DOI:10.5281/zenodo.5905053

openPMD
A Scientific, Community Meta-Data Standard

Plenary talk by: Axel Huebl

Lawrence Berkeley National Laboratory

On behalf of the openPMD Community (see last slide)

LPA Online Workshop on Control Systems and Machine Learning

Munich, Germany (virtual)
January 26th, 2022

National Nuclear Security Administration

CAMPA Consortium for Advanced Modeling of Particle Accelerators

ATAP ACCELERATOR TECHNOLOGY & APPLIED PHYSICS DIVISION

BERKELEY LAB Lawrence Berkeley National Laboratory

U.S. DEPARTMENT OF ENERGY | Office of Science

Acknowledgements

This project has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 654220. Supported by the Consortium for Advanced Modeling of Particles Accelerators (CAMPA), funded by the U.S. DOE Office of Science under Contract No. DE-AC02-05CH11231.

CAMPA Consortium for Advanced Modeling of Particle Accelerators

This work was partially funded by the Center of Advanced Systems Understanding (CASUS), which is financed by Germany's Federal Ministry of Education and Research (BMBF) and by the Saxon Ministry for Science, Culture and Tourism (SMWK) with tax funds on the basis of the budget approved by the Saxon State Parliament.

ECP EXASCALE COMPUTING PROJECT

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 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 and of 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.

Outline

- **openPMD Concepts**
  - design principles of our meta-data standard
  - common challenges
  - base standard and extensions

- **Common Challenges & Possible Solutions**
  - data bottlenecks: HPC & Control Systems
  - data reduction & backlog criteria
  - transition to data streams
  - faster reads: data layouts & in situ statistics

- **Open Ecosystem**
  - projects, libraries, tooling, examples
  - open community & possible future directions

Outline

- **openPMD Concepts**
  - design principles of our meta-data standard
  - common challenges
  - base standard and extensions

- **Common Challenges & Possible Solutions**
  - data bottlenecks: HPC & Control Systems
  - data reduction & backlog criteria
  - transition to data streams
  - faster reads: data layouts & in situ statistics

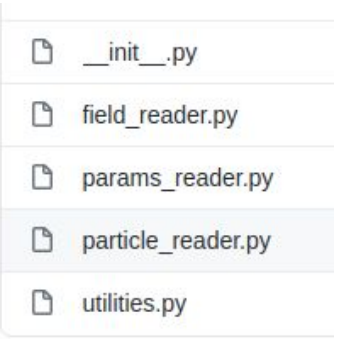
- **Open Ecosystem**
  - projects, libraries, tooling, examples
  - open community & possible future directions

Design Principles

Open Standard for Particle-Mesh Data

- **high-level** description
- **minimal**: users can add more
- **human** readable & **machine** actionable

h5py reader:
<400 lines
of code



- file format **agnostic, portable**
- **scalable** from desktop to supercomputer
- **forward-updatable:**

  start “strict“, relax later


```
openPMD_updater_h5 -i example.h5
```

Common Data Challenges

- **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

extensions
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

Organizing Scientific Records

- **electric field** $\vec{E}(\vec{r})$

  / … / meshes / E /

  x y z

- **temperature** $T(\vec{r})$

  / … / meshes /

  T

- **electron position** $\vec{r}$

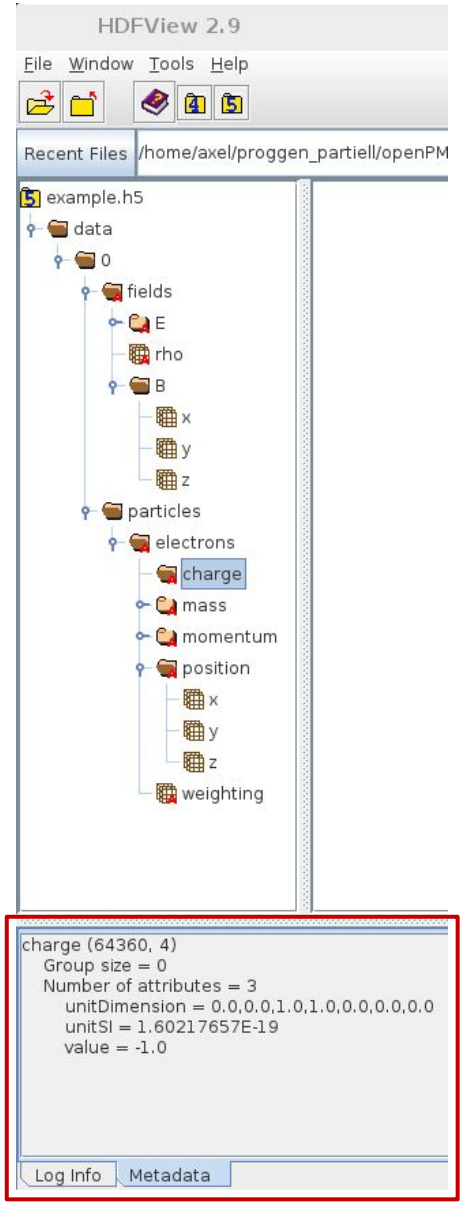
  / … / particles / electrons / position /

  x y z

group

data

attributes



extensions to the **base standard** define additional, compatible keywords

Unit System

- **unitDimension**

  automated description of physical dimension

  only powers of base dimensions

  length **L**, mass **M**, time **T**, electric current **I**, thermodynamic temperature **theta**, amount of substance **N**, luminous intensity **J**

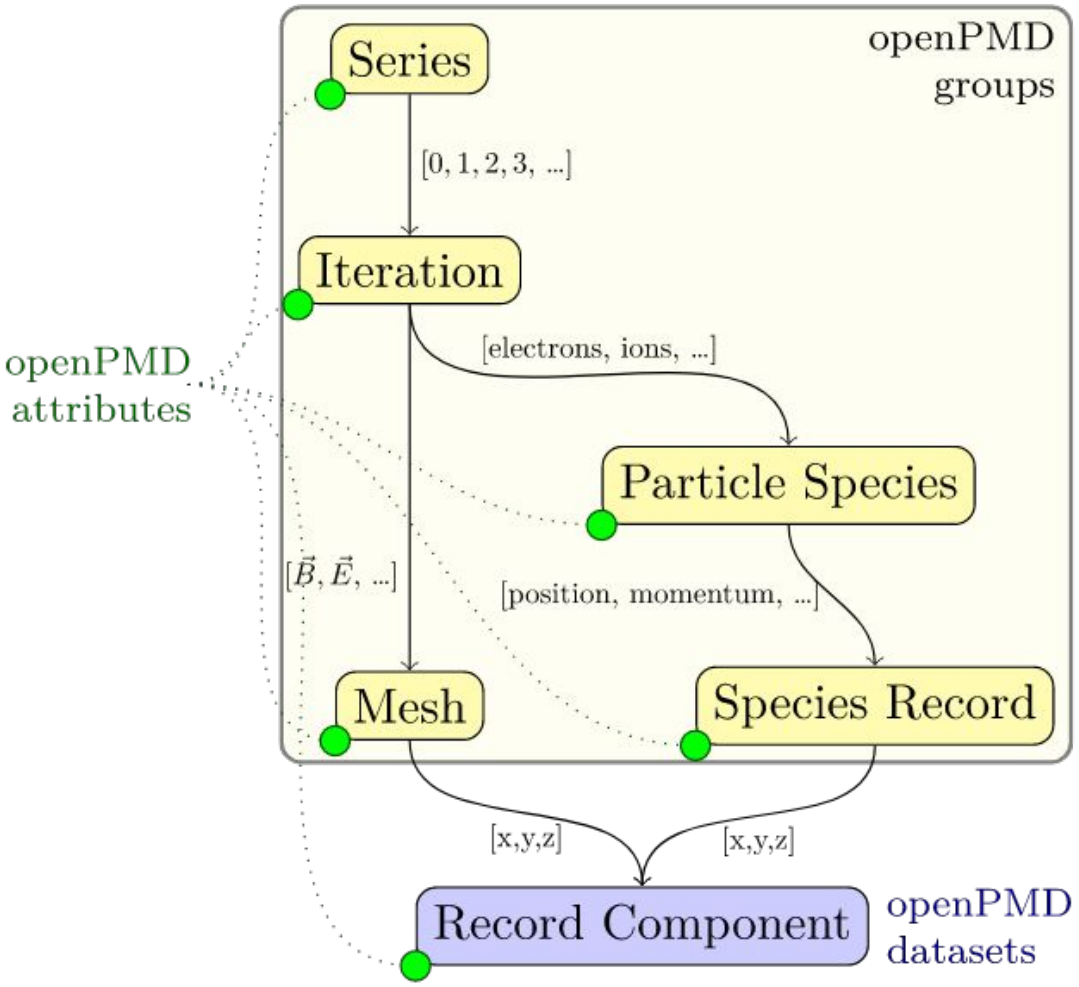
  magnetic field: [B] = M / (I * T^2)
  → **(0., 1., -2., -1., 0., 0., 0.)**

