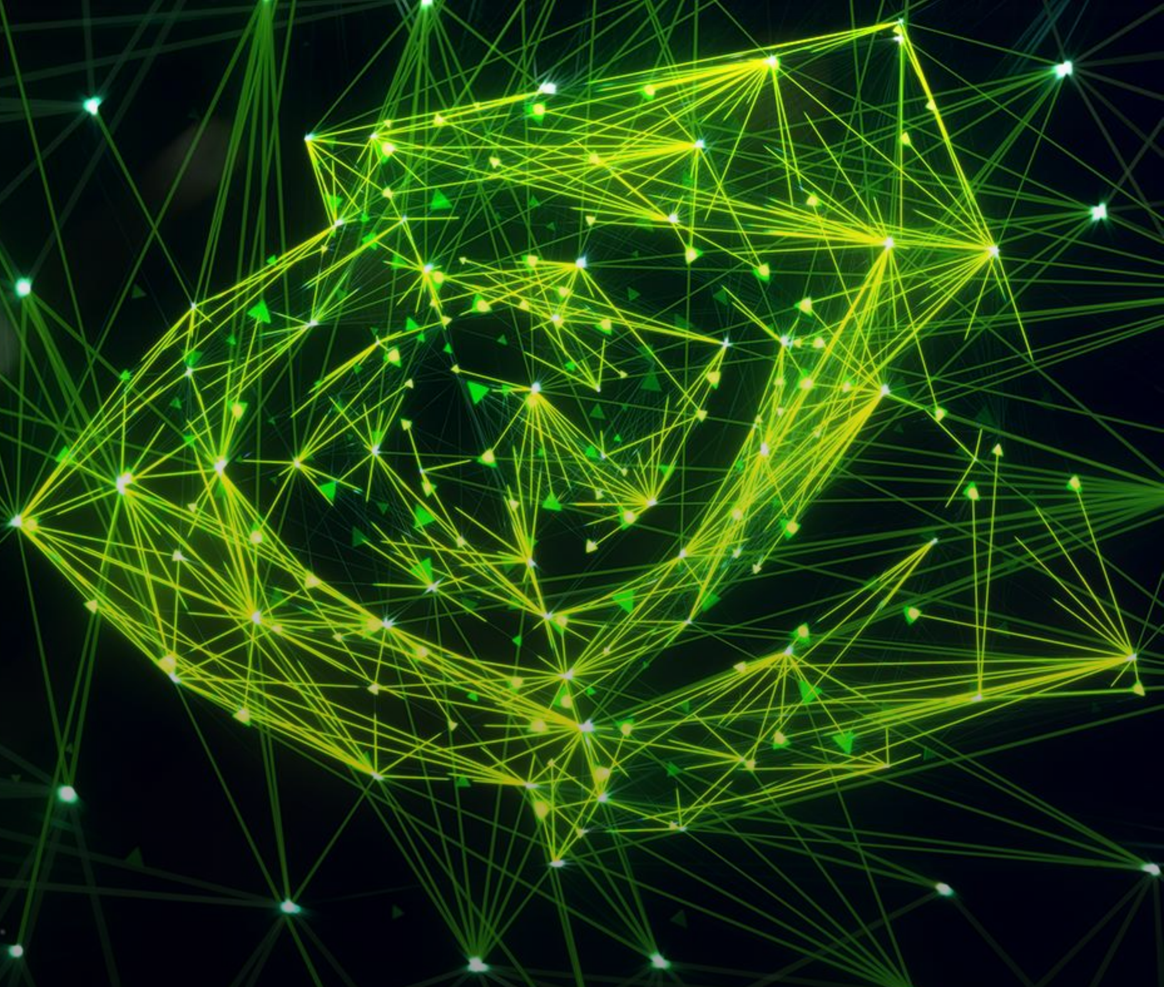


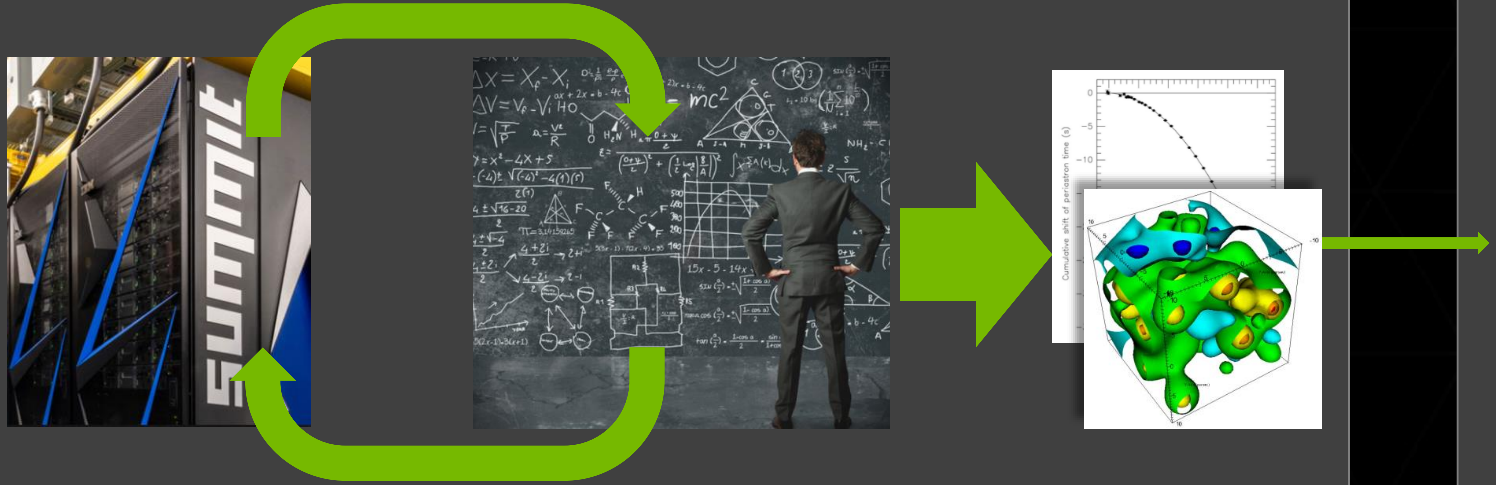
NVIDIA.

VISUALIZE YOUR LARGE DATASETS!

Peter Messmer, 3/20/2019

SCIENTIFIC VIS VS. EDUTAINMENT

Science



Extract information, gain insight

Visual cues, interactivity enhance focus

Helps to understand data

ParaView, VisIt, Matlab, Python,…

Edutainment

Tell a story

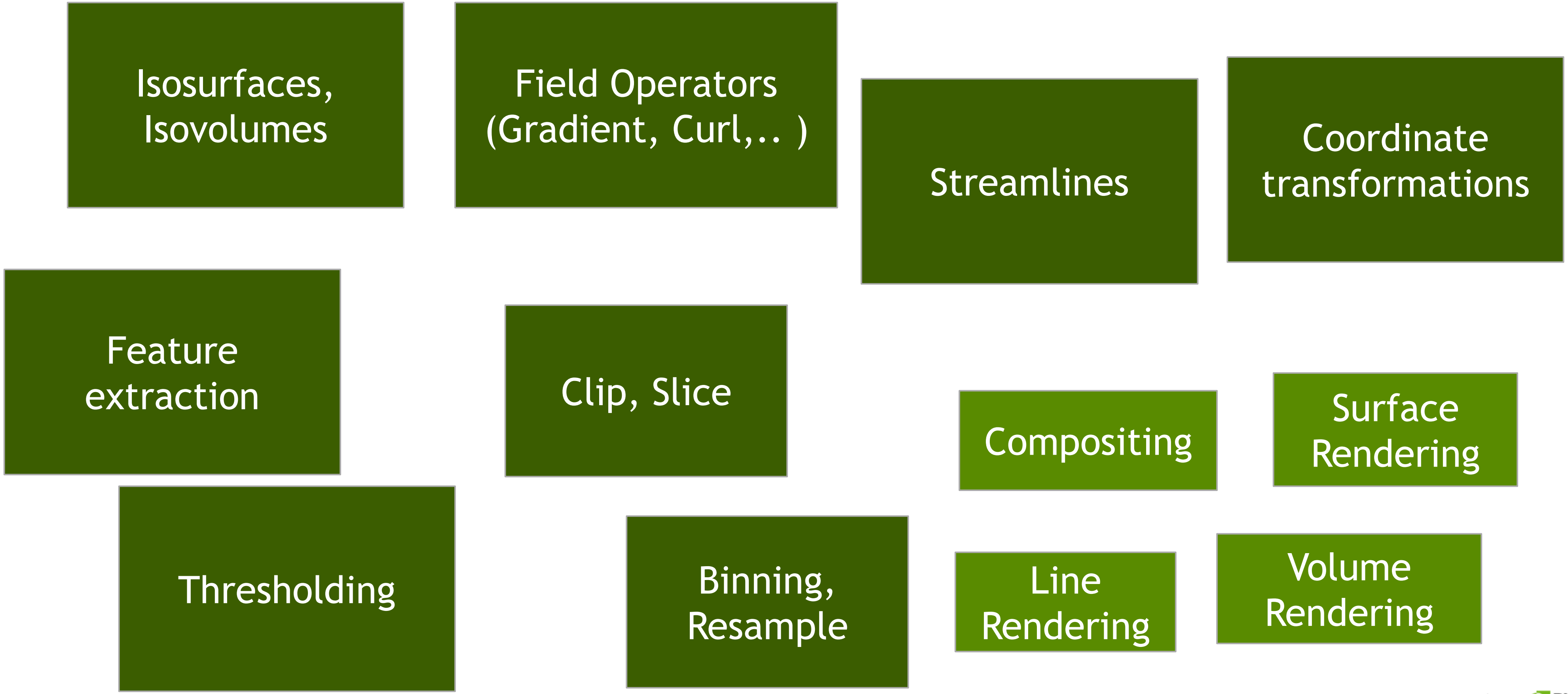
Support story with visual FX

Catch viewer’s attention

Houdini, Blender, Maya, ..

VISUALIZATION ≠ RENDERING *

* but it’s a part of it



Isosurfaces, Isovolumes

Field Operators (Gradient, Curl,..)

