Phillip Miller
Director of Product Management
NVIDIA Advanced Rendering

Agenda

1. What is NVIDIA Advanced Rendering ?
2. Progress in NVIDIA Iray
3. Progress in NVIDIA OptiX
4. GPU Ray Tracing Basics (if there's time)

NVIDIA Ray Tracing Options

- **CUDA** – language and computing platform
  - The choice for building *entirely custom GPU solutions from scratch*

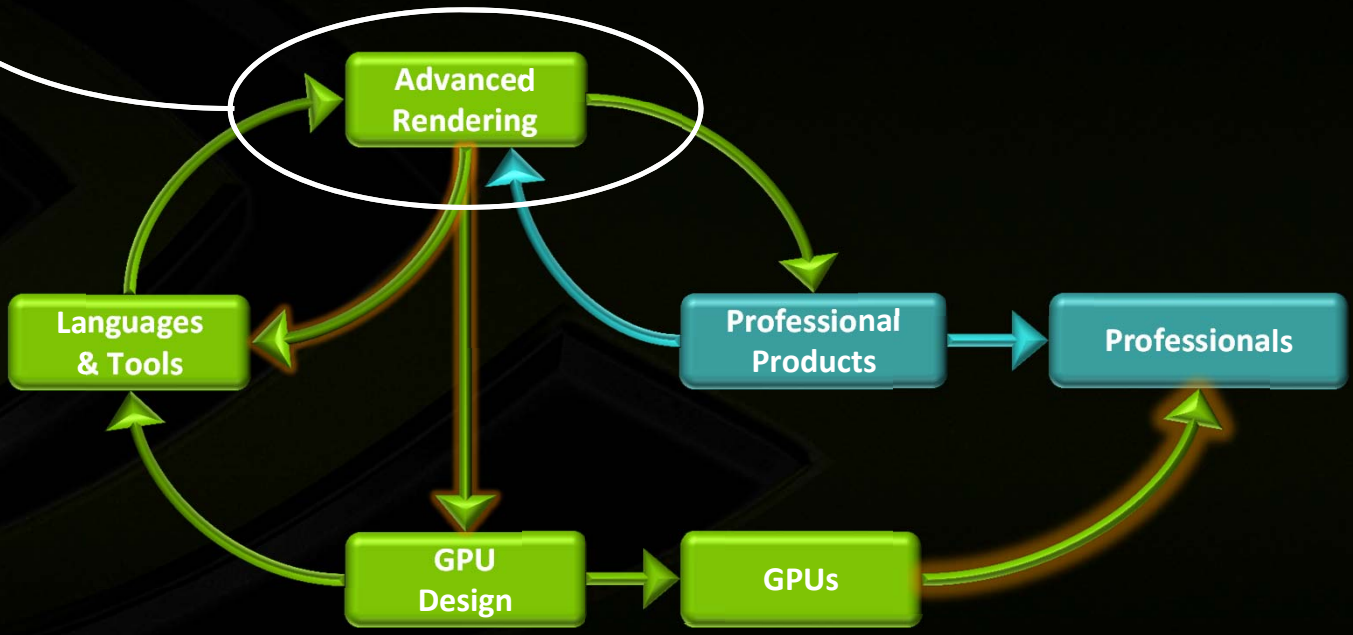
NVIDIA Advanced Rendering:

- **OptiX** – freely licensed middleware for ray tracing developers
  - Good choice for developers *with domain expertise* building *custom ray tracing solutions* who prefer *leaving GPU issues (and ray tracing basics) to NVIDIA*

- **Iray & mental ray** – commercially licensed rendering products
  - Good choice for companies wanting a *ready-to-integrate solution* which is *maintained and advanced for them*
  - Iray focuses on the needs of Design markets, while mental ray focuses on Film Production

NVIDIA Commercial Rendering Offerings

- State-of-the-Art rendering exemplifying what's possible on the latest GPU technology
- Completes a vital Feedback Loop to influence NVIDIA (long before products alone can)



- Result: best of class solutions for end users, licensees and third party developers


NVIDIA Iray 2013

Iray Physically Based Easy to Use

NVIDIA

- Rapid Adoption
- Simplicity:
  - Photorealism is a goal that both artists & developers can relate to
  - Physical model ensures consistent algorithms
  - Results match real world experiences with light, distance and materials
  - Simplicity = ease of use
    = faster production
    = approachable by more users

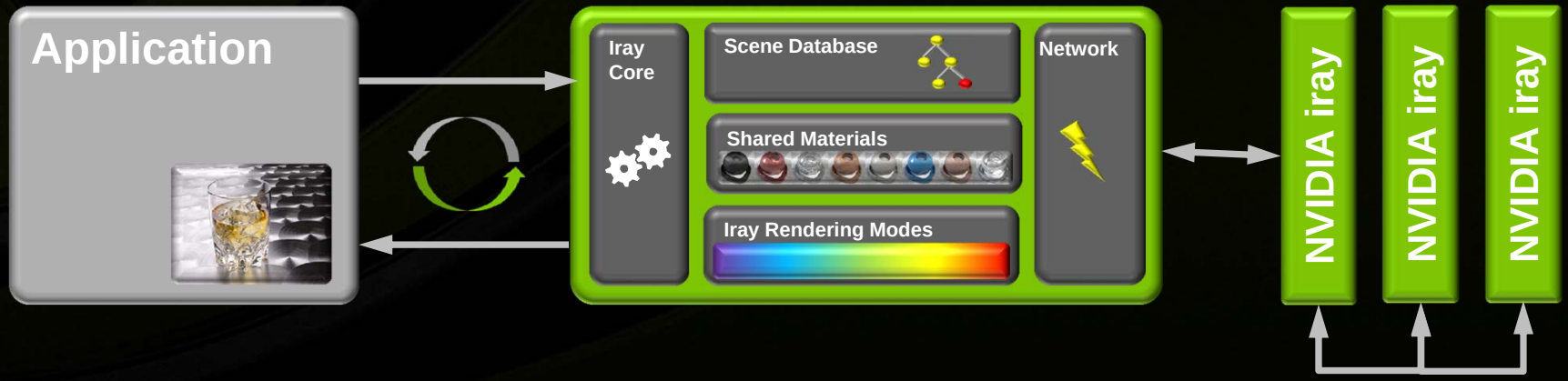
NVIDIA Iray for End Users

- Iray within shipping commercial products:
  - Autodesk **3ds Max** & **3ds Max Design**
  - Dassault Systèmes **Catia V6**
  - Bunkspeed **SHOT**, **MOVE**, **PRO**
  - Cinema 4D **(M4D add-on)**
  - SketchUp **(Bloom Unit add-on)**

- Now let's discuss what's available to these products to include in their future updates from NVIDIA Iray

Iray 2013 just released to commercial licensees

- For Software Developers wanting to add physically based rendering to their applications that is easy to use, highly interactive and scalable
- Now with render modes, shared materials, cluster management, and cloud rendering
- A rich API handshakes with Application manipulation for interactive updates



Iray 2013 – contains Iray 3

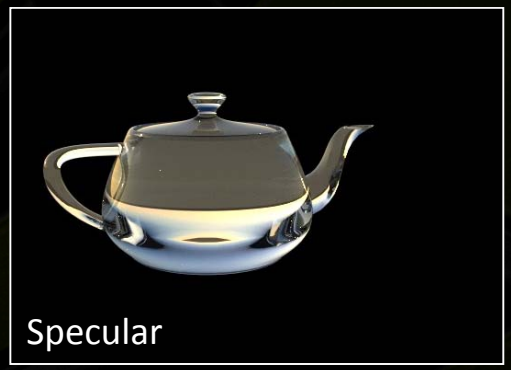
Major focus for this Iray release:

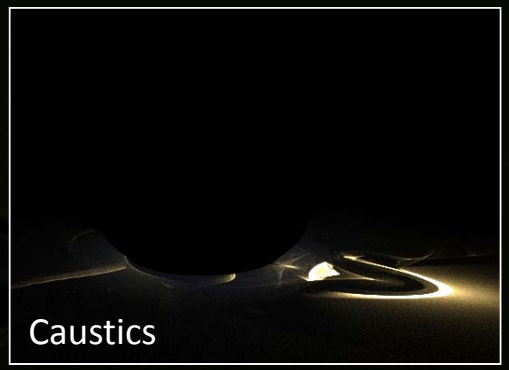
- Greatly expand post production possibilities
- Increase rendering quality for challenging scenarios
- Improve usability
  - Reduce noise and artifacts of convergence
  - Improve data handling to use less memory
  - Improve interactivity via better balancing of Display GPU
  - Support new shared material model

