DOI:10.5281/zenodo.10120349

State of In Situ Visualization in Simulations: We are fast. But are we inspiring?

EXASCALE COMPUTING PROJECT

In Situ Infrastructures for Enabling Extreme-scale Analysis and Visualization (ISAV23)

In conjunction with:
The International Conference for High Performance Computing, Networking, Storage, and Analysis (SC23)

Denver (CO), USA
November 13th, 2023

Axel Huebl
Arianna Formenti, Marco Garten, Jean-Luc Vay

Lawrence Berkeley National Laboratory

On behalf of the WarpX team (PI: Jean-Luc Vay)
LBNL, LLNL, SLAC, CEA, DESY, TAE, CERN

Special thanks to ECP Alpine - Ascent
Cyrus Harrison, Matt Larsen, Nicole Marsaglia et al.

NNSA National Nuclear Security Administration
SciDAC Scientific Discovery through Advanced Computing
U.S. DEPARTMENT OF ENERGY | Office of Science

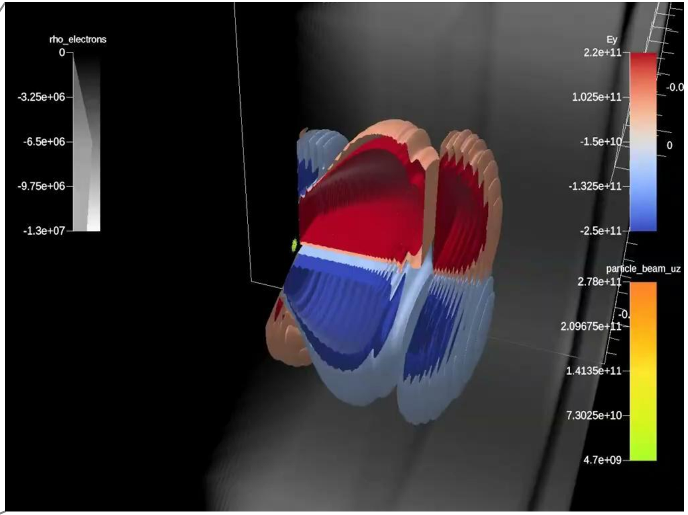
State of In Situ Visualization in Simulations

- **We are fast.** *Scalable Simulations 🤝 Visualization*
  - Our Domain Science
  - Scalable In Situ Analysis & Visualization

- **But are we inspiring?** *Exciting, unsolved challenges*
  - quality: expectations
  - workflows: inputs, animations
  - asynchronous algorithms: stitched, spatially-sliced data

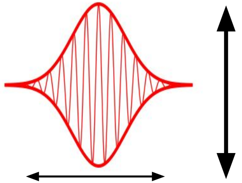
Kinetic Modeling Ecosystem

- Laser-Plasma, Accelerators & Beams
- 4+ Multi-GPU Codes, Libraries, Standards
- **WarpX**: 3D Time-Integrated PIC Code



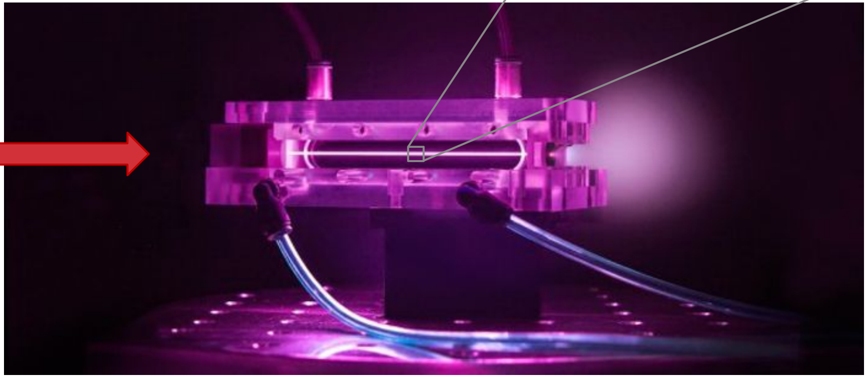
Staging of Laser-Wake-field Accelerators for Next-Gen Colliders