Streamlines

Coordinate transformations

Feature extraction

Clip, Slice

Compositing

Surface Rendering

Thresholding

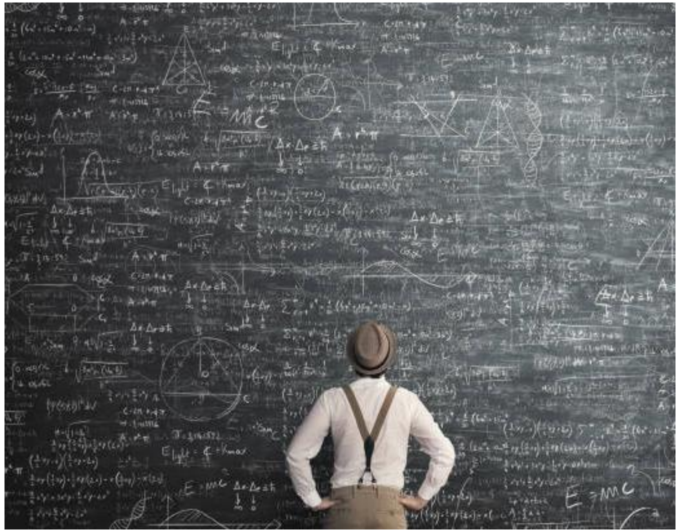
Binning, Resample

Line Rendering

Volume Rendering

CHALLENGES AT LARGE SCALE

Locality



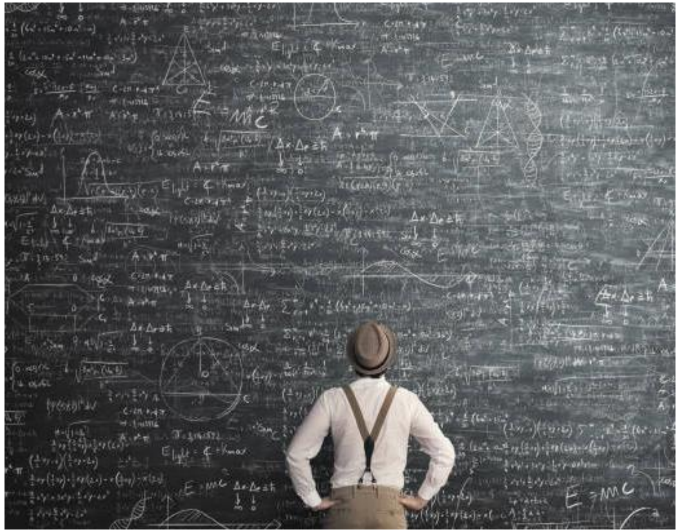
Complexity

Tools

CHALLENGES AT LARGE SCALE

Locality

Leave it where it is



Complexity

Use optimal resource

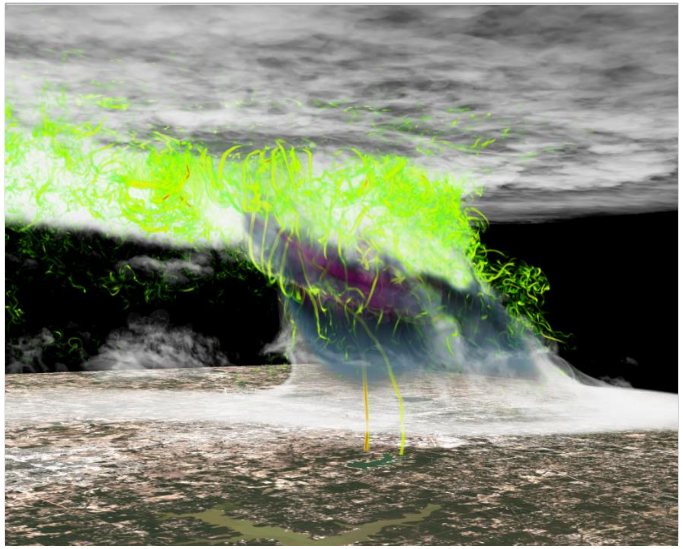
Tools

Minimal intrusion

VISUALIZATION IN THE DATACENTER

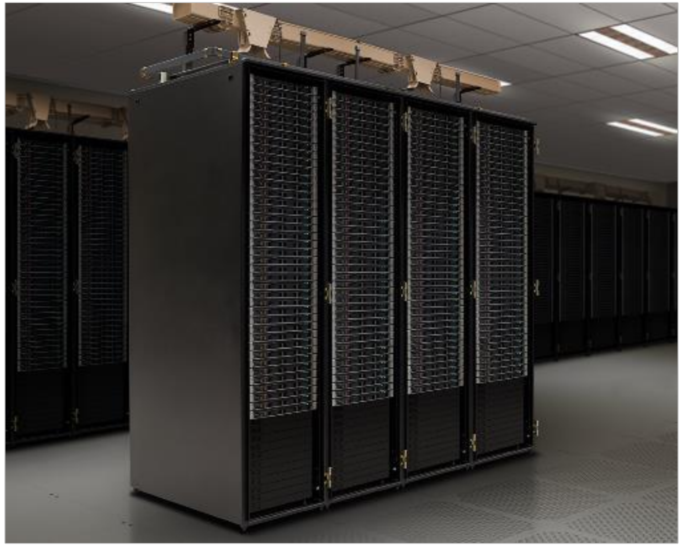
VISUALIZATION IN THE DATACENTER

Benefits of Rendering on Supercomputer



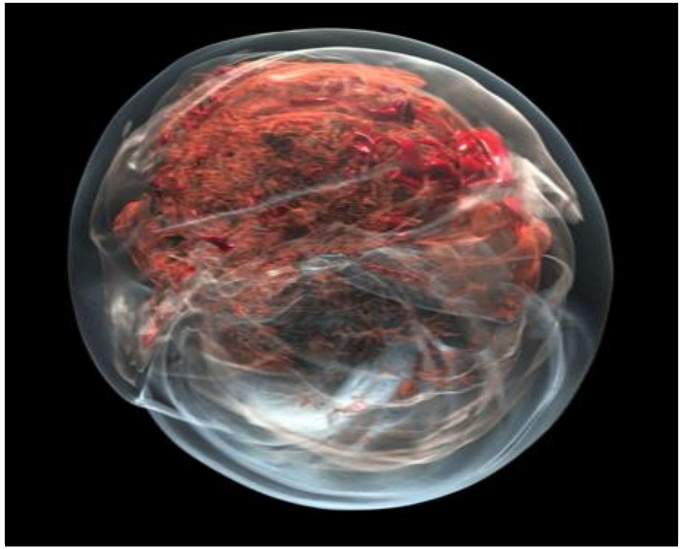
Scale with Simulation

No Need to Scale Separate Vis Cluster



Cheaper Infrastructure

All Heavy Lifting Performed on the Server



Interactive High-Fidelity Rendering

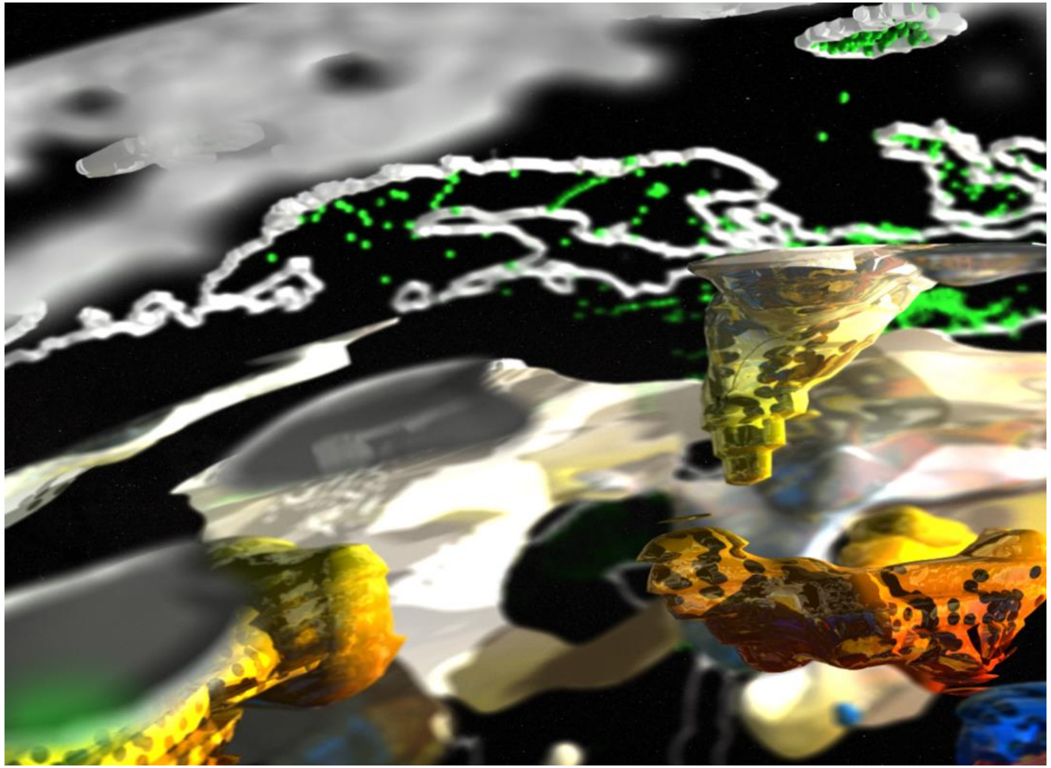
Improves Perception and Scientific Insight

CHALLENGES IN THE DATACENTER

Headless rendering

Remoting

Vis Software Stack



HEADLESS RENDERING

HEADLESS RENDERING

How to rasterize without an attached display

OpenGL context management

Two approaches for context handling:

- X server: mgmt. by separate process
- EGL: mgmt. by driver

X SERVER ON HEADLESS

How to rasterize without an attached display

Recommended if code modification is not an option

```
nvidia-xconfig –o xorg.conf --allow-empty-initial-configuration -a
```

`-o` output file

`-a` enables all GPUs (`-–enable-all-gpus`)

`--allow-empty-initial-configuration` start even if no attached display detected

CONTEXT MANAGEMENT WITH EGL

How to rasterize without an attached display

Requires minor application modification of GLX context initialization

```
// 1. Initialize EGL
EGLDisplay eglDpy = eglGetDisplay(EGL_DEFAULT_DISPLAY);
EGLint major, minor;
eglInitialize(eglDpy, &major, &minor);

// 2. Select an appropriate configuration
EGLint numConfigs; EGLConfig eglCfg;
eglChooseConfig(eglDpy, configAttribs, &eglCfg, 1, &numConfigs);

// 3. Create a surface
EGLSurface eglSurf = eglCreatePbufferSurface(eglDpy, eglCfg, pbufferAttribs);

// 4. Bind the API
eglBindAPI(EGL_OPENGL_API);
```

https://devblogs.nvidia.com/egl-eye-opengl-visualization-without-x-server/

REMOTING

FLEXIBLE GPU ACCELERATION ARCHITECTURE

Independent CUDA Cores & Video Engines

Decode HW*

Formats:

- MPEG-2
- VC1
- VP8
- VP9
- H.264
- H.265
- Lossless

Bit depth:

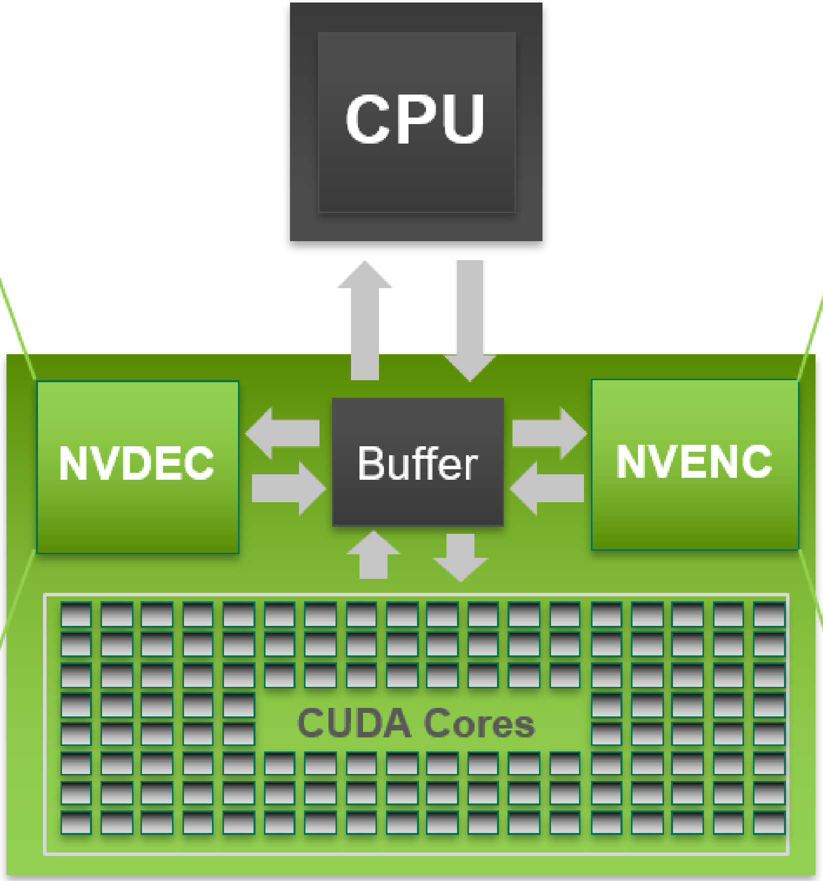
- 8 bit
- 10 bit
- 12 bit

Color**

- YUV 4:2:0
- YUV 4:4:4

Resolution

- Up to 8K***



Encode HW*

Formats:

- H.264
- H.265
- Lossless

Bit depth:

- 8 bit
- 10 bit

Color**

- YUV 4:4:4
- YUV 4:2:0

Resolution

- Up to 8K***

** Diagram represents support for the NVIDIA Turing GPU family*

*** 4:2:2 is not natively supported on HW*

**** Support is codec dependent*

VIDEO CODEC SDK

APIs For Hardware Accelerated Video Encode/Decode

What's New with Turing GPUs and Video Codec SDK 9.0

- Up to 3x decode throughput with multiple decoders on professional cards (Quadro & Tesla)
- Higher quality encoding - H.264 & H.265
- Higher encoding efficiency (15% lower bitrate than Pascal)
- HEVC B-frames support
- HEVC 4:4:4 decoding support

NVIDIA GeForce Now is made possible by leveraging **NVENC** in the datacenter and streaming the result to end clients

https://developer.nvidia.com/nvidia-video-codec-sdk

NVPIPE

A Lightweight Video Codec SDK Wrapper

Simple C API

H.264, HEVC

RGBA32, uint4, uint8, uint16

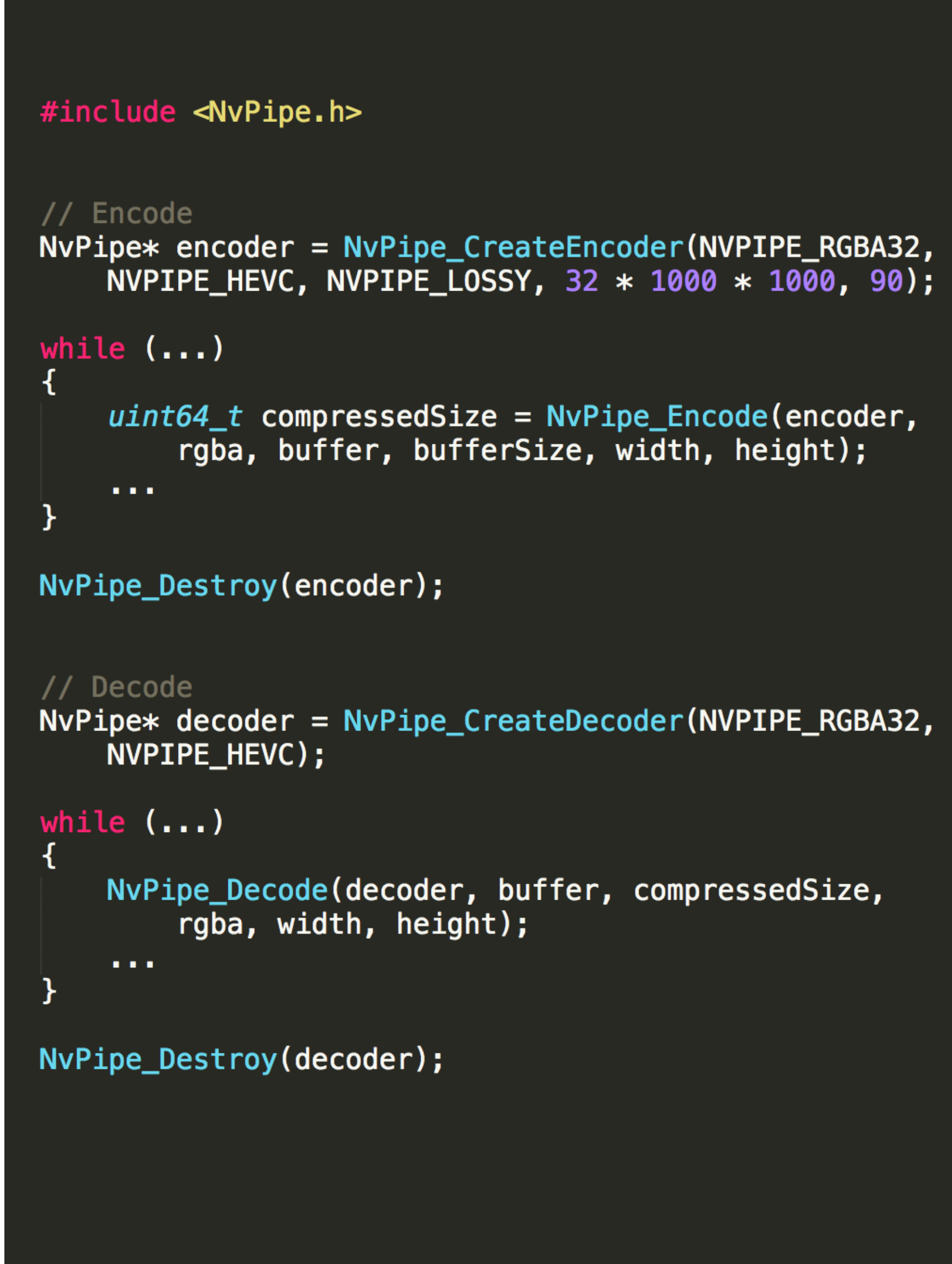
Lossy, Lossless

Host/Device memory, OpenGL textures/PBOs

https://github.com/NVIDIA/NvPipe

Issues? Suggestions? Feedback welcome!

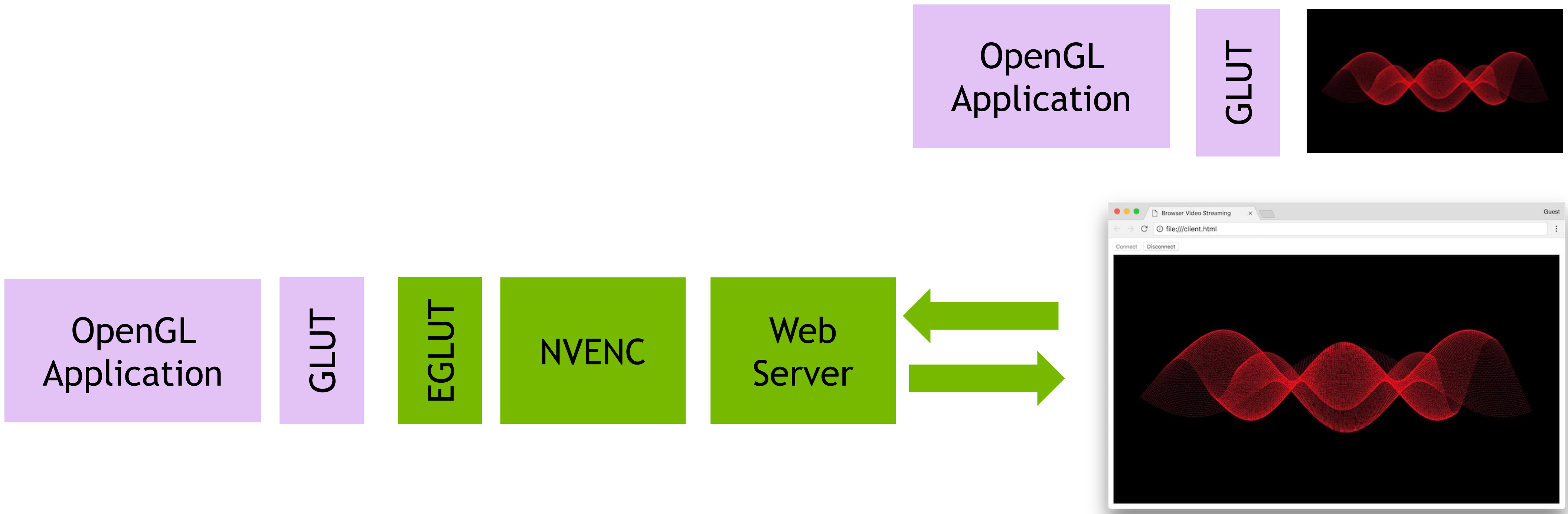
S9490 – GPU-Enhanced Collaborative Scientific Visualization,
Wed 3/20, 11:00-11:50



```
#include <NvPipe.h>


// Encode
NvPipe* encoder = NvPipe_CreateEncoder(NVPIPE_RGBA32,
    NVPIPE_HEVC, NVPIPE_LOSSY, 32 * 1000 * 1000, 90);

while (...)
{
    uint64_t compressedSize = NvPipe_Encode(encoder,
        rgba, buffer, bufferSize, width, height);
    ...
}

NvPipe_Destroy(encoder);


// Decode
NvPipe* decoder = NvPipe_CreateDecoder(NVPIPE_RGBA32,
    NVPIPE_HEVC);

while (...)
{
    NvPipe_Decode(decoder, buffer, compressedSize,
        rgba, width, height);
    ...
}

NvPipe_Destroy(decoder);
```

EGL RENDERING + BROWSER STREAMING

Powerful combo for rendering in the cloud



OpenGL Application | GLUT

OpenGL Application | GLUT | EGLUT | NVENC | Web Server

mp4 stream

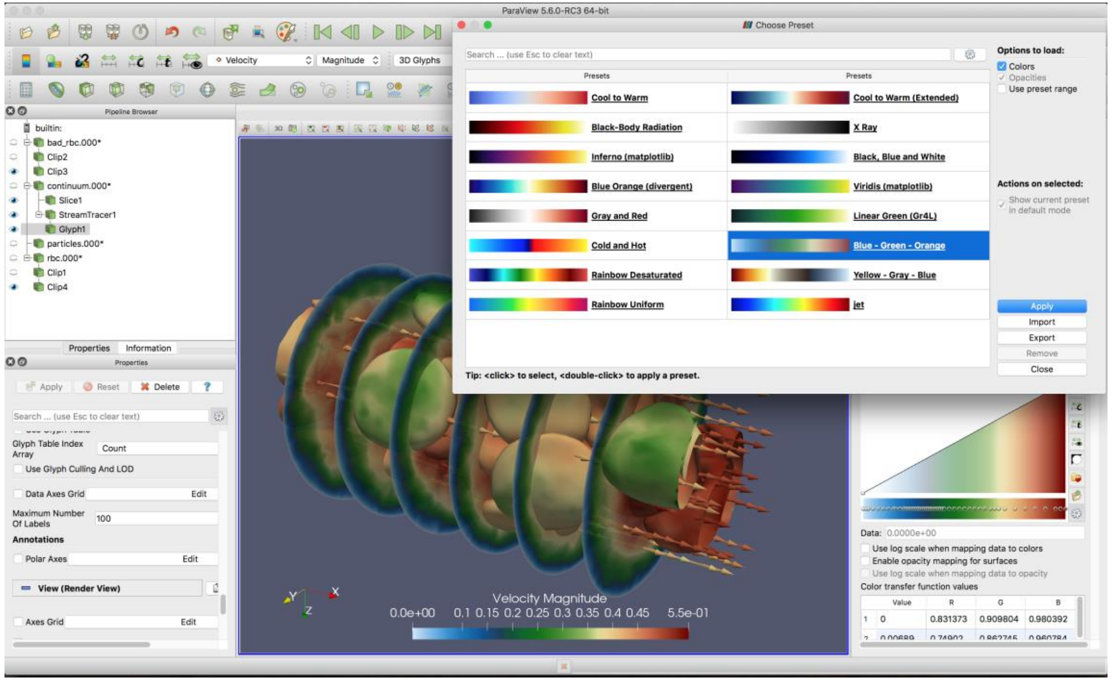
Java Script Client

Available upon request

TOOL COMPLEXITY

KITWARE PARAVIEW

Open-Source (Distributed) Visualization Package



Search ... (use Esc to clear text)

Options to load:

☑ Colors

☑ Opacities

□ Use preset range

Actions on selected:

☑ Show current preset in default mode

Tip: <click> to select, <double-click> to apply a preset.