- **All while maintaining speed!**

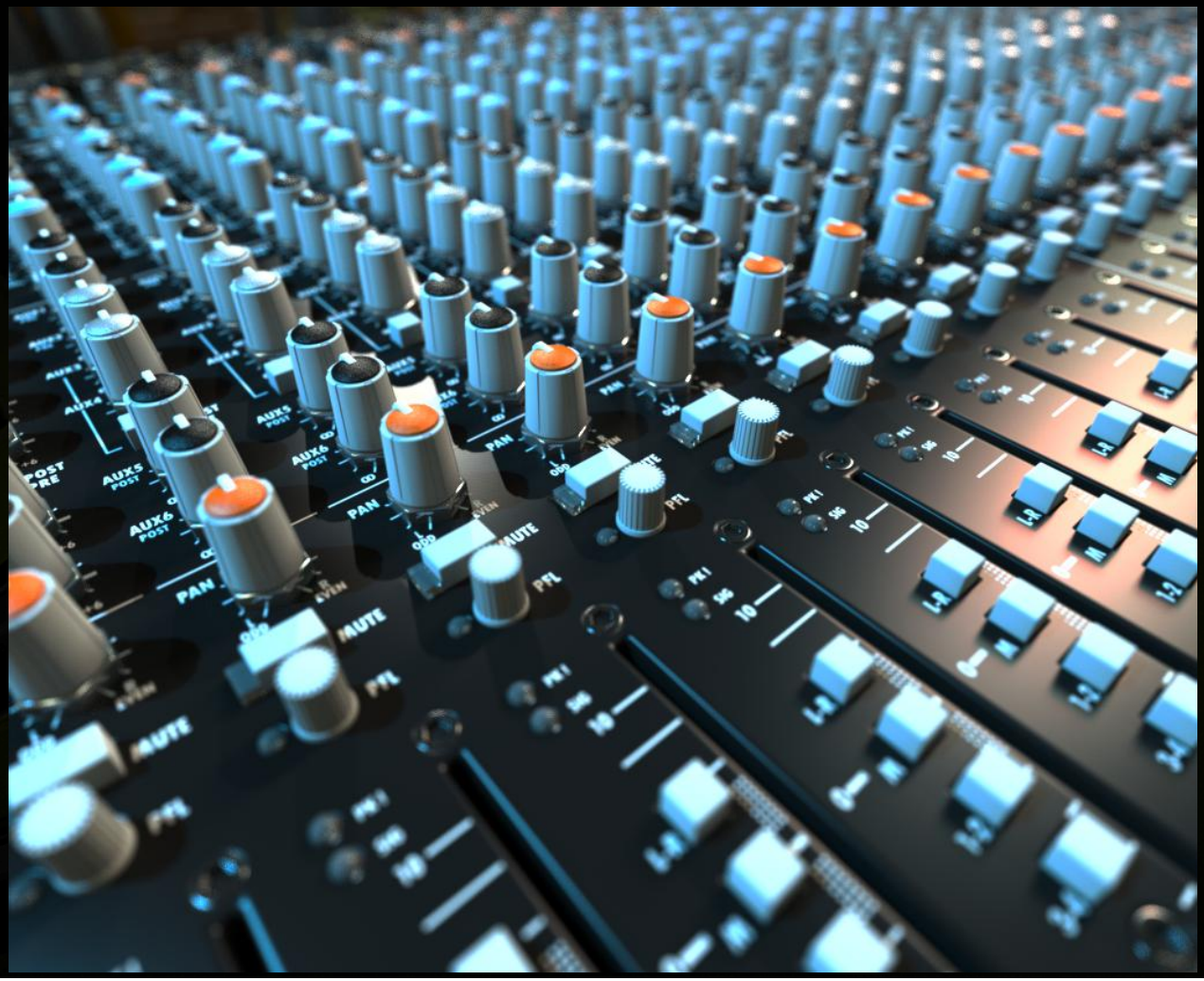
New with Iray 3 - Light Path Expressions

- Similar to traditional “Render Passes” – but on steroids
- Freely Configurable – can be edited by end users
- Can be on a light group or per-light basis