10^{14}W to 10^{16}W



τ=30fs

high power laser pulse

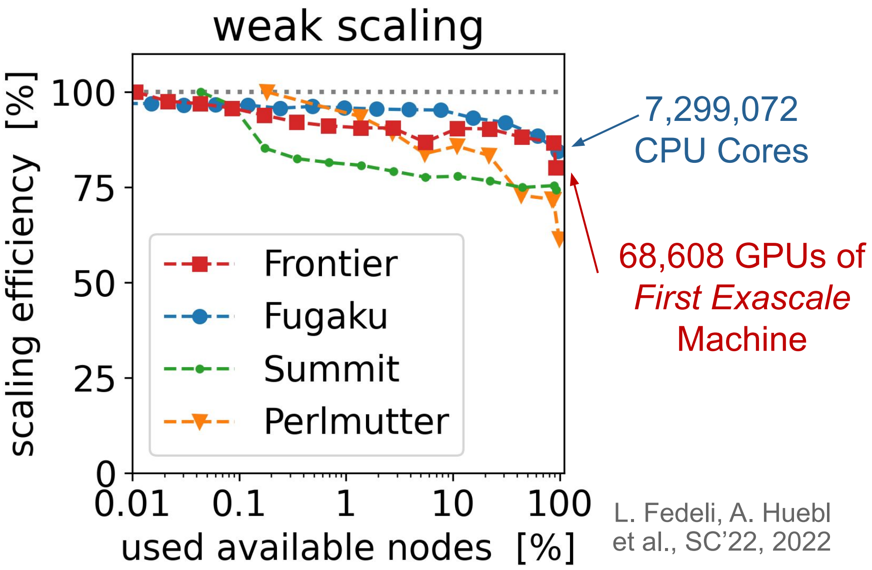


WarpX on 552 Frontier Nodes (4,416 GPUs/GCDs): transv. electric field in an LPA - *Ascent & VTK-m*
N Marsaglia, M Larsen, C Harrison, A Huebl, J-L Vay DOI:10.5281/zenodo.8226853

Plasma	←**Conventional**
100 GV / m	←20 MV/m

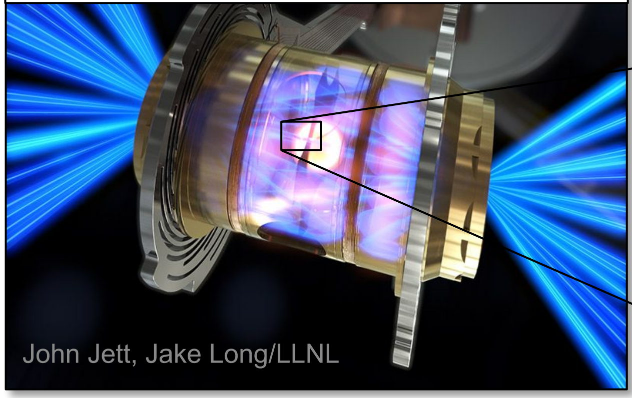
Kinetic Modeling Ecosystem

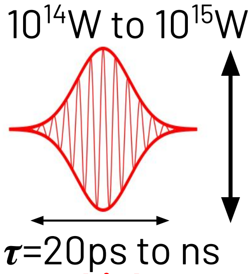
- Laser-Plasma, Accelerators & Beams
- **WarpX:** 2022 ACM Gordon Bell Prize



