Next Generation Computational Tools for the Modeling and Design of Particle Accelerators at Exascale

Axel Huebl
Lawrence Berkeley National Laboratory
On behalf of the BLAST team (lead: Jean-Luc Vay @ LBNL)
LBNL, LLNL, SLAC, CEA, DESY, Modern Electron, CERN

Invited Oral - TUYE2
Computing and Data Science for Accelerator Systems

North American Particle Accelerator Conference (NAPAC22)

location: Albuquerque, NM, USA
August 9th, 2022
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. This work was supported by the Laboratory Directed Research and Development Program of Lawrence Berkeley National Laboratory under U.S. Department of Energy Contract No. DE-AC02-05CH11231. 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 using NERSC award ASCR-ERCAP0022112. This work used computational resources of the supercomputer Fugaku provided by RIKEN through the HPCI System Research Project (Project ID: ra010013)
- BLAST: Beam pLasma Accelerator Simulation Toolkit
- IO, Standardization & Open Development
- HPC: The Exascale Computing Project and Beyond
Simulation of Beam Sources & Dynamics Requires Different Types of PIC Codes

Imagine a future, *hybrid particle accelerator*, e.g., with conventional and plasma elements.

**Goal**
Start-to-end modeling in an open software ecosystem.
Simulation of Beam Sources & Dynamics Requires Different Types of PIC Codes

Imagine a future, *hybrid particle accelerator*, e.g., with conventional and plasma elements.

**Goal**
Start-to-end modeling in an open software ecosystem.
Simulation of Beam Sources & Dynamics Requires Different Types of PIC Codes

Imagine a future, **hybrid particle accelerator**, e.g., with conventional and plasma elements.

**Goal**
Start-to-end modeling in an open software ecosystem.

3D visualization of the plasma proton density during the acceleration process of a few-fs, 1.15nC beam
Hilz, Ostermayr, Huebl et al.; Nat. Comm. 9.432, 2018
Imagine a future, *hybrid particle accelerator*, e.g., with conventional and plasma elements.

**t-based electrostatic or electromagnetic PIC**

**Quasistatic PIC**
*separates the timescale for plasma wakefield and beam evolution*

**Goal**
Start-to-end modeling in an open software ecosystem.

3D visualization of the plasma proton density during the acceleration process of a few-fs, 1.15nC beam.

Hilz, Ostermayr, Huebl et al.; Nat. Comm. 9.432, 2018
Simulation of Beam Sources & Dynamics Requires Different Types of PIC Codes

Imagine a future, **hybrid particle accelerator**, e.g., with conventional and plasma elements.

Quasistatic PIC separates the timescale for plasma wakefield and beam evolution

**Goal**
Start-to-end modeling in an open software ecosystem.

s-based PIC uses \( s \) instead of \( t \) as independent variable + symplectic maps for accelerator elements
Imagine a future, **hybrid particle accelerator**, e.g., with conventional and plasma elements.

Quasistatic PIC separates the timescale for plasma wakefield and beam evolution.

**Goal**

Start-to-end modeling in an open software ecosystem.

Left: Laser-driven wakefield accelerator (LWFA) stage with the drive laser propagating to the right shown in red; right: plasma wakefield accelerator (PWFA) driven by the electron beam from the LWFA stage (figure credits: Thomas Heinemann/Strathclyde and Alberto Martinez de la Ossa/DESY). T. Kurz, T. Heinemann, et al. Nat. Comm. 12.2895 (2021)
Imagine a future, hybrid particle accelerator, e.g., with conventional and plasma elements.

**t-based electrostatic or electromagnetic PIC**

Quasistatic PIC separates the timescale for plasma wakefield and beam evolution.

**Goal**
Start-to-end modeling in an open software ecosystem.

s-based PIC uses $s$ instead of $t$ as independent variable + symplectic maps for accelerator elements.
Imagine a future, **hybrid particle accelerator**, e.g., with conventional and plasma elements.

**S-based PIC** uses $s$ instead of $t$ as independent variable + symplectic maps for accelerator elements.

**Quasistatic PIC** separates the timescale for plasma wakefield and beam evolution.

**Goal**: Start-to-end modeling in an open software ecosystem.
Simulation of Beam Sources & Dynamics Requires Different Types of PIC Codes

Imagine a future, **hybrid particle accelerator**, e.g., with conventional and plasma elements.