- **unitSI** (recommended)

  relation to an *absolute* unit system

attributes

openPMD - Hierarchy for Data Series

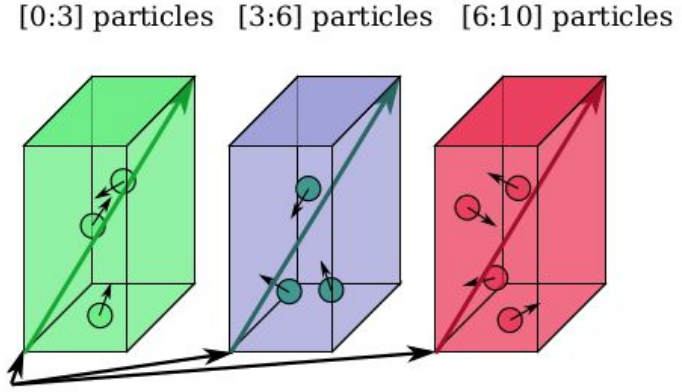


- **Structure** for series & snapshots/iterations
- Records for **physical observables**
- Attributes: **conversion, description**

- **Group-based**: file per series or subfiles
  File-based: file per iteration
  Variable-based: use ADIOS2 variables/steps

- Constants, mixed precision, complex numbers

- domain-decomposition



openPMD - A Self-Describing, FAIR Standard

Findable: Standardized metadata to identify the data producer

```
string    /author              attr    = "franz"
string    /software            attr    = "PIConGPU"
string    /softwareVersion     attr    = "0.5.0-dev"
```

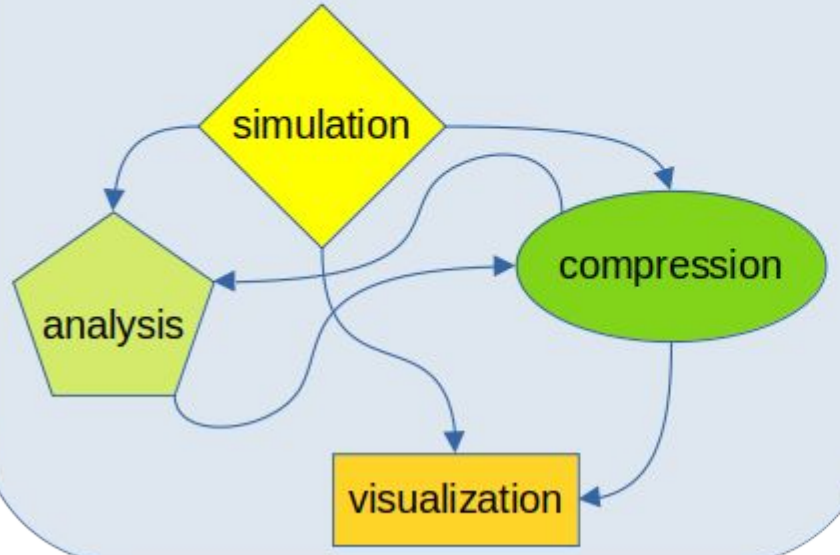
Accessible: Open standard, implementable in various formats

*currently implemented, but not limited to

"The FAIR Guiding Principles for scientific data management and stewardship" (Mark D. Wilkinson et al.)

openPMD - A Self-Describing, FAIR Standard

Interoperable:
Data exchange spans applications, platforms and teams



Reusable:
Rich and standardized description for physical quantities

Name	Value
axisLabels	[b'z' b'y' b'x']
dataOrder	b'C'
fieldSmoothing	b'none'
geometry	b'cartesian'
gridGlobalOffset	[0. 0. 0.]
gridSpacing	[4.252342 1.0630856 4.252342]
gridUnitSI	4.1671151662e-08
position	[0. 0. 0.]
timeOffset	0.0
unitDimension	[-3. 0. 1. 1. 0. 0. 0.]
unitSI	15399437.98944343

"The FAIR Guiding Principles for scientific data management and stewardship" (Mark D. Wilkinson et al.)

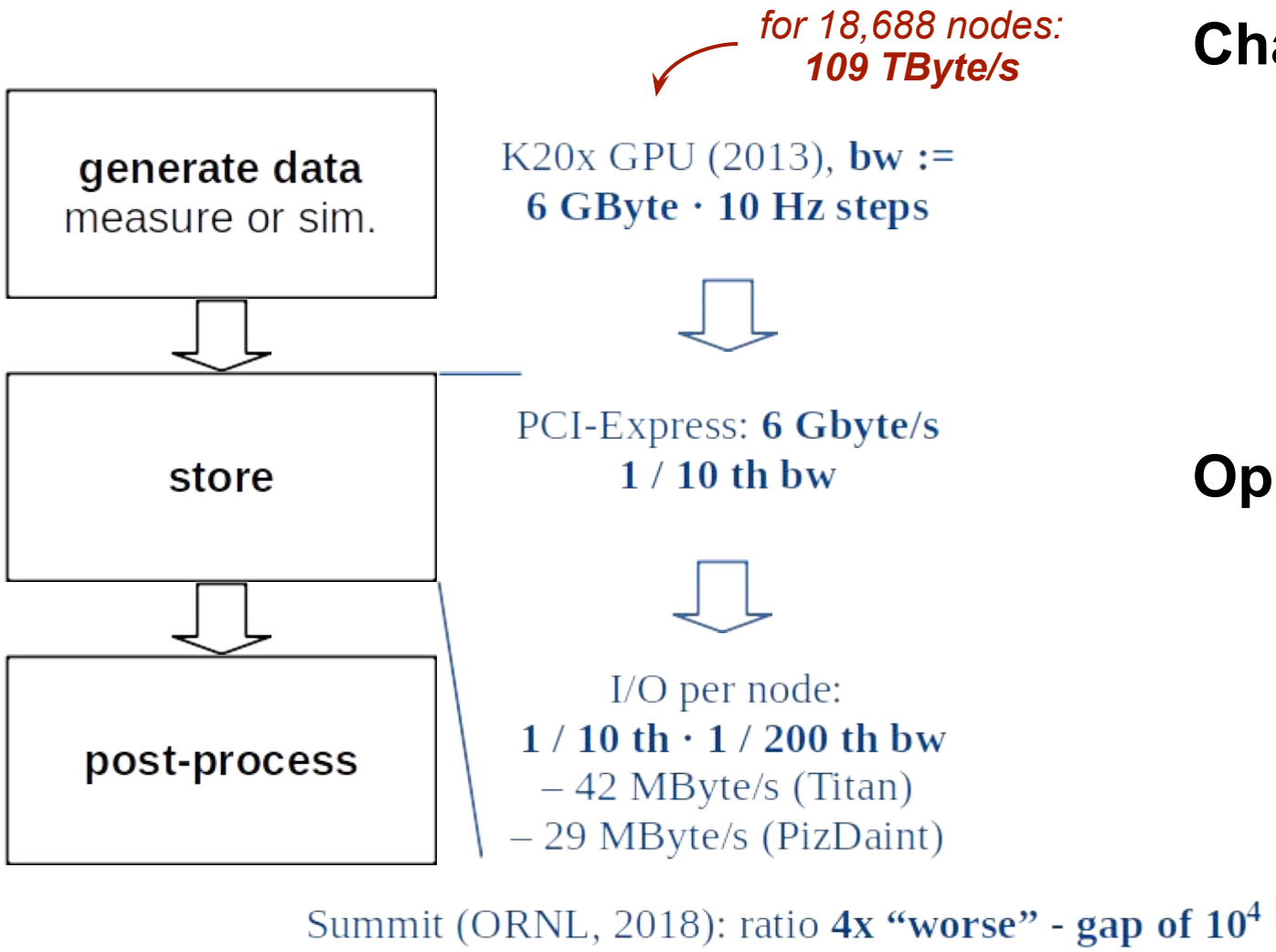
Outline

- **openPMD Concepts**
  - design principles of our meta-data standard
  - common challenges
  - base standard and extensions

