DOI:10.5281/zenodo.8362549

Exascale and ML Models for Accelerator Simulations

EXASCALE COMPUTING PROJECT

Axel Huebl

Lawrence Berkeley National Laboratory

On behalf of the WarpX, ImpactX & pyAMReX teams
LBNL, LLNL, SLAC, CEA, DESY, TAE, CERN

6th European Advanced Accelerator Concepts workshop (EAAC'23)

WG3: Theory and simulations

Elba, Italy
September 20th, 2023

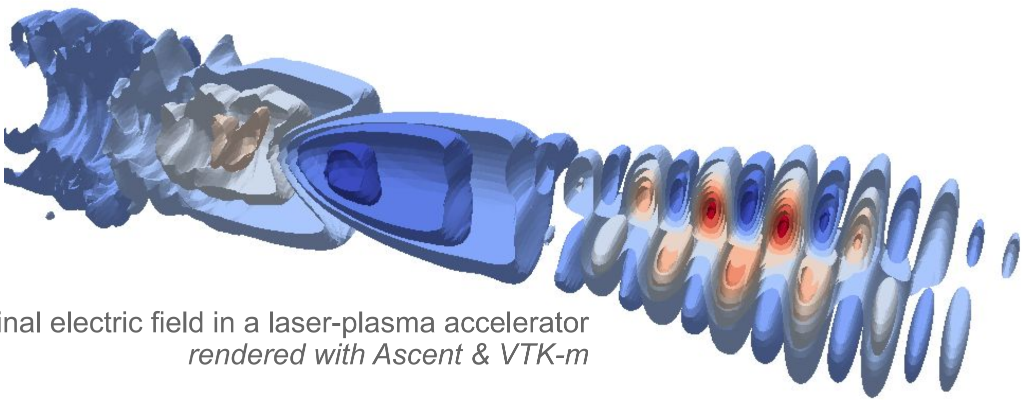
National Nuclear Security Administration

BERKELEY LAB LDRD

SciDAC Scientific Discovery through Advanced Computing

U.S. DEPARTMENT OF ENERGY | Office of Science

Funding Support



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**.

The **EAAC23 Workshop** was supported by the EU I.FAST project. This h om the European Union's **Horizon 2020** Research and Innovation programme under Grant Agreement No 101004730.

EUROPEAN NETWORK FOR NOVEL ACCELERATORS
EuroNNAc4
NPACT supported by EU via I-FAST

Outline

- **Advanced Accelerator Modeling at Exascale**
  - WarpX and ImpactX in the Beam, Plasma and Accelerator Simulation Toolkit
  - Addressing a Cambrian Explosion of Compute Architectures
  - Plasma Mirror Simulations on the Full Size of the First Exascale Supercomputer

- **Across Scales: Advanced and Conventional Accelerators**
  - Connecting Exascale and ML
  - ML-Enabled, Hybrid Beamlines

Advanced Accelerator Modeling at Exascale

Ultimate goal: *virtual accelerator* with *on-the-fly tunability* of physics & numerics complexity to users

Fast

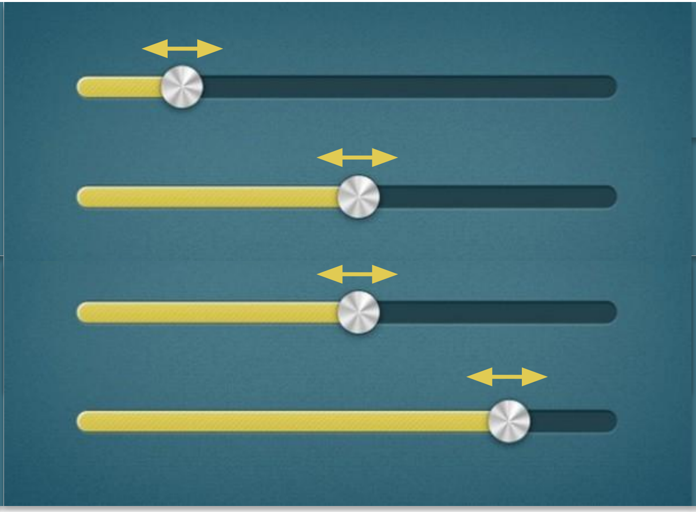
Great for ensemble runs for ***design studies***

Incomplete physics

1D-1V

Low resolution

Reduced models



Full physics

3D-3V

High resolution

First principles

Accurate

Great for detailed runs for ***physics studies***

Goal
Start-to-end model-ing in an open software ecosystem.

BLAST
BEAM PLASMA & ACCELERATOR SIMULATION TOOLKIT

Start-to-End Modeling R&D

- advanced models: numerics, AI/ML surrogates
- speed & scalability: team science with computer sci.
- flexibility & reliability: modern software ecosystem

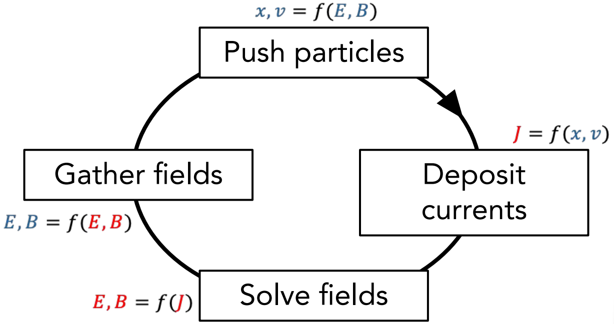
ECP EXASCALE COMPUTING PROJECT

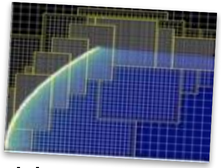
SciDAC Scientific Discovery through Advanced Computing

WarpX is a GPU-Accelerated PIC Code for Exascale

Available Particle-in-Cell Loops

- electrostatic & electromagnetic (fully kinetic)





Advanced algorithms

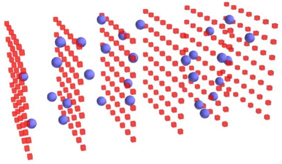
boosted frame, spectral solvers, Galilean frame, embedded boundaries + CAD, MR, …

Multi-Physics Modules

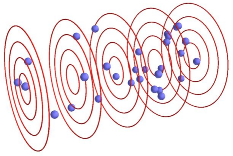
field ionization of atomic levels, Coulomb collisions, QED processes (e.g. pair creation), macroscopic materials

Geometries

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



3D Cartesian grid



Cylindrical grid (schematic)

Multi-Node parallelization

- MPI: 3D domain decomposition
- dynamic load balancing

On-Node Parallelization

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

Scalable, Standardized I/O

- PICMI Python interface
- openPMD (HDF5 or ADIOS)
- in situ diagnostics

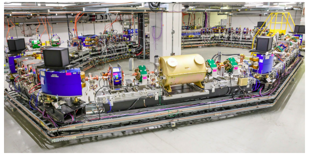
BLAST
BEAM PLASMA & ACCELERATOR SIMULATION TOOLKIT

ECP EXASCALE COMPUTING PROJECT

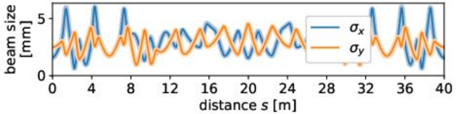
WarpX: conceived & developed by a multidisciplinary, multi-institution team

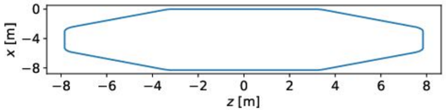
ImpactX: GPU-, AMR- & AI/ML-Accelerated Beam Dynamics

Particle-in-Cell Loop



- electrostatic
  - with space-charge effects
- s-based
  - relative to a reference particle
  - elements: <u>symplectic maps</u>



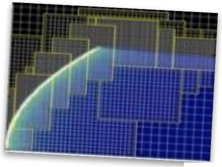


Fireproof Numerics

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

Triple Acceleration Approach

- GPU support
- Adaptive Mesh Refinement
- AI/ML & Data Driven Models



User-Friendly

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

Same Script
CPU/GPU & MPI

Multi-Node parallelization

- MPI: domain decomposition
- dynamic load balancing (in dev.)

