IXPUG In Situ Workshop Report – Best Practices and Lessons Learned

2017 IXPUG US Annual Meeting

September 27, 2017

PRESENTED BY:

Paul A. Navrátil, Ph.D.

Deputy Director of Visualization

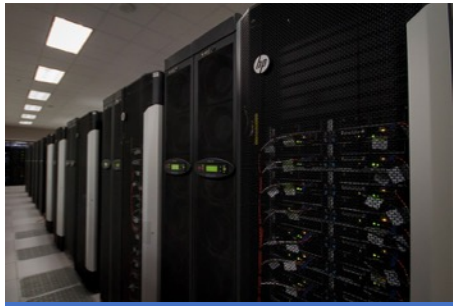
pnav@tacc.utexas.edu

History of Remote Visualization at TACC

Maverick – Sun Fire E25K 3dfx subsystem

Spur – 8 node Sun AMD NVIDIA cluster

Longhorn – 256 node Dell Intel NVIDIA cluster



Maverick – 132 node HP Intel NVIDIA cluster

same data center

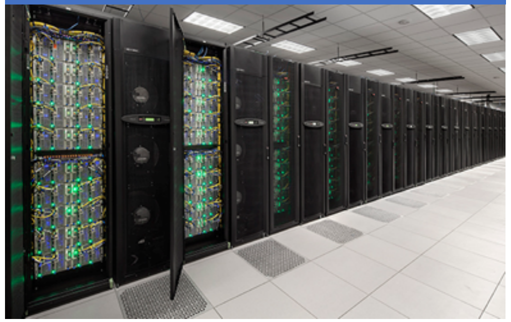
2004 2008 2010 2011 2013 2014

same interconnect fabric

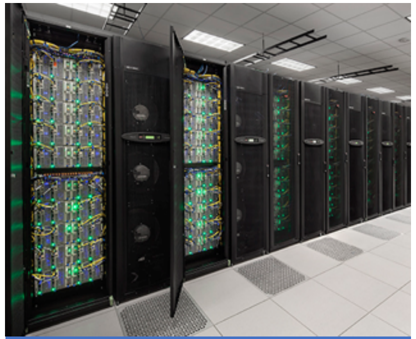
Ranger – 8 node Sun AMD NVIDIA subsystem

Lonestar – 16 node Dell Intel NVIDIA subsystem

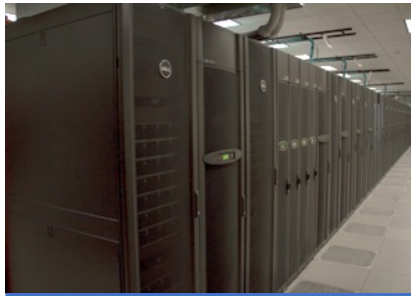
Stampede – 128 node Dell Intel NVIDIA subsystem



2015 – present same machine



Stampede-KNL – 508 node Dell Intel KNL cluster



Stampede2– 4200 node Dell Intel KNL cluster

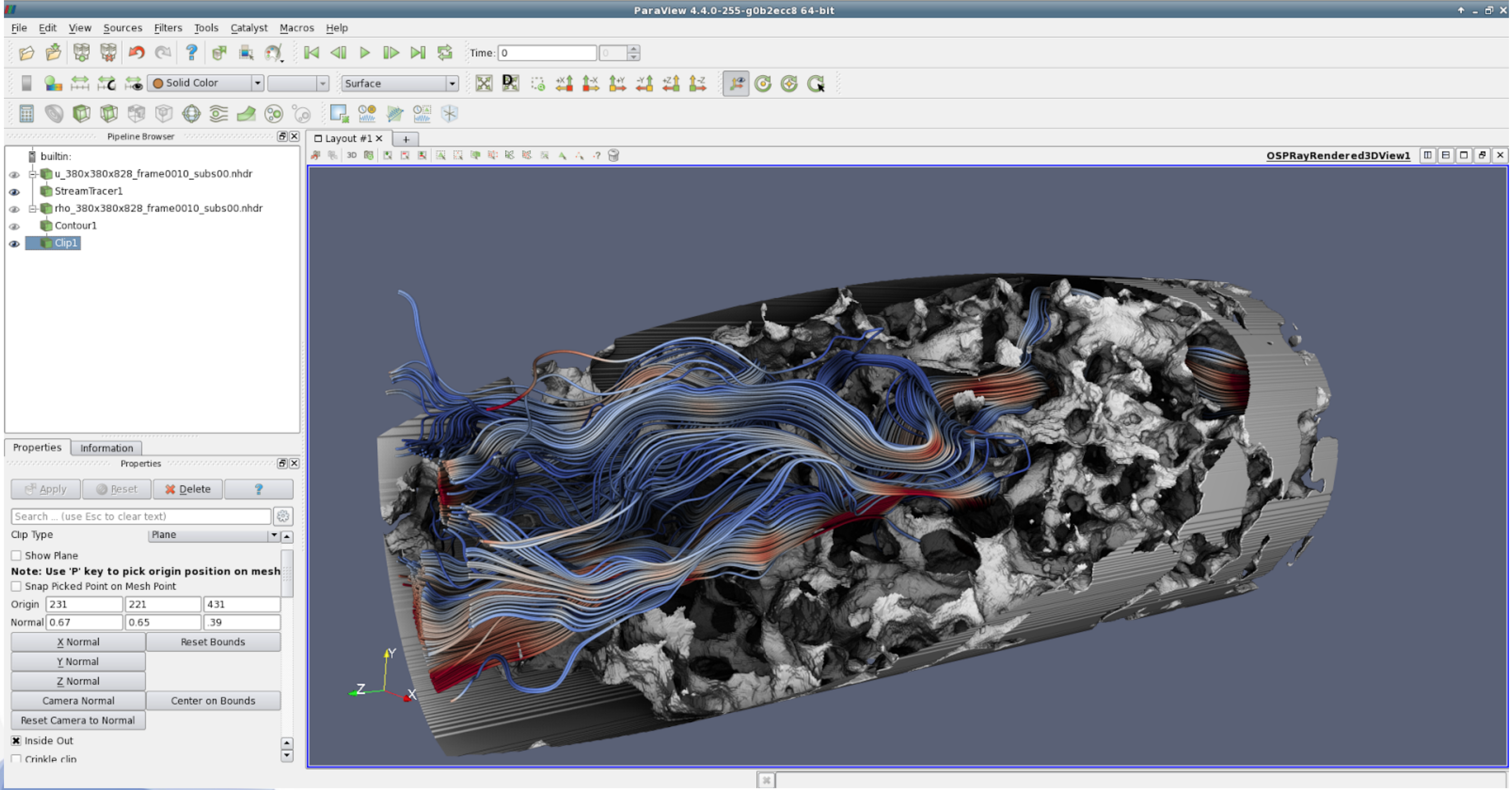
TACC

Stampede2 Visualization Overview

Stampede2 Architectural Vision for Visualization

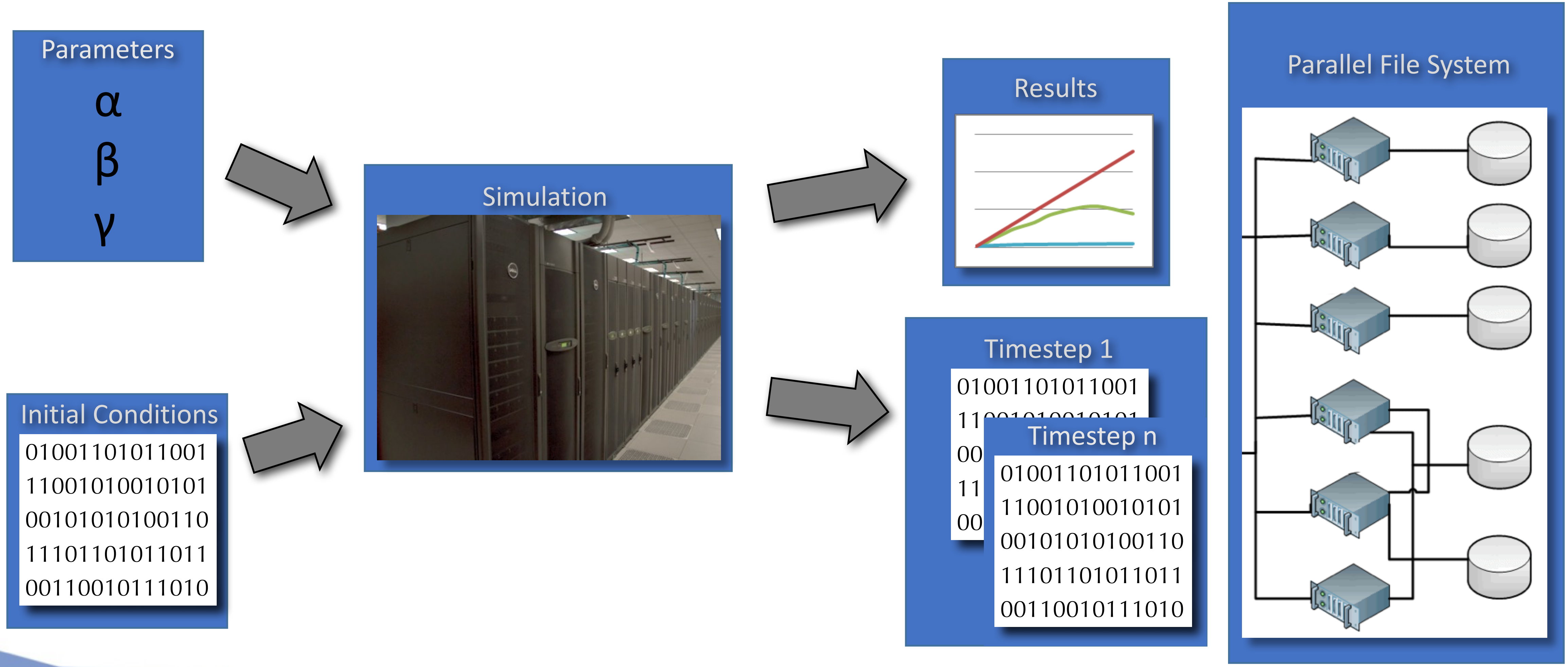
- Current and near-future machines will use processors with many cores
- Each core contains wide vector units: use them for max utilization (e.g., *-AVX512)
- Fortunately the Software-Defined Visualization stack is optimized for such processors!
- Use your preferred rendering method independent of the underlying hardware
  - Performant rasterization
  - Performant ray tracing
  - Visualization and analysis on the simulation machine

TACC



High-Fidelity Visualization Natively on Xeon and Xeon Phi

Typical HPC Workflow

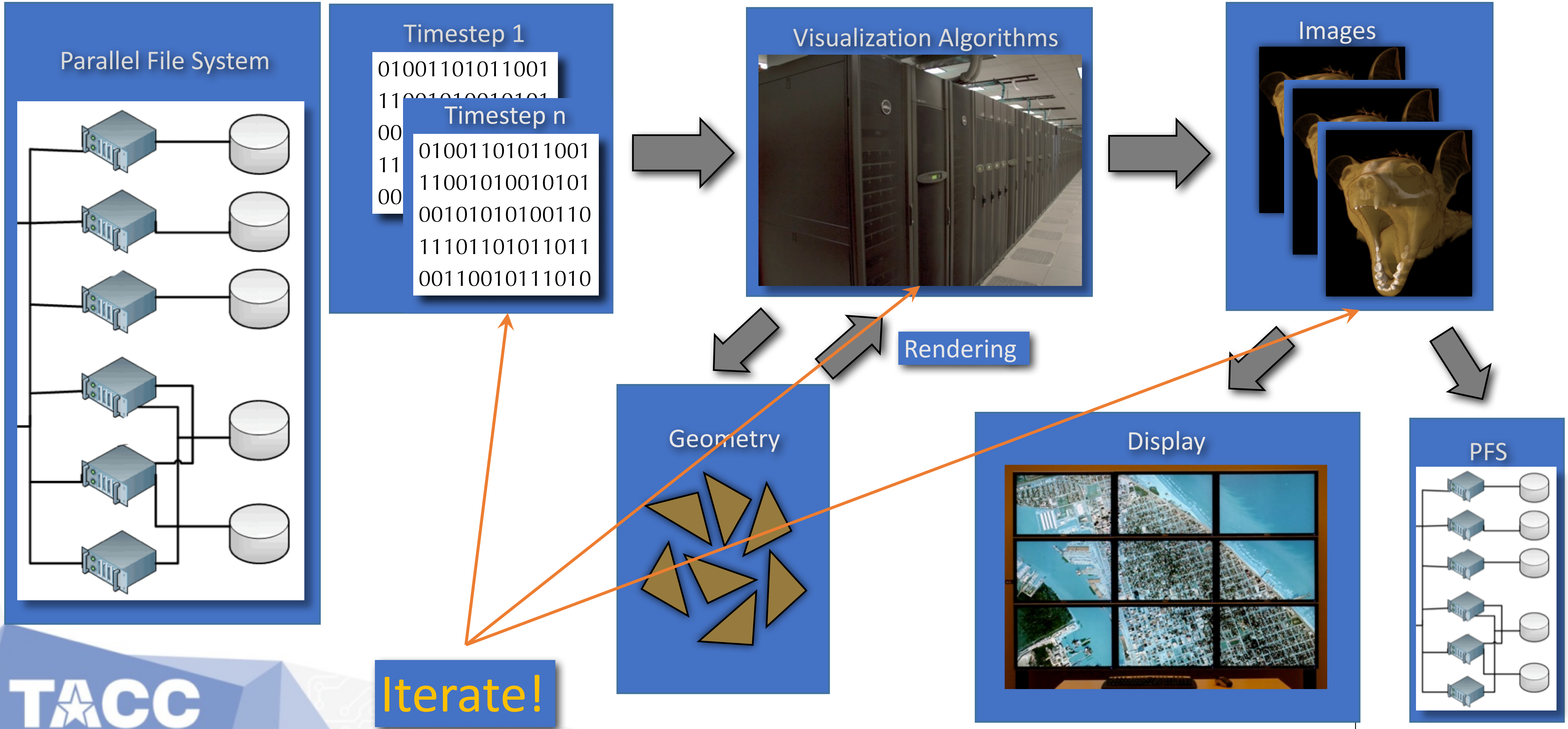


Software-Defined Visualization – Why?

