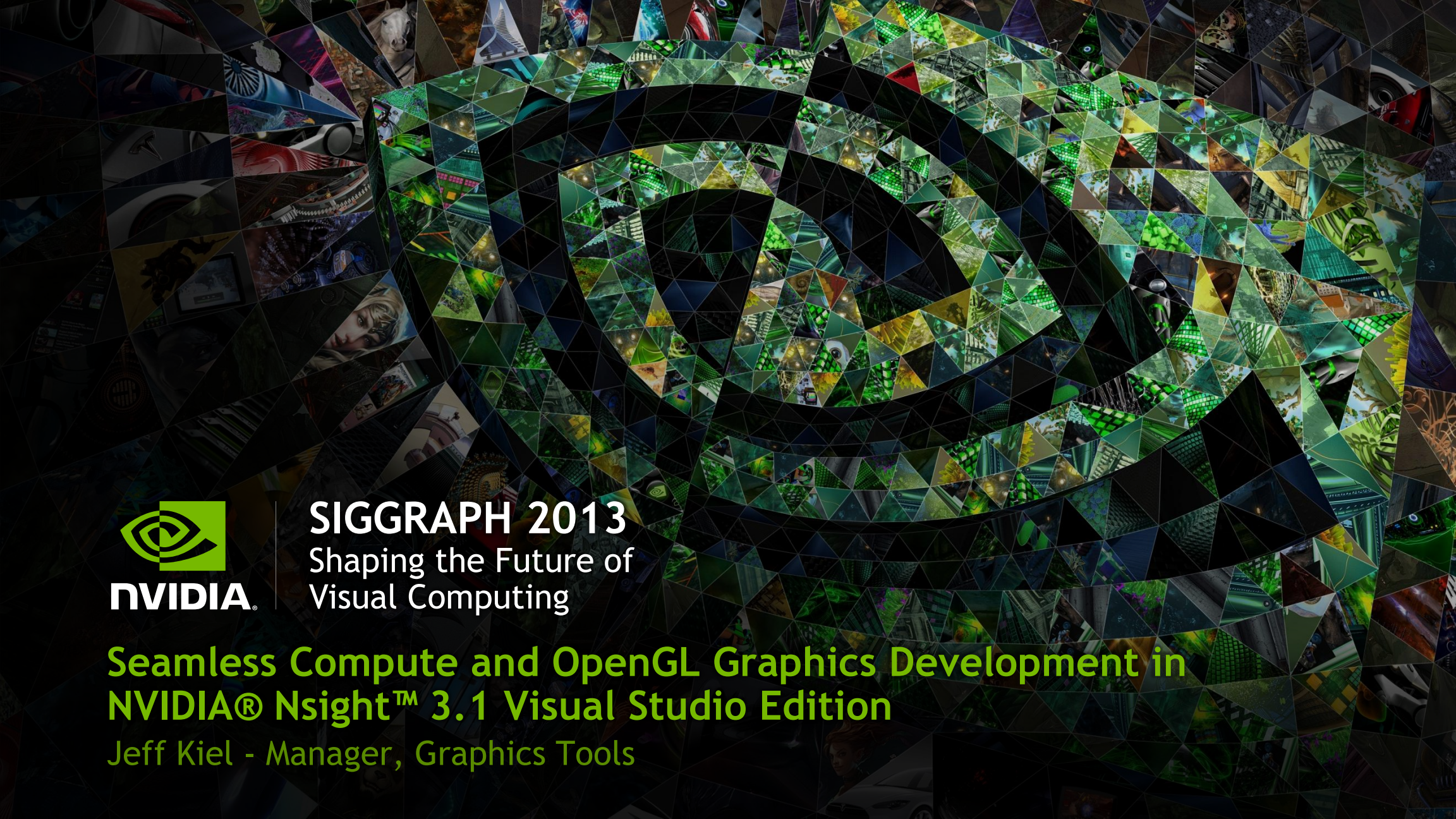


NVIDIA

SIGGRAPH 2013
Shaping the Future of
Visual Computing

Seamless Compute and OpenGL Graphics Development in NVIDIA® Nsight™ 3.1 Visual Studio Edition

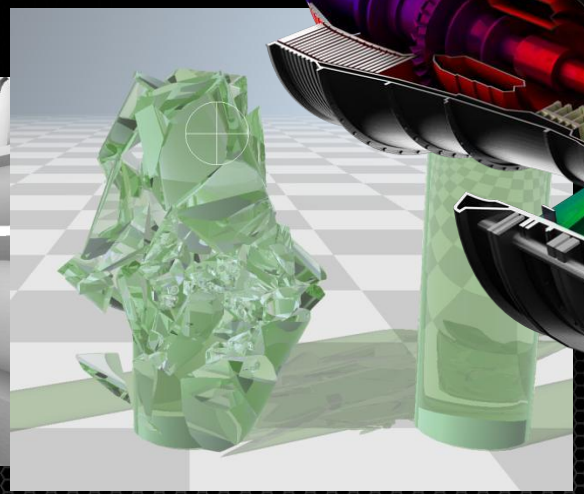
Jeff Kiel - Manager, Graphics Tools

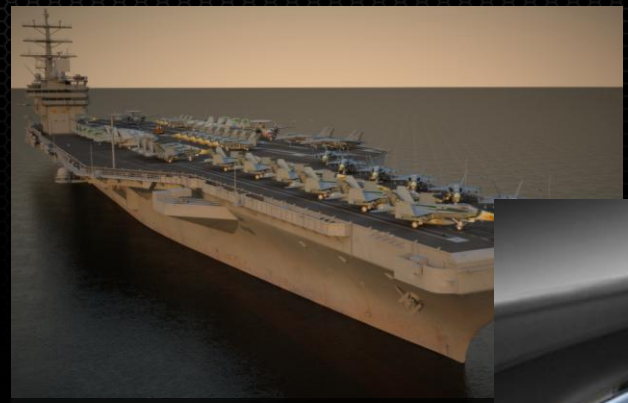
Agenda

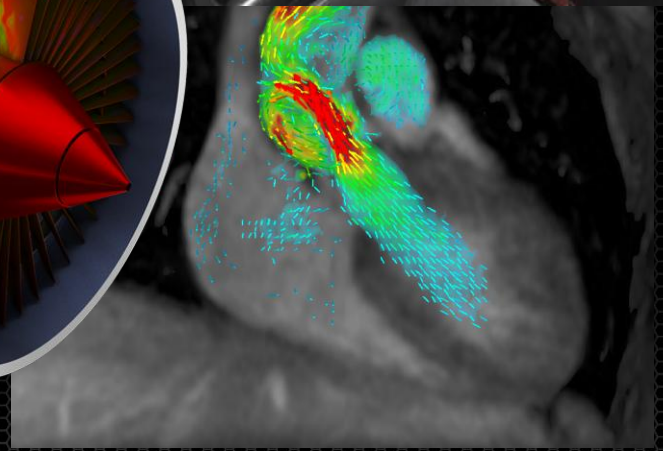
- Computational Graphics and Visual Computing
- Maximus™
- Developer Challenges
- Debugging Computational Graphics with Nsight Visual Studio Edition
- Conclusion and Q&A

Computational Graphics - Visual Computing

- Deferred Shading
- Ambient Occlusion
- Simulation & Visualization
- Medical imaging
- Manufacturing

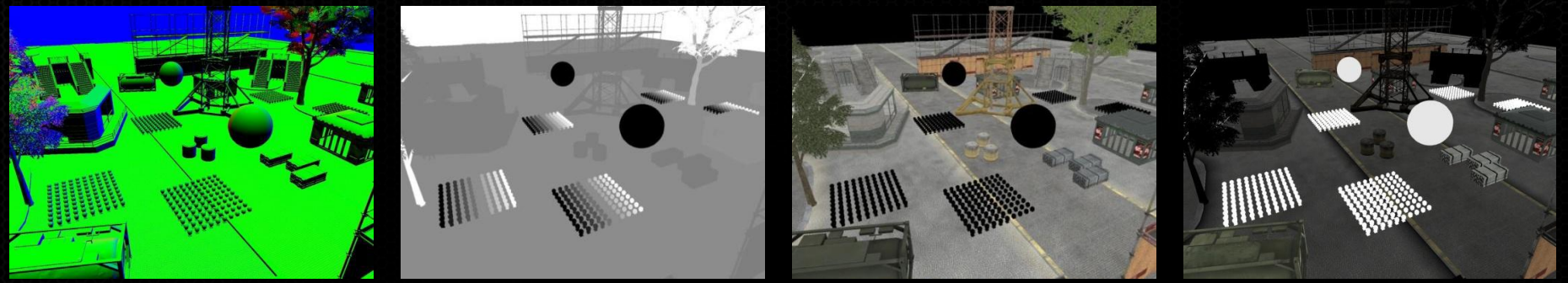






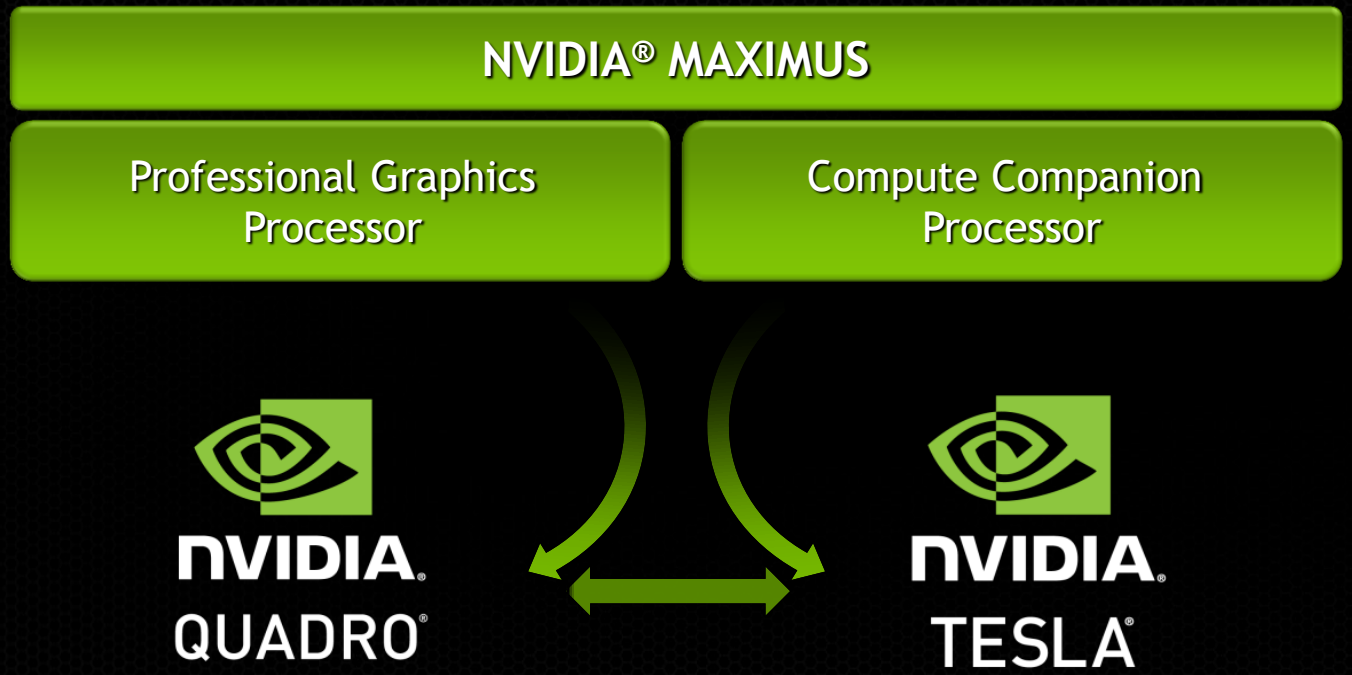
Compute Shader Based Deferred Shading

- Graphics pipeline rasterizes gbuffers for opaque surfaces

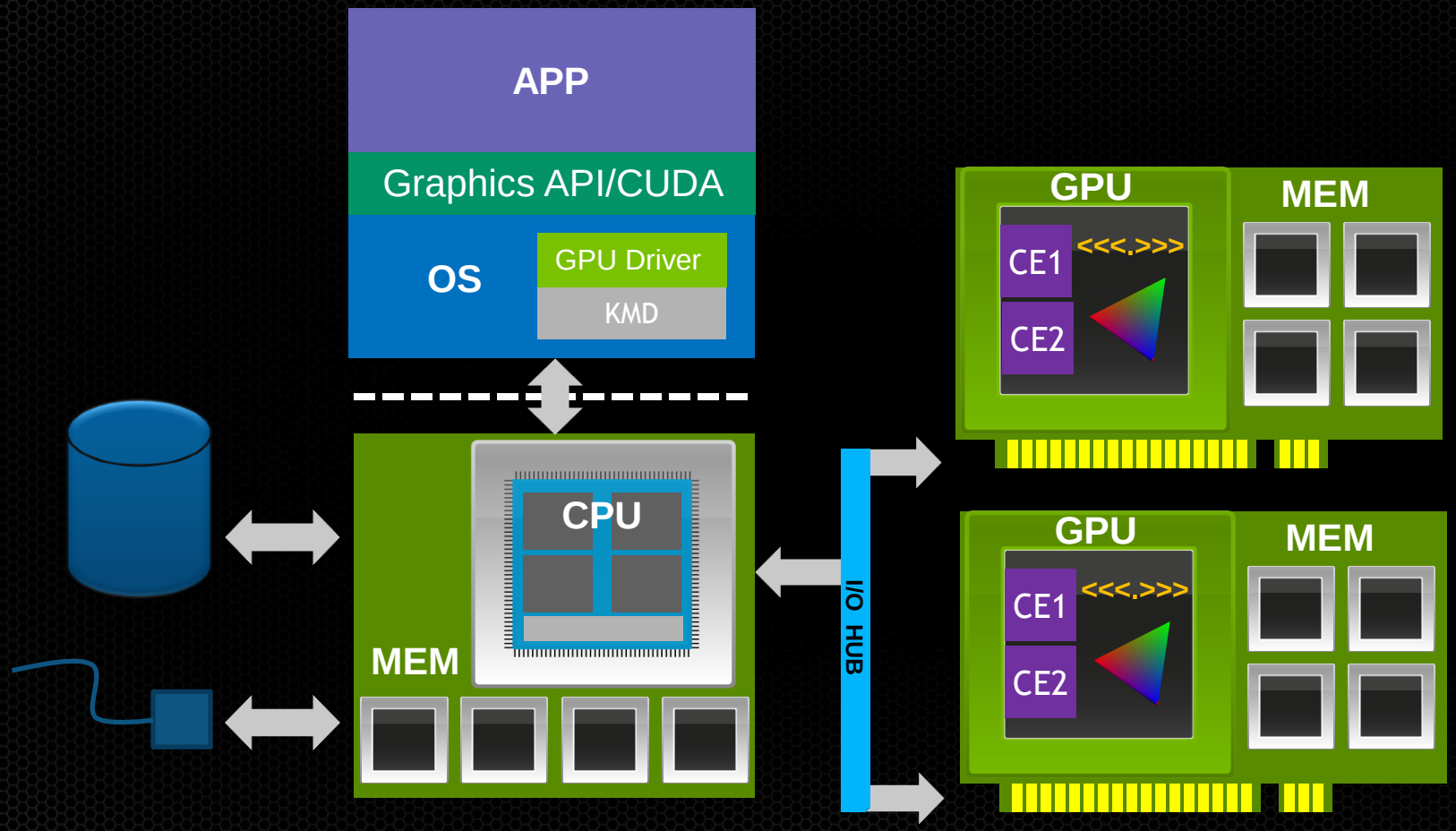


- Compute pipeline uses gbuffers
  - cull light sources
  - compute lighting
  - shading

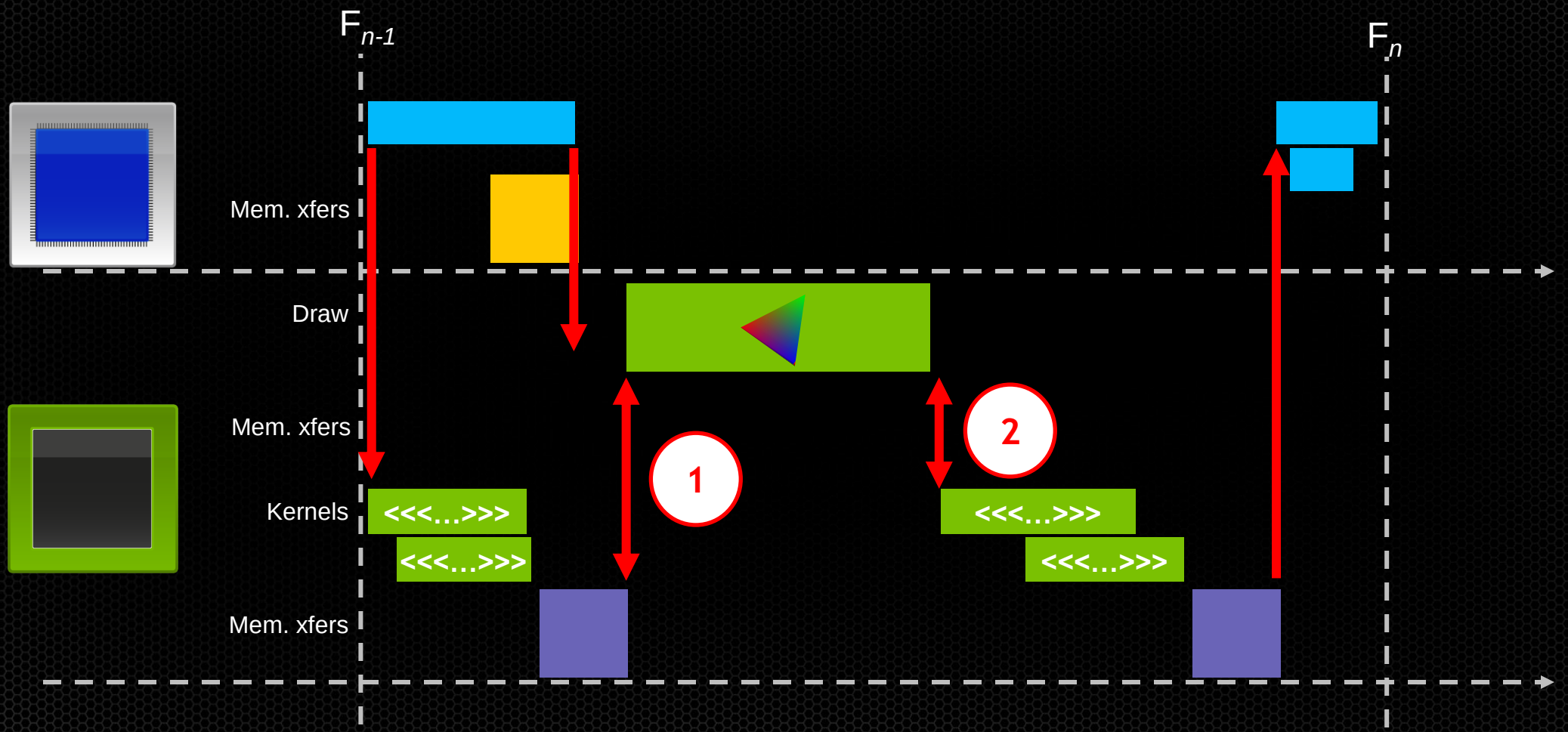
NVIDIA® Maximus™



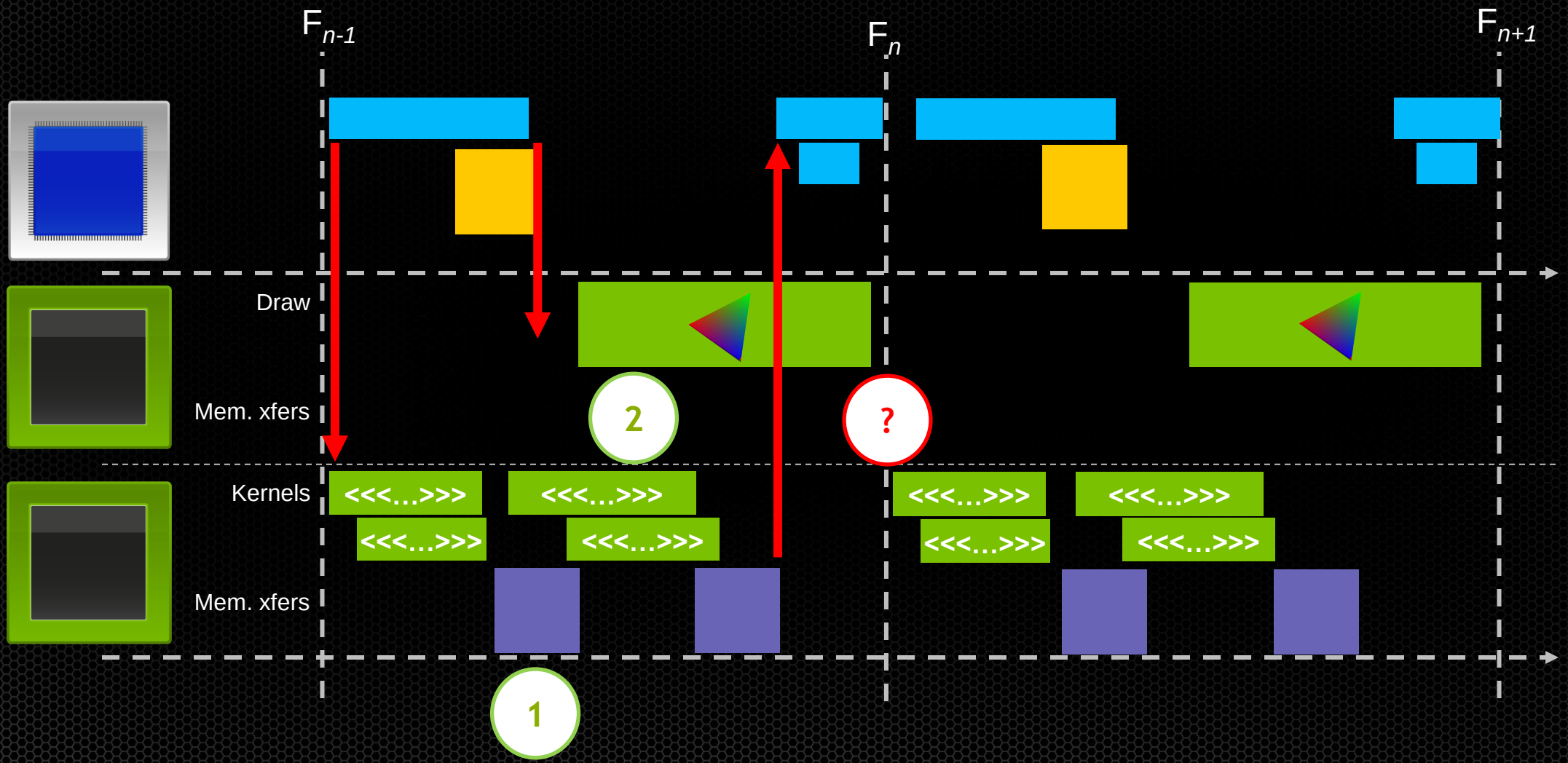
Multi-GPU Platform Overview



Single-GPU Simulation and Visualization



Multi-GPU Simulation and Visualization



Developer Challenges

- Debugging GPU Compute and Graphics with interop
- Context switching overhead
- Multi-Core/Multi-GPU race conditions
- Data transfers – host-to-device, device-to-host, P2P
- Asynchronous transfers
- Multi-core/Multi-threaded
- Multi-GPU – Graphics/Graphics, Graphics/Compute
- Concurrent kernel execution
- Driver models (TCC/WDDM)

