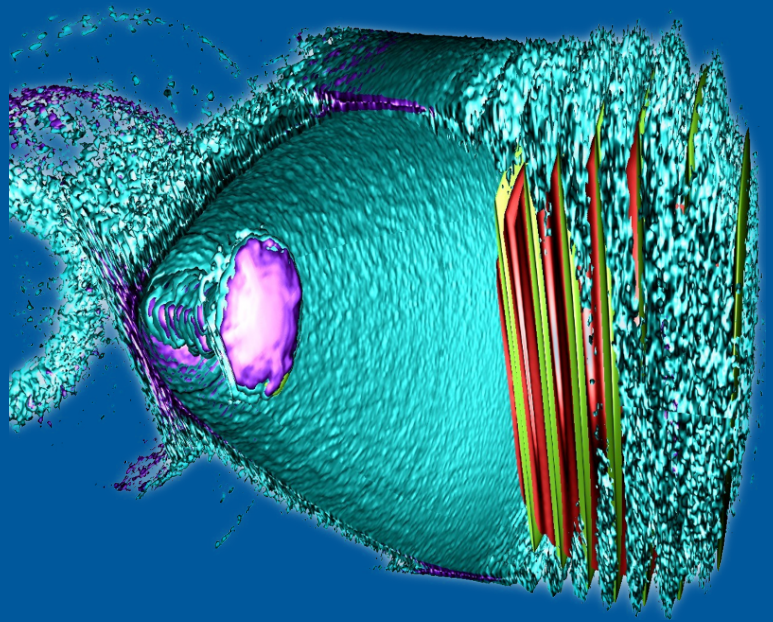
HPC Start-to-End Simulations & Digital Twins



A. Huebl[1,2], **R. Widera**[1], **H. Meissner**[1], **J. Kelling**[1], **G. Juckeland**[1], **and M. Bussmann**[1]

[1] Helmholtz-Zentrum Dresden - Rossendorf

[2] Technische Universität Dresden

Matter meets Information

Common Challenges and Perspectives

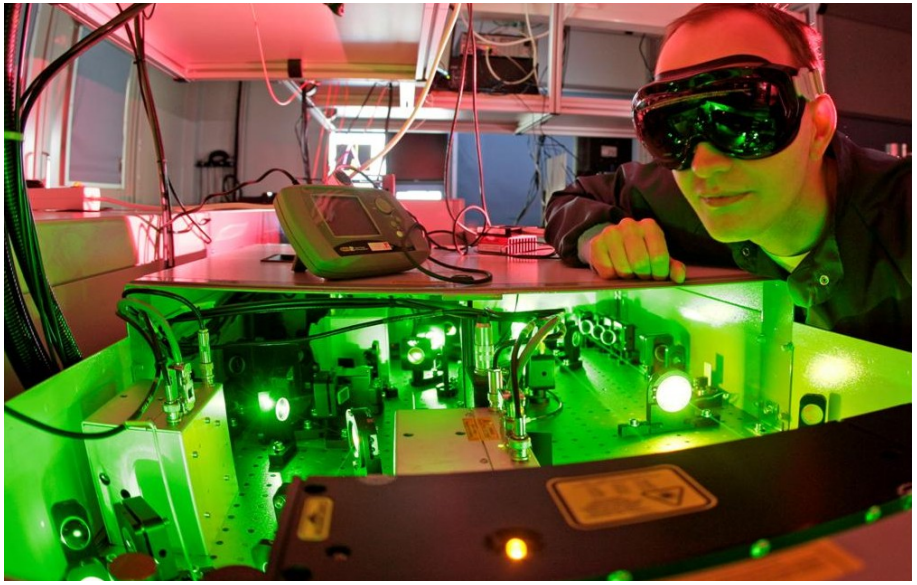
Hamburg, January 14th 2019

Application-Driven

Start-to-End Simulations are Data-Driven

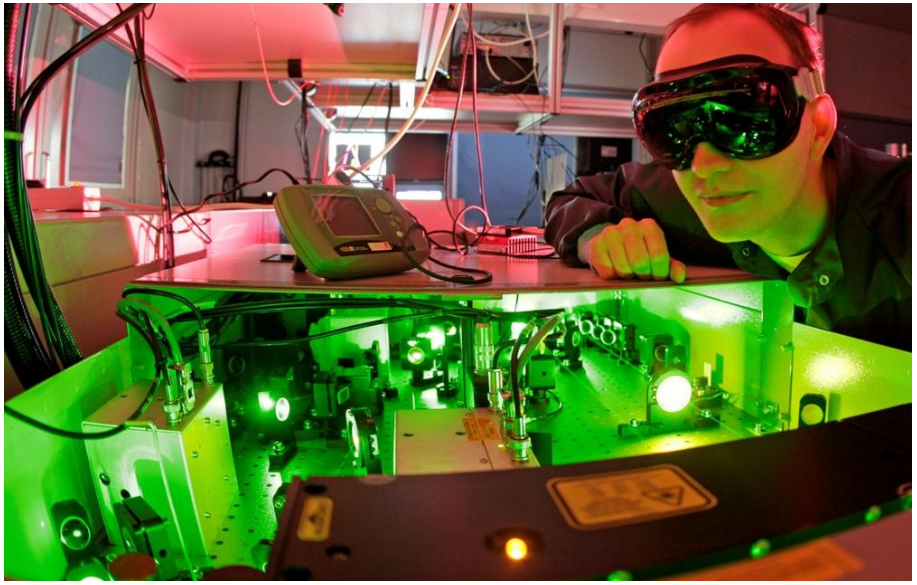
Application- & Data-Driven
Start-to-End Modeling

