Ascent: Flyweight In Situ Visualization and Analysis for HPC Simulations

OLCF User Conference Call

Wednesday September 27th, 2023

Cyrus Harrison (LLNL),
Nicole Marsaglia (LLNL)

LLNL-PRES-852351
This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

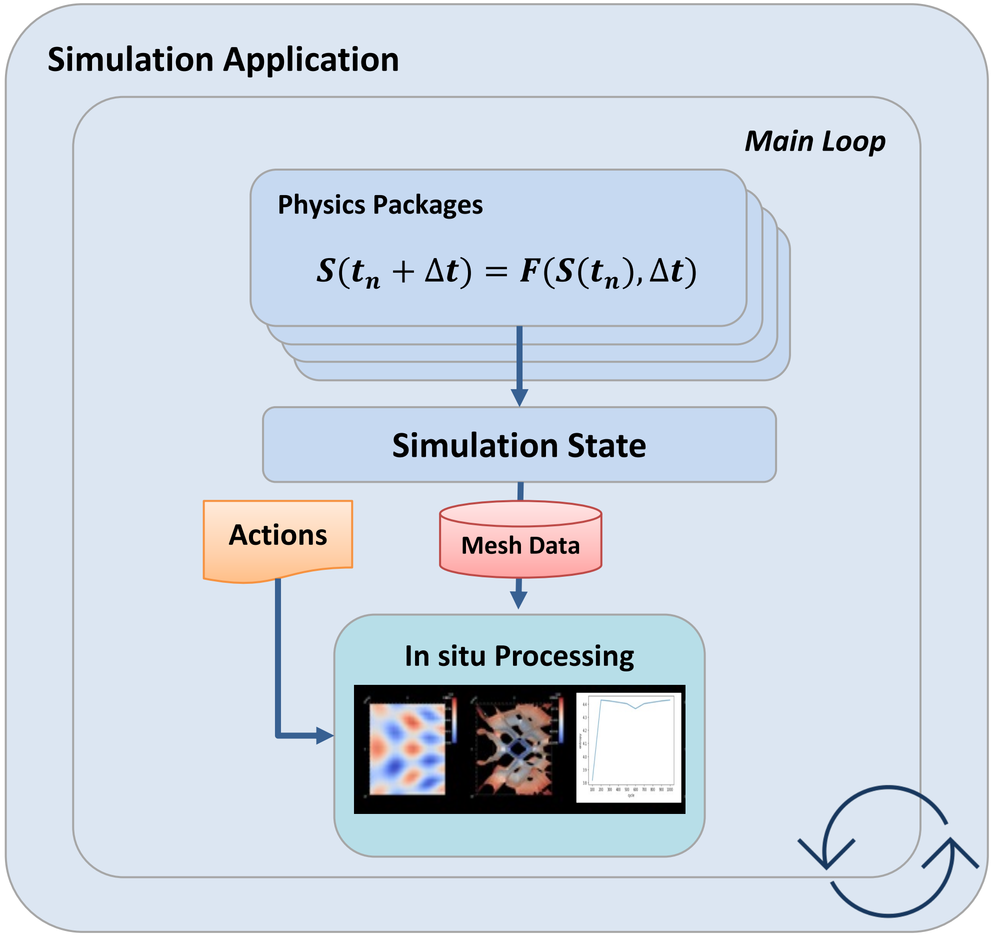
Acknowledgements

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344.
Lawrence Livermore National Security, LLC

This research was supported by the Exascale Computing Project (17-SC-20-SC), a joint project of the U.S. Department of Energy's Office of Science and National Nuclear Security Administration, responsible for delivering a capable exascale ecosystem, including software, applications, and hardware technology, to support the nation's exascale computing imperative.

What Is In Situ Processing?

- Defined:
  - Process data while it is generated
  - Couple visualization and analysis routines with the simulation code (avoiding file system I/O)
- Pros:
  - No or greatly reduced I/O vs post-hoc processing
  - Can access all the data
  - Computational power readily available
- Cons:
  - More difficult when lacking a priori knowledge of what to visualize/analyze
  - Increasing complexity
  - Constraints (memory, network)



(Slide Acknowledgement: Hank Childs)

Important links and contact info:

Ascent Resources:

- Github: https://github.com/alpine-dav/ascent
- Docs: http://ascent-dav.org/
- Tutorial Landing Page: https://www.ascent-dav.org/tutorial/

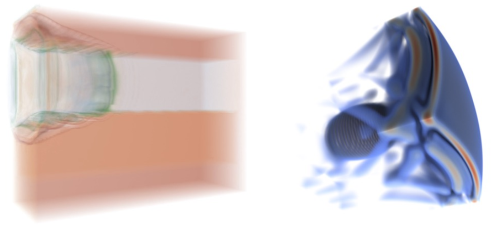
Contact Info:

Cyrus Harrison: cyrush@llnl.gov
Nicole Marsaglia: marsaglia1@llnl.gov

Ascent is an easy-to-use flyweight in situ visualization and analysis library for HPC simulations

