Scalable Visualization, Query, and I/O for Particle Accelerator Simulations using Warp

Oliver Rübel*, Burlen Loring*, Jean-Luc Vay†‡, David Grote†§, Cameron G. R. Geddes†‡, Hari Krishnan*, Alex Romosan*, Kesheng Wu*, Prabhat*, Arie Shoshani*, and E. Wes Bethel*

*Computational Research Division, LBNL | †Accelerator & Fusion Research Division, LBNL | §Fusion Energy Sciences Program, LLNL | ‡Members of ComPASS Partnership

1) Science Problem

The available I/O bandwidth and data storage capabilities are decreasing relative to computation, limiting the amount of data that can be saved for post-process analysis. To enable particle-in-cell (PIC) simulations to: i) more effectively utilize high-performance computing resources, ii) perform analysis at high temporal resolution, and to iii) enable knowledge discovery from large-scale simulations, we are implementing a three-fold strategy as part of the WARP [4] simulation framework:

1. Couple general-purpose, state-of-the-art *in situ* visualization technology using VisIt [3] with Warp to make new advanced analysis capabilities accessible to Warp and to enable *in situ* processing of the complete data in parallel, which is not possible using the current approach based on OpenDX (see Fig 1.1).
2. Integrate *in situ* query capabilities with Warp and the visualization to enable identification of data features and data subsets of interest, reducing the amount of data that needs to be visualized and stored.
3. Integrate high-performance I/O libraries with Warp to improve parallel I/O performance.

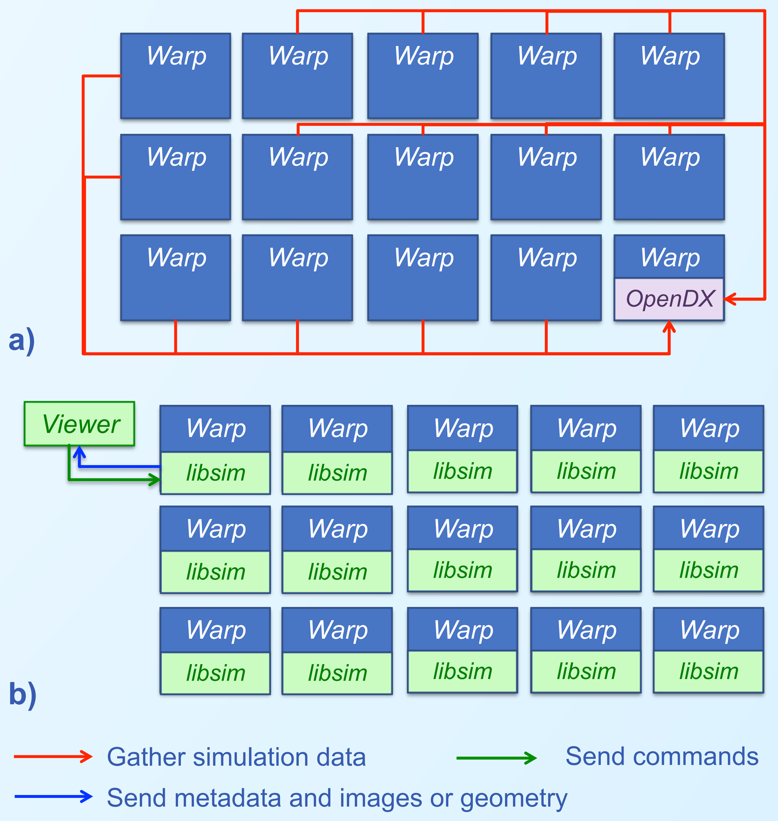


Fig. 1.1 *In situ* visualization using a) Warp+OpenDX and b) Warp+VisIt. VisIt processes the simulation data in parallel and in place, enabling large-scale visualization of the complete data while reducing communication cost.

2) Controlling the Simulation and Visualization

Monitoring Mode

The simulation runs independently while the user connects periodically to the simulation via the visualization to check results and perform *in situ* analysis.

Batch Mode

The simulation and visualization are executed in concert without external user control.