- Laser-Plasma Accelerators

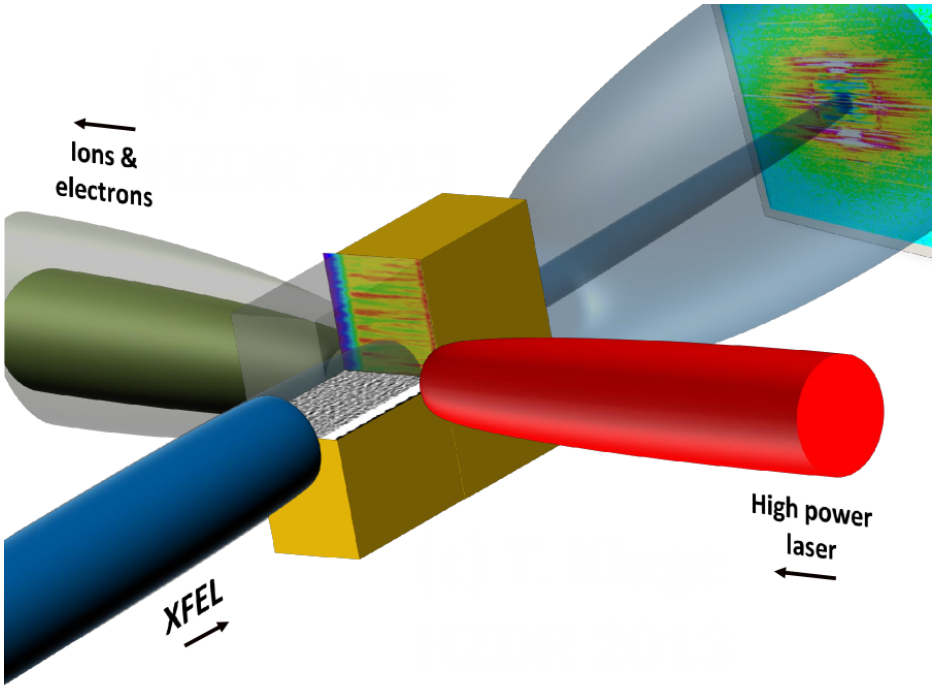


Application- & Data-Driven
Start-to-End Modeling

- Laser-Plasma Accelerators



- Light-Sources
  - EU XFEL



T. Kluge, et al., Phys. Rev. X **8**, 031068 (2018)

Particle-in-Cell Simulations

PIConGPU

- Ab initio, electro-magnetic plasmas

Particle-in-Cell Simulations

PIConGPU

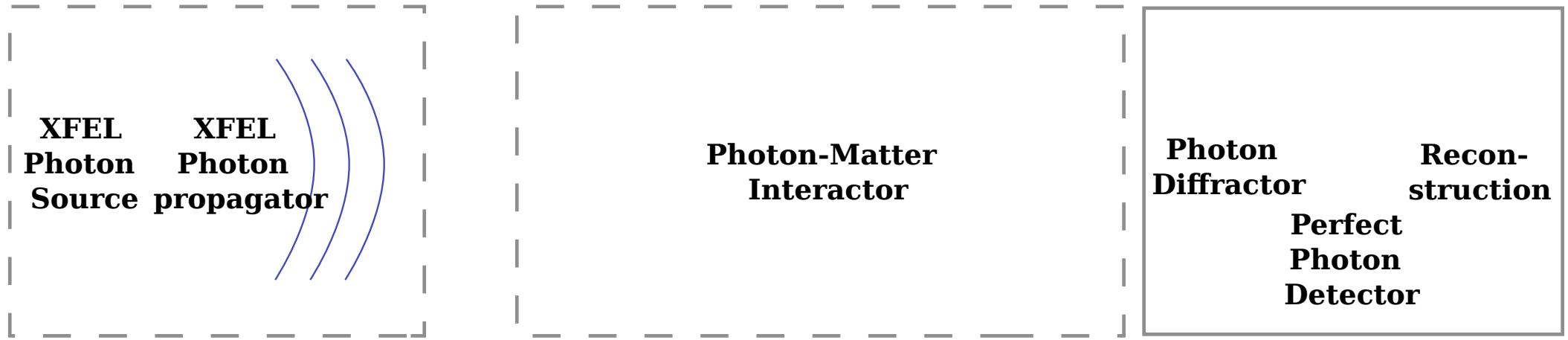
- Ab initio, electro-magnetic plasmas
- Scaling to the full-size of Titan & Piz Daint