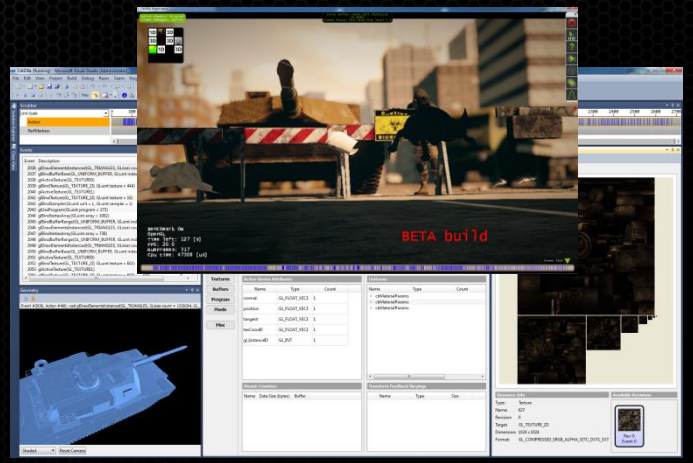
NVIDIA® Nsight™ Visual Studio Edition

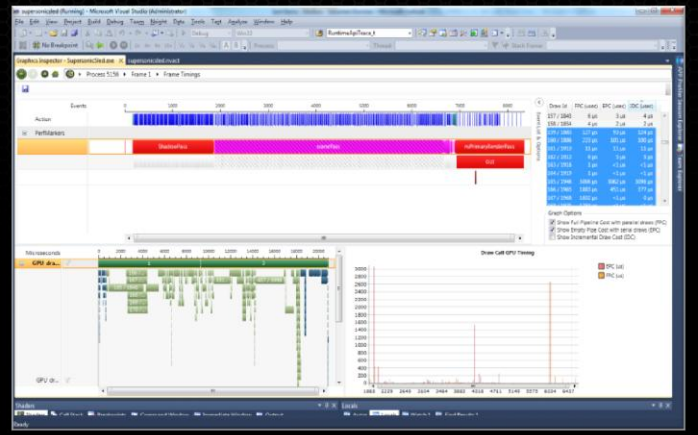
Visual Studio integrated development for GPU and CPU

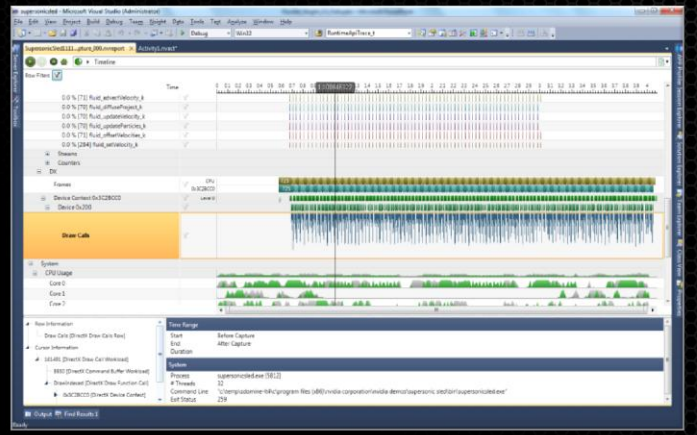
Build **Debug** **Profile**

Microsoft® Visual Studio® PARTNER

NVIDIA Nsight for Graphics Developers
OpenGL 4.2, Direct3D 9/11







Frame debugger - OpenGL 4.2, Direct3D 9/11

- Draw call and state inspection
- Frame capture and playback (source code gen D3D9/11)
- Nsight HUD for draw call scrubbing and inspection

HLSL and GLSL Shader debugger

- Native GPU shader debugging and GPU memory views
- Complex condition breakpoints and Pixel History
- Local single GPU shader debugging

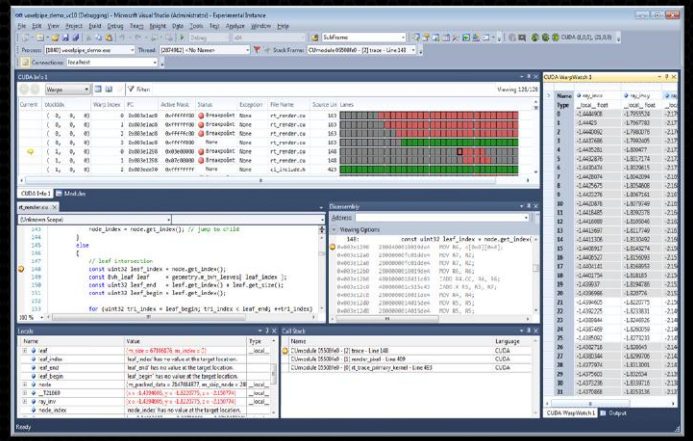
Frame profiler - OpenGL 4.2, Direct3D 9/11

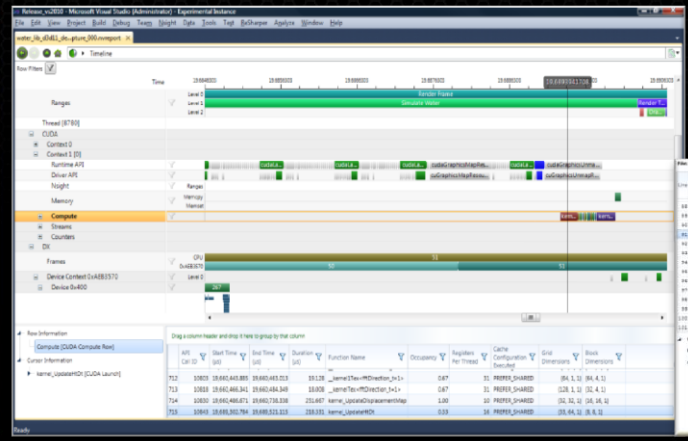
- Automatic GPU bottleneck determination
- Draw call and frame timings
- Grouping/sorting by NVTX/D3D Perf Markers and state

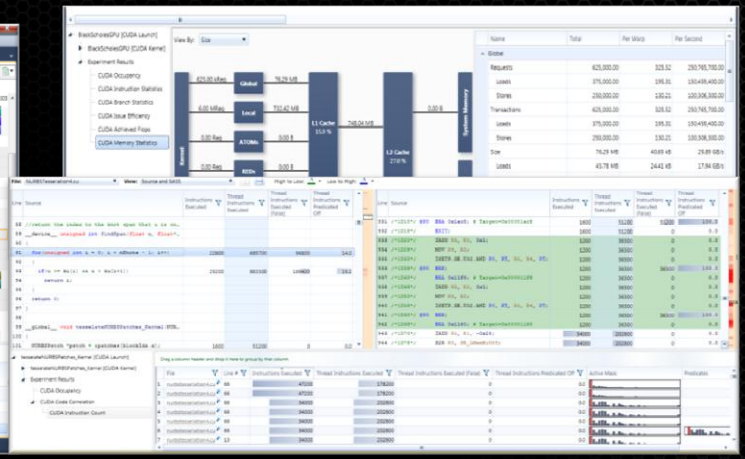
Application and system trace

- Inspect OpenGL and Direct3D workloads, CPU and GPU
- Correlate threads, call stack, API calls, WDDM kernel queues and resulting GPU workloads
- Concurrent draw call execution and memory transfer trace

NVIDIA Nsight for Compute Developers
CUDA 4.2 and 5.0 and Maximus™







CUDA debugger

- Debug CUDA kernels directly on GPU hardware
- Info page and Parallel warp watch view
- Use on-target conditional breakpoints to locate errors

CUDA memory checker

- Out of bounds memory access detection
- Enables precise error detection

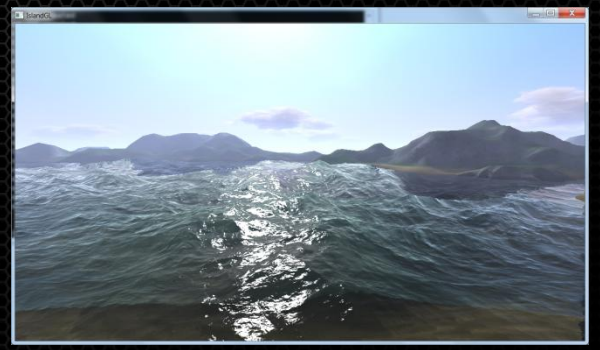
Application and system trace

- Review CUDA workloads across CPU and GPU
- Activity correlation panel

CUDA profiler

- Source code correlation
- Deep kernel analysis to detect factors limiting maximum performance
- Unlimited experiments on live kernels

Debugging and Profiling

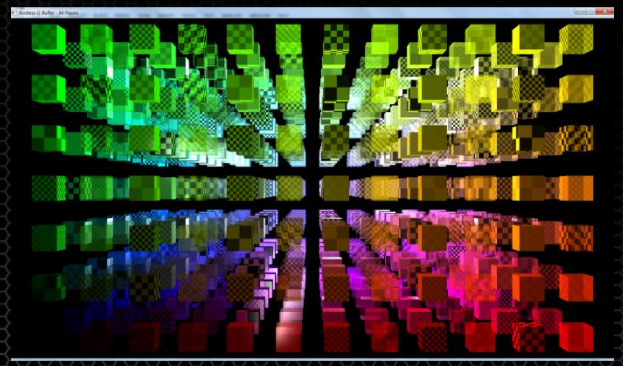


IslandsGL – GPU water simulation

Debugging

GIVoxels – Indirect illumination using voxel tracing

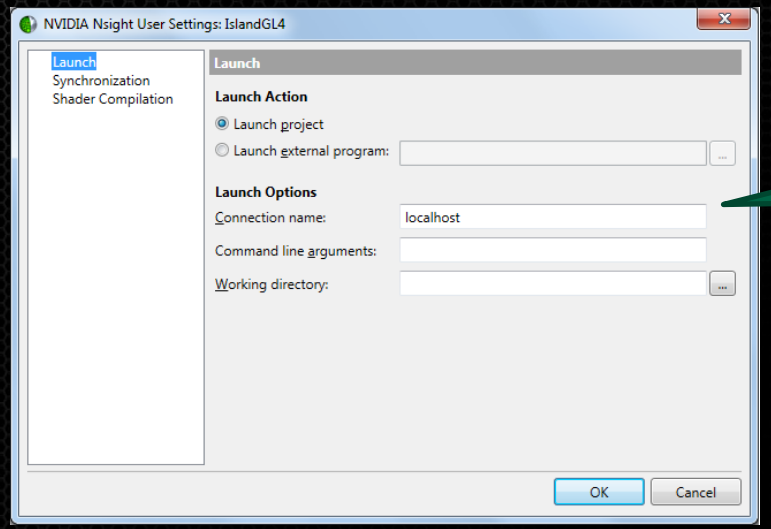
Application Tracing



Bindless GBuffer - Deferred rendering with bindless

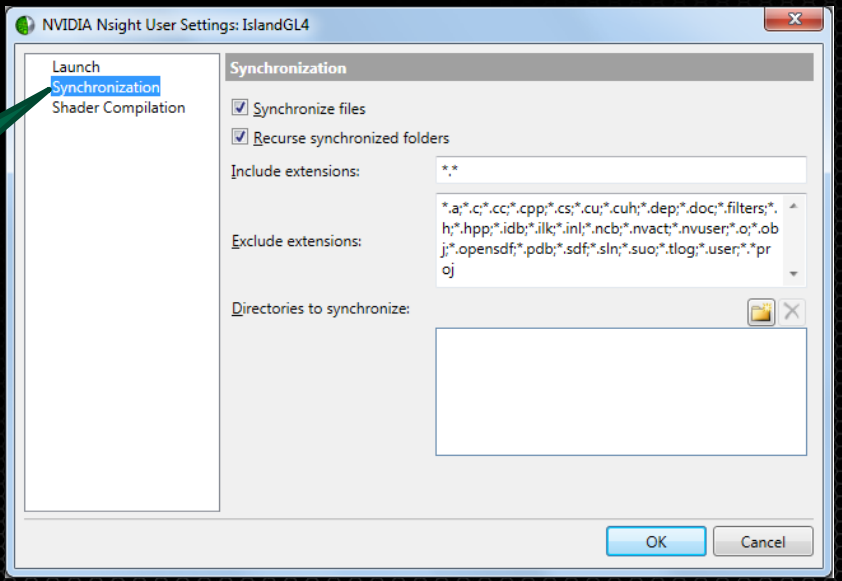
Profiling

Demo: Setting up your application

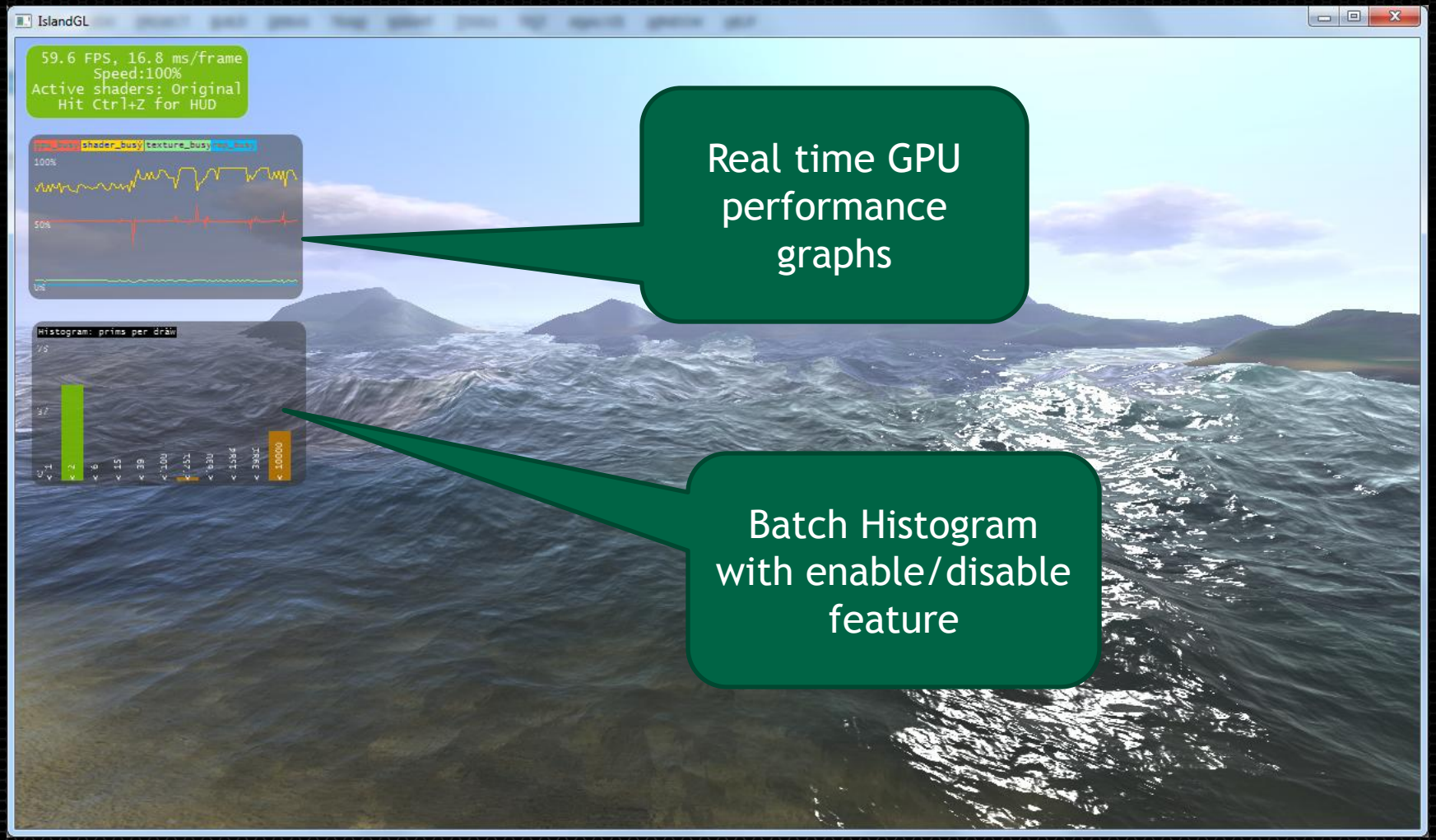


Set up target application, working directory, arguments, and machine

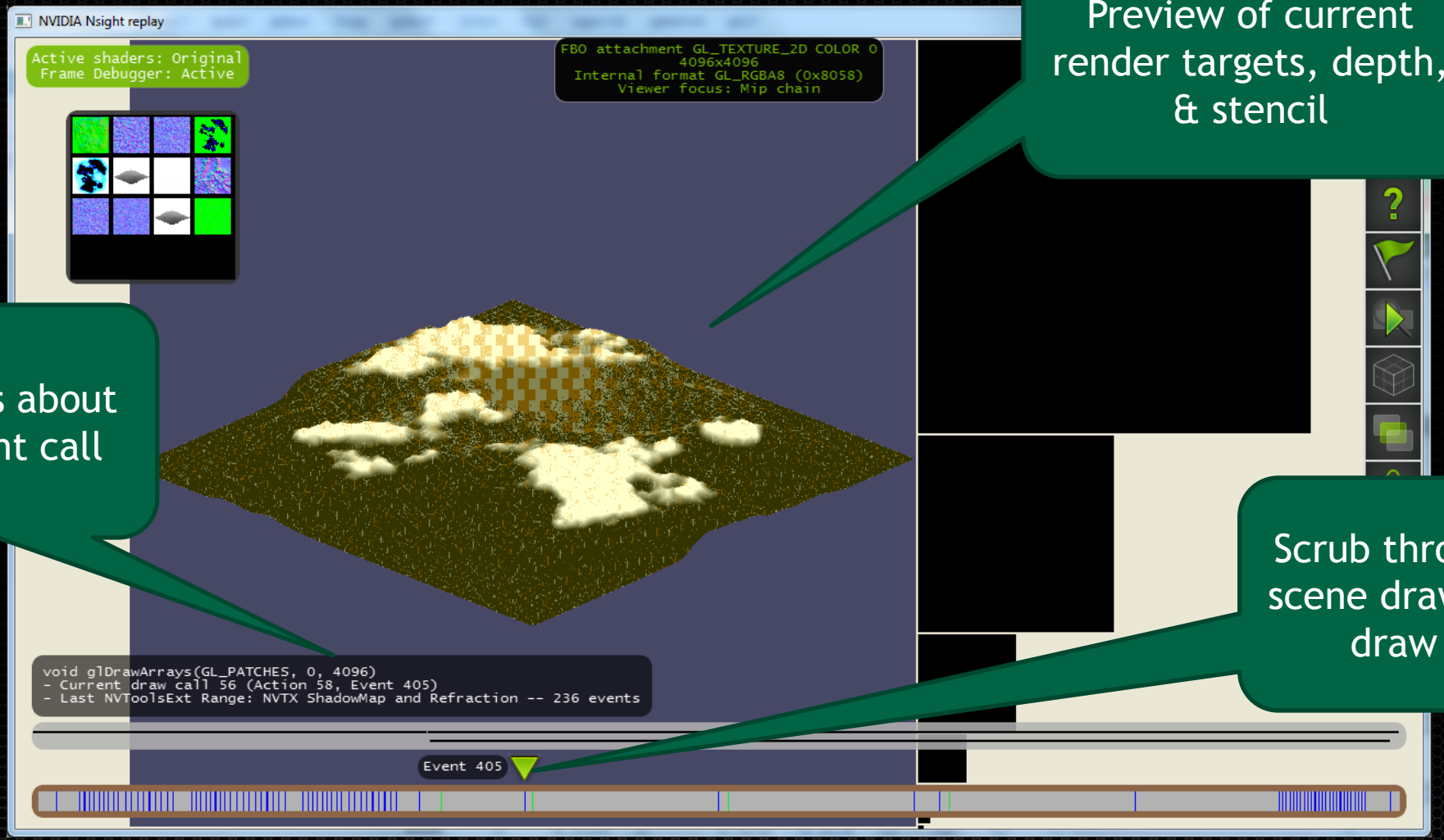
Enable serialization of application and assets to target machine



