GPU-Powered Particle-in-Cell Community Frameworks for Laser-Plasma Interaction

DOI:10.5281/zenodo.3900296

Axel Huebl

Lawrence Berkeley National Laboratory, U.S.

WarpX Collaboration, Lawrence Berkeley National Laboratory, U.S.
previously with: PIConGPU Collaboration,
Helmholtz-Zentrum Dresden-Rossendorf, Germany

SIAM-PP: GPU Computing for Solving Large Scale Scientific Problems

February 14th, 2020

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

ACCELERATOR TECHNOLOGY & APPLIED PHYSICS DIVISION ATAP

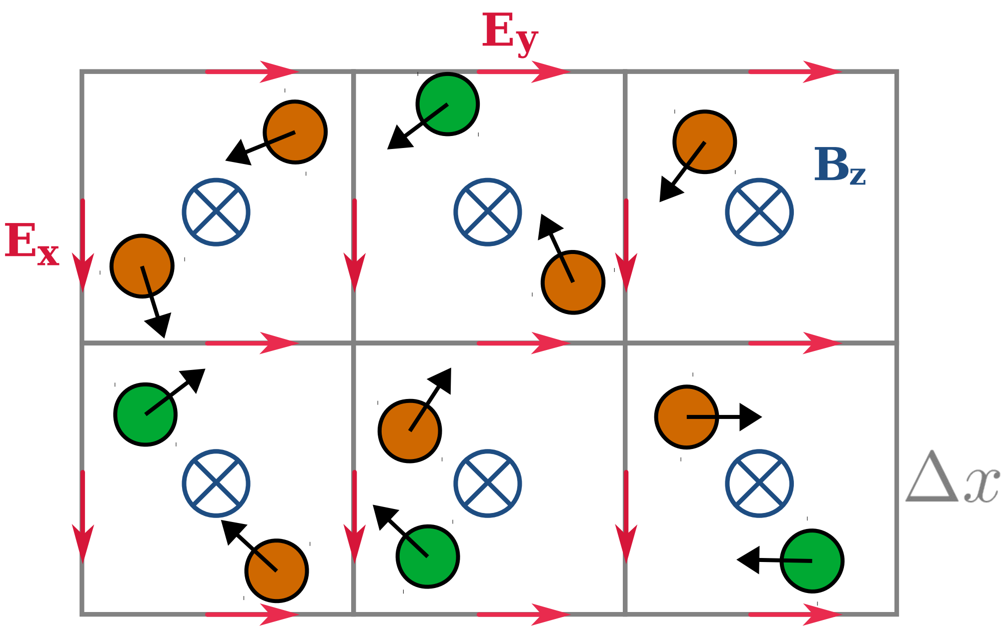
ECP EXASCALE COMPUTING PROJECT

Electromagnetic Particle-in-Cell on GPU

GPU-centric Data Challenges

- **Algorithm**
- **Open Exascale Software Stacks (Examples):**
  - WarpX
  - PIConGPU
- **Application Memory Footprint**
  - Motivation
  - Implementation choices
  - Code Comparison: Optimization vs. flexibility
  - Mixed Precision Benchmarks

EM Particle-in-Cell
Basic Principle



- Eulerian: electro-magnetic fields
- Lagrangian: particles in Vlasov-equation

initial & boundary conditions:

$$\nabla \cdot \mathbf{E} = \frac{1}{\varepsilon_0} \sum_s \rho_s$$

$$\nabla \cdot \mathbf{B} = 0$$

self-consistent, linearized time step:

$$\frac{\partial \mathbf{A}}{\partial t} \rightarrow \frac{\Delta \mathbf{A}}{\Delta t}$$

$$c\Delta t \lessapprox \Delta x$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{j} + \mu_0 \epsilon_0 \frac{\partial \mathbf{E}}{\partial t}$$