Quasistatic PIC separates the timescale for plasma wakefield and beam evolution.

**Goal**
Start-to-end modeling in an open software ecosystem.

modeling of radiative & space-charge effects

**s-based PIC uses s instead of t as independent variable + symplectic maps for accelerator elements**
Simulation of Beam Sources & Dynamics Requires Different Types of PIC Codes

Imagine a future, **hybrid particle accelerator**, e.g., with conventional and plasma elements.

**Quasistatic PIC**
- Separates the timescale for plasma wakefield and beam evolution

**(S)RF Gun**
- LPA/LPI
- Source

**WarpX**
- T-based electrostatic or electromagnetic PIC

**HiPACE++**
- S-based PIC uses $s$ instead of $t$ as independent variable + symplectic maps for accelerator elements

**ImpactX**

**Legend**
- **BLAST**: Exascale
- **Exascale Computing Project**
- **SciDAC**: Scientific Discovery through Advanced Computing

---

**Goal**
- Start-to-end modeling in an open software ecosystem.
Simulation of Beam Sources & Dynamics Requires Different Types of PIC Codes

Imagine a future, **hybrid particle accelerator**, e.g., with conventional and plasma elements.

*Quasistatic PIC separates the timescale for plasma wakefield and beam evolution*

**Goal**
Start-to-end modeling in an open software ecosystem.

**Legend**
- in BLAST
- BLAST: Exascale
- other

**ImpactX**
- s-based PIC uses s instead of t as independent variable + symplectic maps for accelerator elements
Ultimate goal: offer on-the-fly tunability of physics & numerics complexity to users

Great for ensemble runs for design studies

- Reduced physics
- 1D-1V
- Low resolution
- Surrogate models

Great for detailed runs for physics studies

- Full physics
- 3D-3V
- High resolution
- First principles

Goal
Start-to-end modeling in an open software ecosystem.

Start-to-End Modeling R&D
- advanced models: numerics, AI/ML surrogates
- speed & scalability: team science with computer sci.
- flexibility & reliability: modern software ecosystem
Overview of the Particle-In-Cell code WarpX

Available Particle-in-Cell Loops
- electrostatic & electromagnetic (fully kinetic)

Advanced algorithms
- boosted frame, spectral solvers, Galilean frame, embedded boundaries + CAD, MR, ...

Multi-Physics Modules
- field ionization of atomic levels, Coulomb collisions, QED processes (e.g. pair creation), macroscopic materials
Overview of the Particle-In-Cell code WarpX

Available Particle-in-Cell Loops
- electrostatic & electromagnetic (fully kinetic)

\[ x, v = f(E, B) \]

Push particles

\[ E, B = f(I) \]

Solve fields

\[ I = f(x, v) \]

Deposit currents

Gather fields

Geometries
- 1D3V, 2D3V, 3D3V and RZ (spectral cylindrical)

Multi-Node parallelization
- MPI: 3D domain decomposition
- dynamic load balancing

On-Node Parallelization
- GPU: CUDA, HIP and SYCL
- CPU: OpenMP

Advanced algorithms
boosted frame, spectral solvers, Galilean frame, embedded boundaries + CAD, MR, ...

Multi-Physics Modules
field ionization of atomic levels, Coulomb collisions, QED processes (e.g. pair creation), macroscopic materials

Scalable, Parallel I/O
- AMReX plotfile and openPMD (HDF5 or ADIOS)
- in situ diagnostics
WarpX supports a growing number of applications

- Plasma accelerators (LBNL, DESY, SLAC)
- Laser-ion acceleration - advanced mechanisms (LBNL)
- Plasma mirrors and high-field physics + QED (CEA Saclay/LBNL)
- Plasma confinement, fusion devices (Zap Energy, Avalanche Energy)
- Laser-ion acceleration - laser pulse shaping (LLNL)
- Thermionic converter (Modern Electron)
- Pulsars, magnetic reconnection (LBNL)
- Magnetic fusion sheaths (LLNL)
- Microelectronics (LBNL) - ARTEMIS
- Pulsars, magnetic reconnection (LBNL)
Last month, we open sourced ImpactX as an early developer preview.

Particle-in-Cell Loop
- electrostatic
  - with space-charge effects (in dev.)
- s-based
  - relative to a reference particle
  - elements: symplectic maps

Fireproof Numerics
Based on IMPACT suite of codes, esp. IMPACT-Z and MaryLie

Triple Acceleration Approach
- GPU support
- Adaptive Mesh Refinement (in dev.)
- AI/ML & Data Driven Models (in dev.)
github.com/ECP-WarpX/impactx
Last month, we open sourced ImpactX as an early developer preview.