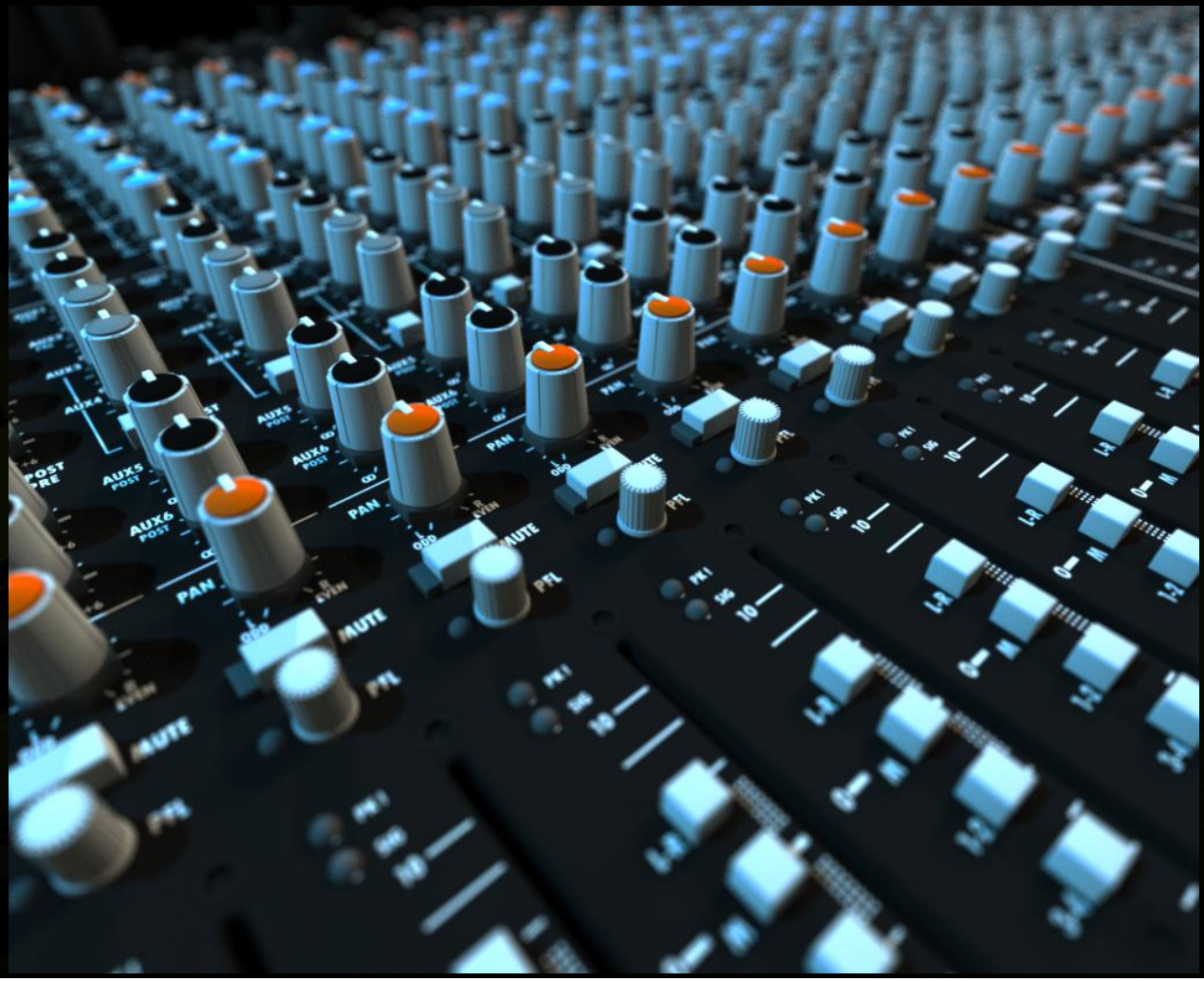




Iray Beauty Pass

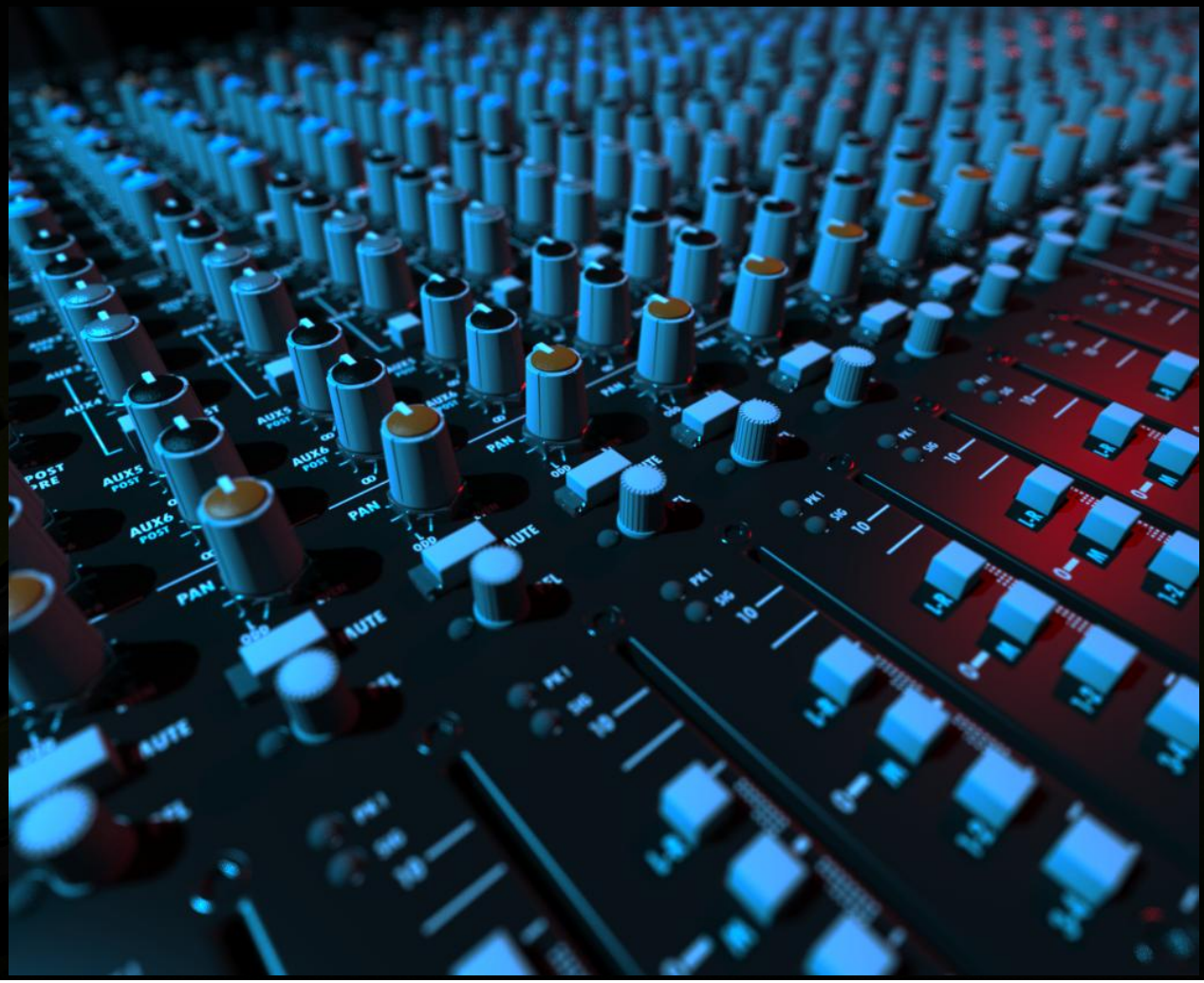


Diffuse LPE



Specular & Glossy LPE

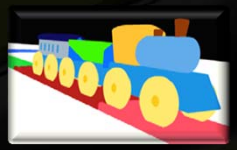
Light1 and Light2 LPE



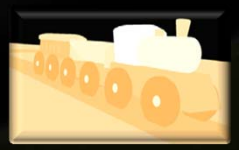
Environment Lighting LPE

Light Path Expressions

- Regular expressions
  - BSDF components
  - Emitting light handles & Interacting geometry handles
  - Considerable flexibility in Post
- Minimal render time overhead (< 5%)
  - All buffers render simultaneously
  - Faster in some cases (e.g. direct illumination only)