File Size	100 Gbps	10 Gbps	1 Gbps	300 Mbps	54 Mbps
1 GB	< 1 sec	1 sec	10 sec	35 sec	2.5 min
1 TB	~100 sec	~17 min	~3 hours	~10 hours	~43 hours
1 PB	~1 day	~12 days	~121 days	>1 year	~5 years

Increasingly Difficult to Move Data from Simulation Machine

Typical Post-Hoc Visualization Workflow



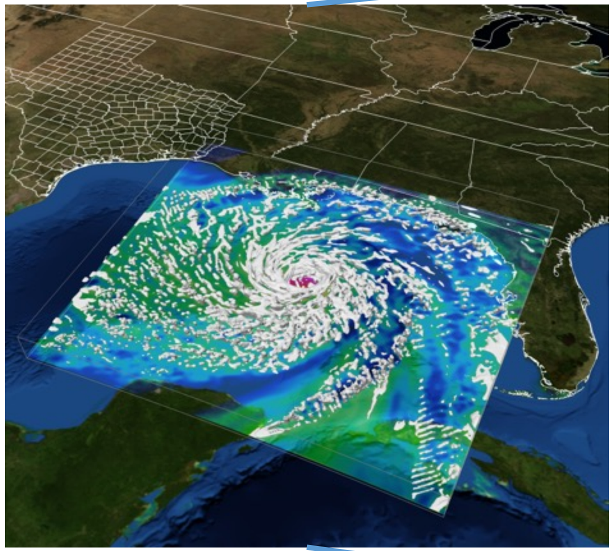
In-Situ Visualization – Why?

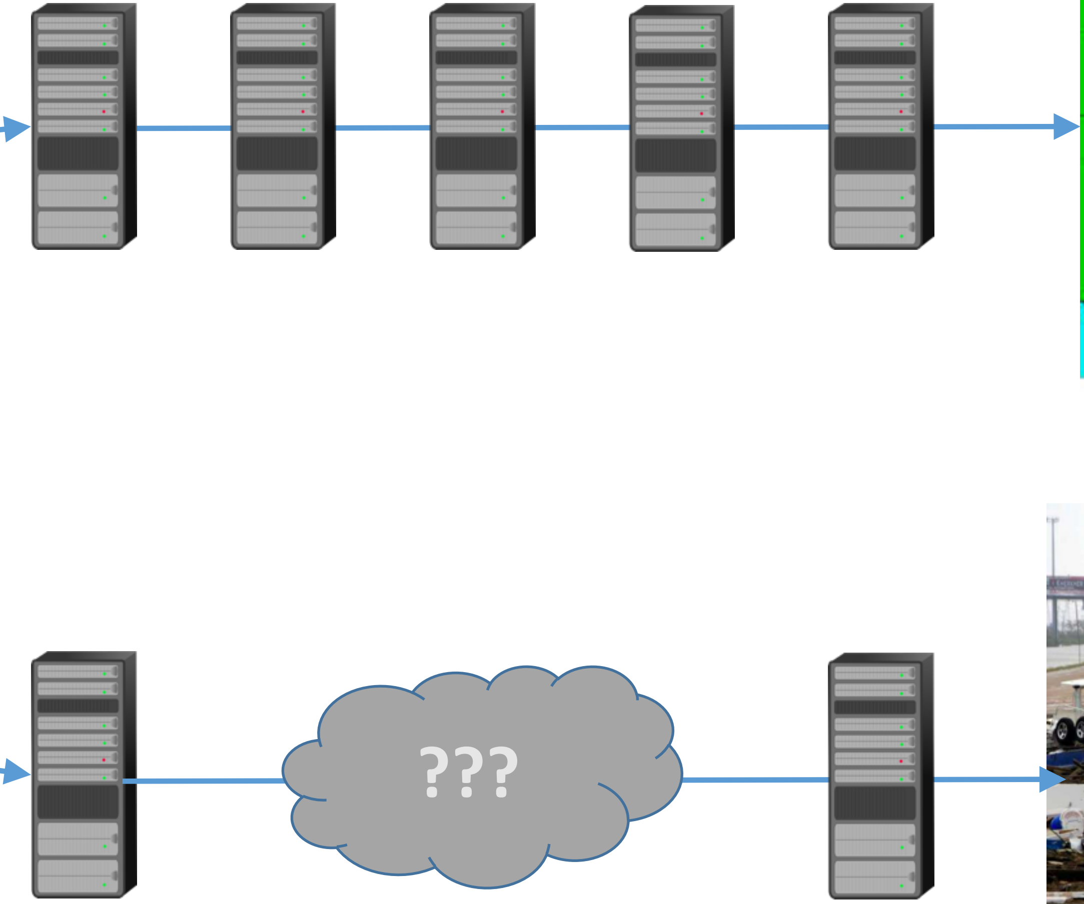
FILE SIZE	1000 GBPS	100 GBPS	10 GBPS	1 GBPS
1 TB	1 sec	~ 10 sec	~ 2 min	~ 17 min
1 PB	~ 17 min	~ 3 hours	~ 1 day	12 days
1 XB	12 days	124 days	3 ½ years	34 years

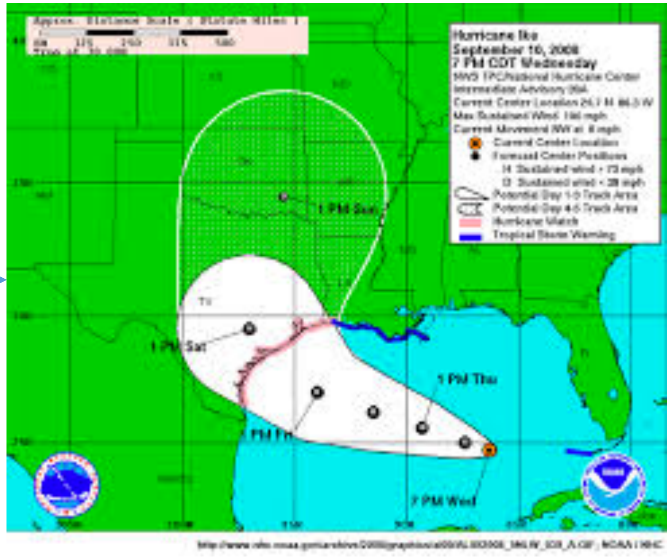
Increasingly Difficult to Write Data from Simulation to Disk

In-Situ Visualization – Why?

High-frequency writes

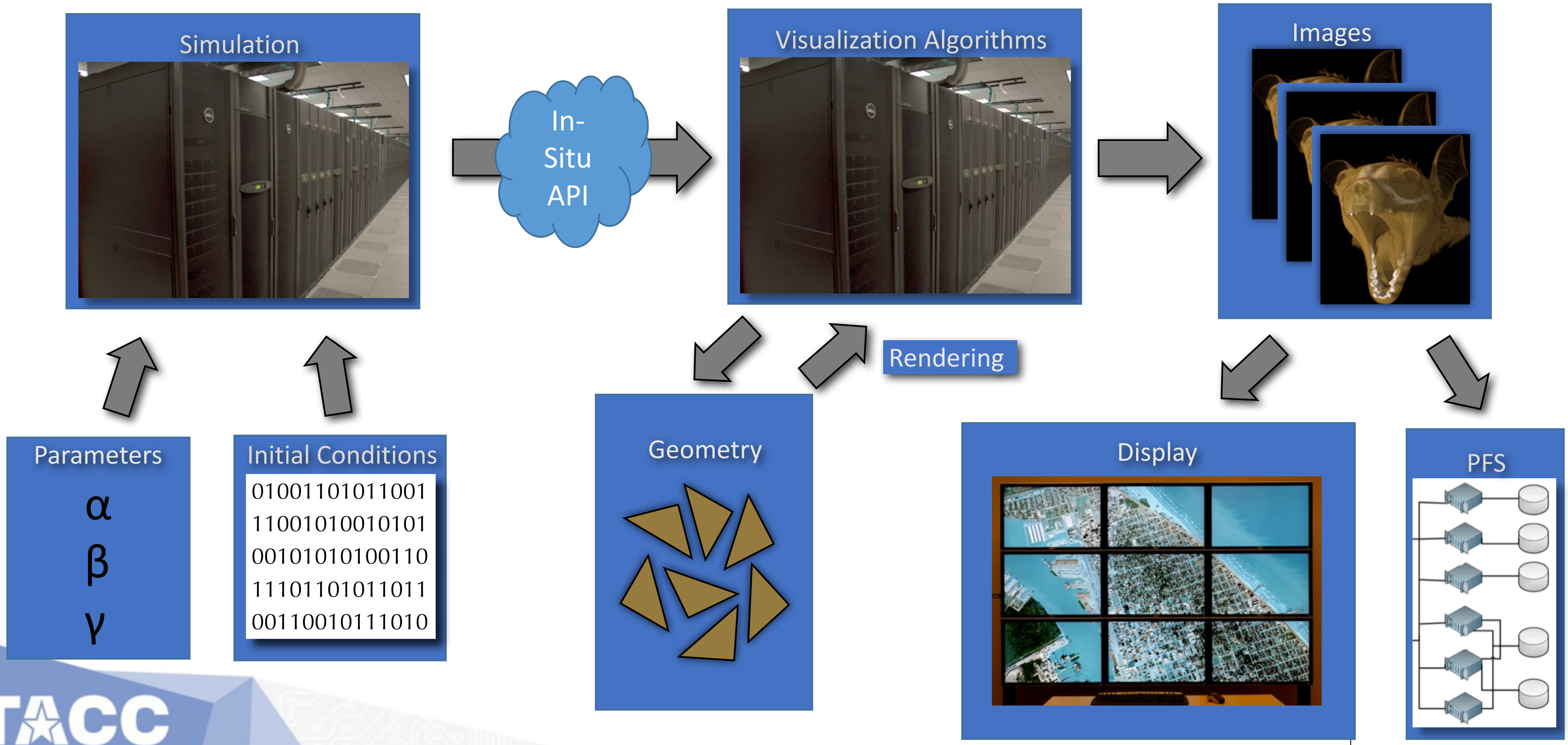


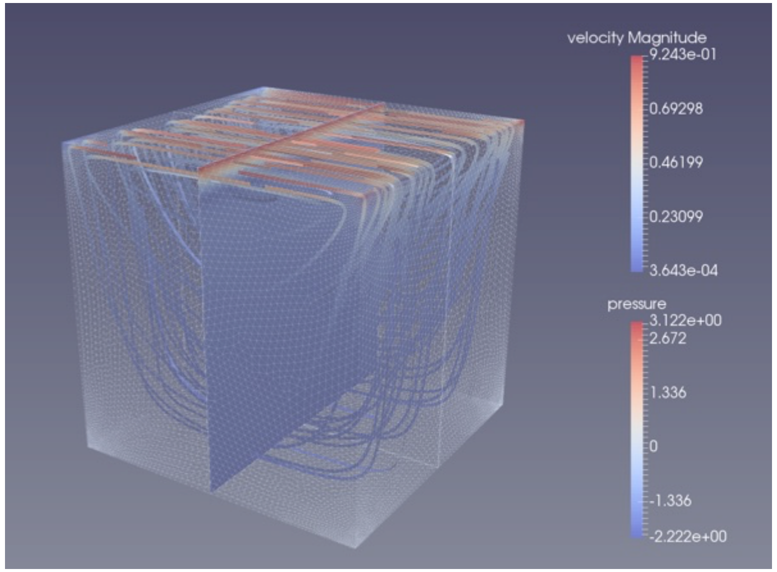


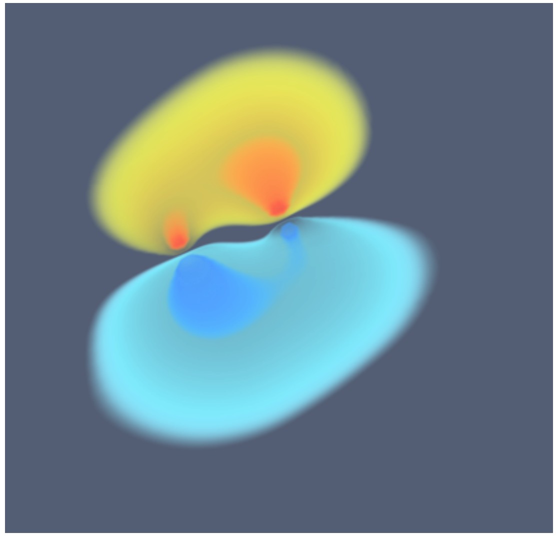


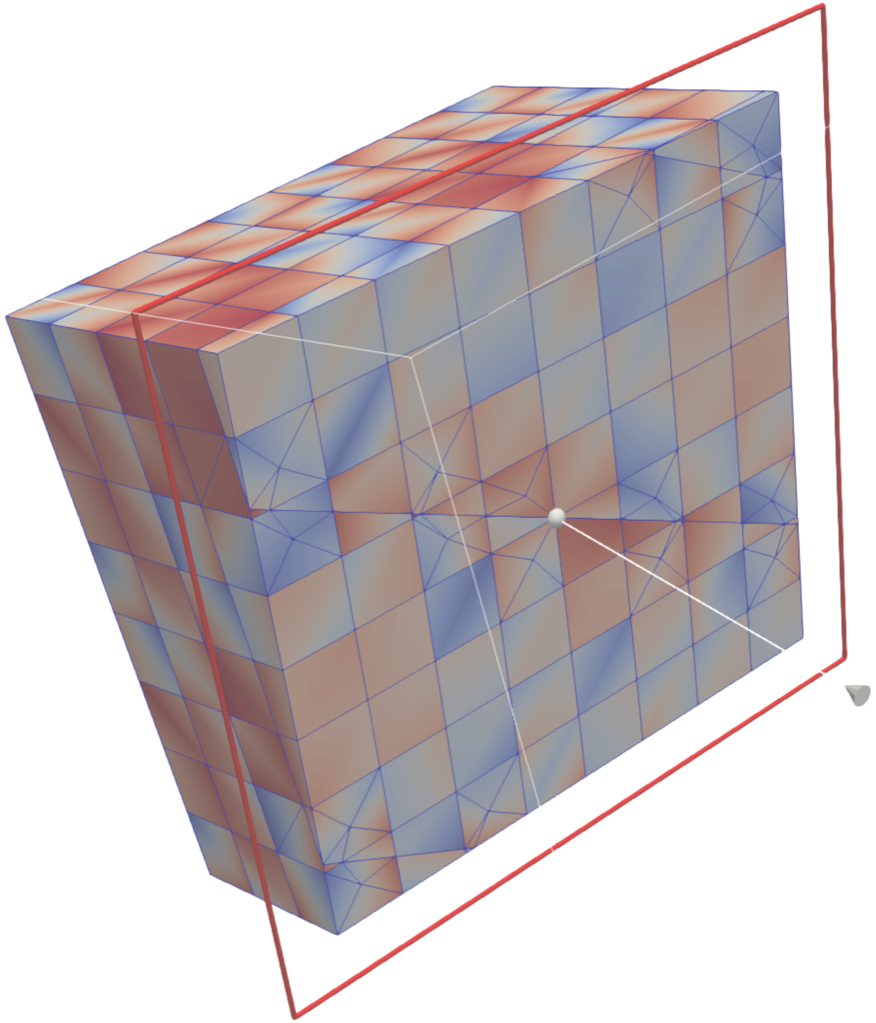
Low-frequency writes

In-Situ Visualization Workflow









In-Situ Software Stack

In Situ Terminology Project (courtesy Ken Moreland, Sandia)

Sandia National Laboratories

Integration Type	Proximity	Access	Division of Execution	Operation Controls	Output Type
Bespoke	Same Memory	Direct Shallow Copy	Time Division	Automatic Adaptive	Subset
Dedicated API	On-node Distinct Memory	Direct Deep Copy	Space Division	Automatic Non-adaptive	Transform
Multi-purpose API	Off-node Same Computing Resource	Indirect		Human-in-the-loop Blocking	Derived Fixed
Inter-position	Distinct Computing Resource			Human-in-the-loop Non-blocking	Derived Proportional
Inspection					

In-Situ Options

courtesy Hank Childs and In-Situ Terminology Group

- VTK-Based APIs
  - ParaView Catalyst – https://www.paraview.org/in-situ/
  - VisIt LibSim - https://www.visitusers.org/index.php?title=Libsim_Batch
  - LLNL ALPINE - https://github.com/Alpine-DAV/alpine
- I/O API
  - ADIOS - https://www.olcf.ornl.gov/center-projects/adios/
- Meta API
  - Sensei - http://www.sensei-insitu.org/
  - Damaris - http://damaris.gforge.inria.fr/doku.php
- Ensemble Post-Process
  - Cinema - http://cinemaviewer.org/

Software-Defined Visualization Stack

- OpenSWR Software Rasterizer
  - openswr.org
  - Performant rasterization for Xeon and Xeon Phi
  - Thread-parallel vector processing
    (previous parallel Mesa3D only has threaded fragments)
  - Support for wide vector instruction sets, particularly AVX2, AVX512
    Integrated into Mesa3D since v12.0 as gallium driver (mesa3d.org)
  - Current rev v17.x installed on Stampede2 and other TACC systems!