Glyph Table Index Array: Count

□ Use Glyph Culling And LOD

□ Data Axes Grid — Edit

Maximum Number Of Labels: 100

Annotations

□ Polar Axes — Edit

□ Axes Grid — Edit

Data: 0.0000e+00

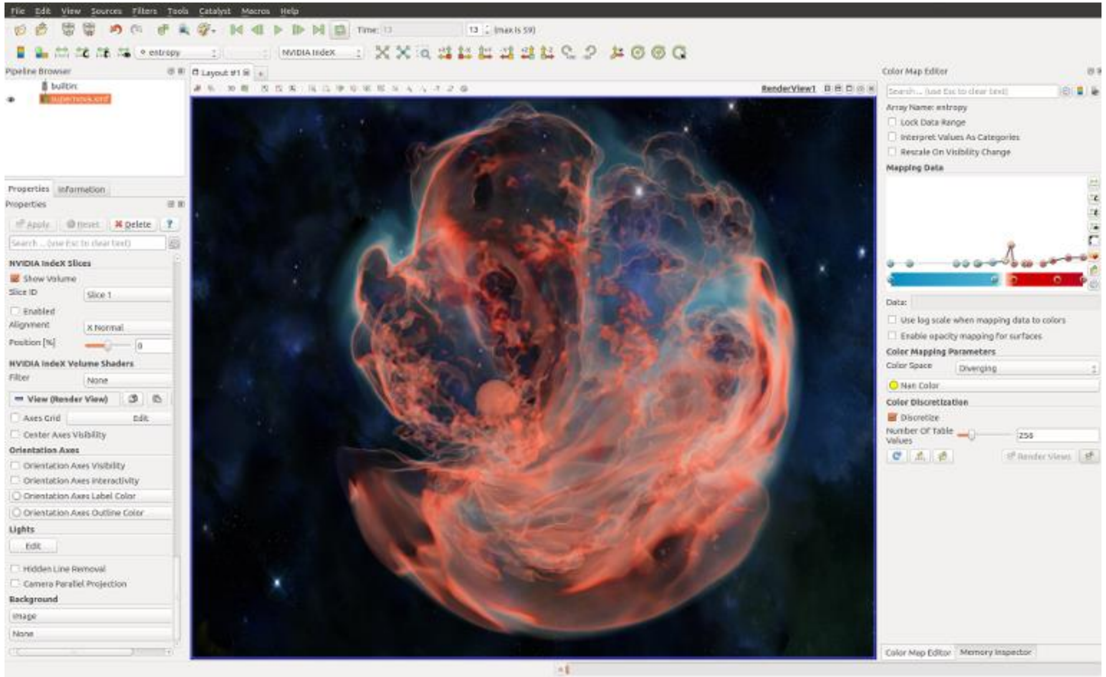
□ Use log scale when mapping data to colors

□ Enable opacity mapping for surfaces

□ Use log scale when mapping data to opacity

Color transfer function values

	Value	R	G	B
1	0	0.831373	0.909804	0.980392
2	0.00689	0.74902	0.862745	0.960784



Search ... (use Esc to clear text)

NVIDIA IndeX Slices

☑ Show Volume

Slice ID: Slice 1

□ Enabled

Alignment: X Normal

Position [%]: 0

NVIDIA IndeX Volume Shaders

Filter: None

View (Render View)

□ Axes Grid — Edit

□ Center Axes Visibility

Orientation Axes

□ Orientation Axes Visibility

□ Orientation Axes Interactivity

○ Orientation Axes Label Color

○ Orientation Axes Outline Color

Lights — Edit

□ Hidden Line Removal

□ Camera Parallel Projection

Background: Image

None

Array Name: entropy

□ Lock Data Range

□ Interpret Values As Categories

□ Rescale On Visibility Change

Mapping Data

Data:

□ Use log scale when mapping data to colors

□ Enable opacity mapping for surfaces

Color Mapping Parameters

Color Space: Diverging

○ Nan Color

Color Discretization

☑ Discretize

Number Of Table Values: 256

OpenGL

NVIDIA IndeX Plugin

ParaView

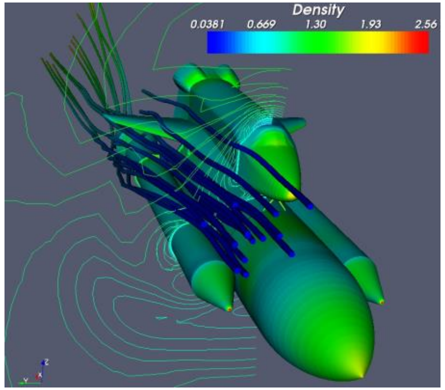
VTK: VISUALIZATION TOOLKIT

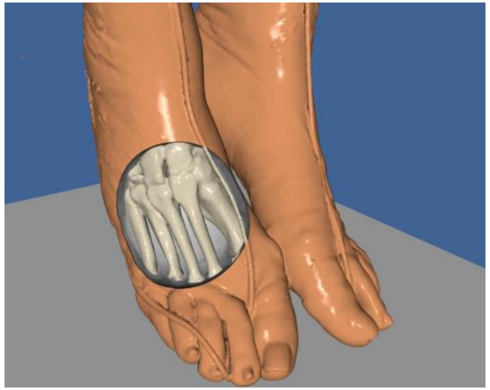
Open Source Scientific Visualization Toolbox

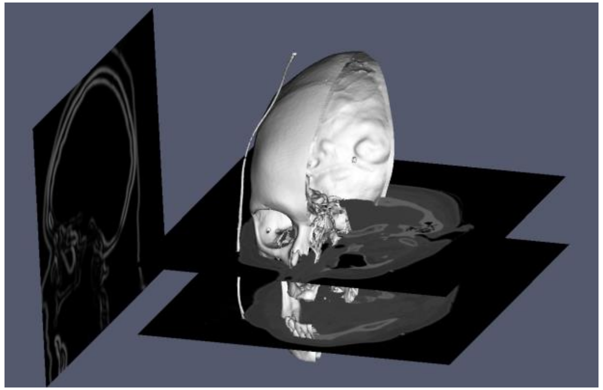
Process data using pipelines made up of filters

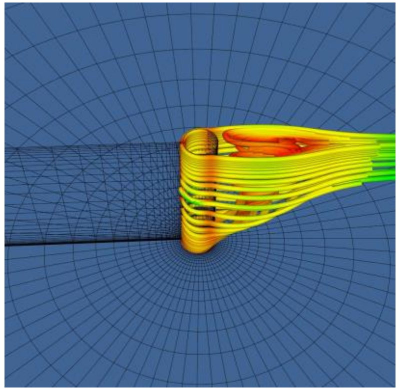
Forms the foundation of ParaView, VisIt and many other vis tools

OpenGL, Software raytracing









S9458 – VTK-m: Lessons from Building a Visualization Toolkit for Massively Threaded Architectures, Wed 3/20, 3:00-3:50

CONTAINERS: SIMPLIFYING WORKFLOWS

WHY CONTAINERS

Simplifies Deployments

- Eliminates complex, time-consuming builds and installs

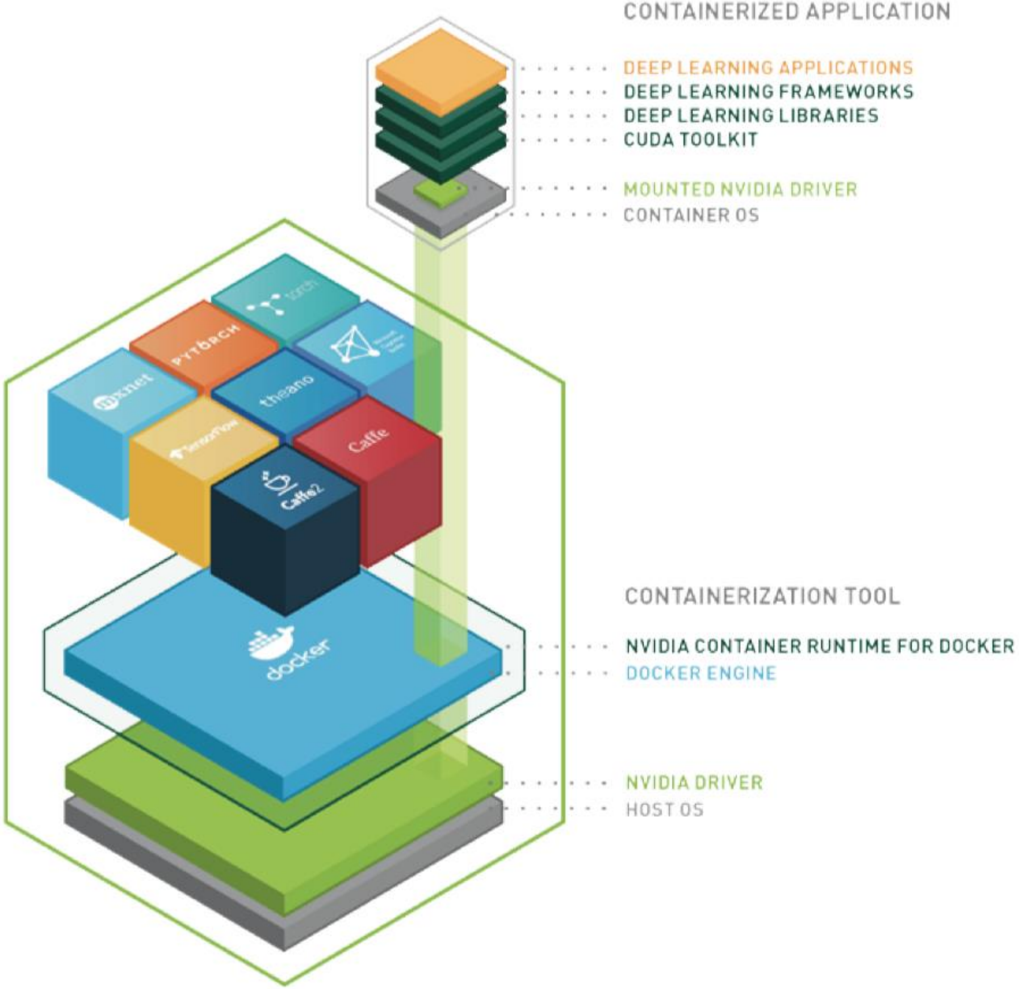
Get started in minutes

- Simply Pull & Run the app

Portable

- Deploy across various environments, from test to production with minimal changes