L. Fedeli, A. Huebl et al., SC'22, 2022

Kinetic effects in Inertial Confinement Fusion

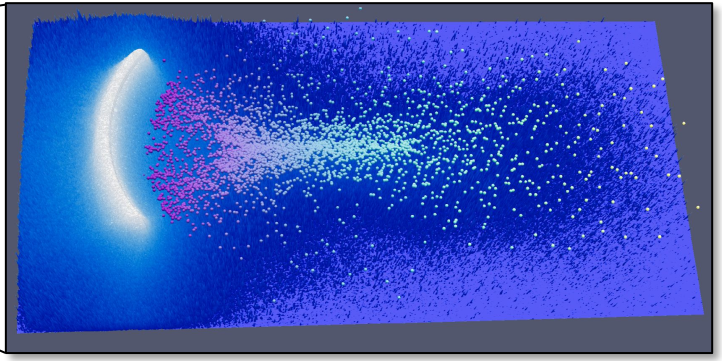




many high power laser pulses

WarpX on Frontier - *ParaView*

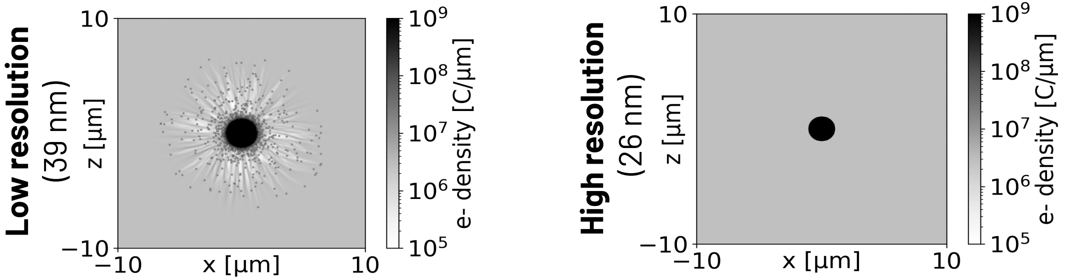
A Huebl, M Garten, J-L Vay, J Ludwig, S Wilks, A Kemp



Through Visualization, we want to Stimulate & Develop Insight

Selected *visual* questions we want to address *in situ*

- **Physics:** Which effects of scale & dimensionality are overlooked in lower fidelity?
- **Dynamics:** Is a (costly) simulation evolving as anticipated?
- **Analytics:** What is the response on a (virtual) detector?
- **Correctness:** Are numerical options and resolution sufficient & stable?



L. Fedeli, A. Huebl et al., SC'22, 2022

  - Are any hardware or software issues/bugs appearing at scale?

EXASCALE COMPUTING PROJECT

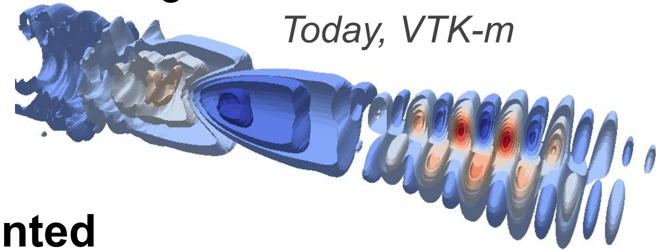
Opportunity 1: Stimulate Insight, Inspire Ideas

Quality of In Situ Generated Vis

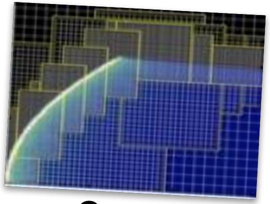
2005
OpenDX

- **Expectations** from traditional vis
  - movies, games, advertising
  - stimulate minds, inspire awe

Today, VTK-m



- **Scalable Methods Wanted**
  - casting soft+hard shadows, tracing reflections, semi-transparent iso-contours, smoothing, volume-rendering >1 overlapping source
    - sorting collisions with objects, etc.
    - notoriously non-local and are thus challenging for multi-GPU



What we willing to trade for this?
Add/exploit artificial locality from refinement, reduction, occlusion/defocus/fog, …?

