CLOUD GAMING AND NVIDIA GRID™

Agatha Hu, GRID DevTech Engineer
AGENDA

Why Cloud Gaming

Cloud Gaming Architecture

Register and use AWS

Onboard Games Onto GRID

GRID Link SDK
CLOUD GPUs FOR 6 MARKETS

- Cloud Gaming (GRID)
- Application Streaming (GRID SDK)
- Realtime Encode (NVENC SDK)
- Accelerated Virtual Desktops (vGPU)
- Remote Workstations (Quadro)
- High Performance Computing (TESLA)
PC GPU vs CONSOLES

GPU Flops

- GeForce 3
- Xbox
- Xbox 360
- GeForce 7800
- PS3
- GeForce 8800
- PS4
- Xbox One
- GeForce 980 GTX


5 years 8 years 3x 2x
CLOUD GAMING ADVANTAGES

- Mobility
- Uniform High Quality
- Play Instantly
- No Piracy
- Ease of Updating Games
- Precise Accounting

CLOUD GAMING SERVICE
NVIDIA GRID GAME-STREAMING SERVICE

1080p 60 fps • Play in a Minute • AAA titles
GAME STREAMING ARCHITECTURE

“HALF THE BLINK OF AN EYE”
Game Input Latency

console
- Game Engine: 90 ms
- Encoder: 10 ms
- Network: 30 ms
- Decoder: 10 ms
- Controller: 10 ms
- HDMI TV: 30 ms

grid
- Game Engine: 60 ms
- Encoder: 10 ms
- Network: 30 ms
- Decoder: 10 ms
- Controller: 10 ms
- HDMI TV: 30 ms
GRID API SUPPORT

Supported APIs

- DirectX
- OpenGL
- PhysX (CUDA)
CAPTURE AND ENCODE

Graphic Commands

Host Interface

NVIDIA GRID™ SDK

H.264 Streams

NVENC

NVIFR

NVFBC

DRAM Interface

Render Target

Front Buffer
One Game Stream Per VM Per GPU
One VM Instance Connects to One GPU
## Kepler vs Maxwell

### Key Cloud Metrics

<table>
<thead>
<tr>
<th></th>
<th>Kepler GK104</th>
<th>Maxwell GM204</th>
<th>Speed Up</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GPU Specs</strong></td>
<td>1536 Cores 800 Mhz</td>
<td>2048 Cores 1100 Mhz</td>
<td>2X</td>
</tr>
<tr>
<td><strong>TFLOPS</strong></td>
<td>2.4</td>
<td>4.8</td>
<td></td>
</tr>
<tr>
<td><strong>Texture Fillrate (GT/s)</strong></td>
<td>102</td>
<td>151</td>
<td>1.5X</td>
</tr>
<tr>
<td><strong>Video Memory</strong></td>
<td>4GB</td>
<td>8GB</td>
<td>2X</td>
</tr>
<tr>
<td><strong>1080p @ 30fps H.264 streams</strong></td>
<td>8</td>
<td>32</td>
<td>4x</td>
</tr>
</tbody>
</table>
WHERE TO FIND GRID?

Dedicated Capacity
- Data centers located around the world
- Choose the data center closer to user to cut down on latency

GRID Hardware Available On Amazon Web Services
- Scale on Demand and Flexible to configure
- No initial investment in hardware - Rent what you need
- Lots of Bandwidth from Servers to the Internet Backbone

powered by
amazon web services™
Register and Use AWS

• Register an AWS account (http://aws.amazon.com)
Register and Use AWS

- Choose GPU Instance from EC2 control panel (g2.2xlarge)
- Configure and run the Instance
Prepare for Game Streaming
- NVIDIA GRID™ Service connected to SHIELD Devices
- Work with us to get your Game on GRID
GAMEPAD SUPPORT!

Add GAMEPAD Controller Support
- Crucial for a good gaming experience on GRID™

Use XINPUT API
- Preferred for Universal Controller Support
- Easily Implemented for
  - Desktops (Windows, Linux, OSX)
  - Consoles (Xbox 360/One, PS3/PS4)
  - Mobile (Android & IOS)

Refer to the NvGamepad Library
- API Layer for GamePad support for Windows and Android Games
CONFIGURABLE GAME SETTINGS

• Developer: Game’s settings must be configurable
  – Use game config files that are modified externally (by GRID)
  – Do not encrypt config files
    • Makes onboarding difficult or not possible
    • Optionally have the video settings reside in a separate config file