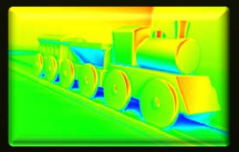
Object ID



Material ID

Z-Depth

Specular



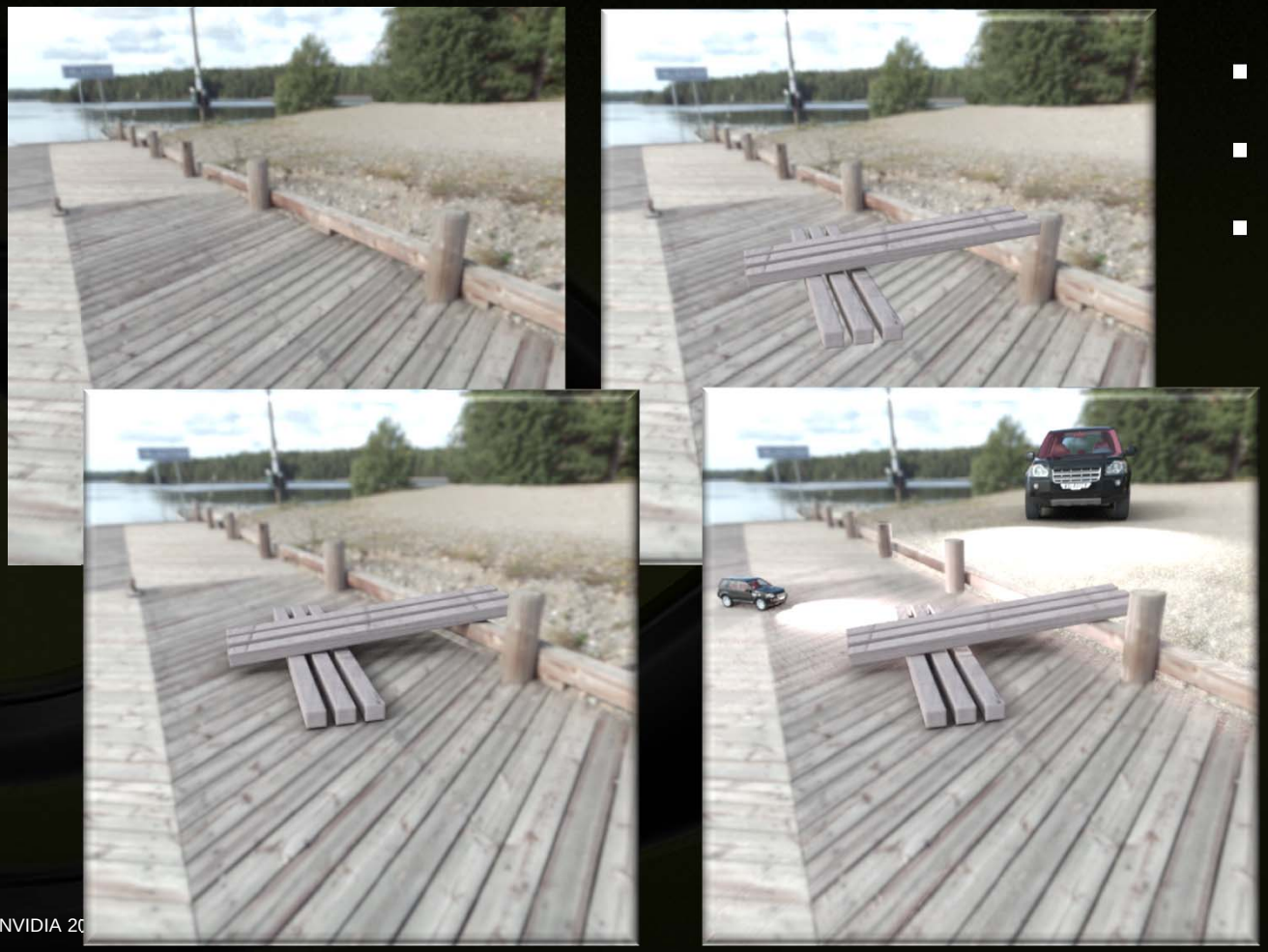
Irradiance

Glossy

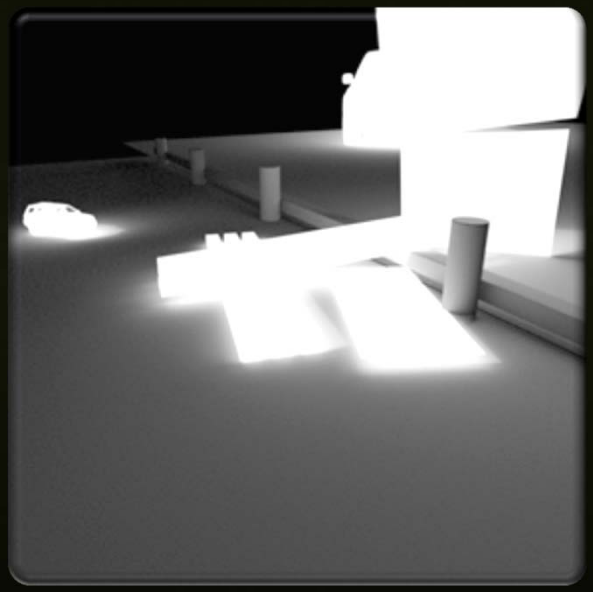
New with Iray 3 – Matte Objects

- Classic workflow + more
- Supports full GI, MBlur, DOF

New with Iray 3 – Matte Objects



- Classic workflow + more
- Supports full GI, MBlur, DOF
- Lighting/Shadow “bloom” essential for realism



New with Iray 3 – Matte Objects

- Back Plate

- Chrome Sphere Reference

New with Iray 3 – Matte Objects

- Stand-In Geometry

- Match Materials & Flag Matte Objects

New with Iray 3 – Matte Objects

- Add synthetic geometry at will

- The Iray matte making it possible

New with Iray 3 – Additional Samplers

- “Caustic”
  for doing just that
- “Architectural”
  a robust path sampler
  for highly indirect
  challenges

Iray 3 with new “caustic” sampler

New with Iray 3 – Additional Samplers

Iray 2.x

Iray 3 with new “caustic” sampler

New with Iray 3 – Additional Samplers

Iray 2.x

Iray 3 with new “architectural” sampler

New with Iray 3 – Additional Samplers

Iray 2.x

Iray 3 with new “architectural” sampler

Iray 3 Improved Convergence

2 minutes

Iray 2.x

Iray 3

Iray 3 Improved Convergence 10 minutes

Iray 2.x

Iray 3

Iray 3 Improved Convergence 30 minutes

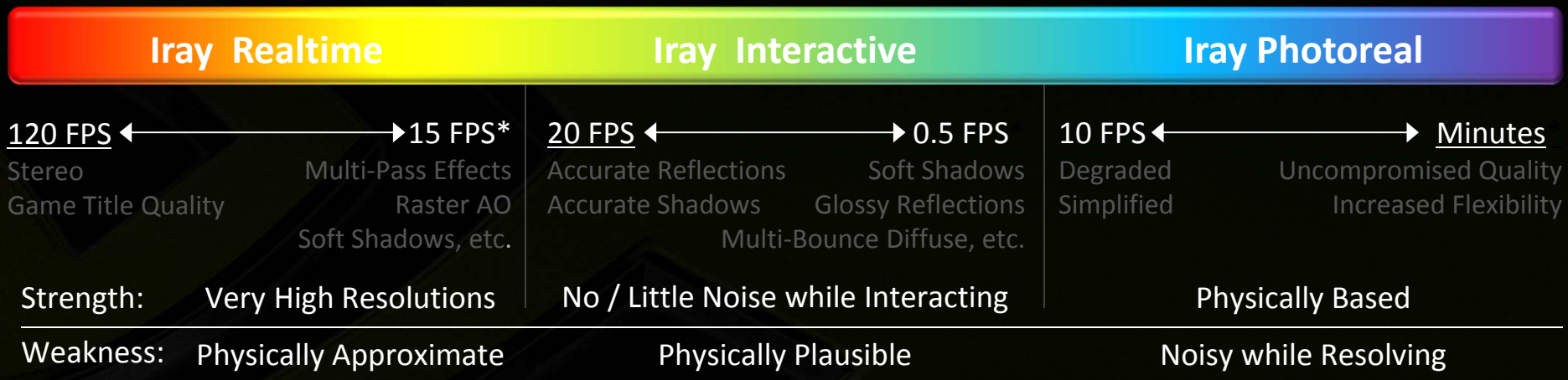
Iray 2.x

Iray 3

Iray 2013 Render Modes

NVIDIA

- Multiple Rendering Modes, providing a quality/speed continuum



- API calls for which mode to use, with what features, what to do on mouse-up, etc. enable custom personalities for behavior and look

Iray 2013 Rendering Modes

- Iray Photoreal
  - Interactive but “noisy”
  - The overall scene resolves in a couple of minutes
- Iray Interactive
  - Interactive with minimal noise
  - Shadows, glossiness and AA resolve in a couple of seconds
  - Can be made faster with lower quality settings

Iray Photoreal in a few minutes

Iray Interactive in a few seconds

Iray Shared Materials

- Physically Based for accuracy (BSDF)
- Layered for great flexibility
- Consistent appearance

Iray Interactive

Iray Photoreal

- Processed via the Iray API
- No plans yet for licensing the language processing separately

MDL – Material Description Language

- NOT a shading language, but a canonical representation which renderers can target as they see fit

For material artists:

- Easy to parse and understand
- No algorithmic knowledge required
- Parameters easily exposed

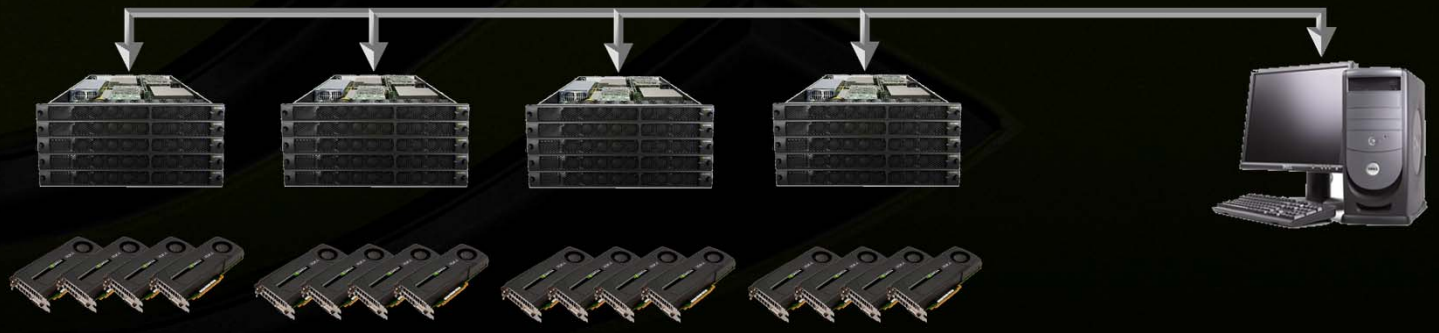
For end users:

- Assign and edit parameters at will

Iray Cluster

- Near linear scaling for production rendering
- Also usable for interactive rendering (on low latency networks)
- Includes a cluster configuration front end

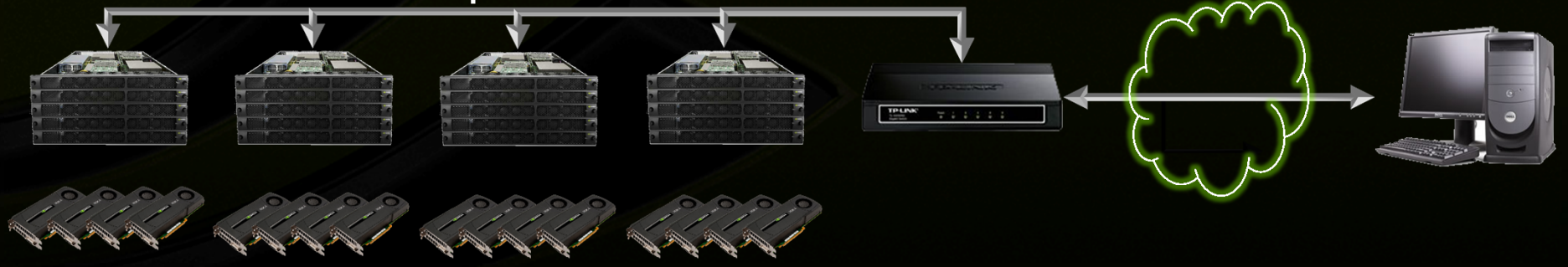
- Additional license required



Iray Cloud

- Scalable rendering power on demand
- Network protocols to handle private and public clouds
- Assets are only ever sent once – for minimal upload times
- Scene edits are handled incrementally – for fast iterations

