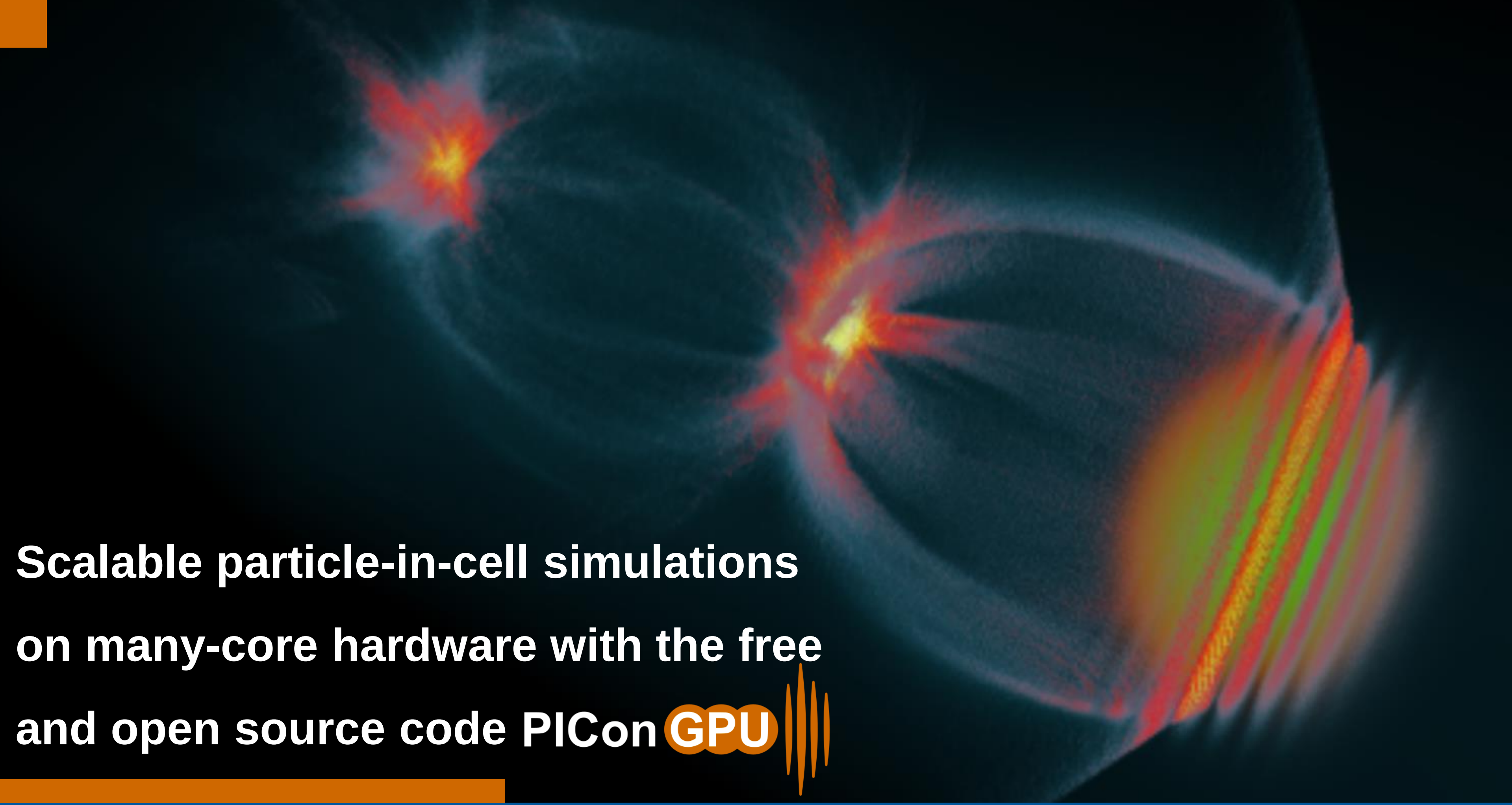


Scalable particle-in-cell simulations on many-core hardware with the free and open source code PIConGPU

K. Steiniger, S. Bastrakov, A. Debus, S. Ehrig,
M. Garten, A. Huebl, A. Matthes, F. Meyer, R. Pausch,
F. Poeschel, S. Rudat, S. Starke, M. Werner, R. Widera,
B. Worpitz, M. Bussmann

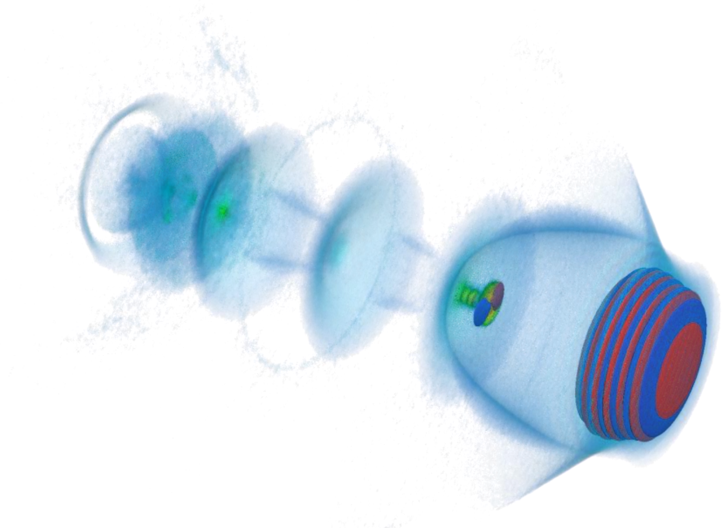
Partner von DRESDEN concept exzellenz-dresden.de

HZDR
HELMHOLTZ ZENTRUM DRESDEN ROSSENDORF

Why are we developing a PIC code for supercomputers?

For fast 3D high iteration, high resolution studies of real world setups, in e.g.,

Electron acceleration with lasers



Recently: **3D Start-to-end** simulations of a staged LWFA-driven plasma wakefield accelerator (*next talk by A. Debus*).

Includes: real gas profile,
measured Laguerre-Gauss laser modes

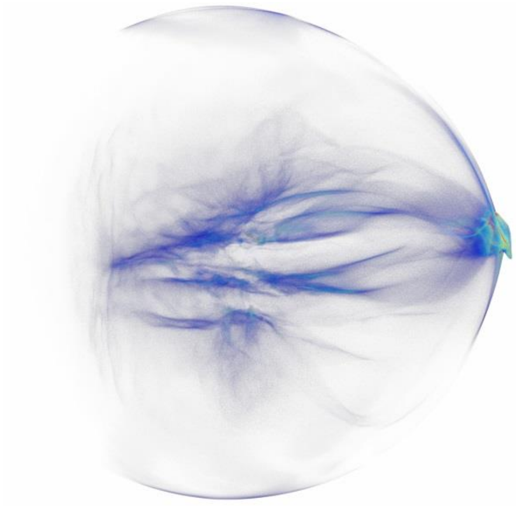
Volume: 2.7×10^9 cells

Macro-Particles: 7×10^8

Time: 300k iterations

Hardware: 192 Nvidia K80 GPUs (12GB)

Ion acceleration with lasers



© Huebl (HZDR), Matheson (ORNL)

Recently: 3D simulations of proton acceleration from ultrathin foils with **non-ideal laser pulse contrast** (*M. Garten, Tue 7pm*).