On-Node Parallelization

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

Scalable, Parallel I/O

- openPMD
- in situ analysis

mac OS

github.com/**ECP-WarpX/impactx**

BERKELEY LAB LDRD
SciDAC Scientific Discovery through Advanced Computing

ImpactX: Easy to Use, Extent, Tested and Documented

```python
from impactx import ImpactX, elements

sim = ImpactX()
# ...

# design the accelerator lattice)
ns = 25  # number of slices per ds in the element
fodo = [
    elements.Drift(ds=0.25, nslice=ns),
    elements.Quad(ds=1.0, k=1.0, nslice=ns),
    elements.Drift(ds=0.5, nslice=ns),
    elements.Quad(ds=1.0, k=-1.0, nslice=ns),
    elements.Drift(ds=0.25, nslice=ns),
    monitor,
]
# assign a fodo segment
sim.lattice.extend(fodo)

# run simulation
sim.evolve()
```

Example: ImpactX FODO Cell Lattice

INSTALLATION
Users
Developers
HPC

USAGE
Run ImpactX
Parameters: Python
Parameters: Inputs File

⊟ Examples
FODO Cell
Chicane
Constant Focusing Channel
Constant Focusing Channel with Space Charge
Expanding Beam in Free Space
Kurth Distribution in a Periodic Focusing Channel
Kurth Distribution in a Periodic Focusing Channel with Space Charge
Acceleration by RF Cavities
FODO Cell with RF
FODO Cell, Chromatic
Chain of thin multipoles
A nonlinear focusing channel based on the IOTA nonlinear lens
The "bare" linear lattice of the Fermilab IOTA storage ring

/ Examples   Edit on GitHub

Examples

This section allows you to **download input files** that correspond to different physical situations or test different code features.

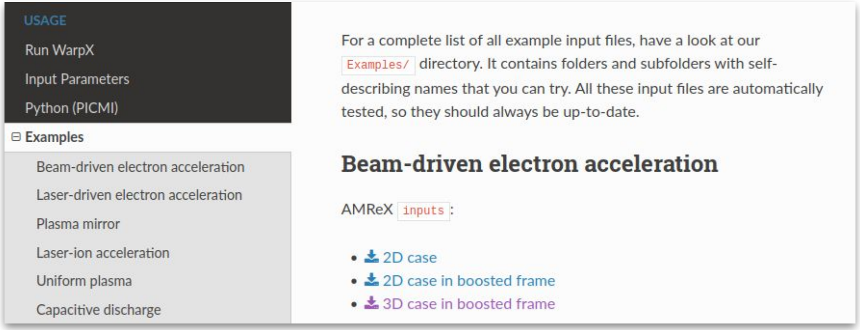
- FODO Cell
- Chicane
- Constant Focusing Channel
- Constant Focusing Channel with Space Charge
- Expanding Beam in Free Space
- Kurth Distribution in a Periodic Focusing Channel
- Kurth Distribution in a Periodic Focusing Channel with Space Charge
- Acceleration by RF Cavities
- FODO Cell with RF
- FODO Cell, Chromatic
- Chain of thin multipoles
- A nonlinear focusing channel based on the IOTA nonlinear lens
- The "bare" linear lattice of the Fermilab IOTA storage ring
- Solenoid channel
- Drift using a Pole-Face Rotation
- Soft-edge solenoid
- Soft-Edge Quadrupole
- Positron Channel
- Cyclotron
- Combined Function Bend
- Ballistic Compression Using a Short RF Element
- Test of a Transverse Kicker

github.com/**ECP-WarpX/impactx**

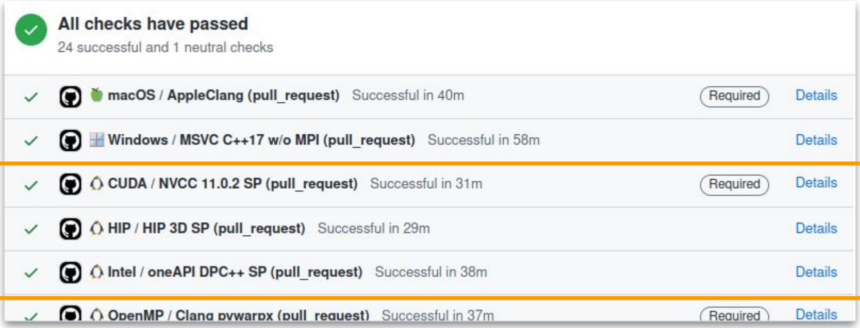
We Develop Openly with the Community

BLAST
BEAM PLASMA & ACCELERATOR SIMULATION TOOLKIT

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



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



230 physics benchmarks *run on every code change* of WarpX
19 physics benchmarks + 106 tests *for* ImpactX

Rapid and easy installation on any platform:

conda install -c conda-forge warpx

spack install warpx
spack install py-warpx

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

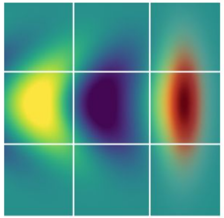
python3 -m pip install .

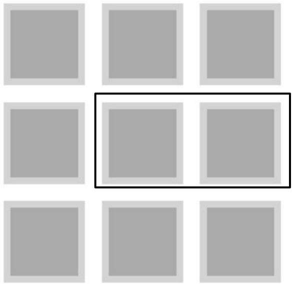
brew tap ecp-warpx/warpx
brew install warpx

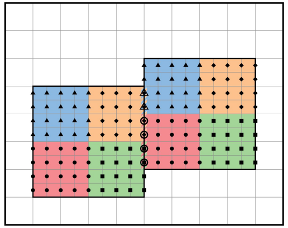
module load warpx
module load py-warpx

Power-Limits Seed a *Cambrian Explosion* of Compute Architectures

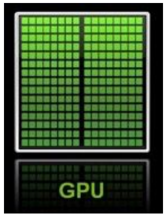
distribute *one* simulation over **10,000s of computers** for **millions of cores**

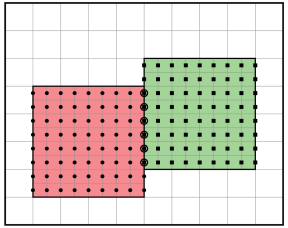






with tiling





without tiling

AMReX

potential future

Field-Programmable Gate Array (FPGA)

Application-Specific Integrated Circuit (ASIC)

Quantum-Circuit

First-of-their-kind platforms: NERSC (Intel, then Nvidia)→Exascale: OLCF (AMD), ALCF (Intel)

ECP EXASCALE COMPUTING PROJECT

J-L Vay, A Huebl et al., PoP 28.2, 023105 (2021); A Myers et al, JParCo 108.102833, (2021); L Fedeli, A Huebl et al., SC22 (2022)

Ranks and Details the 500 most powerful non-distributed Computer Systems, TOP500.org (June 2023)