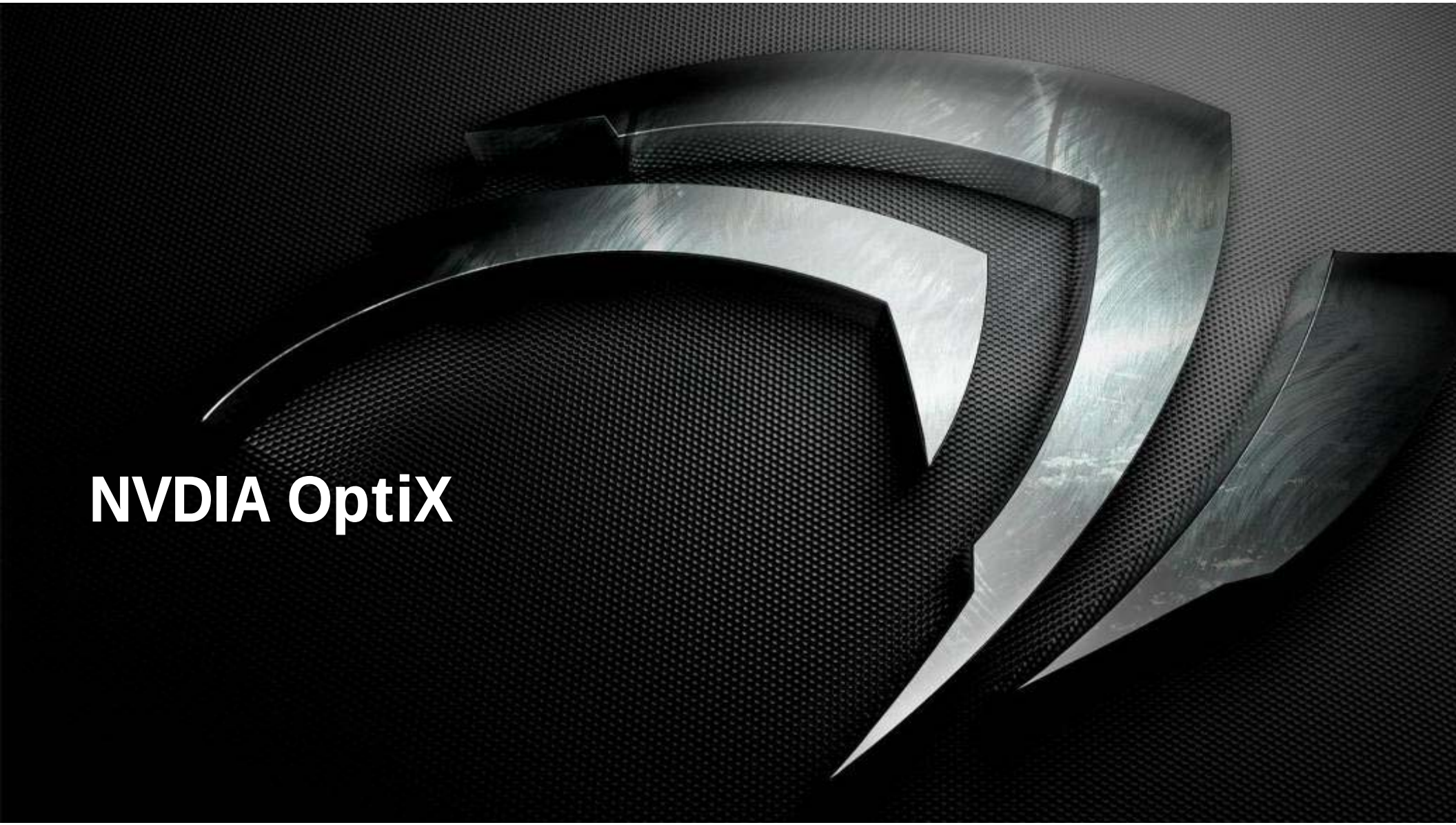
- Additional license required




NVIDIA Iray 2013
Plug it in

Developers wanting to try Iray 2013

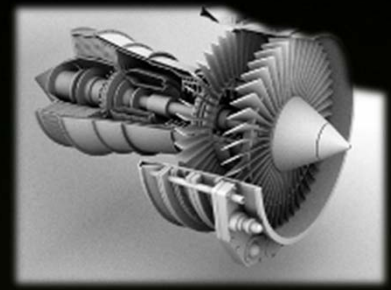
- Procedure:
  1. Register your interest at www.mentalimages.com
     http://www.mentalimages.com/products/iray/iray-integration-framework/software-download.html
  2. NVIDIA reviews application, and grants access to SDK
  3. Integrate the SDK within your Application
     Result is full featured, but output constrained
     Requires a GPU with at least 2GB of memory
  4. Once satisfied, obtain a commercial license from NVIDIA


NVDIA OptiX

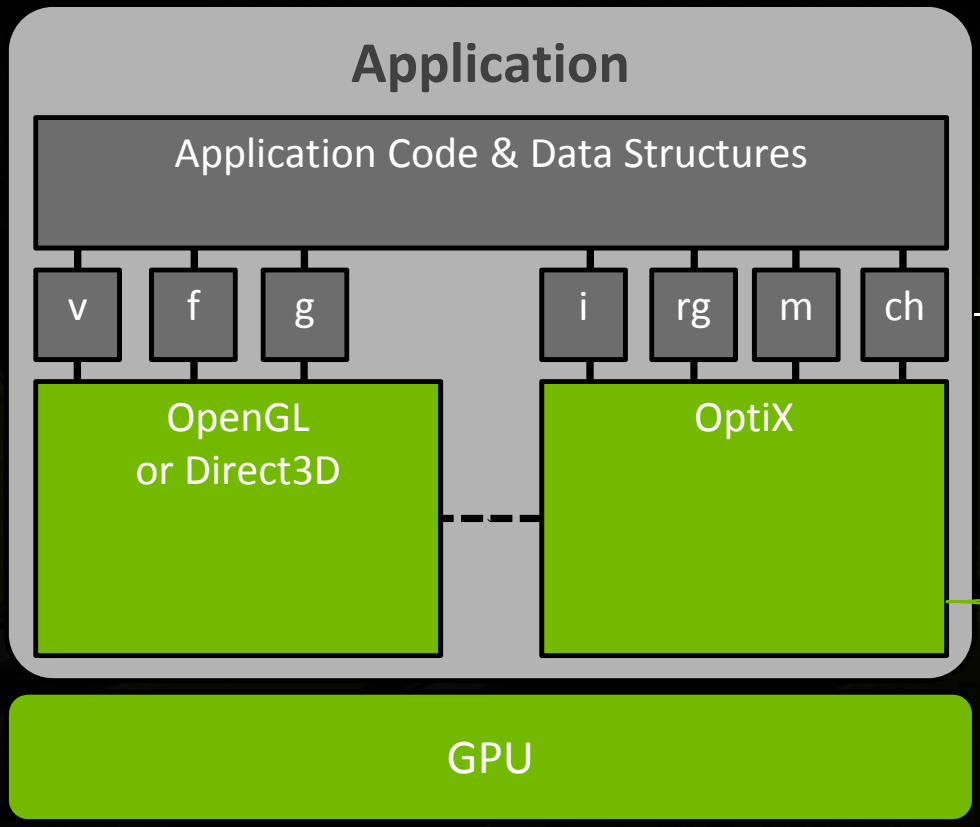
NVIDIA® OptiX™ ray tracing engine

A programmable ray tracing framework enabling the rapid development of high performance ray tracing applications – from complete renderers to discrete functions
(collision, acoustics, ballistics, radiation reflectance, signals, etc.)