S9525 – Containers Democratize HPC, Tue 3/19



NGC CONTAINERS: ACCELERATING WORKFLOWS

WHY CONTAINERS

Simplifies Deployments

- Eliminates complex, time-consuming builds and installs

Get started in minutes

- Simply Pull & Run the app

Portable

- Deploy across various environments, from test to production with minimal changes

WHY NGC CONTAINERS

Optimized for Performance

- Monthly DL container releases offer latest features and superior performance on NVIDIA GPUs

Scalable Performance

- Supports multi-GPU & multi-node systems for scale-up & scale-out environments

Designed for Enterprise & HPC environments

- Supports Docker & Singularity runtimes

Run Anywhere

- Pascal/Volta/Turing-powered NVIDIA DGX, PCs, workstations, servers and top cloud platforms

GPU-OPTIMIZED SOFTWARE CONTAINERS

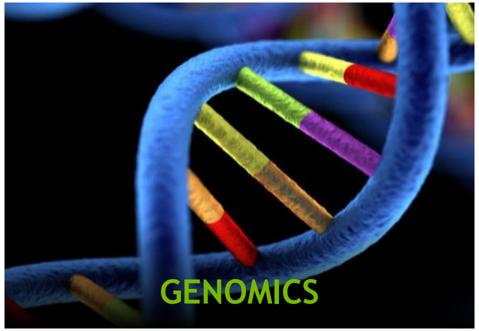
Over 50 Containers on NGC

TensorFlow | PyTorch | more

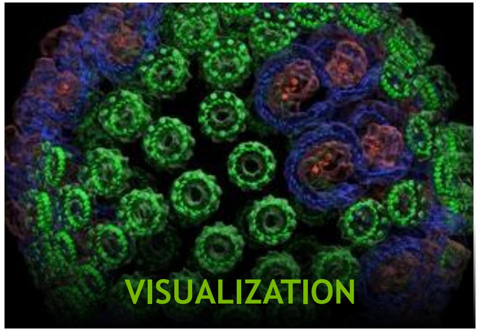
RAPIDS | H2O | more

TensorRT | DeepStream | more

NAMD | GROMACS | more



Parabricks



ParaView | IndeX | more

RENDERING: 2D

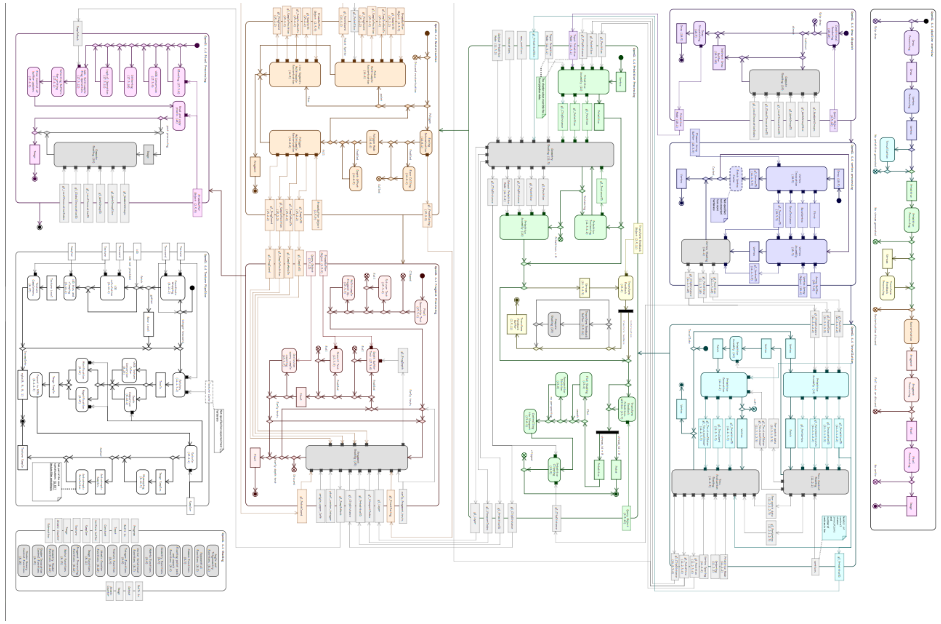
GPU ACCELERATED VECTOR GRAPHICS

Acceleration of 2D Graphics

GPUs primary rendering focus on 3D

2D rendering is so much more common

Often served out via web pages



Examples

graphs, diagrams, networks, flow charts, maps, vector artwork, Flash-like animation, etc. etc.

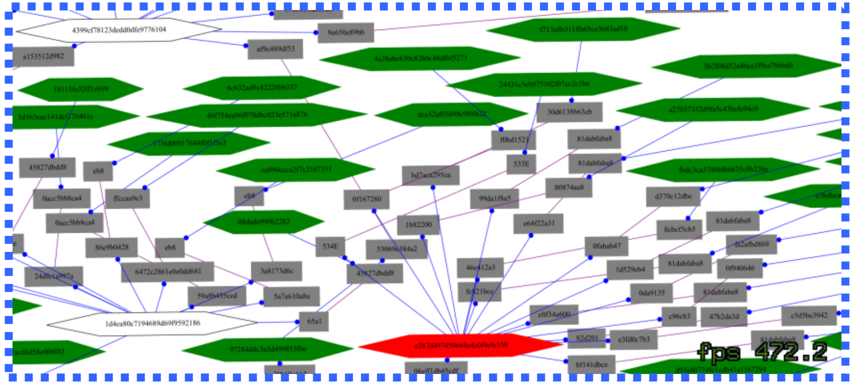
SCALABLE VECTOR GRAPHICS (SVG)

Pros:

- Wide support, efficient implementations
- Very powerful feature set

Cons:

- Slow due to client-side rendering in browser
- SVG contains data, not just pixels

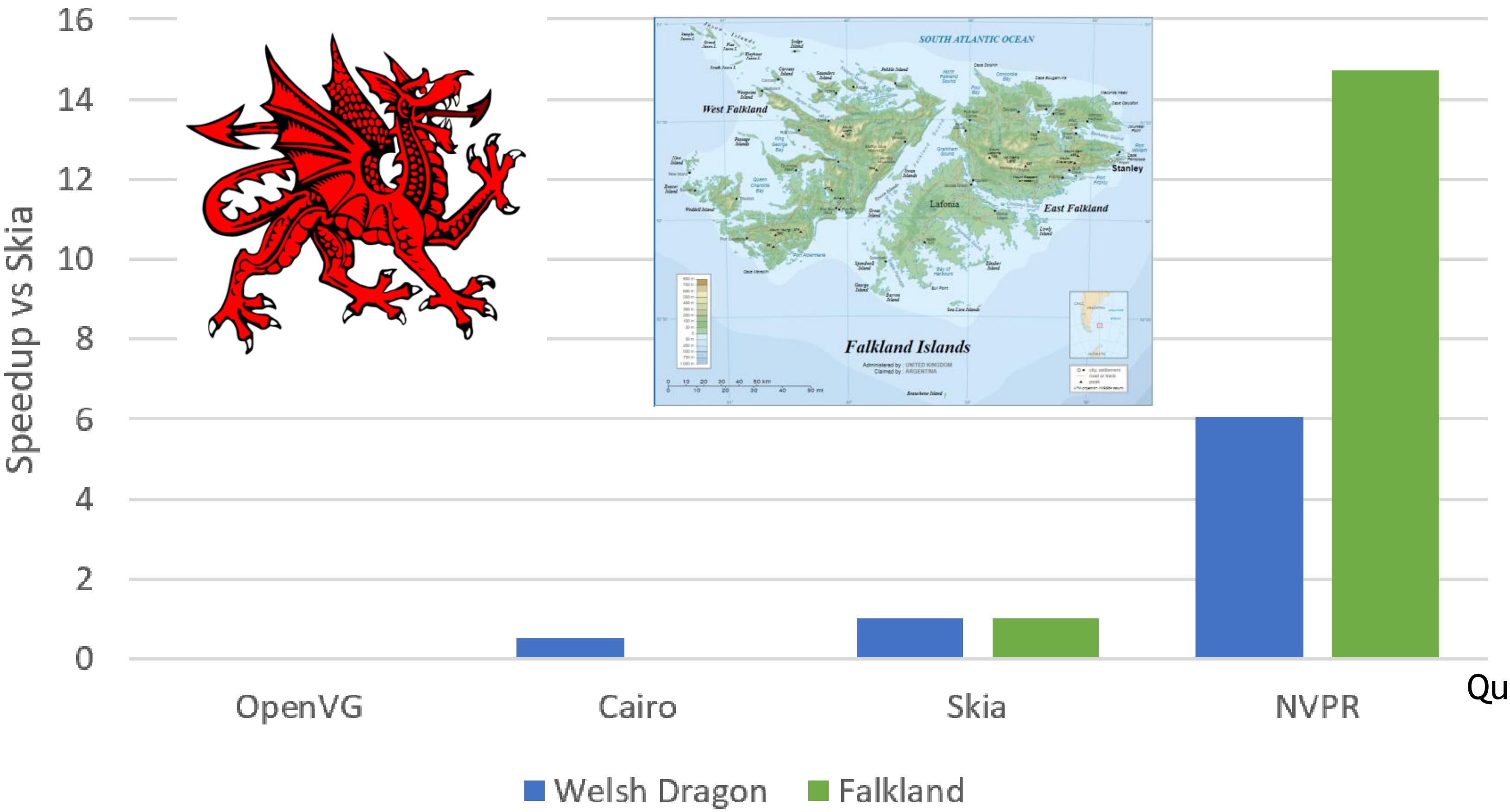


⇒ GPU cloud rendering addresses <u>both</u> downsides

⇒ Support via NV_path_rendering OpenGL extension

SVG RENDERING PERFORMANCE

Bigger benefit for more complex scenes



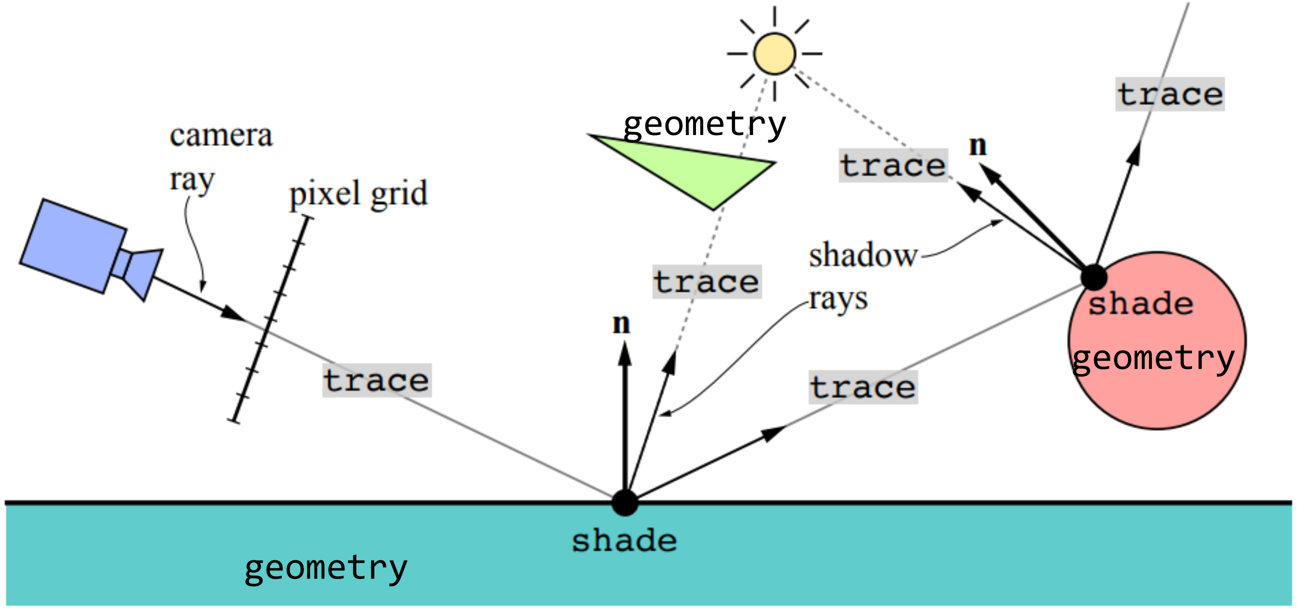
Quadro RTX6000
i7-5820k

RENDERING: RAYTRACING

ANATOMY OF A RAY-TRACING APP

Interplay of Rays and Geometry

- Intersection of rays with geometry
- Arbitrary new rays started at arbitrary locations
- Arbitrary operations at intersectior points
- Typically in 3D space
- Hierarchical spatial decomposition as acceleration structure