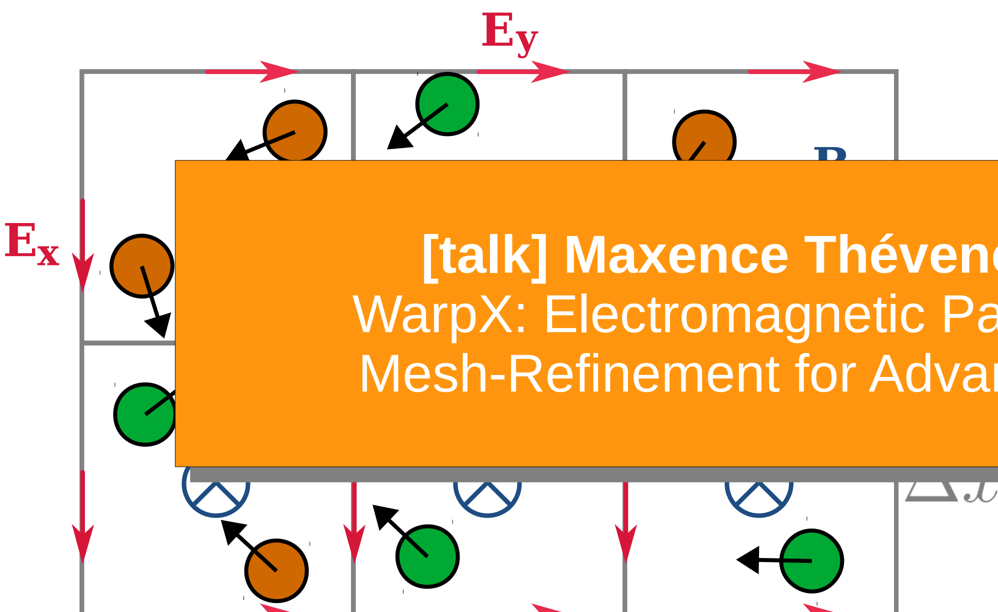
EM Particle-in-Cell

Basic Principle

initial & boundary conditions:

$$\nabla \cdot \mathbf{E} = \frac{1}{\varepsilon_0} \sum_s \rho_s$$

$$\nabla \cdot \mathbf{B} = 0$$



[talk] Maxence Thévenet – Sat 1:50PM in 507
WarpX: Electromagnetic Particle-in-Cell with Adaptive Mesh-Refinement for Advanced Particl eAccelerators

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$

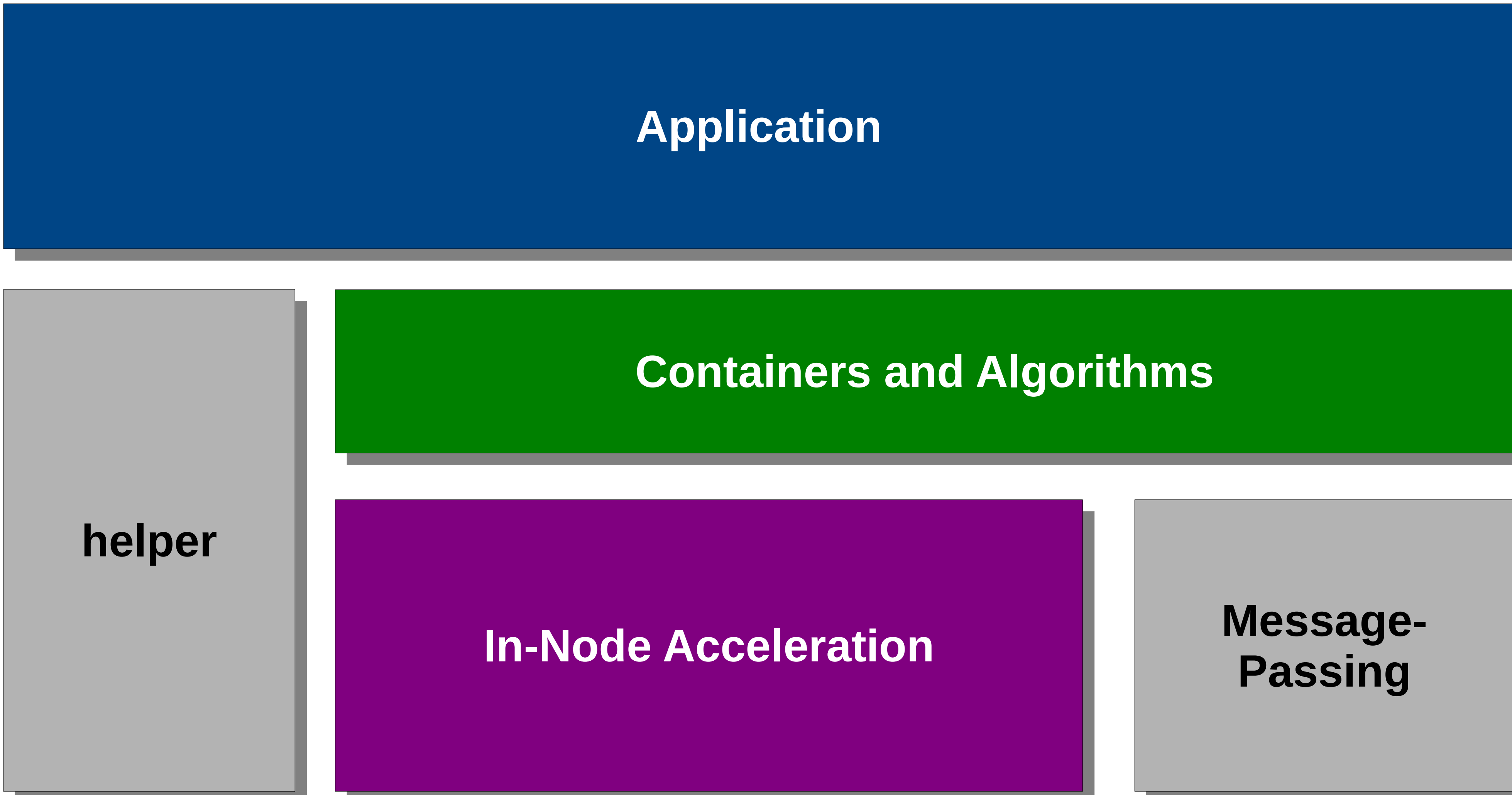
$$\nabla \times \mathbf{B} = \mu_0 \mathbf{j} + \mu_0 \epsilon_0 \frac{\partial \mathbf{E}}{\partial t}$$