Community Approaches to Exascale Programming

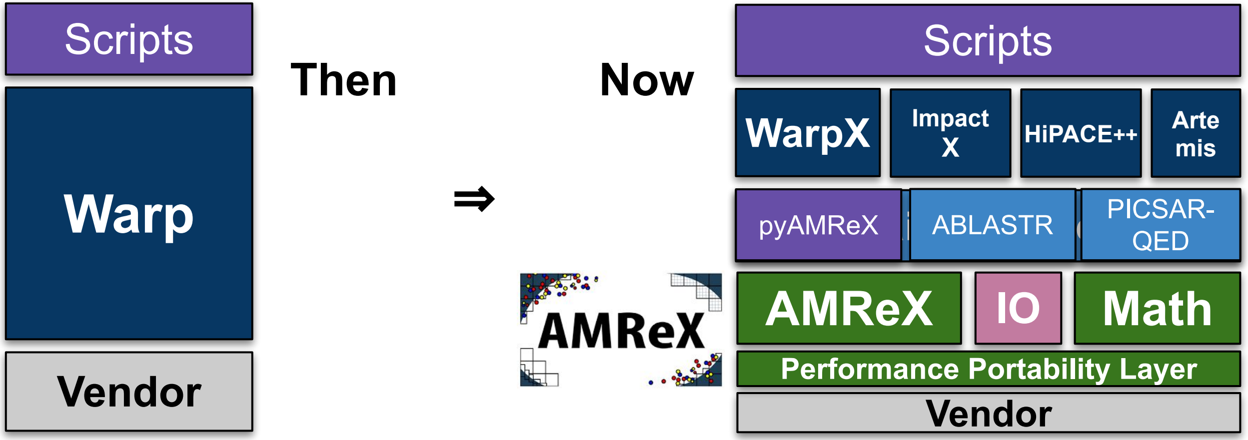
Applications

Libraries

PIC Algorithms

Communication

Performance Portability



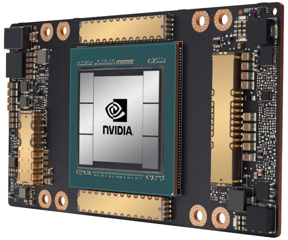
B Worpitz, MA (2015); E Zenker, A Huebl et al., IPDPSW (2016); E Zenker, A Huebl et al., IWOPH (2017); A Matthes, A Huebl et al., P3MA (2017); A Myers et al., JPARCO (2021); HC Edwards et al., SciProg (2012); RD Hornung et al., OSTI TR (2014)

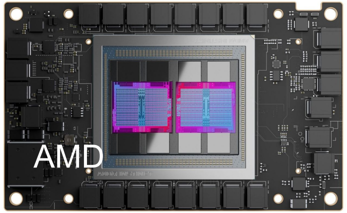
Programming Models

OpenACC Directives for Accelerators | OpenMP | NVIDIA CUDA | AMD ROCm / HIP | SYCL

Hardware

Power

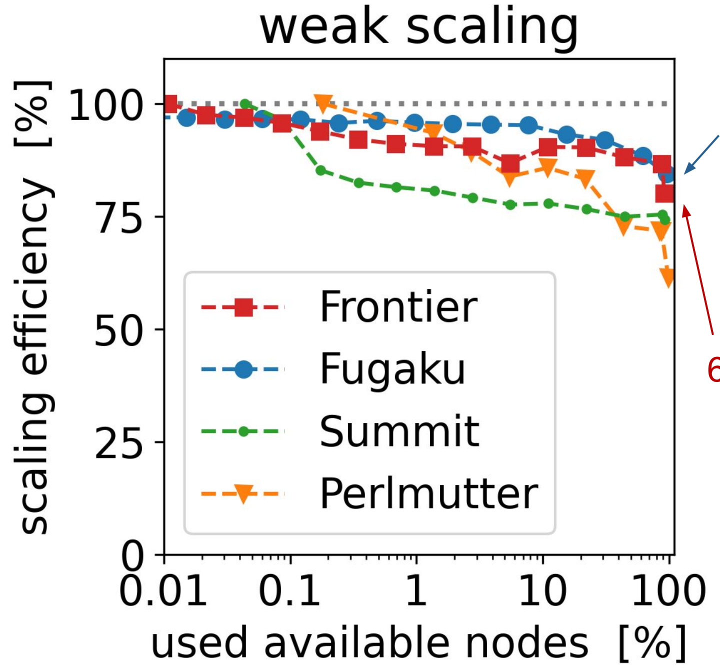




WarpX is now 500x More Performant than its Baseline

April-July 2022: WarpX on **world's largest HPCs**

L. Fedeli, A. Huebl et al., *Gordon Bell Prize Winner* at SC'22, 2022



7,299,072 CPU Cores

68,608 GPUs of *First Exascale* *Machine*

Note: Perlmutter & Frontier were pre-acceptance measurements!

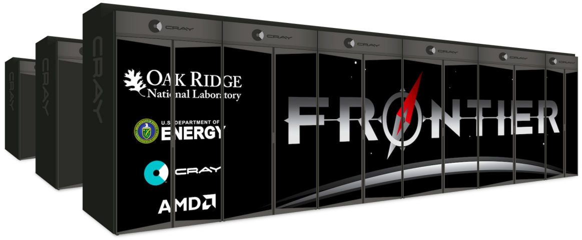


Figure-of-Merit: weighted updates / sec

Date	Code	Machine	N_c/Node	Nodes	FOM
3/19	Warp	Cori	0.4e7	6 625	2.2e10
3/19	WarpX	Cori	0.4e7	6 625	1.0e11
6/19	WarpX	Summit	2.8e7	1 000	7.8e11
9/19	WarpX	Summit	2.3e7	2 560	6.8e11
1/20	WarpX	Summit	2.3e7	2 560	1.0e12
2/20	WarpX	Summit	2.5e7	4 263	1.2e12
6/20	WarpX	Summit	2.0e7	4 263	1.4e12
7/20	WarpX	Summit	2.0e8	4 263	2.5e12
3/21	WarpX	Summit	2.0e8	4 263	2.9e12
6/21	WarpX	Summit	2.0e8	4 263	2.7e12
7/21	WarpX	Perlmutter	2.7e8	960	1.1e12
12/21	WarpX	Summit	2.0e8	4 263	3.3e12
4/22	WarpX	Perlmutter	4.0e8	928	1.0e12
4/22	WarpX	Perlmutter†	4.0e8	928	1.4e12
4/22	WarpX	Summit	2.0e8	4 263	3.4e12
4/22	WarpX	Fugaku†	3.1e6	98 304	8.1e12
6/22	WarpX	Perlmutter	4.4e8	1 088	1.0e12
7/22	WarpX	Fugaku	3.1e6	98 304	2.2e12
7/22	WarpX	Fugaku†	3.1e6	152 064	9.3e12
7/22	WarpX	Frontier	8.1e8	8 576	1.1e13

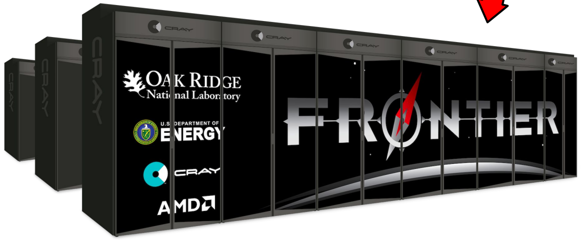
110x

500x

2022 ACM Gordon Bell Prize: using the First Exascale Supercomputer

April-July 2022: WarpX on **world's largest HPCs**

L. Fedeli, A. Huebl et al., *Gordon Bell Prize Winner* at SC'22, 2022



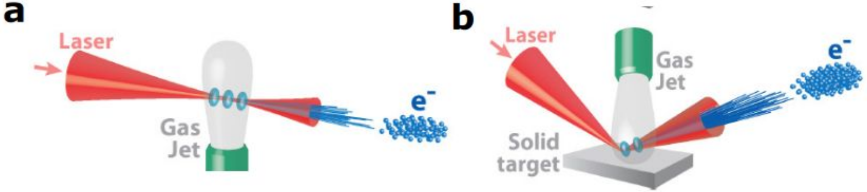


Fig. 1: Sketches showing the focusing of a high-power femtosecond laser (a) into a gas jet (b) onto a hybrid solid-gas target.

cea PARIS-SACLAY · BERKELEY LAB · cnrs · GENCI · RIKEN · arm · Atos

A success story of a multidisciplinary, multi-institutional team!