- **Easy to use in-memory visualization and analysis**
  - Use cases: ***Making Pictures, Transforming Data,*** and ***Capturing Data***
  - Young effort, yet already supports most common visualization operations
  - Provides a simple infrastructure to integrate custom analysis
  - Provides C++, C, Python, and Fortran APIs
- **Uses a flyweight design targeted at next-generation HPC platforms**
  - Efficient distributed-memory (MPI) and many-core (CUDA, HIP, OpenMP) execution
    - Demonstrated scaling: In situ filtering and ray tracing across ***16,384 GPUs*** on LLNL's Sierra Cluster
  - Has lower memory requirements than current tools
  - Requires less dependencies than current tools (ex: no OpenGL)
    - Builds with Spack https://spack.io/

Ascent



Visualizations created using Ascent

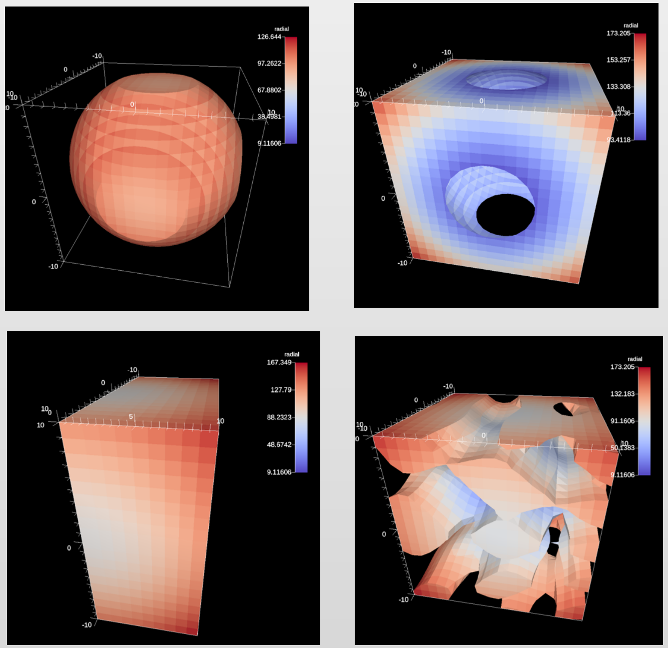
python™ HDF CINEMA

Extracts supported by Ascent

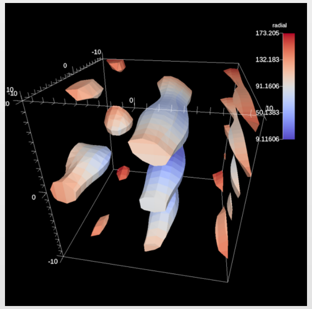
http://ascent-dav.org
https://github.com/Alpine-DAV/ascent

Website and GitHub Repo

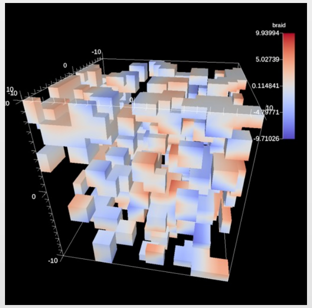
Ascent supports common visualization use cases



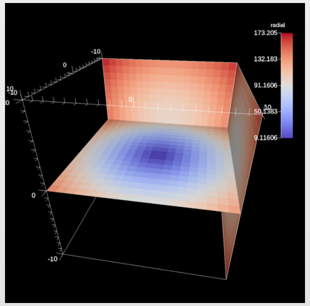
Clips



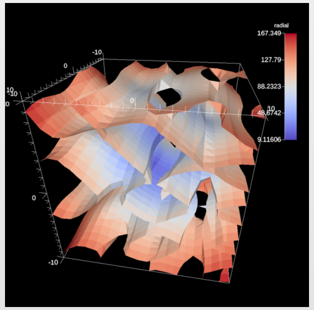
Iso-Volume



Threshold



Slice

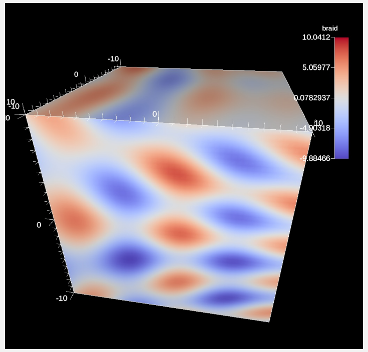


Contour

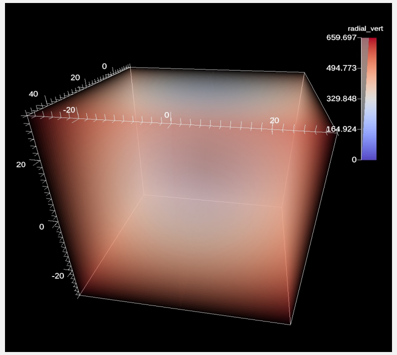
[powered by]