- **Common Challenges & Possible Solutions**
  - data bottlenecks: HPC & Control Systems
  - data reduction & backlog criteria
  - transition to data streams
  - faster reads: data layouts & in situ statistics

- **Open Ecosystem**
  - projects, libraries, tooling, examples
  - open community & possible future directions

HPC Background: Our Data Processing Bottlenecks Look Alike



A Huebl et al., ISC 2017, DOI:10.1007/978-3-319-67630-2_2 (2017)

Challenges

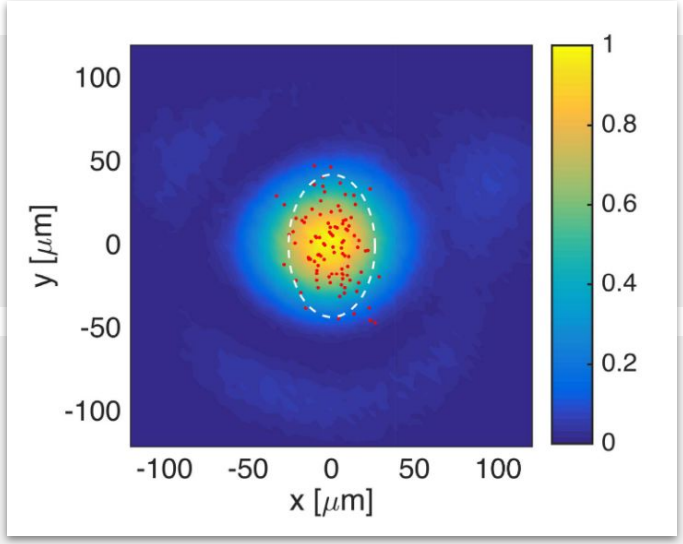
- **3 orders of magnitude gap** between producing devices and storage
- "store & analyze everything" is *unaffordable*

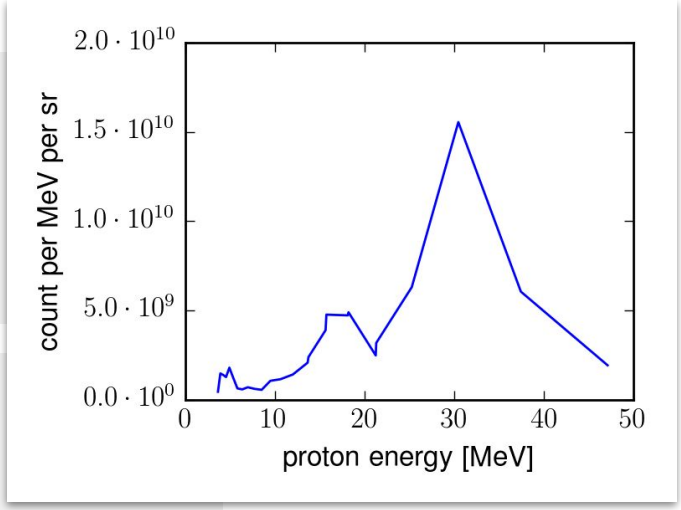
Opportunities

- analysis tasks have varying **fidelity** needs
- many common tasks can be done **in situ**
- manual steps: limit the sampling of raw data to **setup phase**

Data Processing & Reduction Examples

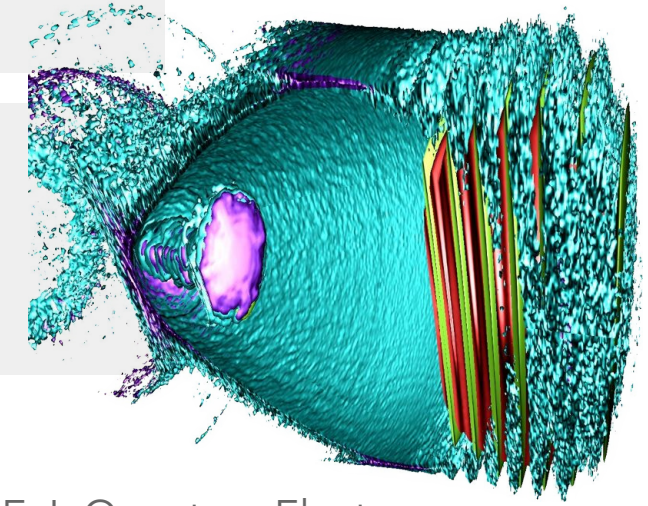
Binning of a **spectrogram**
Fitting of an **ellipsoid**





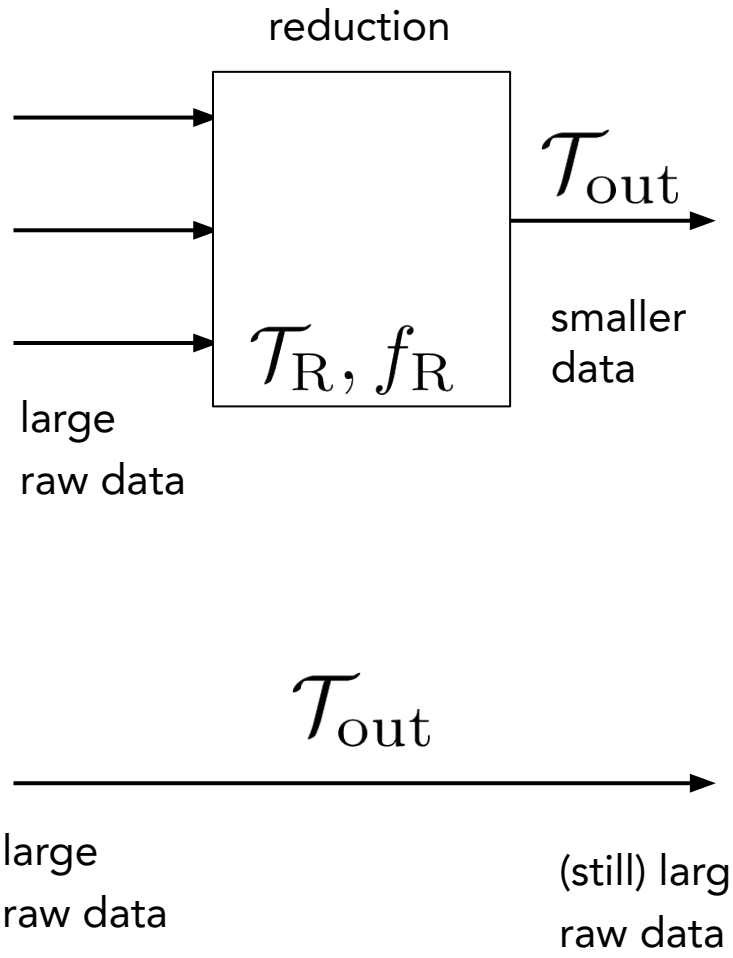
Compression (lossless/lossy)

Ray-casting 3D data,
training a neural network, etc.