L. Fedeli, A. Huebl et al., IEEE, SC22 (2022)

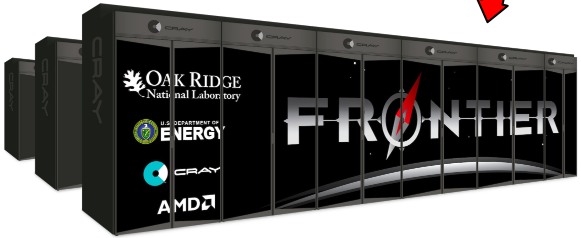
M. Thévenet et al., Nat. Phys 12 (2016)

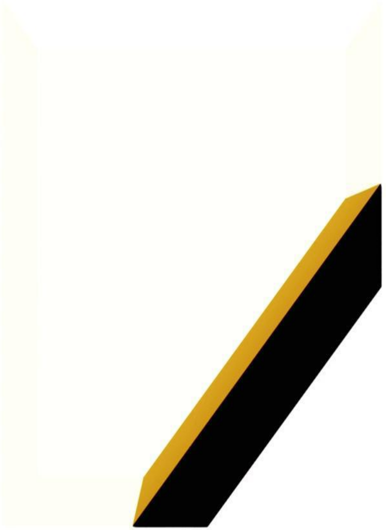
ECP EXASCALE COMPUTING PROJECT

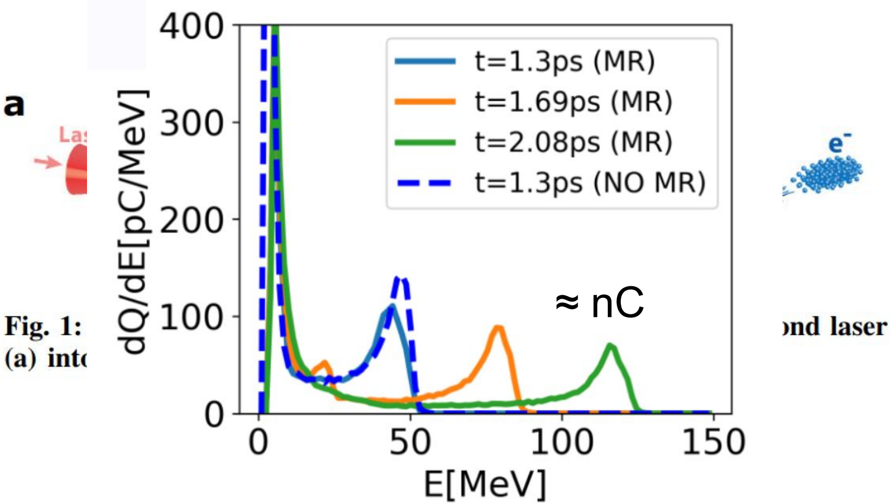
2022 ACM Gordon Bell Prize: using the First Exascale Supercomputer

April-July 2022: WarpX on **world's largest HPCs**

L. Fedeli, A. Huebl et al., *Gordon Bell Prize Winner* at SC'22, 2022







A success story of a multidisciplinary, multi-institutional team!

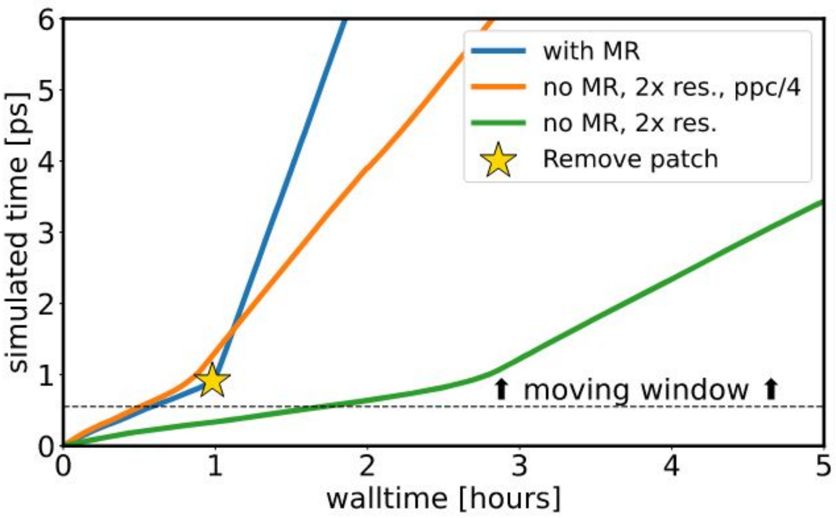
L. Fedeli, A. Huebl et al., IEEE, SC22 (2022)

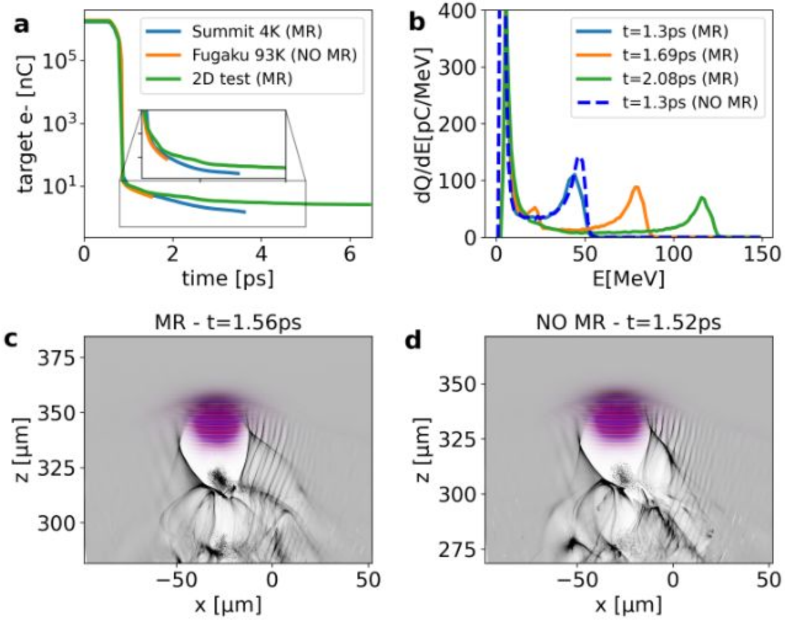
M. Thévenet et al., Nat. Phys 12 (2016)

2022 ACM Gordon Bell Prize: using the First Exascale Supercomputer

April-July 2022: WarpX on **world's largest HPCs**

L. Fedeli, A. Huebl et al., *Gordon Bell Prize Winner* at SC'22, 2022





A success story of a multidisciplinary, multi-institutional team!