• NVIDIA: Game Onboarding process for GRID
  – Tested with different game settings
  – Settings are optimized for GRID
    • Visual Quality (HQ) and Frame rate (30fps & 60fps)
    • Streaming settings (H.264)
• Target these Screen Sizes and Resolutions
  – Shield screen sizes: 5” to 8” LCD screens
  – Large screen sizes: 32” to 85” TVs
  – Screen resolutions: 720p and 1080p
  – Scale User Interface Elements and Text accordingly

• Add support for HQ Video Settings
  – High resolution textures = images less blurry and stretched
  – Anti-aliasing (TXAA) + more 3D geometry
    • Better H.264 quality (high PSNR & lower bitrate)
  – More Particle Effects
    • Add more emitters, patterns for larger screen
    • Calibrate colors in your game for TVs
VIDEO SETTINGS ON GRID™

• Follow Multiple Screen Size recommendations
  – Screen Sizes Range from 5-8 Inches to 40-65 Ft + Large Screens
  – User Interface Elements and Text Font Sizes Must be Scaled Accordingly
  – Support 16:9 Aspect Ratio = Landscape Orientation
  – Support Overscan = TVs Loose Some Rendered Space Along the Edges

• Refer to These Guides
OPTIMIZING YOUR GAME FOR GRID™

• Game launch suggestions
  – Games need to be directly launchable (without Game Launcher)
  – Allow pre-game screens & video startups to be skipped by Gamepad buttons

• Some Game Menu options should be hidden
  – Shield X sets the game settings before launch
  – Settings should be hidden from user when running on GRID
  – Hide UI settings for Video, audio, and performance
  – Integrate directly with the GRID Link SDK
HANDLING TEXT INPUT

• Text Input
  – If your game has an in-game On Screen (OSC) Keyboard
  – You are GamePad and ready for NVIDIA GRID

• Alternatively refer to GRID Link SDK
  – Text input through a Dialog?
  – A non-native OSC Keyboard = Bad experience for GamePad users
  – Integrate with GRID Link SDK for a better experience

• ALL the Menus & Dialogs Need to be Navigable by Gamepad
  – UI Components should be properly highlighted and resized for visibility
  – UI Elements should be clearly actionable when Selected
  – Do not use Touch Screen controls
VALIDATE WITH GAMESTREAM

- Test GameStream using a Shield & NVIDIA GPU
  - Windows 7 or 8 PC with NVIDIA GTX 650 or higher
  - Latest NVIDIA GeForce drivers.
  - WiFi Router: 802.11a/g, 802.11n, 802.11ac dual band recommended.
    - 5Ghz wireless recommended
  - A Shield device
VALIDATE WITH GAMESTREAM

• On Your PC, by Windows System
• Open NVIDIA GeForce Experience
• Choose Preference -> GameStream
• Click + to “Add Your Game”

• On Your Shield Launch Shield HUB
• My PC Games -> Settings -> GameStream Ready PCs
• Enter the IP Address of Your PC
• Shield Shows a 4 Digit Code
• Enter on Your PC
NVIDIA GRID™ LINK SDK

• What is it?
  – A C/C++ Library that can be linked into your game to allow integration with the NVIDIA GRID™ Cloud Gaming Platform

• What does it do?
  – Allows us to work with game developers to solve some cloud gaming challenges

• Why do this?
  – Better user experience when playing streaming games
  – Reduced barriers to entry - Higher convergence rates for developers
## NVIDIA GRID™ LINK API MODULES

<table>
<thead>
<tr>
<th>PROGRAM CONTROL</th>
<th>USER DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pause Game</td>
<td>Cloud Save and Restore</td>
</tr>
<tr>
<td>Trigger Game Save</td>
<td>of Saved Games and Preferences</td>
</tr>
<tr>
<td>Exit Game</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GAME SETTINGS</th>
<th>USER INTERFACE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Localization</td>
<td>Native Text Entry</td>
</tr>
<tr>
<td>Disable User Changes</td>
<td></td>
</tr>
<tr>
<td>Optimize Graphics Settings</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PATCH CONTROL</th>
<th>USER ACCOUNT CONTROL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check for Required Patches</td>
<td>Account Federation for Seamless</td>
</tr>
<tr>
<td>Keep Game Patched to Newest Version</td>
<td>Game Login and Sign Up</td>
</tr>
</tbody>
</table>

---

**Localization**

**Disable User Changes**

**Optimize Graphics Settings**

**Cloud Save and Restore of Saved Games and Preferences**

**Native Text Entry**

**Account Federation for Seamless Game Login and Sign Up**

**Check for Required Patches**

**Keep Game Patched to Newest Version**
ARCHITECTURE OVERVIEW

GAME BINARY

GAME CODE

All the fun stuff is here!