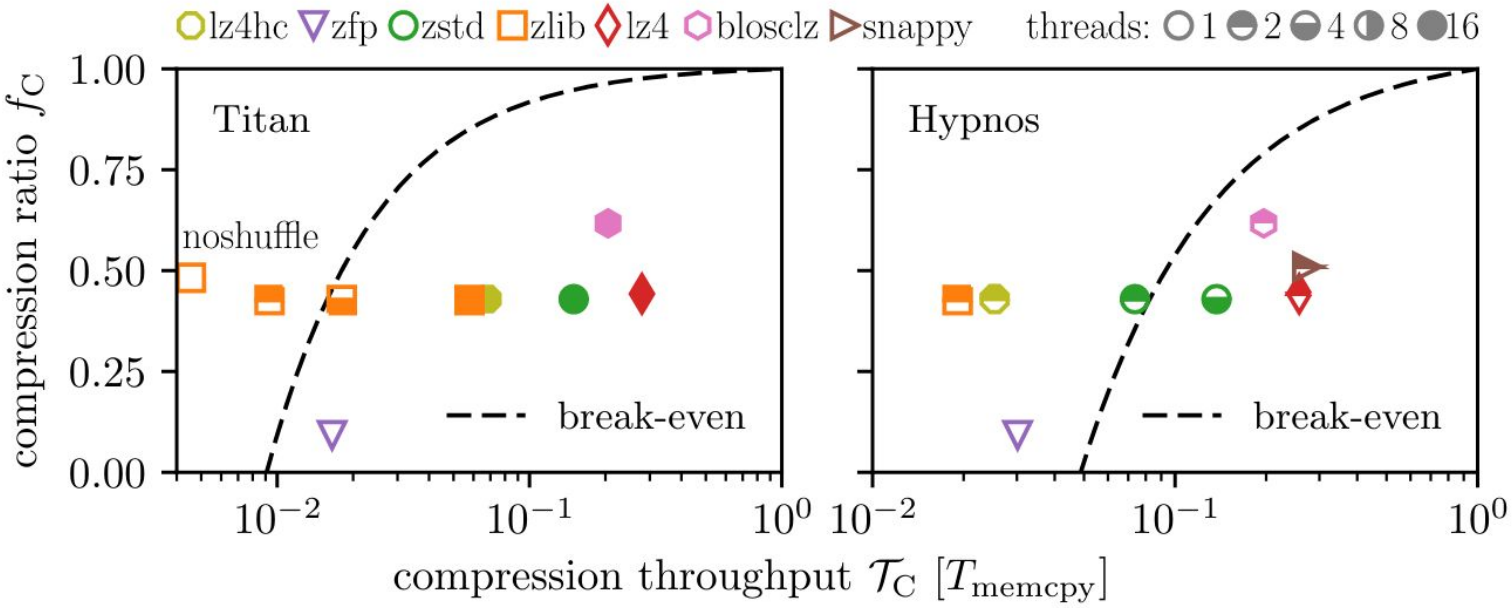


A Matthes, A Huebl et al., ISC 2017, DOI:10.14529/jsfi160403 (2017);
A Huebl et al., ISC 2017, DOI:10.1007/978-3-319-67630-2_2 (2017);
K Nakamura et al., IEEE J. Quantum Electron, DOI:10.1109/JQE.2017.2708601 (2019)

Avoid Backlog: Design Criteria for Data Reduction Pipelines



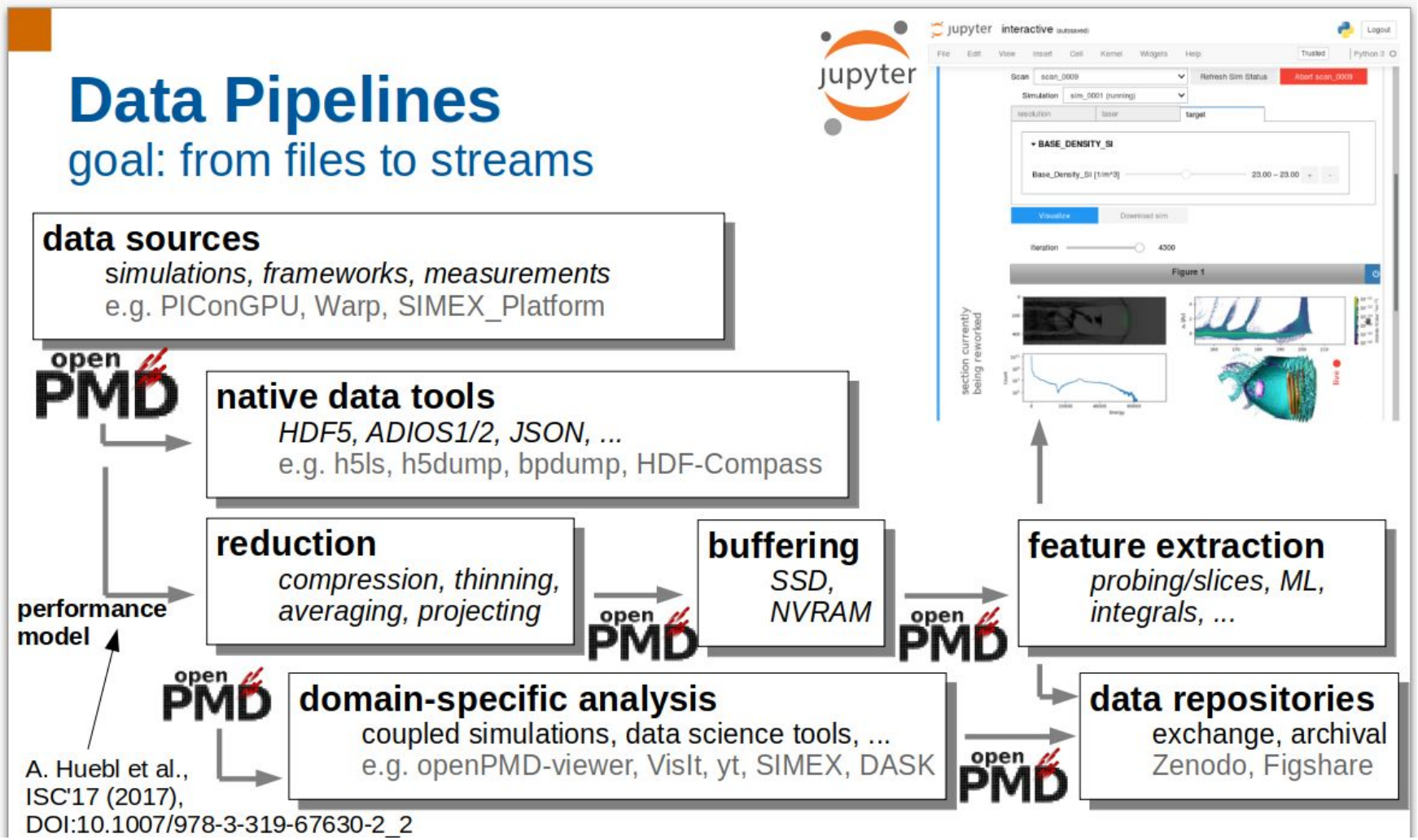
$$\frac{\mathcal{T}_R \times (1 - f_R)}{1 - \mathcal{T}_R} > \mathcal{T}_{out}$$