L. Fedeli, A. Huebl et al., IEEE, SC22 (2022)

M. Thévenet et al., Nat. Phys 12 (2016)

If You Want to Go Far, Go Together

Standardization…

- Inputs
- Data
- Reference Implementations

strong int. partnerships

Particle-In-Cell Modeling Interface

PICMI Standard

↓

Code A

Code B

…

↓

open Particle Mesh Data standard

openPMD

… Accelerates Innovation

- *LASY*
  github.com/**LASY-org**
- optimas
  github.com/**optimas-org**
- BLAST + Geant4
  github.com/LDAmorim/**GPos**
- **easy ML training**

open source initiative®

SciDAC-5

LDRD

A Huebl et al., DOI:10.5281/zenodo.591699 (2015)
DP Grote et al., *Particle-In-Cell Modeling Interface (PICMI)* (2021)
LD Amorim et al., *GPos* (2021); M Thévenet et al., DOI:10.5281/zenodo.8277220 (2023)
A Ferran Pousa et al., DOI:10.5281/zenodo.7989119 (2023)
RT Sandberg et al., IPAC23, DOI:10.18429/JACoW-IPAC-23-WEPA101 (2023)

Across Scales: Advanced and Conventional Accelerators

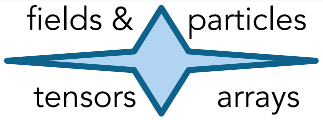
BLAST is Now An Accelerated, Machine-Learning Boosted Ecosystem

BLAST
BEAM PLASMA & ACCELERATOR SIMULATION TOOLKIT

GPU Workflows are blazingly fast

- PIC simulations (EXASCALE COMPUTING PROJECT)
- Machine learning (PyTorch)

Can we augment & accelerate on-GPU PIC simulations with on-GPU ML models?



Numba

CuPy

Cross-Ecosystem, In Situ Coupling

Consortium for Python Data API Standards data-apis.org

Very easy to:

- connect
- vary ML models

```python
from pywarpx import picmi
import torch
# ...

# iterate all density boxes
for i in rho_device:
    rho = torch.as_tensor(
        rho_device.array(i),
        device="cuda")

    # apply ML in-memory
    with torch.no_grad():
        surrogate_model(rho)
```

A) Training

- Offline: WarpX openPMD → Neural Network
- Online (*in situ*): advanced ML methods

B) Inference: *in situ* to codes

- Zero-copy data access: *persistently on GPU*
- Example: an *ML map* in beam dynamics

A Huebl et al., NAPAC22, DOI:10.18429/JACoW-NAPAC2022-TUYE2 (2022)
RT Sandberg et al and A Huebl, IPAC23, DOI:10.18429/JACoW-IPAC-23-WEPA101 (2023)
A Huebl et al., AAC22, arXiv:2303.12873 (2023); RT Sandberg et al. and A Huebl, *in preparation* (2023)

BERKELEY LAB LDRD

A Huebl (PI), R Sandberg, R Lehe, CE Mitchell et al.

Modeling Time: ML-Acceleration of Plasma Elements for Beamlines

LPA integration via AI/ML for rapid beamline design & operations.

Model Speed: for accelerator elements

Plasma Source	Trans-port	Plasma Stage	Plasma Stage	Injector
WarpX	ImpactX	WarpX	HiPACE++	WarpX-ES

Simulation time: full geometry, full physics

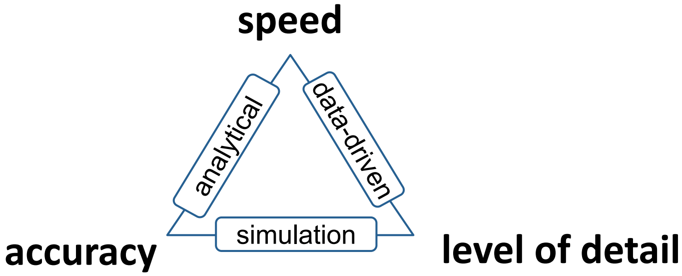
hrs	sec	hrs	hrs	min

ML boosted: for a *specific* problem

LWFA w/ iinj.	Trans-port	LWFA Stage	PWFA Stage	Kicker Magnet
ML	ImpactX	ML	ML	ML

- start-to-end collider modeling
- digital twin / ‘real-time’

Model Choice: for complex, nonlinear, many-body systems ***pick two*** of the following



Fast surrogates: Data-driven modeling is a potential middle ground between

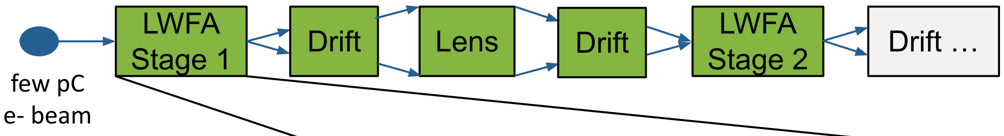
- analytical modeling and
- full-fidelity simulations.

BERKELEY LAB LDRD
A Huebl (PI), R Sandberg, R Lehe, CE Mitchell et al.

A Huebl et al., NAPAC22, DOI:10.18429/JACoW-NAPAC2022-TUYE2 (2022)
RT Sandberg et al and A Huebl, IPAC23, DOI:10.18429/JACoW-IPAC-23-WEPA101 (2023)
A Huebl et al., AAC22, arXiv:2303.12873 (2023); RT Sandberg et al. and A Huebl, *in preparation* (2023)

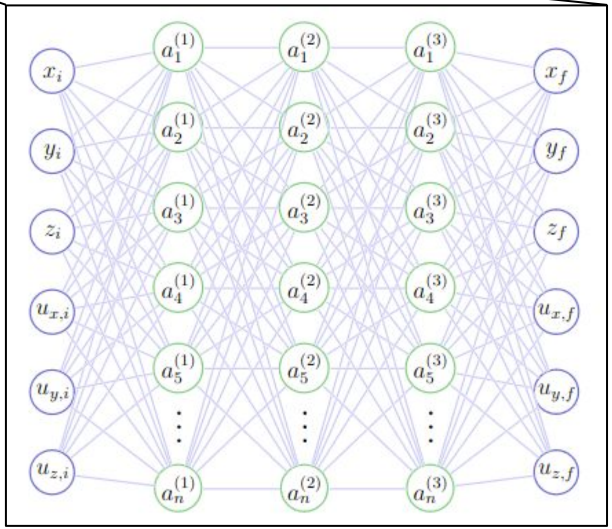
We Trained a Neural Net with WarpX for Staging of Electrons

one-time cost: few hr WarpX sim + 10min training



A Neural Net is a non-linear transfer map!

Assumption: purely tracking



A **single NN** can learn details of **multiple stages** (e.g, 10, 20, 30 GeV).

Assumption: laser-plasma parameters stay the same.