VTK-m RAJA mfem Devil Ray

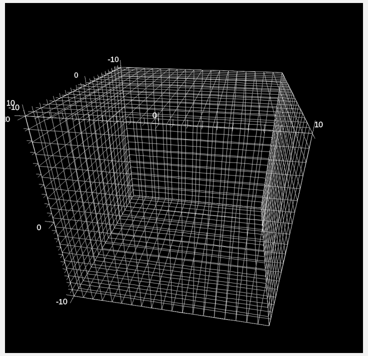
Rendering



Pseudocolor



Volume

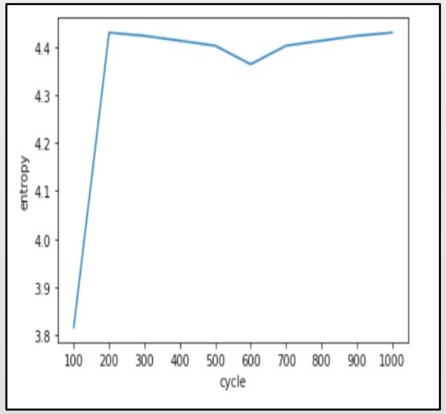


Mesh

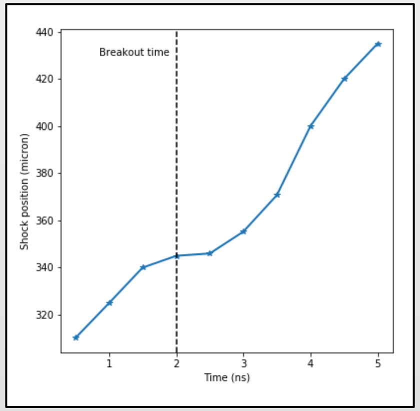
Ascent supports common analysis use cases

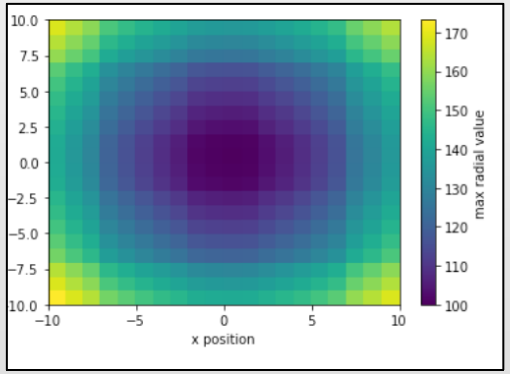
```
expression: |
  du = gradient(field('velocity','u'))
  dv = gradient(field('velocity','v'))
  dw = gradient(field('velocity','w'))
  w_x = dw.y - dv.z
  w_y = dw.z - dv.x
  w_z = dw.x - dv.y
  vector(w_x,w_y,w_z)
name: vorticity
```

Derived Fields



Time Histories



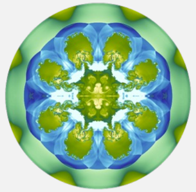


Lineouts and Spatial Binning

```
condition:
  entropy - history(entropy,
          relative_index = 1) > 0.5
```

Triggers

Extracts



Scalar Images

HDF5 Files

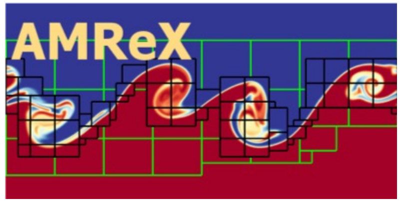
Cinema Databases

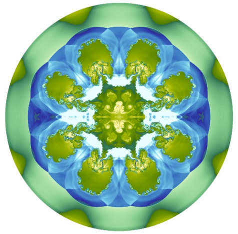
python powered

jupyter

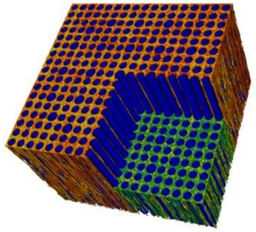
We are working to integrate and deploy Ascent with HPC simulation codes (ECP and beyond)