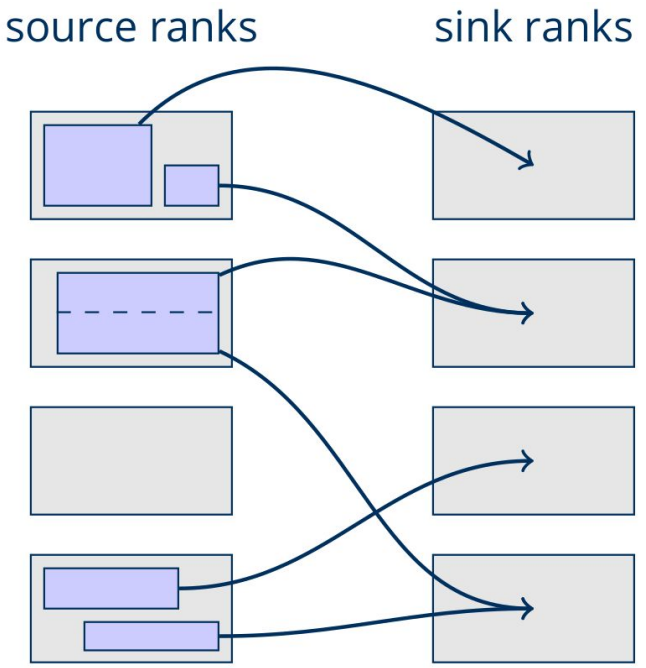


Result: **trade compute for throughput**, >100 GByte perceived application throughput and **280 GByte/s** peak parallel filesystem throughput

A Huebl et al., "On the Scalability of Data Reduction Techniques in Current and Upcoming HPC Systems from an Application Perspective," ISC High Performance Workshops, DOI:10.1007/978-3-319-67630-2_2 (2017)

Streaming: Avoid Files Altogether





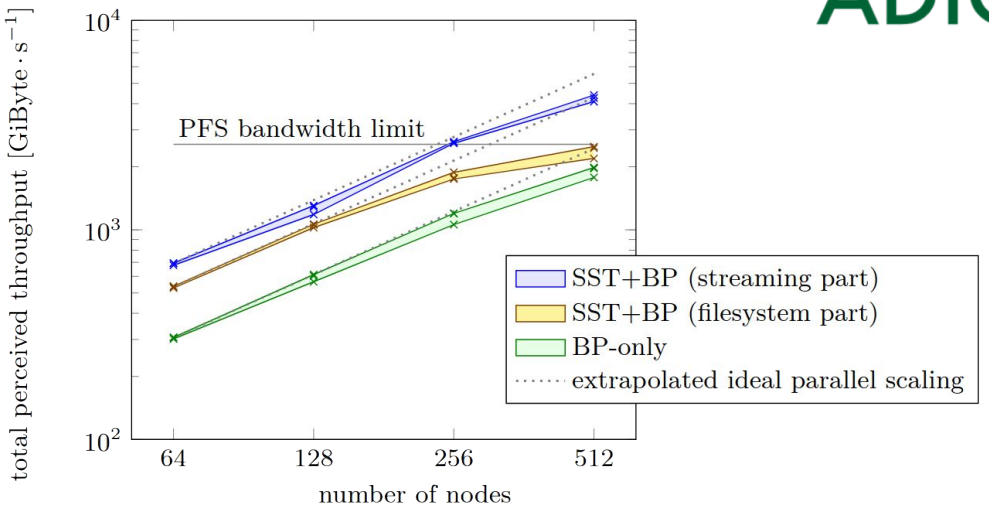
Right: For each arrow in the left figure, one could change the mapping between computing nodes (ranks).

F. Poeschel, … A. Huebl. "Transitioning from file-based HPC workflows to streaming data pipelines with openPMD and ADIOS2," accepted in SMC21, https://arxiv.org/abs/2107.06108 (2021)

Performance, Data Layouts and Fileless I/O

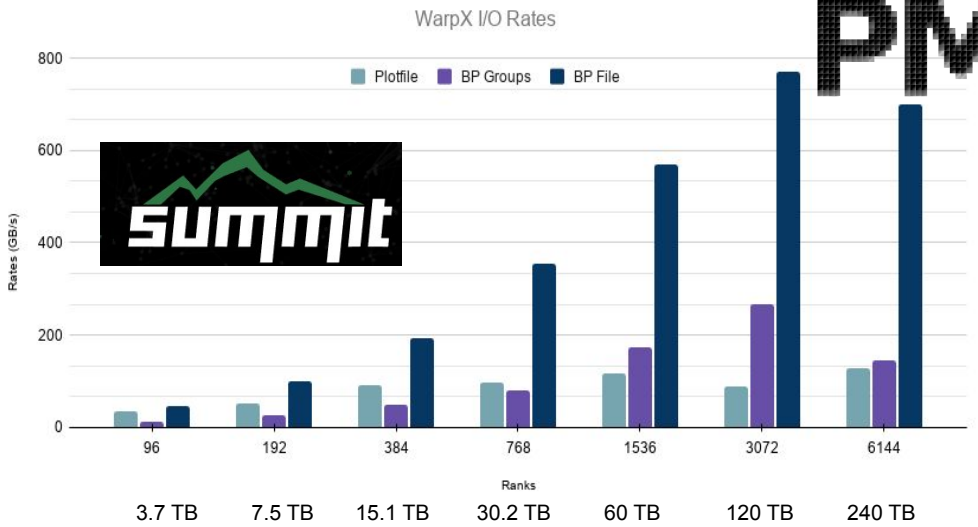
Application Challenges

- R&D in: scalable techniques, data layouts, libraries
- scientific data analysis & sharing

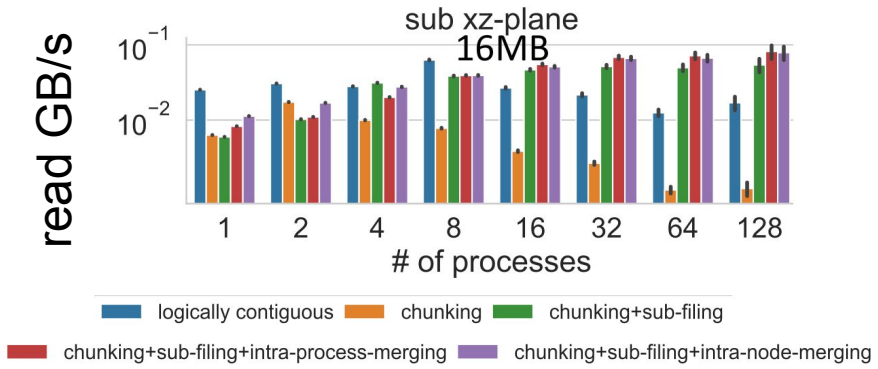


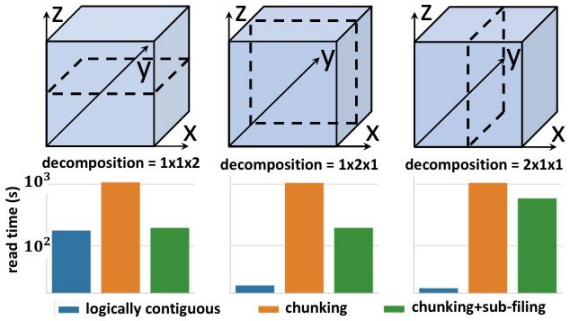
Streaming Data Pipelines: arXiv:2107.06108
by F Poeschel, A Huebl et al., SMC21 (2021)

Online Data Layout Reorganization:
DOI:10.1109/TPDS.2021.3100784
by L Wan, A Huebl et al., TPDS (2021)