- Best Uses
  - OpenGL-based codes
  - Low geometry count, many geometry changes
  - Non-physically-based shading effects

Software-Defined Visualization Stack

- OSPRay Ray Tracer
  - ospray.org
  - Performant ray tracing for Xeon and Xeon Phi incorporating Embree kernels
  - Thread- and wide-vector parallel using Intel ISPC (including AVX512 support)
  - Parallel rendering support via distributed framebuffer

- Best Uses
  - Photorealistic rendering
  - Realistic lighting
  - Realistic material effects
  - Large geometry, few geometry changes
  - Implicit geometry (e.g., molecular ”ball and stick” models)

Software-Defined Visualization Stack

- GraviT Scheduling Framework
  - tacc.github.io/GraviT/
  - Large-scale, data-distributed ray tracing
    (uses OSPRay for rendering engine target)
  - Parallel rendering support via distributed ray scheduling
  - Funded by US NSF awards ACI-1339863, ACI-1339840, ACI-1339881
    program officers Dan Katz and Rajiv Ramnath

- Best Uses
  - Large distributed data
  - Data outside of renderer control
  - Incoherent ray-intensive sampling (e.g., global illumination approximations)

SDVIS PERFORMANCE UPDATE

Performance slides courtesy Jim Jeffers, Intel Corp.

Notices and Disclaimers

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.

The cost reduction scenarios described are intended to enable you to get a better understanding of how the purchase of a given Intel based product, combined with a number of situation-specific variables, might affect future costs and savings. Circumstances will vary and there may be unaccounted-for costs related to the use and deployment of a given product. Nothing in this document should be interpreted as either a promise of or contract for a given level of costs or cost reduction.

Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice Revision #20110804.

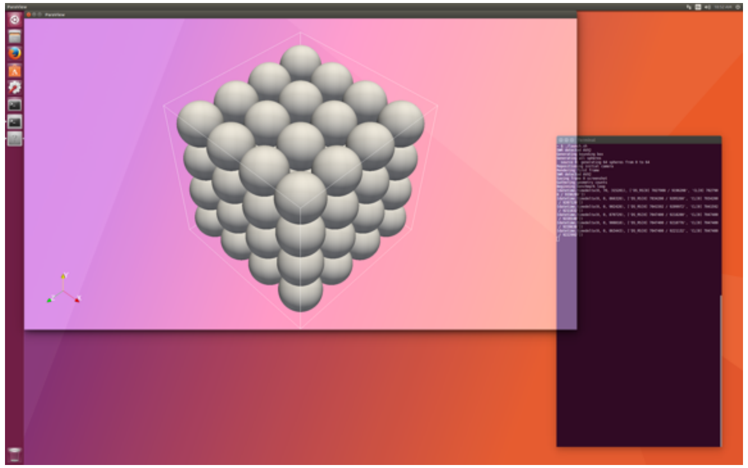
No computer system can be absolutely secure.

Intel® Advanced Vector Extensions (Intel® AVX)* provides higher throughput to certain processor operations. Due to varying processor power characteristics, utilizing AVX instructions may cause a) some parts to operate at less than the rated frequency and b) some parts with Intel® Turbo Boost Technology 2.0 to not achieve any or maximum turbo frequencies. Performance varies depending on hardware, software, and system configuration and you can learn more at http://www.intel.com/go/turbo.

Intel processors of the same SKU may vary in frequency or power as a result of natural variability in the production process.

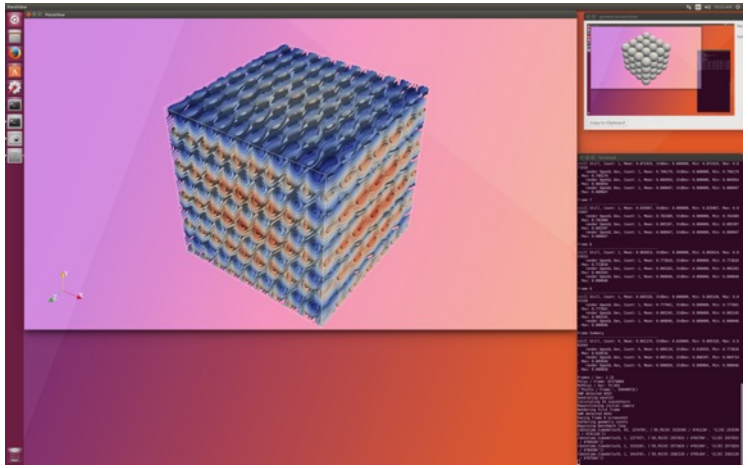
SPEC, SPECfp and SPECint are registered trademarks of the Standard Performance Evaluation Corporation (SPEC).

