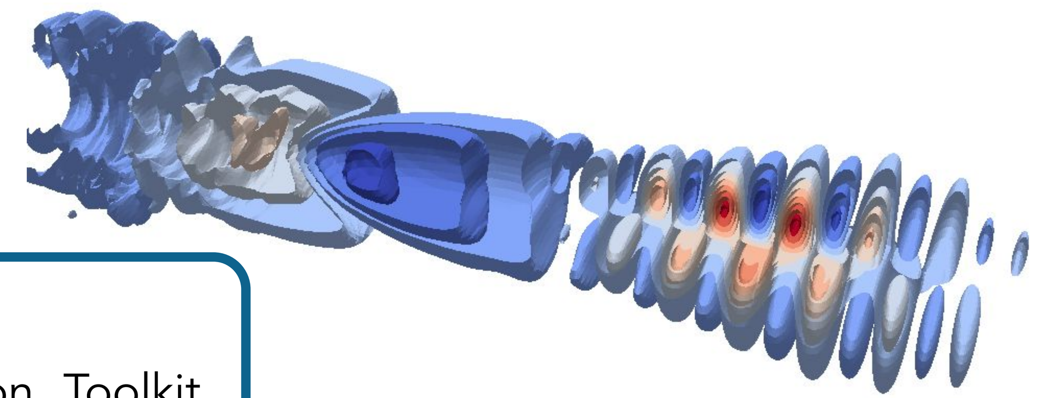
Design Optimization and In Situ Surrogate Modeling Activities in the Beam, Plasma & Accelerator Simulation Toolkit (BLAST)

A. Huebl, R. Lehe, J.-L. Vay – *LBNL* Á. Ferran Pousa, S. Jalas, M. Kirchen, A. Martinez de la Ossa, M. Thévenet – *DESY* J. Larson, S. Hudson – *ANL*



In situ visualization of a laser-driven plasma accelerator modelled with *WarpX* using the *boosted frame* technique.

Goal

Particle accelerators are a vital part of the DOE-supported infrastructure of discovery science and applications, but demand increasingly sophisticated computational tools for their design and optimization. Integrating and advancing promising technologies, such as plasma-based sources and boosters, into mainstream scientific tools depends critically on high-performance, high-fidelity modeling of complex processes. We want to provide a versatile, extensible and ultra-fast modeling framework for particle accelerator design and research that takes advantage of the latest hardware & modern software engineering.

Solution

The Beam, Plasma & Accelerator Simulation Toolkit (BLAST) is set of open-source particle accelerator codes with unique algorithmic versatility, from plasma based accelerators to beam transport dynamics and beam-beam interaction. We combine the speed boosts of Graphical Processing Units (GPUs), adaptive mesh-refinement (AMR) and Artificial Intelligence / Machine Learning (AI/ML) in a modular framework.

Rewritten BLAST Applications for Exascale

The U.S. DOE Exascale Computing Project (ECP) develops *open source* computer science **software technology** and computational science **applications.** That includes:

- **WarpX** is a time-integrated, electromagnetic and electrostatic particle-in-cell code
- **AMReX**, an adaptive mesh-refinement and performance portability library.

J.-L. Vay, A. Huebl et al., *Phys. Plasmas* ***28**, 023105* (2021) as well as *J. Instr.* ***16**, T10003* (2021)

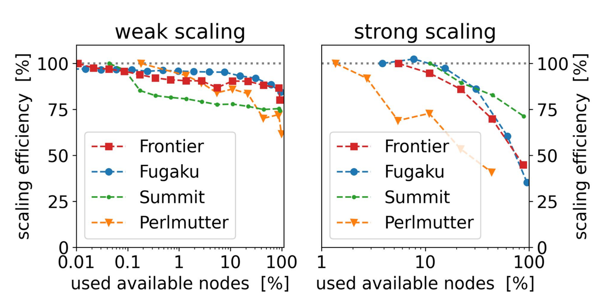
In a new seed project, we generalize the Exascale particle-in-cell routines from WarpX to redesign special purpose codes such as:

- **ImpactX**, an s-based particle tracking code with upcoming space charge effects and mesh-refinement A. Huebl et al., *NAPAC22, arXiv:2208.02382* (2022)

to establish a ***cohesive modeling framework***:

- **BLAST** Beam, Plasma & Accelerator Simulation Toolkit
- **ABLASTR** accelerated BLAST recipes

Scaling to the **full size of the world's largest** HPC machines (TOP500):



Fedeli L, Huebl A, et al., accepted in The International Conference for High Performance Computing, Networking, Storage, and Analysis (SC22), 2022

Algorithmically, we leverage and advance the *numerics and expertise* from the predecessor codes *WARP*, the *IMPACT suite* of codes & *MARYLIE*.

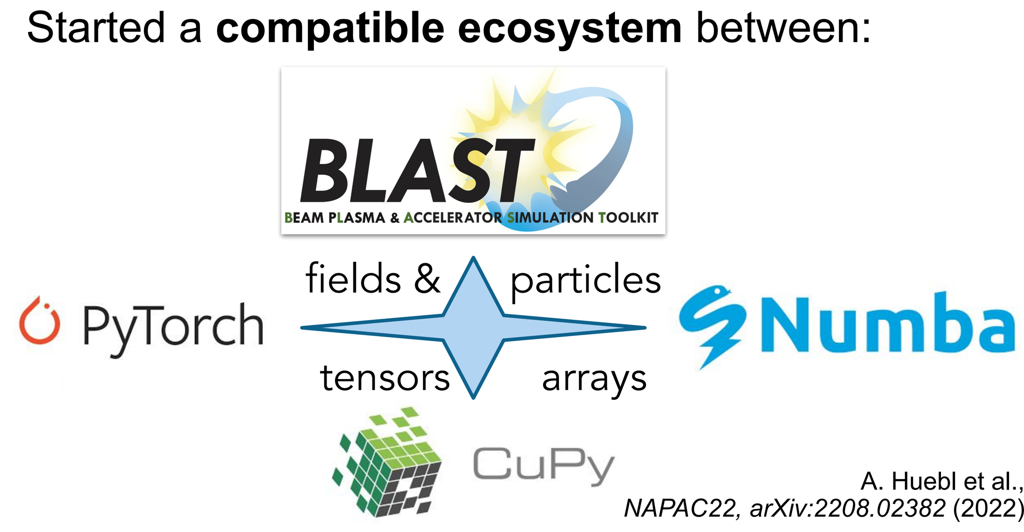
Zero-Copy ML Integration

In-the-loop coupling of GPU-accelerated exascale codes and GPU-accelerated ML models:

- persistent GPU data placement
- no transformations & copies
- Python: control and even GPU-code injection

Cross-Ecosystem, In Situ Coupling: Consortium for Python Data API Standards *data-apis.org*

Started a **compatible ecosystem** between:



A. Huebl et al., NAPAC22, arXiv:2208.02382 (2022)

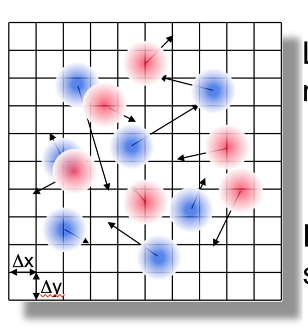
Example: pyTorch Interplay

Every **BLAST application** can reuse transparent **pyAMReX data interfaces**:

```python
import impactx
import torch

sim = impactx.ImpactX()

# ...

rho = sim.rho(lev=0)

gm = sim.Geom(lev=0)
dr = gm.data().CellSize()
dV = np.prod(dr)

rs = rho.sum_unique(comp=0, local=False)
beam_charge = dV * rs  # in C

for mfi in rho:
    rho_arr = rho.array(mfi)
    rho_torch = torch.as_tensor(rho_arr, device="cuda")
    rho_torch[:, :, :] = 42
```

Next: coupling of surrogates, such as advanced lattice elements, into **hot loops** of ImpactX.