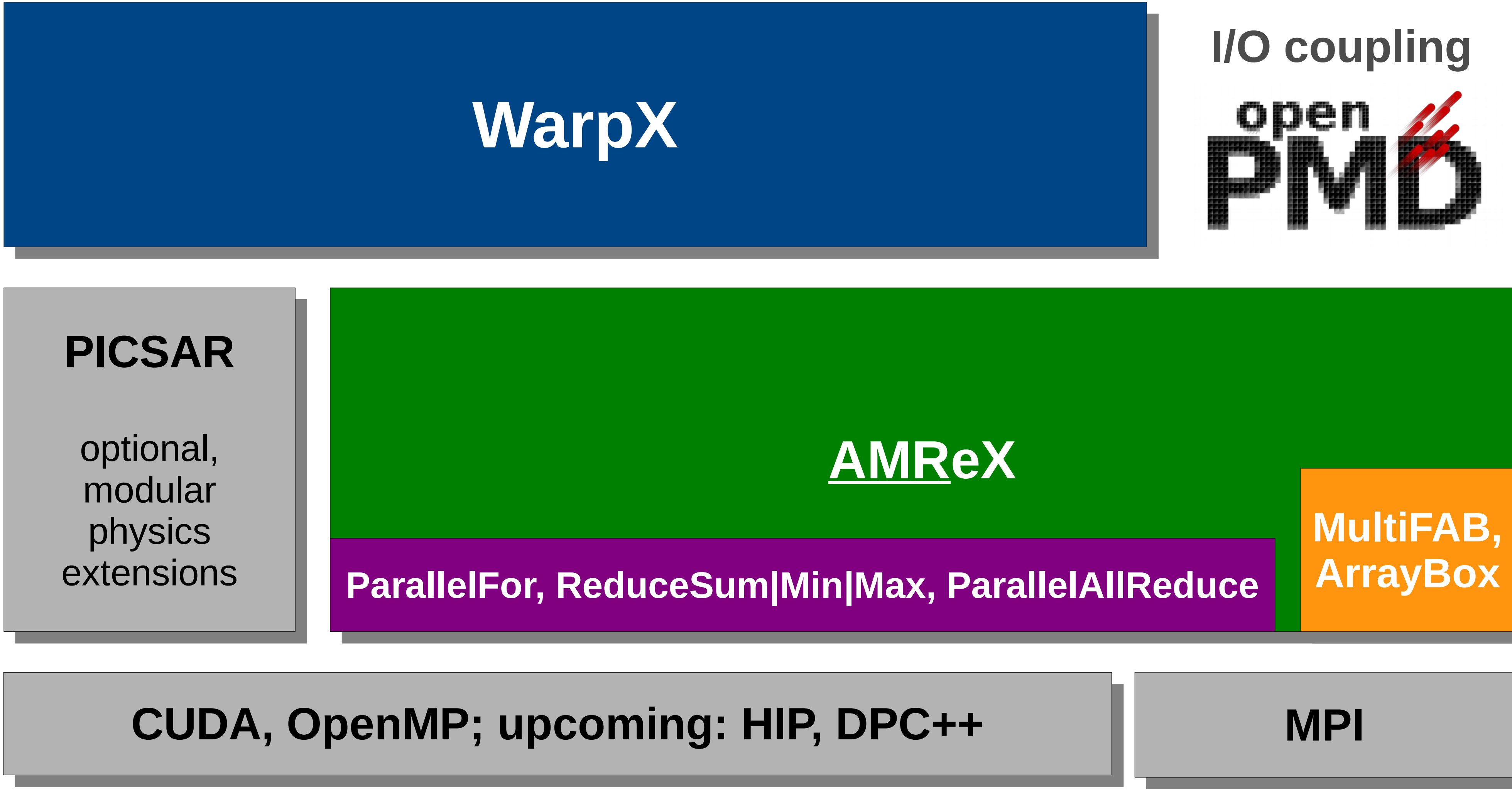
- Eulerian: electro-magnetic fields
- Lagrangian: particles in Vlasov-equation