Demo: Debugging IslandsGL



Demo: Debugging IslandsGL

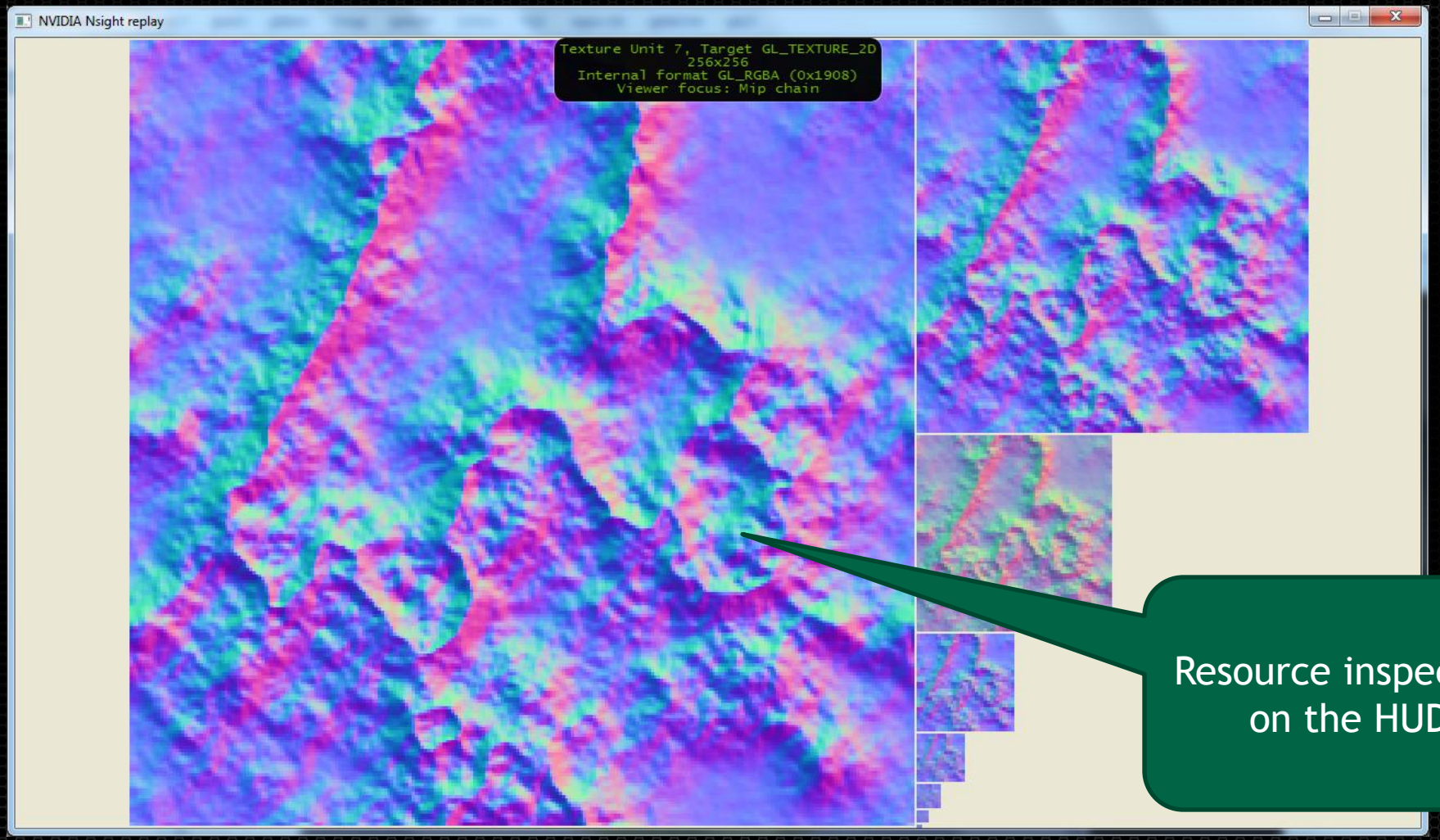


Preview of current render targets, depth, & stencil

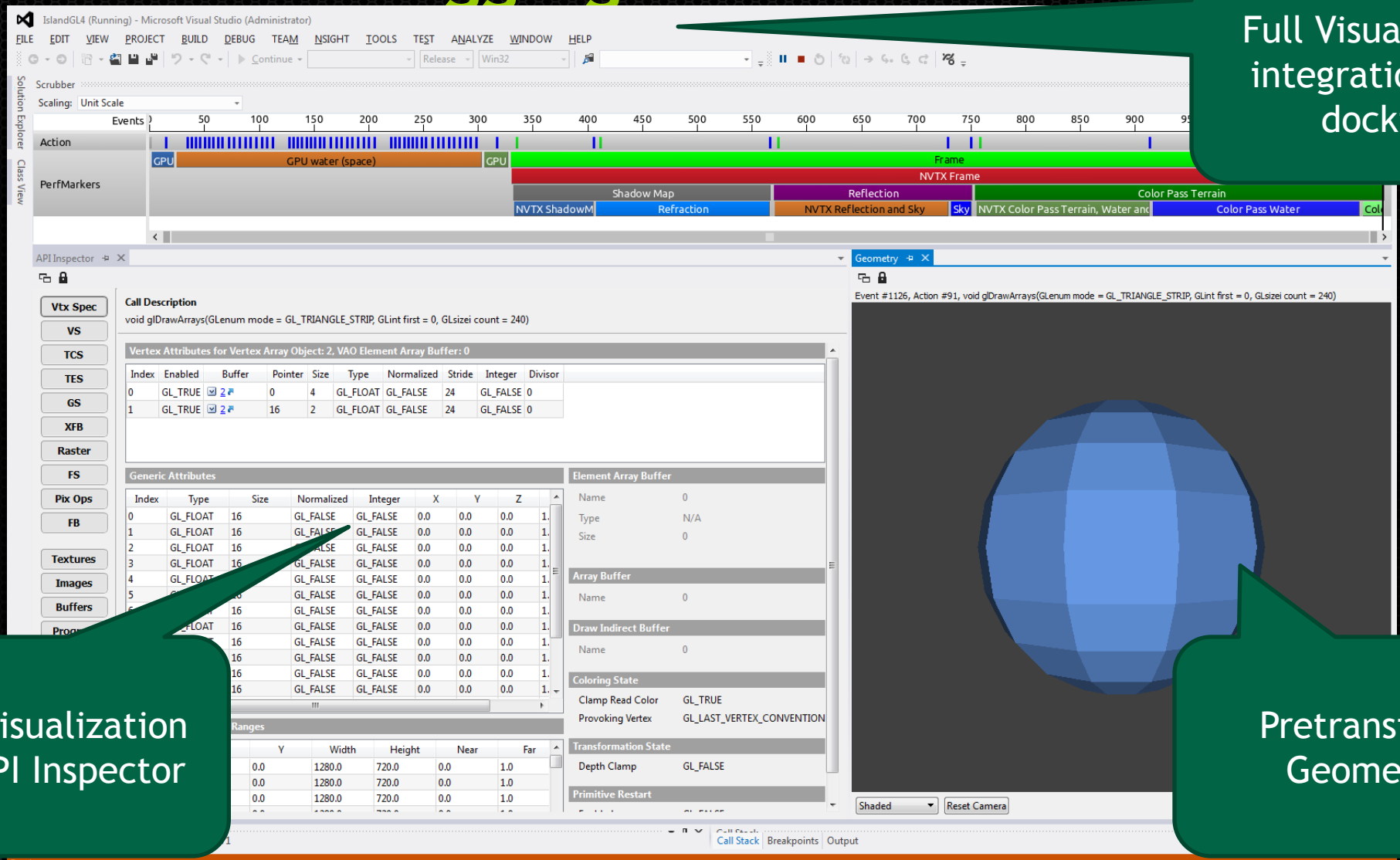
See details about the current call

Scrub through the scene draw call by draw call

Demo: Debugging IslandsGL



Demo: Debugging IslandsGL



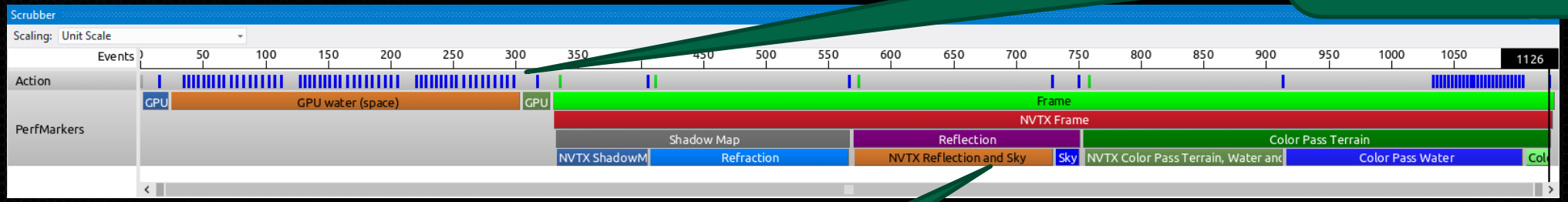
Full Visual Studio integration with docking

State visualization with API Inspector

Pretransform Geometry

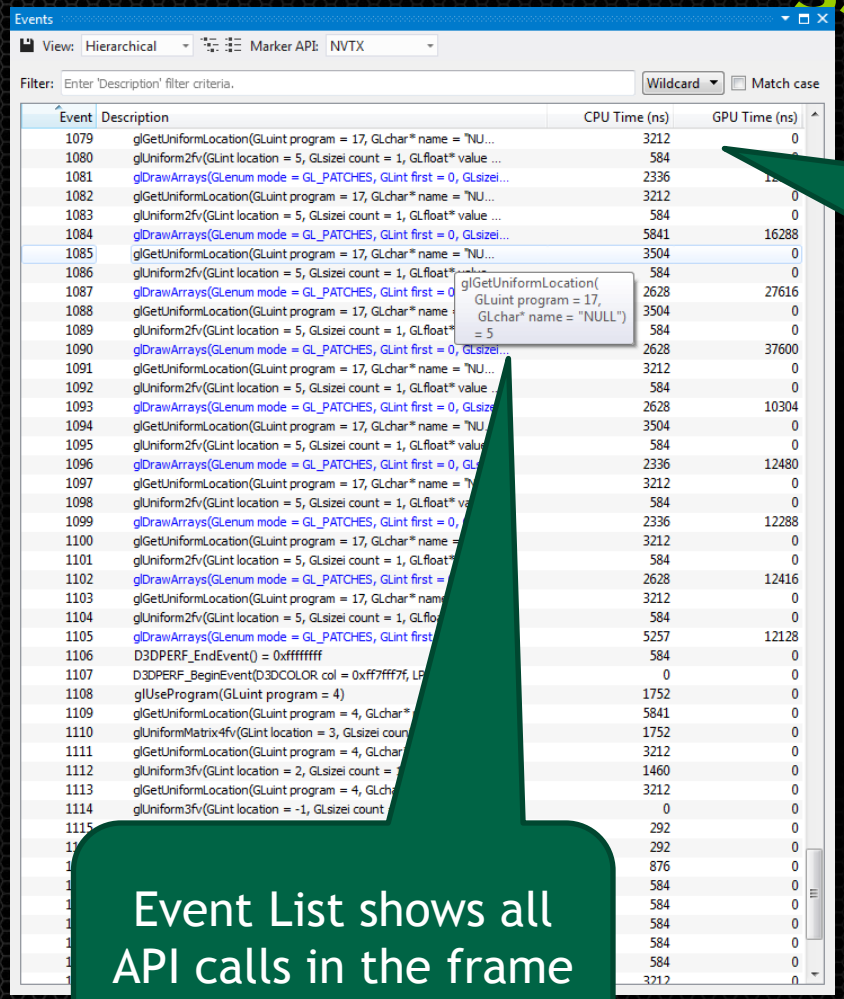
Demo: Debugging IslandsGL

Scrub through scene draw call by draw call



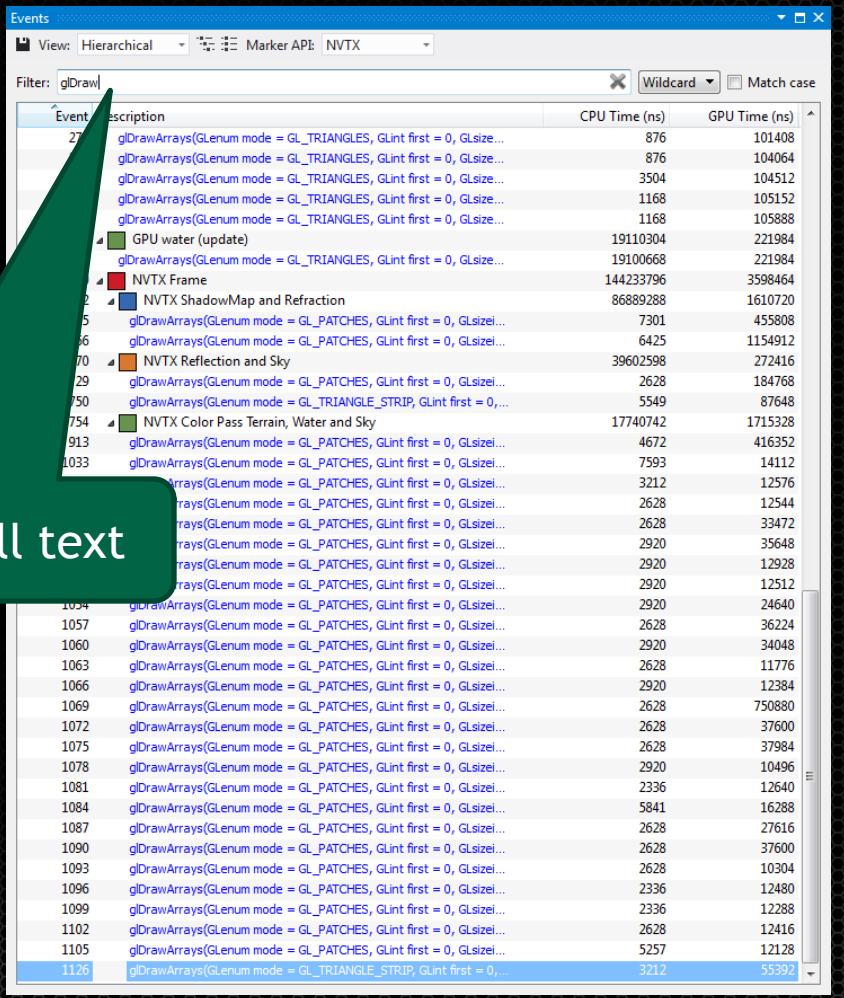
Visualize scene ranges using NV Tools Extension

Demo: Debugging IslandsGL



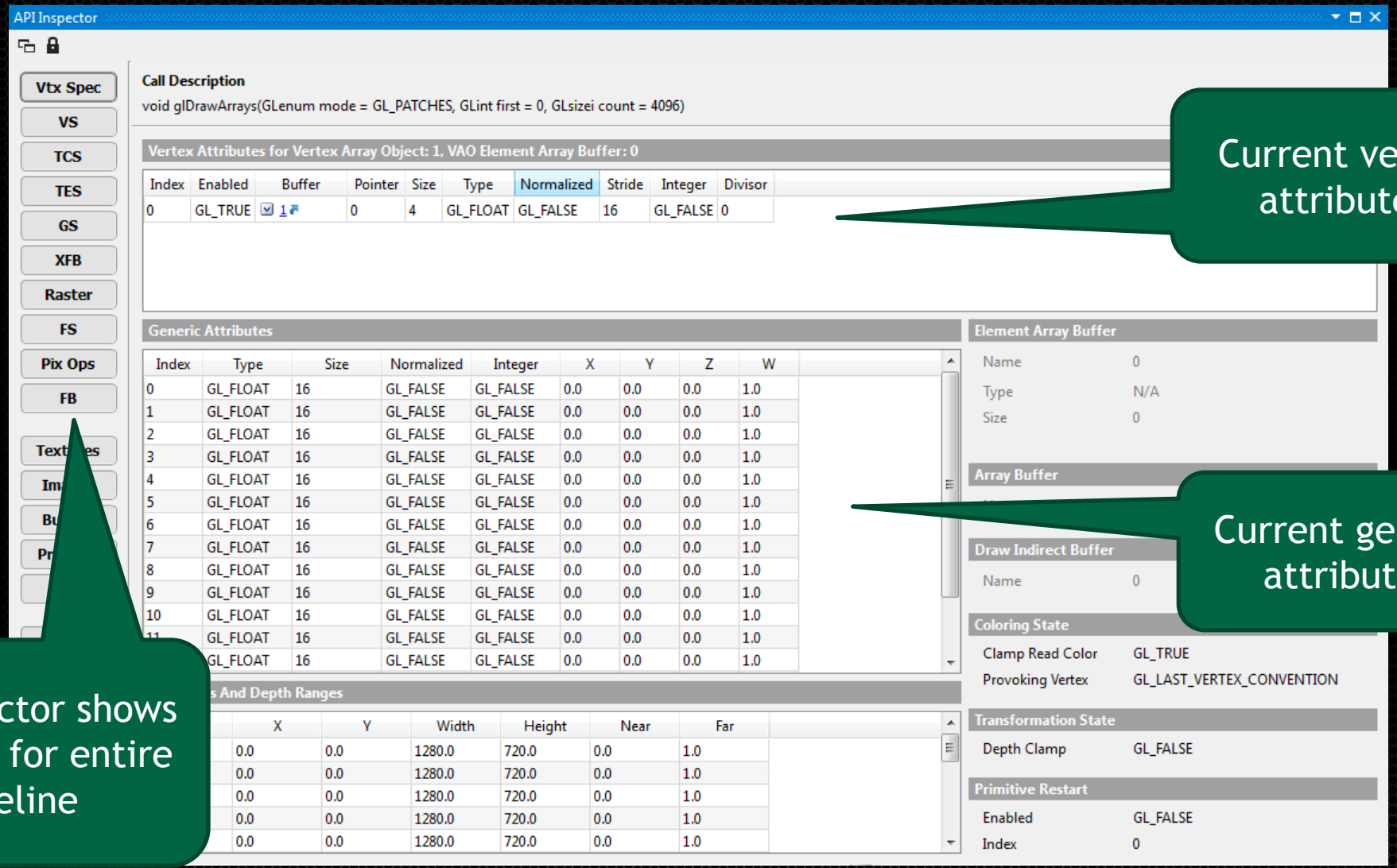
Performance stats including CPU and GPU call duration

Filter based on API call text

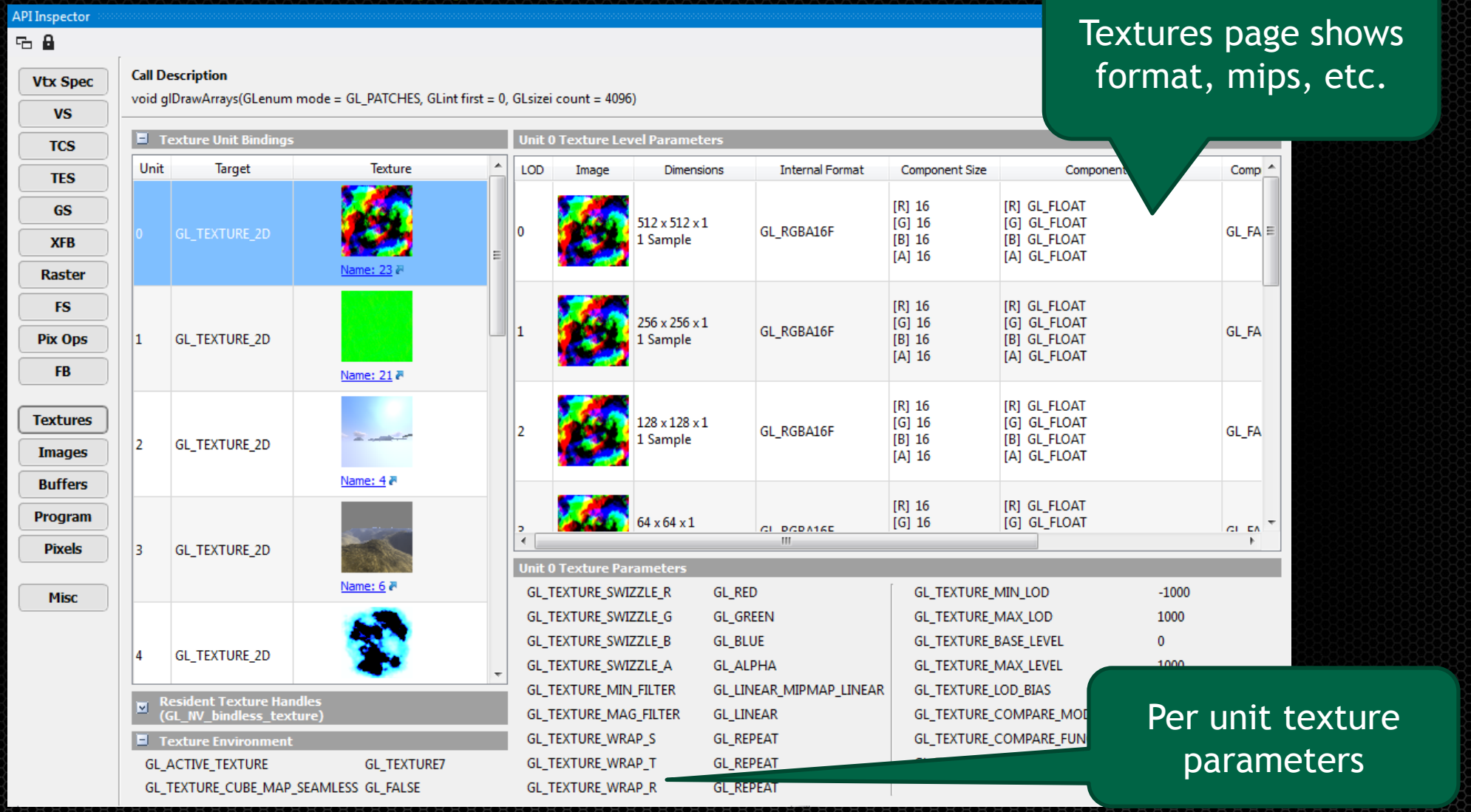


Event List shows all API calls in the frame (with parameters)