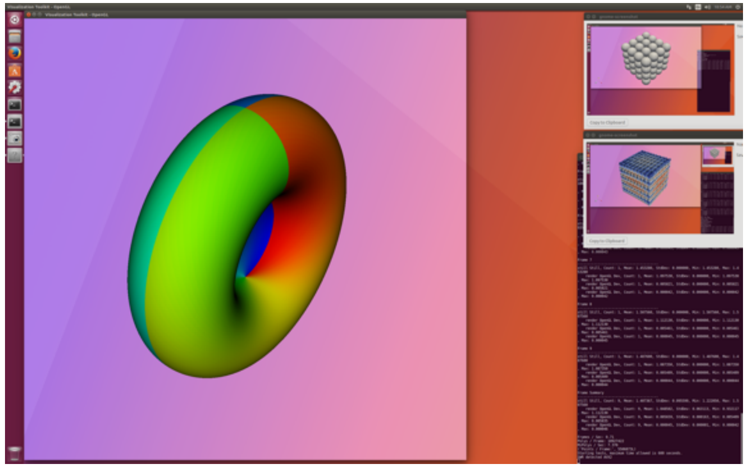
OPENGL (OpenSWR) benchmarks



manyspheres.py
67 MiPolys

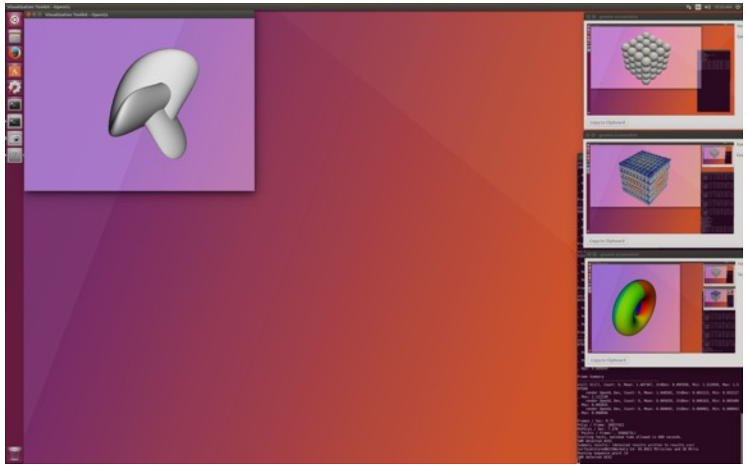
wavelets.py
11 MiPolys



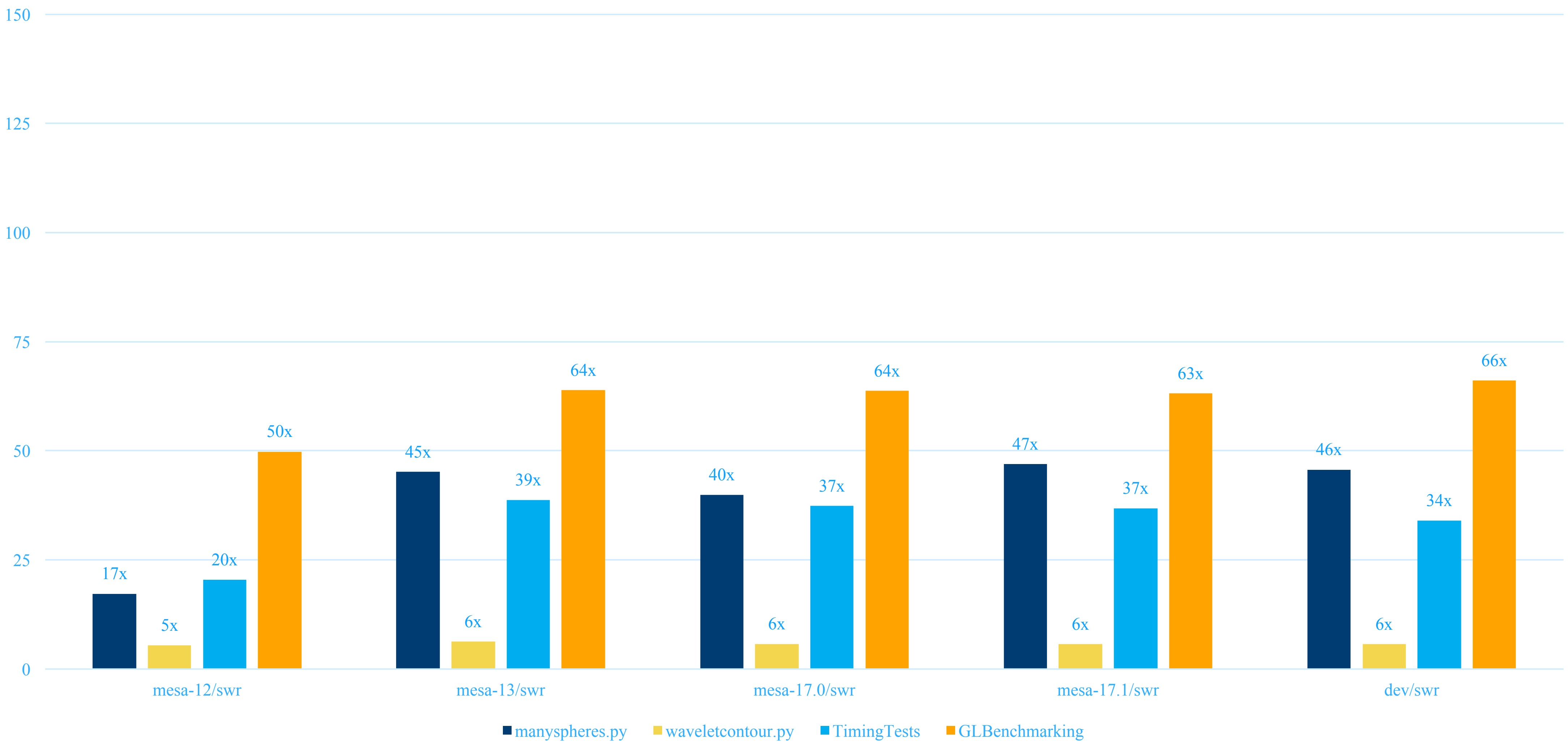


TimingTests
30 MiTris

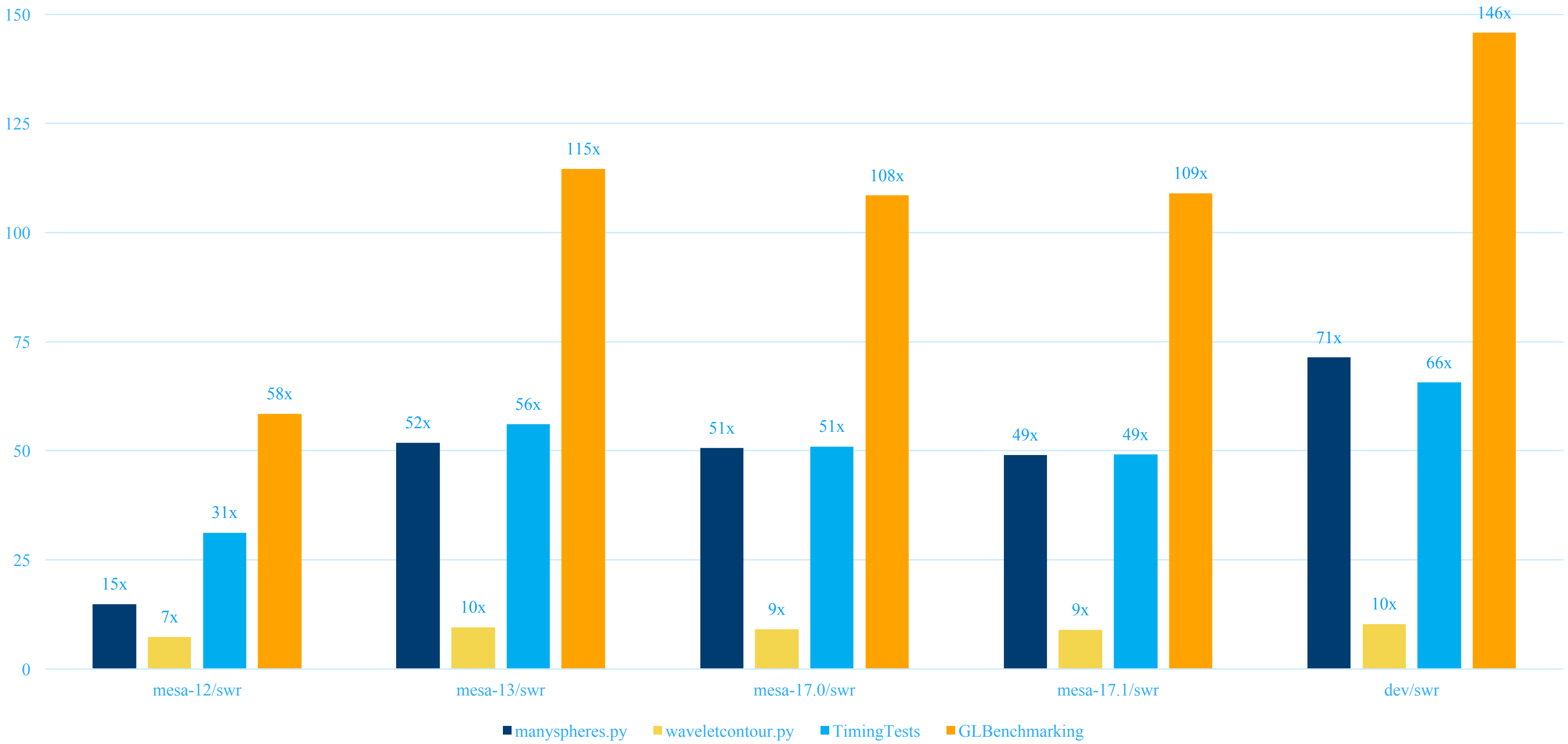
GLBenchmarking
30MiTris



INTEL® Xeon® E5 v4 OPENSWR/LLVMPIPE PERFORMANCE RATIO

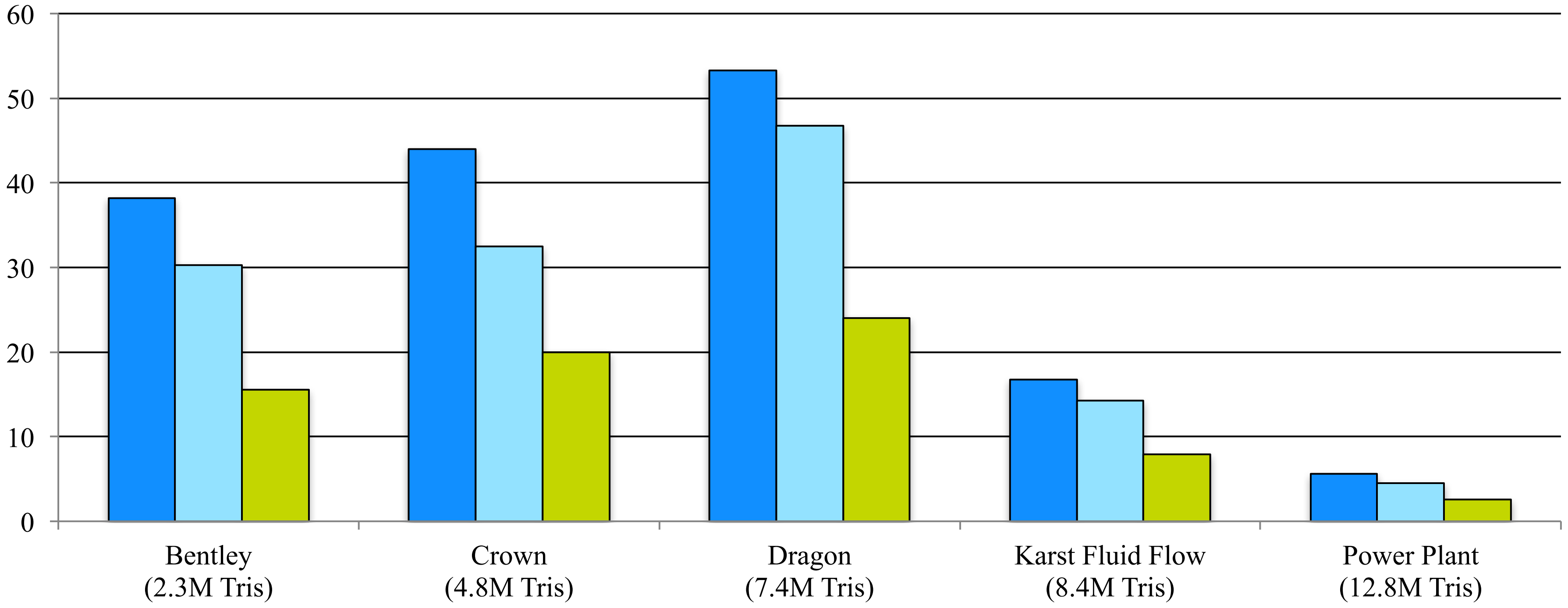


INTEL® Xeon Phi™ 7250 OPENSWR/LLVMPIPE PERFORMANCE RATIO



Performance: Embree vs. NVIDIA* OptiX*

Frames Per Second (Higher is Better), 1024x1024 image resolution



Intel® Xeon® E5-2699 v4 Processor
2 x 22 cores, 2.2 GHz

Intel® Xeon Phi™ 7250 Processor
68 cores, 1.4 GHz

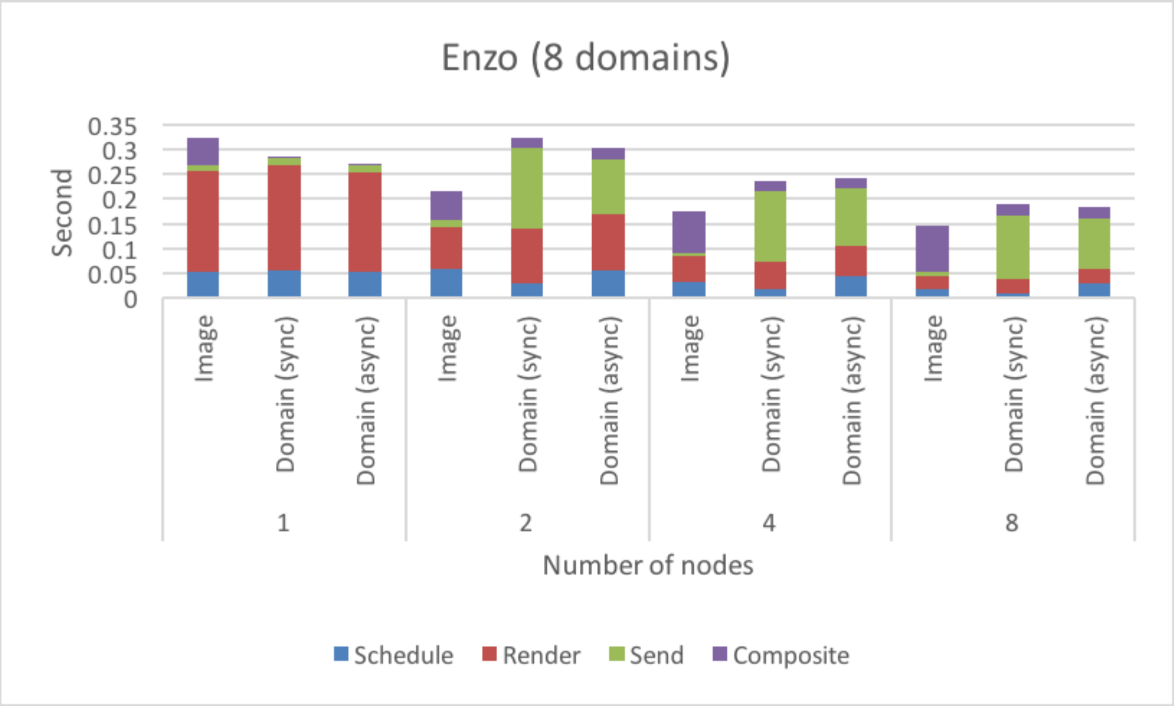
NVIDIA Tesla P100 Coprocessor
PCIe, 16 GB RAM

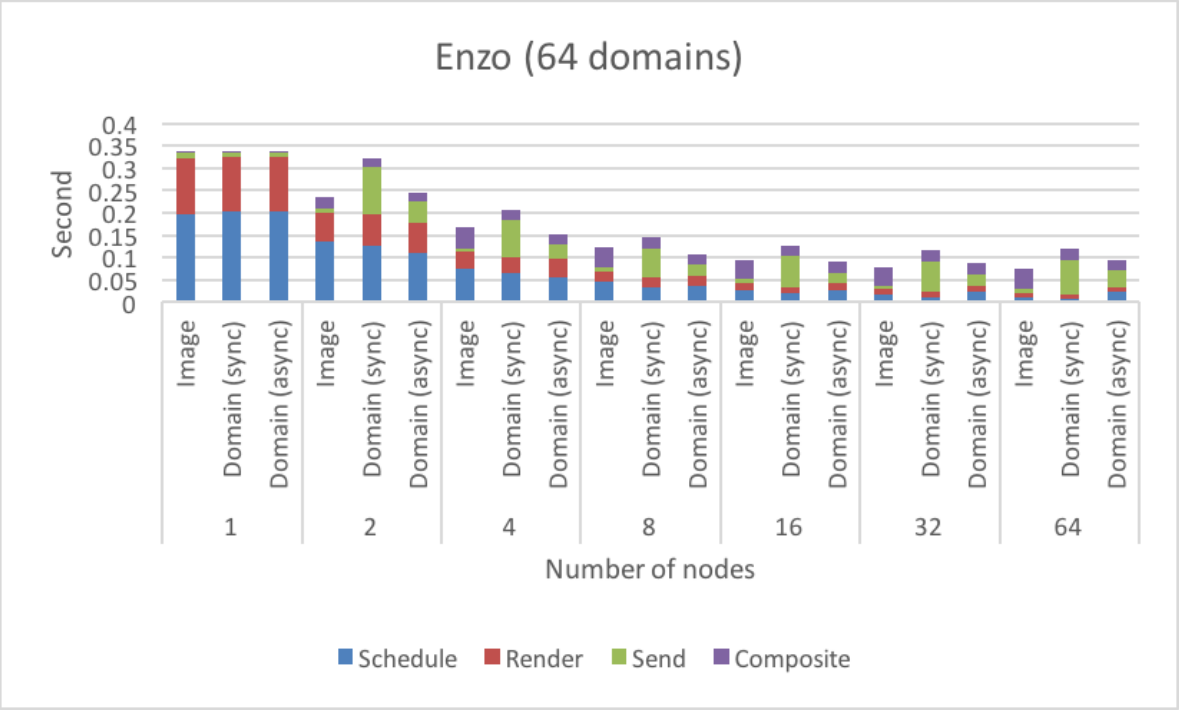
Embree 2.16.1, Intel® C++ Compiler 17.0.2, Intel® SPMD Program Compiler (Intel® ISPC) 1.9.1

NVIDIA* OptiX* 4.0.2, CUDA* 8.0.44

Source: Intel

GraviT Distributed RT Performance





Stampede2 Early Science:
IXPUG In-Situ Workshop and Hackathon

Workshop Goals

- Bring simulation developers and visualization developers together with explicit expectation to develop code
- Organize respondents into "tiger teams" of sim + vis folks
  - Get early system access to handle builds, shake out installs
  - Maximize usefulness of in-person cycles
- Build community, identify best practices, advance adoption

IXPUG In-Situ Workshop Participation

Forty-two registered participants

Fourteen simulation teams

Seventeen institutions

Five countries

- Argonne National Laboratory
- Cambridge University
- Federal University of Rio de Janeiro
- Intel Corporation
- Intelligent Light
- Kitware Inc.
- Lawrence Livermore National Laboratory
- Los Alamos National Laboratory
- SCI Institute
- SURVICE Engineering
- Texas Advanced Computing Center
- University of Chicago
- University of Oregon
- University of Stuttgart
- University of Tennessee
- University of Texas – ECE
- University of Texas – ICES

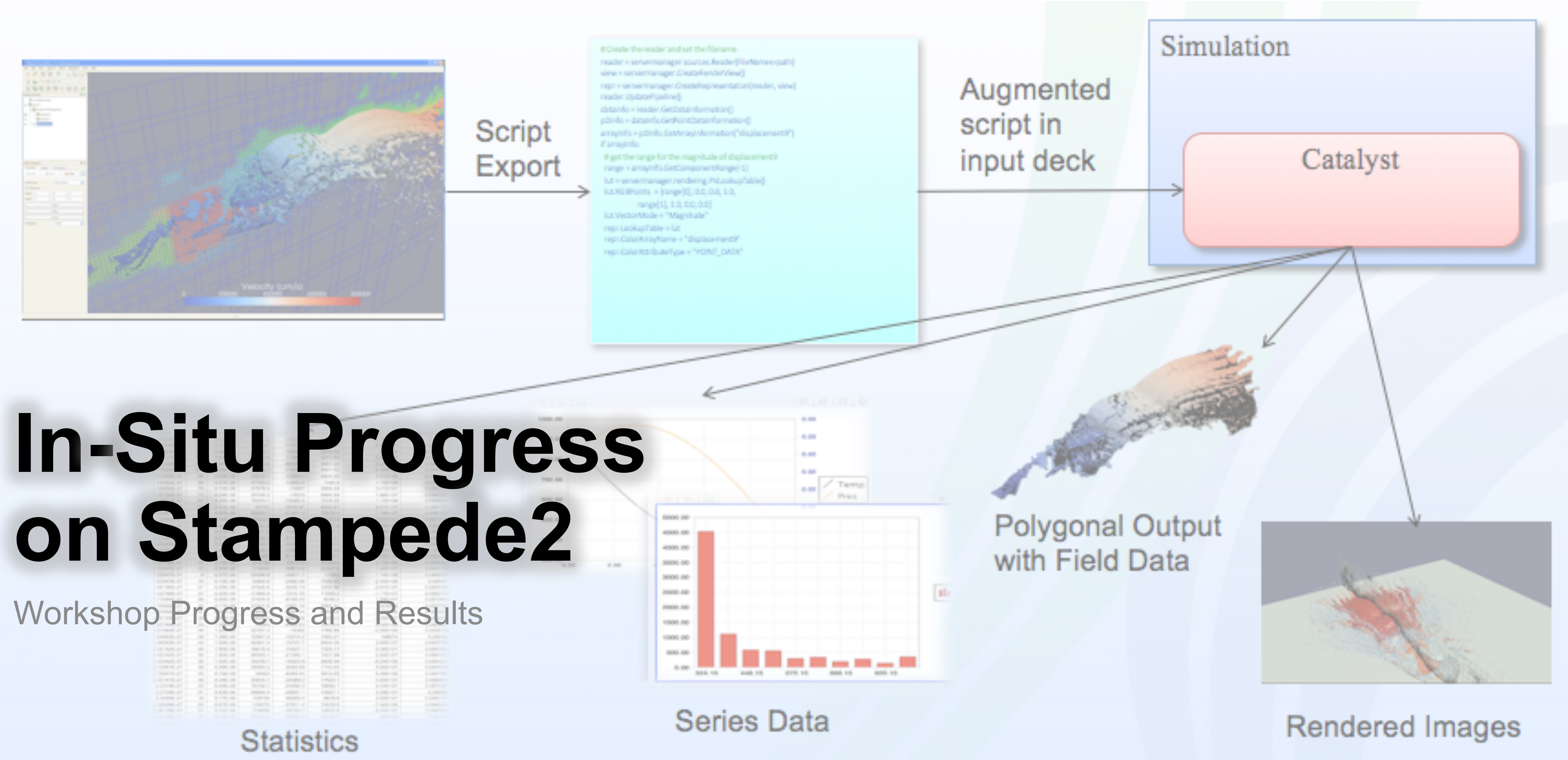
TACC

Workshop "Hackathon" Format

- Three days of worktime over four days
  - Monday afternoon – Thursday morning
- In-Situ Environment Update
  - In-Situ Terminology Project presentation
  - Stampede2 capabilities
  - ParaView Catalyst and VisIt LibSim deep-dives
  - In-Situ community lightning talks
- "Tiger Team" breakouts each day
  - Monday – sync, planning, system access
  - Tuesday - hacking
  - Wednesday – hacking
  - Thursday – lessons learned and next seteps