Includes: picosecond leading pulse edge, Bremsstrahlung photons

Volume: 6.4×10^{10} cells

Particles: 5×10^{10}

Time: 120k iterations

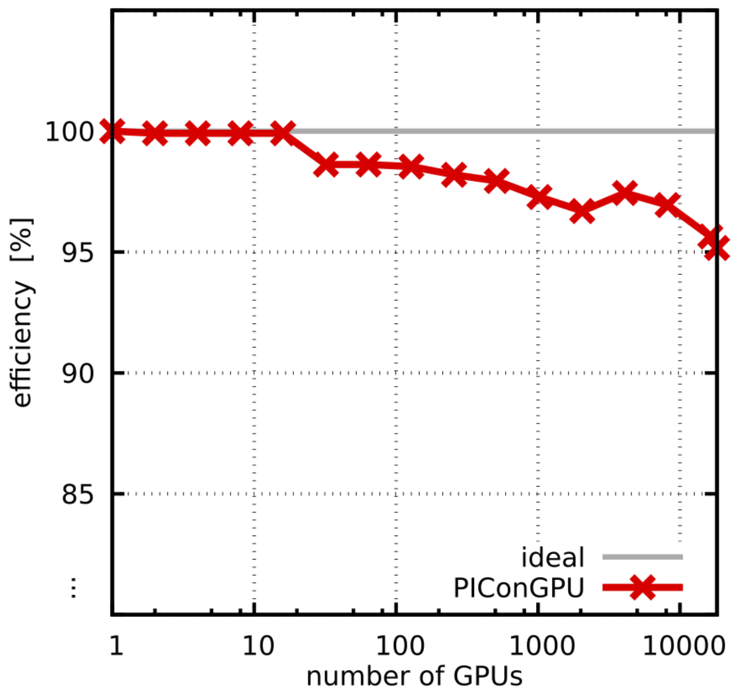
Hardware: 2400 Nvidia P100 GPUs (16GB)

Scale well, or waste precious compute time

PIConGPU uses all levels of parallelism and utilizes memory efficiently

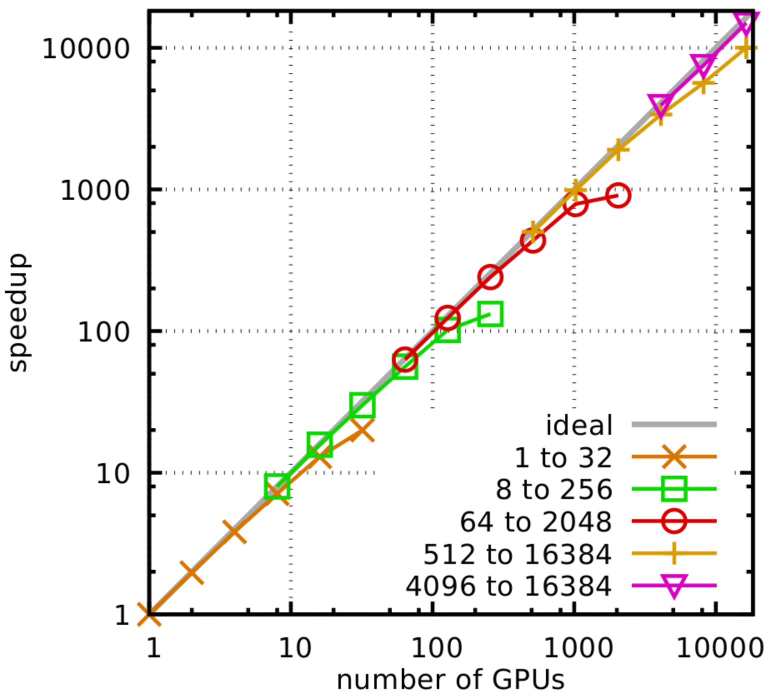
Weak scaling

Increase problem size with number of ranks but keep problem size per rank.
efficiency = t_{base} / $t_{solution}$



Strong scaling

Increase number of ranks but keep total problem size.
speedup = t_{base} / $t_{solution}$



PIConGPU

github.com/ComputationalRadiationPhysics/picongpu

picongpu.readthedocs.io

- **Open source**, fully relativistic, 3D3V, manycore, performance portable PIC code with a single code base

- Implements **various numerical schemes**, e.g.:
  > *Villasenor-Buneman*, *Esirkepov* and *ZigZag* current deposition
  > *NGP* (0th) to *P4S* (4th) macro particle shape orders
  > *Boris* and *Vay* particle pusher
  > *Yee* and *Lehe* field solver

- Available **self-consistent additions** to the PIC cyle, e.g.:
  > QED synchrotron radiation and Bremsstrahlung (photon emission)
  > *Thomas-Fermi* collisional ionization
  > *ADK* and *BSI* field ionization
  > Classical radiation reaction

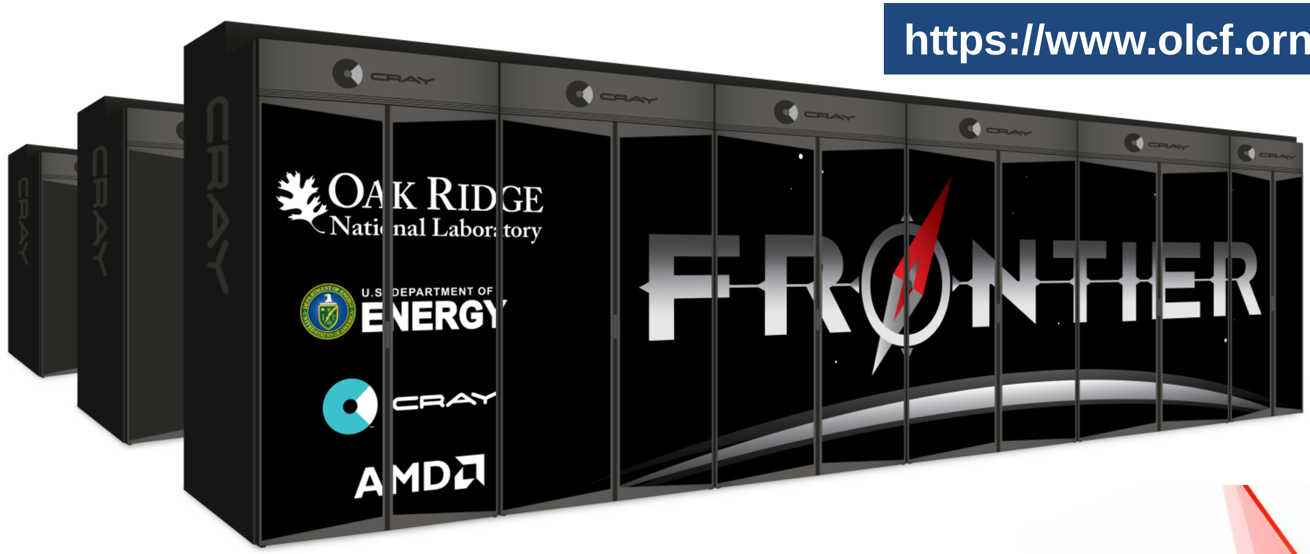
- **Tools and diagnostics**, e.g.:
  > Extensible selection of plugins for online analysis of particle and field data
  > In situ calculation of coherent and incoherent far field radiation
  > Scalable I/O for restarts and full output in openPMD with parallel *HDF5* and *ADIOS*