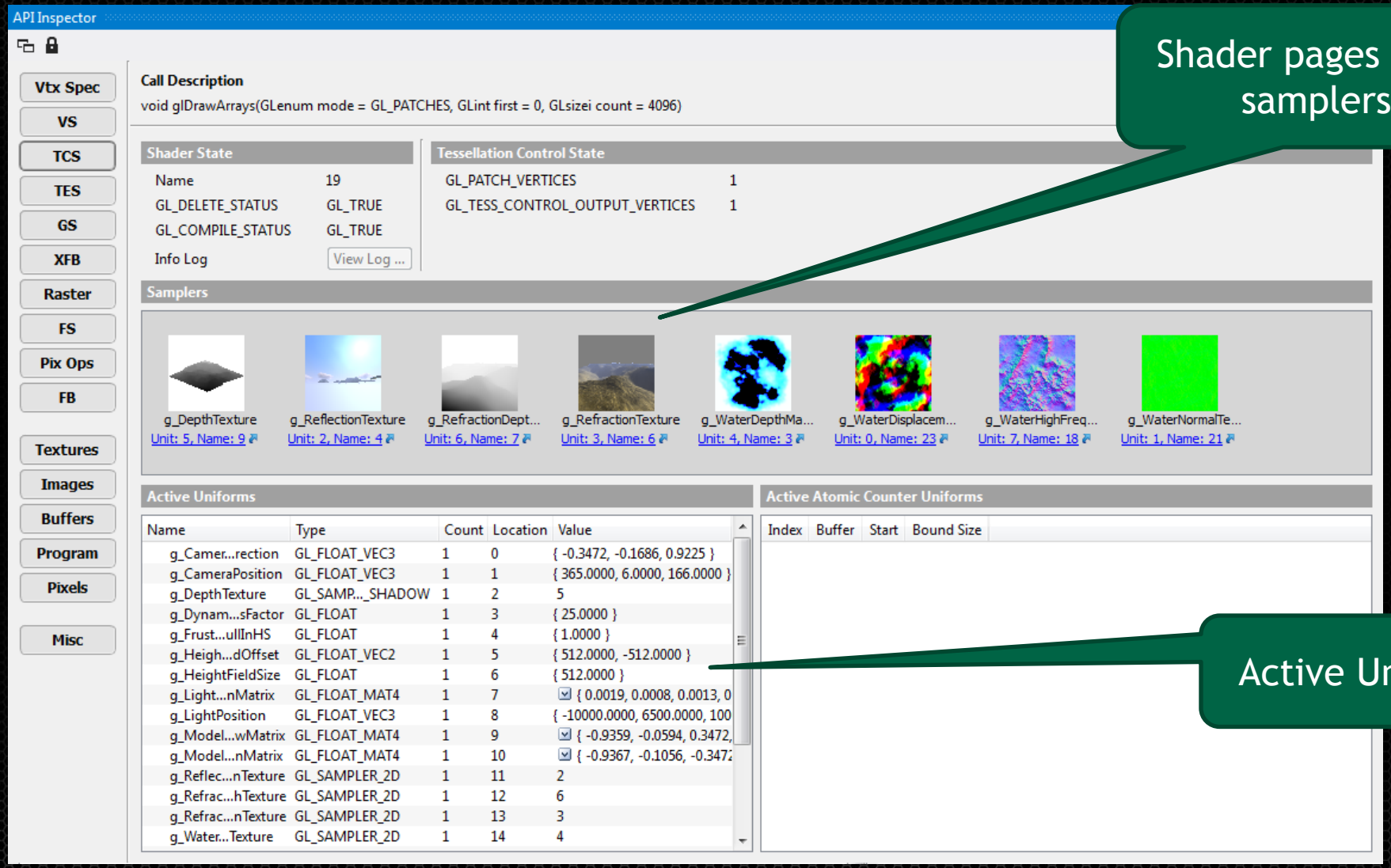
Demo: Debugging IslandsGL



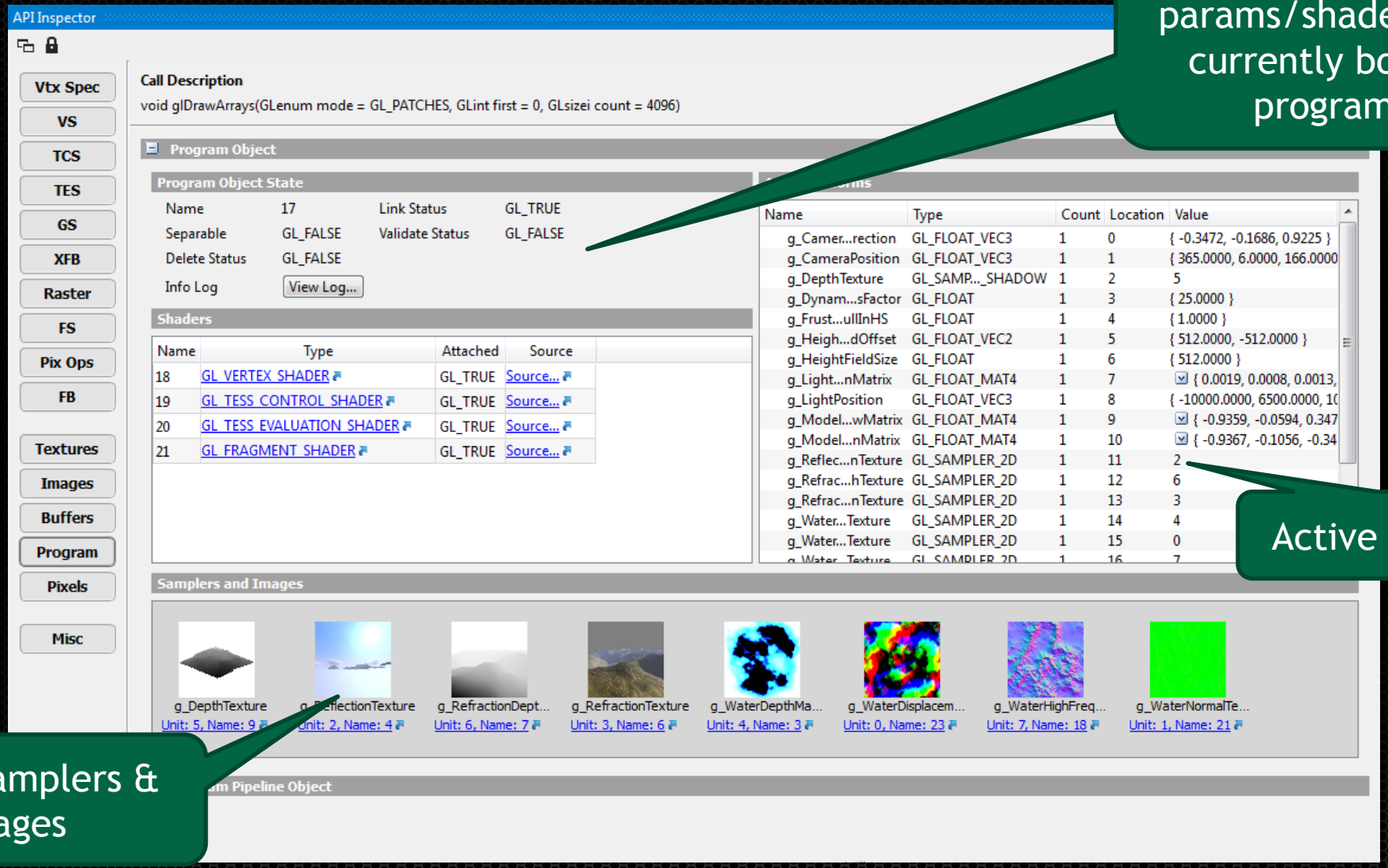
Demo: Debugging IslandsGL



Demo: Debugging IslandsGL



Demo: Debugging IslandsGL



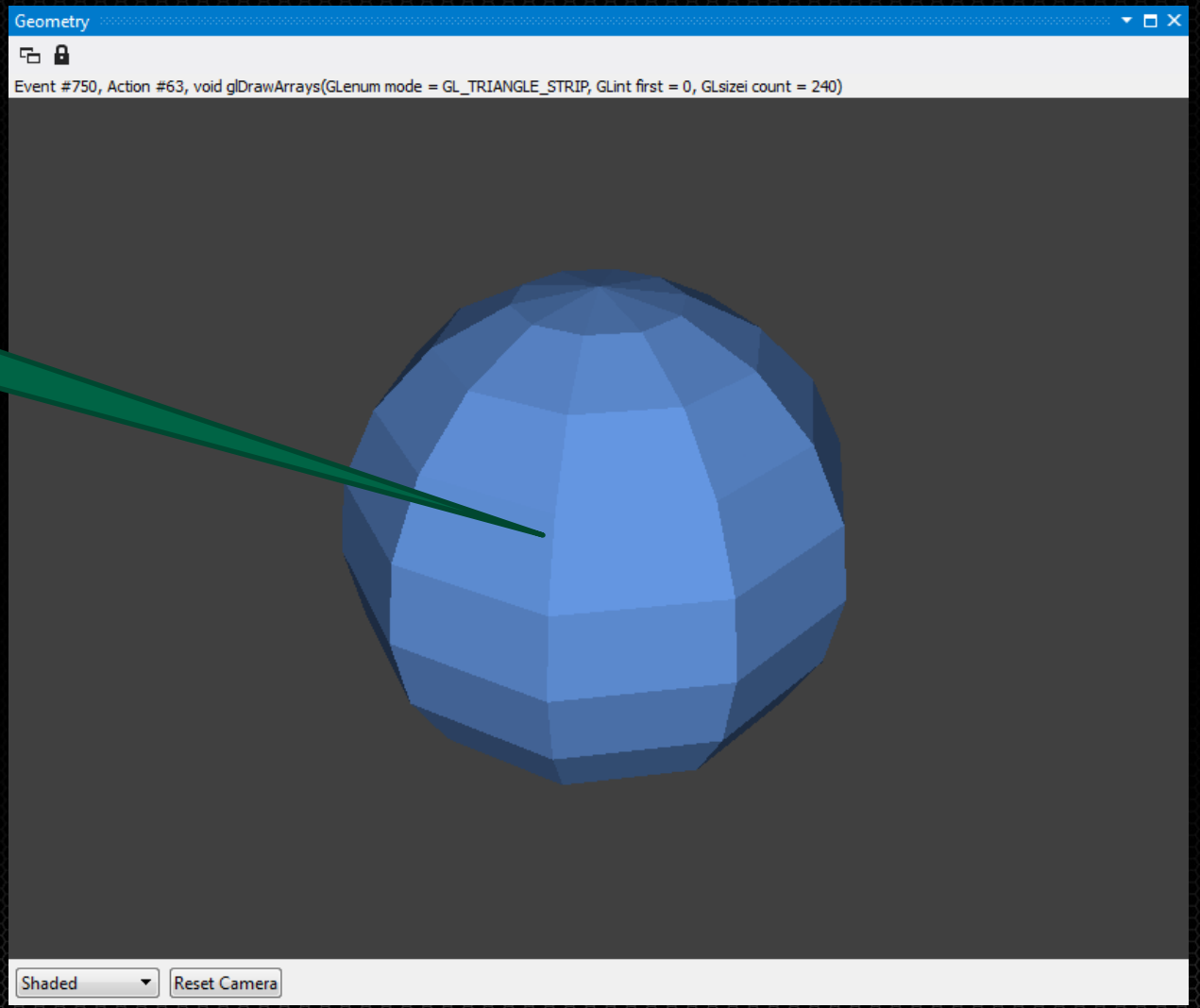
Program page shows params/shaders for currently bound program

Active uniforms

Bound samplers & images

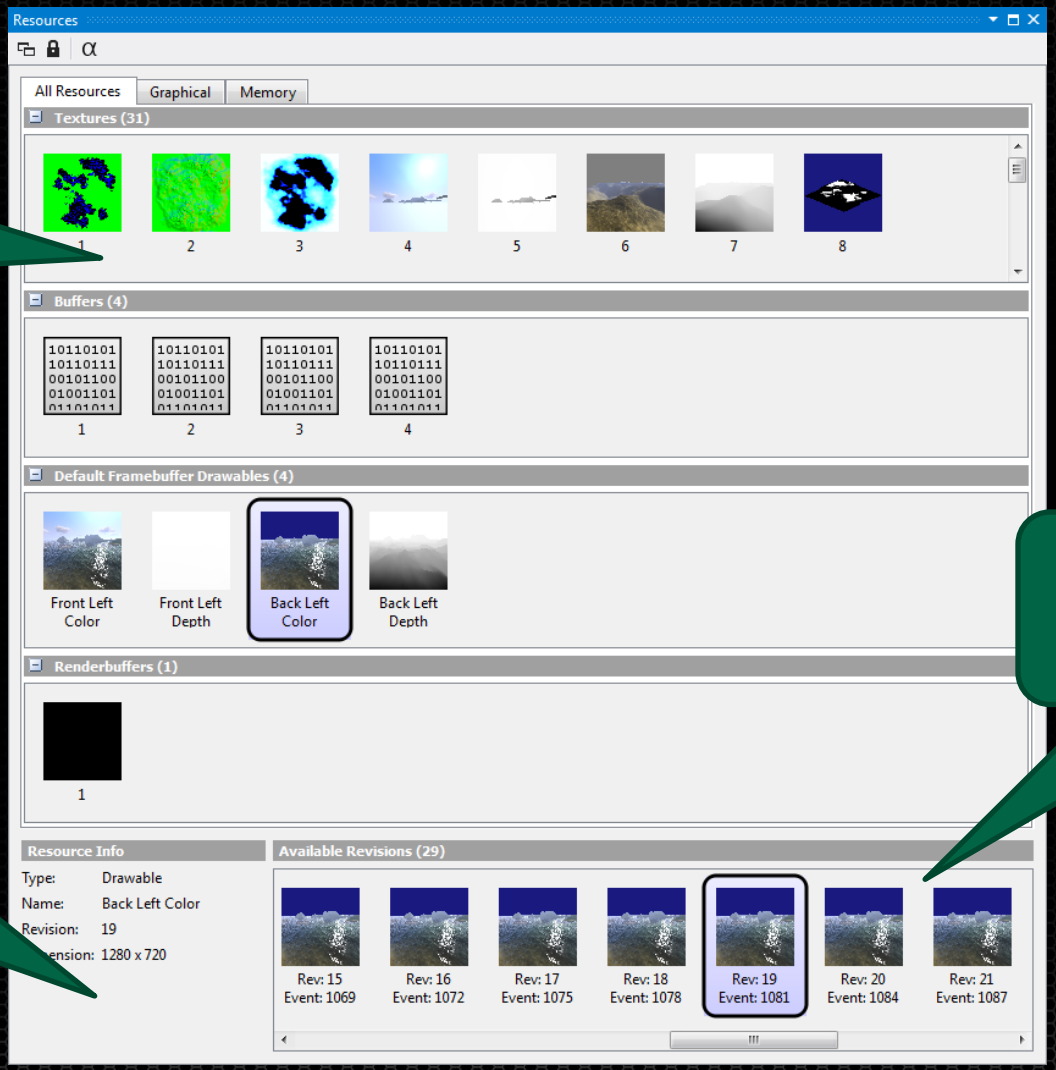
Demo: Debugging IslandsGL

Pretransform Geometry



Demo: Debugging IslandsGL

Resource window shows all used resources



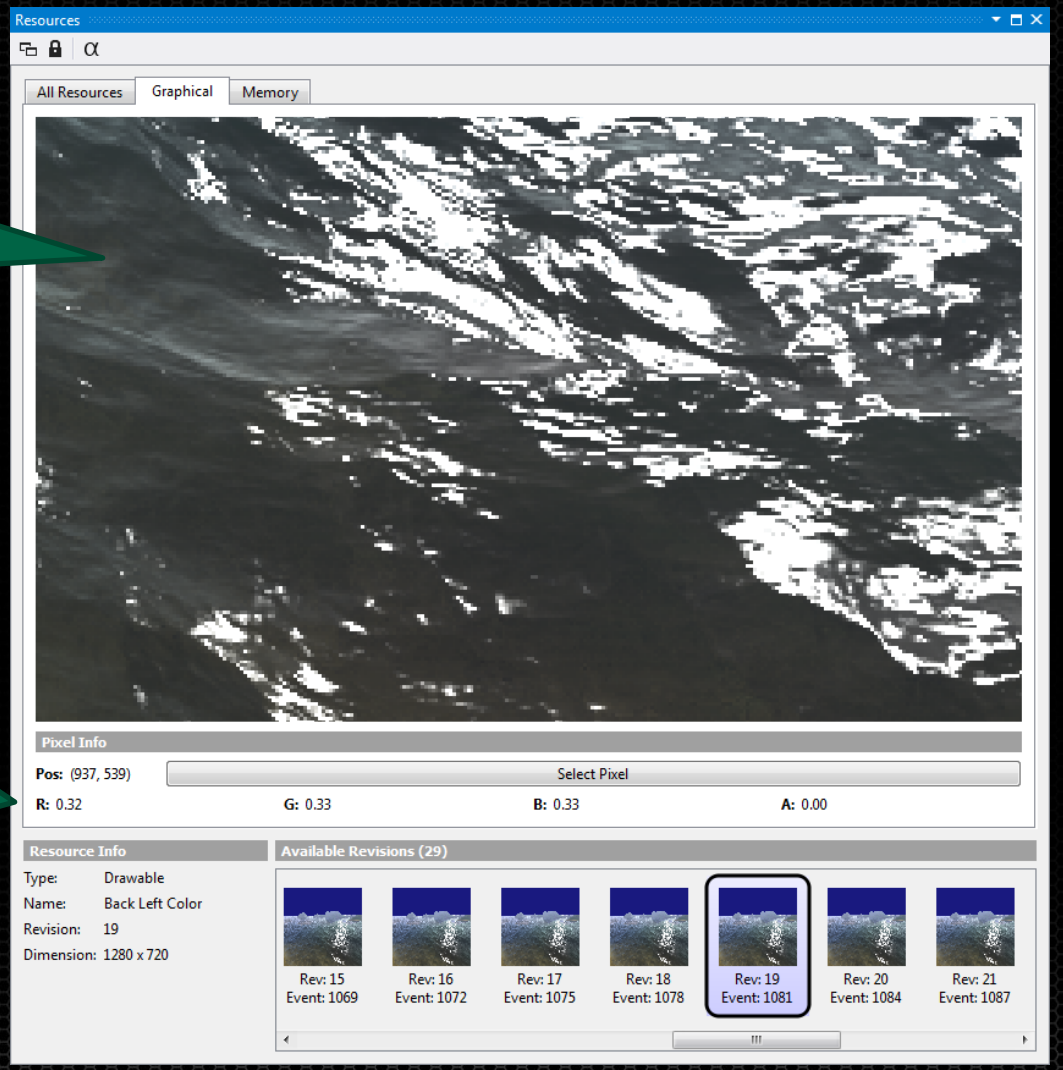
Resource details like format, size, etc.