In-Situ Progress on Stampede2

Workshop Progress and Results



Lessons Learned

- Good news: First users on Stampede2!
- Bad news: First users on Stampede2 …

- Stampede2 rollout presented unique logistical challenge
  - Pre-workshop access to Stampede-KNL
  - Stampede2 access Thursday before workshop
  - Stampede-KNL decommissioned Friday before workshop
  - Updated compiler and MPI required recompile of entire vis stack
  - Users had the weekend to update their builds …
    assuming all prereqs present …

Lessons Learned

- Gathering people together worked!
  - Had to convince people to pause hacking for free food and beer
- In-Situ capabilities established and expanded
  - GR-CHOMBO, ALPINE+WALLS, LibMesh, RHEA
- Issues identified and solutions iterated
  - VTK zero-copy, AMR data, OSPRay and OpenSWR integrations
- Impromptu projects undertaken
  - Villi simulation vis, LAMMPS + Sensei + OSPRay, KNL optimizations
- Significant demand for additional workshops
  - Broaden reach into additional communities (DOE, DARPA, etc)

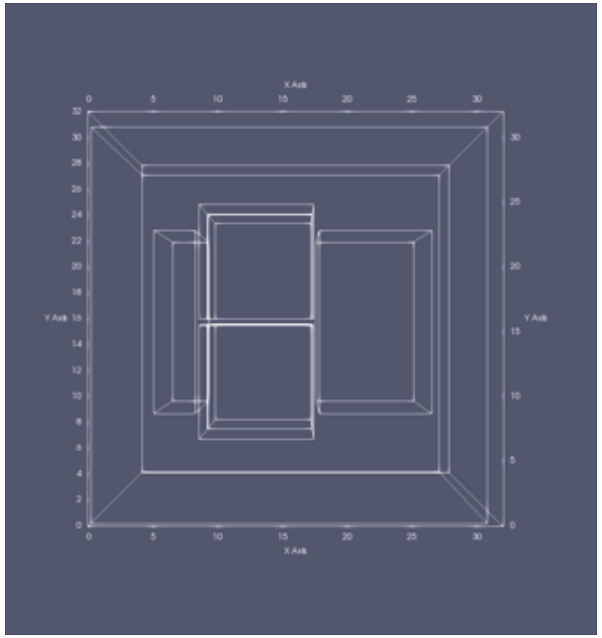
TACC

Annual hackathon + BOFs at ISC and SC

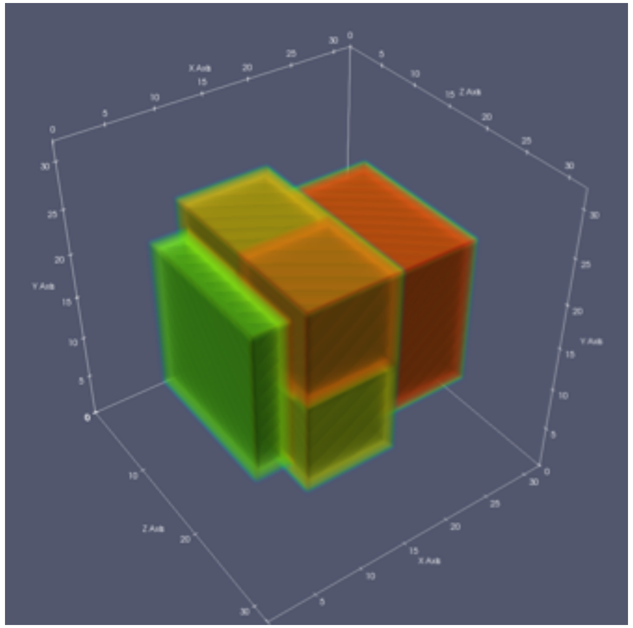
GR-CHOMBO + Catalyst

David Daverio (Cambridge), Kacper Kornet (Cambridge),
Dave DeMarle (Kitware), Andy Bauer (Kitware)

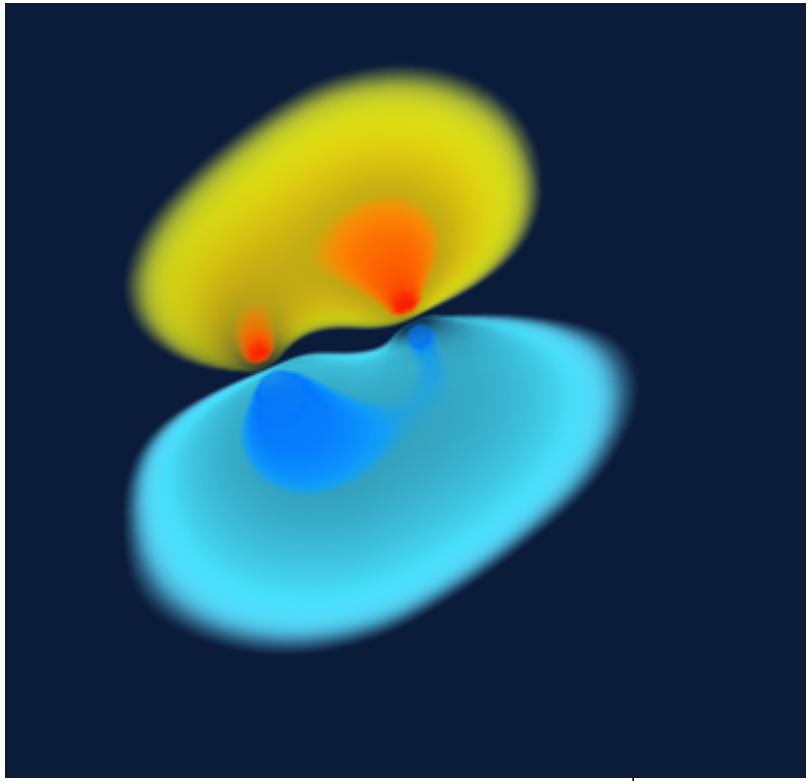
first light



recent



goal : match pvOSPRay to post-hoc LIGO images



ALPINE + WALLS

Matt Larsen (LLNL), David Daverio (Cambridge), Kacper Kornet (Cambridge)

Integration

```c
conduit_node_set_path_char8_str(mesh,"topologies/mesh/type", "uniform");
conduit_node_set_path_char8_str(mesh,"topologies/mesh/coordset", "coords");
double dim_x = extents[1] - extents[0]+1;
double dim_y = extents[3] - extents[2]+1;
double dim_z = extents[5] - extents[4]+1;
conduit_node_set_path_char8_str(mesh,"coordsets/coords/type","uniform");
conduit_node_set_path_float64(mesh,"coordsets/coords/dims/i",dim_x);
conduit_node_set_path_float64(mesh,"coordsets/coords/dims/j",dim_y);
conduit_node_set_path_float64(mesh,"coordsets/coords/dims/k",dim_z);

conduit_node_set_path_float64(mesh,"coordsets/coords/origin/x",(double)extents[0]);
conduit_node_set_path_float64(mesh,"coordsets/coords/origin/y",(double)extents[2]);
conduit_node_set_path_float64(mesh,"coordsets/coords/origin/z",(double)extents[4]);

conduit_node_set_path_float64(mesh,"coordsets/coords/spacing/dx",spacing);
conduit_node_set_path_float64(mesh,"coordsets/coords/spacing/dy",spacing);
conduit_node_set_path_float64(mesh,"coordsets/coords/spacing/dz",spacing);
```

```c
conduit_node* mesh = conduit_node_create();
conduit_node_set_path_int32(mesh,"state/time", 0);
conduit_node_set_path_int32(mesh,"state/cycle", 0);
conduit_node_set_path_int32(mesh,"state/domain", 0);
conduit_node_set_path_char8_str(mesh,"state/info", "walls");

conduit_node_set_path_char8_str(mesh,"topologies/mesh/type", "uniform");
conduit_node_set_path_char8_str(mesh,"topologies/mesh/coordset", "coords");
double dim_x = extents[1] - extents[0]+1;
double dim_y = extents[3] - extents[2]+1;
double dim_z = extents[5] - extents[4]+1;
conduit_node_set_path_char8_str(mesh,"coordsets/coords/type","uniform");
conduit_node_set_path_float64(mesh,"coordsets/coords/dims/i",dim_x);
conduit_node_set_path_float64(mesh,"coordsets/coords/dims/j",dim_y);
conduit_node_set_path_float64(mesh,"coordsets/coords/dims/k",dim_z);

conduit_node_set_path_float64(mesh,"coordsets/coords/origin/x",(double)extents[0]);
conduit_node_set_path_float64(mesh,"coordsets/coords/origin/y",(double)extents[2]);
conduit_node_set_path_float64(mesh,"coordsets/coords/origin/z",(double)extents[4]);

conduit_node_set_path_float64(mesh,"coordsets/coords/spacing/dx",spacing);
conduit_node_set_path_float64(mesh,"coordsets/coords/spacing/dy",spacing);
conduit_node_set_path_float64(mesh,"coordsets/coords/spacing/dz",spacing);

double (* P1)[N2+2][N3+2][N4+2] = (double (*)[N2+2][N3+2][N4+2]) field;
double *out_data = malloc(sizeof(double)*num_points);
//
// strip out the ghost data
//
int index = 0;
for(int i=1;i<N4+2;i++)
{
  for(int j=1;j<N3+2;j++)
  {
    for(int k=1;k<N2+2;k++)
    {
      out_data[index] = P1[0][k][j][i];
      index++;
    }
  }
}

conduit_node_set_path_char8_str(mesh,"fields/data/association","vertex");
conduit_node_set_path_char8_str(mesh,"fields/data/type","scalar");
conduit_node_set_path_char8_str(mesh,"fields/data/topology","mesh");
conduit_node_set_path_float64_ptr(mesh, "fields/data/values", out_data, num_points);
free(out_data);
return mesh;
```

https://github.com/Alpine-DAV/alpine

ALPINE + WALLS

Matt Larsen (LLNL), David Daverio (Cambridge), Kacper Kornet (Cambridge)

Isosurfaces

- VTK-m currently does not include filters in the library
  - So, I hacked on into the rendering library