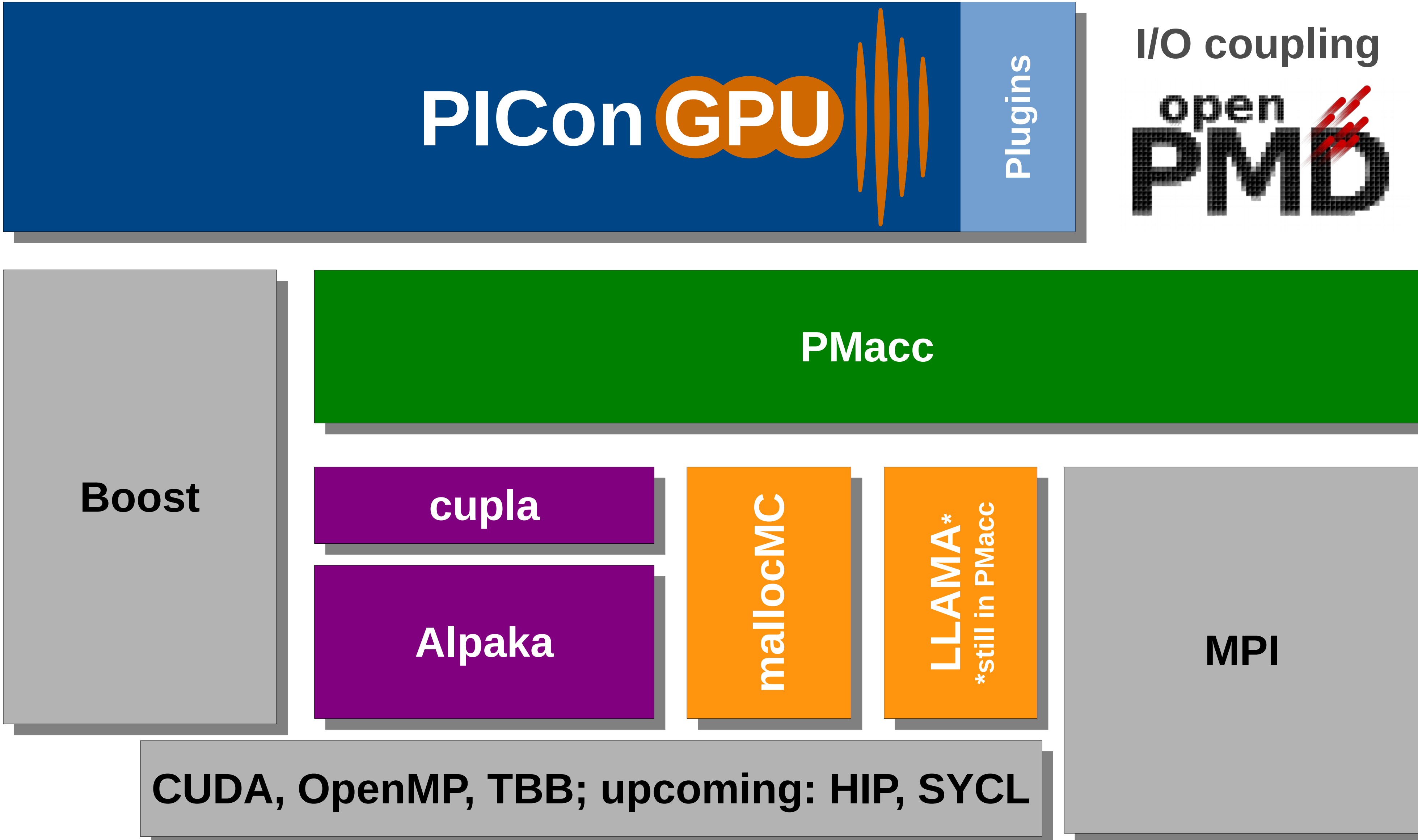
Exascale PIC
Software Stacks

Examples: WarpX & PIConGPU

HPC Application Software Stack




WarpX
I/O coupling
openPMD
PICSAR
optional,
modular
physics
extensions
AMReX
ParallelFor, ReduceSum|Min|Max, ParallelAllReduce
MultiFAB,
ArrayBox
CUDA, OpenMP; upcoming: HIP, DPC++
MPI


PIConGPU
Plugins
I/O coupling
openPMD
PMacc
Boost
cupla
Alpaka
mallocMC
LLAMA*
*still in PMacc
MPI
CUDA, OpenMP, TBB; upcoming: HIP, SYCL