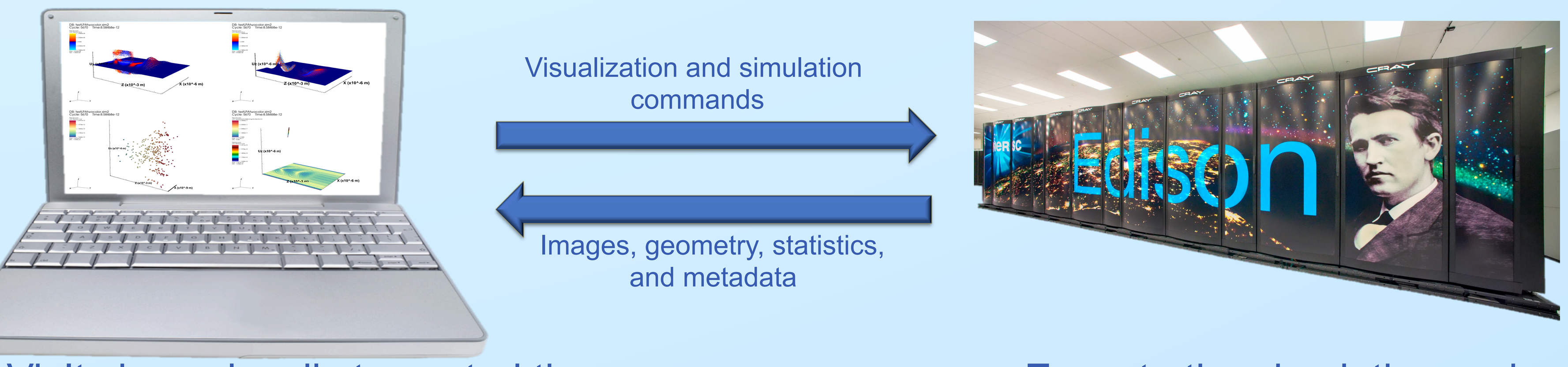
Interactive Mode

The user controls the simulation from the visualization GUI or Python shell while exploring the simulation data as it is being generated.

Shell Mode

The user controls the simulation and the visualization directly from the simulation shell.

3) Enabling *In Situ* Remote Data Analysis and Visualization



Run the VisIt viewer locally to control the visualization and simulation and to view visualization and analysis results

Execute the simulation and perform *in situ* visualization and analysis on remote HPC system

4) Multiple Species and Field Support

Warp uses a flexible multi-species PIC model and provides a broad variety of particle movers and field solvers. The type and number of particle species, fields, and associated variables often vary between simulations depending on the accelerator design modeled and solvers used. To enable users to easily interact with this dynamic collection of data and to facilitate the integration of VisIt with Warp, we have extended Warp to support introspection of the available particles, fields, and associated metadata and adapted the Warp+VisIt interface to enable dynamic variable access.

5) Memory Profiling and Optimization

A central challenge for *in situ* visualization is to minimize the impact of the visualization on the simulation. A central cost factor for *in situ* visualization is memory usage. To reduce memory usage and enable long production runs, we have:

- Added NumPy support for VisIt's libsim simulation interface, avoiding the need for additional data copies to convert Warp data arrays to Python lists originally required for integration with VisIt.
- Added support for zero copy, enabling VisIt to directly use Warp data arrays in place.
- Profiled memory usage for Warp+VisIt in detail and fixed various memory leaks (see Fig. 5.1).