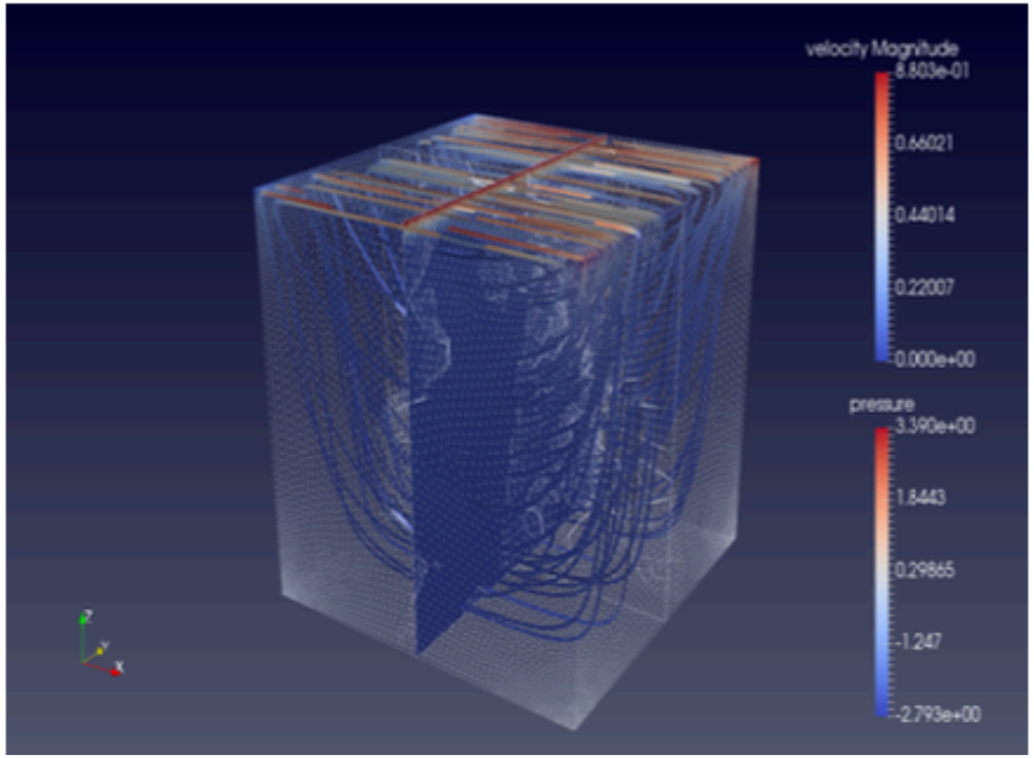
Cycle 1

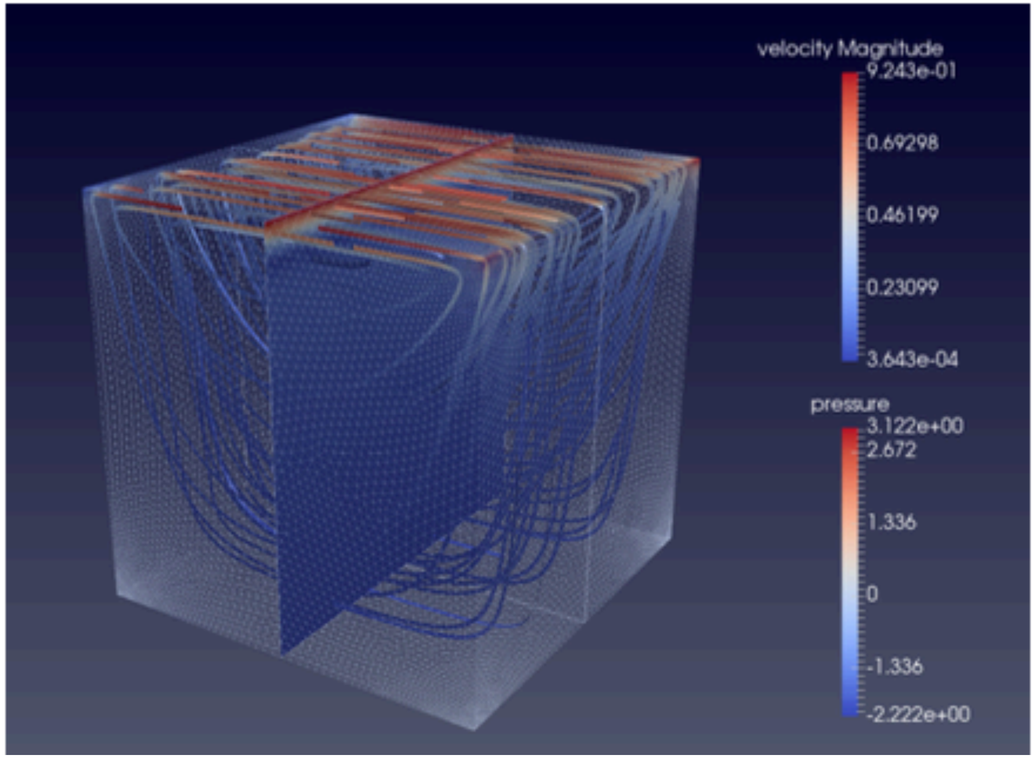
Cycle 230

https://github.com/Alpine-DAV/alpine

libMesh + Catalyst

Jose Camata (Rio de Janeiro), Dave DeMarle (Kitware), Andy Bauer (Kitware)

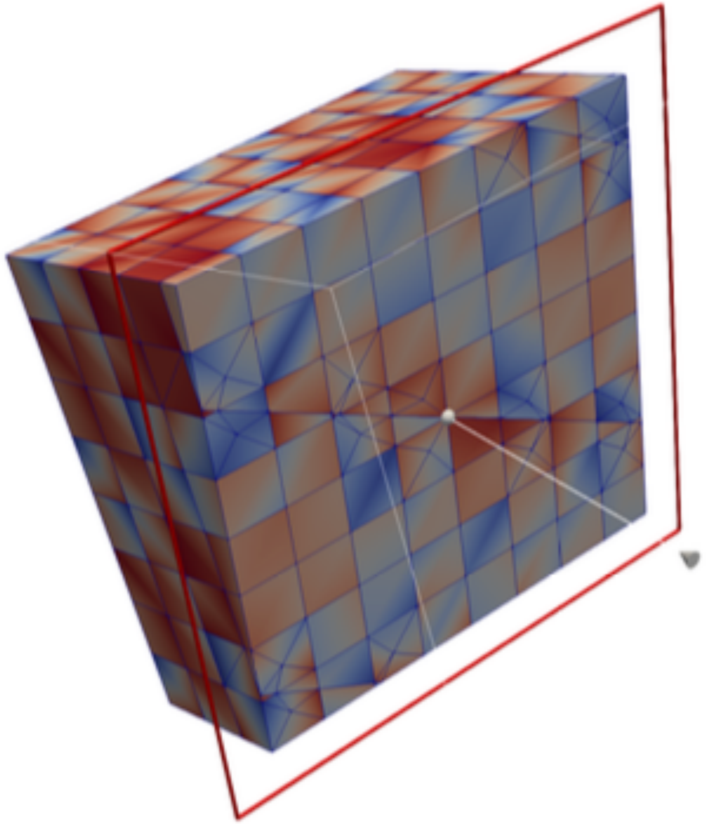




RHEA + Catalyst

Johann Rudi (ICES), Dave DeMarle (Kitware), Andy Bauer (Kitware)

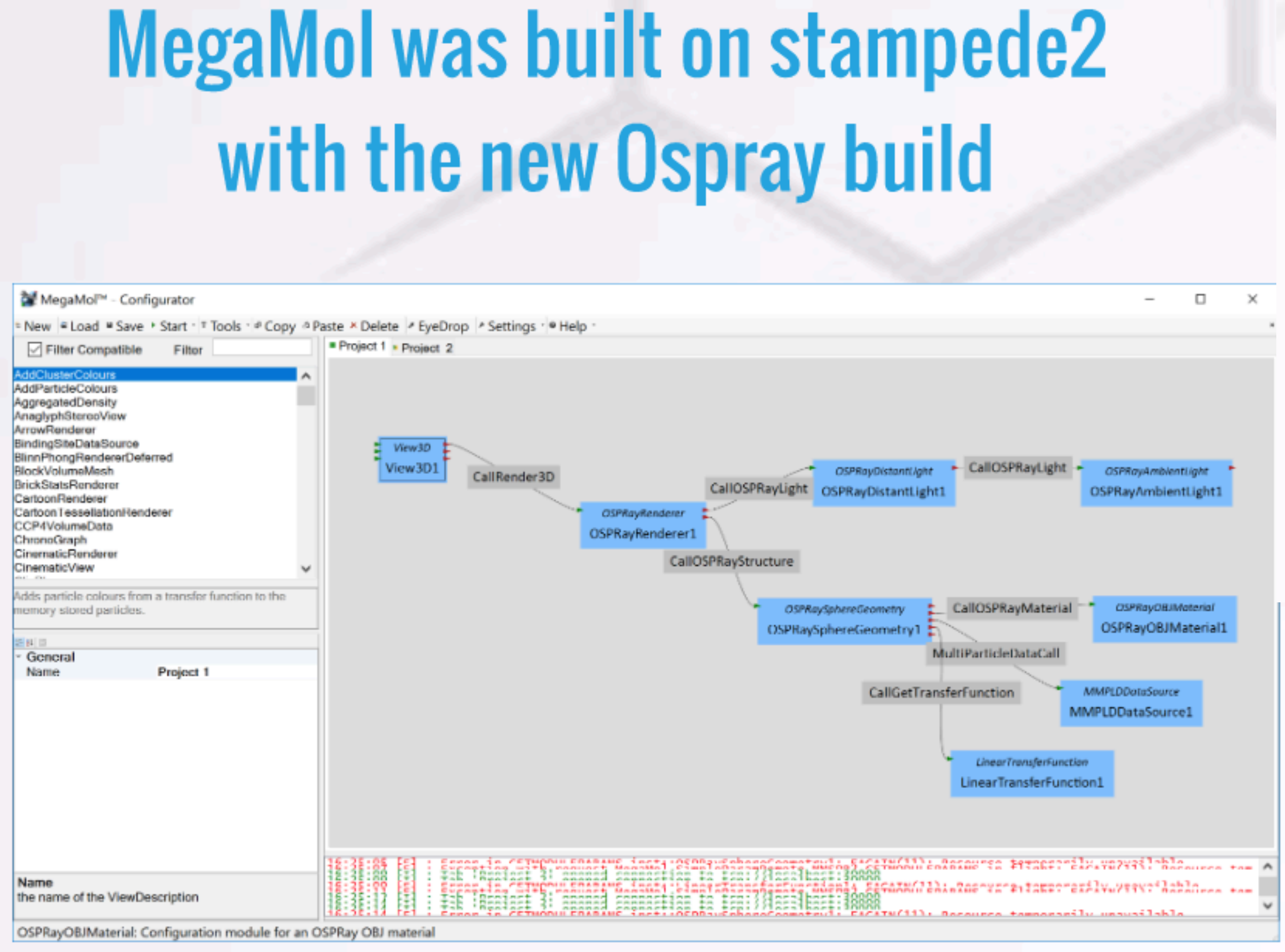
- Skeleton integration for mantle convection simulations
- Leverage zero-copy array support in VTK
- KNL simulation tuning

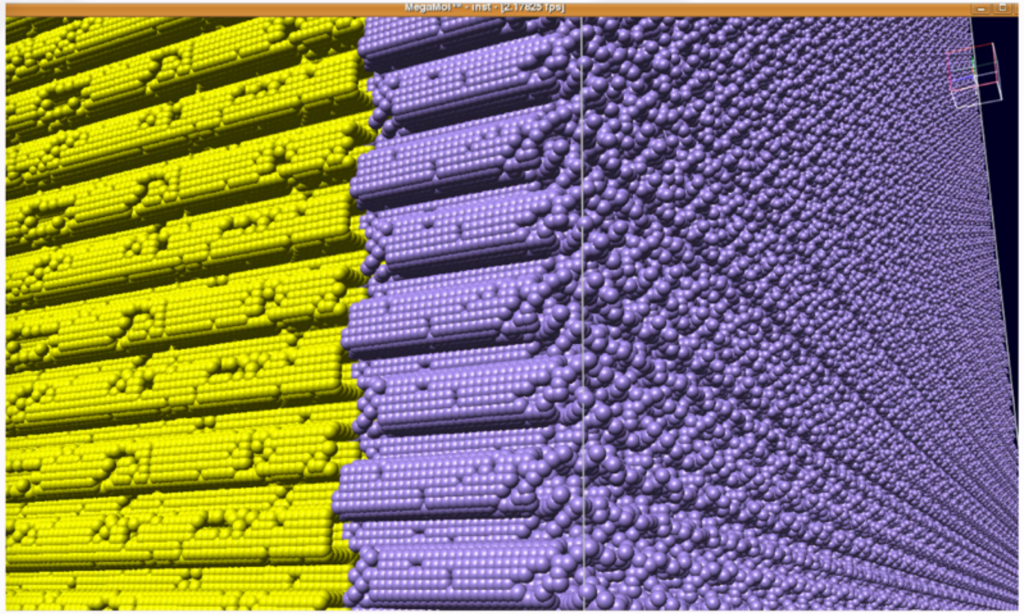


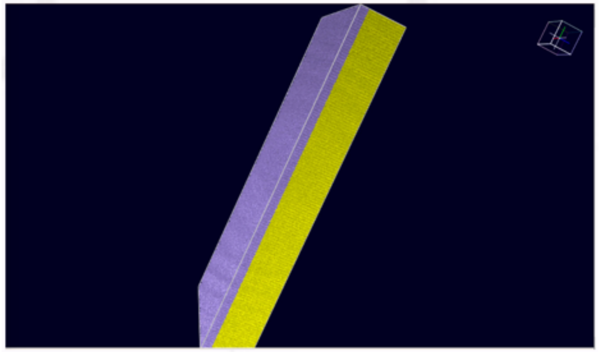
RHEA Catalyst output

Intestinal Villi Simulation

Teodora Szasz (Chicago), Ayat Mohammed (Virginia Tech), Anne Bowen (TACC)