Opportunity 2: User-Facing Workflows

Usage could be easier

- New tool = New input
    - standardize visualization scenes[1]
    - rapid scene design
- exchange scenes: post-processing GUIs ⇆ in situ
    - ambience: load external/STL geometries

[1] Conduit, OpenUSD.org, ANARI

```yaml
-
  action: add_pipelines
  pipelines:
    contour_pipeline:
      f0:
        params:
          field: Ey
          levels: 16
        type: contour
-
  action: add_scenes
  scenes:
    ey_contour_pc:
      plots:
        p0:
          field: Ey
          pipeline: contour_pipeline
          type: pseudocolor
      renders:
        r1:
          camera:
            azimuth: 90.0
          image_prefix: "ey_contour_pc_"
    ey_volrend:
      plots:
```

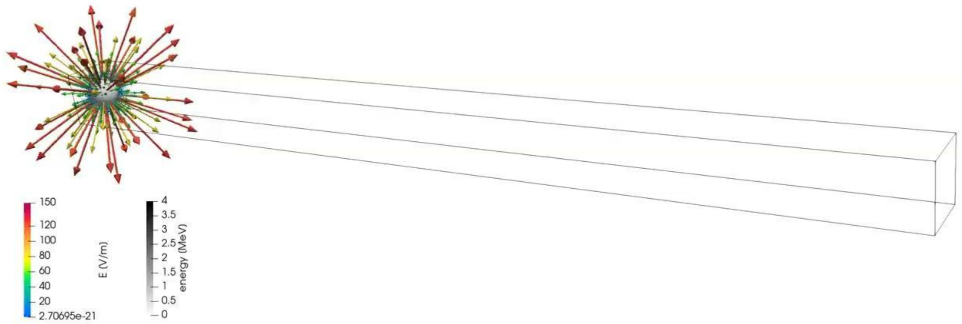
EXASCALE COMPUTING PROJECT

Opportunity 2: User-Facing Workflows

Usage could be easier

[1] Conduit, OpenUSD.org, ANARI

- New tool = New input
  - standardize visualization scenes[1]
  - rapid scene design
- exchange scenes: post-processing GUIs ⇆ in situ



WarpX - *ParaView*
A Formenti

- **Animations**
  - flicker: iso-contours, glyphs, streamlines
  - reason: roughness of simulation data and steps selected
  - challenge: smooth transitions/animations as in web/CSS?

EXASCALE COMPUTING PROJECT

Opportunity 3: Visualize Partial Data - Stitch it Over Time

BLAST
BEAM PLASMA & ACCELERATOR SIMULATION TOOLKIT

Often, we cannot yet in situ visualize the *right* data.

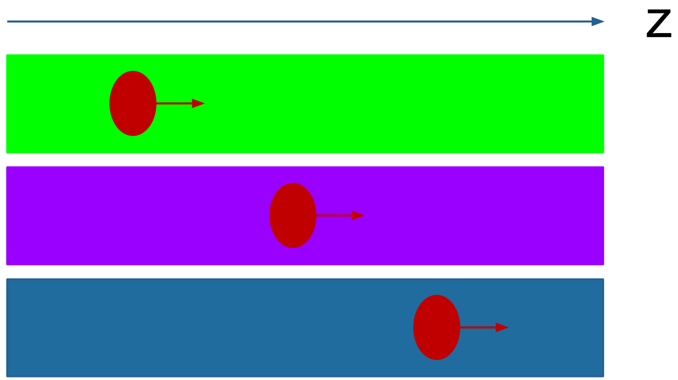
- Traditional, time-based iteration
  - every cell & particle are modeled at the same time t
  - $t_{n+1} = t_n + \Delta t$

Traditional Domain Decomposition

$t_1 = t_0 + \Delta t$

$t_2 = t_1 + \Delta t$

$t_3 = t_2 + \Delta t$



sort-last parallel render

sort-last parallel render

sort-last parallel render

K Moreland and B Wylie, IEEE PVG (2001)   K Moreland et al., SC11 (2011)
https://www.kennethmoreland.com/scalable-rendering

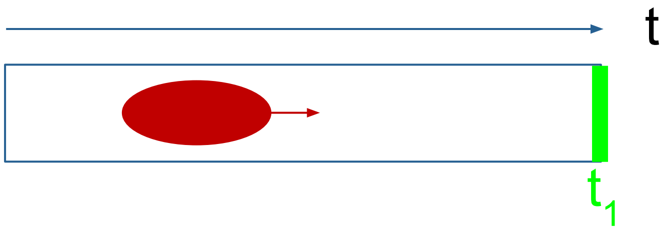
EXASCALE COMPUTING PROJECT

Opportunity 3: Visualize Partial Data - Stitch it Over Time

Often, we cannot yet in situ visualize the *right* data.