Application Memory Footprint

GPU Memory Footprint

Motivation

Titan (ORNL): 109 TByte GPU RAM

- **GPU-specific Challenge**
  - Device utilization: data **persistently on device**
  - GPU weak-scaling: to solve memory**-size** bound setups
  - **Memory utilization** peaks: move particles, buffer communications
- **Resource Occupation**
  - Scalability: essential; methods (and codes) scale well
  - Time-to-solution: from week(s) to half-days due to GPUs
  - **Node-hours-per-run: linear to resulting science / campaign**

Particle Implementation Choices

Memory Layout and Management

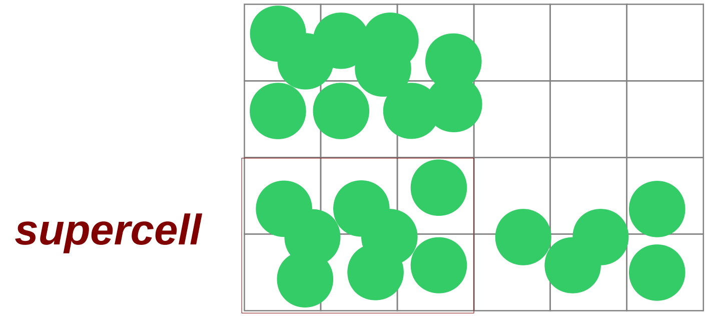
[talk] Andrew Myers – 3:45PM in 507
An Overview of Particles in AMReX

- **SoA**/AoS
  - unsorted or sorted (e.g. for particle-particle)
  - cache blocking: sort + index; often hindered due to multiple kernels
  - resize: costly or pre-partitioning

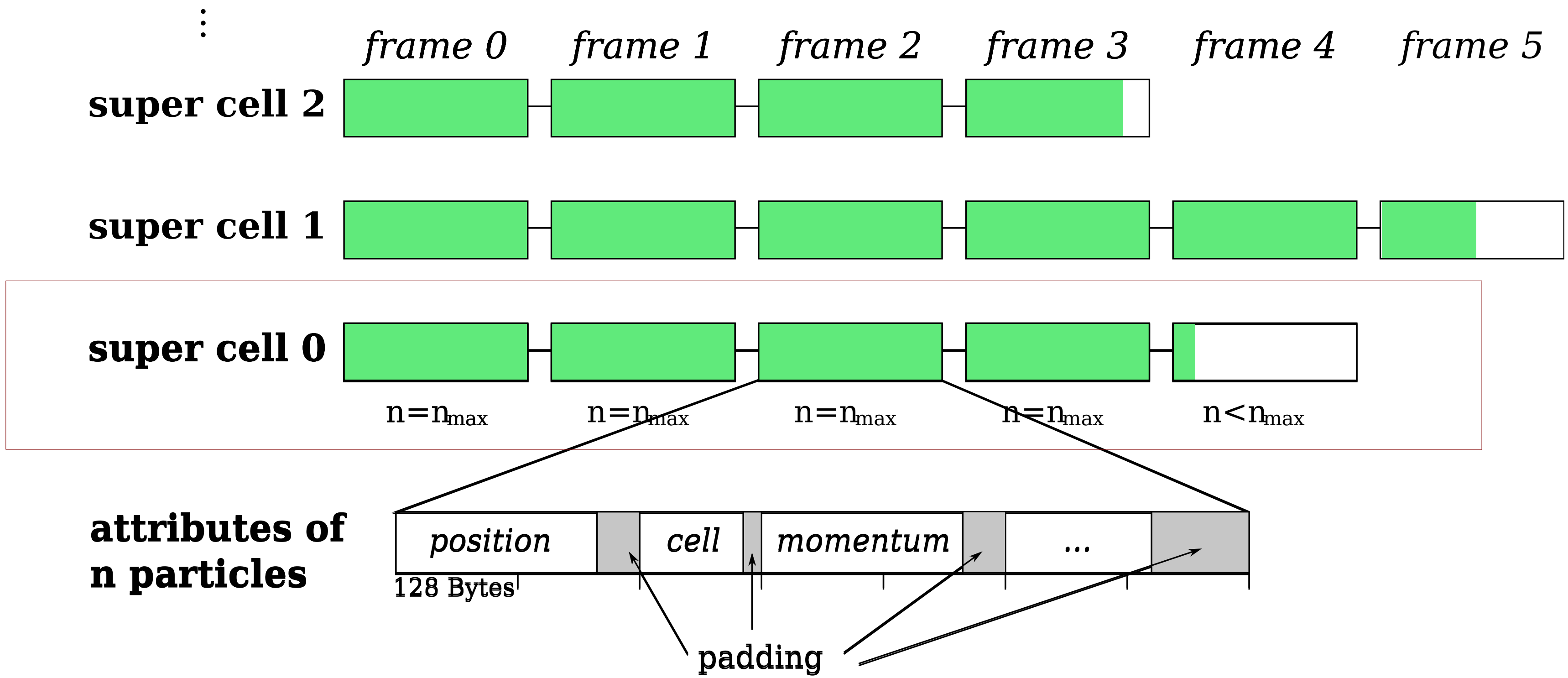
- **Tiled SoA**/AoS
  - unsorted, sorted or bucket sorted (in-bucket index/sort for particle-particle)
  - cache blocking: iterate tiles of spatially close particles by bucket (supercell)
  - resize: flexible; add chunks in lock-free algorithms