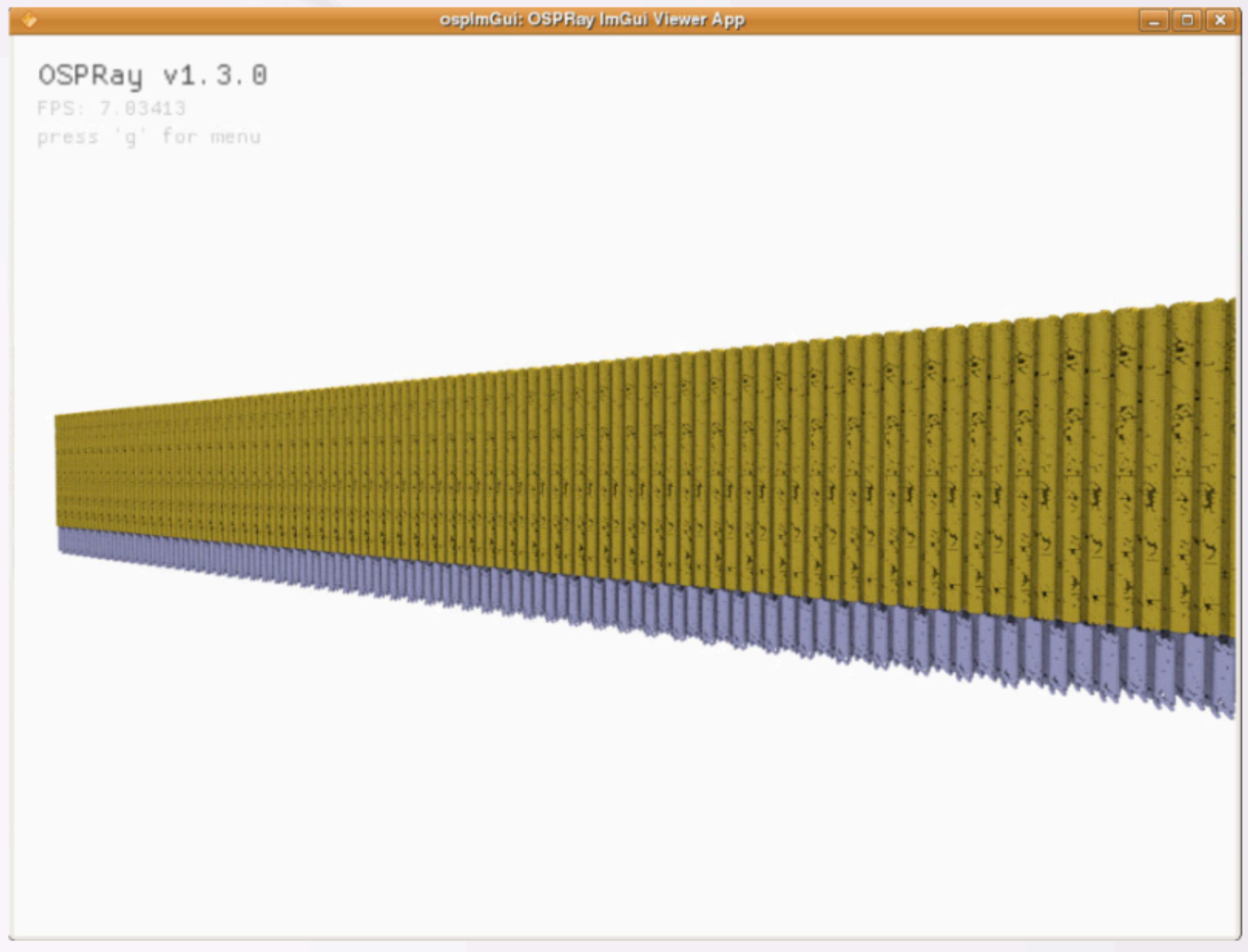


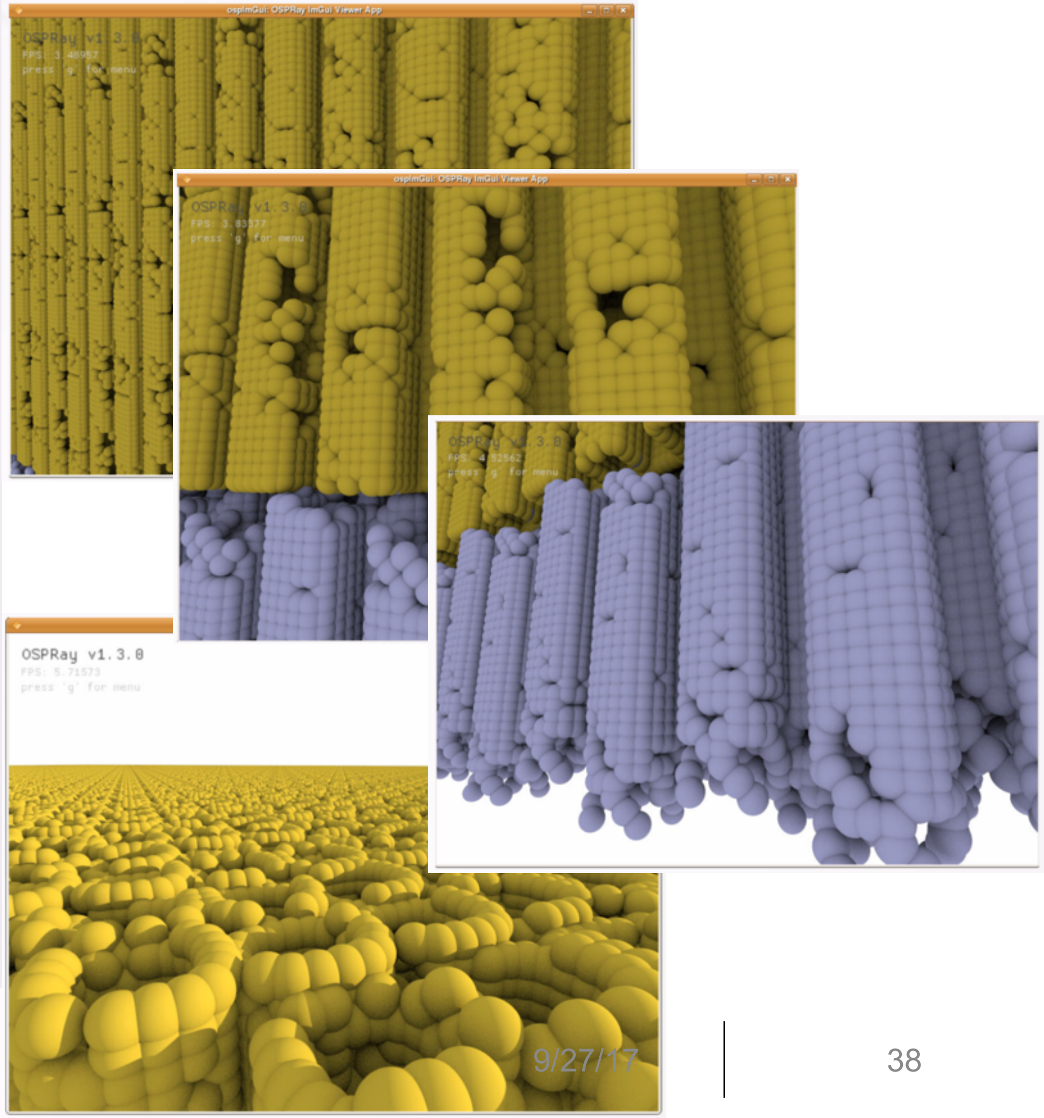




Intestinal Villi Simulation

Teodora Szasz (Chicago), Ayat Mohammed (Virginia Tech), Anne Bowen (TACC)





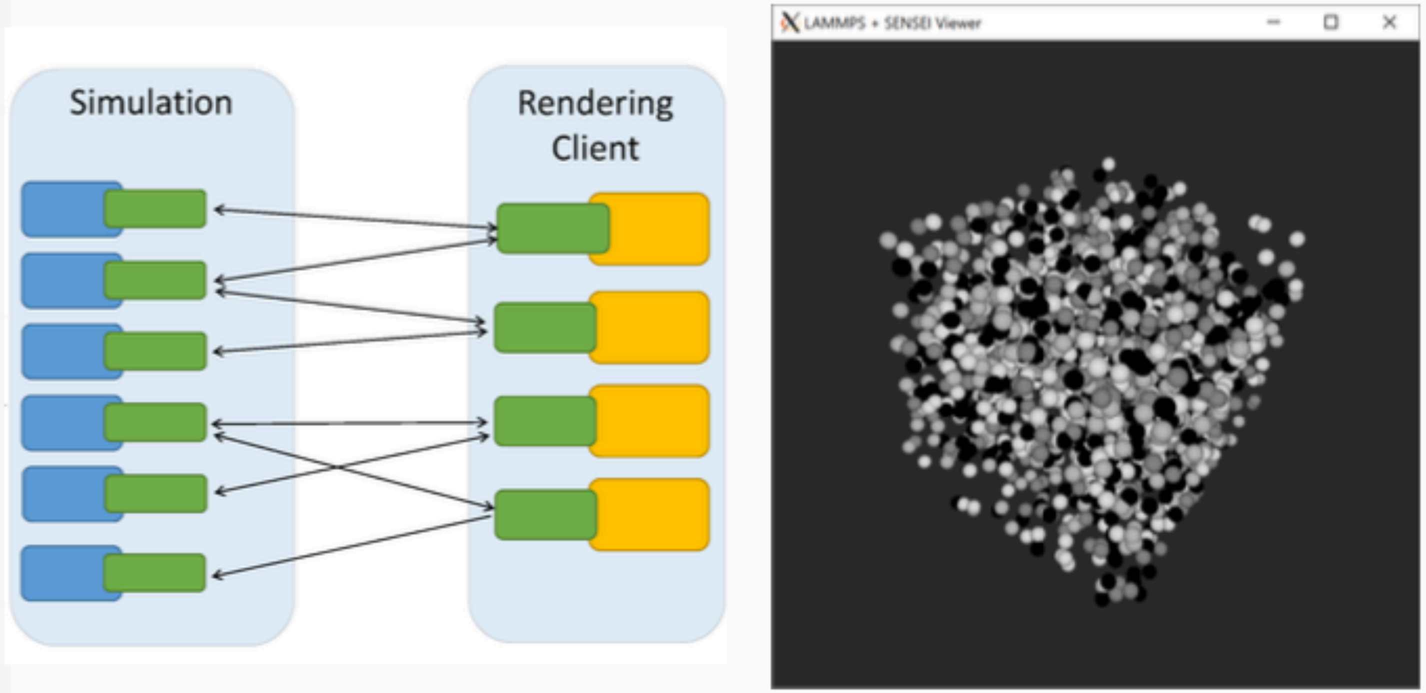
LAMMPS + Sensei + OSPRay

Will Usher (Intel), Aaron Knoll (SCI), Silvio Rizzi (Argonne), Joe Insley (Argonne)

Client Viewer in OSPRay

Connect to simulation running a server with SENSEI to query data

Pull data back to distributed OSPRay client app running using OSPRay's distributed device to provide an interactive viewer of the latest timestep

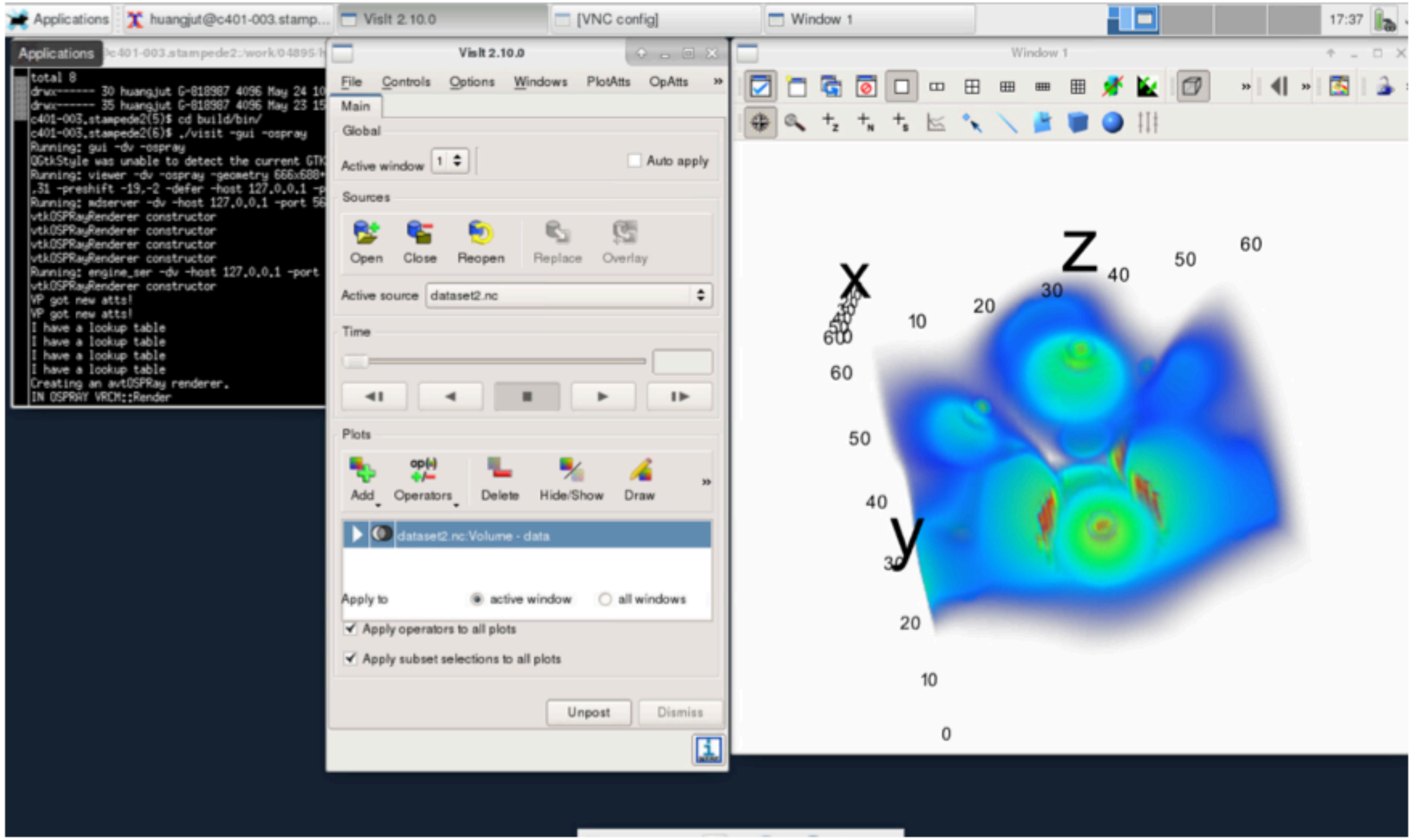


https://xgitlab.cels.anl.gov/fl/lammps_sensei_ospray

VisIt + LibSim + OSPRay

Alok Hota (Tennessee), Jian Huang (Tennessee), Hank Childs (Oregon)

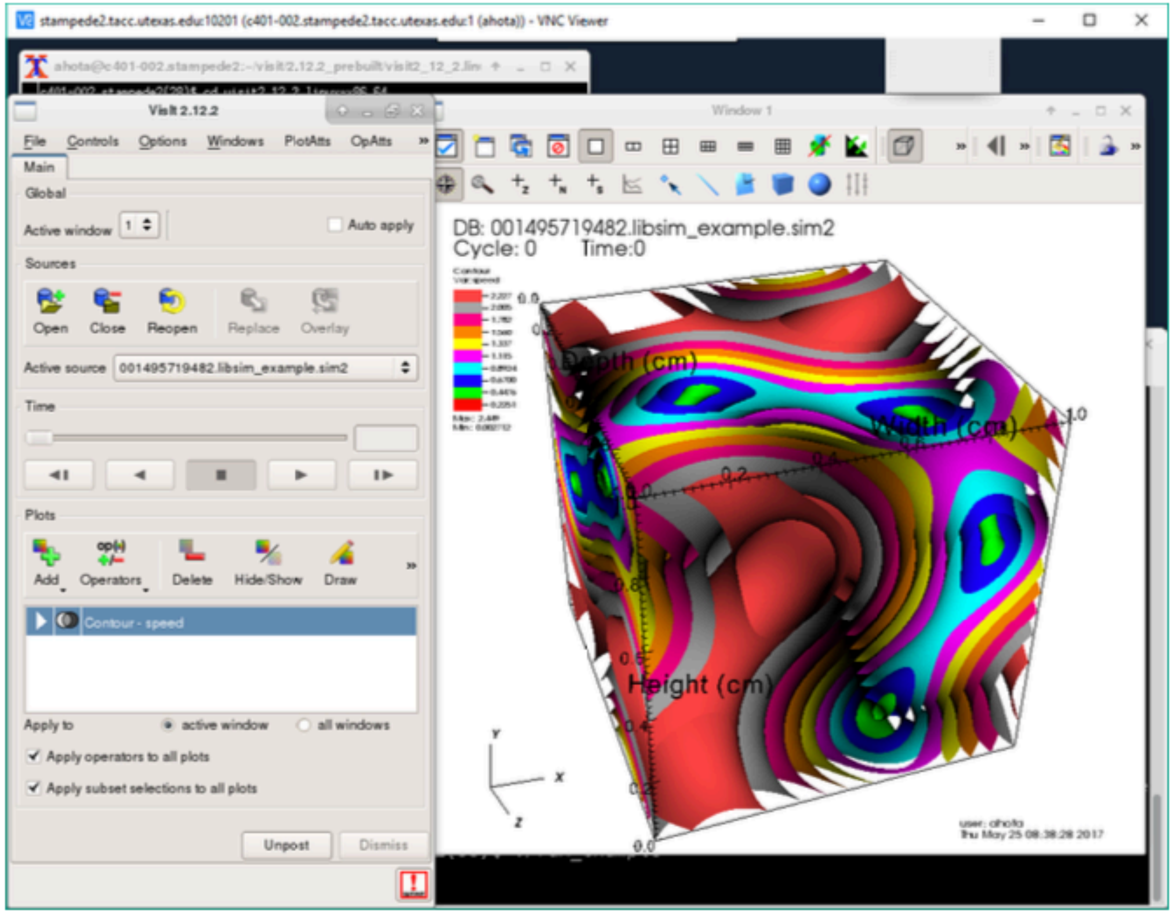
- VisIt 2.10
- OSPRay 1.1.0
- Build guide updated