- Codes in BLAST: WarpX w/ boosted frame, HiPACE++, ImpactX
- domain-decomposition: space (2D) + time (1D)
- render streamed, spatially-sliced data

Optimal Ref. Frame for Compute



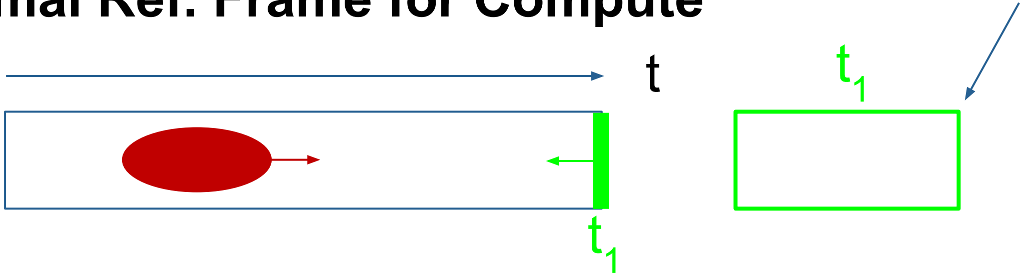
Opportunity 3: Visualize Partial Data - Stitch it Over Time

Often, we cannot yet in situ visualize the *right* data.

- Codes in BLAST: WarpX w/ boosted frame, HiPACE++, ImpactX
- domain-decomposition: space (2D) + time (1D)
- render streamed, spatially-sliced data

Optimal Ref. Frame for Compute

back-transformed data

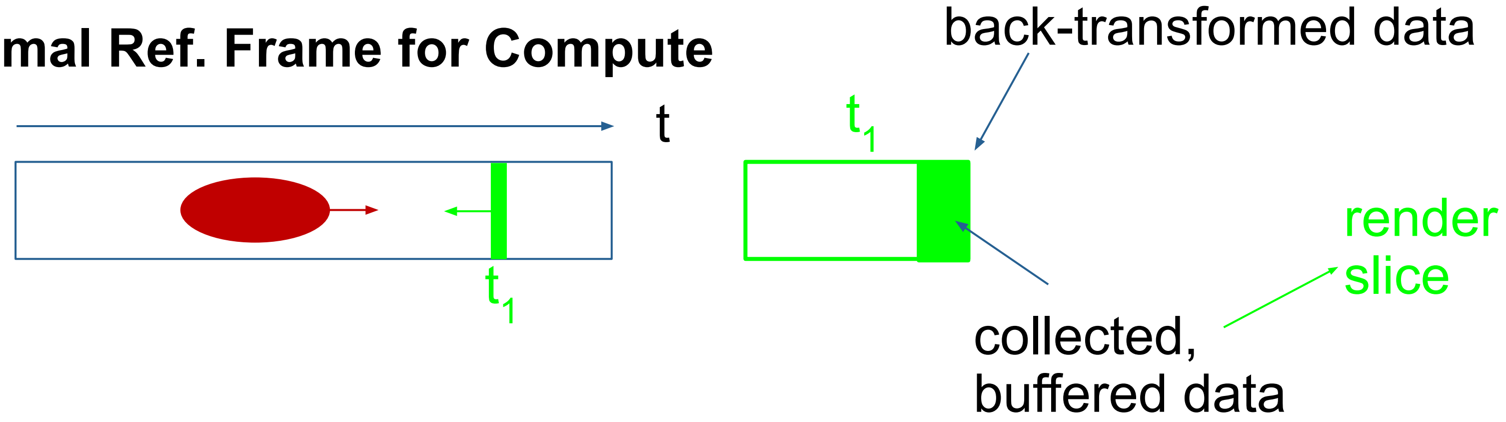


Opportunity 3: Visualize Partial Data - Stitch it Over Time

Often, we cannot yet in situ visualize the *right* data.