Open Source Modules and Contributions



Lagrangian macro-particles

Eulerian fields on structured grids

Special-Purpose Particle-in-Cell Loops

- electromagnetic (fully kinetic)
- electrostatic: t- and s-based
- quasi-electrostatic (2D Poisson)

Advanced algorithms

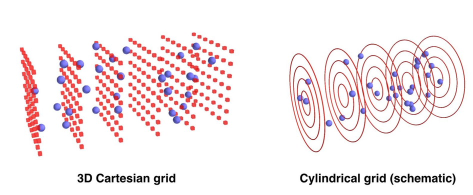
boosted frame, spectral solvers, Galilean frame, mesh-refinement, embedded boundaries, ...

Multi-Physics Modules

field ionization of atomic levels, collisional operators, QED processes (e.g. pair creation), macroscopic materials

Geometries

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



Multi-Node parallelization

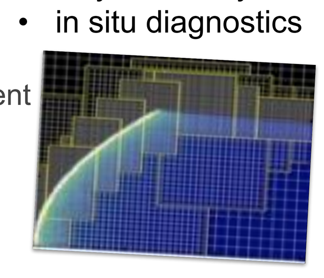
- MPI: 3D domain decomposition
- dynamic load balancing

On-Node Parallelization

- GPU: CUDA, HIP and SYCL
- CPU: OpenMP

Scalable, Parallel I/O

- AMReX plotfile and openPMD (HDF5 or ADIOS): *PBytes at TByte/s rate*
- in situ diagnostics



mesh-refinement

Beams & Lattices

- common distributions + custom file input/coupling
- lattice elements & segments: bends, quads, multipole, …

Multidisciplinary, Multi-Institutional Contributor Team

Open Science: Accessible, Usable, Testable, Contributable

Fully Open Source

- Public GitHub:
  - github.com/ECP-WarpX
  - github.com/Hi-PACE
  - github.com/openPMD
  - github.com/AMReX-Codes
- open issue reports
- support & discussions

User Friendly

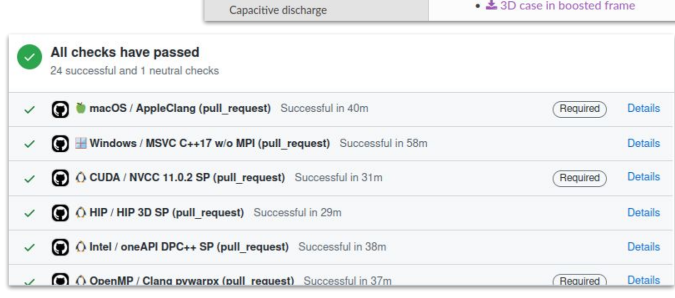
- efficient C++17 core exposed with
  - extensible python modules
  - customizable callback hooks
- documented interfaces, algorithms/theory, examples, workflows & tutorials
  - impactx.readthedocs.io hipace.readthedocs.io
  - warpx.readthedocs.io openpmd-api.rtfd.io

Reliable Releases

- package manager support
  - desktop to HPC
  - pre-compiled binaries on all major OS
- persistent identifiers & reproducible environments
- changelogs

Sustainable Development

- code reviews
  - CI
  - many benchmark cases
- continuous documentation
- permissive licensing (BSD-3-Clause-LBNL)



python3 -m pip install .