HELMHOLTZ ZENTRUM DRESDEN ROSSENDORF

PIConGPU will run on an Exascale machine by 2022

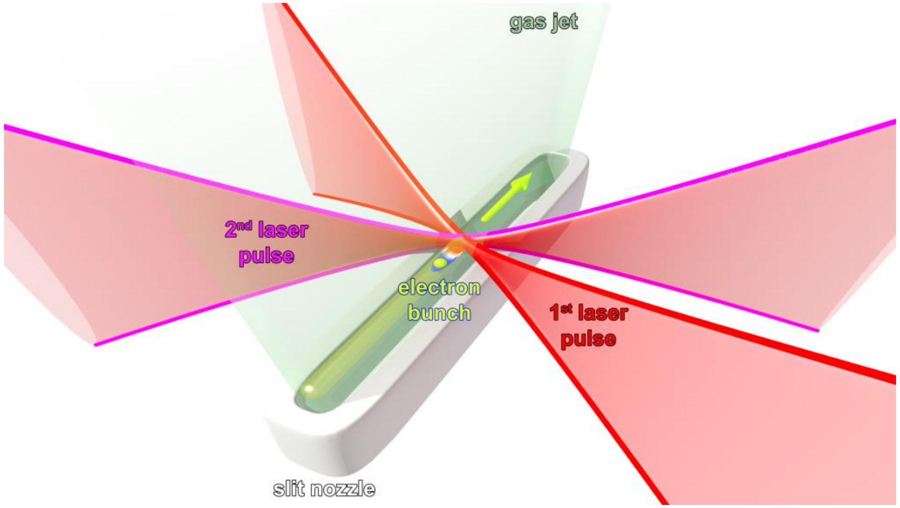
Collaboration of HZDR and Sunita Chandrasekaran‘s team (U. Delaware) selected to participate in the Frontier Center for Accelerated Application Readiness (CAAR) program.

https://www.olcf.ornl.gov/caar/Frontier-CAAR/



Specs: AMD EPYC CPUs and Radeon Instinct GPUs with 4:1 GPU-to-CPU ratio. Expected peak performance of 1.5 EFlop/s.

Physics case: LWFA in a TWEAC geometry utilizing two pulse-front tilted laser pulses obliquely incident to a slit nozzle for electron acceleration beyond 10 GeV.
(*A. Debus talk on Thursday, 16:40*)

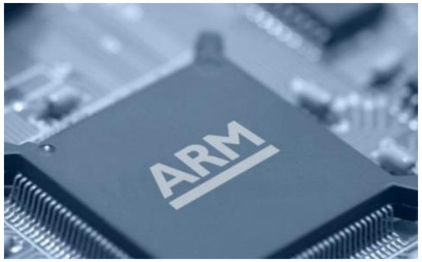


PIConGPU DRESDEN concept HZDR

Todays TOP10 HPCG systems reflect the multitude of modern compute architectures

HPCG List for June 2019

Rank	TOP500 Rank	System	Cores	Rmax (TFlop/s)	HPCG (TFlop/s)
1	1	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148,600.0	2925.75
2	2	**Sierra** - IBM Power System S922LC, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480	94,640.0	1795.67
3	20	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect , Fujitsu RIKEN Advanced Institute for Computational Science (AICS) Japan	705,024	10,510.0	602.74
4	7	**Trinity** - Cray XC40, Xeon E5-2698v3 16C 2.3GHz, Intel Xeon Phi 7250 68C 1.4GHz, Aries interconnect , Cray Inc. DOE/NNSA/LANL/SNL United States	979,072	20,158.7	546.12
5	8	**AI Bridging Cloud Infrastructure (ABCI)** - PRIMERGY CX2570 M4, Xeon Gold 6148 20C 2.4GHz, NVIDIA Tesla V100 SXM2, Infiniband EDR , Fujitsu National Institute of Advanced Industrial Science and Technology (AIST) Japan	391,680	19,880.0	508.85
6	6	**Piz Daint** - Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect , NVIDIA Tesla P100 , Cray Inc. Swiss National Supercomputing Centre (CSCS) Switzerland	387,872	21,230.0	496.98
7	3	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway , NRCPC National Supercomputing Center in Wuxi China	10,649,600	93,014.6	480.85

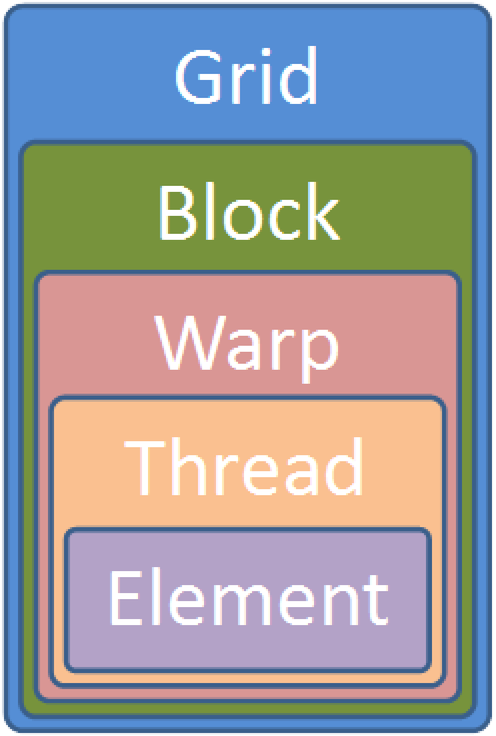


We aim for a single source, performance portable code to avert code branching!

