC Video Capture
- Audio Capture
- NvENC H.264 Hardware Encoder

Streaming Client
- Decode Audio and Video on GeForce & Tegra
- Keyboard, Mouse, Gamepad Input
Game Streaming Prototype on Amazon G2

Client
- Game Client
  - GRID SDK Decode

Internet
- Client Input Controls
- Video + Audio Stream

G2 GRID Server
- Game Program
  - GRID SDK Capture Encode
  - GRID Driver

Client Input Controls
- Video + Audio Stream
Overview

- Game Streaming
- Amazon G2 and GRID
- Onboarding Games for GRID
Step 1 - Setup AWS Account

- Register for AWS Amazon Account (http://aws.amazon.com)
Step 2 - Create and Launch an Instance

- Choose GPU Instance \textit{(g2.2xlarge)} from EC2 Management Console
- Configure and Launch the Instance

<table>
<thead>
<tr>
<th>All instance types</th>
<th>GPU instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Micro instances</td>
<td></td>
</tr>
<tr>
<td>General purpose</td>
<td>GPU instances provide graphics processing units (GPUs) along with high CPU and network performance for applications benefiting from highly parallelized processing, including 3D graphics, HPC, rendering, and media processing applications.</td>
</tr>
<tr>
<td>Memory optimized</td>
<td>Size: 26 ECU, 8 vCPU, 15 GB memory, 1 x 60 GB Storage Capacity</td>
</tr>
<tr>
<td>Storage optimized</td>
<td>EBS-Optimized Available: Yes, Network Performance: High</td>
</tr>
<tr>
<td>Compute optimized</td>
<td></td>
</tr>
</tbody>
</table>

G2 Instances are backed by 1 x NVIDIA GRID GPU (Kepler GK104) and 8 x hardware hyperthreads from an Intel Xeon E5-2670.
AWS Management Console

Access and manage Amazon Web Services through a simple and intuitive web-based user interface. You can also use the companion AWS Console for iOS and Android to quickly view resources on-the-go.

Features

Access all AWS services in one place

The AWS Management Console is a single destination for managing all your AWS resources, from EC2 instances to DynamoDB tables. Use the Console to perform any number of tasks, from deploying new applications to monitoring the health of your application.

Administer your AWS account

The Console enables you to manage all aspects of your AWS account, including accessing your monthly spending by service, managing security credentials, or even setting up new IAM Users.

Manage AWS resources from any device
Step 3 - Install the Game on the Instance

- Remotely connect to your Instance using VNC
- Copy and install the game to the Instance
- Start up the Streaming Server (waiting for client)
Step 4 - Connect Client for Game Streaming

- Launch the NVIDIA Streaming Client from Local System with Server IP Address to connect
- Launch Game from your Client
- Start Playing!
Onboarding Goals

- Test the Gameplay experience
  - Validate Visual Quality for the end user
  - Test end to end playability

- Additional Guidelines for Game Developers

- Contact (GRID-devtech-support@nvidia.com) if you are interested having your game in the GRID Beta
Samuel Gateau - Devtech Engineer

- Checklist for boarding (your game in the cloud)
- Render, Capture & Encode
- Cloud Game Engine
- Perspectives
Onboard your game in the cloud

- Help us run your game on cloud gaming services
  - GRID (beta) for SHIELD
  - Cloud Gaming Partners
Configure game settings

- Support config files to assign game settings
  - Need to assign the optimal video settings
  - Find the proper balance of CPU / GPU usage for the best quality

- Ideally just rely on the GeForce Experience interoperability API
  - Your game can be configured externally
  - And you benefit the integration with the GeForce Experience program!
Support the GAMEPAD

- Fully support GAMEPAD controllers through XINPUT
  - The only universal controller solution
  - Easily emulated on mobile or desktop
  - Make sure ALL the menus & dialogs are navigable with the gamepad

- Take a look at NvGamepad library
  - A crossplatform gamepad abstraction layer for Windows and Android
Entering credential information is painfull

- Just like for settings, the game should provide a way to get the logging information from command line or a file
- Cloud service can automatically fill the credential for the user making it seamless

We will propose an API to delegate any form filling or dialog box to the client side

- Better integration with the client’s device using the local interface kit
- Just make it actually usable with any client’s platform
- Call for volunteers 😊
Improve the cloudy user experience

- Skip all these amazing pre-game screens & videos
  - Time is money
- Hide any video/audio/performance settings
  - Taken care of by the service
- Follow recommendations for Valve’s “Big Picture”
  - Gamepad !!!
  - Screen size goes from 4” to 4 whatever
Introduce a streaming mode to the game that facilitate the onboarding & support the cloud constraints

- Improve the user experience

Improve the performance profile of the game by capping the rendering framerate to what is needed

- The game logic and simulation can still run at 1000Hz
- Less power required
Checklist for boarding (your game in the cloud)

Render, Capture & Encode

Cloud Game Engine Perspectives
Rendering the video stream

- Framerate of the game is not very stable
- Capturing these frames for video encoding is not good enough
- Streaming needs a fixed rate
Encoding the video stream

- Introduce a new thread in charge of encoding the video stream at the target framerate
- Broadcast to remote client
Rendering Thread

?? Hz

Shared Surface

Encoding Thread

30 Hz

E-1

copy to

R
Rendering Thread

30 Hz

Encoding Thread

E

E-1

Shared Surface

R

R+1

copy to

R+1
Rendering Thread

30 Hz

Encoding Thread

E

R

R+1

R+2

Shared Surface

copy to
Rendering Thread

30 Hz

Encoding Thread

E

E-1

R

R+2

R+1

R+2

E+1

copy from

Shared Surface
Asynchronous capture & encode

- Minimal latency for encoding
- Does not block physics engine/input
- Game loop is running free
- Need 2 extra color surfaces
- Rendering thread could be capped above the targeted framerate to avoid unnecessary rendering
Checklist for boarding (your game in the cloud)

- Render, Capture & Encode

Cloud Game Engine Perspectives
GRID server as a platform
Your next next gen

With Baremetal, a game can take over the full server
- Run several game sessions at the same time
- Graphics engine use all the GPUs and SHARE common resources
- Game engine use all the CPU cores
- Render & stream as many game sessions as possible (think 100)

Partners working on it already
- Square Enix Project “Flare”
One Game Engine

- Same game, many sessions, share all the graphics resource
- If the game is happening live for each session
  - Perform all the animations and update once
  - Compute the "world" physics simulation once
  - All the shadows and lighting can be evaluated and shared
  - Then render each session’s view and encode
Many graphics techniques are tolerant of latency but expensive to calculate. NVIDIA Research created a demo where global illumination was calculated once for a number of clients using an additional cloud based GPU.
Amortized Computation

- Not limited to lighting techniques
- Any state that is expensive to calculate yet shared between clients
  - AI
  - Physics
  - Simulation
  - Animation
Thanks !!

Questions ?

eyoung@nvidia.com  fdiard@nvidia.com  sgateau@nvidia.com