brew tap ecp-warpx/warpx
brew install warpx

cmake -S . -B build
cmake --build build --target install

conda install -c conda-forge warpx

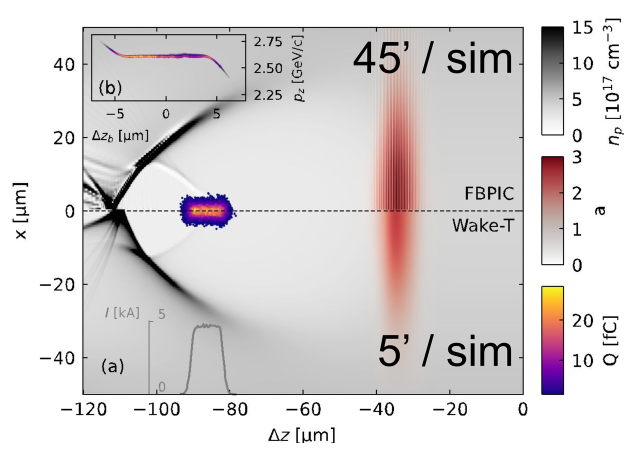
spack install warpx
spack install py-warpx

module load warpx
module load py-warpx

Bayesian Multi-Task Optimization with Different Fidelities for LWFA

Low- & high-fidelity data passed to → Multi-Task Gaussian Process (MTGP) model:

- auto-eval. **level of correlation:** low- & high-fidelity data
- where strongly-correlated: can use low-fidelity data to **inform predictions** on high-fidelity data

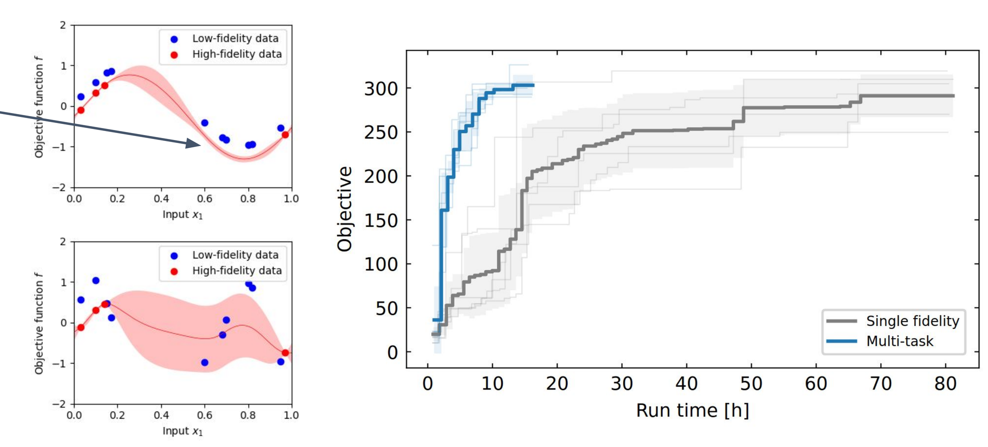


4 params: **position & long. profile** of the beam

Beam quality encoded as *objective function*:

$$f \propto \frac{Q \times E}{\sigma_E}$$

Optimization can be **drastically faster** when using the generalized, multi-task algorithm:



Á. Ferran Pousa et al., IPAC22, DOI:10.18429/JACoW-*IPAC2022-WEPOST030* (2022)
Bonilla et al., NIPS (2007); Swersky et al., NIPS (2013)
Letham et al., arxiv:1904.01049 (2019) as implemented in *Ax*