PIConGPU DRESDEN concept HZDR

Handling the variety of different heterogeneous architectures with ALPAKA

ALPAKA: Abstraction Library for Parallel Kernel Acceleration



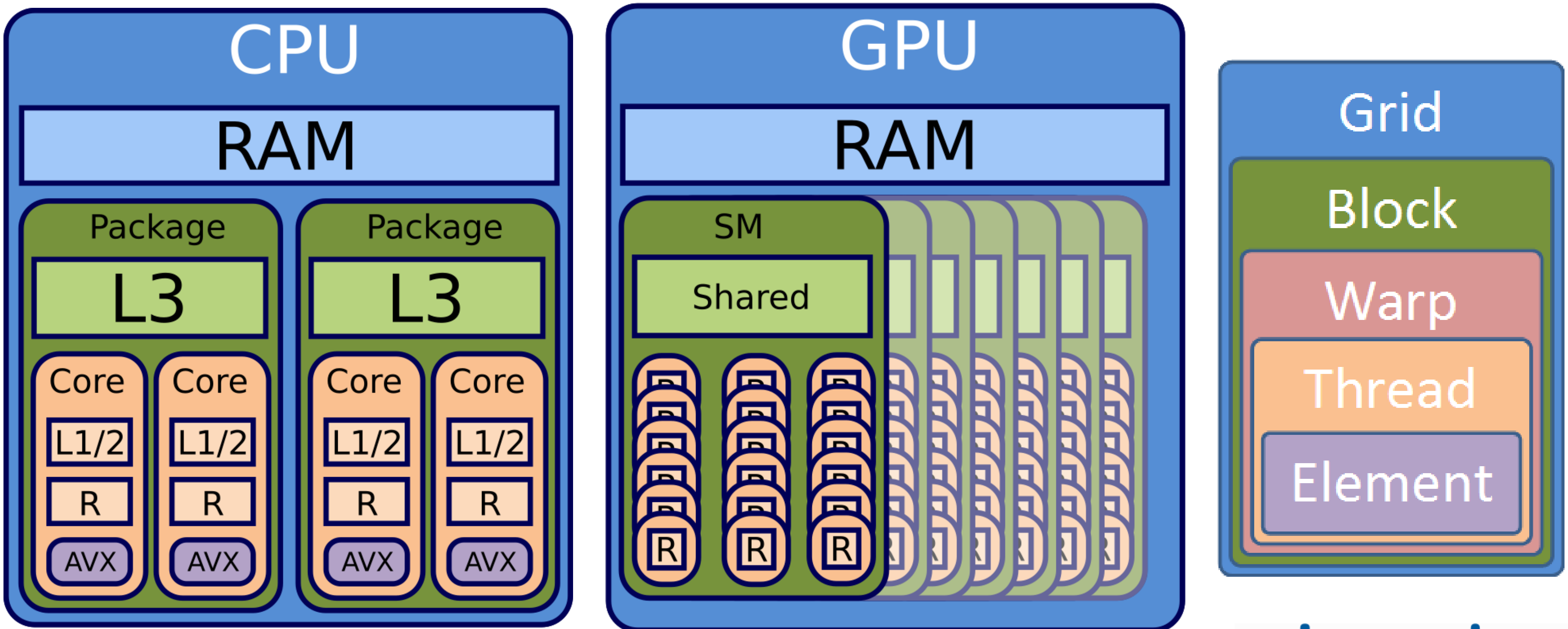
Hirarchy of parallelization offered by ALPAKA:

Grid	**whole parallel task**
Block	**fully independent part of the grid**
Warp	**group of synchronous threads**
Threads	**executed concurrently**
Elements	**sequential lock-step, vectorization**

Zenker E. et al. (2016) IEEE Xpress doi:10.1109/IPDPSW.2016.50 and ISC (2016) ISC High Performance doi:10.1007/978-3-319-46079-6_21; Matthes A. et al., (2017) ISC High Performance (pp 496-514) doi: 10.1007/978-3-319-67630-2_2

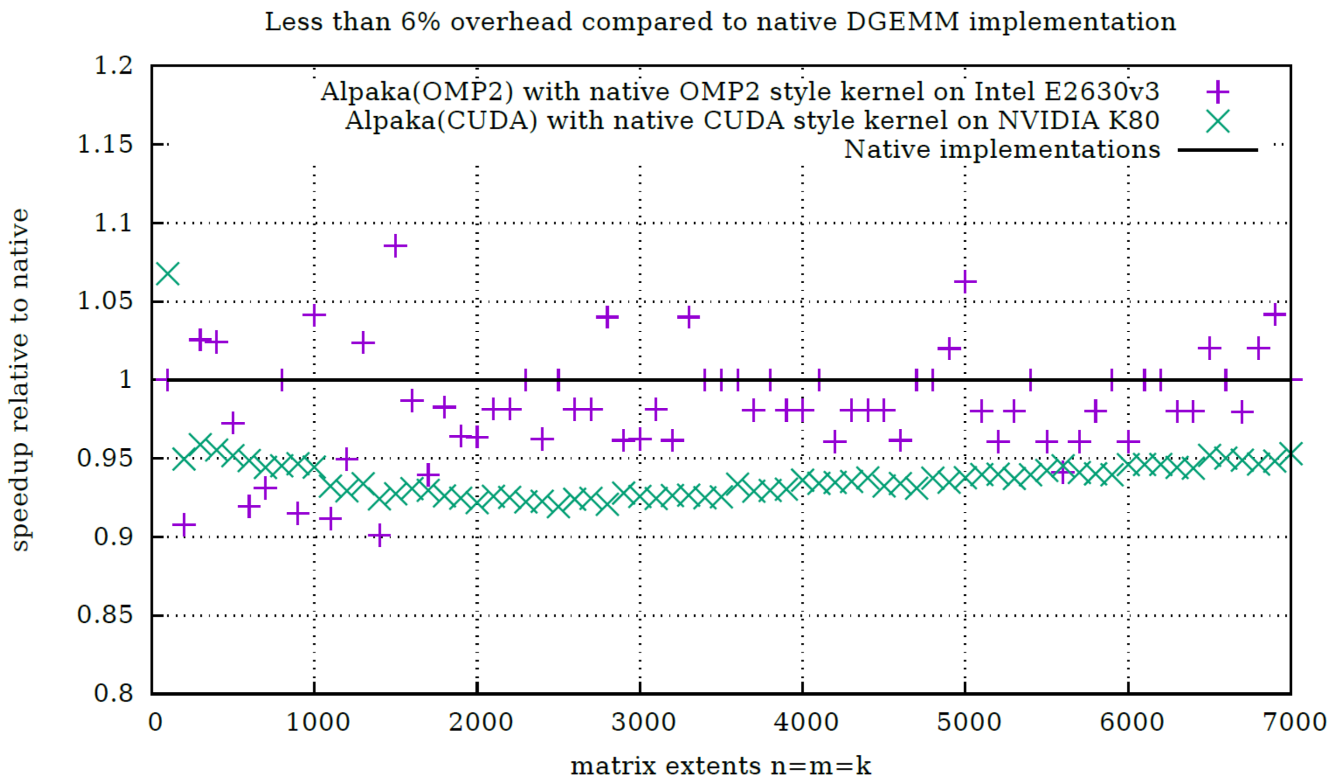
ALPAKA – one C++ interface to rule them all

Alpaka maps the parallel programming model to different architectures



Zenker E. et al. (2016) IEEE Xpress doi:10.1109/IPDPSW.2016.50 and ISC (2016) ISC High Performance doi:10.1007/978-3-319-46079-6_21; Matthes A. et al., (2017) ISC High Performance (pp 496-514) doi: 10.1007/978-3-319-67630-2_2