VisIt + LibSim + OpenSWR

Alok Hota (Tennessee), Jian Huang (Tennessee), Hank Childs (Oregon)

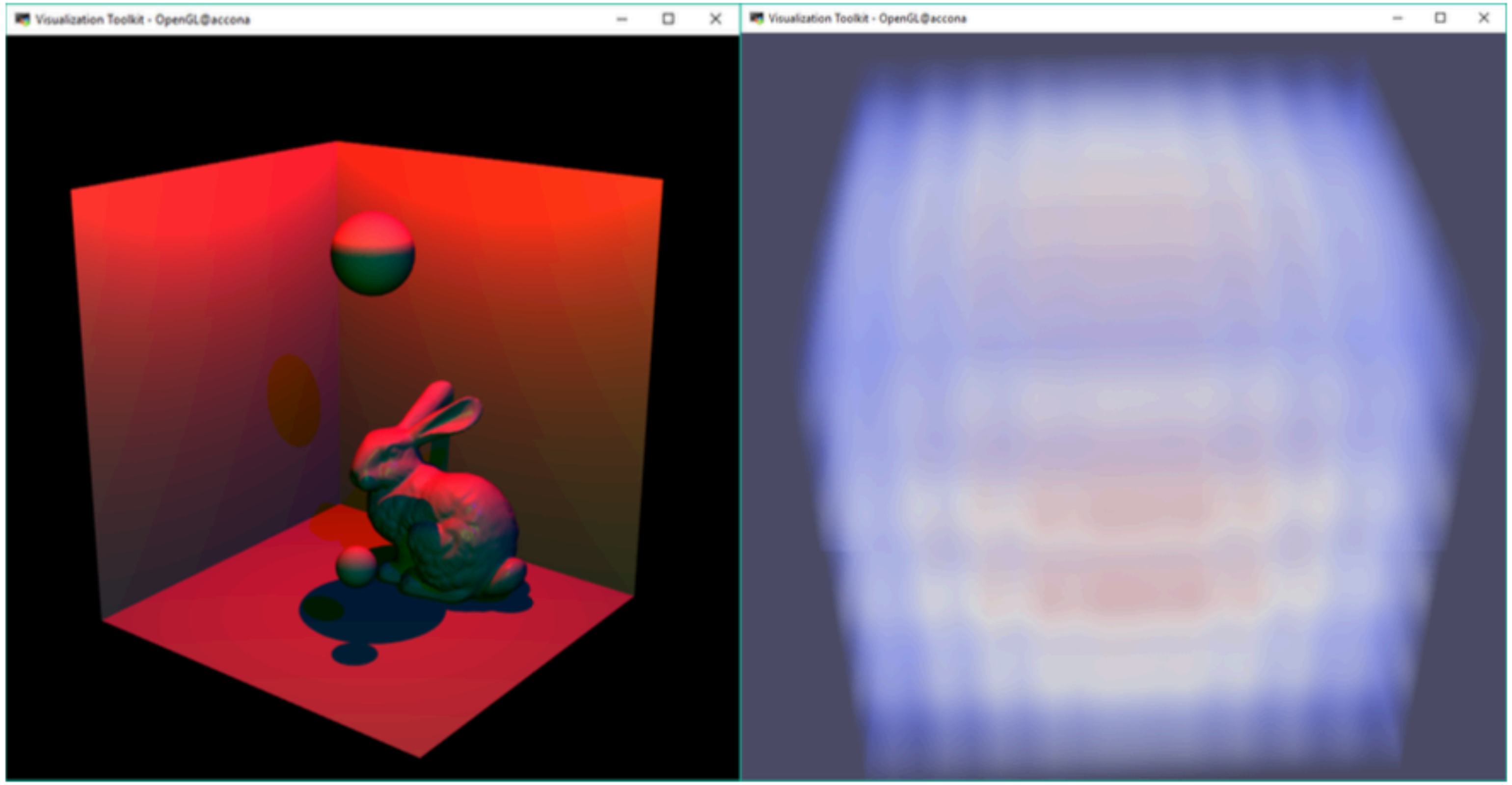
- VisIt 2.12
- SWR 17
- Arnold-Beltrami-Childress analytic vector field
- Thanks to Brad Whitlock



VisIt + LibSim + VTK Upgrade

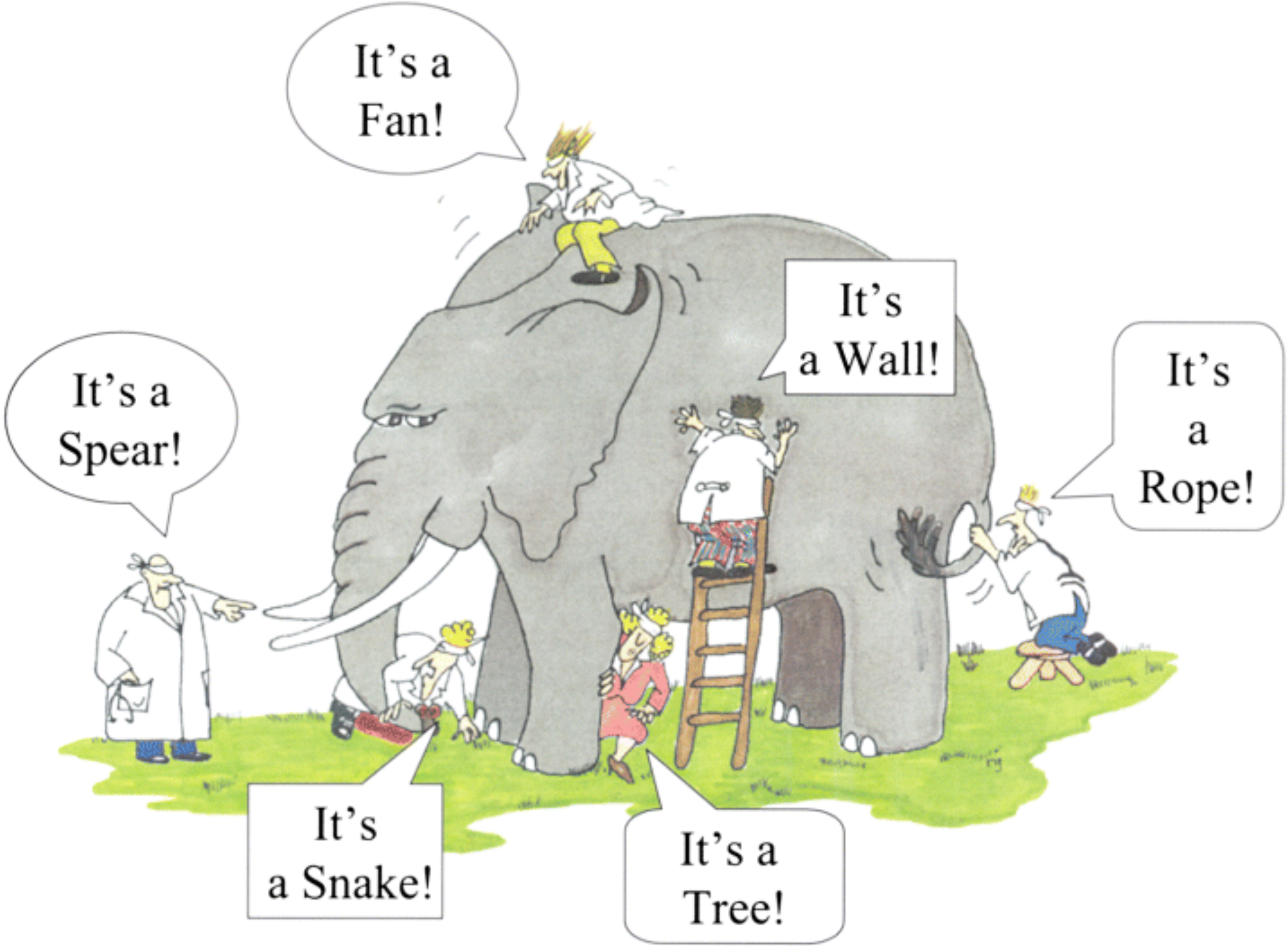
Alok Hota (Tennessee), Jian Huang (Tennessee), Hank Childs (Oregon)

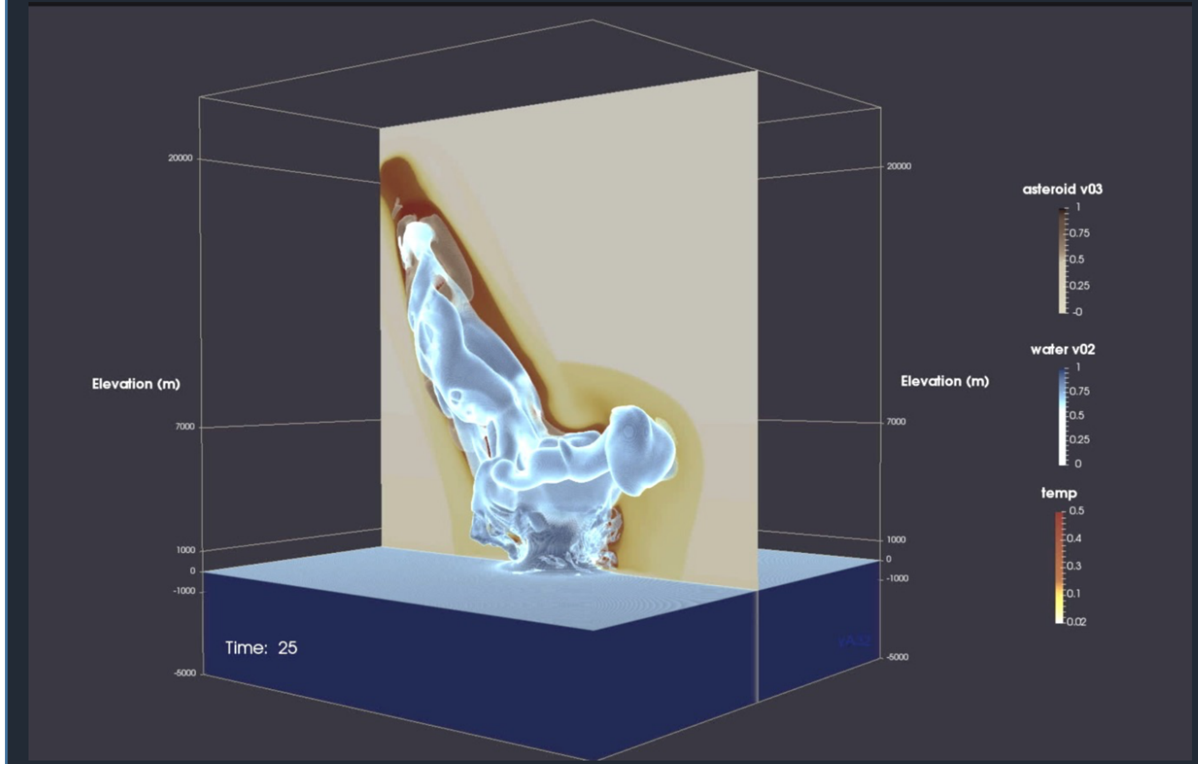
- For VisIt migration to VTK 7
- Thanks to Dave DeMarle

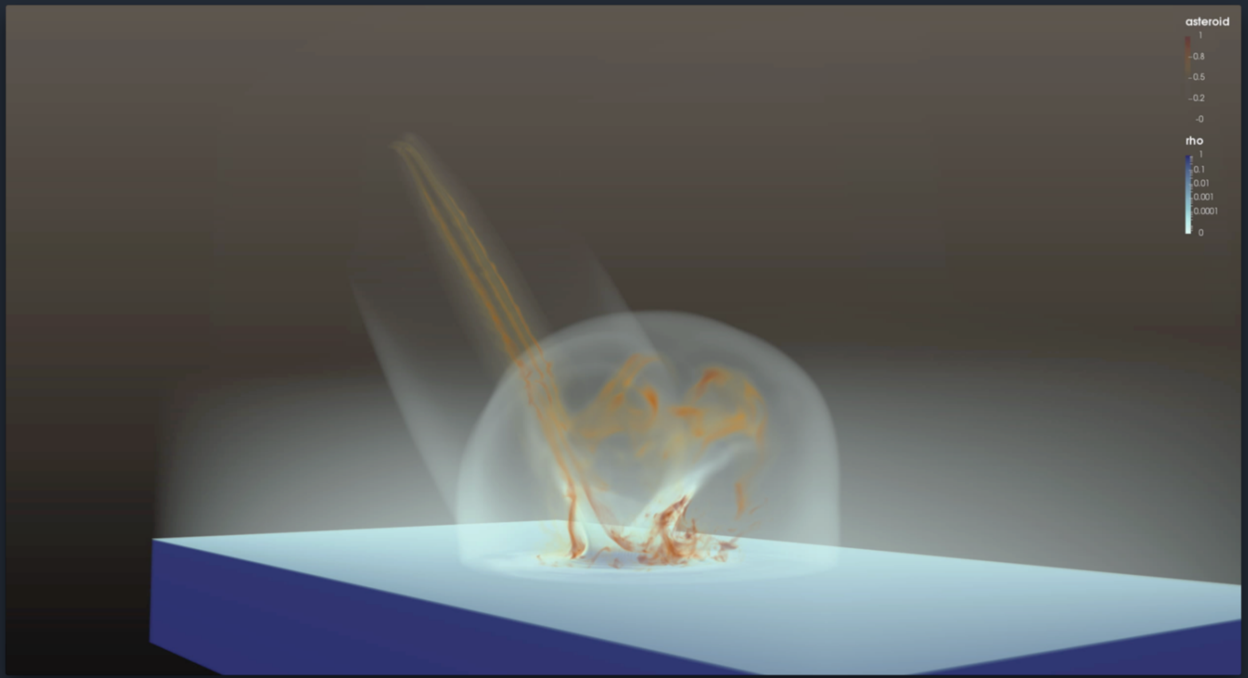


http://web.eecs.utk.edu/~ahota/visitospray/

Discussion







Thank you!

pnav@tacc.utexas.edu

TACC