- Codes in BLAST: WarpX w/ boosted frame, HiPACE++, ImpactX
- domain-decomposition: space (2D) + time (1D)
- render streamed, spatially-sliced data

Optimal Ref. Frame for Compute

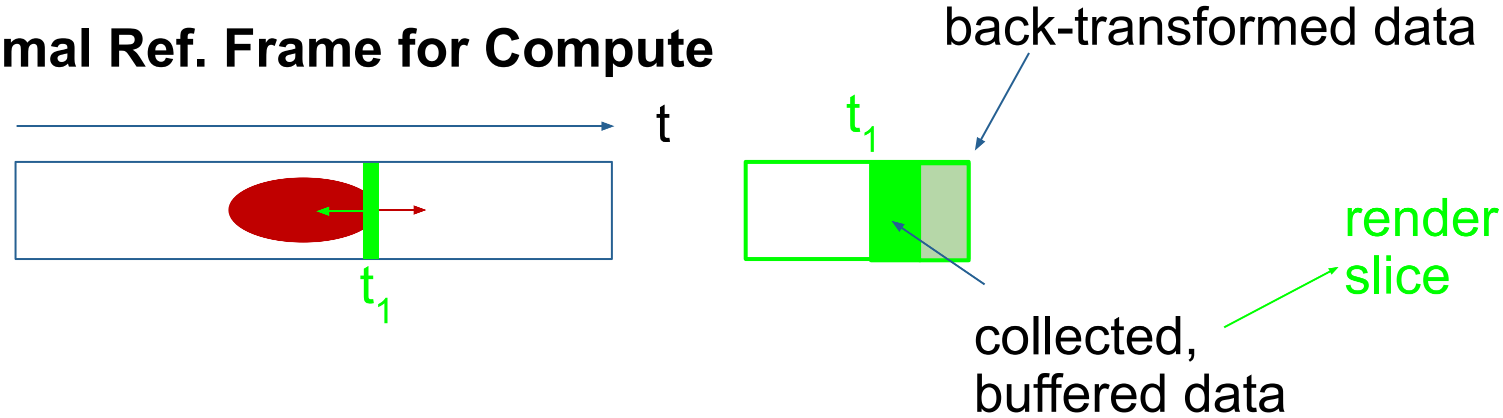


Opportunity 3: Visualize Partial Data - Stitch it Over Time

Often, we cannot yet in situ visualize the *right* data.

- Codes in BLAST: WarpX w/ boosted frame, HiPACE++, ImpactX
- domain-decomposition: space (2D) + time (1D)
- render streamed, spatially-sliced data