Particle-in-Cell Loop
- electrostatic
  - with space-charge effects (in dev.)
- s-based
  - relative to a reference particle
  - elements: symplectic maps

Fireproof Numerics
based on IMPACT suite of codes, esp. IMPACT-Z and MaryLie

Triple Acceleration Approach
- GPU support
- Adaptive Mesh Refinement (in dev.)
- AI/ML & Data Driven Models (in dev.)

User-Friendly
- single-source C++, full Python control
- fully tested
- fully documented

Multi-Node parallelization
- MPI: 2D/3D domain decomposition
- dynamic load balancing (in dev.)

On-Node Parallelization
- GPU: CUDA, HIP and SYCL
- CPU: OpenMP

Scalable, Parallel I/O (in dev.)
- openPMD
- in situ analysis/visualization

github.com/ECP-WarpX/impactx
ImpactX: Physics Benchmark Examples

- FODO cell
- magnetic bunch compression chicane
- stationary beam in a const. focusing channel
- Kurth-distr. beam in periodic isotropic focusing channel
- stable FODO cell + short RF (buncher) cavities for longitudinal focusing
- chain of thin multipoles
- nonlin. focusing channel (IOTA nonlin. lens)
- Fermilab IOTA storage ring (linear optics)
ImpactX: Physics Benchmark Examples

- FODO cell
- magnetic bunch compression chicane
- stationary beam in a const. focusing channel
- Kurth-distr. beam in periodic isotropic focusing channel
- stable FODO cell + short RF (buncher) cavities for longitudinal focusing
- chain of thin multipoles
- nonlin. focusing channel (IOTA nonlin. lens)
- Fermilab IOTA storage ring (linear optics)

Berlin-Zeuthen Chicane
- rms-matched 5 GeV electron beam with initial normalized transverse rms emittance of 1 μm
- LCLS (@5GeV) & TESLA XFEL (@500MeV)-like
- longitudinal phase space: 10x compression
- emittance coupling: recovered at exit
ImpactX: Physics Benchmark Examples

- FODO cell
- magnetic bunch compression chicane
- stationary beam in a const. focusing channel
- Kurth-distr. beam in periodic isotropic focusing channel
- stable FODO cell + short RF (buncher) cavities for longitudinal focusing
- chain of thin multipoles
- nonlin. focusing channel (IOTA nonlin. lens)
- Fermilab IOTA storage ring (linear optics)

**FODO Cell**
- stable FODO lattice with a zero-current phase advance of 67.8 degrees per cell
- rms-matched 2 GeV electron beam with initial unnormalized rms emittance of 2 nm
  - test also checks if emittance stays flat
ImpactX: Physics Benchmark Examples

```python
from impactx import ImpactX, RefPart, 
    distribution, elements
sim = ImpactX()  # simulation object
# set numerical parameters and IO control
sim.set_particle_shape(2)  # B-spline order
sim.set_slice_step_diagnostics(True)
sim.set_space_charge(False)
# domain decomposition & space charge mesh
sim.init_grids()
# load a 2 GeV electron beam with an initial
# unnormalized rms emittance of 2 nm
energy_MeV = 2.0e3  # reference energy
charge_C = 1.0e-9  # used with space charge
mass_MeV = 0.510998950  # mass
qm_qeeV = -1.0e-6/mass_MeV  # charge/mass
npart = 10000  # number of macro particles

distr = distribution.Waterbag(
    sigmaX = 3.9964884770e-5,
    sigmaY = 3.9964884770e-5,
    sigmaT = 1.0e-3,
    sigmaPx = 2.662353879e-5,
    sigmaPy = 2.662353879e-5,
    sigmaPt = 2.0e-3,
    mupx = -0.846574929020762,
    mupy = 0.846574929020762,
    mutp = 0.0)
sim.add_particles(
    qm_qeeV, charge_C, distr, npart)
# set the energy in the reference particle
sim.set_energy_MeV(energy_MeV, mass_MeV)
# design the accelerator lattice
ns = 25  # number of steps slicing through ds
fodo = [
    elements.Drift(ds=0.25, nalice=ns),
    elements.Quad(ds=1.0, k=1.0, nalice=ns),
    elements.Drift(ds=0.5, nalice=ns),
    elements.Quad(ds=1.0, k=-1.0, nalice=ns),
    elements.Drift(ds=0.25, nalice=ns)]
# assign a fodo segment
sim.lattice.extend(fodo)
# run simulation
sim.evolve()
```