CEED
EXASCALE DISCRETIZATIONS

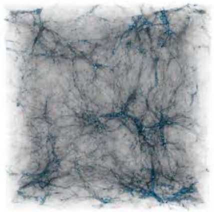




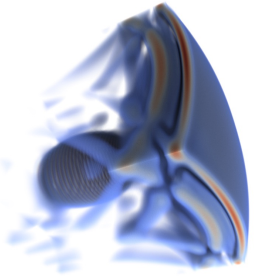
MARBL



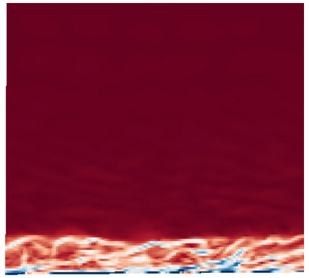
NekRS



Nyx



WarpX

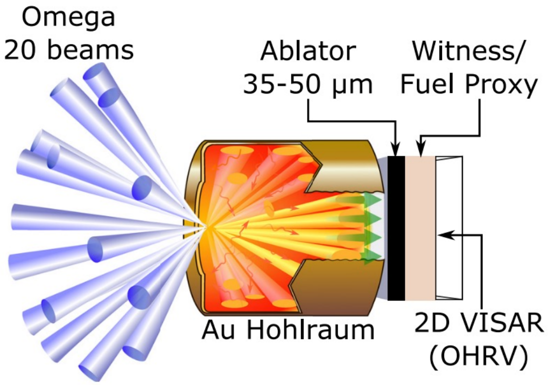


AMRWind

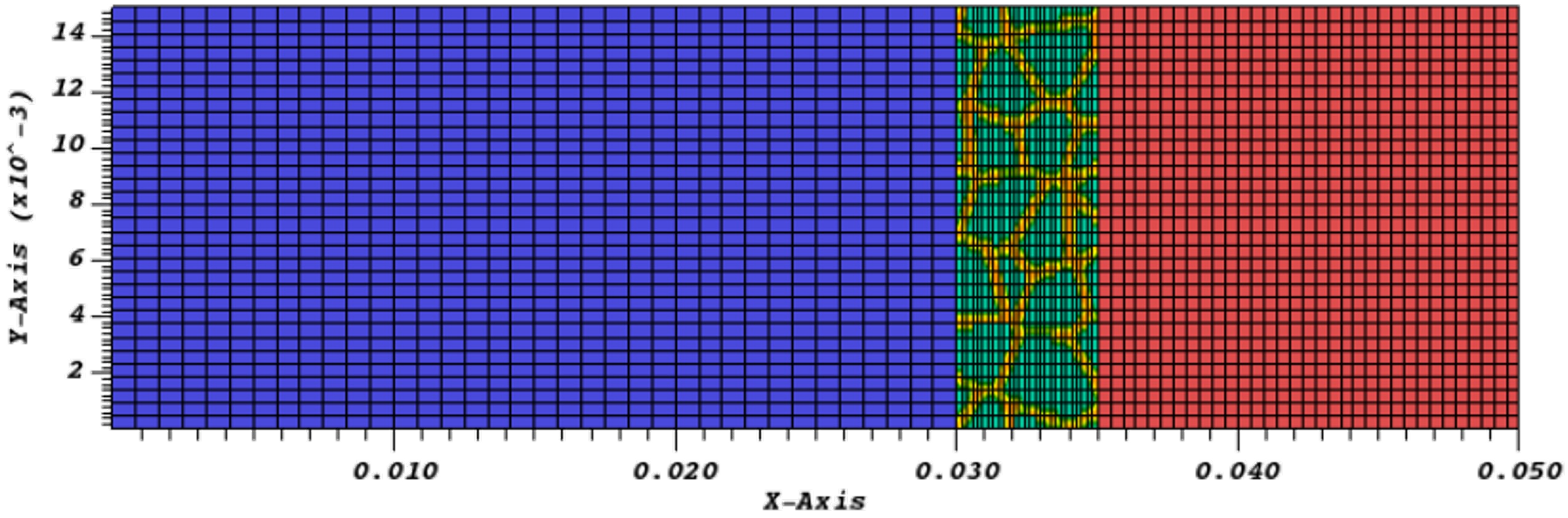
Pele

Ascent

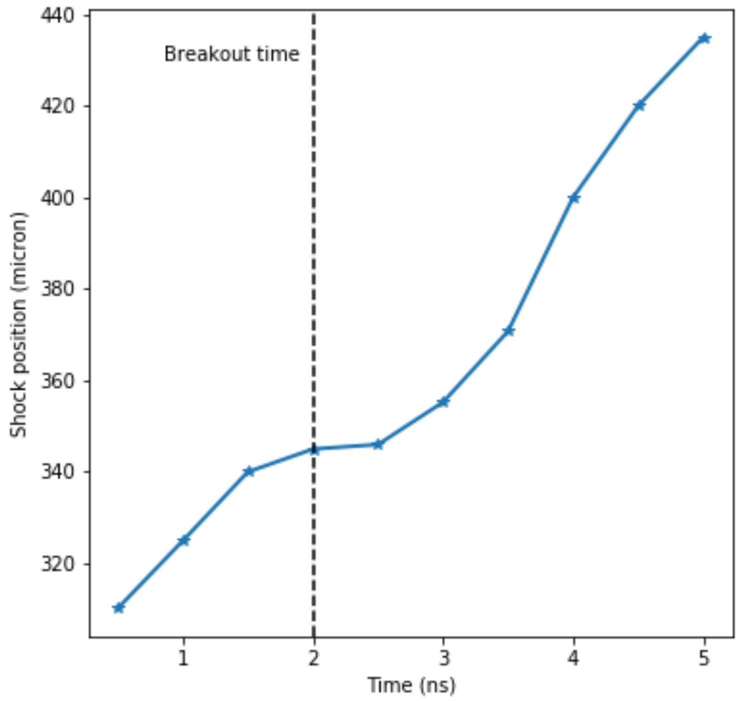
Science Enabling Results: Shock Front Tracking (VISAR)



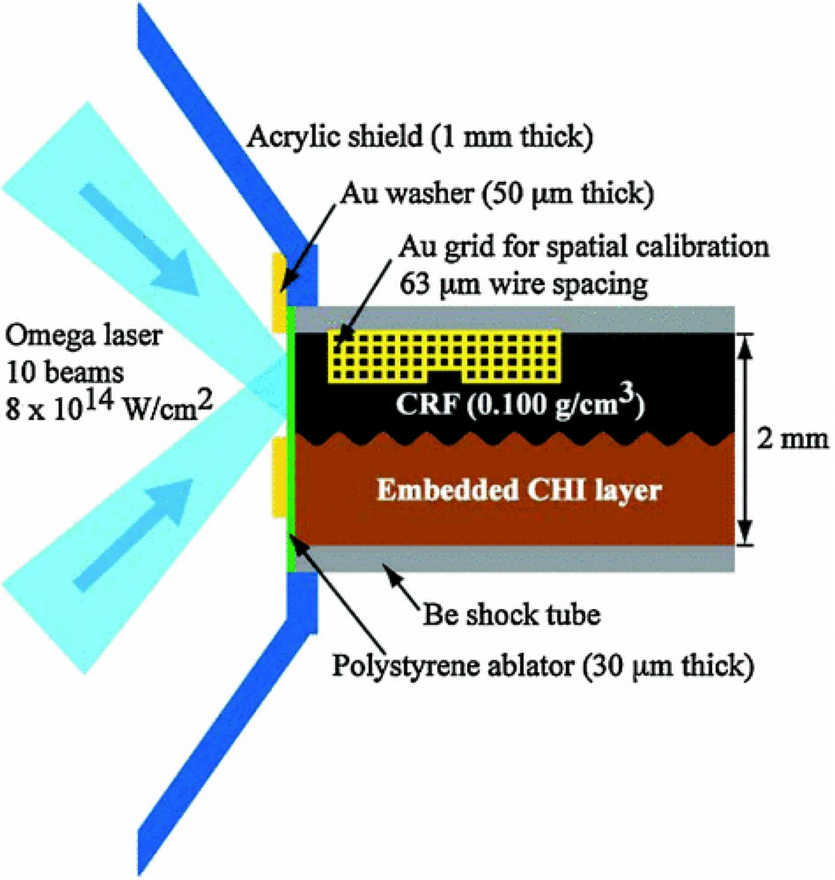
Velocity interferometer system for any reflector (VISAR)