TURING RT CORES

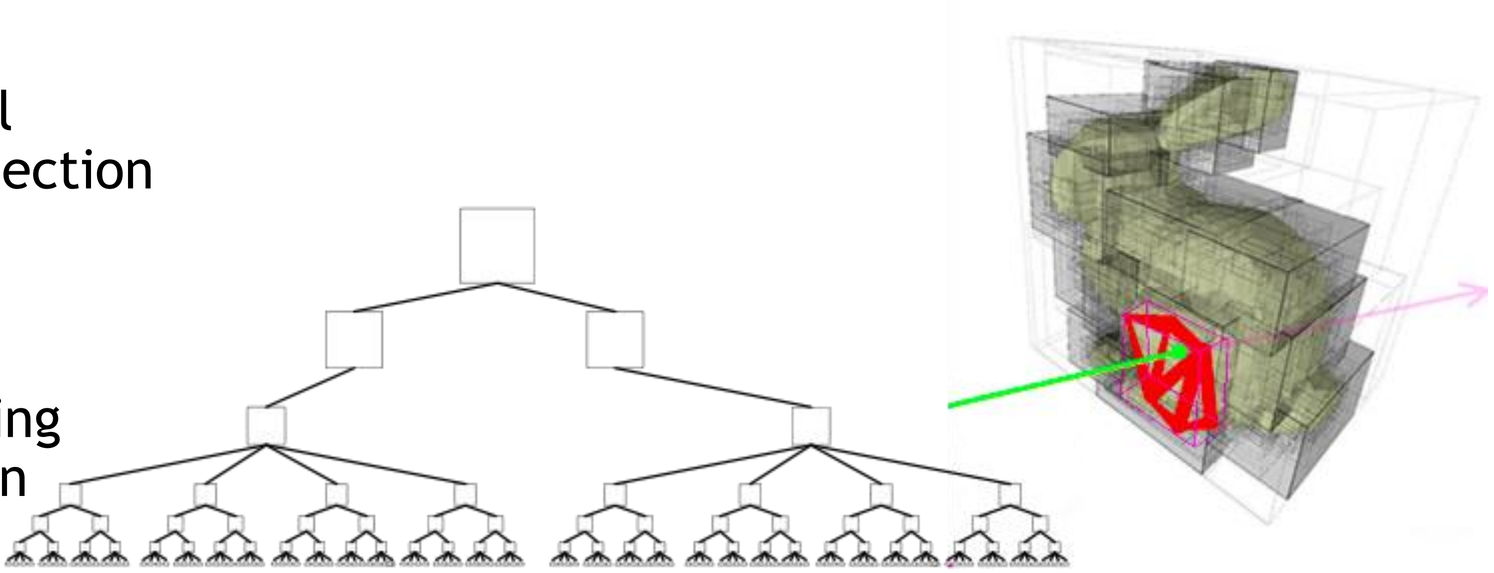
Hardware Accelerated Ray Tracing

RT Cores perform

- Ray-BVH Traversal
- Instancing: 1 Level
- Ray-Triangle Intersection

Return to SM for

- Multi-level Instancing
- Custom Intersection
- Shading

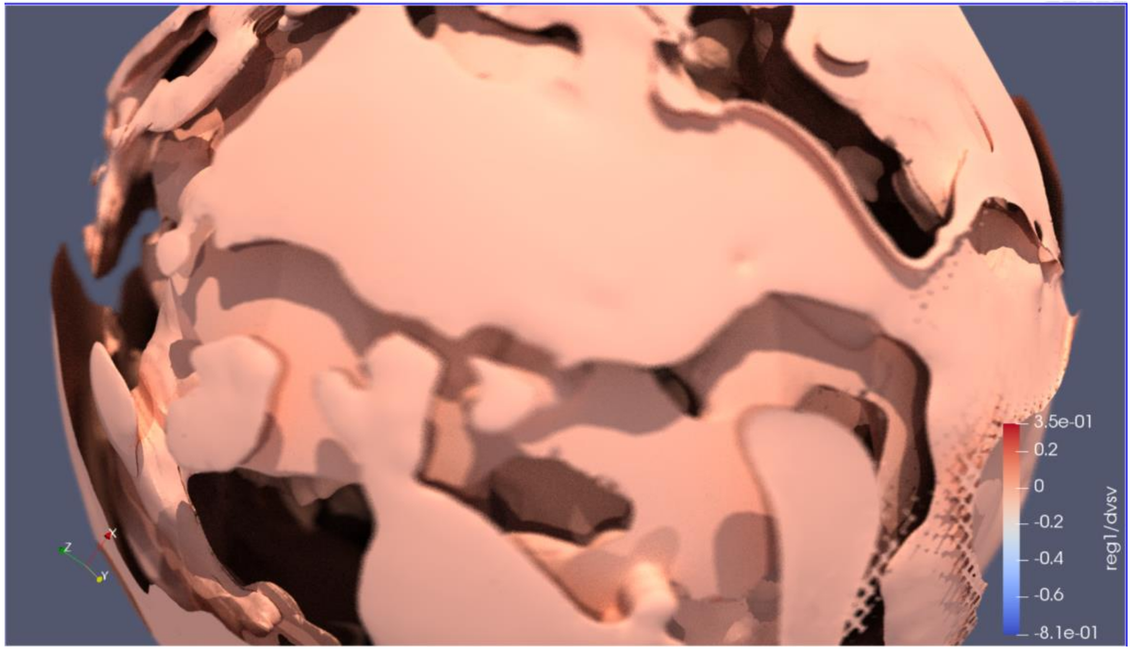


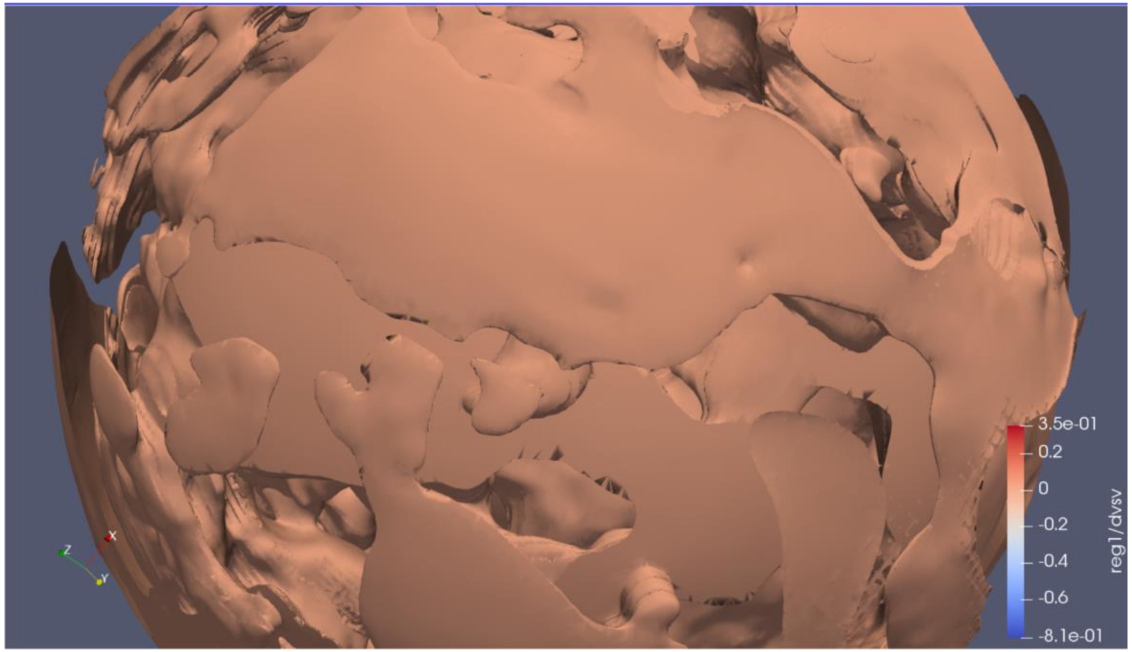
Programming via OptiX RT framework
Low overhead interop with CUDA