ALPAKA provides performance portability



(almost)
no overhead
compared to native
implementation

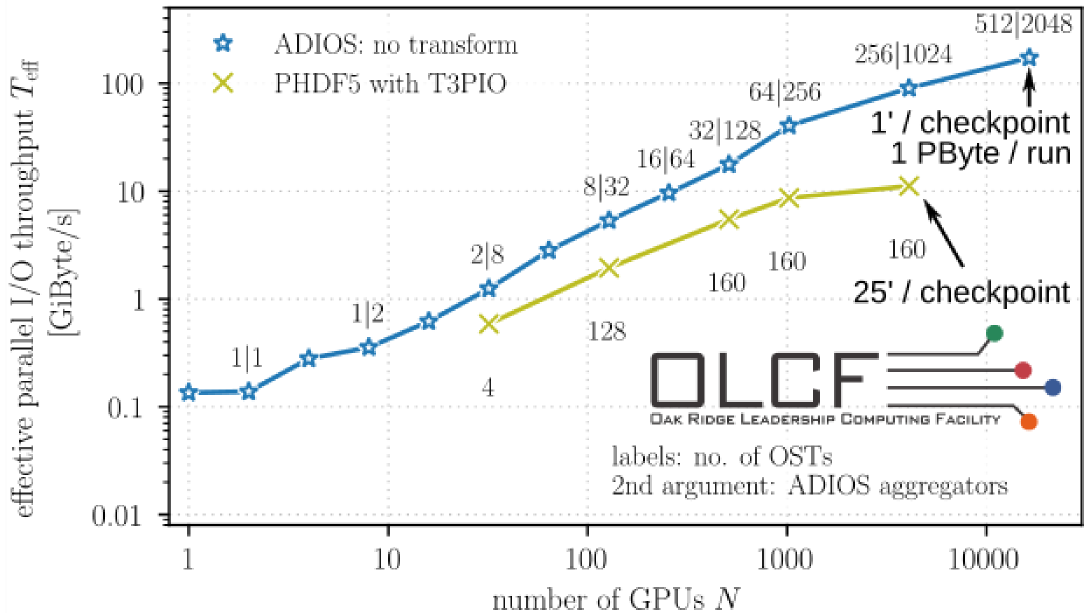
Zenker E. et al. (2016) IEEE Xpress doi:10.1109/IPDPSW.2016.50 and ISC (2016) ISC High Performance doi:10.1007/978-3-319-46079-6_21; Matthes A. et al., (2017) ISC High Performance (pp 496-514) doi: 10.1007/978-3-319-67630-2_2

Even if your computations scale, do not underestimate I/O

Data production on compute device
NVIDIA K20X Kepler GPU
6GB GPU RAM updated by 10 iterations / s
⇨ **60GB/s data creation rate / Node (GPU)**

In-node memory copy throughput
PCI-Express 3.x
15.75GB/s bus speed
⇨ **25% creation rate**

File write out
Lustre Filesystem
55MB/s data throughput per node
⇨ **0.1% creation rate**

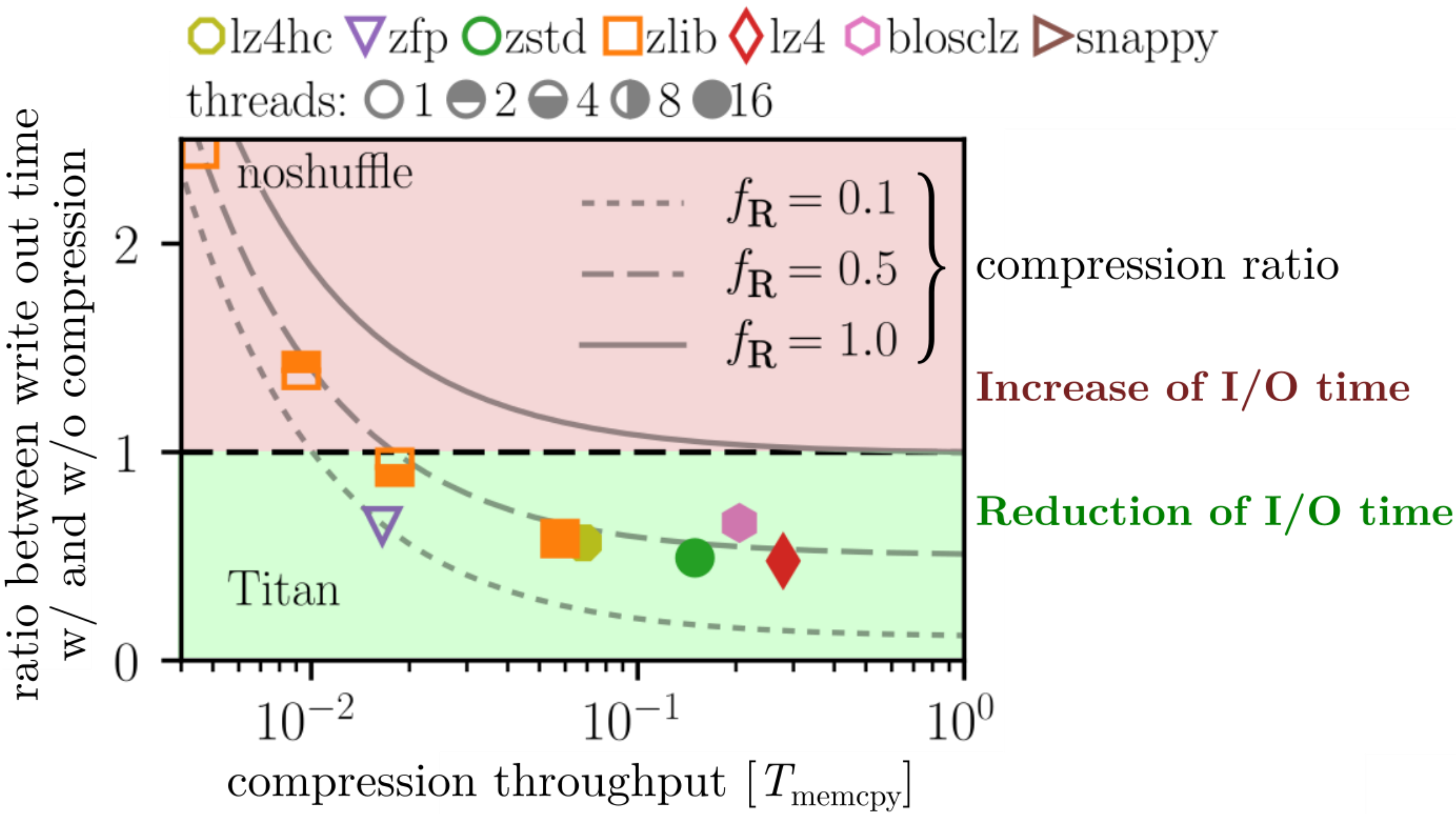


> **Technical limit for I/O per node**

> **But global re-ordering and synchronization (with parallel HDF5) results in significant slow down**

PIConGPU DRESDEN concept HZDR

Compressing data before write out does not help per se to cope with the technical I/O limitation