Shock position tracked in Ascent

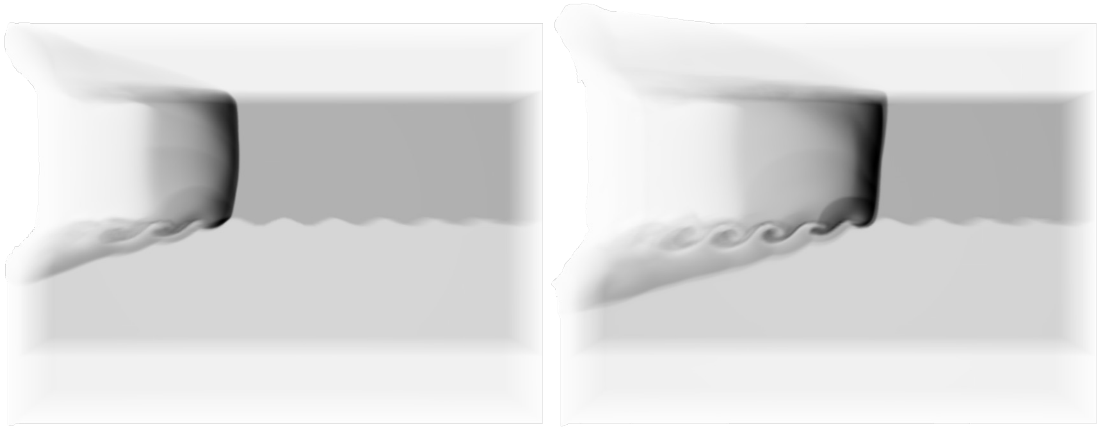


Science Enabling Results: Simulation Validation

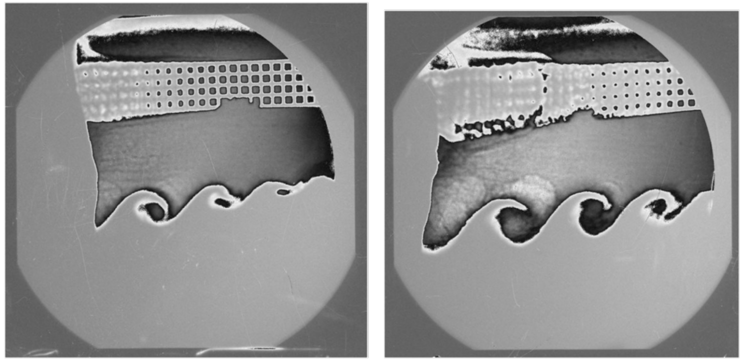


Radiographs

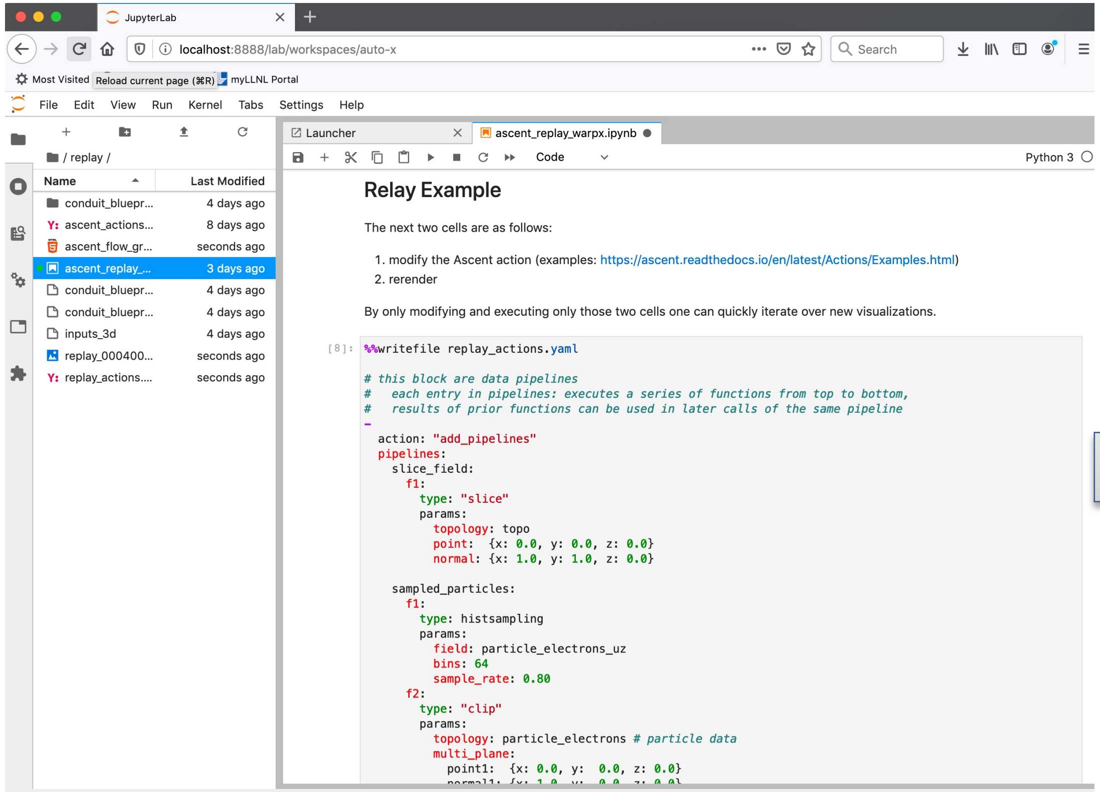
Simulated



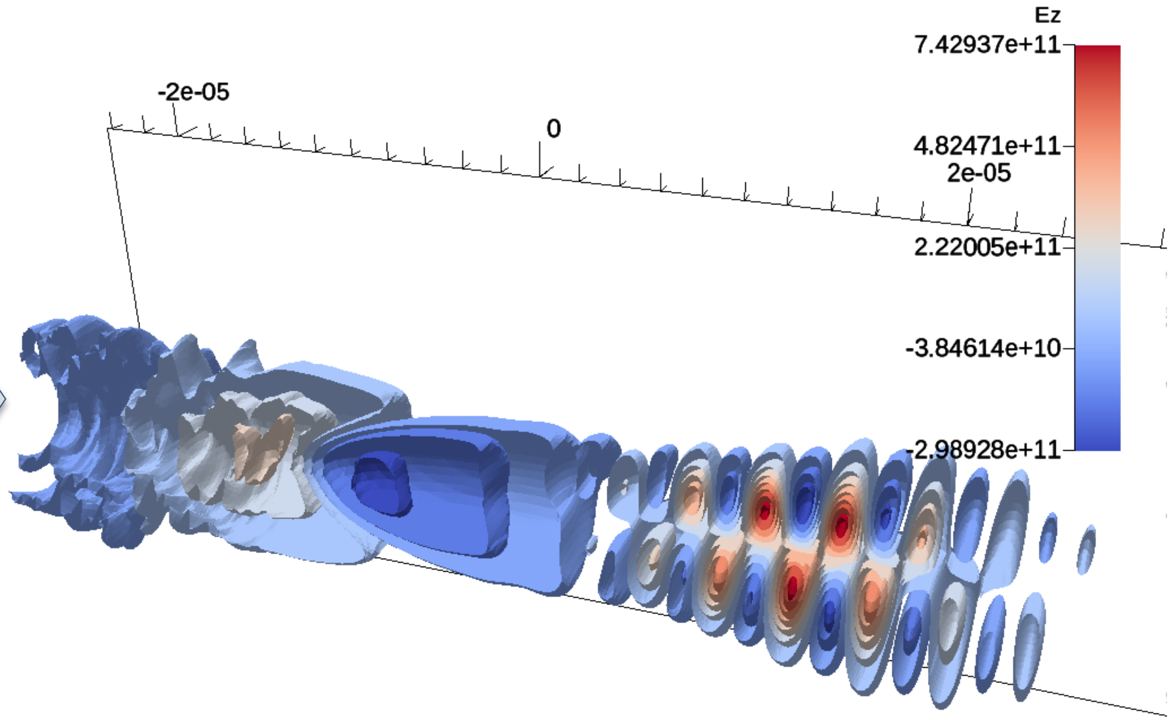
Experimental



Science Enabling Results: WarpX Workflow Tools (Jupyter Lab)



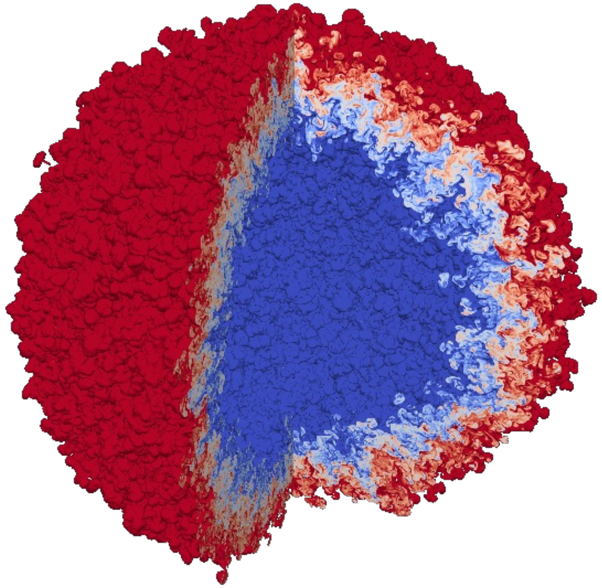
Jupyter Lab Interface



Resulting Image

Science Enabling Results: Rendering At Scale (2018)

- The ***97.8 billion*** element simulation ran across ***16,384 GPUs*** on ***4,096 Nodes***
- The simulation application used **CUDA** via **RAJA** to run on the GPUs
- Time-varying evolution of the mixing was visualized in-situ using **Ascent**, also leveraging 16,384 GPUs
- Ascent leveraged **VTK-m** to run visualization algorithms on the GPUs