- Use your techniques, methods, and data for your application with simple programs and a single ray programming model
- OptiX makes it easy to implement by doing the "heavy lifting" of ray tracing with easy-to-use APIs, for traversal, intersection, and (optionally) shading.
- OptiX makes it run fast on the GPU, by handling load balancing, parallelism, paging, and optimizing per GPU architecture.



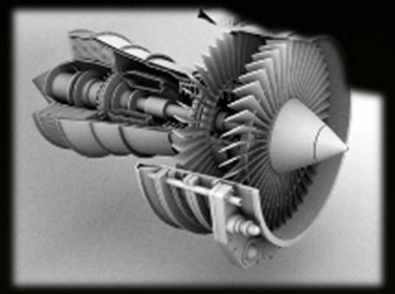
OptiX - similar “in approach” to OpenGL

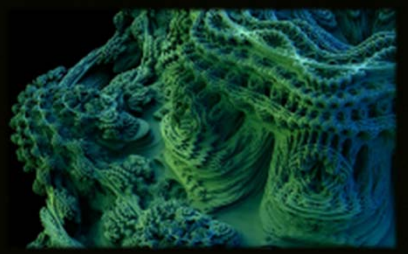


- C-based Shaders/Functions (minimal CUDA exp. reqd.)
  - Small, Custom Programs
  - Acceleration Structures Build & Traversal
- Optimal GPU parallelism and Performance
- Memory Management
- Paging

OptiX Across Markets and Disciplines

As many as 1/3 of OptiX developers don't "render", and a very simple Traversal API makes this even easier





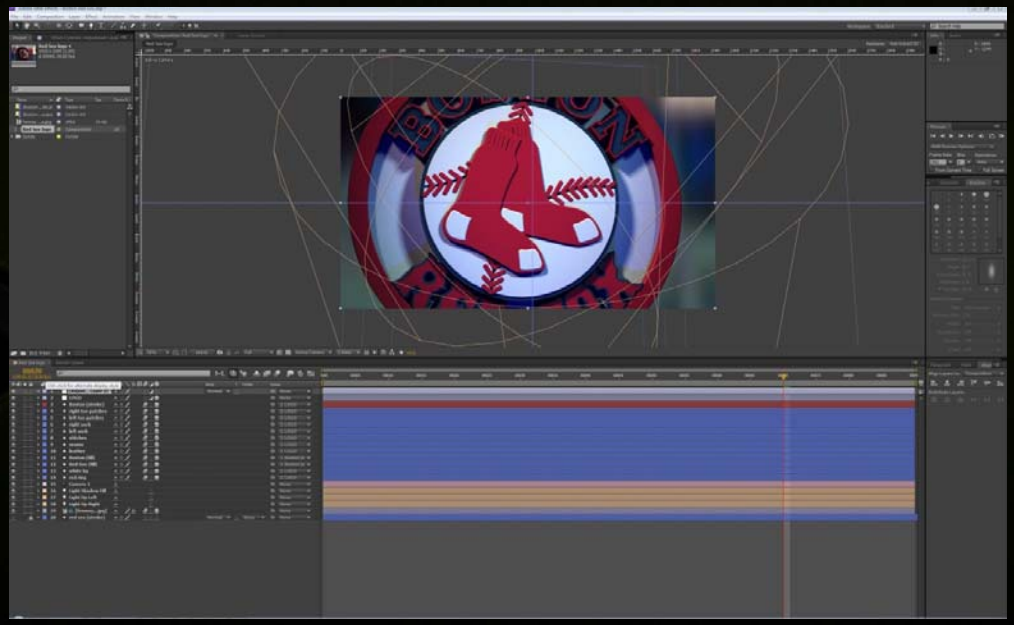
OptiX generality includes:

- No assumptions on technique, shading language, geometry type, or data structure
- Supports custom ray generation, material shading, object intersection, scene traversal, ray payloads
- Programmable intersection for custom surface types (procedurals, patches, NURBS, displacement, hair, fur, etc.)

Adobe After Effects CS6 – using OptiX

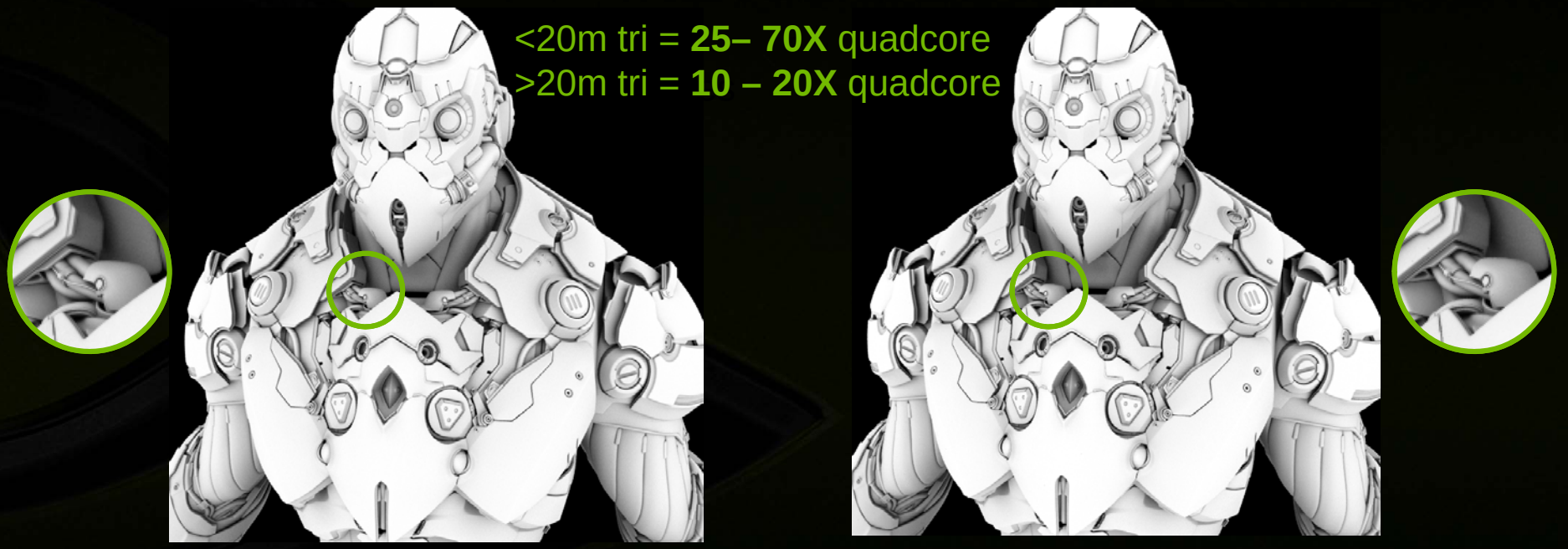
New 3D compositing with ray traced production renderer

- Built from scratch, in 1 release cycle
- 100% OptiX – no x86 code
- Includes CPU Fallback
  - Via LLVM in OptiX
  - Currently unique to Adobe



OptiX for mental ray Ambient Occlusion

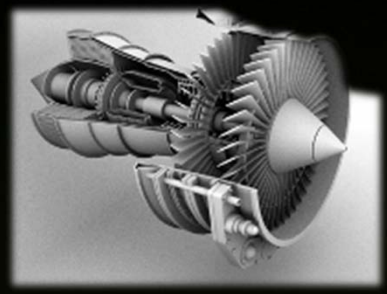
- mental ray 3.11 (released to licensees) pipeline accelerated



- 1.5sec HLBVH build + 15sec on Quadro 6000 vs. 20 minutes on CPU

OptiX 2.6 - this past August

The OptiX 2.5 feature set with Kepler support using CUDA 4.2



- Optimized for NVVM (aka LLVM for CUDA)
  - Note: CUDA 1.0 to 4.0 used Open64 compiler front end

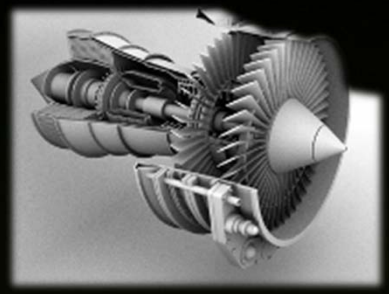
- NVVM code generation is very different, but it's worth it.
- LLVM is a great leap forward for CUDA, allowing any language to work on the CUDA platform and thus on OptiX