**FODO Cell**

- stable FODO lattice with a zero-current phase advance of 67.8 degrees per cell
- rms-matched 2 GeV electron beam with initial unnormalized rms emittance of 2 nm
  - test also checks if emittance stays flat

---

💡 Same Script

CPU/GPU & MPI

---

ECP EXASCALE COMPUTING PROJECT
BROOKLYN LAB

LDRD
bare (linear) lattice of the Fermilab IOTA storage ring; an rms-matched proton beam with an un-normalized emittance of 4.5 μm propagates over a single turn.
bare (linear) lattice of the Fermilab IOTA storage ring; an rms-matched proton beam with an un-normalized emittance of 4.5 μm propagates over a single turn

Preservation of Second Moments
- nnl. element: conserve invariants of motion
- check emittance preservation
- rms beam size evolution:
  IMPACT-Z vs ImpactX

Preliminary Performance
- on Perlmutter (NERSC) CPU / GPU
- order-of-magnitude perf. w/o dyn. LB (yet)
An open interface with the community

Online Documentation: warpx|hipace|impactx.readthedocs.io

Open-Source Development & Benchmarks: github.com/ECP-WarpX

188 physics benchmarks run on every code change of WarpX
8 physics benchmarks + 32 tests for ImpactX
An open interface with the community

Online Documentation:
warpx|hipace|impactx.readthedocs.io

Open-Source Development & Benchmarks:
github.com/ECP-WarpX

Rapid and easy installation on any platform:

- python3 -m pip install .
- brew tap ecp-warpx/warpx
- brew install warpx
- conda install -c conda-forge warpx
- spack install warpx
- spack install py-warpx
- cmake -S . -B build
- cmake --build build --target install
- module load warpx
- module load py-warpx

188 physics benchmarks run on every code change of WarpX
8 physics benchmarks + 32 tests for ImpactX
Portable Performance through Exascale Programming Model

**AMReX library**

- Domain decomposition & MPI communications: MR & load balance

**Performance-Portability Layer:** GPU/CPU/KNL

![Graph showing the number of particles per ns (full PIC loop) vs. number of Summit nodes.](image)

A. Myers et al., “Porting WarpX to GPU-accelerated platforms,” Parallel Computing 108, 102833 (2021)
Portable Performance through Exascale Programming Model

AMReX library

- Domain decomposition & MPI communications: MR & load balance
- Performance-Portability Layer: GPU/CPU/KNL

| Data Structures |
|-----------------|
| without tiling  |
| with tiling     |

- Write the code once, specialize at compile-time
- Parallel linear solvers (e.g. multi-grid Poisson solvers)
- Embedded boundaries
- Runtime parser for user-provided math expressions (incl. GPU)

A. Myers et al., “Porting WarpX to GPU-accelerated platforms,” Parallel Computing 108, 102833 (2021)
Transitioning to an Integrated Ecosystem

Desktop to HPC

MPI

CUDA, OpenMP, SYCL, HIP
Transitioning to an Integrated Ecosystem

AMReX

Containers, Communication, Portability, Utilities

MPI

CUDA, OpenMP, SYCL, HIP

FFT
on- or multi-device

Lin. Alg.
BLAS++ LAPACK++

Desktop to HPC

mac OS

BLAST
BEAM PLASMA & ACCELERATOR SIMULATION TOOLKIT
Transitioning to an Integrated Ecosystem