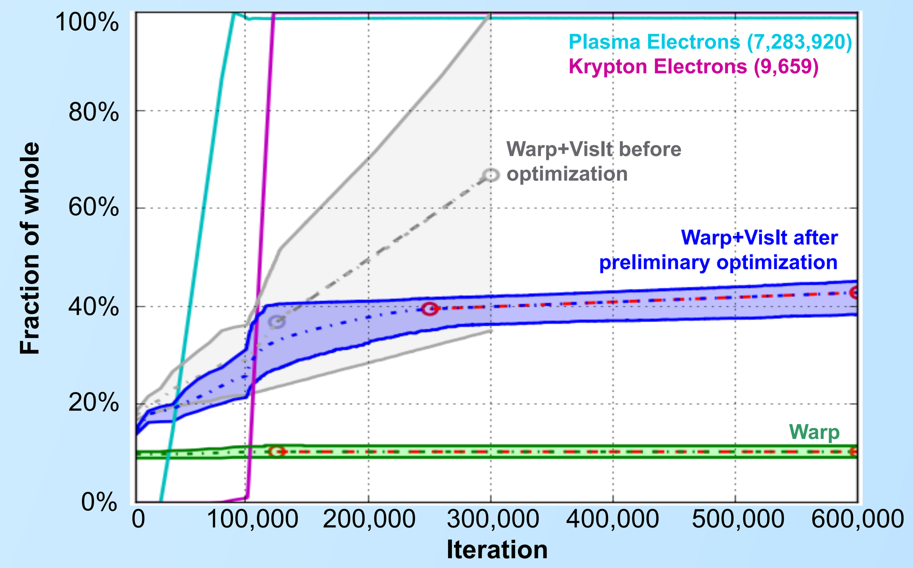


Fig. 5.1 Per-processor memory usage for a laser-plasma accelerator simulation using 768 Processes. During *in situ* visualization, we generated every 500 simulation steps four complex particle data views (see Fig 8.1 below)

6) *In Situ* Filtering and Query-driven Visualization

A central challenge in the analysis of complex particle simulations arises from the fact that while 10^7 to 10^9 particles are required for accurate simulation, only a small fraction of the particles form features of interest, e.g., a particle beam.

To address this challenge, we have extended Warp+VisIt via the concept of filtered species. Filtered species define custom, derived particle species while exposing to the analysis the same species interface. This concept enables: i) flexible *in situ* analysis of particle features, ii) analysis and collection of data subsets of interest at higher temporal frequency, and iii) reduces cost for subsequent visualization and I/O. Users can easily extend the filtered species design via custom filters while a set of customizable, reusable particle filters—e.g., threshold, particle Id, accumulative query, and halo filter—facilitate common use cases.

In future we plan to integrate the index/query system FastBit [5] with Warp to accelerate *in situ* particle query and enable *in situ* query-driven data exploration.

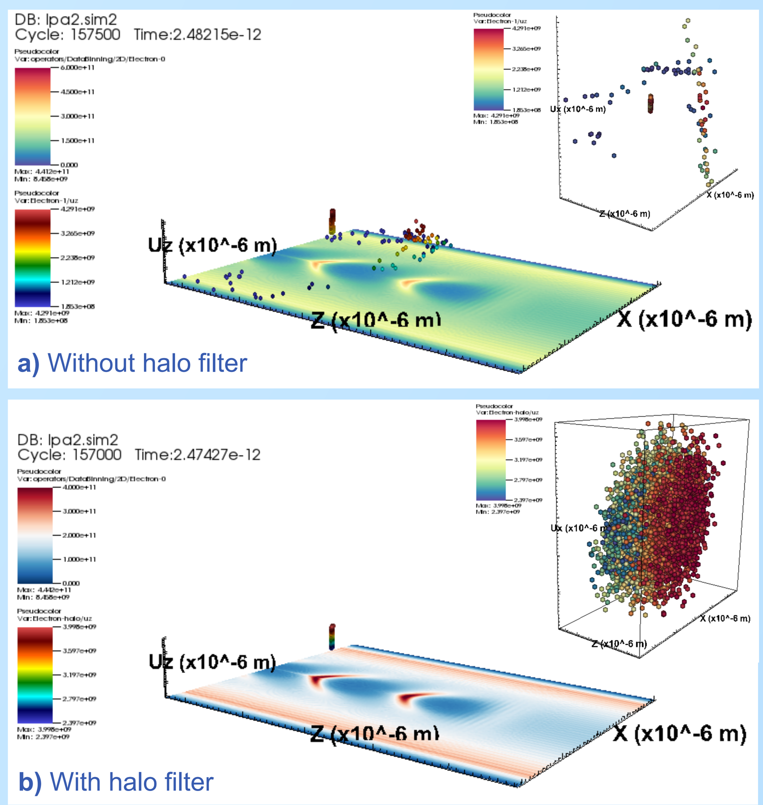


Fig. 6.1 Using the concept of *in situ* query via filtered species enables the visualization and analysis to focus on the main particle beam of interest.