PIConGPU

Data Structures



$\vec{E}, \vec{B}$



H. Burau et al., IEEE Trans. Plasma Sci. (2010), DOI:10.1109/TPS.2010.2064310
M. Bussmann et al., SC'13 (2013), DOI:10.1145/2503210.2504564
A. Huebl, Diploma Thesis (2014), DOI:10.5281/zenodo.15924

Memory Footprint Comparison

Warm, Uniform, 3D3V Electron-Positron Plasma

Particles per GPU: 16GByte V100 (12.6 Mio cells)

- **WarpX** 20.02
  - ≤ **101M** particles / GPU
- **PIConGPU** 0.4.3-781-dev
  - ≤ **403M** particles / GPU

One Particle [concat. Bytes]

- **112 DP** / 60 SP
- 59 DP / **31 SP**

1.9x

Less Attributes by Kernel Fusion

- drop **6 floats** by fusing field gather into push kernel(s): 60 DP / 36 SP

1.6x

Memory Management, Padding & Utilization Peaks

- projected: 314M

Δ = 1.3x

- 403M

Optimization Details

Chances and Risks

- **Single Precision / Mixed Precision**
  - Validation of precision of physical observables
    - PIConGPU: benchmarked, normalized units
- **Kernel Fusion**
  - “push” kernels: gather fields multiple times
    - relatively small cost, often only one field gathered
  - slight register increase
    - less of a problem in recent GPUs
- **Tiled memory management**
  - libraries available, algorithm prototyping can be more complex

Runtime Benchmarks

Single vs. Double Precision in PIConGPU

- **Runtime Cost Increase**
  - homogeneous, warm electron-positron plasma test
  - arbitrary-order particle splines

	CIC pw linear	TSC pw quad.	PCS pw cubic
SP	**1x**	1.79x	3.38x
DP	1.50x	2.91x	

…

A. Huebl, PIConGPU 0.4.2 on Nvidia P100, https://github.com/ComputationalRadiationPhysics/picongpu/issues/2815

Summary

Strategies for Memory Optimizations in GPU PIC Libraries

- **Controlling the Memory Footprint**
  - **Memory is node-hours: in practice just as costly as walltime**
  - reduced precision; fuse kernels instead of global-memory helpers
- **Particle Memory Management**
  - **Contiguous** (AMReX / WarpX)
    - STL-like algorithms (+), rapid prototyping (+), multiple kernels (-/0), ~1.3x memory overhead (-/0)
  - **Tiled, bucket-sorted** (PMacc / PIConGPU)
    - Cache blocking (+), additional parallel algorithms (-/0)

Rely on a community library: e.g. AMReX, CoPA-Cabana, or PMacc

Meet the Teams

WarpX team*: physicists + applied mathematicians + computer scientists

ECP

BERKELEY LAB

Jean-Luc Vay (PI) | Diana Amorim | Axel Huebl | Rémi Lehe | Olga Shapoval | Maxence Thévenet | Yinjian Zhao | Edoardo Zoni | Glenn Richardson | Daniel Belkin

ATAP
ACCELERATOR TECHNOLOGY & APPLIED PHYSICS DIVISION

Ann Almgren (coPI) | John Bell | Kevin Gott | Revathi Jambunathan | Andrew Myers | Michael Rowan | Cameron Yang | Weiqun Zhang

Kevin Gott (NESAP) | Michael Rowan (NESAP)

CRD AMReX

NERSC NESAP

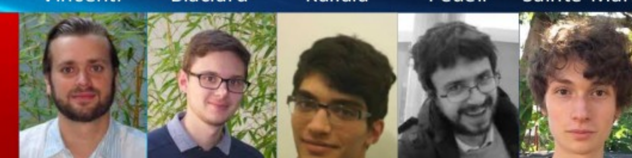
David Grote (coPI)