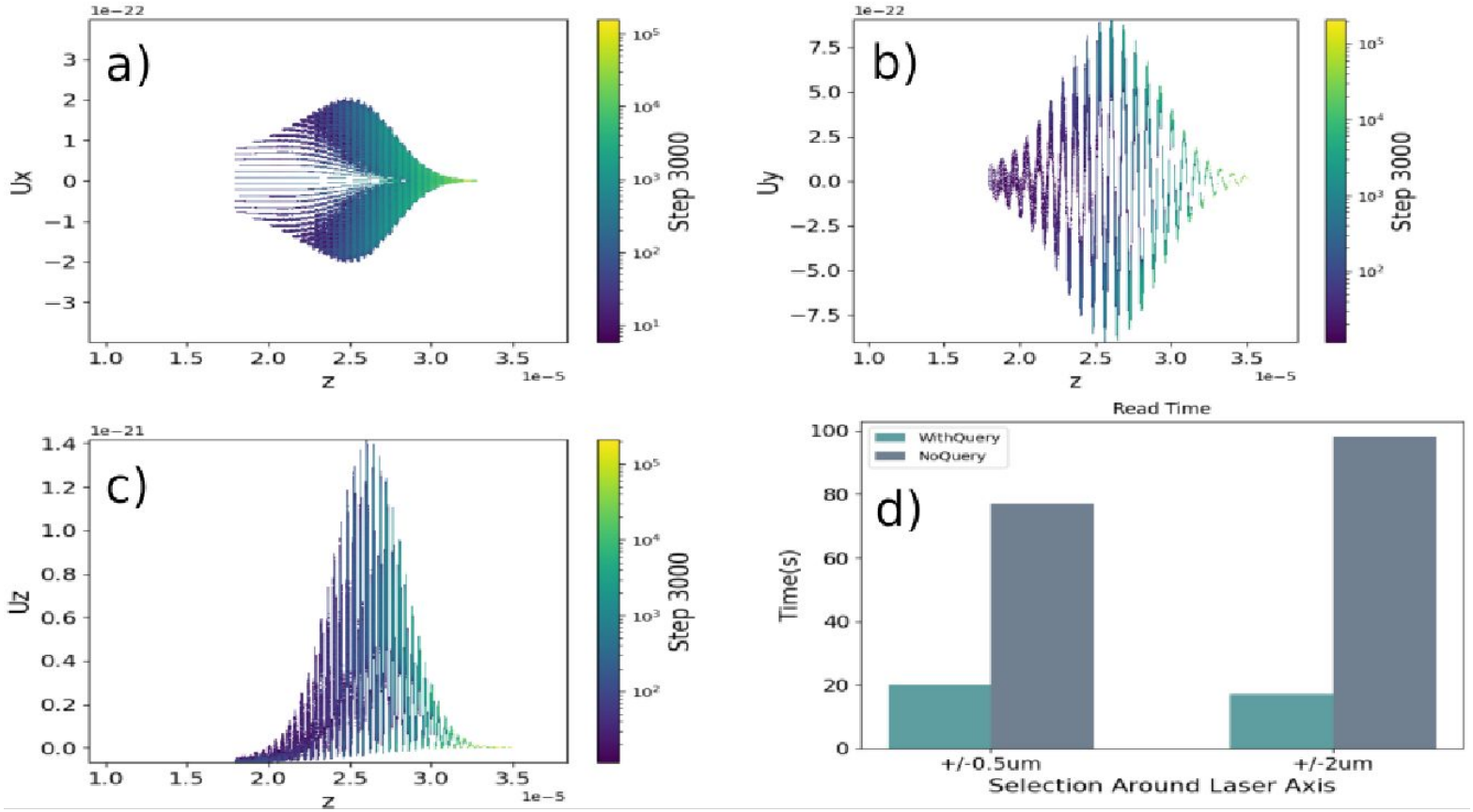
Write: plotfiles → ADIOS BP per rank & step → append to files





Impact of decomposition schemes when reading

In-Transport Data Processing: Data Statistics



Data analysis using region of interest filtering with ADIOS queries. a)-c) Phase space projections of plasma particles oscillating in a laser pulse, filtered close to the laser axis. d) Read time comparison between conventional reads and pre-filtered reads with queries.

J Gu, P Davis, G Eisenhauer, W Godoy, A Huebl, S Klasky, M Parashar, N Podhorszki, F Poeschel, J-L Vay, L Wan, R Wang, and K Wu, "Organizing Large Data Sets for Efficient Analyses on HPC Systems," submitted (2021)

Outline

- **openPMD Concepts**
  - design principles of our meta-data standard
  - common challenges
  - base standard and extensions

- **Common Challenges & Possible Solutions**
  - data bottlenecks: HPC & Control Systems
  - data reduction & backlog criteria
  - transition to data streams
  - faster reads: data layouts & in situ statistics

- **Open Ecosystem**
  - projects, libraries, tooling, examples
  - open community & possible future directions

First Steps

https://github.com/openPMD

openPMD

Open Standard for Particle-Mesh Data Files

https://www.openPMD.org axelhuebl@lbl.gov

Overview | Repositories 17 | Packages | People 55 | Teams 5 | Projects | Settings

Pinned

Customize your pins

openPMD-standard Public
Open Standard for Particle-Mesh Data
☆ 52 ⑂ 20

openPMD-projects Public
Overview on Projects around openPMD
☆ 6 ⑂ 6

openPMD-viewer Public
Python visualization tools for openPMD files
Jupyter Notebook ☆ 45 ⑂ 36

openPMD-api Public
C++ & Python API for Scientific I/O
C++ ☆ 75 ⑂ 36

openPMD-example-datasets Public
HDF5 Example Files
Python ☆ 6 ⑂ 2

openPMD-validator Public
Validator and Example Scripts
Python ☆ 4 ⑂ 8

Repositories

People

View all

Invite someone

Top languages

Python C++ Jupyter Notebook
C CSS

Application Examples in Particle Acceleration & Laser-Plasma

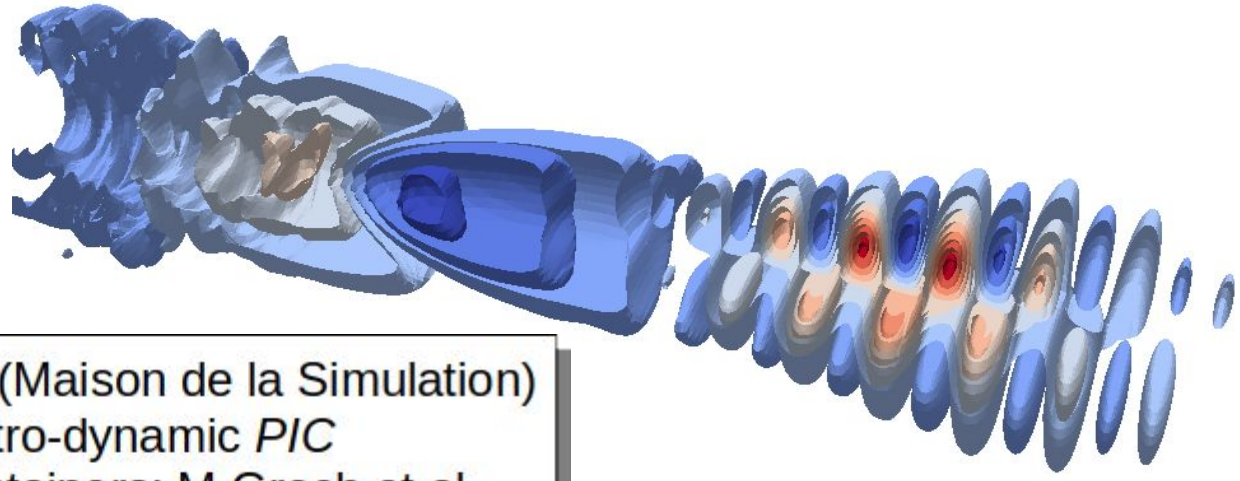
Simulations:

- WarpX
- PIConGPU
- HiPACE++
- Wake-T
- VPIC (prototype)
- ...

Analysis:

- openPMD-viewer
- ParaView
- VisualPIC
- APtools
- various scripts
- ...

openPMD/openPMD-projects



PIConGPU (HZDR)
electro-dynamic particle-in-cell code
maintainers: A Huebl, M Bussmann et al.

Warp (LBNL, LLNL)
electro-dynamic/static particle-in-cell code
maintainers: JL Vay, D Grote, R Lehe et al.

FBPIC (LBNL, DESY)
spectral, fourier-bessel particle-in-cell code
maintainers: R Lehe, M Kirchen et al.

SIMEX Platform (EUCALL, European XFEL)
simulation of advanced photon experiments
maintainer: C Fortmann-Grote

Smilei (Maison de la Simulation)
electro-dynamic *PIC*
maintainers: M Grech et al.