- Continued to include Paging for out-of core memory situations

OptiX 3 – Shipping December 12, 2012

Now based on CUDA 5.0, OptiX 3 includes many highly requested features:

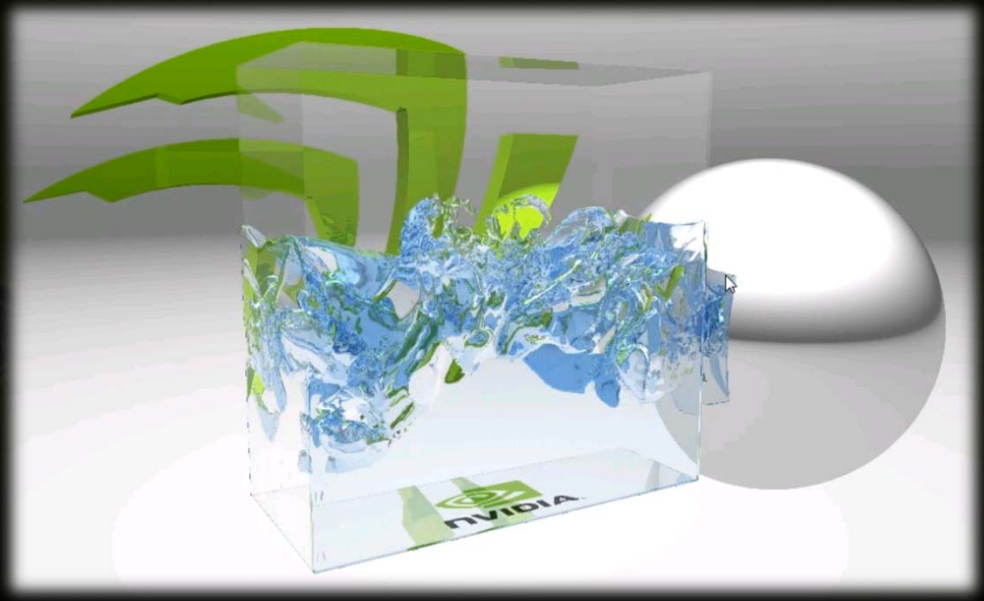
- CUDA Interop – for sharing CUDA contexts and pointers with other CUDA programs
  - See new samples: Collision, Ocean
  - Includes Multi-GPU support
- Callable Programs – for Shade Trees, etc.
- Much faster Acceleration Structure building
  - SBVH is up to 8X faster (for large assemblies) and compiles +2X faster
  - BVH refitting on all AS Builders
- GPU Direct for faster GL interop buffers



OptiX 3 CUDA Interop Water Demo

Using OptiX and PhysX together

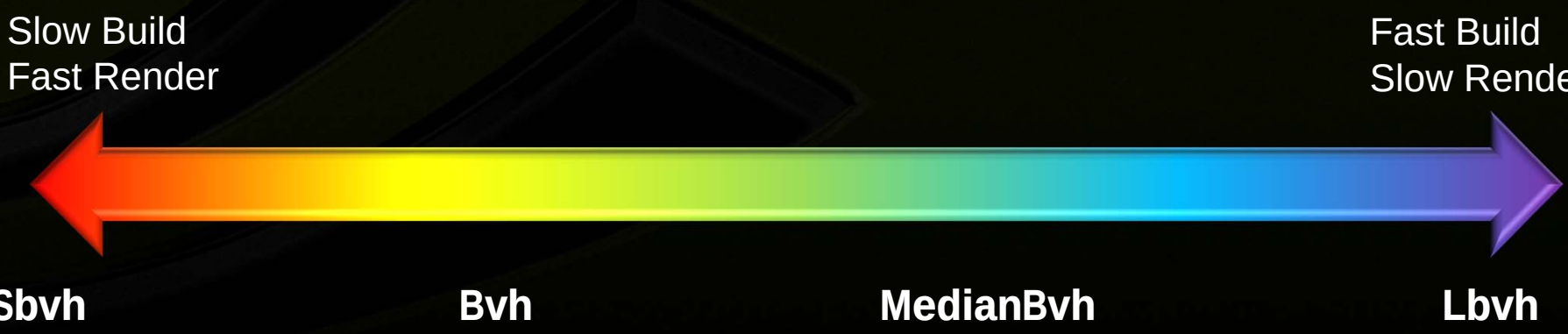
- PhysX CFD water simulation in a 128x128x64 volume
- Custom OptiX intersection object for water
- Fresnel dielectric model for water shading w/ 12 reflection & refraction bounces
- CUDA Interop exchanges data without extra copies
  – in this case across GPUs
- Uses FXAA for anti-aliasing (a fantastic new option for interactive ray tracing)



OptiX 3

BVH Refinement

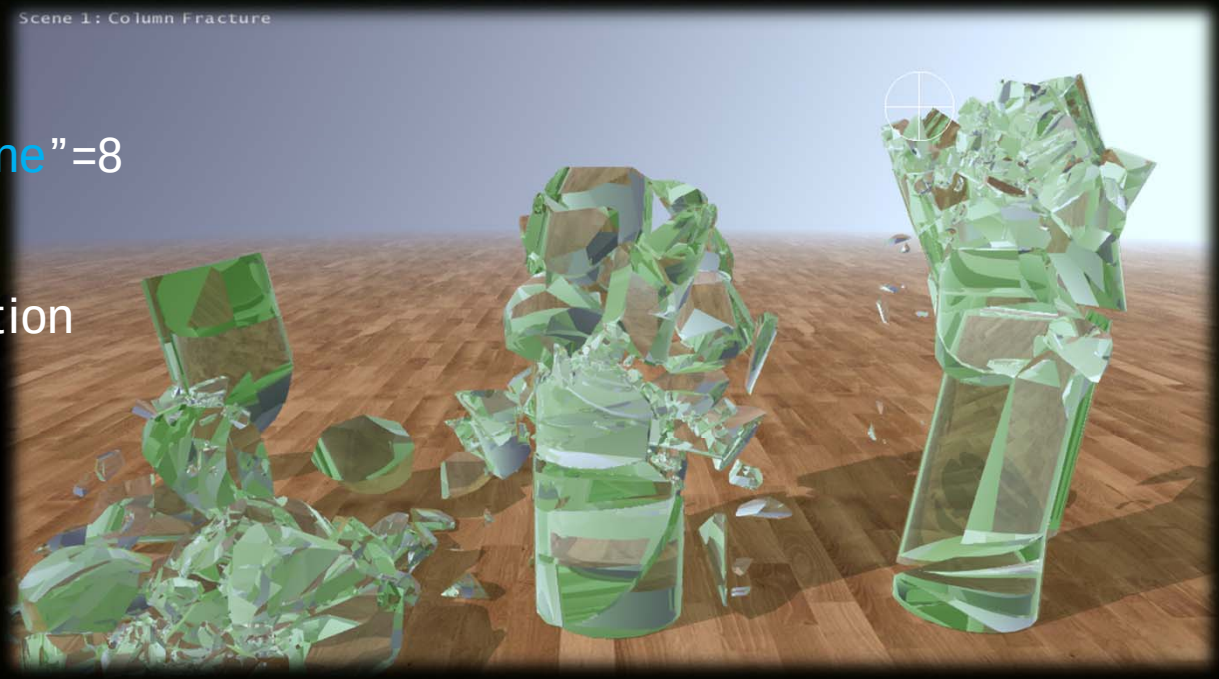
- "Sbvh" is up to 8X faster
- "Lbvh" is extremely fast and works on very large datasets
- BVH Refinement optimizes the quality of a BVH
  - Smoother scene editing
  - Smoother animation



OptiX 3 CUDA Interop Fracture Demo

Using OptiX and PhysX together

- NVIDIA PhysX GPU Rigid Bodies
- CUDA Interop for geometry
- BVH Refinement: “refit”=1 “refine”=8
- OpenGL Interop for TXAA
- Glass shader with Fresnel reflection
  - Max ray depth of 12
  - About 350,000 triangles



Developers wanting to trying OptiX

- Procedure:
  1. Go to NVIDIA Developer Zone
     https://developer.nvidia.com/optix
  2. Grab the OptiX SDK
  3. Start coding
  4. OptiX is completely free to use and deploy