Thumbnails of resource revisions

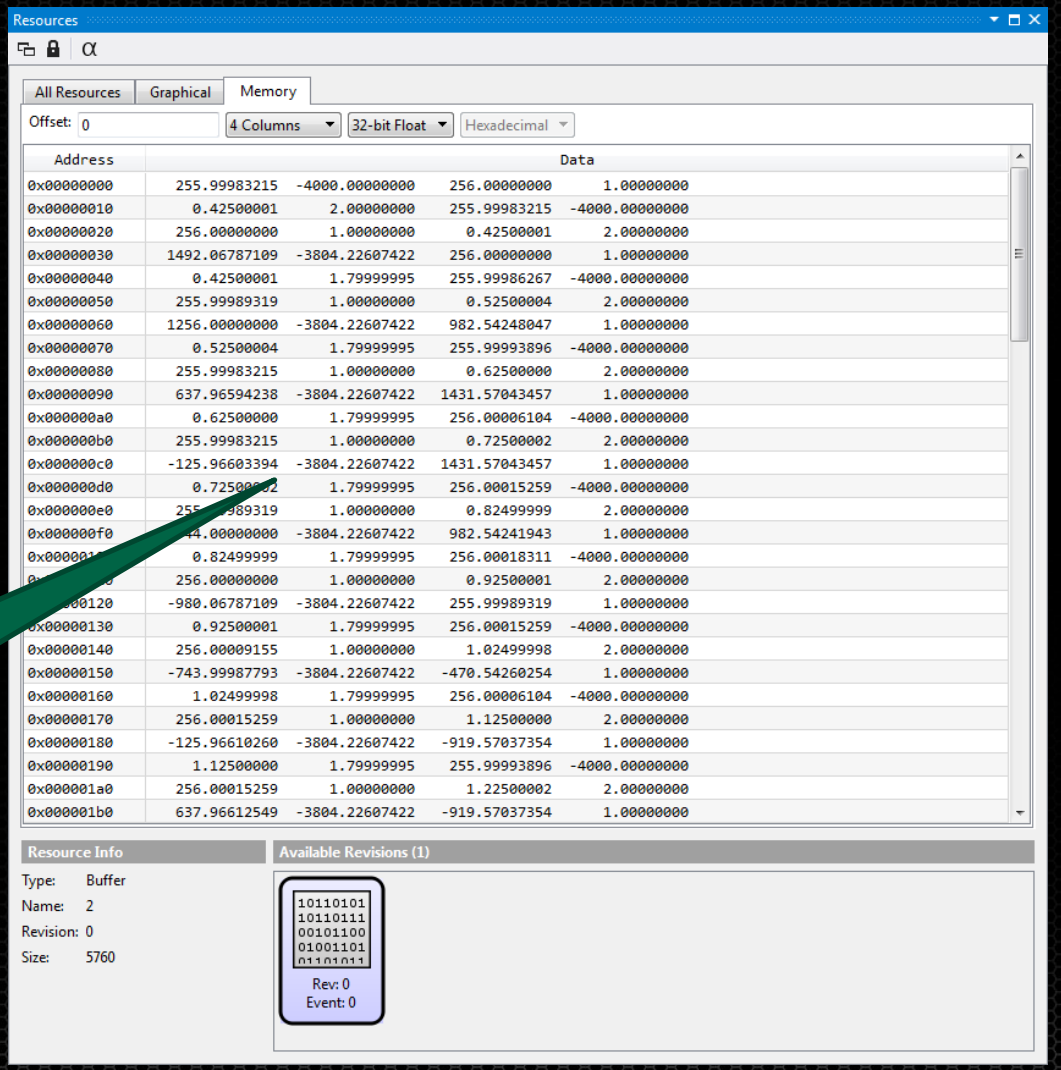
Demo: Debugging IslandsGL

Use mouse wheel to zoom in and inspect bitmap resources

Text readout of color values



Demo: Debugging IslandsGL



Display binary buffers according to the underlying data type

Demo: Debugging IslandsGL

Zoom resource display

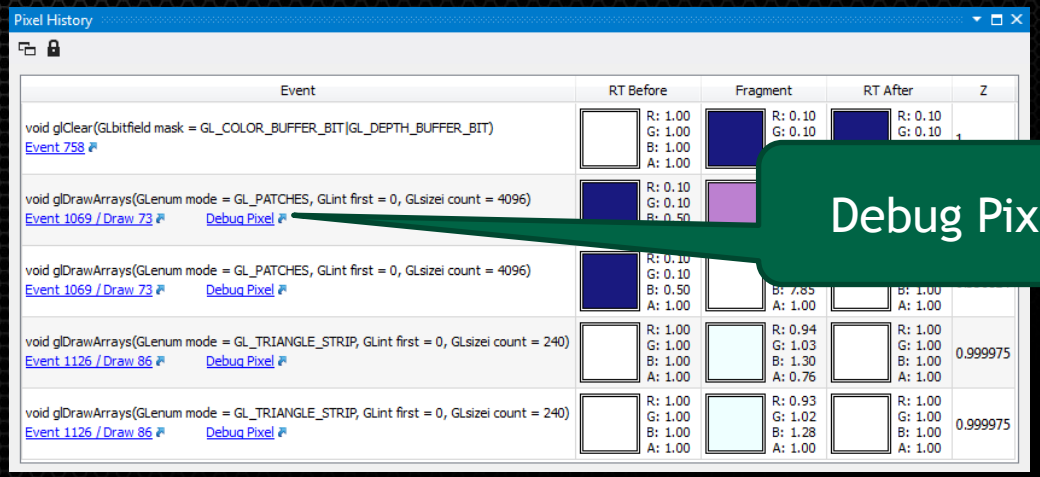
Select pixel to gather history on

Display list of contributing fragments

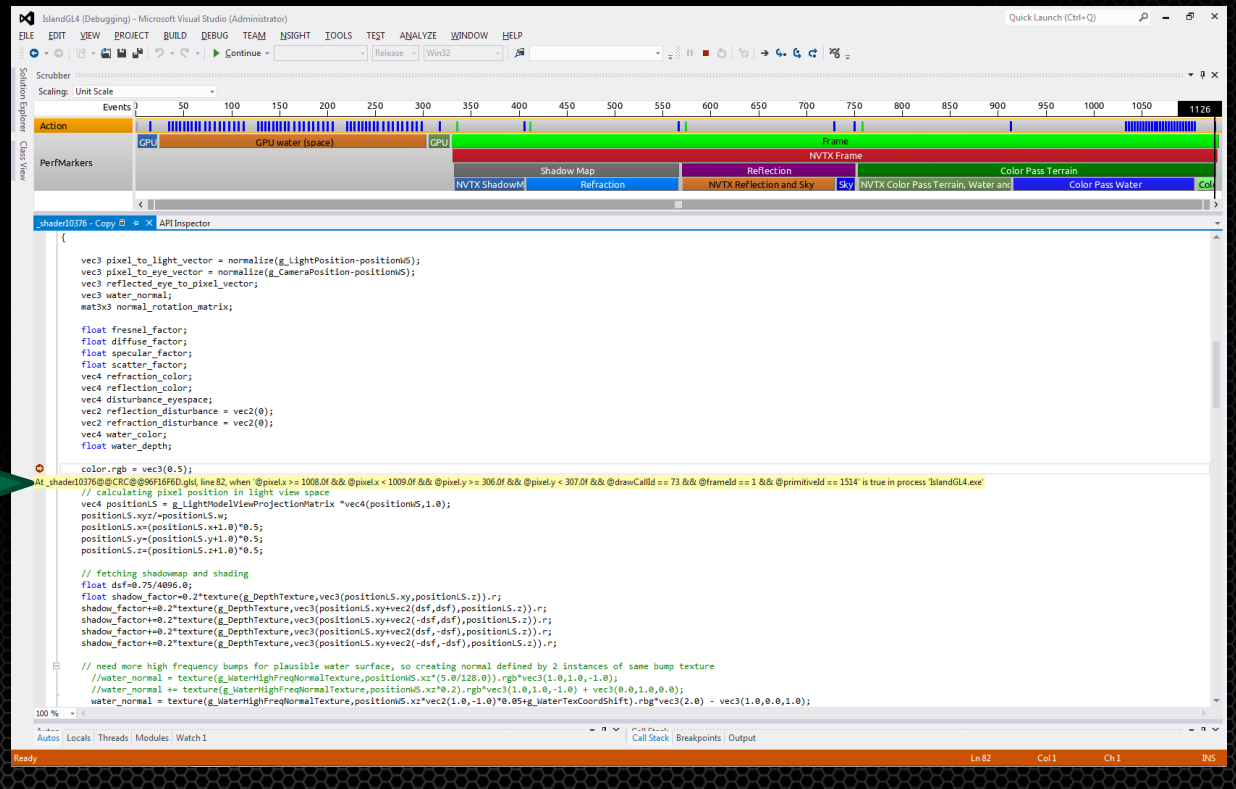
Pixel History

Event	RT Before	Fragment	RT After	Z
void glClear(GLbitfield mask = GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT) Event 758	R: 1.00 G: 1.00 B: 1.00 A: 1.00	R: 0.10 G: 0.10 B: 0.50 A: 1.00	R: 0.10 G: 0.10 B: 0.50 A: 1.00	1
void glDrawArrays(GLenum mode = GL_PATCHES, GLint first = 0, GLsizei count = 4096) Event 1069 / Draw 73 Debug Pixel	R: 0.10 G: 0.10 B: 0.50 A: 1.00	R: 0.74 G: 0.50 B: 0.81 A: 1.00	R: 1.00 G: 1.00 B: 1.00 A: 1.00	0.996814
void glDrawArrays(GLenum mode = GL_PATCHES, GLint first = 0, GLsizei count = 4096) Event 1069 / Draw 73 Debug Pixel	R: 0.10 G: 0.10 B: 0.50 A: 1.00	R: 7.78 G: 7.83 B: 7.85 A: 1.00	R: 1.00 G: 1.00 B: 1.00 A: 1.00	0.996814
void glDrawArrays(GLenum mode = GL_TRIANGLE_STRIP, GLint first = 0, GLsizei count = 240) Event 1126 / Draw 86 Debug Pixel	R: 1.00 G: 1.00 B: 1.00 A: 1.00	R: 0.94 G: 1.03 B: 1.30 A: 0.76	R: 1.00 G: 1.00 B: 1.00 A: 1.00	0.999975
void glDrawArrays(GLenum mode = GL_TRIANGLE_STRIP, GLint first = 0, GLsizei count = 240) Event 1126 / Draw 86 Debug Pixel	R: 1.00 G: 1.00 B: 1.00 A: 1.00	R: 0.93 G: 1.02 B: 1.28 A: 1.00	R: 1.00 G: 1.00 B: 1.00 A: 1.00	0.999975

Demo: Debugging IslandsGL



Debug Pixel



Conditional breakpoint to inspect fragment's shader instance

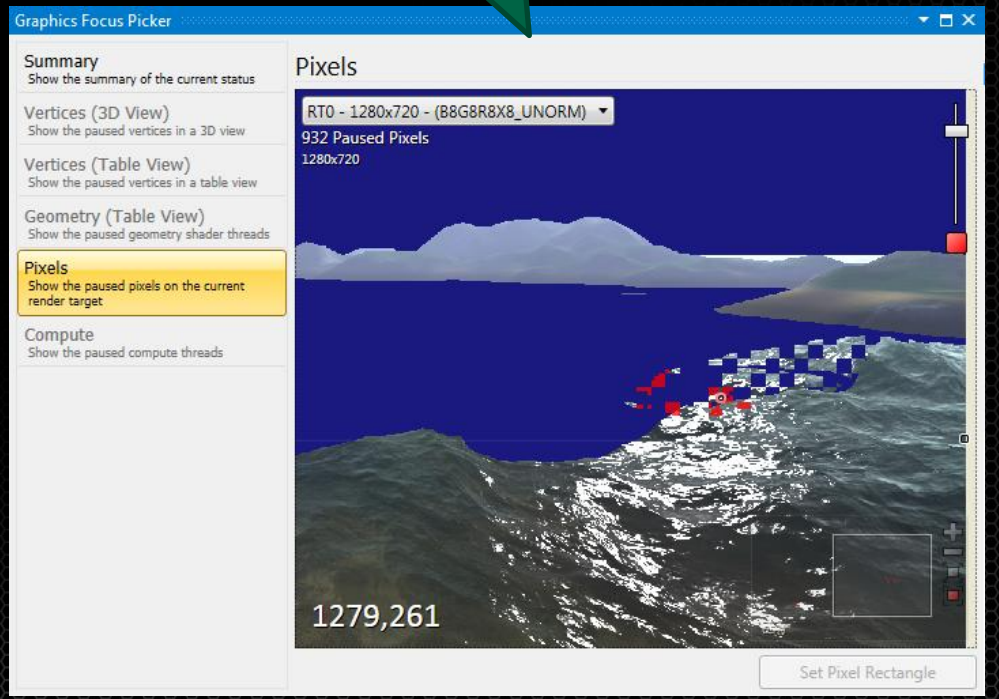
Demo: Debugging IslandsGL

Locals

Name	Value	Type
pixel_to_light_vector	{x = -0.65988165, y = 0.41504461, z = 0.62633395}	vec3
pixel_to_eye_vector	{x = 0.76135629, y = 0.21701783, z = -0.6109336}	vec3
reflected_eye_to_pixel_vector	{x = -0.64161855, y = 0.46083534, z = 0.6131528}	vec3
water_normal	{x = 0.17394866, y = 0.98474944, z = 0.0032239738}	vec3
normal_rotation_matrix	{{0.9866547, 0.16271445, 0}, {-0.16280109, 0.98582, 0.017720262}, {-0.0028853309, 0.017471733, -	mat3x3
fresnel_factor	0.20285697	float
diffuse_factor	0.12959486	float
specular_factor	0.10574586	float
scatter_factor	0.018670255	float
refraction_color	{x = 0.11210445, y = 0.12237978, z = 0.15450107, w = 1}	vec4
reflection_color	{x = 0.87772948, y = 1, z = 1, w = 1}	vec4
disturbance_eyespace	{x = -0.16393672, y = -0.0098230857, z = 0.057417676, w = 0}	vec4
reflection_disturbance	{x = -0.0081968363, y = -0.026925821}	vec2
refraction_disturbance	{x = -0.015622481, y = -0.0009360988}	vec2
water_color	{x = 0.10927379, y = 0.11950034, z = 0.15134811, w = 0}	vec4
water_depth	2.147438	float

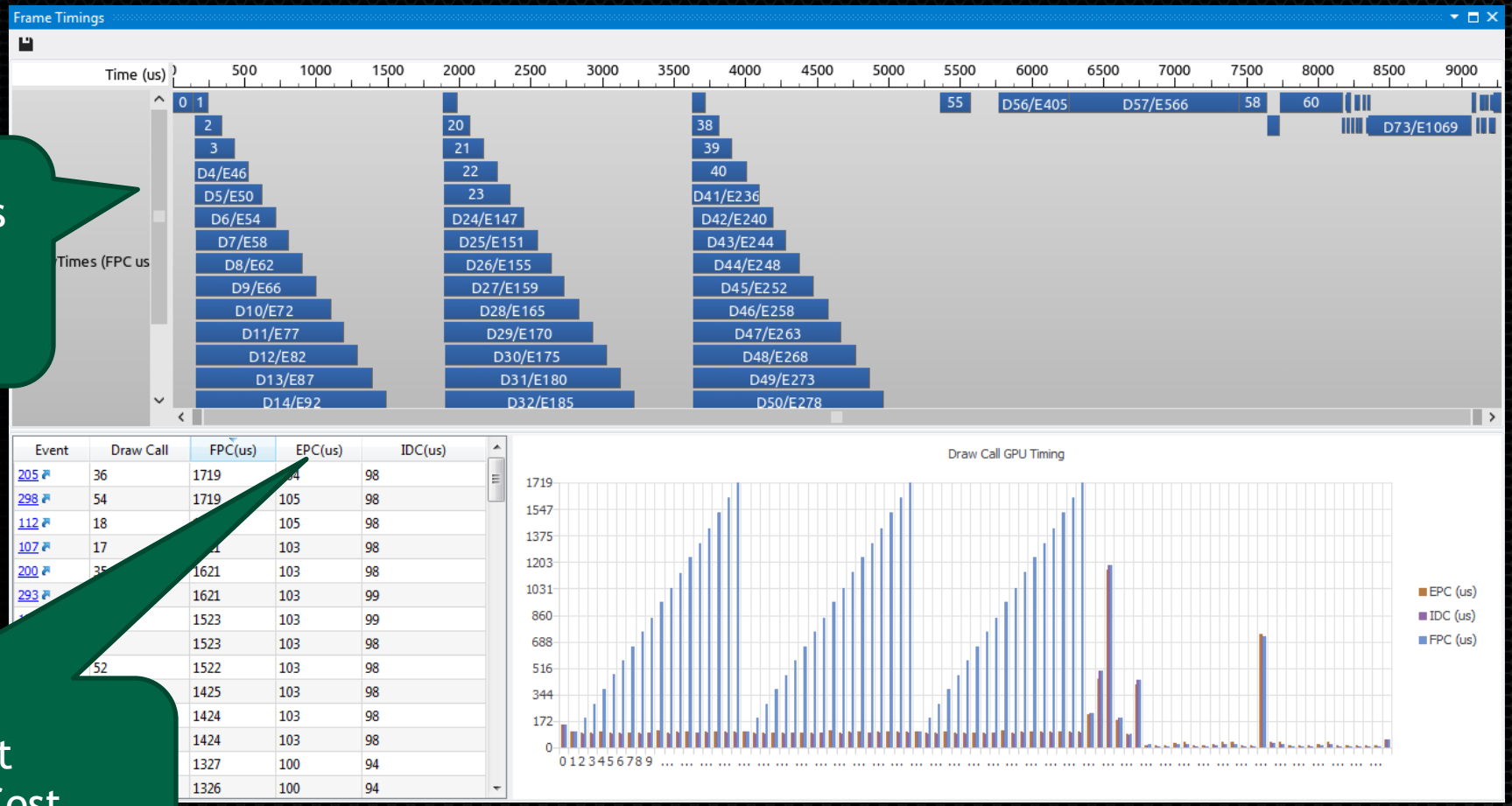
Shader debugging – standard VS windows work as expected