AMReX
Containers, Communication, Portability, Utilities

Diagnostics
I/O code coupling
openPMD
ADIOS
S2
HD
F5
ZFP

FFT
on- or multi-device
openPMD
ADIOS
S2
HD
F5
ZFP

Lin. Alg.
BLAS++
LAPACK++

MPI

CUDA, OpenMP, SYCL, HIP

- PByte-scale
- TByte/s Bandwidth

Desktop to HPC

3.7 TB         7.5 TB       15.1 TB        30.2 TB         60 TB           120 TB         240 TB

● PByte-scale
● TByte/s Bandwidth
Transitioning to an Integrated Ecosystem

**AMReX**
- Containers, Communication, Portability, Utilities

**MPI**

**CUDA, OpenMP, SYCL, HIP**

**Diagnostics**
- I/O code coupling
  - openPMD
  - ADIO
  - HD
  - S2
  - F5
  - ZFP
  - Ascend
  - VTK
  - -m

**FFT**
- on- or multi-device

**Lin. Alg.**
- BLAS++
- LAPACK++

**ABLASTR library**: common PIC physics

**PICSAR**
- QED Modules

**QED events**
Transitioning to an Integrated Ecosystem

- **WarpX** full PIC, LPA/LPI
- **HiPACE++** quasi-static, PWFA
- **ARTEMIS** microelectronics
- **ImpactX** accelerator lattice design
- **PICSAR** QED Modules
- AMReX Containers, Communication, Portability, Utilities
- **ABLASTR library**: common PIC physics
- Diagnostics I/O code coupling
  - openPMD
  - ADIOS2
  - HD-F5
  - ZFP
- FFT on- or multi-device
  - openPMD
  - ADIOS2
  - HD-F5
  - ZFP
- **Lin. Alg.** BLAS++ LAPACK++
- MPI
- **CUDA, OpenMP, SYCL, HIP**
Transitioning to an Integrated Ecosystem

- WarpX: full PIC, LPA/LPI
- HiPACE++: quasi-static, PWFA
- ARTEMIS: microelectronics
- ImpactX: accelerator lattice design
- Containers, Communication, Portability, Utilities
- AMReX: Python: Modules, PICMI interface, Workflows
- pyAMReX
- PICSAR: QED Modules
- ABLASTR library: common PIC physics
- Diagnostics & I/O code coupling
  - openPMD
  - ADIOS
  - HD F5
  - ZFP
- FFT: on- or multi-device
  - Ascent
  - VTK-m
- Lin. Alg.
  - BLAS++ LAPACK++
- CUDA, OpenMP, SYCL, HIP
- Desktop to HPC
- HiPACE++ quasi-static, PWFA
- ARTEMIS microelectronics
- ImpactX accelerator lattice design
- Python: Modules, PICMI interface, Workflows
- pyAMReX
- PICSAR: QED Modules
- ABLASTR library: common PIC physics
- Diagnostics & I/O code coupling
  - openPMD
  - ADIOS
  - HD F5
  - ZFP
- FFT: on- or multi-device
  - Ascend
  - VTK-m
- Lin. Alg.
  - BLAS++ LAPACK++
- CUDA, OpenMP, SYCL, HIP
- BLAST: Beam pLasma Accelerator Simulation Toolkit
- IO, Standardization & Open Development
- HPC: The Exascale Computing Project and Beyond
We Standardize & Develop Scalable Data Methods

Start-to-end accelerator modeling requires data compatibility and control usability.

Figure 8: Longitudinal electric field (in V/m) in a laser-driven plasma acceleration stage at two times (top: $t \approx 300\text{fs}$, bottom: $t \approx 600\text{fs}$) along the laser propagation from 2-D PIC simulations with (left) Warp; (right) Osiris. Plots are based on rendering from the openPMD-viewer.
We Standardize & Develop Scalable Data Methods

Start-to-end accelerator modeling requires data compatibility and control usability.

Figure 8: Longitudinal electric field (in V/m) in a laser-driven plasma acceleration stage at two times (top: $t \approx 300$fs, bottom: $t \approx 600$fs) along the laser propagation from 2-D PIC simulations with (left) Warp; (right) Osiris. Plots are based on rendering from the openPMD-viewer.
Start-to-end accelerator modeling requires data compatibility and control usability

We Standardize & Develop Scalable Data Methods

Consortium for Advanced Modeling of Particle Accelerators CAMPA

Communities

DOE HEP GARD - now DOE SciDAC-5

Figure 8: Longitudinal electric field (in V/m) in a laser-driven plasma acceleration stage at two times (top: $t \approx 300$fs, bottom: $t \approx 600$fs) along the laser propagation from 2-D PIC simulations with (left) Warp; (right) Osiris. Plots are based on rendering from the openPMD-viewer.
openPMD: Open Standard for Particle-Mesh Data

- **markup / schema for arbitrary** hierarchical data formats
- truly, scientifically
  **self-describing**
- basis for **open data workflows**
openPMD: Open Standard for Particle-Mesh Data

- **markup / schema for arbitrary** hierarchical data formats
- truly, scientifically **self-describing**
- basis for **open data workflows**

**openPMD standard** (1.0.0, 1.0.1, 1.1.0)
the underlying file markup and definition
A Huebl et al., DOI:10.5281/zenodo.33624

**base standard**
general description
wavefronts, particle species, particle beams, weighted particles, PIC, MD, mesh-refinement, CCD images, ...