7) H5Hut: High-performance I/O for Particle-based Simulations

Large-scale particle-based simulations generate enormous amounts of particle- and field-based data. To enable effective post-processing and visualization and analysis of this data, high-performance I/O and common data standards are critical. H5Hut [2] is an easy-to-use, high-performance I/O library, which implements several data models aimed at particle-based simulations that standardize the data storage and encapsulate the complexity of parallel HDF5. We have developed a Python API for H5Hut to enable the integration of H5Hut with Warp for high-performance parallel I/O.

8) Results:

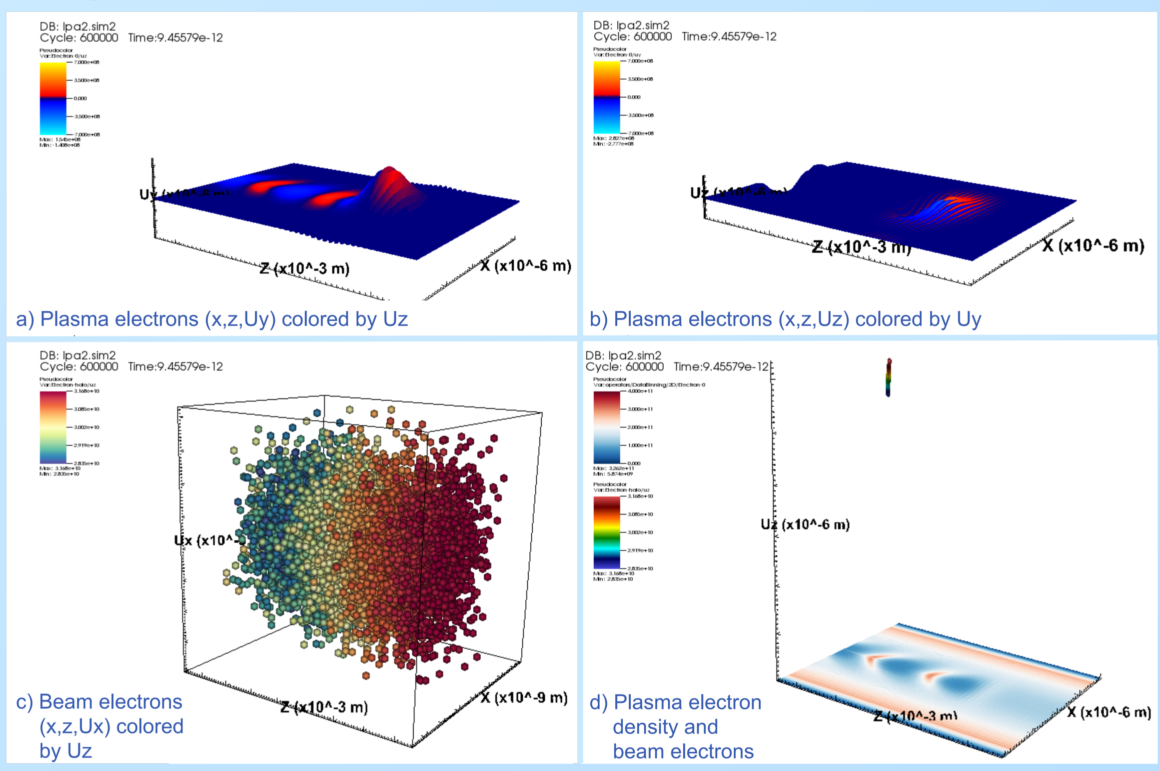


Fig. 8.1 Visualization of a two-color laser-plasma accelerator simulation [1] implemented in Warp and using VisIt for *in situ* visualization. The simulation and *in situ* visualization were performed on Edison@NERSC using 768 cores.