S9768 – New Features in OptiX 6.0
Wed 3/20, 1:00-1:50pm

BETTER INSIGHT VIA RAYTRACING

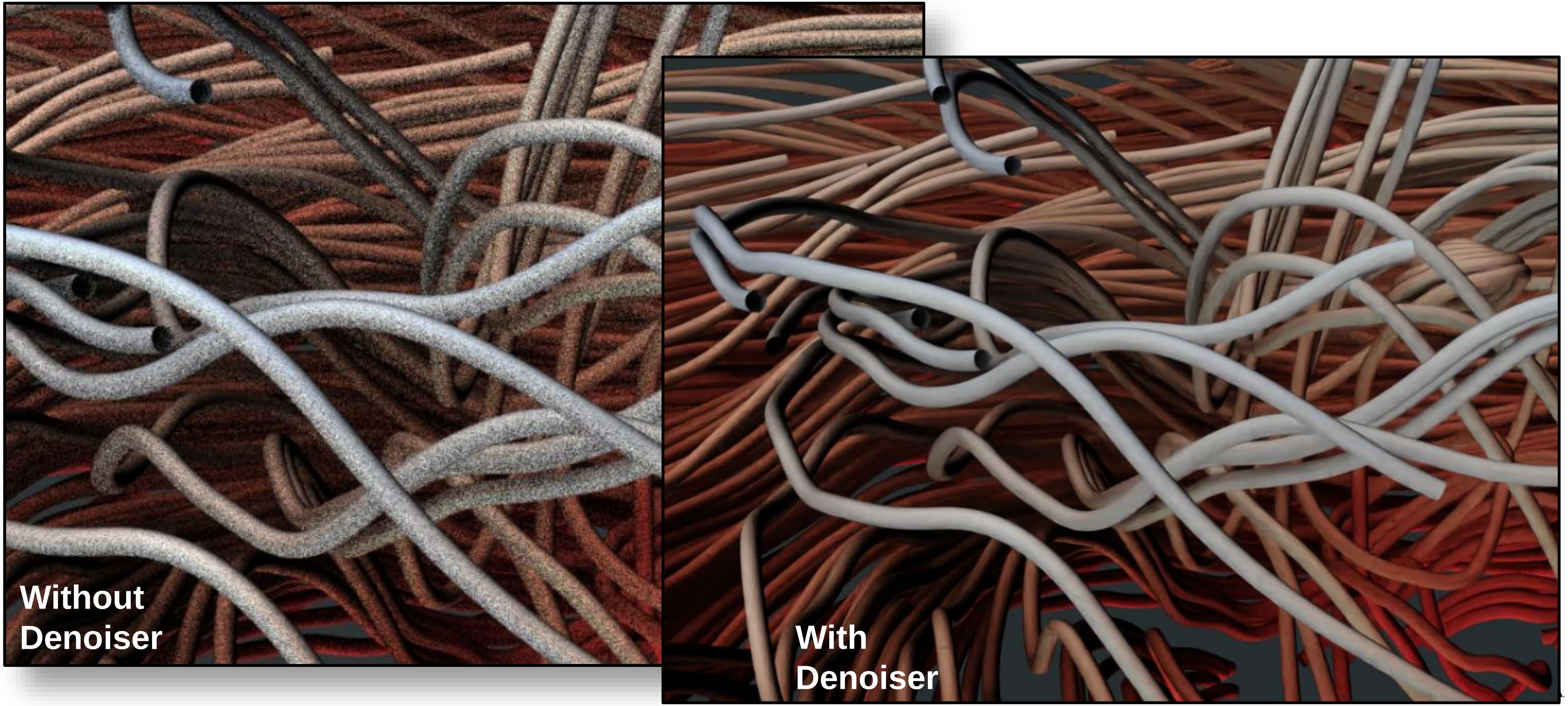
It’s not just pretty pictures





S9589 – Interactive High-Fidelity Biomolecular and Cellular Visualization with RTX Ray Tracing APIs
Wed 3/20, 3:00-3:50pm

OPTIX AI DENOISER IN PARAVIEW



VISRTX

Visualization Framework Powered by NVIDIA RTX Technology

Progressive forward pathtracer with NEE/MIS

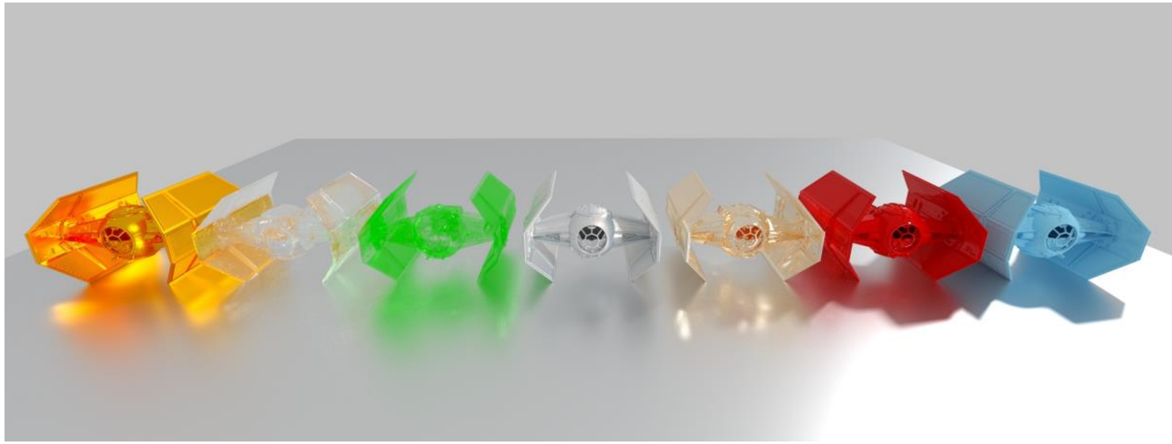
Hardware-acceleration through OptiX

MDL for physically-based materials

AI denoiser

Area lights, Depth of Field, Tone mapping, etc.

Open-source C++ library



http://github.com/NVIDIA/VisRTX

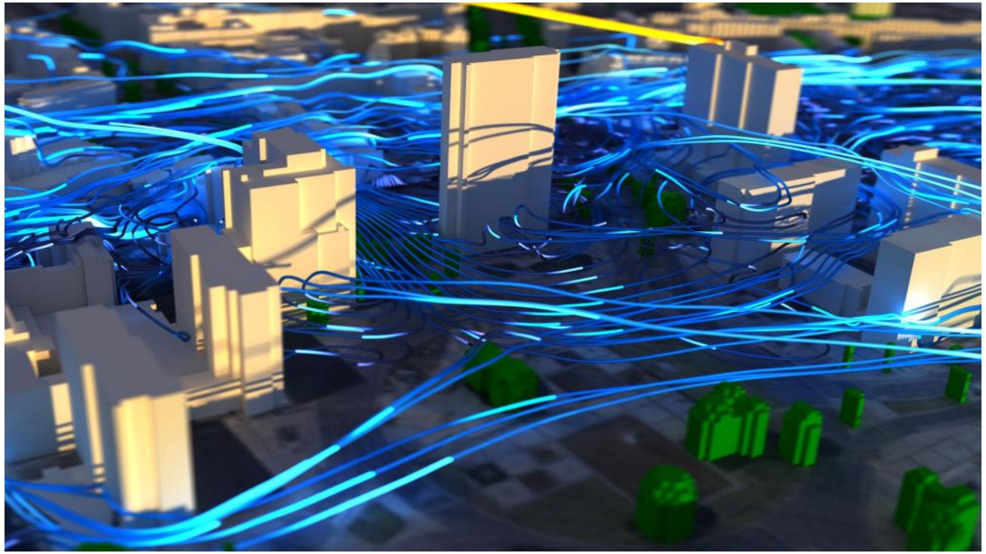
Feedback welcome (issues, PRs, e-mail)!

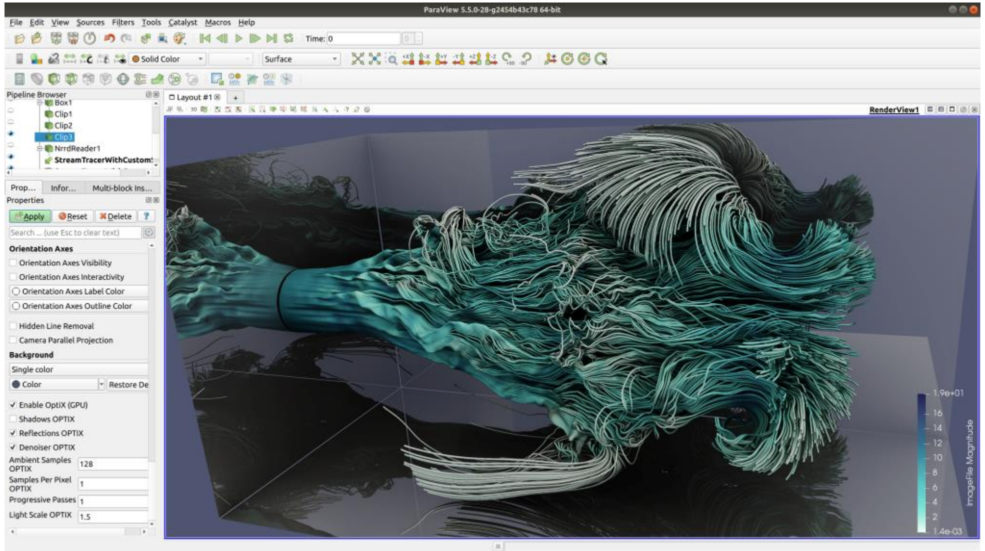
VISRTX + PARAVIEW

VisRTX open-source on GitHub

Shipped with upcoming ParaView 5.7

- **No additional steps necessary!**





Orientation Axes

□ Orientation Axes Visibility

□ Orientation Axes Interactivity

○ Orientation Axes Label Color

○ Orientation Axes Outline Color

□ Hidden Line Removal

□ Camera Parallel Projection

Background

Single color

● Color | Restore De

☑ Enable OptiX (GPU)

□ Shadows OPTIX

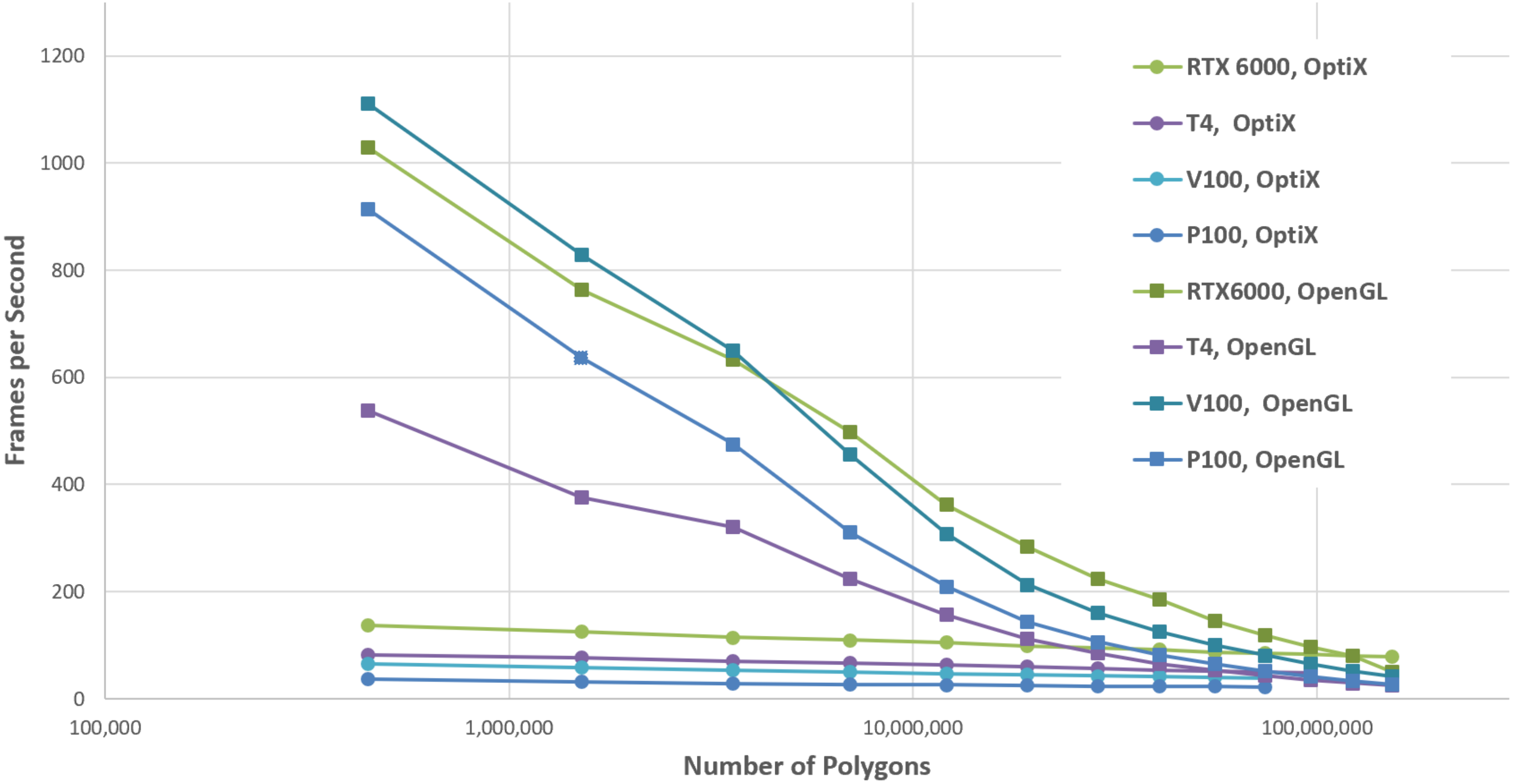
☑ Reflections OPTIX

☑ Denoiser OPTIX

Ambient Samples OPTIX	128
Samples Per Pixel OPTIX	1
Progressive Passes	1
Light Scale OPTIX	1.5