**extensions**
domain specific
openPMD: Open Standard for Particle-Mesh Data

- **markup** / schema for arbitrary hierarchical data formats
- truly, scientifically self-describing
- basis for open data workflows

**openPMD standard** (1.0.0, 1.0.1, 1.1.0)
the underlying file markup and definition
A Huebl et al., DOI:10.5281/zenodo.33624

**base standard**
genral description

**extensions**
domain specific
wavefronts, particle species, particle beams, weighted particles, PIC, MD, mesh-refinement, CCD images, ...

**ADIOS**

**HDF**

**JSON**
openPMD: Open Standard for Particle-Mesh Data

- **markup / schema for arbitrary** hierarchical data formats
- **truly, scientifically** self-describing
- **basis for open data workflows**

openPMD-standard (1.0.0, 1.0.1, 1.1.0)
the underlying file markup and definition
A Huebl et al., DOI:10.5281/zenodo.33624

| base standard | extensions |
|---------------|------------|
| general description | domain specific |
| wavefronts, particle species, particle beams, weighted particles, PIC, MD, mesh-refinement, CCD images, ... |

openPMD-viewer
quick visualization
explore, e.g., in Jupyter

openPMD-api
reference library
file-format agnostics API

openPMD-updater
auto-update to new standard, verify
openPMD-validator
We Standardize & Develop Scalable Data Methods

... and integrate them for scientific productivity including data analytics frameworks & graphical user interfaces
We Standardize & Develop Scalable Data Methods

... and integrate them for scientific productivity including data analytics frameworks & graphical user interfaces
We Standardize & Develop Scalable Data Methods

... and integrate them for scientific productivity
including data analytics frameworks & graphical user interfaces

Open standardization, i.e. openPMD, makes us flexible for I/O libraries, tooling & domain-science needs.
- BLAST: Beam pLasma Accelerator Simulation Toolkit
- IO, Standardization & Open Development
- HPC: The Exascale Computing Project and Beyond
WarpX: Runs Efficiently on the *First Exascale Supercomputer*

April-July 2022: ran on **world’s largest HPCs**
L. Fedeli, A. Huebl et al., *accepted* in SC’22, 2022

*Note: Perlmutter & Frontier are pre-acceptance!*

Demonstrated scaling **4-5 orders** of magnitude
WarpX: Runs Efficiently on the First Exascale Supercomputer

April-July 2022: ran on world’s largest HPCs
L. Fedeli, A. Huebl et al., accepted in SC’22, 2022

Note: Perlmutter & Frontier are pre-acceptance!

Demonstrated scaling 4-5 orders of magnitude
WarpX: Runs Efficiently on the First Exascale Supercomputer

April-July 2022: ran on world’s largest HPCs
L. Fedeli, A. Huebl et al., accepted in SC’22, 2022

Note: Perlmutter & Frontier are pre-acceptance!

Demonstrated scaling 4-5 orders of magnitude

Figure-of-Merit over time

| Date | Machine | N./Node | Nodes | FOM   |
|------|---------|---------|-------|-------|
| 3/19 | Cori    | 0.4e7   | 6625  | 1.0e11|
| 6/19 | Summit  | 2.8e7   | 1000  | 7.8e11|
| 9/19 | Summit  | 2.3e7   | 2560  | 6.8e11|
| 1/20 | Summit  | 2.3e7   | 2560  | 1.0e12|
| 2/20 | Summit  | 2.5e7   | 4263  | 1.2e12|
| 6/20 | Summit  | 2.0e7   | 4263  | 1.4e12|
| 7/20 | Summit  | 2.0e8   | 4263  | 2.5e12|
| 3/21 | Summit  | 2.0e8   | 4263  | 2.9e12|
| 6/21 | Summit  | 2.0e8   | 4263  | 2.6e12|
| 7/21 | Perlmutter | 2.7e8 | 960   | 1.1e12|
| 12/21| Summit  | 2.0e8   | 4263  | 3.3e12|
| 4/22 | Perlmutter | 4.0e8 | 928   | 1.0e12|
| 4/22 | Perlmutter | 4.0e8 | 928   | 1.4e12|
| 4/22 | Summit  | 2.0e8   | 4263  | 3.4e12|
| 4/22 | Fugaku  | 3.1e6   | 98304 | 8.1e12|
| 6/22 | Perlmutter | 4.4e8 | 1088  | 1.0e12|
| 7/22 | Fugaku  | 3.1e6   | 98304 | 2.2e12|
| 7/22 | Fugaku  | 3.1e6   | 152064| 9.3e12|
| 7/22 | Frontier | 8.1e8   | 8576  | 1.1e13|
GPU Computing at Scale Requires Advanced Load Balancing