Optimal Ref. Frame for Compute

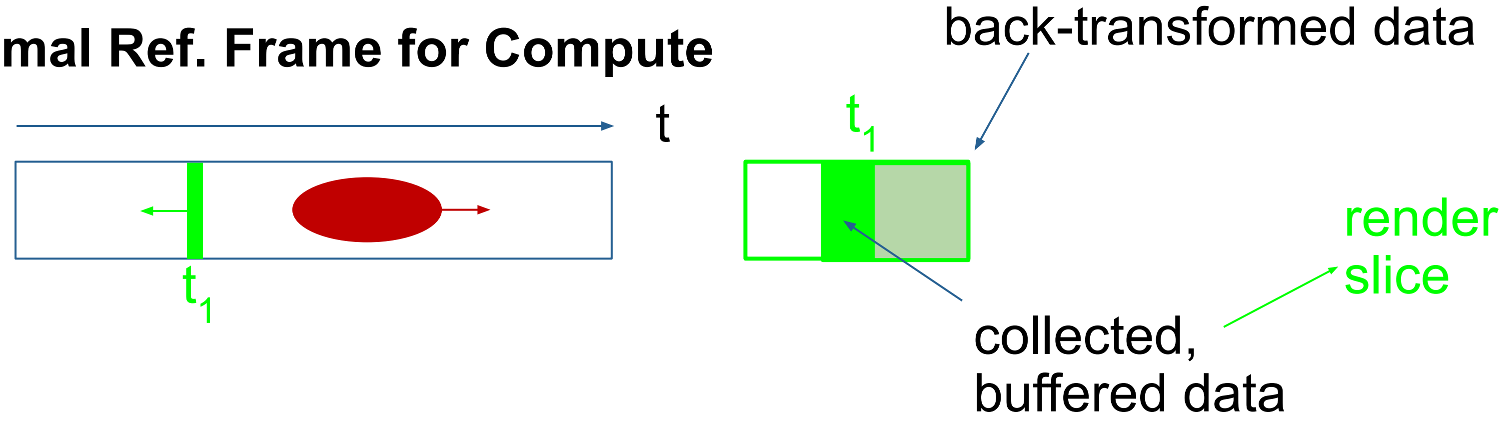


Opportunity 3: Visualize Partial Data - Stitch it Over Time

Often, we cannot yet in situ visualize the *right* data.

- Codes in BLAST: WarpX w/ boosted frame, HiPACE++, ImpactX
- domain-decomposition: space (2D) + time (1D)
- render streamed, spatially-sliced data

Optimal Ref. Frame for Compute

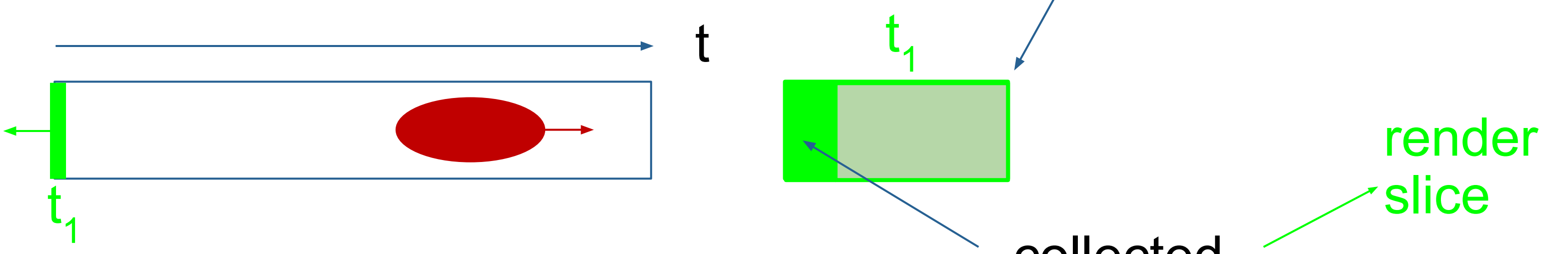


Opportunity 3: Visualize Partial Data - Stitch it Over Time

Often, we cannot yet in situ visualize the *right* data.

- Codes in BLAST: WarpX w/ boosted frame, HiPACE++, ImpactX
- domain-decomposition: space (2D) + time (1D)
- render streamed, spatially-sliced data

Optimal Ref. Frame for Compute



Now: *sort-last parallel* ***composition*** of **all rendered slices** for t_1 .

State-of-the-art: Fallback to slice-wise, full data <u>output</u> - vis. in post!

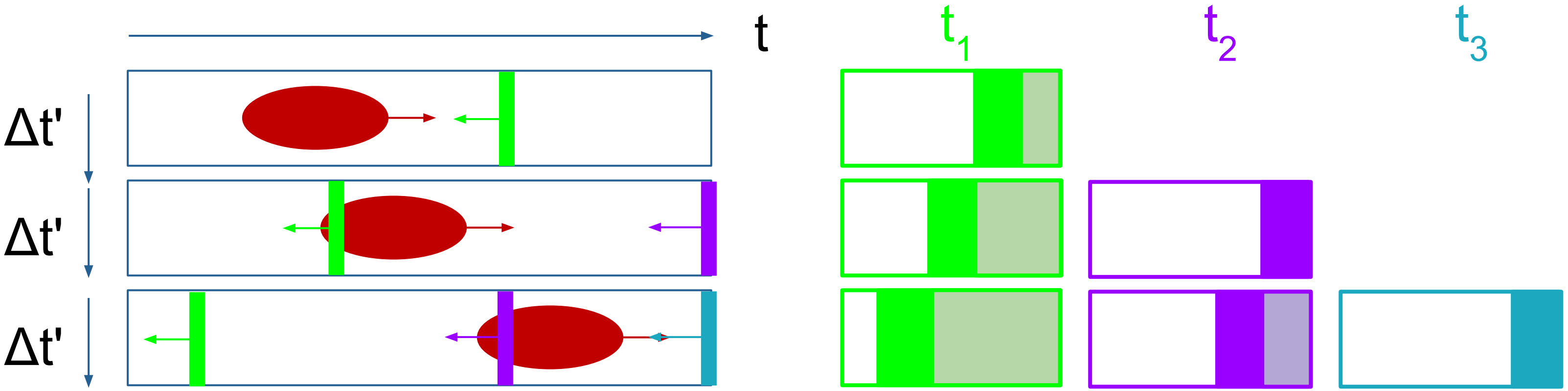
Opportunity 3: Visualize Partial Data - Stitch it Over Time