Nsight focus picker gives context for the graphics threads



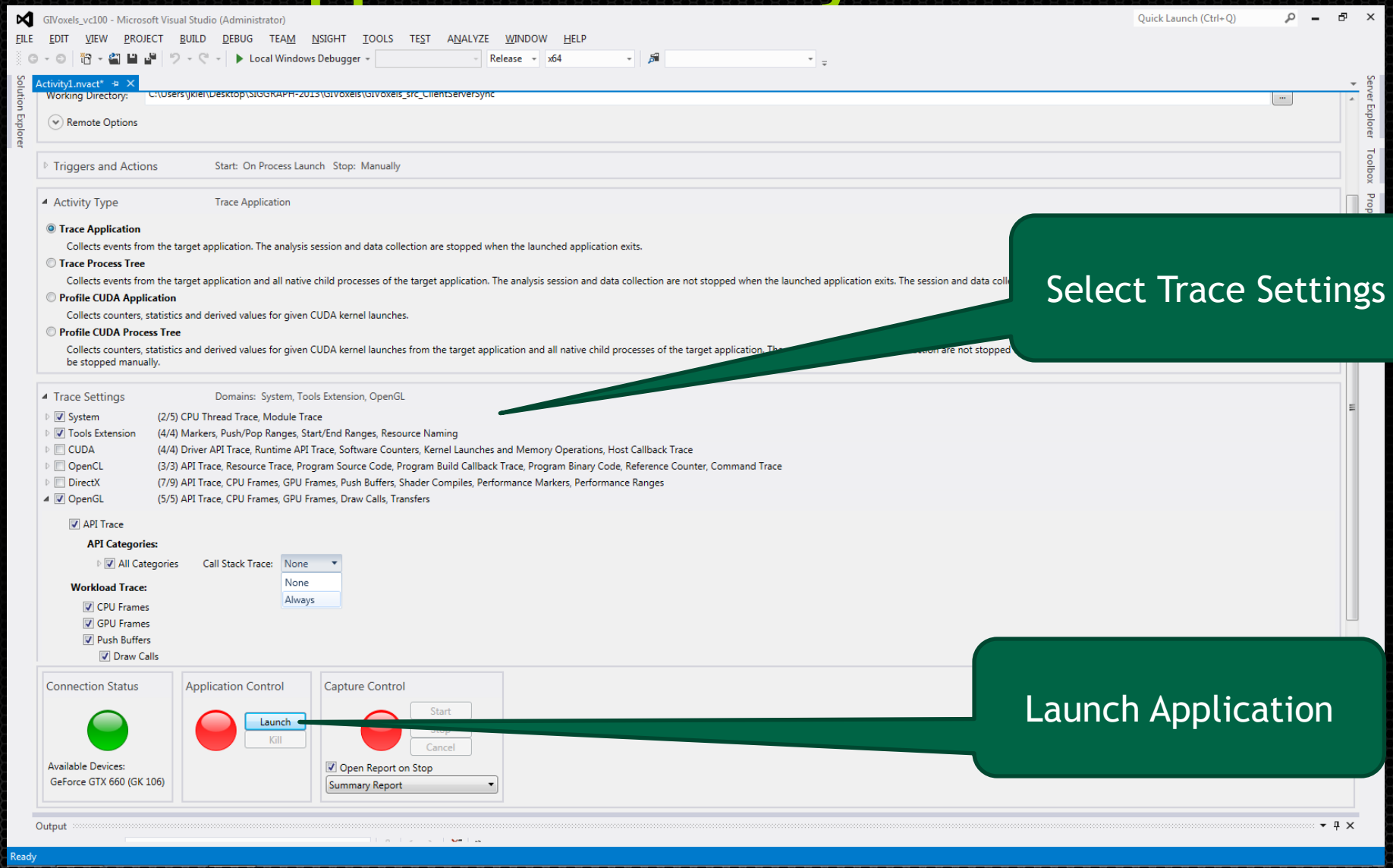
Demo: Debugging IslandsGL

Frame Timings gives quick performance feedback

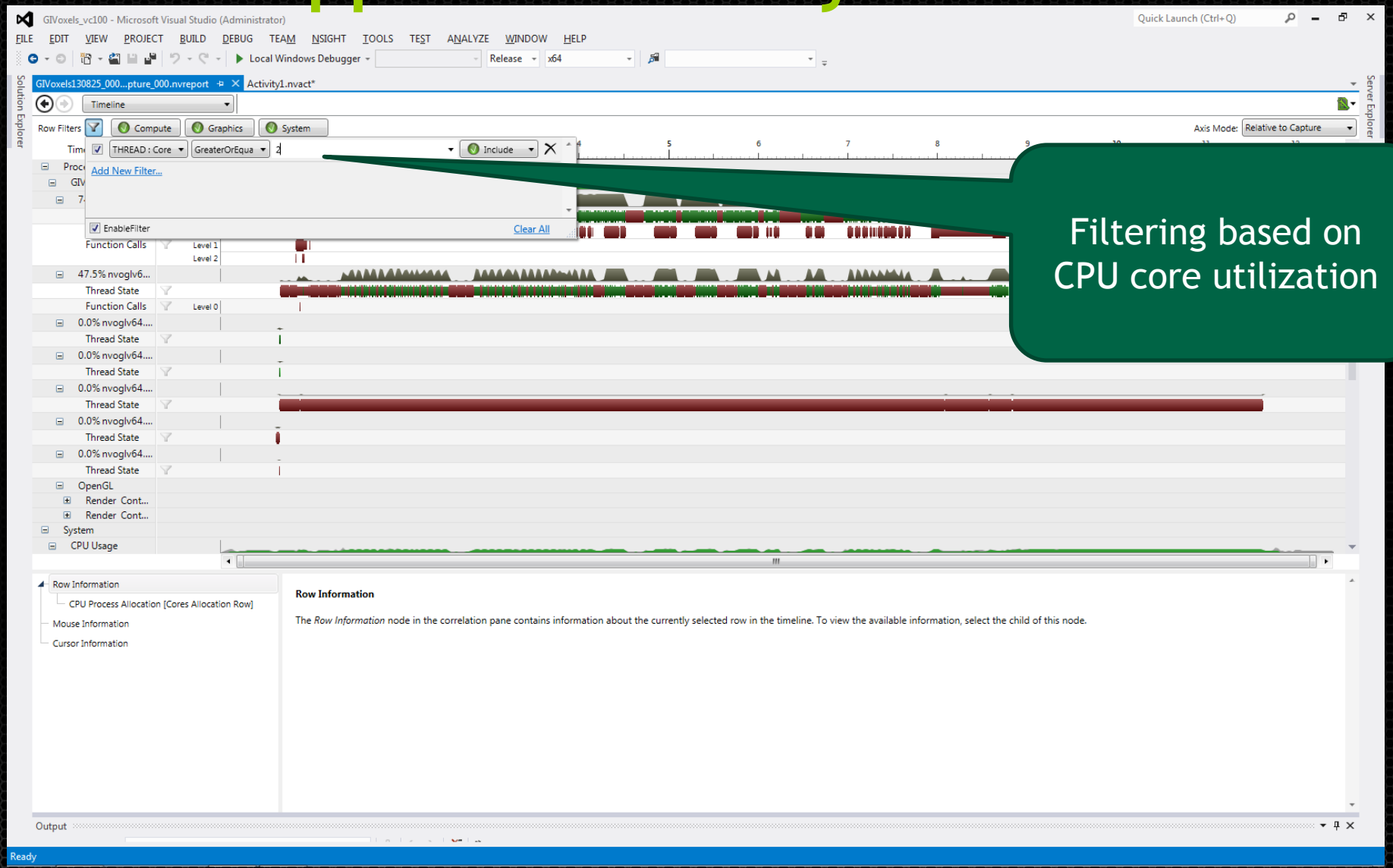


FPC: Full Pipeline Cost
EPC: Empty Pipeline Cost
IDC: Incremental Draw Cost

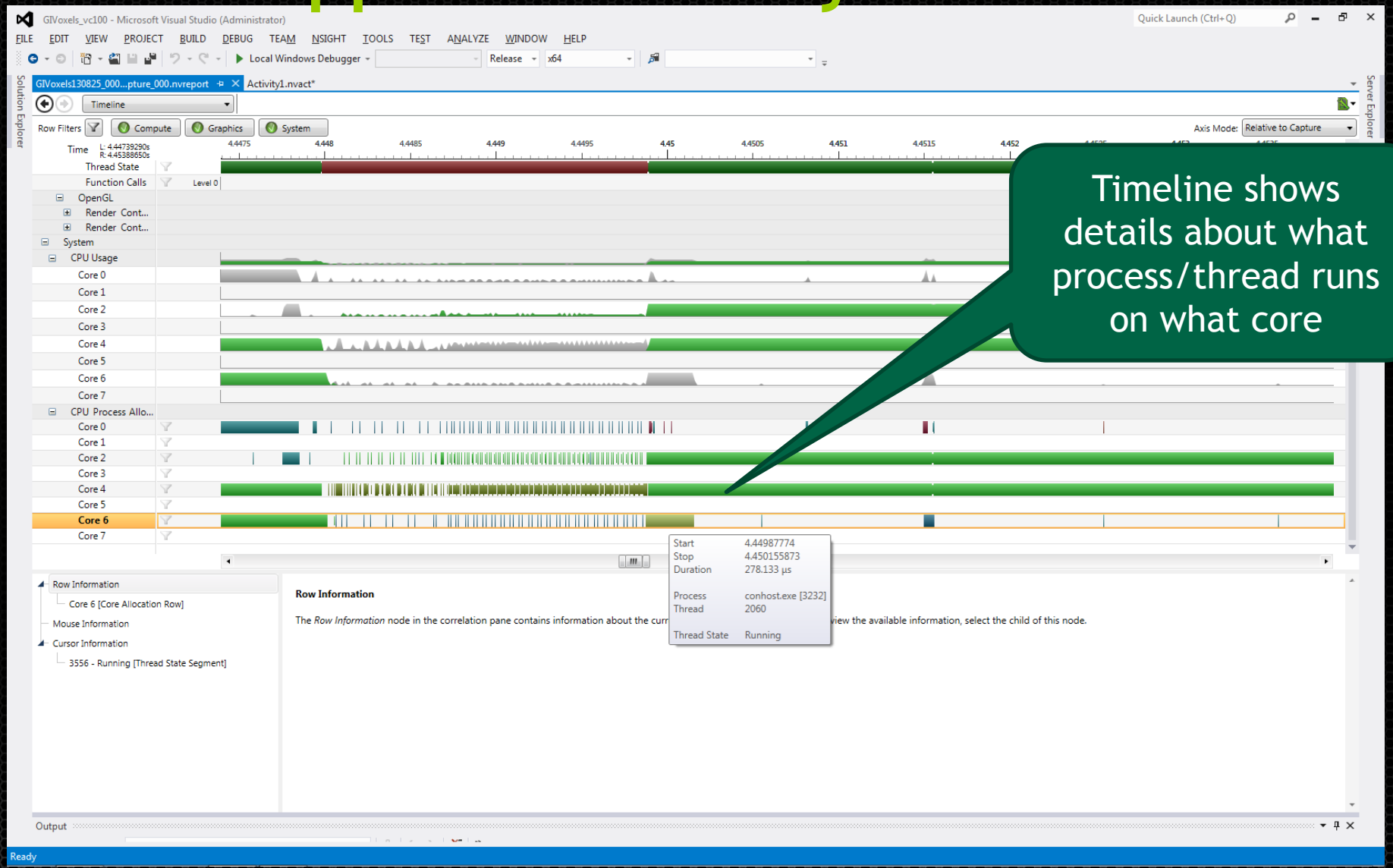
Demo: Application Analysis GIVoxels



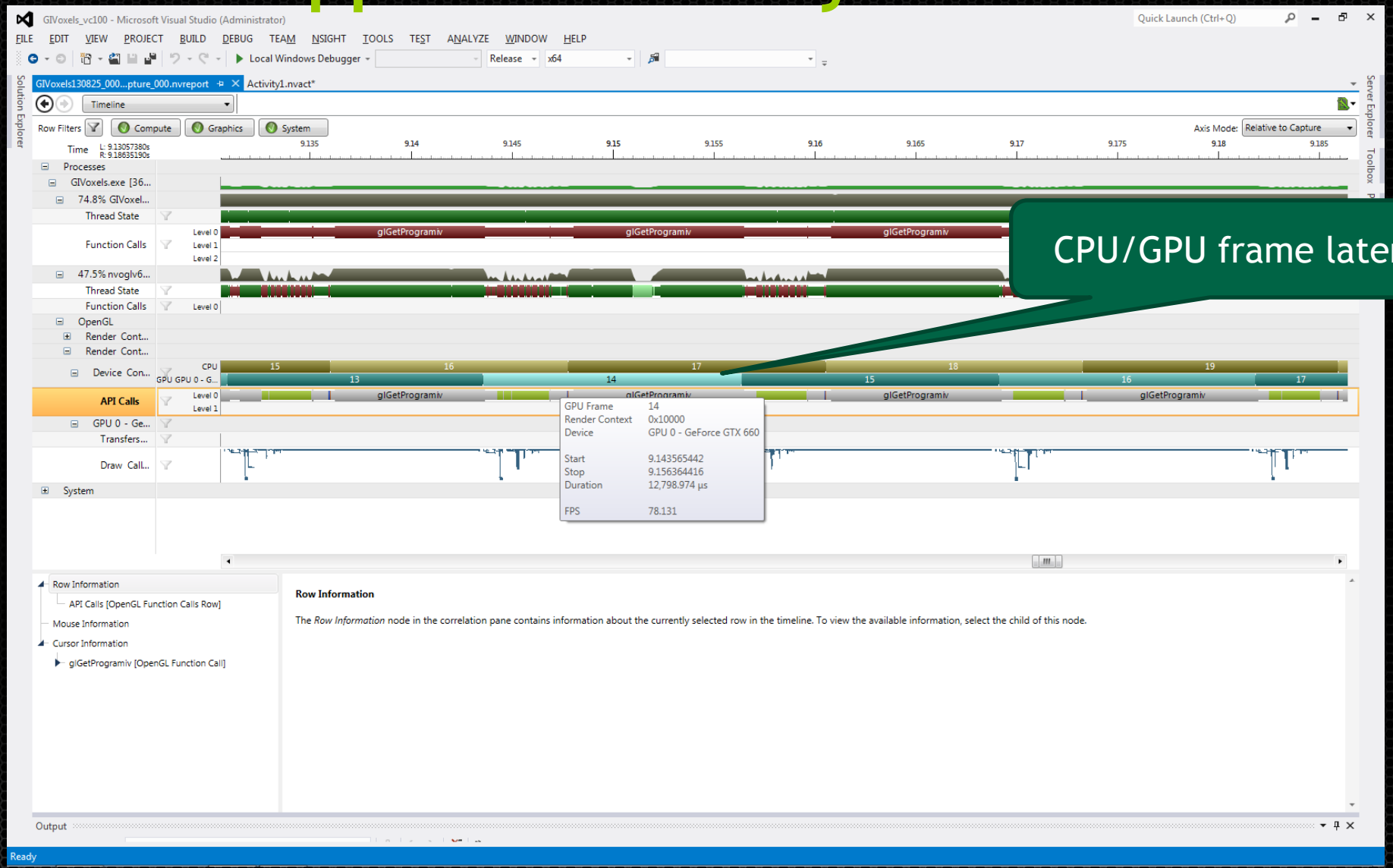
Demo: Application Analysis GIVoxels



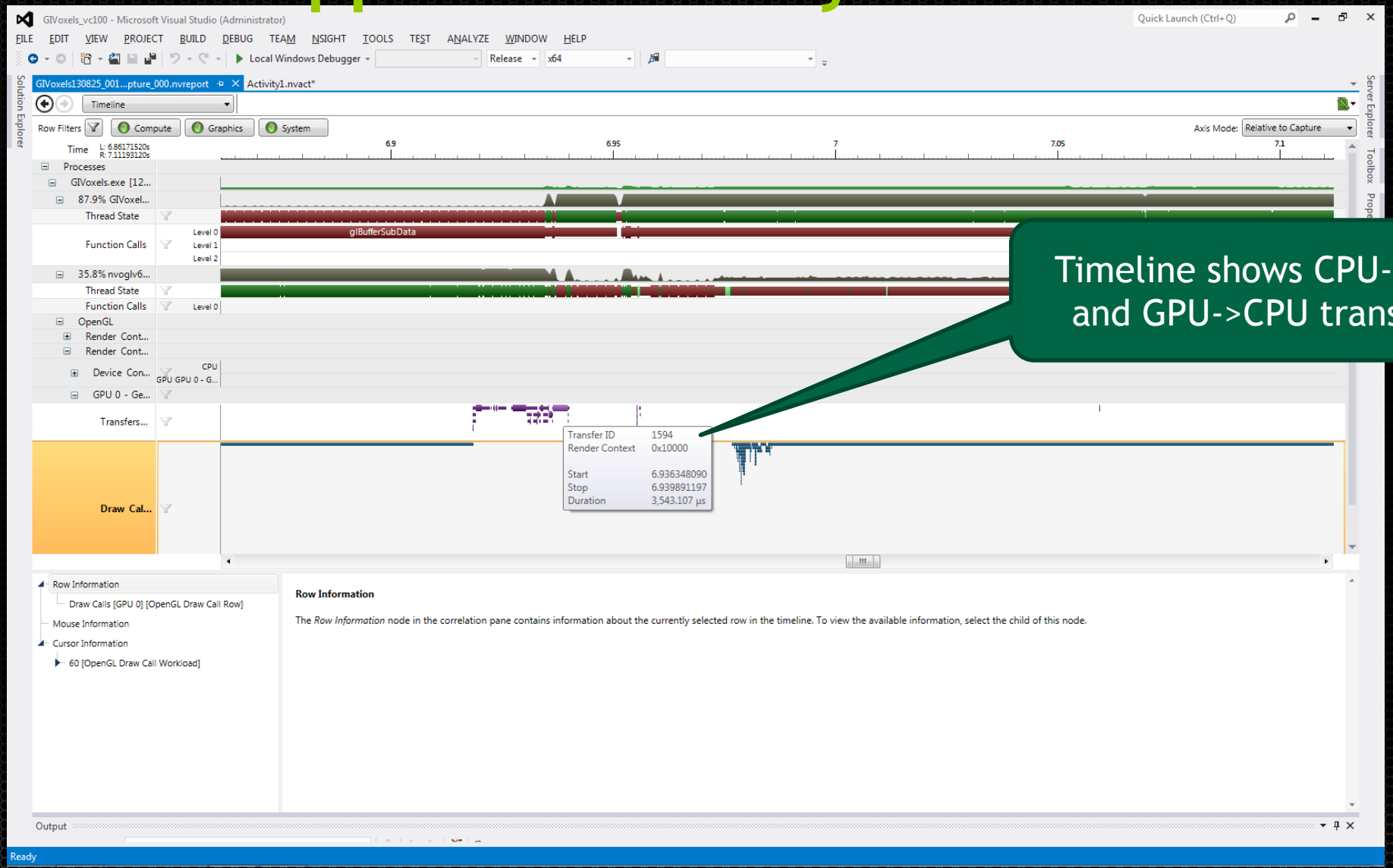
Demo: Application Analysis GIVoxels



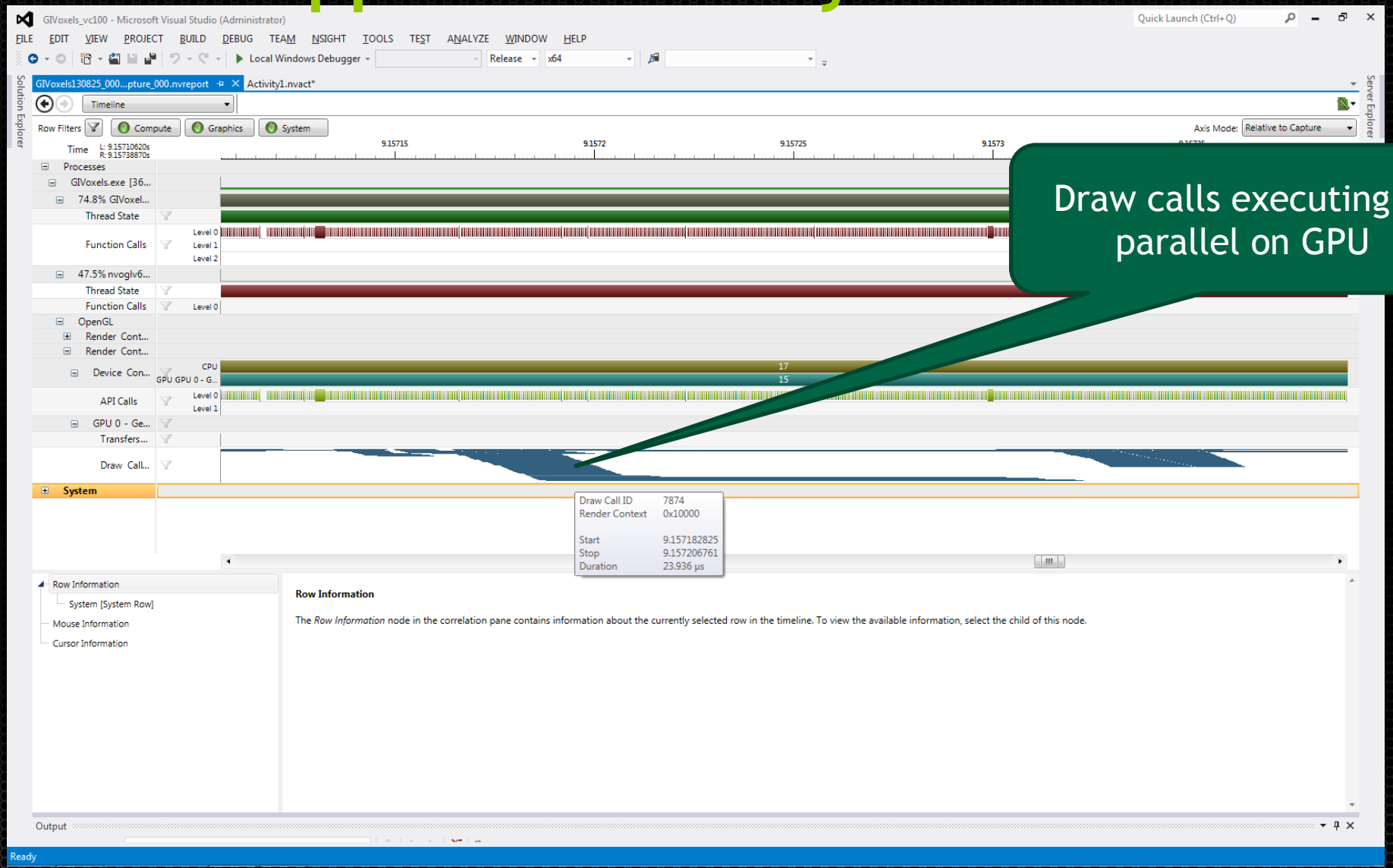
Demo: Application Analysis GIVoxels



Demo: Application Analysis GIVoxels

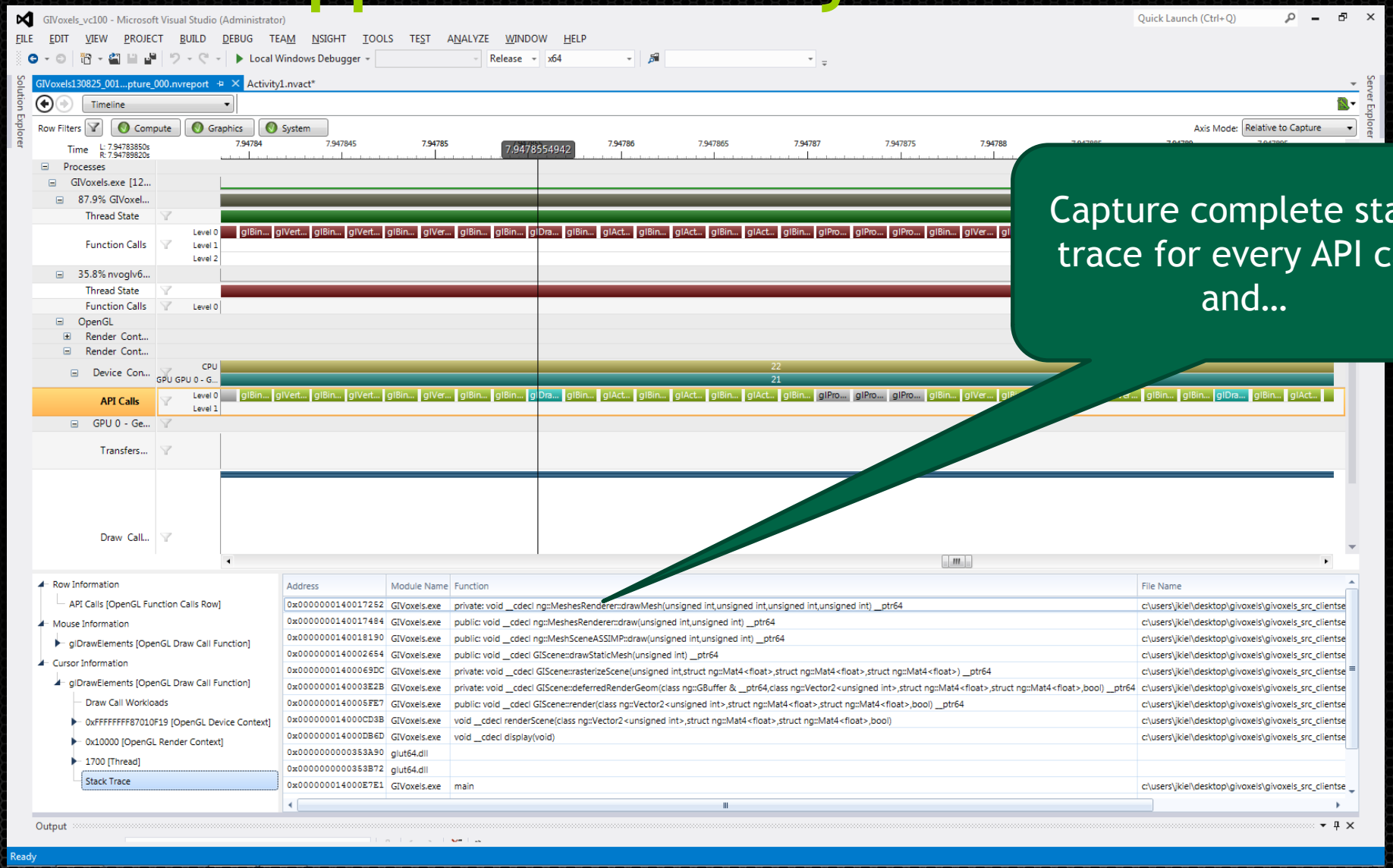


Demo: Application Analysis GIVoxels

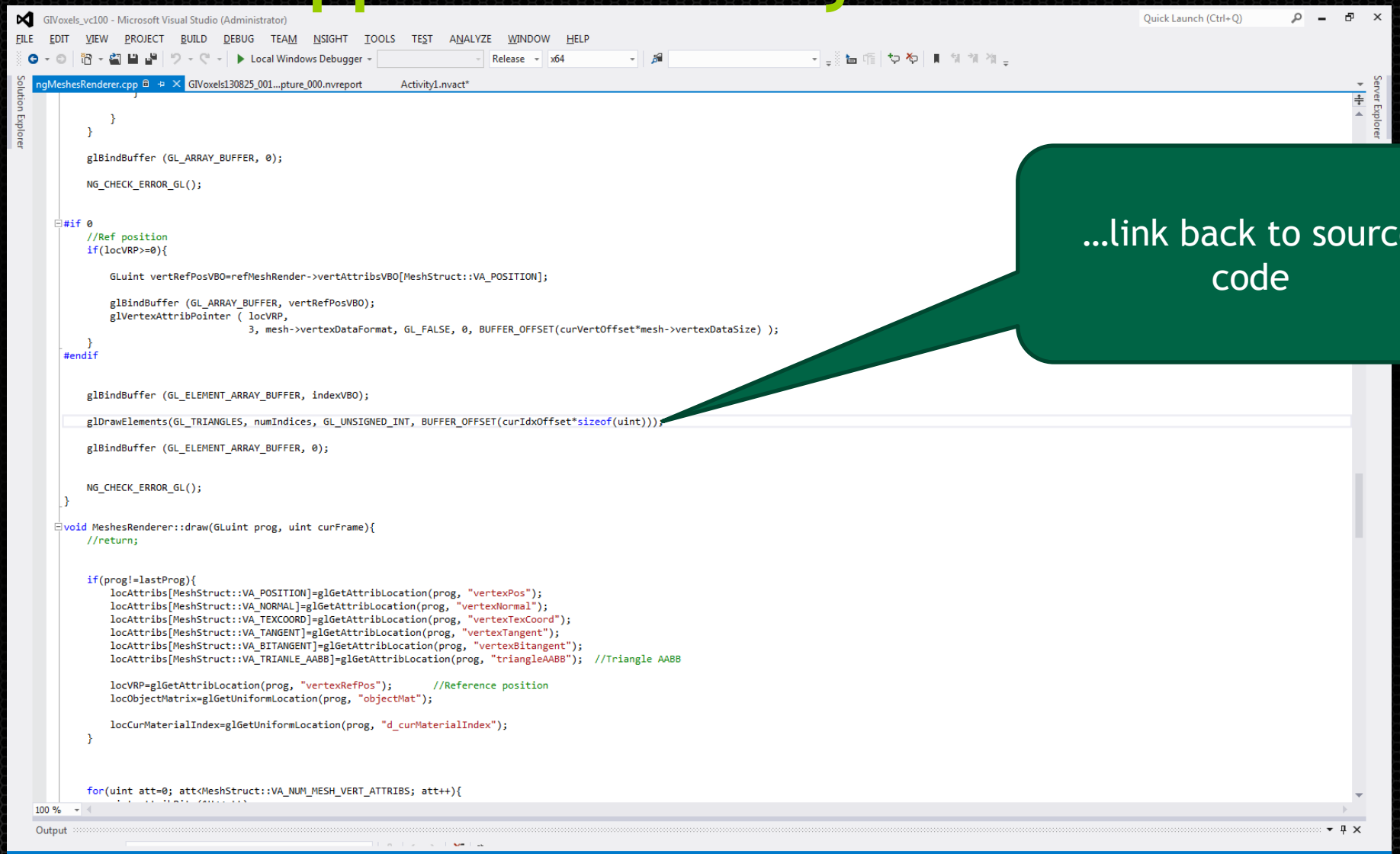


Draw calls executing in parallel on GPU

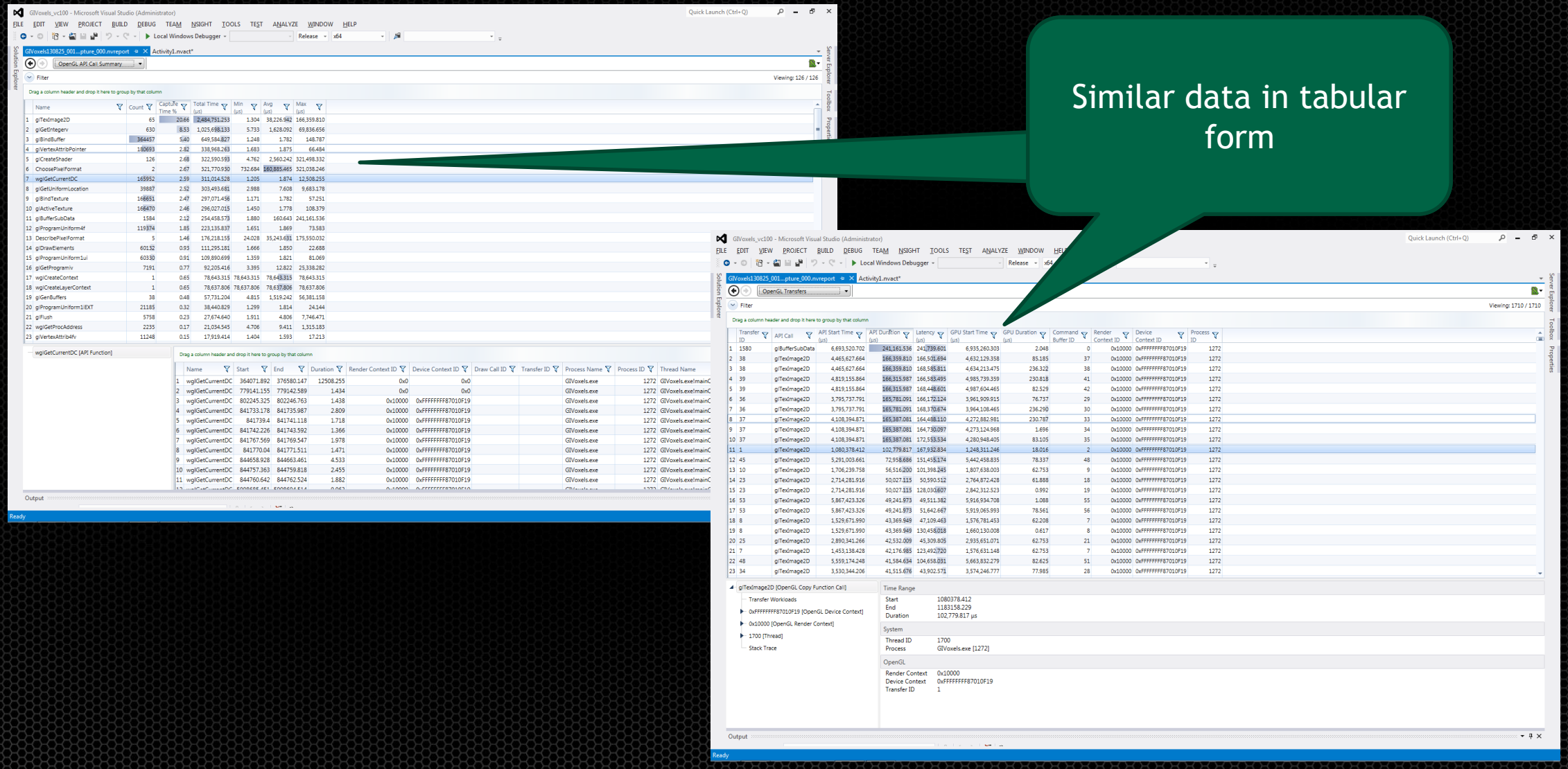
Demo: Application Analysis GIVoxels



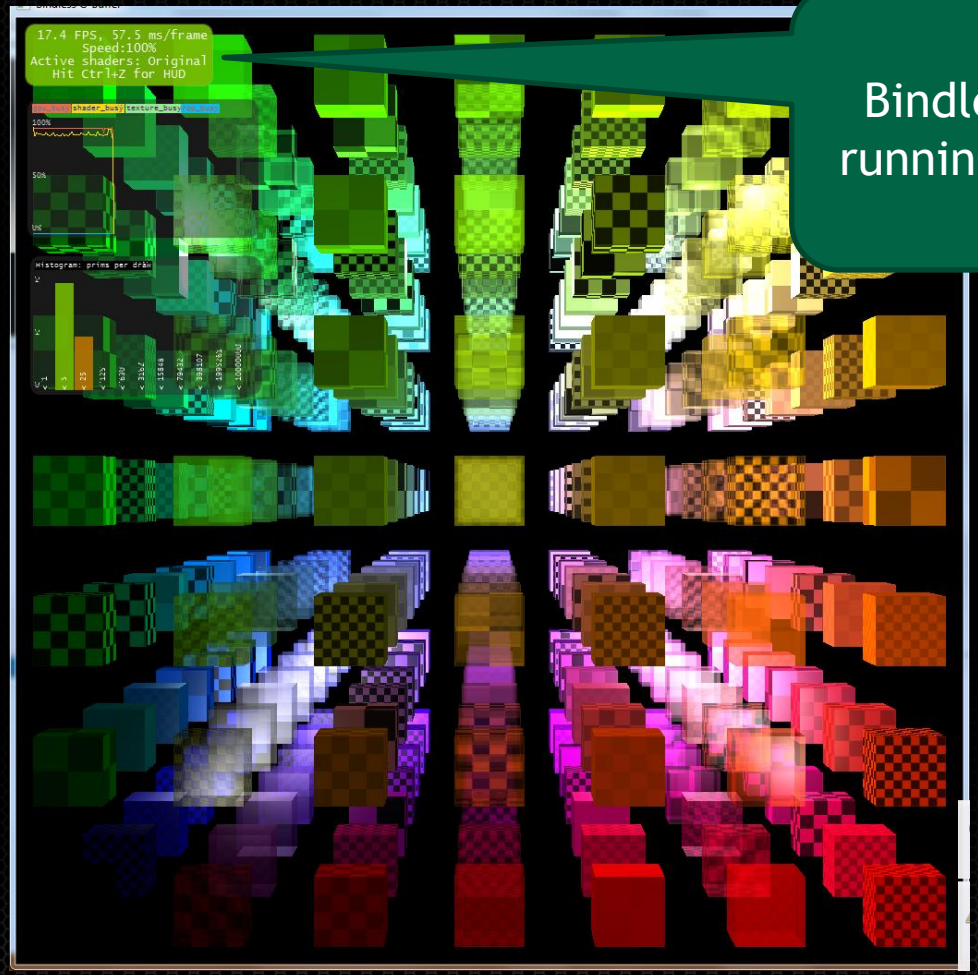