Application Challenges

- Plasma Mirrors & Laser-Ion Acceleration: moving front
- Laser Wakefield Accelerator: Injected Beam Particles

M. Rowan, A. Huebl, K. Gott, R. Lehe, M. Thévenet, J. Deslippe, J.-L. Vay, “In-Situ Assessment of Device-Side Compute Work for Dynamic Load Balancing in a GPU-Accelerated PIC Code,” PASC21, DOI:10.1145/3468267.3470614 (2021)
GPU Computing at Scale Requires Advanced Load Balancing

Application Challenges

- Plasma Mirrors & Laser-Ion Acceleration: moving front
- Laser Wakefield Accelerator: Injected Beam Particles

In Situ Cost Analysis

- basis for distribution functions
- realistic cost: kernel timing

Result: 3.8x speedup!

- production-quality, easy-to-use
- larger simulation: mitigate local memory spikes

M. Rowan, A. Huebl, K. Gott, R. Lehe, M. Thévenet, J. Deslippe, J.-L. Vay, “In-Situ Assessment of Device-Side Compute Work for Dynamic Load Balancing in a GPU-Accelerated PIC Code,” PASC21, DOI:10.1145/3468267.3470614 (2021)
Novel Visualization Techniques

Particle Adaptive Sampling

- emphasis on “uncommon” properties
- inverse sampling to incidence of a property

A. Biswas et al., “In Situ Data-Driven Adaptive Sampling for Large-scale Simulation Data Summarization,” ISAV18 @SC18 (2018)
Novel Visualization Techniques

Particle Adaptive Sampling
- emphasis on “uncommon” properties
- inverse sampling to incidence of a property

Physics-Informed Flow Tracelines
- traditional flow vis. depends only on local field values
- plasma particles:
  - inert: track relativistic momentum on a traceline
  - Lorentz-Force: 6 fields (electromag.), leap-frog
- chance to significantly reduce particle I/O in real-life workflows through savings on temporal fidelity

A. Biswas et al., “In Situ Data-Driven Adaptive Sampling for Large-scale Simulation Data Summarization,” ISAV18 @SC18 (2018)
Postdocs Welcome - Come work with us!

jobs.lbl.gov/jobs/search/3151872

- **Modeling & Theory**
  - Exascale & Wakefields #92244
  - Beam Dynamics & ML #96603

- **Experiment**
  - Wakefields, kHz-MHz (LPA) #96321 #93729
  - Laser-Proton/Ion (LPI) #95498
**Summary**

- **BLAST** is an open suite of PIC codes for **particle accelerator modeling**, increasingly build on top of the AMReX library, using code-sharing through the ABLASTR library and leveraging the U.S. DOE Exascale software stack. **ECP WarpX** is our first Exascale app, for relativistic plasma & beam modeling; **ImpactX** enhances these developments with AI/ML for s-based beam dynamics.

- **AMReX** for CPU/GPU Mesh-Refinement, **ABLASTR** shares PIC methods
  - Portable CPU/GPU frameworks that avoid code duplication
  - Efficient data structures, memory & comms.
  - Reuse numerical methods in various PIC loops

- **Vibrant Ecosystem and Contributions**
  - Runs on any platform: Linux, macOS, Windows
  - Specialized codes & advanced physics modules (QED, collisions, ionization, ...)
  - Advanced computer science research (load-balancing, I/O, visualization, ...)
  - Public development, automated testing, review & documentation
  - Friendly, open & helpful community

---

**Presented by:** Axel Huebl (LBNL)

[axelhuebl@lbl.gov](mailto:axelhuebl@lbl.gov)

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

[github.com/ECP-WarpX](https://github.com/ECP-WarpX)