Hyperparameters

- 6D in 6D out
- <10 layers with few 100s of nodes each are sufficient

Error of Beam Moments

	combined beamline error	combined beamline relative error	stage 1 relative error	stage 2 relative error
$\langle x \rangle$	-2.015e-08	-6.337e-02	5.179e-02	-3.916e-02
σ_x	2.723e-09	8.565e-03	-4.381e-03	4.288e-03
$\langle u_x \rangle$	-3.319e-01	-9.887e-02	-8.609e-02	2.814e-02
σ_{ux}	1.710e-02	5.094e-03	1.047e-02	7.716e-03
ϵ_x	1.844e-08	1.747e-02	7.740e-03	9.912e-03
$\langle y \rangle$	-6.882e-10	-2.155e-03	5.228e-02	1.585e-02
σ_y	9.245e-09	2.895e-02	-8.687e-04	6.412e-03
$\langle u_y \rangle$	-4.540e-01	-1.328e-01	-1.089e-02	-1.243e-01
σ_{uy}	9.856e-02	2.884e-02	3.411e-02	2.491e-03
ϵ_y	5.932e-08	5.509e-02	3.334e-02	5.899e-03
$\langle z \rangle$	-7.686e-09	-7.506e-02	-9.746e-04	-2.561e-02
σ_z	-1.900e-11	-1.855e-04	-3.943e-04	2.927e-03
$\langle u_z \rangle$	1.797e+00	6.148e-05	4.151e-04	-3.769e-05
σ_{uz}	-1.088e+01	-8.394e-02	-8.186e-02	-3.944e-02

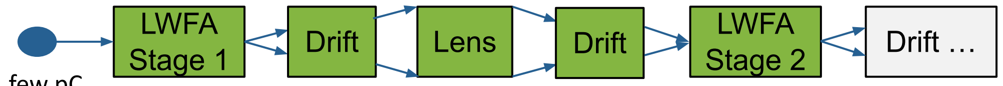
Training data: 50,000 particles / beam

A Huebl et al., NAPAC22, DOI:10.18429/JACoW-NAPAC2022-TUYE2 (2022)
RT Sandberg et al and A Huebl, IPAC23, DOI:10.18429/JACoW-IPAC-23-WEPA101 (2023)
A Huebl et al., AAC22, arXiv:2303.12873 (2023); RT Sandberg et al. and A Huebl, *in preparation* (2023)

BERKELEY LAB LDRD A Huebl (PI), R Sandberg, R Lehe, CE Mitchell et al.

We Trained a Neural Net with WarpX for Staging of Electrons

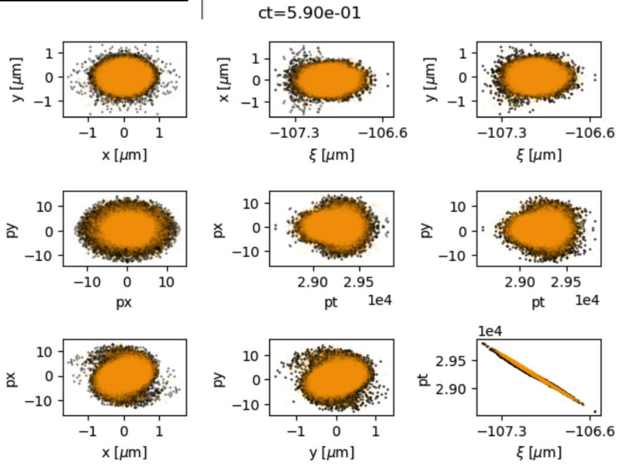
one-time cost: few hr WarpX sim + 10min training



few pC e- beam

ImpactX simulation time: <1 sec

ImpactX: after 2 surrogates
WarpX: 2 stage simulation



Flexible, Hybrid Beamline Sim

- any 6D beam input
- tune lens, transport, …
- modify ML models

Same super-fast evaluation!

Open challenges

Learning microscopic *and* collective effects simultaneously.

Error of Beam Moments

	combined beamline error	combined beamline relative error	stage 1 relative error	stage 2 relative error
$\langle x \rangle$	-2.015e-08	-6.337e-02	5.179e-02	-3.916e-02
σ_x	2.723e-09	8.565e-03	-4.381e-03	4.288e-03
$\langle u_x \rangle$	-3.319e-01	-9.887e-02	-8.609e-02	2.814e-02
σ_{ux}	1.710e-02	5.094e-03	1.047e-02	7.716e-03
ϵ_x	1.844e-08	1.747e-02	7.740e-03	9.912e-03
$\langle y \rangle$	-6.882e-10	-2.155e-03	5.228e-02	1.585e-02
σ_y	9.245e-09	2.895e-02	-8.687e-04	6.412e-03
$\langle u_y \rangle$	-4.540e-01	-1.328e-01	-1.089e-02	-1.243e-01
σ_{uy}	9.856e-02	2.884e-02	3.411e-02	2.491e-03
ϵ_y	5.932e-08	5.509e-02	3.334e-02	5.899e-03
$\langle z \rangle$	-7.686e-09	-7.506e-02	-9.746e-04	-2.561e-02
σ_z	-1.900e-11	-1.855e-04	-3.943e-04	2.927e-03
$\langle u_z \rangle$	1.797e+00	6.148e-05	4.151e-04	-3.769e-05
σ_{uz}	-1.088e+01	-8.394e-02	-8.186e-02	-3.944e-02

Training data: 50,000 particles / beam

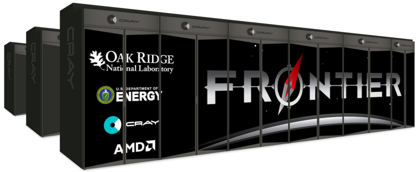
A Huebl et al., NAPAC22, DOI:10.18429/JACoW-NAPAC2022-TUYE2 (2022)
RT Sandberg et al and A Huebl, IPAC23, DOI:10.18429/JACoW-IPAC-23-WEPA101 (2023)
A Huebl et al., AAC22, arXiv:2303.12873 (2023); RT Sandberg et al. and A Huebl, *in preparation* (2023)

BERKELEY LAB LDRD

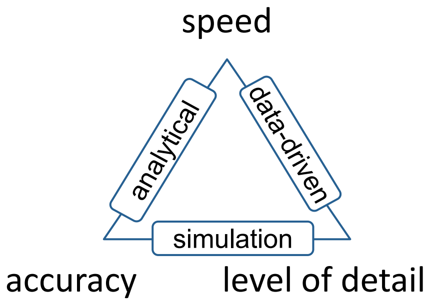
A Huebl (PI), R Sandberg, R Lehe, CE Mitchell et al.

Summary

presented by: Axel Huebl (LBNL)
axelhuebl@lbl.gov



- **BLAST** is a fully open suite of PIC codes for **particle accelerator modeling**, using code-sharing through libraries and leverage the U.S. DOE Exascale software stack.
  - **WarpX** was our first **Exascale** app, for relativistic t-based laser-plasma & beam modeling
  - **ImpactX** leverages these developments for s-based beam dynamics.



- Seamless, **GPU-Accelerated Combination** of **PIC and AI/ML**
  - **zero-copy GPU data access**: in situ models, application coupling
  - Scripted: easy to **vary & research** new data models

- **Vibrant Ecosystem and Contributions**
  - Runs on any platform: Linux, macOS, Windows
  - Public development, automated testing, review & documentation
  - Friendly, open & helpful community

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

open source initiative®

EXASCALE COMPUTING PROJECT
SciDAC Scientific Discovery through Advanced Computing
BERKELEY LAB LDRD

Backup Slides

Abstract (16'+4')

Computational modeling is essential to the exploration and design of advanced particle accelerators. The modeling of laser-plasma acceleration and interaction can achieve predictive quality for experiments if adequate resolution, full geometry and physical effects are included.

Here, we report on the significant evolution in fully relativistic full-3D modeling of conventional and advanced accelerators in the WarpX and ImpactX codes with the introduction of Exascale supercomputing and AI/ML models. We will cover the first PIC simulations on an Exascale machine, the need for and evolution of open standards, and based on our fully open community codes, the connection of time and space scales from plasma to conventional beamlines with data-driven machine-learning models.