Demo: Application Analysis GIVoxels



Demo: Application Analysis GIVoxels

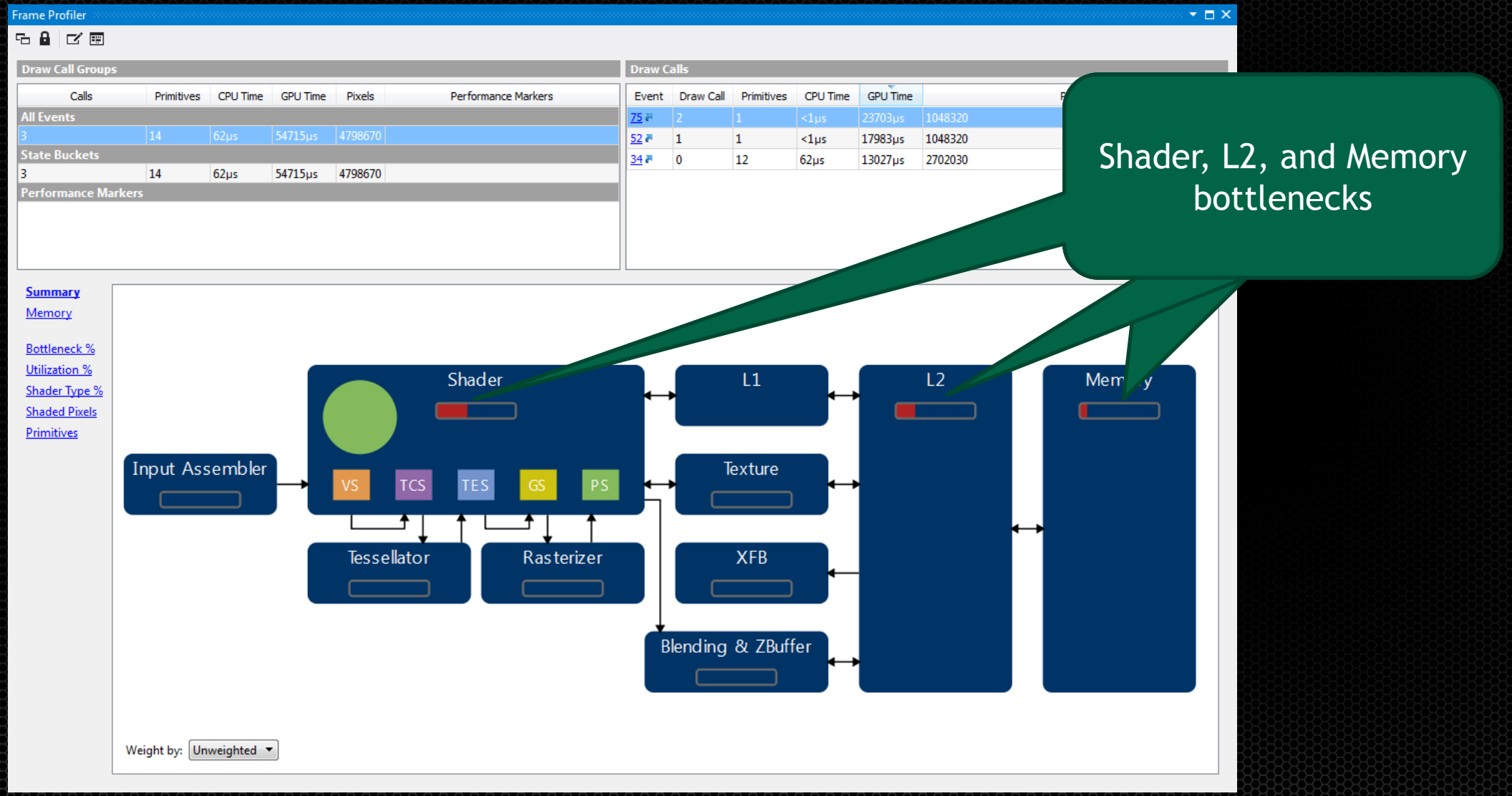


Demo: Performance Bindless GBuffer



Bindless Graphics app running at 17FPS before

Demo: Performance Bindless GBuffer



Demo: Performance Bindless GBuffer

Frame Profiler

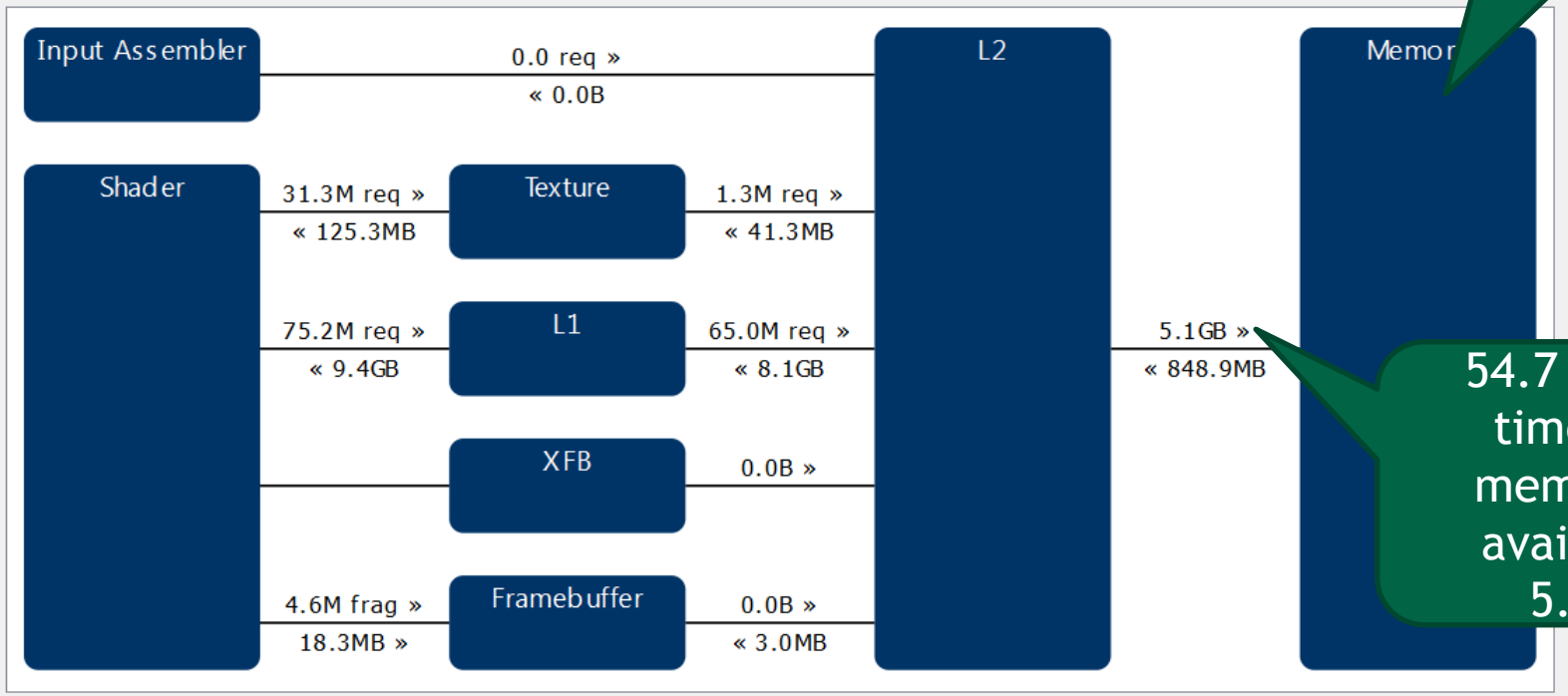
Draw Call Groups

Calls	Primitives	CPU Time	GPU Time	Pixels	Performance Markers
All Events					
3	14	62µs	54715µs	4798670	
State Buckets					
3	14	62µs	54715µs	4798670	
Performance Markers					

Draw Calls

Event	Draw Call	Primitives	CPU Time	GPU Time	Pixels
75	2	1	<1µs	23703µs	1048320
52	1	1	<1µs	17983µs	1048320
34	0	12	62µs	13027µs	2702030

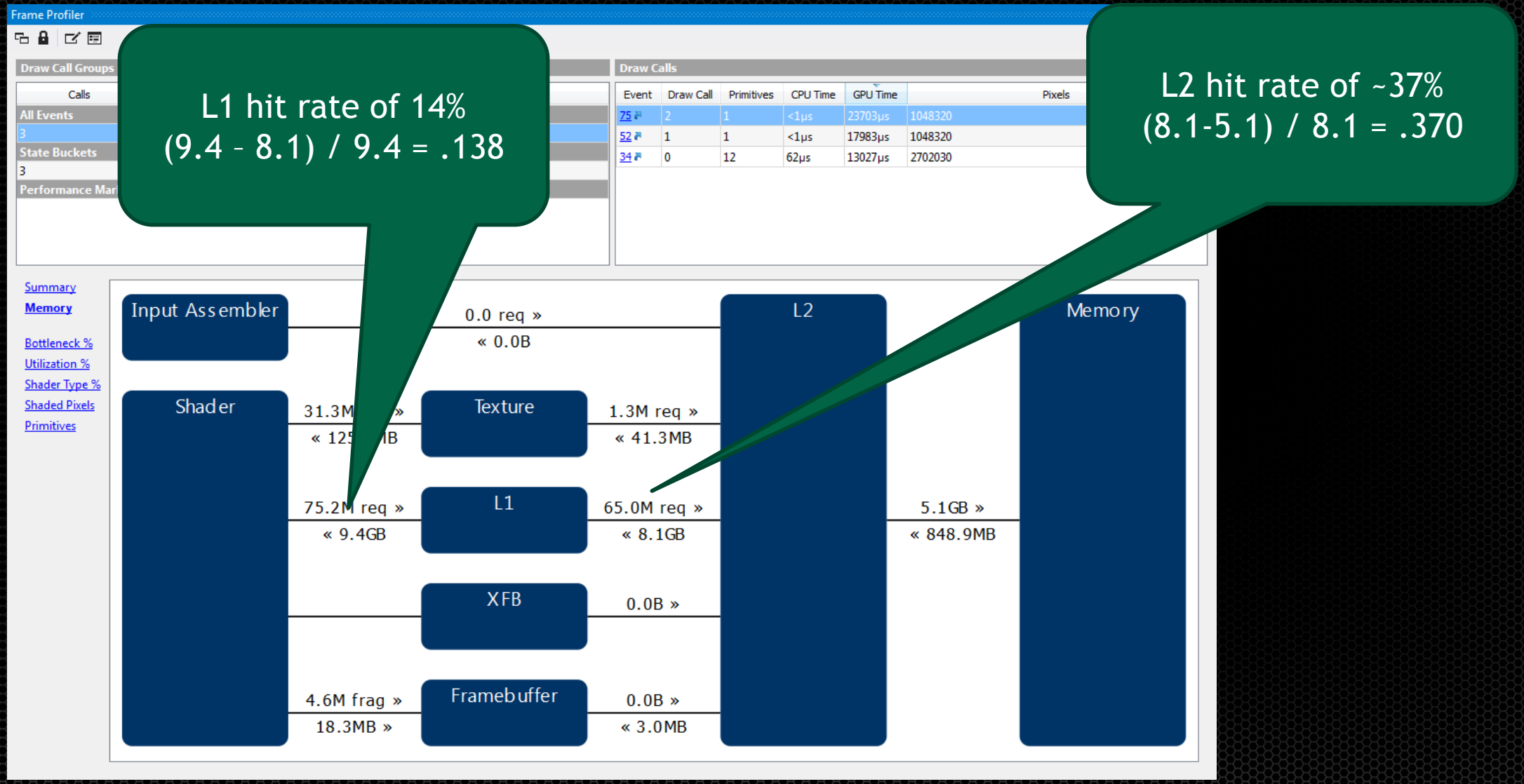
Summary
Memory
Bottleneck %
Utilization %
Shader Type %
Shaded Pixels
Primitives



Running on GeForce GTX660 which has 144.2 GB/s of memory bandwith

54.7 ms total kernel time = 7.89 GB of memory bandwidth available, we used 5.95GB or 75%