Open Source Software: Technology Stack

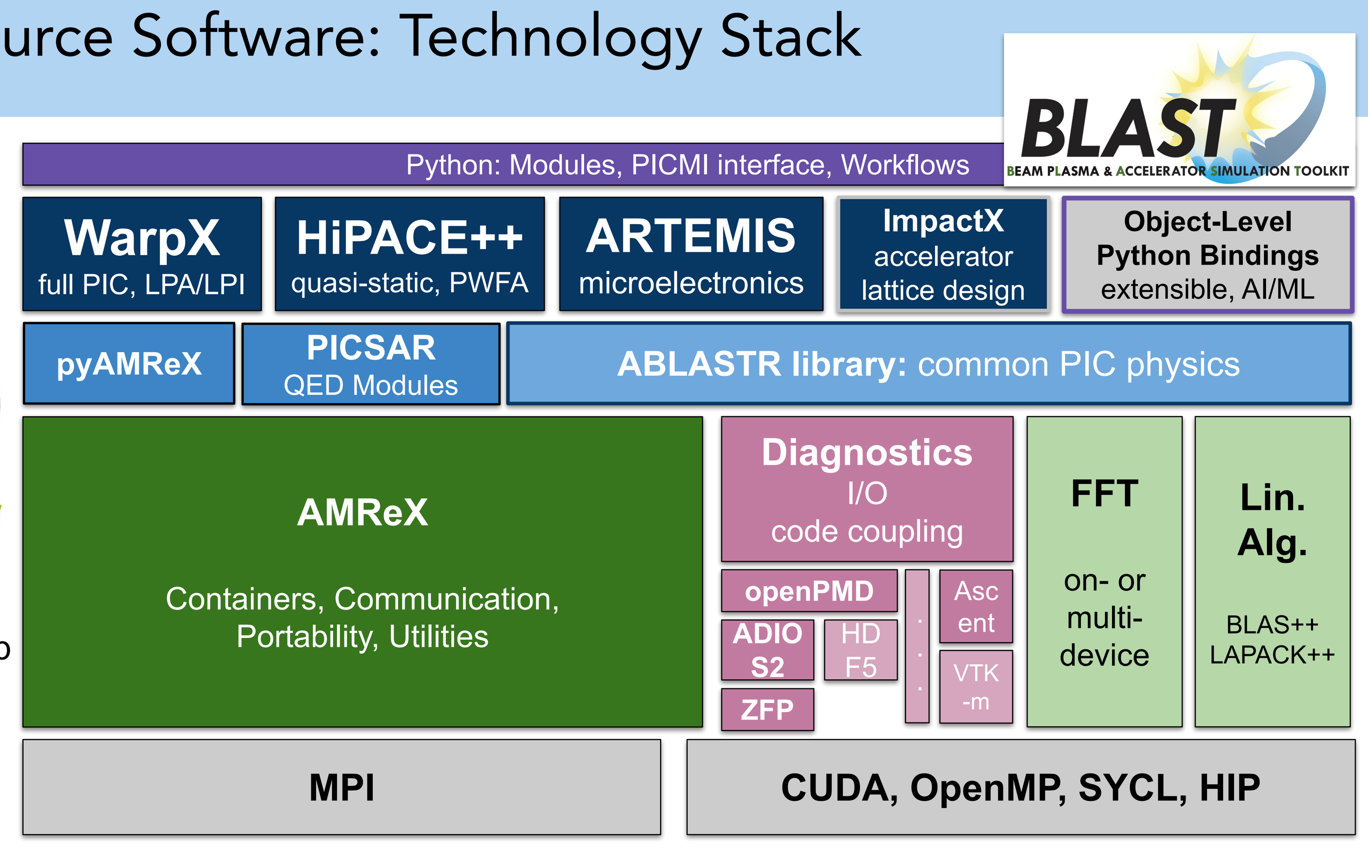
Fully GPU-accelerated

- 10-20x speedup on GPU-enabled systems over using only CPUs
- performance portable:
  - every algorithm runs on every platform
  - single-source: write physics & numerics once

From the Ground-Up Designed for

- AI/ML integration
  - zero-copy access to GPU & CPU memory
- Hybrid Particle Accelerators
  - modular and data exchangeable between segments/elements/codes
- Mesh-Refinement & Load Balancing

Desktop to HPC



Perspectives

- We establish a next-generation, fully open modeling framework that provides fast and machine-guided methods for particle accelerator R&D.
- Innovative numerical methods, such as mesh refinement & pseudo-spectral field solvers.
- Current efforts are dedicated to establishing ImpactX capabilities and expanding data-driven Python interfaces for *in-the-loop* AI/ML training & inference.

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 work 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). Used resources of the National Energy Research Scientific Computing Center and the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.