Marc Hogan (coPI) | Lixin Ge | Cho Ng

SLAC

Henri Vincenti | Guillaume Blaclard | Haithem Kallala | Luca Fedeli | Antonin Sainte-Marie

cea PARIS-SACLAY



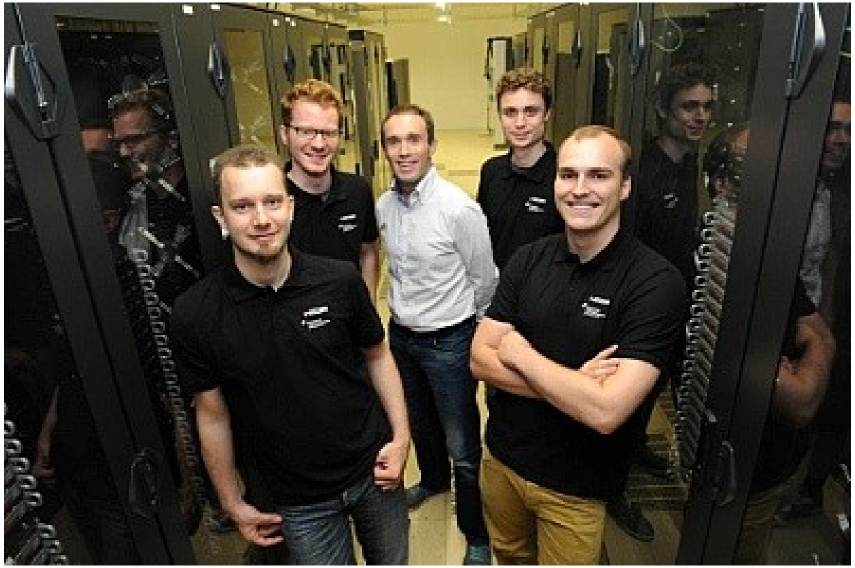
+ collaborators from CEA Saclay (France)

The project also leverages other ASCR (ECP & others) efforts via adoptions of other tools/methods, often via collaboration.

*Many at fraction of time on WarpX.

PIConGPU

github.com/**ComputationalRadiationPhysics**



Axel Huebl,[1,2] R. Widera,[2] M. Garten,[2,3] R. Pausch,[2,3]
K. Steiniger,[2] S. Bastrakov,[2] A. Debus,[2] T. Kluge,[2]
S. Ehrig,[2] F. Meyer,[2] M. Werner,[2,3] B. Worpitz,[4]
A. Matthes,[2,3] K. Bastrakova,[2] F. Poeschel,[2,3] F. Koller,[2]
S. Starke,[2] and M. Bussmann[2]
ORNL Frontier CAAR:
Matt Leinhauser,[4] Josh Davis,[4] Jose Monsalve Davis,[4]
Sunita Chandrasekaran[4]

[1] Lawrence Berkeley National Laboratory [2] Helmholtz-Zentrum Dresden – Rossendorf
[3] Technische Universität Dresden [4] LogMeIn, Inc. [4] University of Delaware

HZDR

HELMHOLTZ ZENTRUM DRESDEN ROSSENDORF

Acknowledgements

Talk by Axel Huebl (LBNL), axelhuebl@lbl.gov

- **WarpX (LBNL + NERSC + LLNL + SLAC, U.S. + CEA, France)**

This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility operated under Contract No. DE-AC02-05CH11231, resources of the Oak Ridge Leadership Computing Facility located in the Oak Ridge National Laboratory, which is supported by the Office of Science of the Department of Energy under Contract DE-AC05-00OR22725, and is 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).

- **PIConGPU (HZDR + CASUS, Germany + UDEL, U.S.)**

This research used resources of the Oak Ridge Leadership Computing Facility located in the Oak Ridge National Laboratory, which is supported by the Office of Science of the Department of Energy under Contract DE-AC05-00OR22725.

This project received funding within the MEPHISTO project (BMBF-Förderkenn-zeichen 01IH16006C). This project has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 654220.