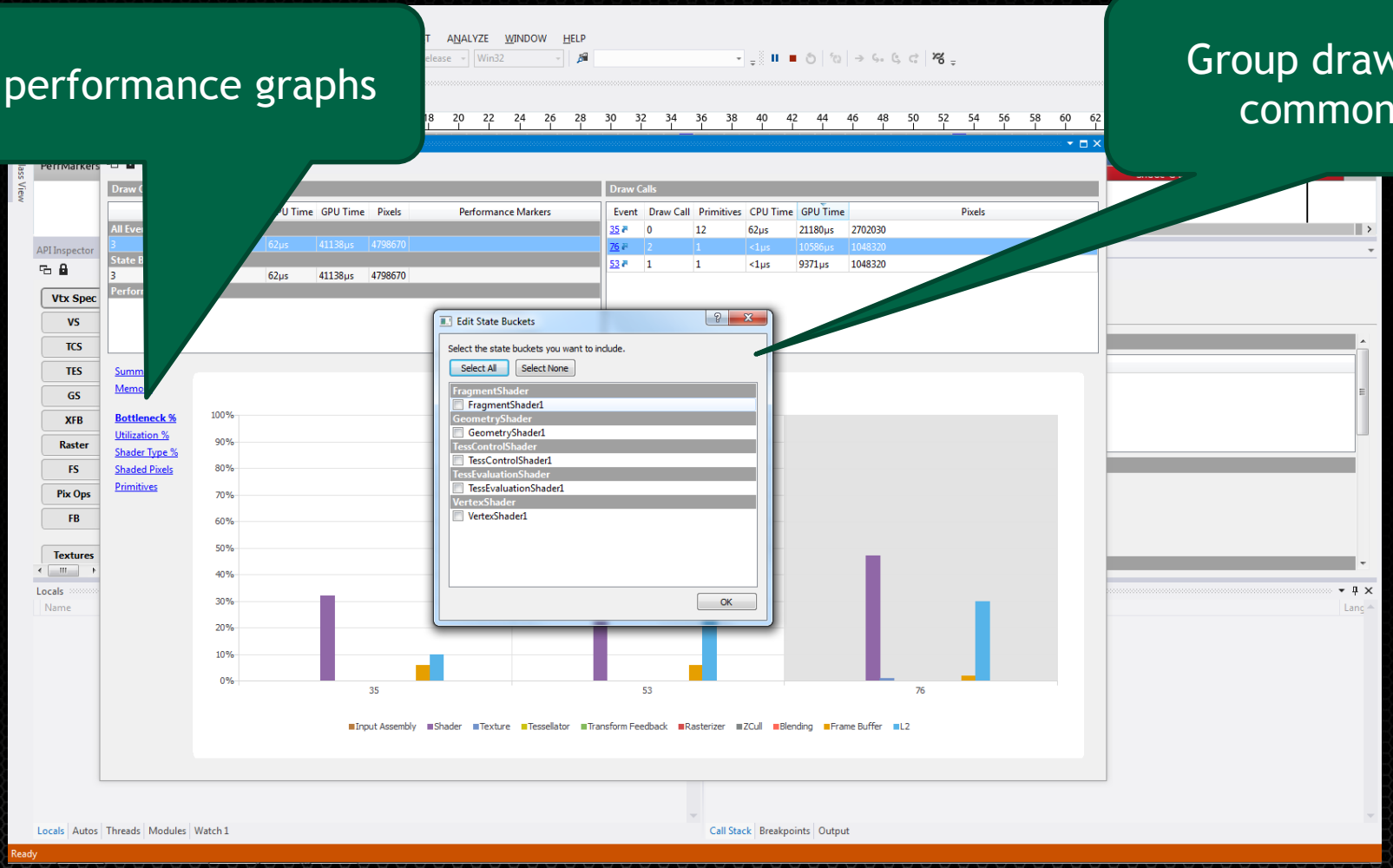
Demo: Performance Bindless GBuffer



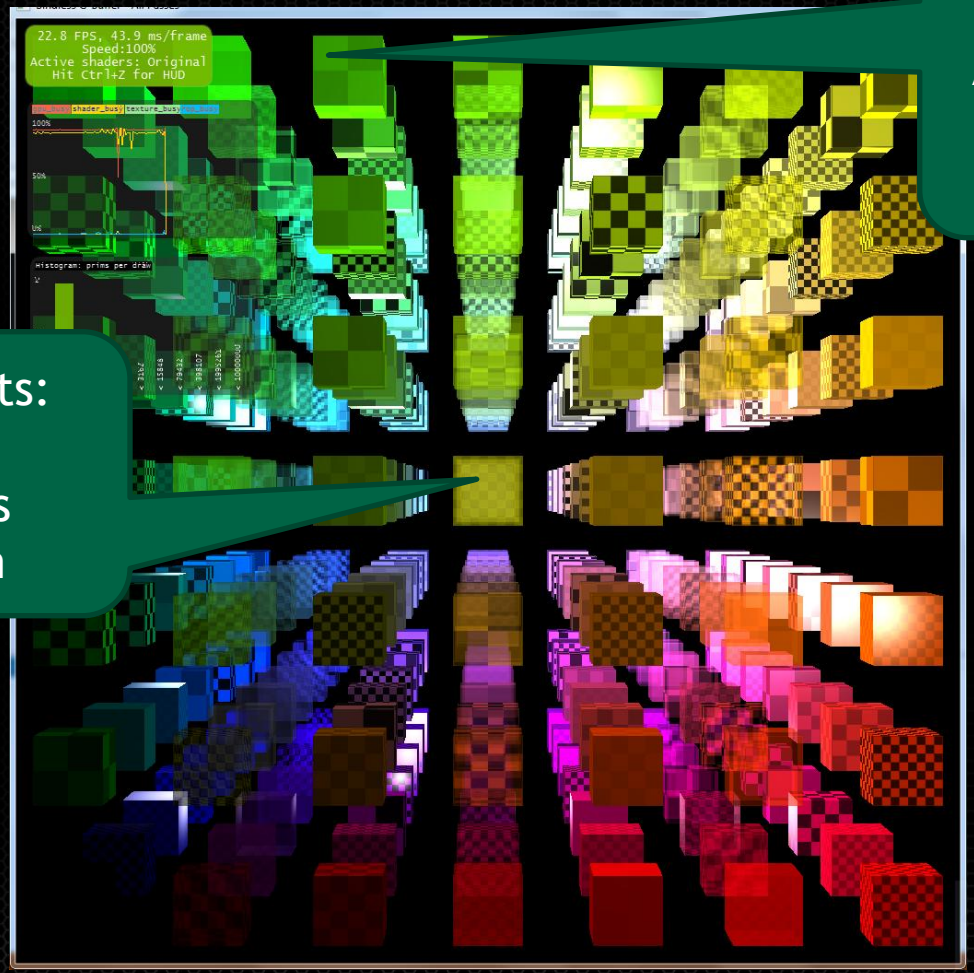
Demo: Performance Bindless GBuffer

Additional performance graphs

Group draw calls by common state



Demo: Performance Bindless GBuffer

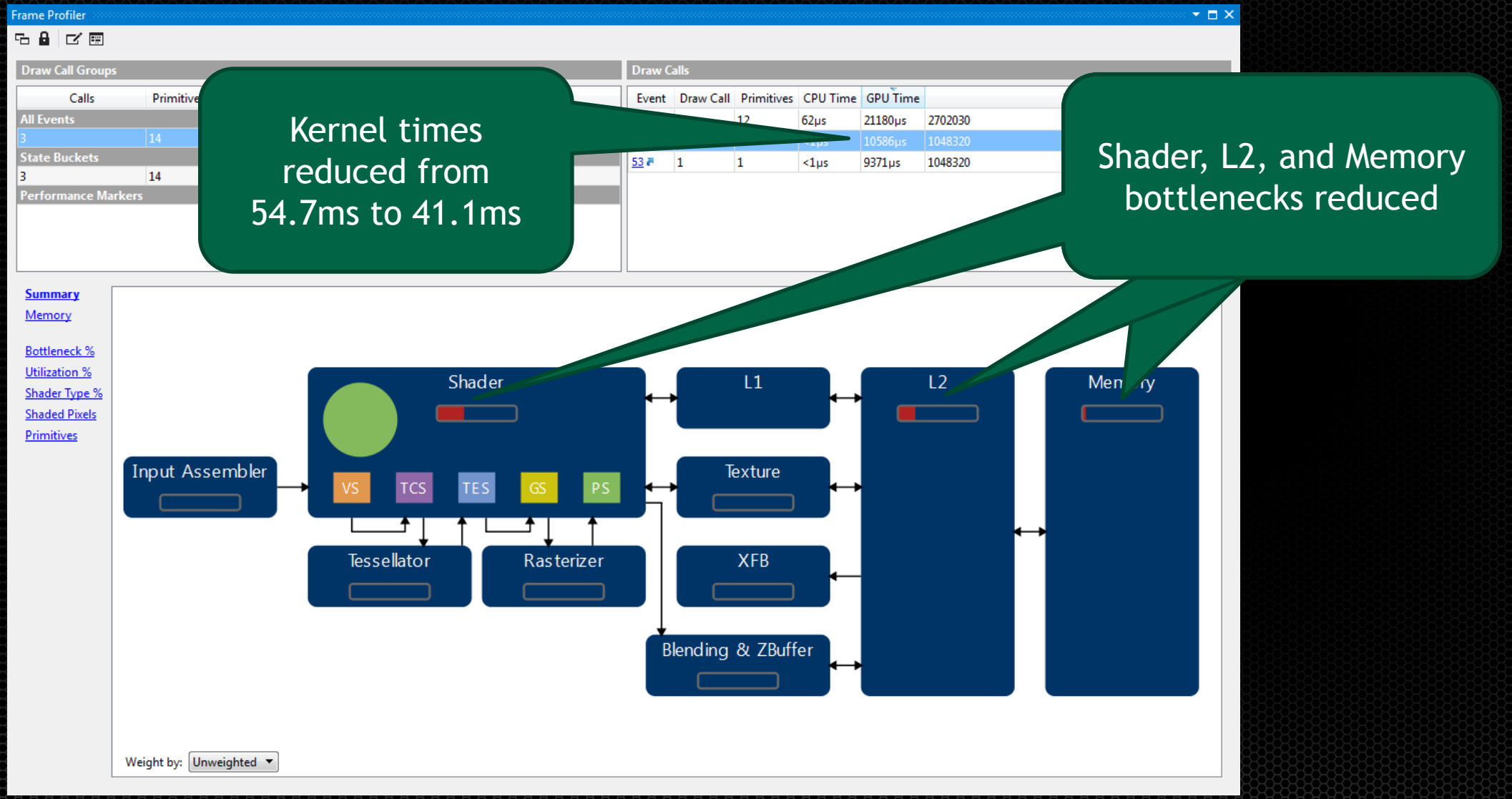


After optimizations running at ~23 FPS

Performance improvements:

1) Multiple tiled allocators
2) Localized memory access
3) Simplified spill algorithm

Demo: Performance Bindless GBuffer



Demo: Performance Bindless GBuffer

Frame Profiler

Draw Call Groups

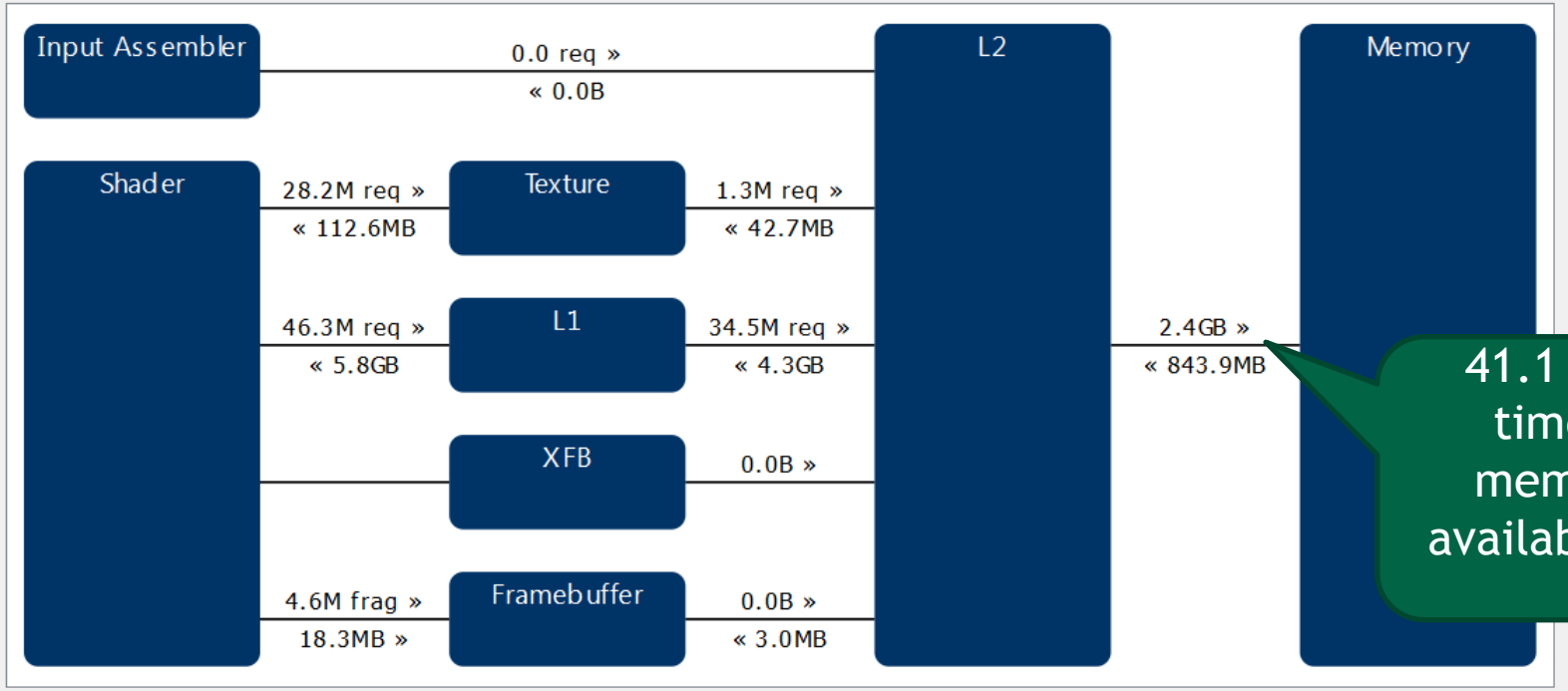
Calls	Primitives	CPU Time	GPU Time	Pixels	Performance Markers
All Events					
3	14	62µs	41138µs	4798670	
State Buckets					
3	14	62µs	41138µs	4798670	
Performance Markers					

Draw Calls

Event	Draw Call	Primitives	CPU Time	GPU Time	Pixels
35	0	12	62µs	21180µs	2702030
76	2	1	<1µs	10586µs	1048320
53	1	1	<1µs	9371µs	1048320

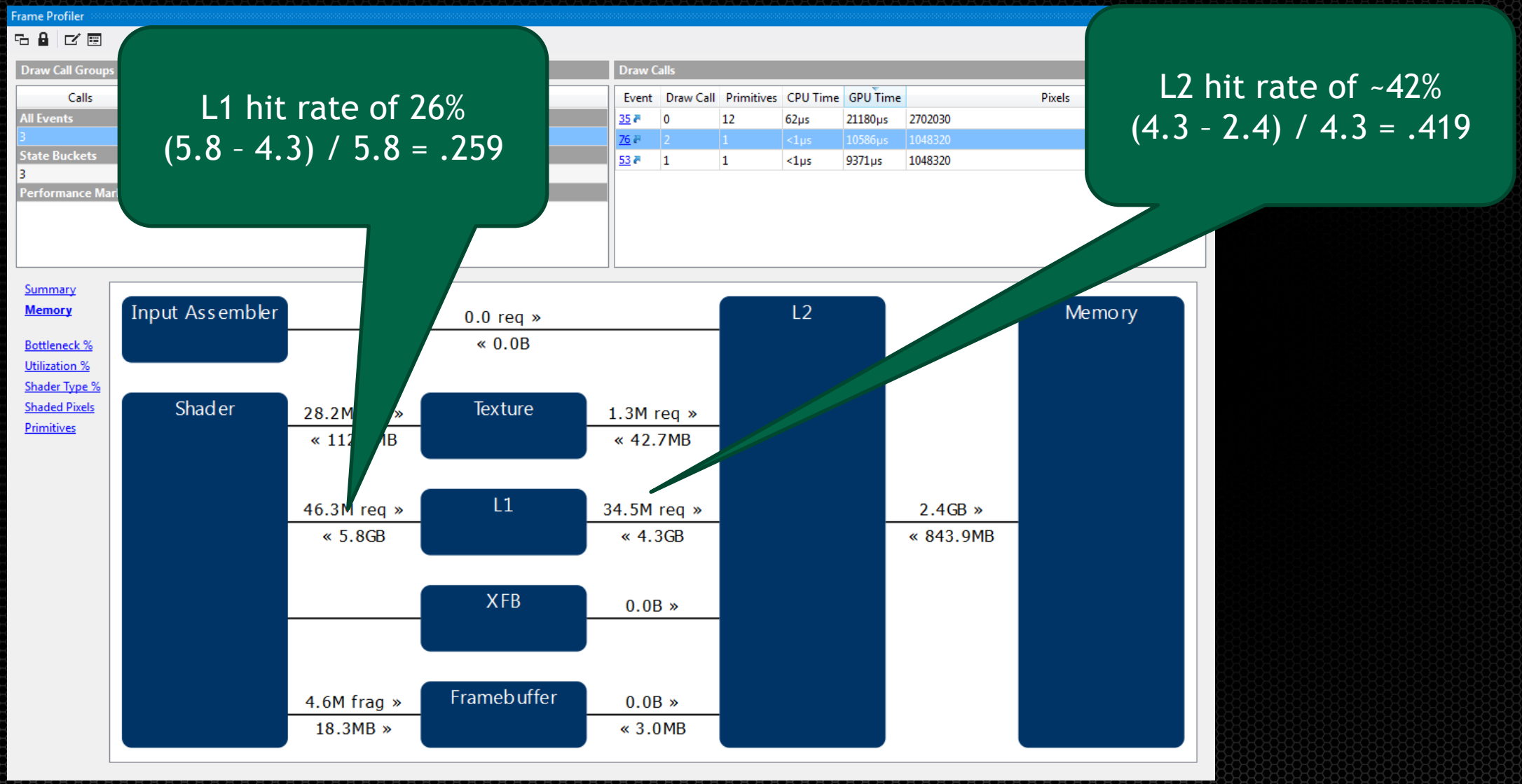
Summary
Memory

Bottleneck %
Utilization %
Shader Type %
Shaded Pixels
Primitives



41.1 ms total kernel time = 5.93 GB of memory bandwidth available, we used 3.24 GB or 55%

Demo: Performance Bindless GBuffer



NVIDIA® Nsight™ Visual Studio Edition

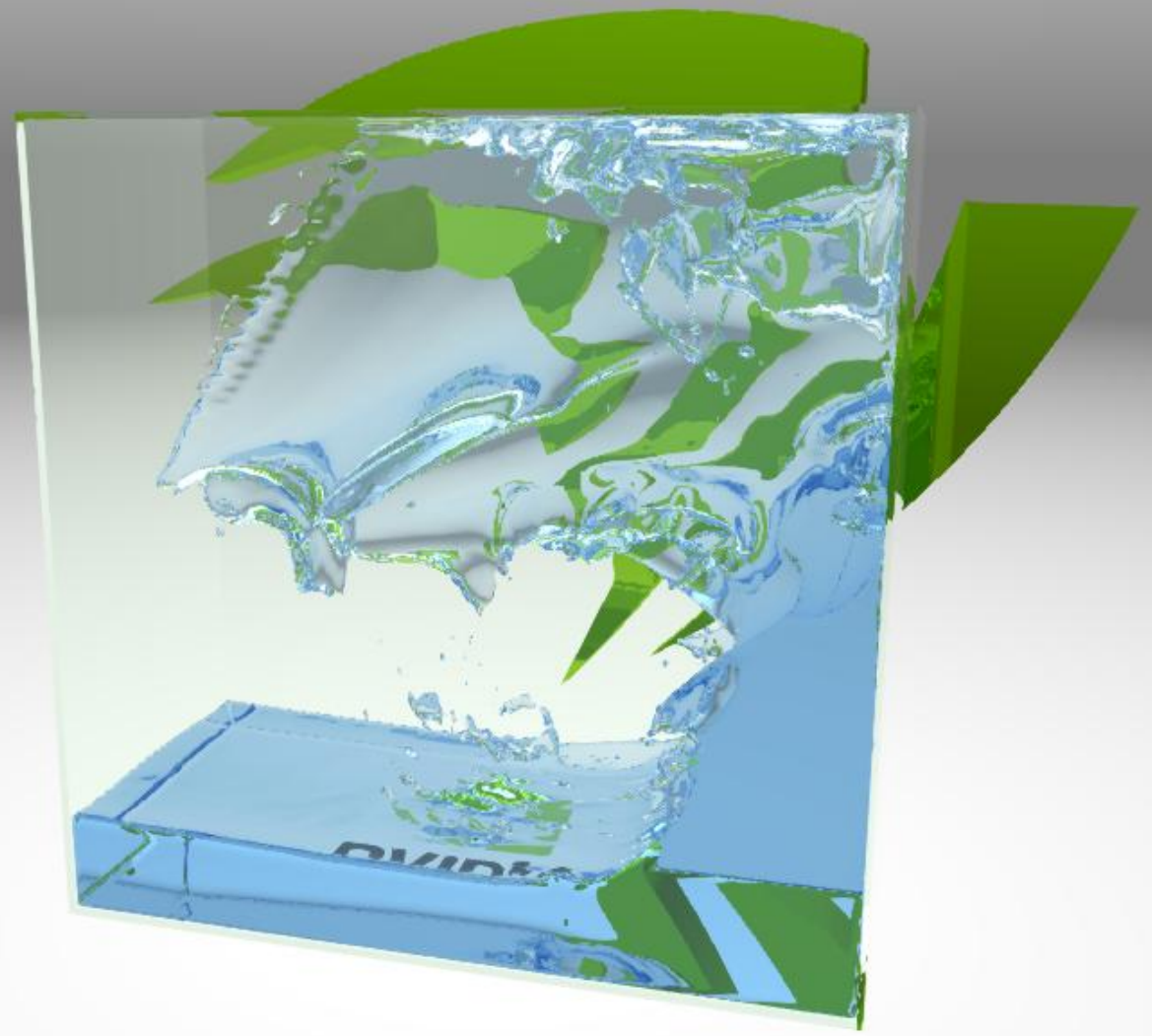
- Download
  - http://www.nvidia.com/nsight
- Developer Forums
  - https://devtalk.nvidia.com/default/board/84/nsight-visual-studio-edition
- Documentation
  - Start → All Programs → NVIDIA Nsight Visual Studio Edition 3.0 → User Guide
  - http://http.developer.nvidia.com/NsightVisualStudio/3.0/Documentation/UserGuide/HTML/Nsight_Visual_Studio_Edition_User_Guide.htm
- Instructional videos
  - http://www.gputechconf.com/object/gtc-express-webinar.html

Come and try Nsight at the Nsight Training Labs

Room 205A (Second Floor)

	Tuesday, July 23	Wednesday, July 24
9:30am - 10:30am	Debugging	Profiling
11:00am - 12:00pm	Profiling	Debugging
1:30pm - 2:30pm	Debugging	Profiling
3:00pm - 4:00pm	Profiling	Debugging
4:30pm - 5:30pm	Debugging	Profiling

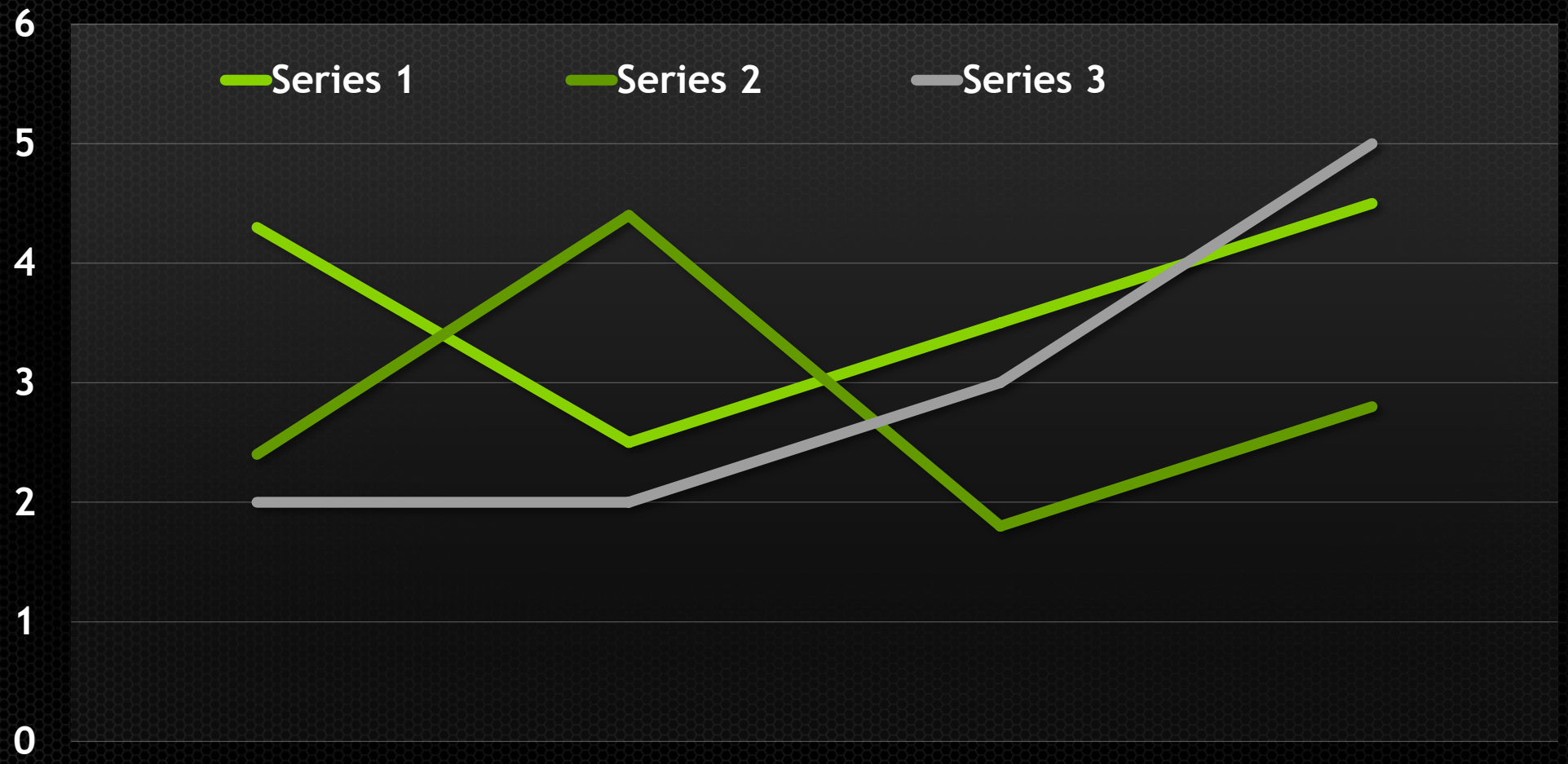
- Backup from here on…



Slide Title

- Bulleted content

Line Chart Sample



Bar Chart Sample

Box Samples

Title
Content

Title
Content

Title
Content

Title
Content

Title
Content

Line Samples