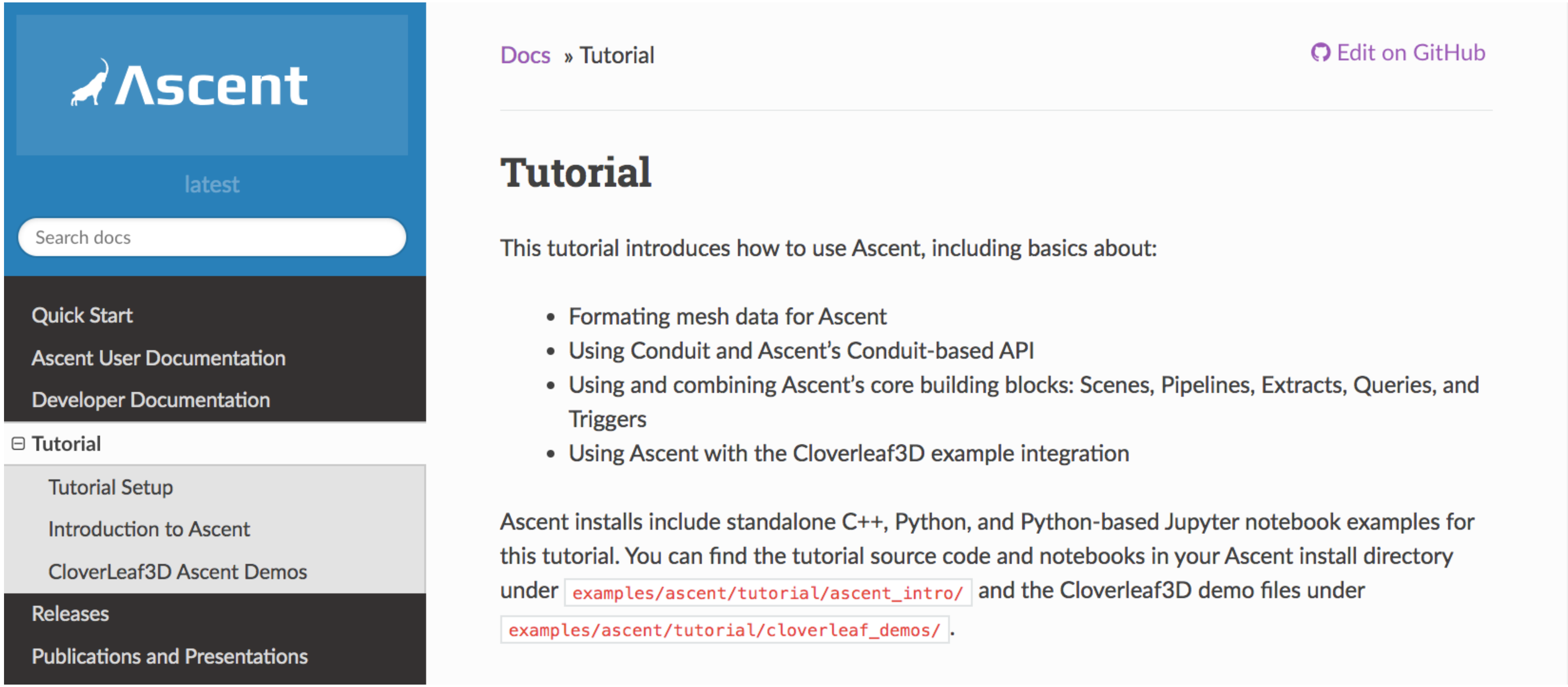
Visualization of an idealized Inertial Confinement Fusion (ICF) simulation of Rayleigh-Taylor instability with two fluids mixing in a spherical geometry.

Today we will teach you about Ascent's API and capabilities

You will learn:

- How to use Conduit, the foundation of Ascent's API
- How to get your simulation data into Ascent
- How to tell Ascent what pictures to render and what analysis to execute

Ascent tutorial examples are outlined in our documentation and included ready to run in Ascent installs



http://ascent-dav.org

Ascent's interface provides five composable building blocks

Scenes
(Render Pictures)

Pipelines
(Transform Data)

Extracts
(Capture Data)

Queries
(Ask Questions)