WarpX in ECP: Staging of Laser-Driven Plasma Acceleration

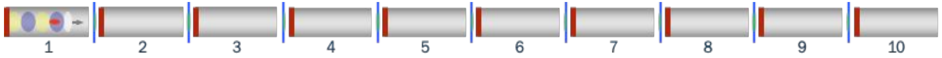
U.S. DEPARTMENT OF ENERGY | Office of Science
NNSA National Nuclear Security Administration

Goal: deliver & scientifically use the nation's first exascale systems

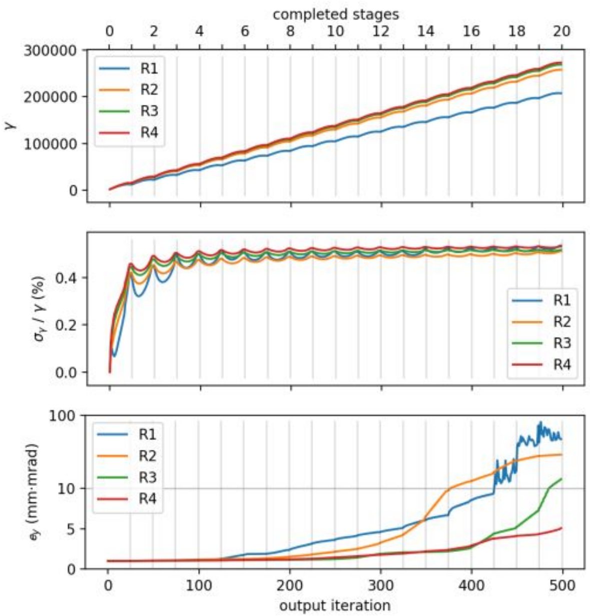
- **ExaFLOP:** a quintillion (10^{18}) calculations per second
- ensure ***all*** the necessary pieces are *concurrently* in place

Our DOE science case is in **HEP**, our methods are **ASCR**:

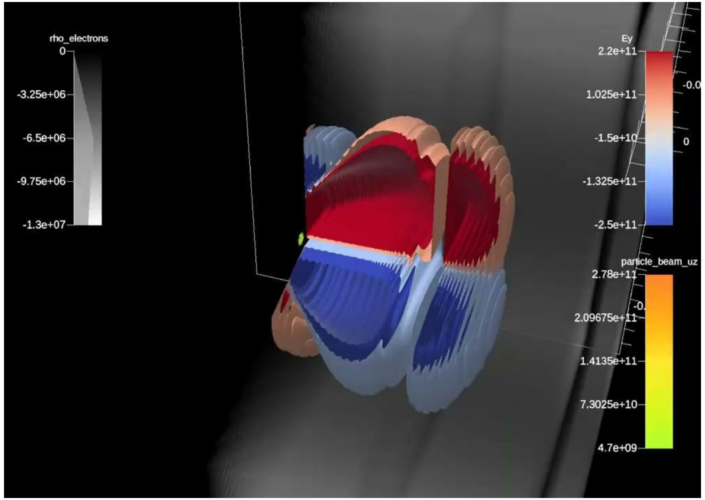
first 3D simulation of a chain of plasma
accelerator stages for future colliders



WarpX in ECP: Staging of Laser-Driven Plasma Acceleration



Ascent VTK-m *In Situ* Visualization:
N. Marsaglia, C. Harrison, A. Huebl



First-of-their-kind platforms: NERSC (Intel, then Nvidia)→Exascale: OLCF (AMD), ALCF (Intel)

J.-L. Vay, A. Huebl et al., ISAV'20 Workshop Keynote (2020) and PoP 28.2, 023105 (2021); L. Fedeli, A. Huebl et al., SC22 (2022)
J.-L. Vay et al., ECP WarpX MS FY23.1; A. Ferran Pousa et al., IPAC23, DOI:10.18429/JACoW-IPAC-23-TUPA093 (2023)

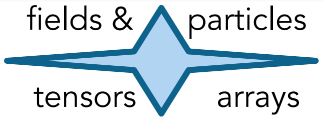
BLAST is Now An Accelerated, ML-Modeling Ecosystem

All-GPU Workflows are blazingly fast

- PIC simulations
- ML Models

Can we augment & accelerate on-GPU PIC simulations with on-GPU ML models?

BLAST
BEAM PLASMA & ACCELERATOR SIMULATION TOOLKIT



Cross-Ecosystem, In Situ Coupling

Consortium for Python Data API Standards data-apis.org

Very easy to:

- connect
- vary to other models

A) Training

- Offline: WarpX openPMD → Neural Network
- Online (*in situ*): advanced ML methods

B) Inference: *in situ* to codes

- Zero-copy data access: *<u>persistently on GPU</u>*
- Example: an *ML map* in beam dynamics

Related Works: Not or only partly GPU accelerated

- bottlenecks in host-device I/O, slower
- quality of prediction C Badiali et al., JPlasmaPhys. 88.6 (2022)

A Huebl et al., NAPAC22, DOI:10.18429/JACoW-NAPAC2022-TUYE2 (2022)
RT Sandberg et al and A Huebl, IPAC23, DOI:10.18429/JACoW-IPAC-23-WEPA101 (2023)
A Huebl et al., AAC22, arXiv:2303.12873 (2023); RT Sandberg et al. and A Huebl, *in preparation* (2023)

LDRD A Huebl (PI), R Sandberg, R Lehe, CE Mitchell et al.

The WarpX Software Stack

BLAST
BEAM PLASMA & ACCELERATOR SIMULATION TOOLKIT

Python: Modules, PICMI interface, Workflows

WarpX full PIC, LPA/LPI

HiPACE++ quasi-static, PWFA

ARTEMIS microelectronics

ImpactX accelerator lattice design

Object-Level Python Bindings extensible, AI/ML

pyAMReX

PICSAR QED Modules

ABLASTR library: common PIC physics

AMReX

Containers, Communication, Portability, Utilities

Diagnostics I/O code coupling

openPMD

ADIOS2

HDF5

ZFP

...

Ascent

VTK-m

FFT on- or multi-device

Lin. Alg. BLAS++ LAPACK++

MPI

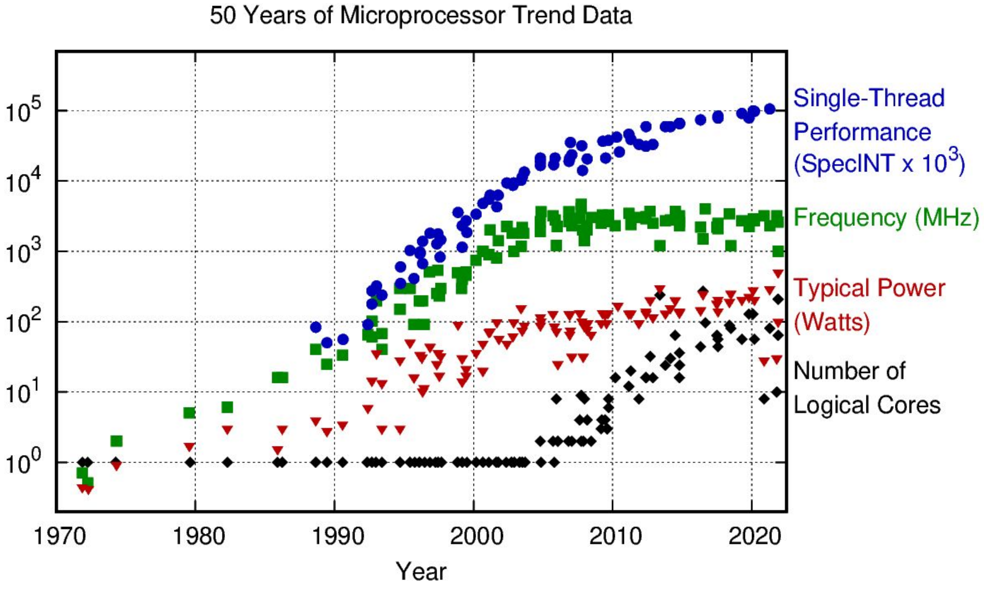
CUDA, OpenMP, SYCL, HIP

mac OS

Desktop to HPC

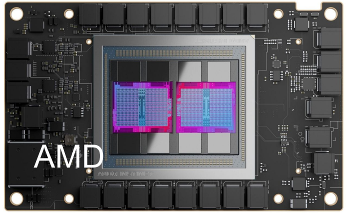
ECP

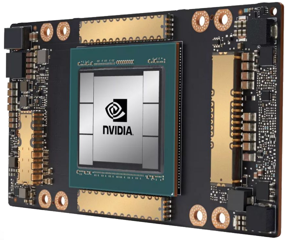
Power-Limits Seed a *Cambrian Explosion* of Compute Architectures