NVIDIA GRID™ LINK DLL

- Keeps weight of static library low – most code is here
- Only present in Grid environment
- Establishes communication with Grid Service
- Passes requests between Grid Service and game binary

NVIDIA GRID™ SERVICE

- Handles connection with streaming client
- Streams frames to client
- Manages game session, time entitlements, etc.
- Monitors seat for security breaches

NVIDIA GRID™ LINK API

NVIDIA GRID™ LINK LIB
FRAMEWORK AND STANDARDS

• C/C++ API provided
• Windows only
• Initialized at game startup, shutdown at exit
• ALL methods are safe to call in ALL cases
  – Outside of NVIDIA GRID™ environment API calls are all simply No-ops
• Most API methods return integer results
  – Zero means success, error otherwise
• All developer implemented methods return Success/Fail or not implemented
  – Developer can implement methods incrementally, or not at all if not desired or needed
GENERAL FUNCTIONALITY

- **InitializeGRIDLinkSDK()**
  - Called once at application start

- **ShutdownGRIDLinkSDK()**
  - Called once at application exit

- **bool IsGRIDEnabled()**
  - Returns true if running in NVIDIA GRID™ environment
  - Allows application to make NVIDIA GRID™ streaming behavior modifications if desired
PROGRAM CONTROL

• **RequestApplicationPause()**
  – Sent if a user pages out of the NVIDIA GRID™ client on client machine or disconnects from the NVIDIA GRID™ session
  – Implemented as a pause for single player games, game specific for multi-player

• **RequestApplicationSave()**
  – Sent if a NVIDIA GRID™ session is ending for any reason
  – Gives application a chance to save data before exit - generally as an autosave

• **RequestApplicationExit()**
  – Sent when a NVIDIA GRID™ session ends
  – Gives application a chance to clean up and exit gracefully - task kill otherwise
GAME SETTINGS

• **LockUserOptions(UserOptions uoOptions)**
  – Tells application to lock certain user options
  – Options will be set optimally for the host VM by NVIDIA’s onboarding process

• **SetLocale(const char* pchLanguageCode)**
  – Informs application of user’s language - country preferences
• **IsUpdateRequired**(bool* pbUpdateRequired)
  – Query made by NVIDIA GRID™ service at application start
  – Should return true if application is not usable at current version
USER DATA

• **const Char* GetStorageLocation()**
  – Provides the application with a directory path that save games and user preferences
  – Previous saved files will be retrieved and placed here prior to application start

• **NotifyFilesUpdated()**
  – Notifies the NVIDIA GRID™ Service that the game has completed saving or updating some files
  – Will be done at end of session - this allows on demand updates
USER INTERFACE

- **RequestKeyboardOverlayOpen(GridScreenPosition position)**
  - Pops up a native text input keyboard for easy text entry on any device

- **RequestKeyboardOverlayClose()**
  - Closes native text input keyboard
bool RequestGRIDAccessToken(byte** token)

- Obtains a NVIDIA GRID™ access token that can be used by application backend in order to authenticate a user without additional credential entry or account sign up

- Returns true on success, false otherwise. False is only expected in cases where GRID does not have an account federation agreement with the application developer, or if outside of GRID environment
AUTHENTICATION FLOW

NVIDIA GRID™ BACKEND
- REQUEST ACCESS TOKEN

GAME SEAT VM
- NVIDIA GRID™ SERVICE
  - LAUNCH GAME
  - PROVIDE ACCESS TOKEN

NVIDIA GRID™ IDM SERVICE
- RETURN ACCESS TOKEN
- VALIDATE, RETURN LOGIN TOKEN
- PROVIDE USER DATA
- UPDATE USER DATA

GAME BACKEND
- OAUTH 2.0 PROTOCOL
  - REQUEST USER DATA
  - AUTHENTICATE USING TOKEN
  - EXISTING ACCOUNT, LOGIN!
  - MATCH ACCOUNTS, LOGIN!
  - CREATE ACCOUNT, LOGIN!
TESTING AND DEBUGGING

• NVIDIA GRID™ environment can be simulated and tested through use of a test dll
  – Developer can test all methods locally on their development machine

• Automated Test Application Provided
  – Calls developer implemented methods to test and verify implementation
RESOURCES

• Signup for GameWorks!

• NVGamePad Library

• Contact:
  – GRID-developer-support@nvidia.com

• GRID Link Library:
  – GridLinkSDK-info@exchange.nvidia.com