S9697 – Path Tracing in ParaView-OptiX: RTX for Scientific Visualization
Thu 3/21, 4:00-4:50

RAYTRACING PAYS OFF AT SCALE



ParaView Manyspheres Benchmark

RENDERING: VOLUMES

NVIDIA IndeX SDK

Large scale and distributed data rendering

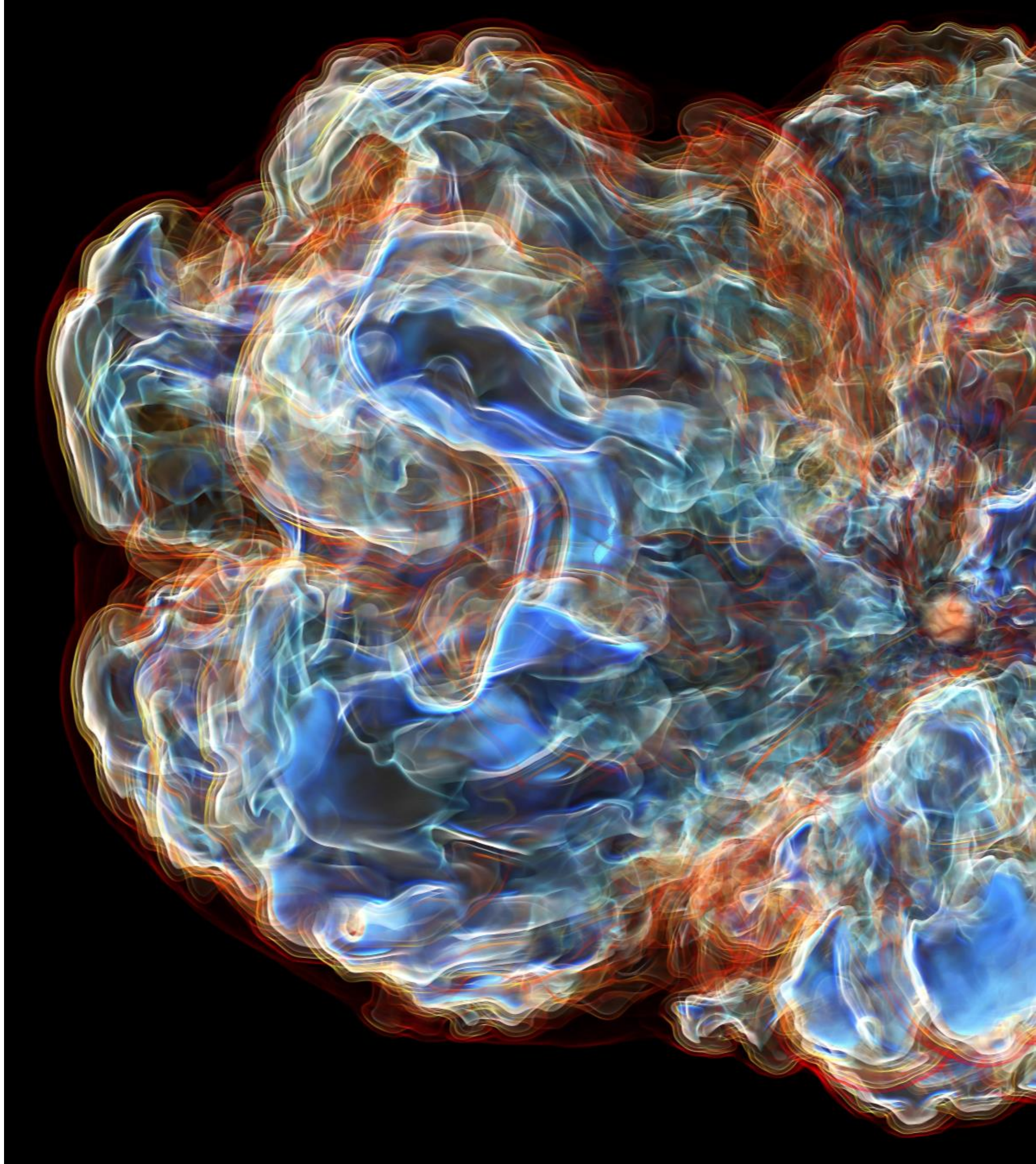
Scene management with volume data

Transparent support for NVLink

Higher-order filtering, advanced lighting & transfer functions

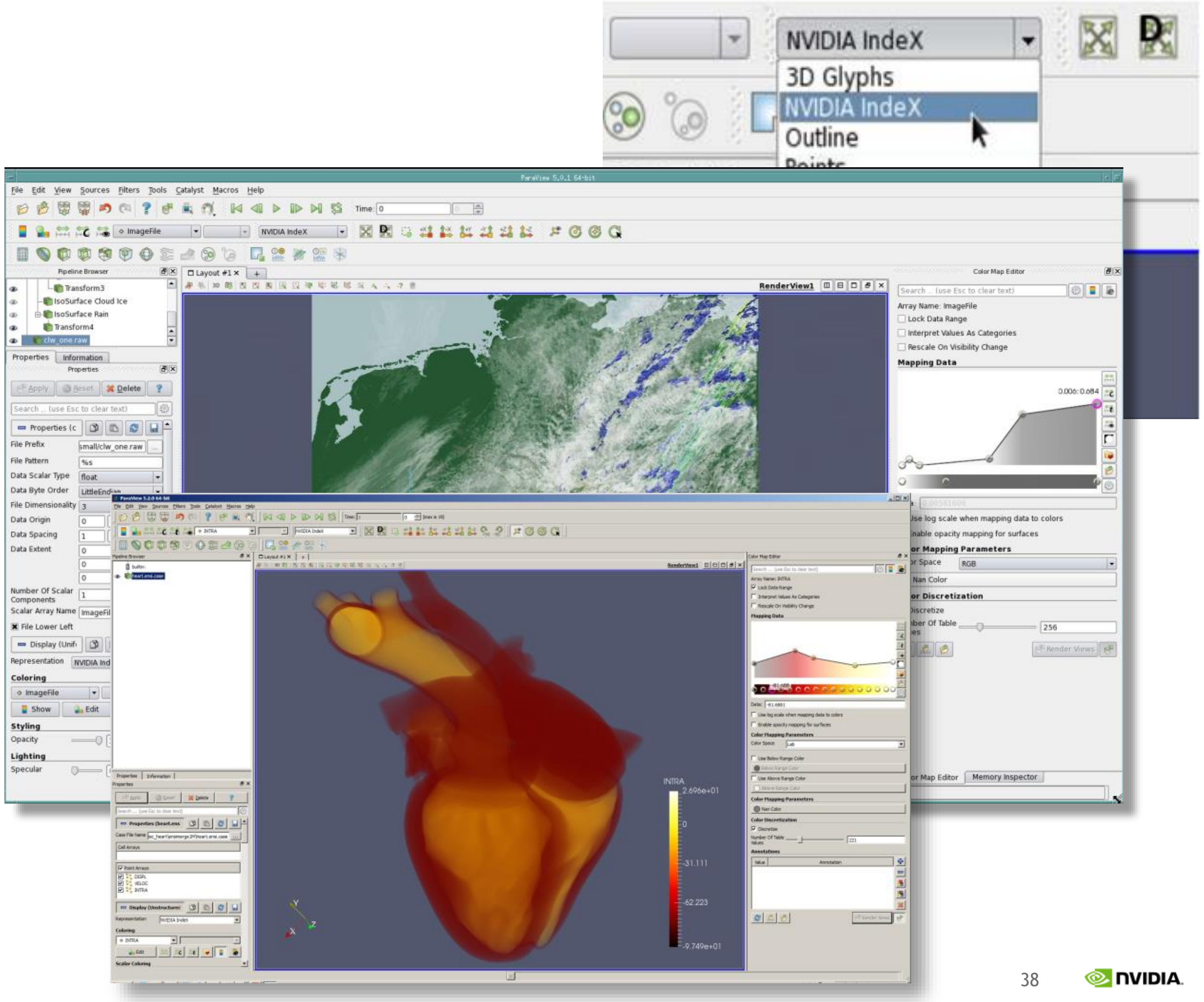
https://developer.nvidia.com/index

S9692 – NVIDIA IndeX – Implementing Cloud Services for Complex Scientific Data Visualization, Tue 3/21, 4:00-4:50



NVIDIA INDEX FOR PARAVIEW PLUGIN

- NVIDIA IndeX rendering in ParaView
- Retain ParaView workflows
- Structured and unstructured meshes

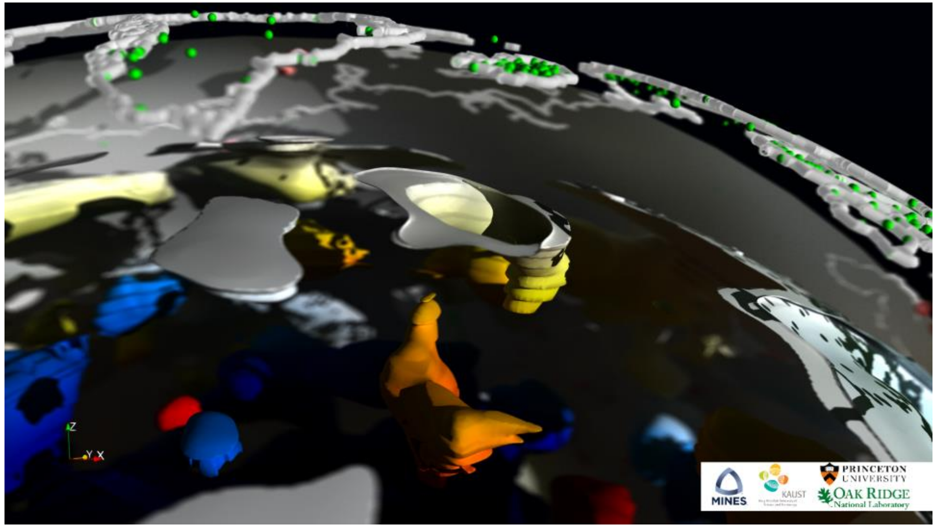


Learn more:

http://www.nvidia.com/object/index-paraview-plugin.html

SUMMARY

Wide Palette for Visualization and Rendering in Datacenter/Cloud



Headless rendering

Accelerated video streaming

2D graphs can benefit from GPUs as well

Raytracing great to enhance vis perception

VisRTX raytracing vis tookit (in ParaView, VTK)

GPU accelerated scalable volume rendering part of open source tools