BEAM PLASMA & ACCELERATOR SIMULATION TOOLKIT

Often, we cannot yet in situ visualize the *right* data.

- Codes in BLAST: WarpX w/ boosted frame, HiPACE++, ImpactX
- domain-decomposition: space (2D) + time (1D)
- render streamed, spatially-sliced data

Optimal Ref. Frame for Compute



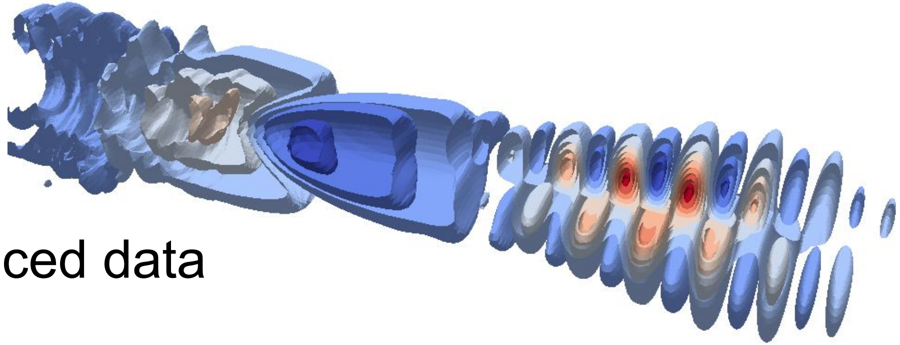
State-of-the-art: Fallback to slice-wise, full data output - vis. in post!

EXASCALE COMPUTING PROJECT

Thank you for your Attention - Let's Address those Opportunities Together

Opportunity Recap

1) **quality**: expectations
2) **workflows**: inputs, animations
3) **algorithms**: stitched, spatially-sliced data



WarpX: longitudinal electric field in a laser-plasma accelerator
rendered with Ascent & VTK-m

github.com/**ECP-WarpX**
github.com/**openPMD**
github.com/**AMReX-Codes**
github.com/**picmi-standard**

This research was supported by the **Exascale Computing Project** (17-SC-20-SC), a collaborative effort of two **U.S. Department of Energy organizations (Office of Science and the National Nuclear Security Administration)** responsible for the planning and preparation of a capable exascale ecosystem, including software, applications, hardware, advanced system engineering and early testbed platforms, in support of the nation's exascale computing imperative. This work was also performed in part by the **Laboratory Directed Research and Development Program of Lawrence Berkeley National Laboratory** under U.S. Department of Energy Contract No. DE-AC02-05CH11231, **Lawrence Livermore National Laboratory** under Contract No. DE-AC52–07NA27344 and **SLAC National Accelerator Laboratory** under Contract No. AC02–76SF00515. Supported by the **CAMPA collaboration**, a project of the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research and Office of High Energy Physics, **Scientific Discovery through Advanced Computing (SciDAC)** program. This research used resources of the **Oak Ridge Leadership Computing Facility**, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725, the **National Energy Research Scientific Computing Center (NERSC)**, a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05CH11231, and the supercomputer Fugaku provided by **RIKEN**.