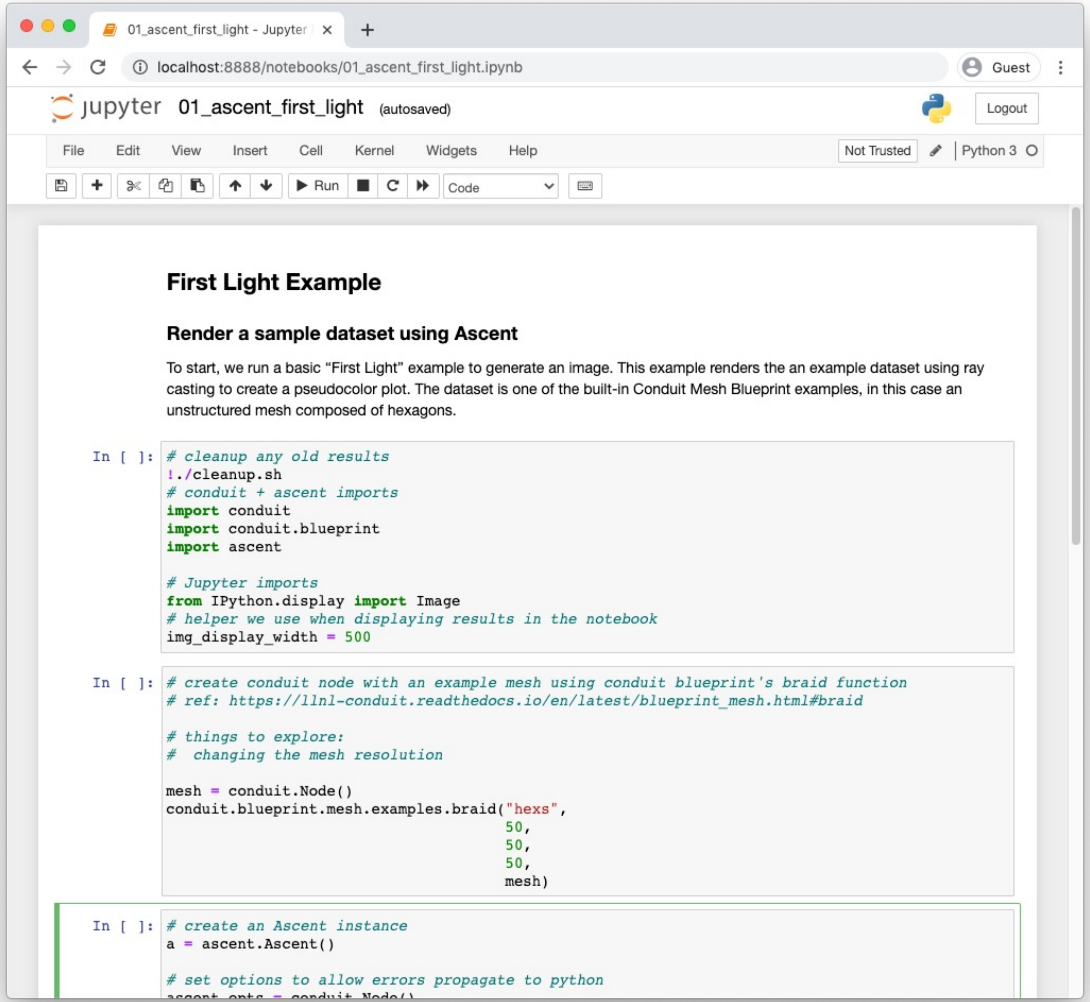
Triggers
(Adapt Actions)

The tutorial provides examples for all of these.

For the reminder of the tutorial, we will run the Ascent Tutorial examples using Jupyter Notebooks

NOTE:

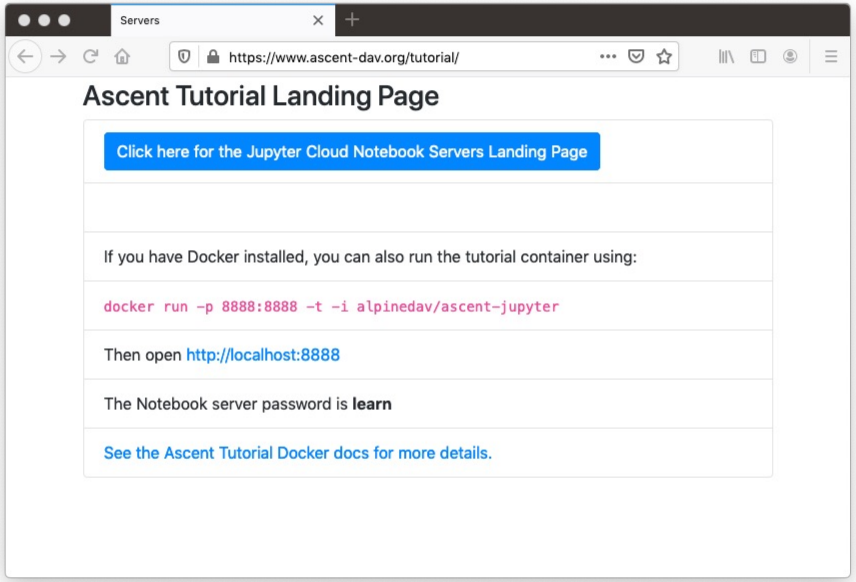
- VPNs or firewalls may block access to general AWS IP addresses and ports
- You may need to disconnect from VPN or request a firewall exemption
- LLNL attendees, you can use the EOR process: https://cspservices.llnl.gov/eor/



You can run our tutorial examples using cloud hosted Jupyter Lab servers

Start here:

https://www.ascent-dav.org/tutorial/



Thanks!

Ascent Resources:

- Github: https://github.com/alpine-dav/ascent
- Docs: http://ascent-dav.org/
- Tutorial Landing Page: https://www.ascent-dav.org/tutorial/

Contact Info:

Cyrus Harrison: cyrush@llnl.gov
Nicole Marsaglia: marsaglia1@llnl.gov

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344.
Lawrence Livermore National Security, LLC

This research was supported by the Exascale Computing Project (17-SC-20-SC), a joint project of the U.S. Department of Energy's Office of Science and National Nuclear Security Administration, responsible for delivering a capable exascale ecosystem, including software, applications, and hardware technology, to support the nation's exascale computing imperative.

Lawrence Livermore National Laboratory