UPIC-Emma 2.0 (UCLA)
spectral *PIC*
maintainers: F Tsung et al.

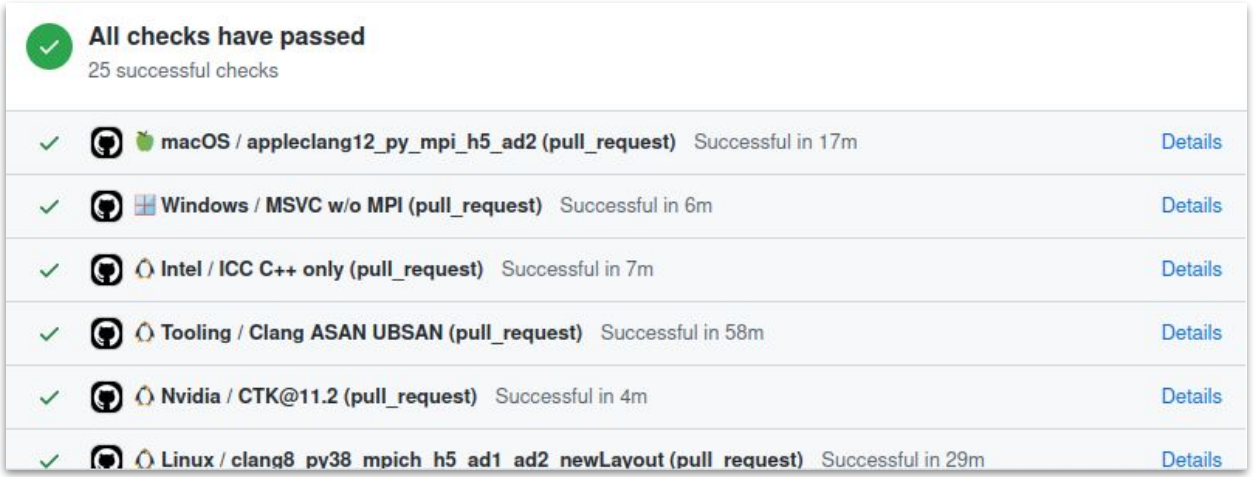
Bmad (Cornell)
particle accelerator
maintainers: D Sagan et al.

scientific simulations

Open Science: Easy to Install, Use, Modify and Reproduce

Online Documentation:
openpmd-api.readthedocs.io

Open-Source Development & Tests:
github.com/openPMD/openPMD-api



Rapid and easy installation on any platform:

python3 -m pip install openpmd-api

brew tap openpmd/openpmd
brew install openpmd-api

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

conda install -c conda-forge openpmd-api

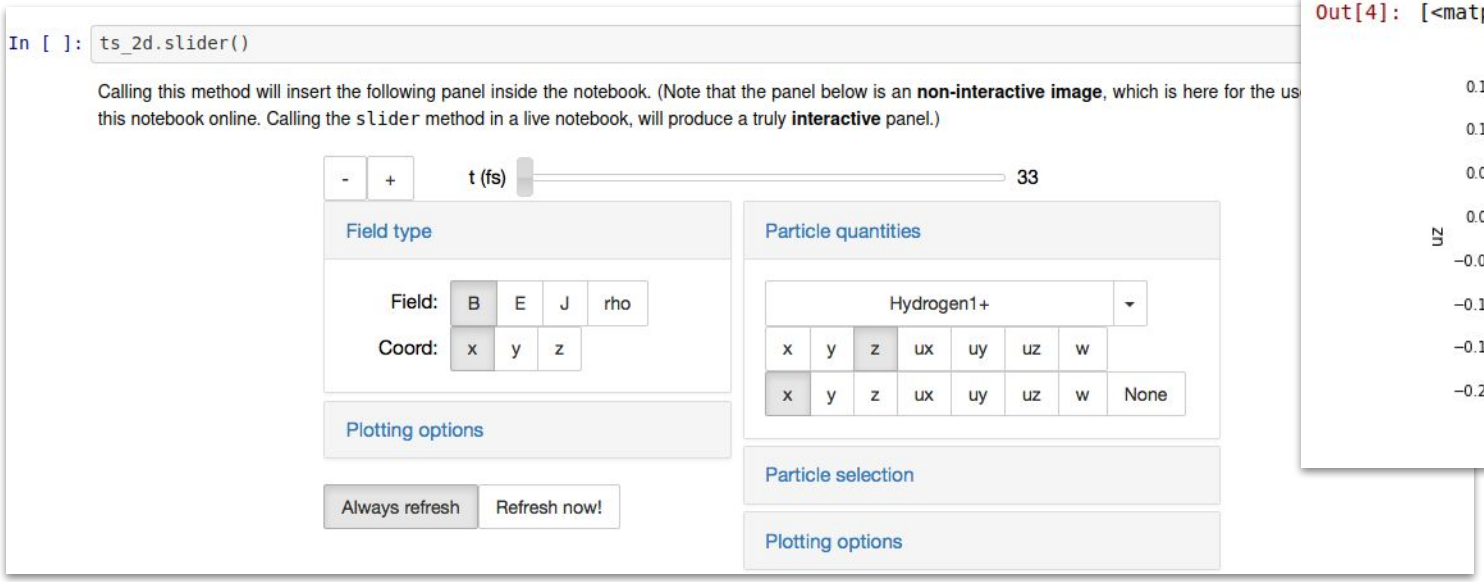
spack install openpmd-api

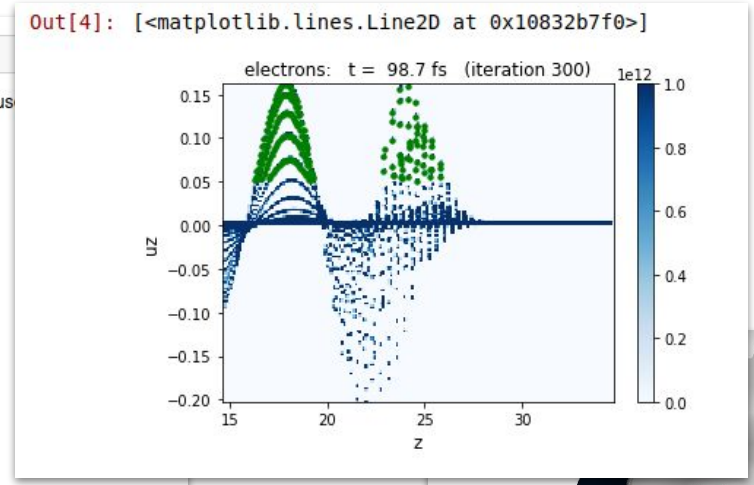
module load openpmd-api

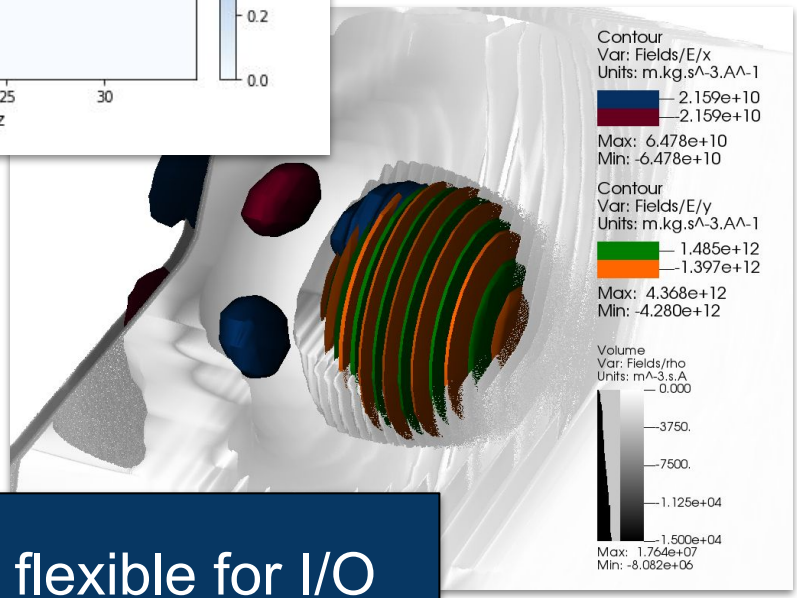
We Standardize & Develop Scalable Data Methods

… and integrate them for scientific productivity

including data analytics frameworks & graphical user interfaces







openPMD DASK ParaView yt VisIt

Open standardization, i.e. openPMD, makes us flexible for I/O libraries, tooling & domain-science needs.