CPU Fallback is available for license to developers with commercial product


General GPU Ray Tracing
Topics relating to most
GPU ray tracing applications

GPU Ray Tracing Similarities – Performance

- Single GPU Ray Tracing Speed
  - Usually linear to GPU cores and Core Clock – *for a given GPU generation*
  - Gains between GPU generations often vary per application / technique
- Multi-GPU Ray Tracing Speed
  - Solution dependent, Common in Renderers, OptiX supports by default
  - Scaling efficiency varies by solution, with slower techniques usually scaling better than fast ones
- Cluster Speed (multi-machine rendering)
  - Solution dependent, capabilities vary (e.g., Iray supports it, OptiX doesn't)

Multi-GPU Configurations

- "SLI" configuration is not needed for multi-GPU ray tracing (and can actually interfere, especially with 3 & 4 way SLI)
- Dual GPU = Easy
- 3 or 4 GPUs = usually a matter of having enough power
- 5 to 8 GPUs = usually requires motherboards with a much large VBIOS
  IMPORTANT to CHECK with YOUR SUPPLIER for what they support

- Late model multi-CPU motherboards:
  - The incorrect pairing of PCI-EX slots and CPUs can greatly impact performance
  - Dual socket motherboards having only one CPU can leave PCI-EX slots "dark"
-

GPU Ray Tracing Similarities – Hardware

- GPU memory size is most often key to what GPU is "right for you"
  - Entire scene must usually fit within GPU memory – to work AT ALL
  - Multiple GPUs can NOT "pool" memory; entire scene must fit onto each
  - If Out-of-Core is supported (as in OptiX), it's much slower than fitting in memory
- Nearly all renderers are Single Precision (e.g., double precision speed not important)
- ECC (error correction) is not needed
  - Reserves ½ GB on a 3 GB board; No Accuracy Benefit; Slows performance a bit ,
- Windows 7 is a bit slower than Windows XP or Linux
- Consumer GPUs are not designed for "data center" usage, while Pro GPUs are.
  – Failures can happen when using Consumer cards for 24/7 rendering.

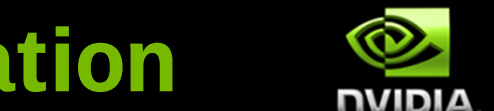
GPU Ray Tracing Similarities – Interaction

- GPU Computing (Ray Tracing) competes with system graphics
  - GPUs are still singularly focused: Compute **or** Graphics – not simultaneous
  - Often the **single biggest** design challenge for interactive app's
- Careful Application Design is needed to achieve balanced interaction
  - Gracefully stopping for user interaction and when app doesn't have focus
  - Controlling mouse pointers in the ray tracing app
- **Or simply use Multi-GPU**
  - One GPU for graphics, additional GPU(s) for compute (Ray Tracing)
  - Becoming mainstream with NVIDIA Maximus = Quadro + Tesla(s)

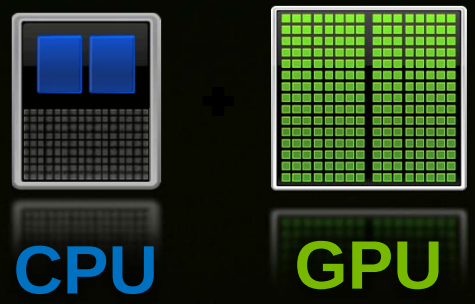
Multi-GPU Considerations for Development

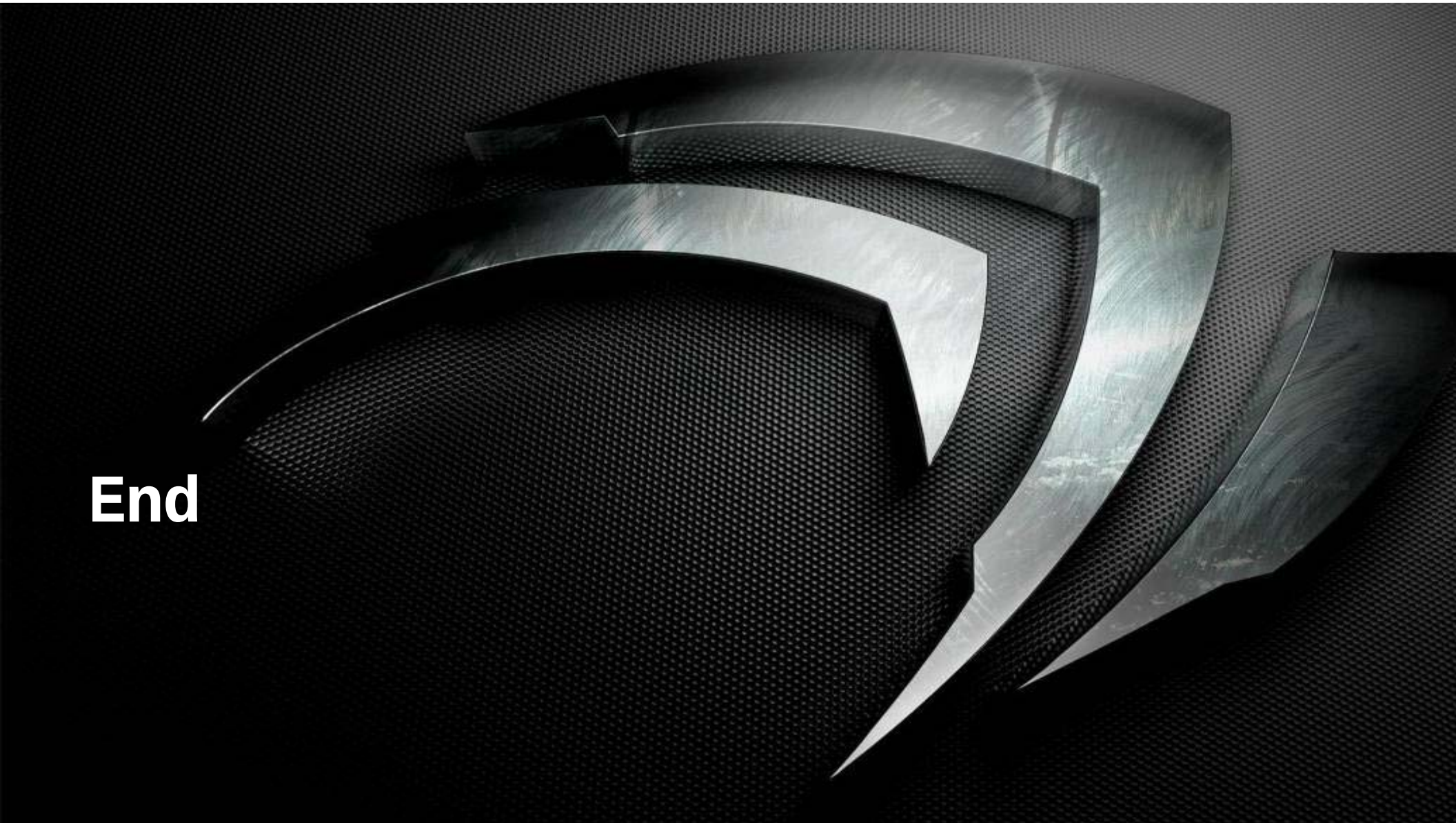
- Differing GPUs can mean different Compute capabilities
  - Not just between architectures (e.g., Fermi vs. Kepler) but sometimes within an architecture (e.g., GF100 vs. GF104 or GK104 and GK110)
  - Either insist on HW consistency from users, program to lowest denominator, or have multiple code paths
- TCC (Tesla Compute Cluster) mode for Windows
  - Default driver mode for new C-Class Tesla's (C2075 and all Kepler class)
  - Compute-only mode; GPU no longer a Windows graphics device
  - Has parity with WDDM driver with CUDA 5.0

Solutions Vary in their GPU Exploitation



- A top end Fermi GPU will typically ray trace 4 to 12 times faster than dedicated x86 code running on a good quad-core CPU
- Constant CPU Compute challenge is to keep the GPU "busy"
  - Gains on complex tasks often greater than for simple ones
  - Particularly evident with multiple GPUs, where data transfers impact simple tasks more
  - Can mean the technique needs to be rethought in how it's scheduling work for the GPU
  - Example OptiX 2.1: previous versions tuned for simple data loads, now tuned for complex loads, with a 30-80% speed increase




End