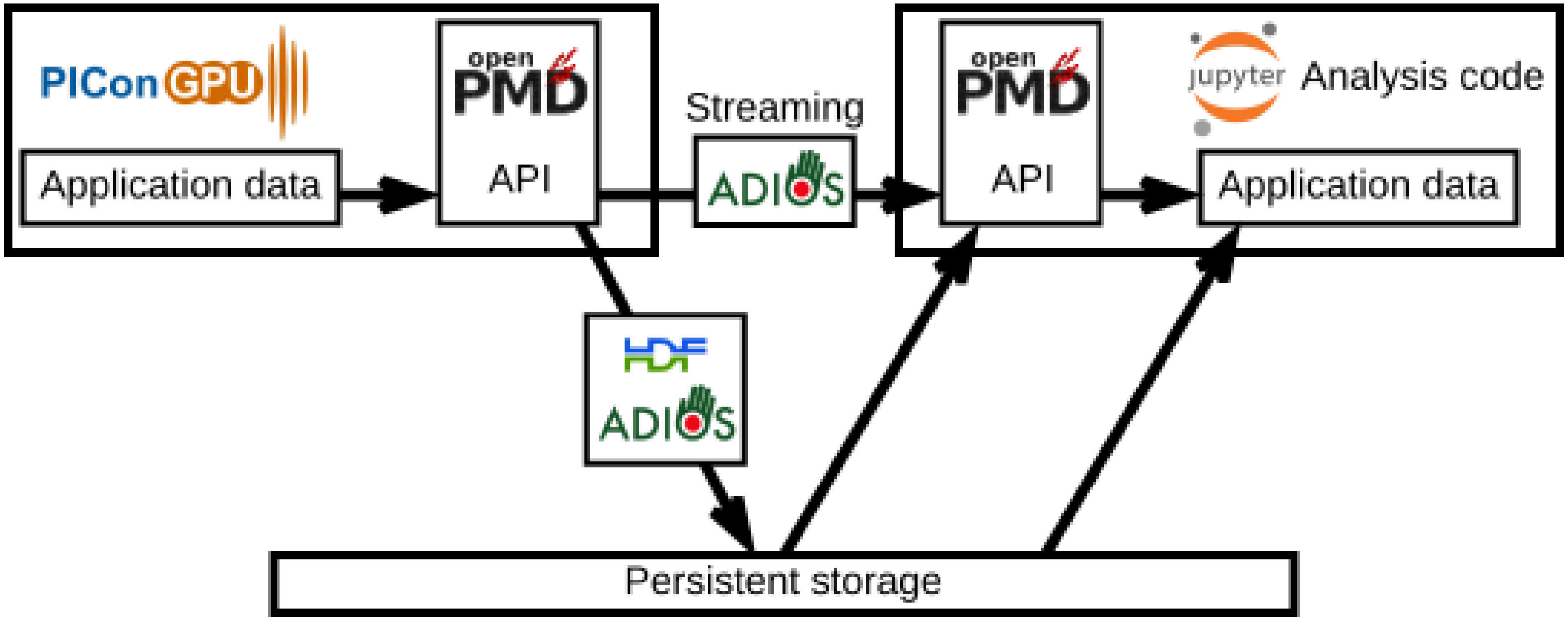


A. Huebl, *Lect. Notes Comput. Sci.* (2017) , doi:10.1007/978-3-319-67630-2_36

But it is not practical to store 1 PB / simulation over a whole campaign …

… MAKE USE OF ONLINE ANALYSIS TO STORE REDUCED DATA

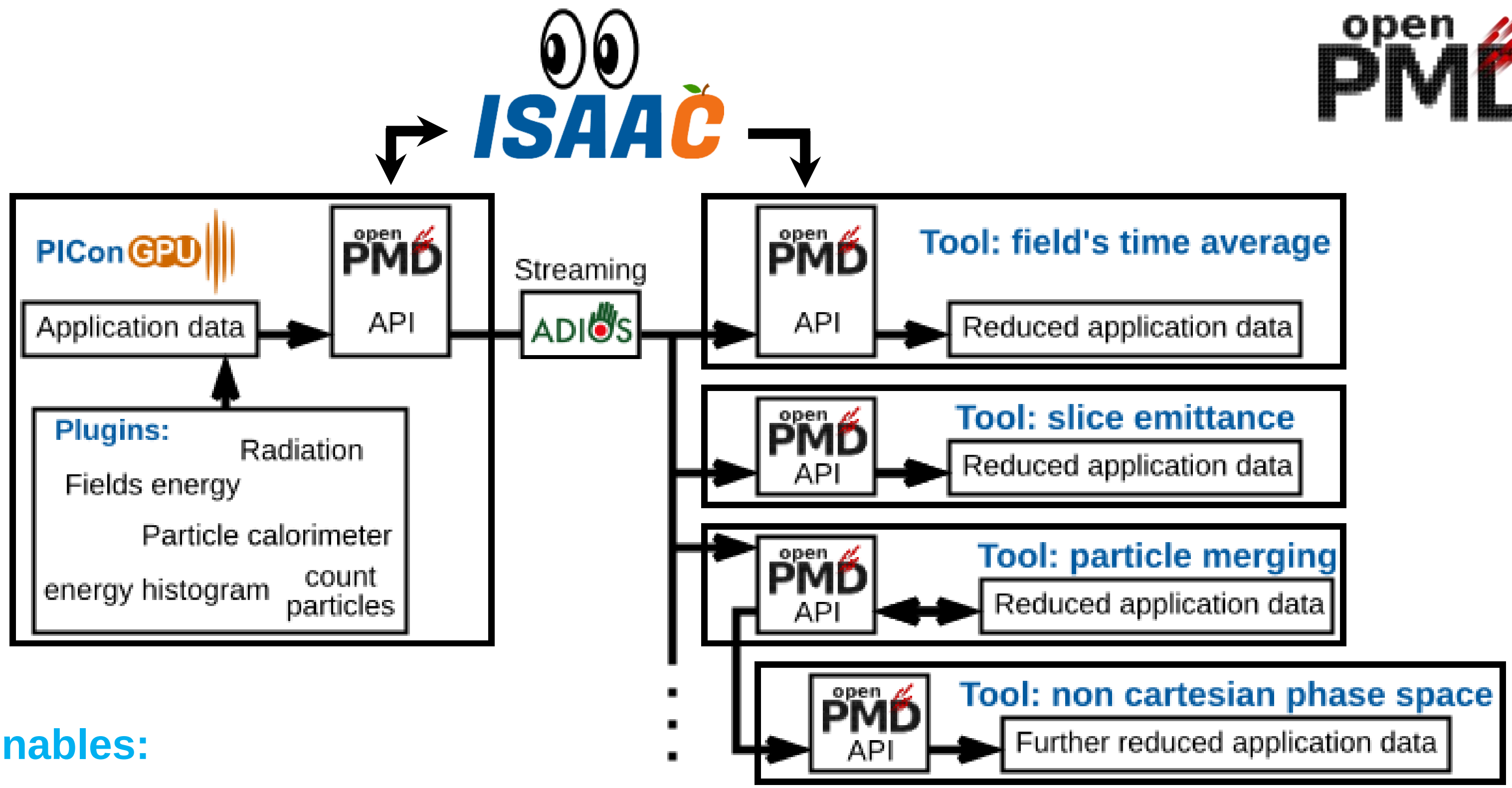
Online data analysis via openPMD-api + ADIOS2