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2021 by K. Rupp

Power



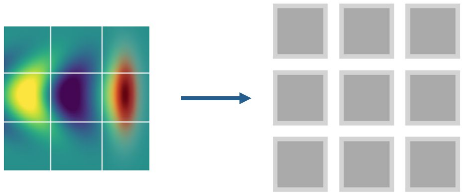


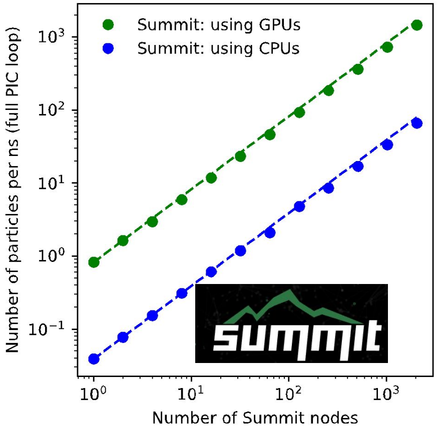
EXASCALE COMPUTING PROJECT

Portable Performance through Exascale Programming Model

AMReX library

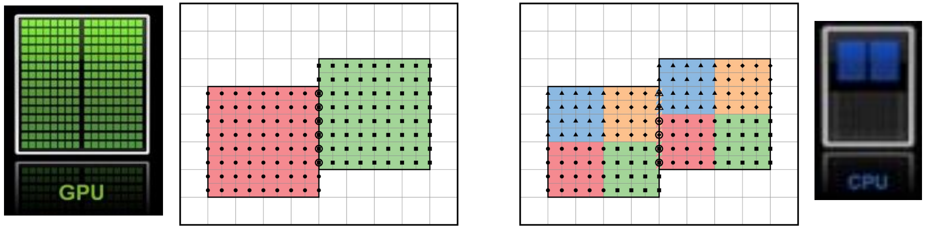
- Domain decomposition & MPI **communications:** MR & load balance





A100 gives additional ~< 2x

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



without tiling

with tiling

Data Structures

- **Write the code once**, specialize at ***compile-time***

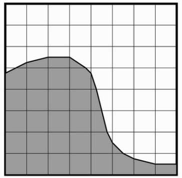
`ParallelFor(/Scan/Reduce)`

```
amrex::ParallelFor( n_particles,
    [=] AMREX_GPU_DEVICE (long i) {

        UpdatePosition( x[i], y[i], z[i],
                ux[i], uy[i], uz[i], dt );

    });
```

- **Parallel linear solvers** (e.g. multi-grid Poisson solvers)
- **Embedded boundaries**

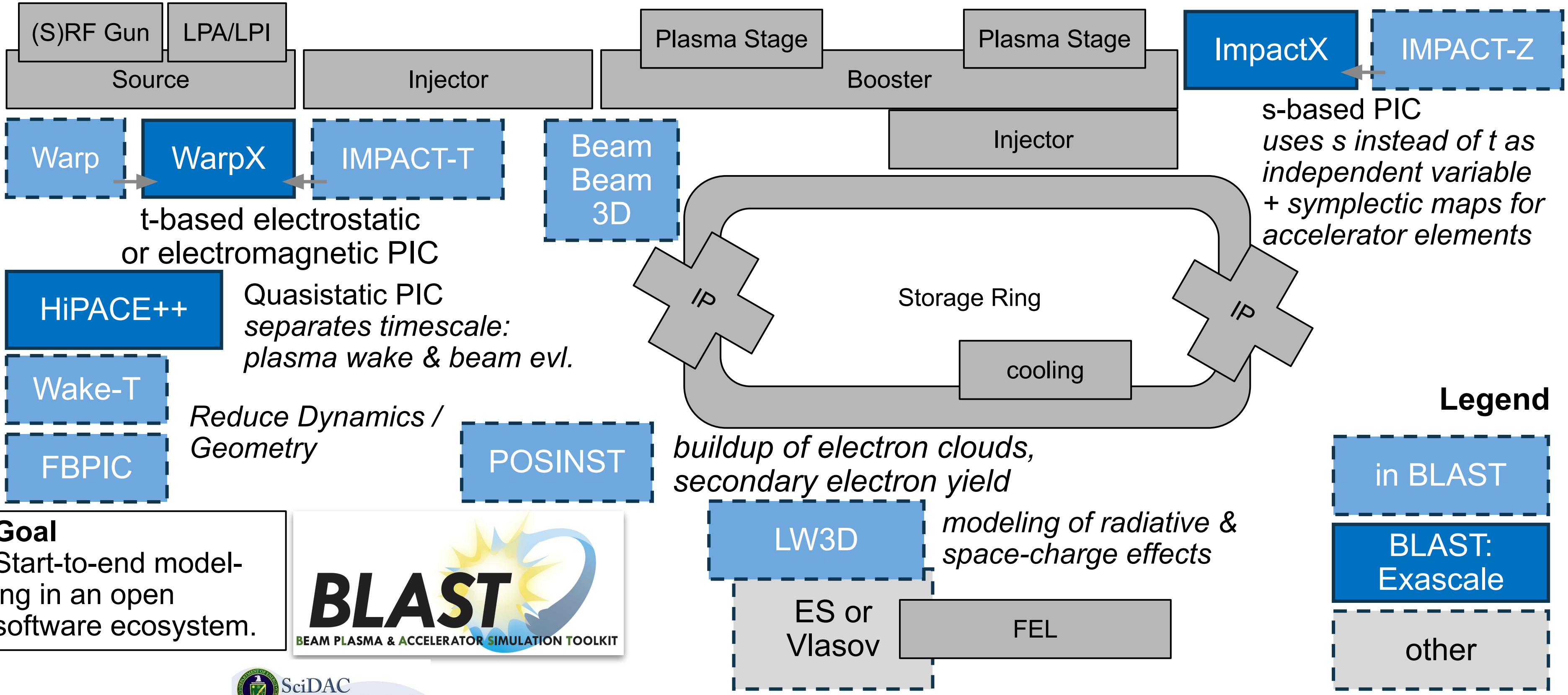


- Runtime parser for user-provided math expressions (incl. GPU)

A. Myers et al., "Porting WarpX to GPU-accelerated platforms," Parallel Computing 108, 102833 (2021)

BLAST Codes: Transition to Exascale

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



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

BLAST
BEAM PLASMA & ACCELERATOR SIMULATION TOOLKIT

SciDAC
Scientific Discovery through Advanced Computing