Integration into the BELLA Control System

BELLA GEECS

- **G**eneralized **E**quipment and **E**xperiment **C**ontrol **S**ystem
- control and data acquisition system

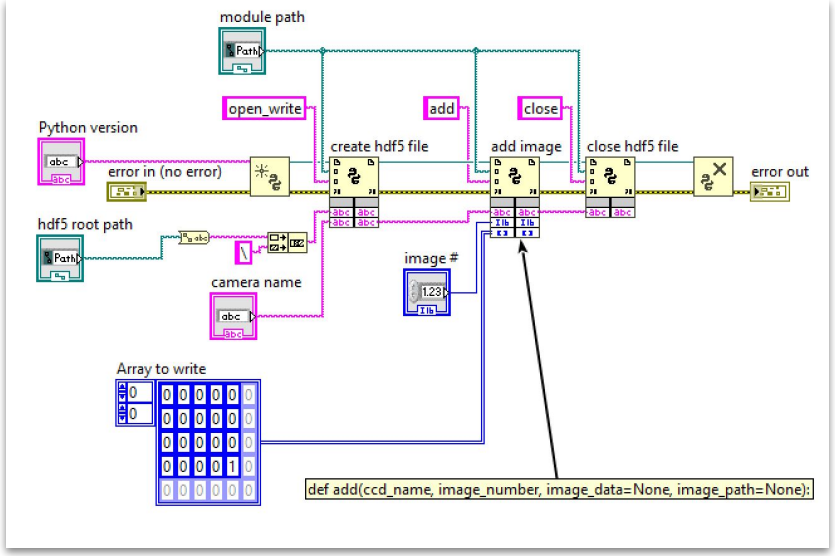
openPMD/openPMD-CCD

GEECS-BELLA

openPMD-CCD

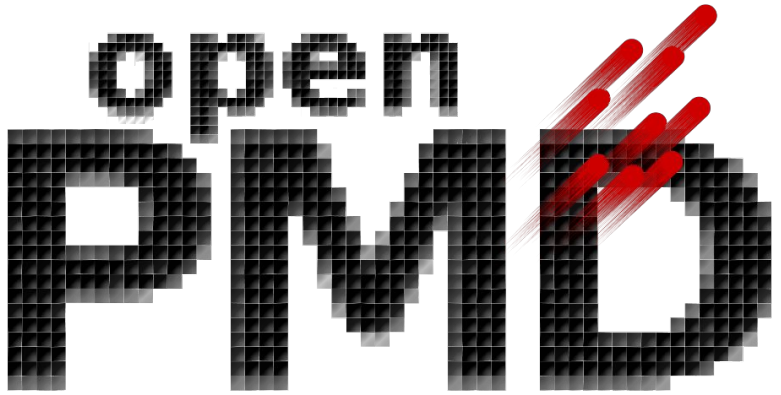
- **Python** module for organizing CCD images with openPMD
- Optional integration with **LabView 2020+**

```python
from openpmd_ccd import CCD

scan = CCD("defaultCam_scan001.h5", overwrite=True,
           name="Go Pro", model="HERO8 Black", serial="12345678",
           operator="Axel Huebl <axelhuebl@lbl.gov>",
           # resolution=None, roi=None, exposure_time=None
)

scan.add(0, "tests/data/Scan005_SimCam_001.png")
scan.add(1, np.array([[1., 2.], [3., 4.]]))

# scan.recalibrate(...)

scan.close()
```



Community

https://www.openPMD.org



HZDR
HELMHOLTZ ZENTRUM DRESDEN ROSSENDORF

BERKELEY LAB

DESY

OAK RIDGE National Laboratory

CASUS
CENTER FOR ADVANCED SYSTEMS UNDERSTANDING

TECHNISCHE UNIVERSITÄT DRESDEN

iOQ JENA
Friedrich-Schiller-Universität

CBG
Max Planck Institute of Molecular Cell Biology and Genetics

UCLA

PI PERIMETER INSTITUTE

Lawrence Livermore National Laboratory

SLAC
NATIONAL ACCELERATOR LABORATORY

MAISON DE LA SIMULATION

CORNELL UNIVERSITY FOUNDED A.D. 1865

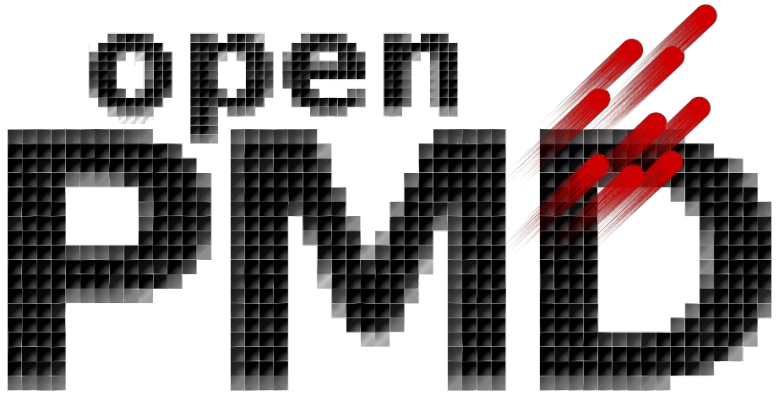
EUCALL

European XFEL

NVIDIA

Community

https://www.openPMD.org



The **openPMD standard** is co-authored by Axel Huebl, Rémi Lehe, Jean-Luc Vay, David P. Grote, Ivo Sbalzarini, Stephan Kuschel, David Sagan, Frédéric Pérez, Fabian Koller, Franz Poeschel, Carsten Fortmann-Grote, Ángel Ferran Pousa, Juncheng E, and Michael Bussmann.

The authors are thankful for the **community contributions** to libraries, software ecosystem, user support, review and integrations. Particularly, thank you to Yaser Afshar, Lígia Diana Amorim, James Amundson, Weiming An, Igor Andriyash, Ksenia Bastrakova, Jean Luca Bez, Richard Briggs, Heiko Burau, Jong Choi, Ray Donnelly, Dmitry Ganyushin, Marco Garten, Lixin Ge, Daniel Grassinger, Alexander Grund, Junmin Gu, Marc W. Guetg, Sören Jalas, Manuel Kirchen, John Kirkham, Scott Klasky, Noah Klemm, Fabian Koller, Mathieu Lobet, Christopher Mayes, Ritiek Malhotra, Paweł Ordyna, Richard Pausch, Norbert Podhorszki, David Pugmire, Felix Schmitt, Erik Schnetter, Dominik Stańczak, Klaus Steiniger, Michael Sippel, Frank Tsung, René Widera, and Erik Zenker!

Summary & Outlook

openPMD is

- a **documented standard** and
- a large **open-source ecosystem**
  - documentation, example data, validation, scripts, integrations, reference libraries

Our community …

- **evolves** and practices **open standardization**
- integrates multiple **state-of-the art** computer science **formats & tools**
  - we are PByte-scale data veterans
  - from parallel, in-transport data processing to file-less scripting workflows using RMDA

Ideas for Control Systems & AI/ML

- Build **compatibility from the bottom up**
  - common data -> shared analysis -> control (loops)
- Start **Data Repositories** & Data Portals
  - AI/ML training, testing, validation, calibration
  - studies over larger, combined data sets
  - preservation, recasting and reinterpretation (see: LHC)

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

A growing number of international contributors

- try the tools, interact with us on what your needs are, contribute/share data & code

openPMD openpmd.slack.com openpmd.org