Instead of two plugins talking directly to the I/O library —using openPMD-api→

- > Flexibly interchange and add backends
- > Unified semantics for data movement
- > Standardized description of data (**openPMD**)

Online data analysis via openPMD-api + ADIOS2



Streaming enables:

- > Extension of plugin-based tightly coupled analysis by loosely coupled, **standalone analysis tools**
- > Running tools concurrently on separate hardware for **online analysis**
- > **Online selection** of data to analyze
- > **Independent development** of tools
- > **Reduction of time to prototype** online analysis

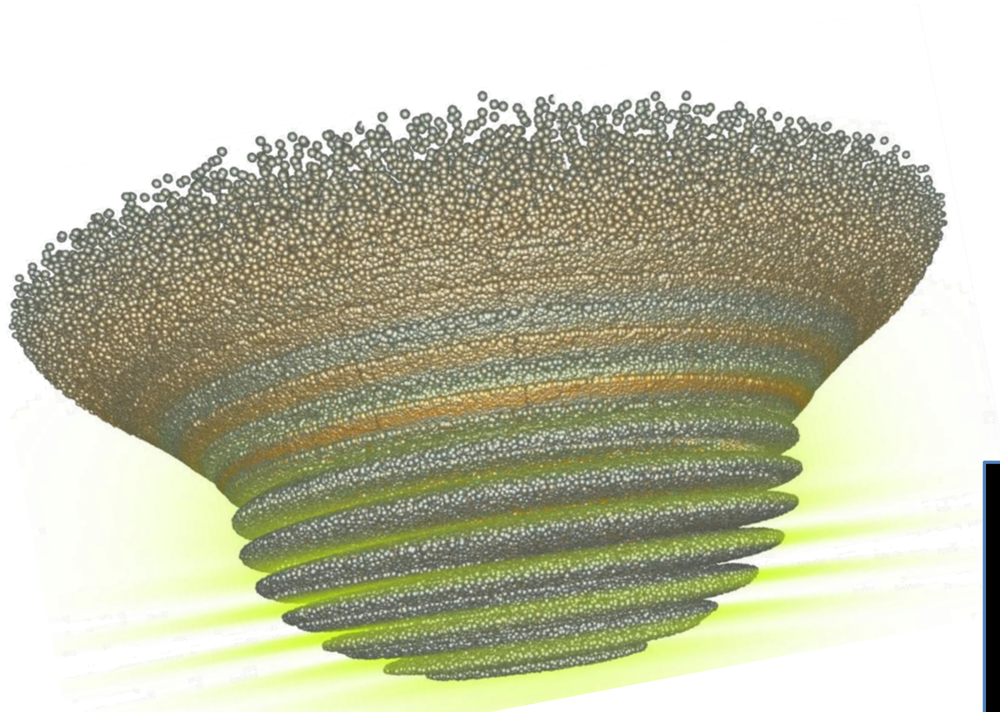
github.com/openPMD

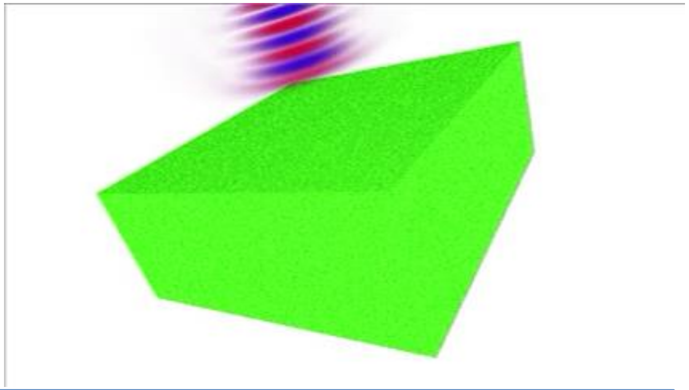
www.openPMD.org

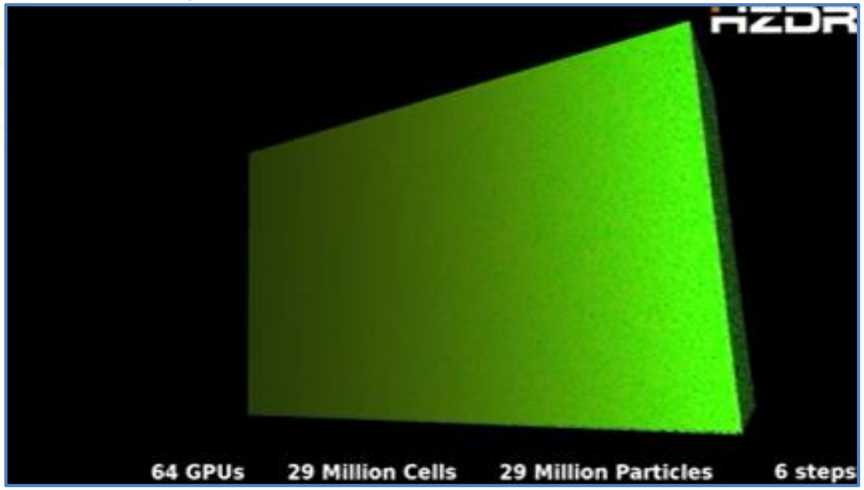
openPMD-api.readthedocs.io

In-situ live visualization

ISAAC





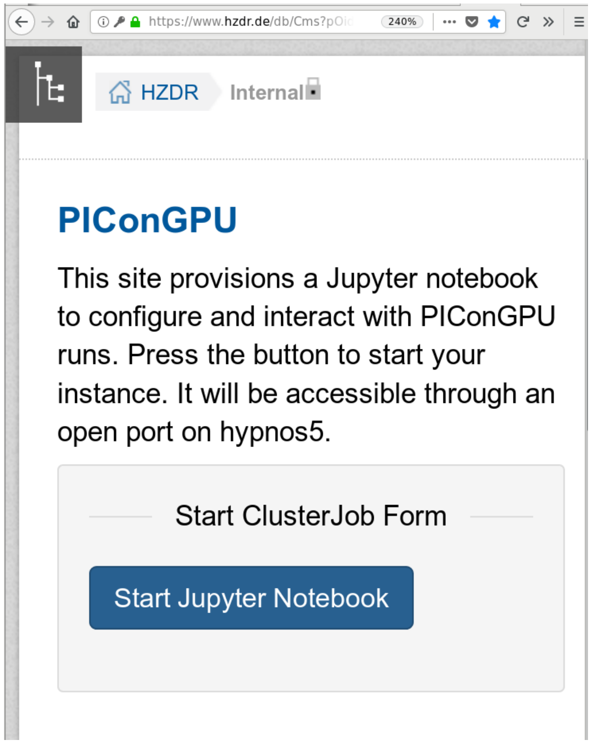


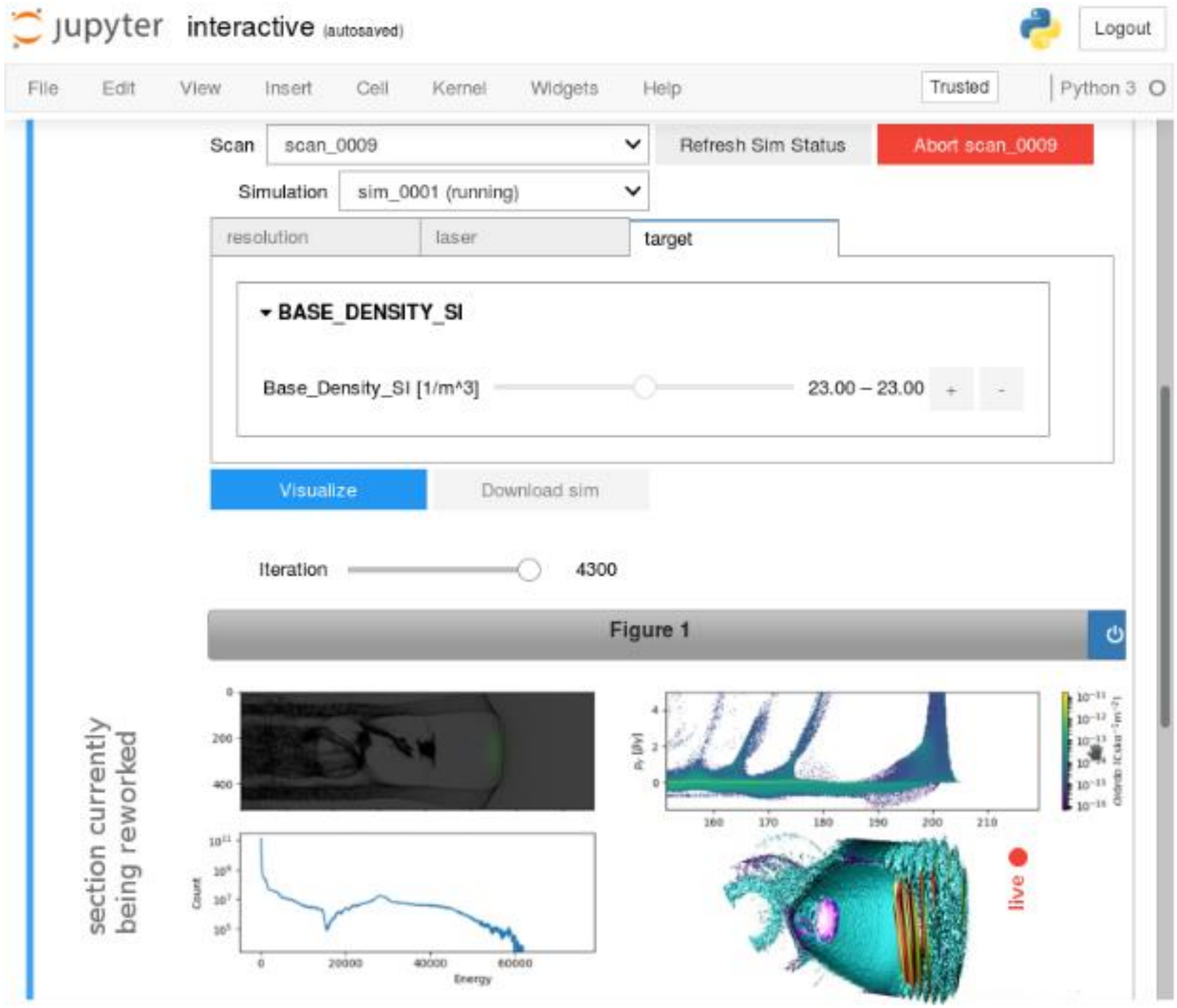
> Tbyte/s throughput
> 10^{11} particles and 10^{9} cells
> few % overhead
> 10fps

Matthes A. et al. (2016) SUPERFRI Vol. 3 No. 4
doi:10.14529/jsfi160403

Towards simulation as a service