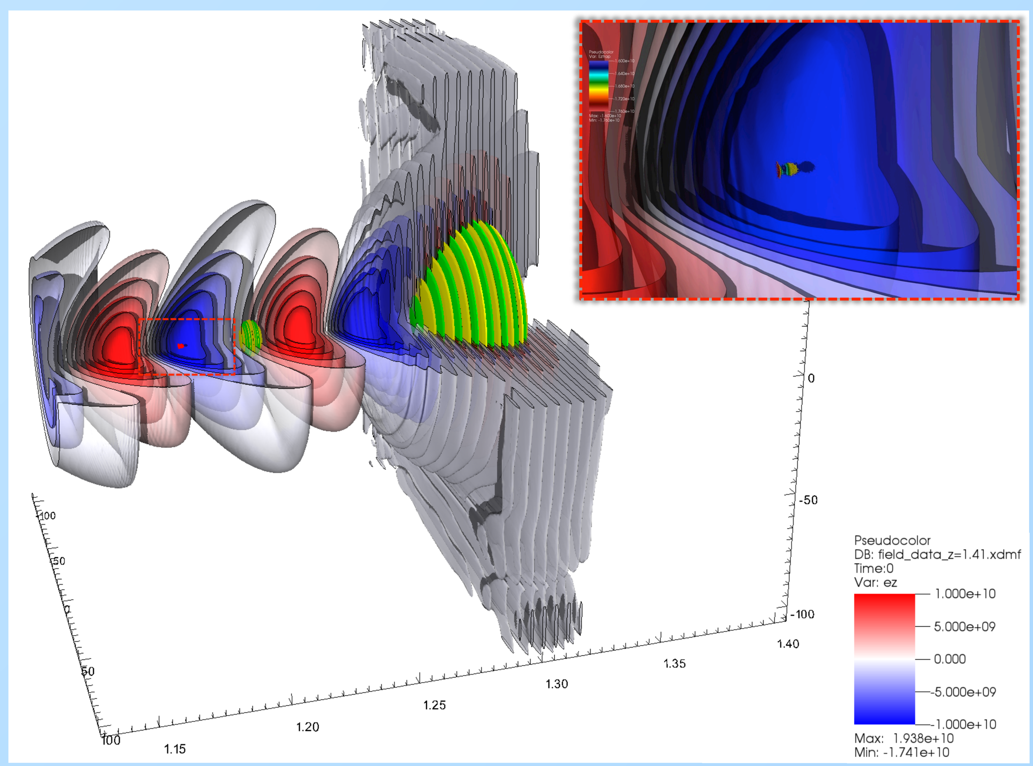


Fig. 8.2 Visualization of a 3D laser-plasma accelerator simulation showing i) the two laser pulses (yellow/green), ii) the wake (red/blue), and iii) the particle beam (red). The inset shows a zoom-in of the beam.

SDAV
Scalable Data Management, Analysis, and Visualization

[1] L.-L. Yu, E. Esarey, C. B. Schroeder, J.-L. Vay, C. Benedetti, C. G. R. Geddes, M. Chen, and W. P. Leemans, "Two-color laser ionization injection," Phys. Rev. Lett. 112, 125001 (2014)
[2] M. Howison, A. Adelmann, E. W. Bethel, A. Gsell, B. Oswald, Prabhat. "H5hut: A High-Performance I/O Library for Particle-based Simulations." In IASDS10, Heraklion, Crete, Greece (2010).
[3] B. Whitlock, J. M. Favre, and J. S. Meredith. "Parallel in situ coupling of simulation with a fully featured visualization system." Proceedings of the 11th Eurographics conference on Parallel Graphics and Visualization, 2011
[4] J.-L. Vay, D. P. Grote, R. H. Cohen, and A. Friedman, "Novel methods in the Particle-In-Cell accelerator Code-Framework Warp," IOP Computational Science & Discovery, 5 014019 (2012)
[5] K. Wu et al. "FastBit: interactively searching massive data," IOP Journal of Physics: Conference Series, 180 012053 (2009)