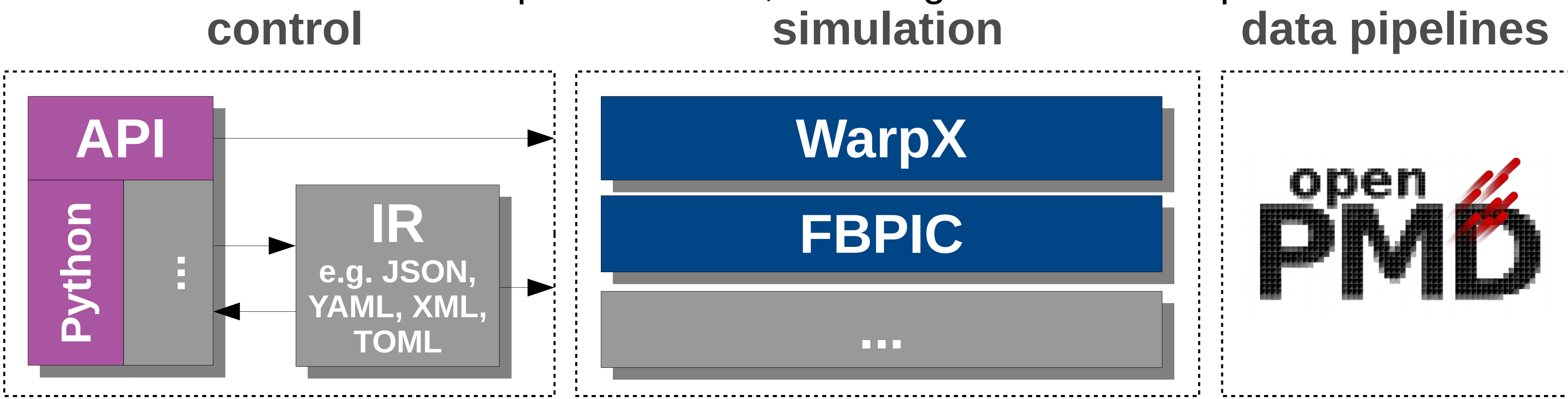
Backup Slides: Standardization Efforts

Particle-In-Cell Modeling Interface

github.com/picmi-standard/picmi

PICMI Standard

- **Standard input format for Particle-In-Cell codes**
  - dictionary as input syntax
  - primary implementation: Python classes
  - extensible for code-specific needs, handling of additional options



Exascale Challenge: I/O Scalability

Titan I/O Weak Scaling with PIConGPU

$$T_{\mathrm{eff}} \equiv \frac{N \times S}{t_{\mathrm{I/O}}}$$

SYSTEM SPECS	TITAN	SUMMIT	FRONTIER
Peak Performance	27 PF	200 PF	> 1.5 EF
Storage	32 PB, 1 TB/s, Lustre Filesystem	250 PB, 2.5 TB/s, GPFS™	2-4x performance and capacity of Summit's I/O subsystem. Frontier will have near node storage like Summit.

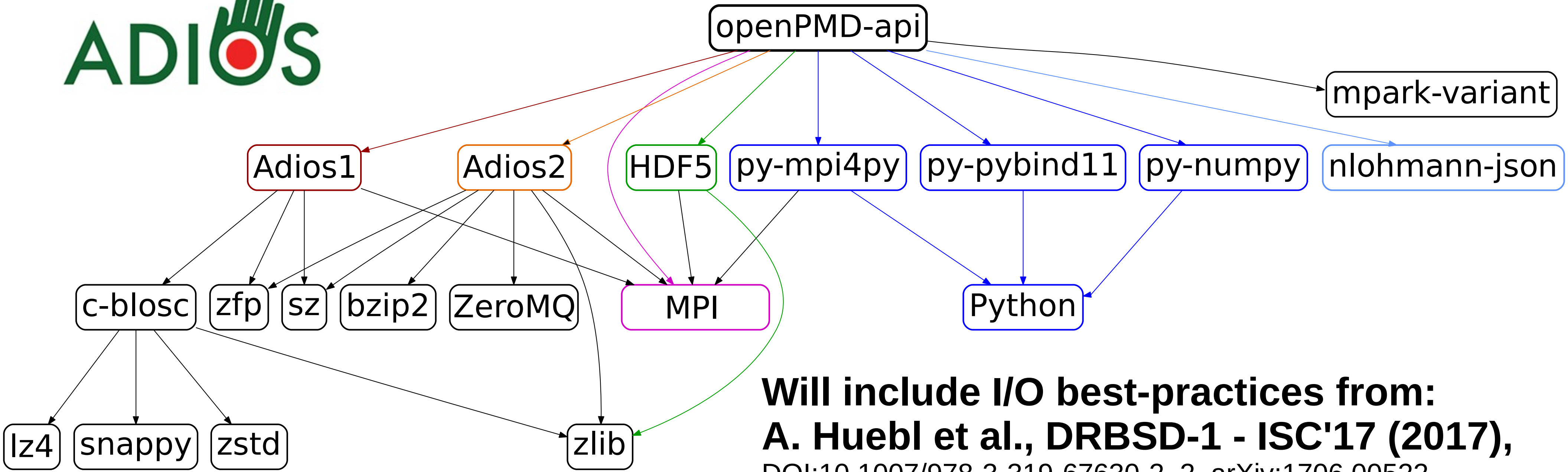
1/3x **1/3x**

In situ approaches: **tightly** versus **loosely** coupled workflows

A. Huebl et al., DRBSD-1 - ISC'17 (2017),
DOI:10.1007/978-3-319-67630-2_2, arXiv:1706.00522

Open Standard for Particle-Mesh Data

Loosely Coupled Pipelines: **openPMD-api**



Will include I/O best-practices from:
A. Huebl et al., DRBSD-1 - ISC'17 (2017),
DOI:10.1007/978-3-319-67630-2_2, arXiv:1706.00522

Available via: Spack, CONDA-FORGE, python Package Index

Open Standard for Particle-Mesh Data

www.openPMD.org

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