Virtual experiment setup, control and analysis with Jupyter notebooks





Goal: Run parameter scans, control simulation output online and analyze results within a Jupyter Notebook and w/o prior knowledge in PIConGPU simulation setup.

Summary

PIConGPU is

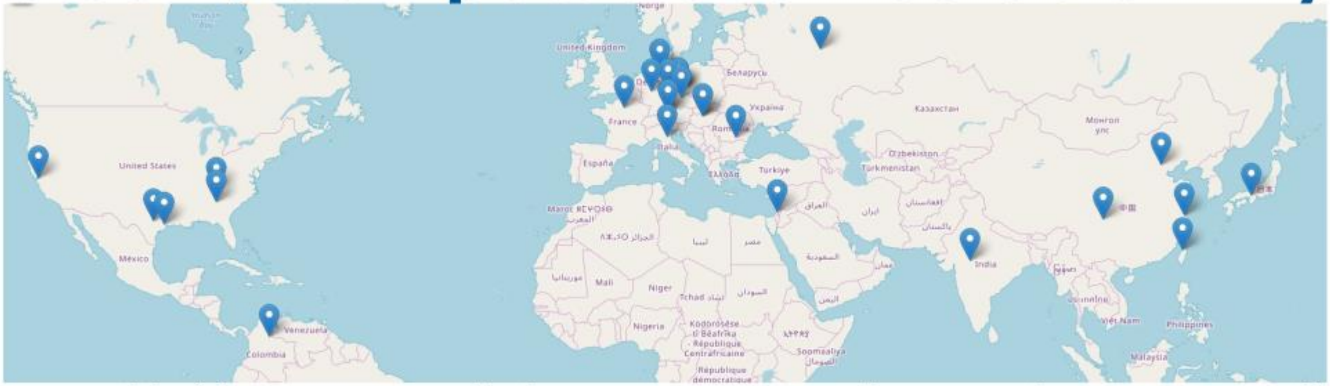
> open source and available from *github.com/ComputationalRadiationPhysics/picongpu*

> scalable in computations, performance portable and single source through ALPAKA

> scalable in I/O through ADIOS

> enabling online, loosely coupled data analysis through openPMD-api and ADIOS2 soon

picongpu.hzdr.de
github.com/ComputationalRadiationPhysics



This project has been enabled by many people in open-source and open-science communities. Great thanks to the communities and developers of: PIConGPU, PMacc, Alpaka, ROOT/Cling, Jupyter, the SciPy ecosystem, ADIOS, HDF5, Boost, CMake, openPMD, Spack, ...

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-Förderkennzeichen 01IH16006C).
This project has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 654220.

PIConGPU DRESDEN concept HZDR