Particle-in-Cell Simulations

PIConGPU

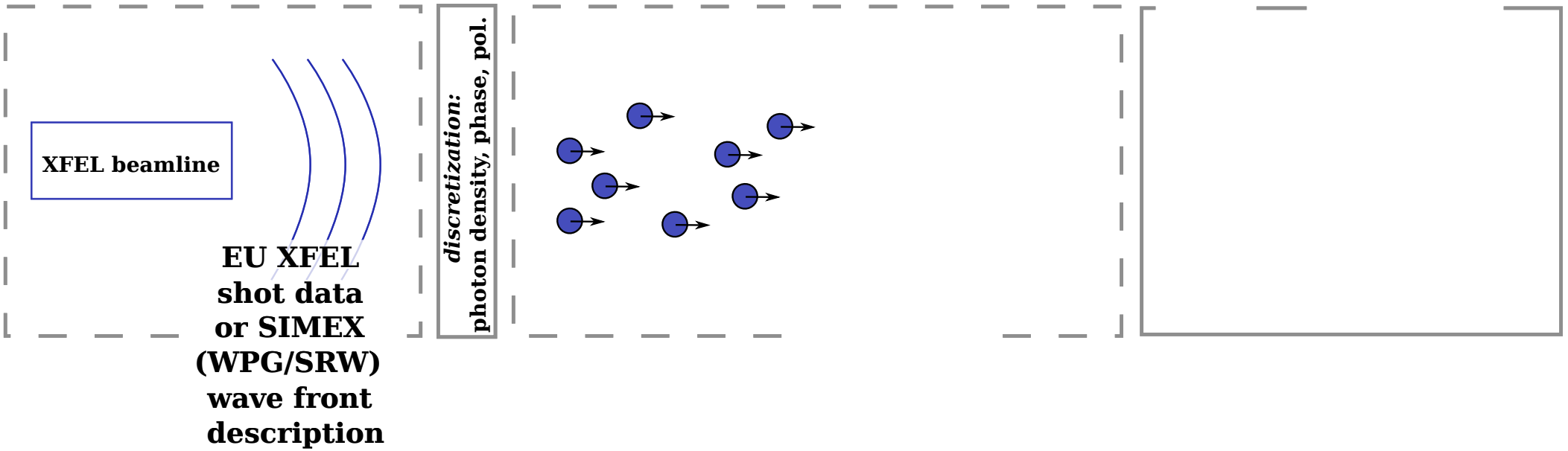
- Ab initio, electro-magnetic plasmas
- Scaling to the full-size of Titan & Piz Daint
- Gordon Bell finalist 2013

SIMEX Platform
Start-to-End Modeling



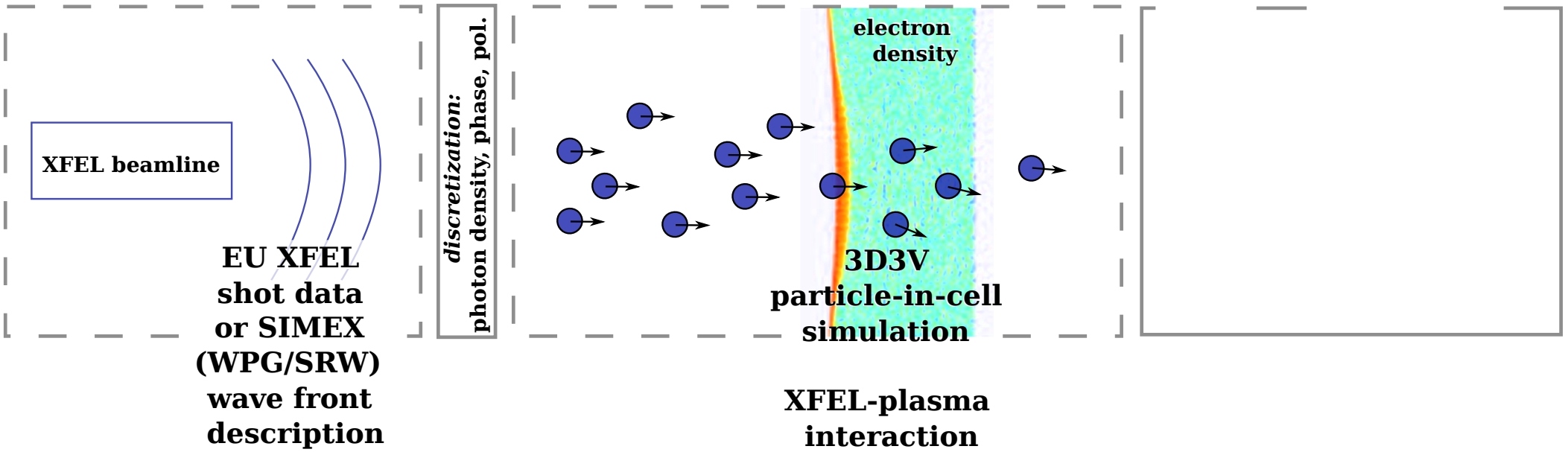
C. Fortmann-Grote, et al., Advances in X-ray Free-Electron Lasers Instrumentation **IV**, 10237 (2017)

SIMEX Platform
Start-to-End Modeling



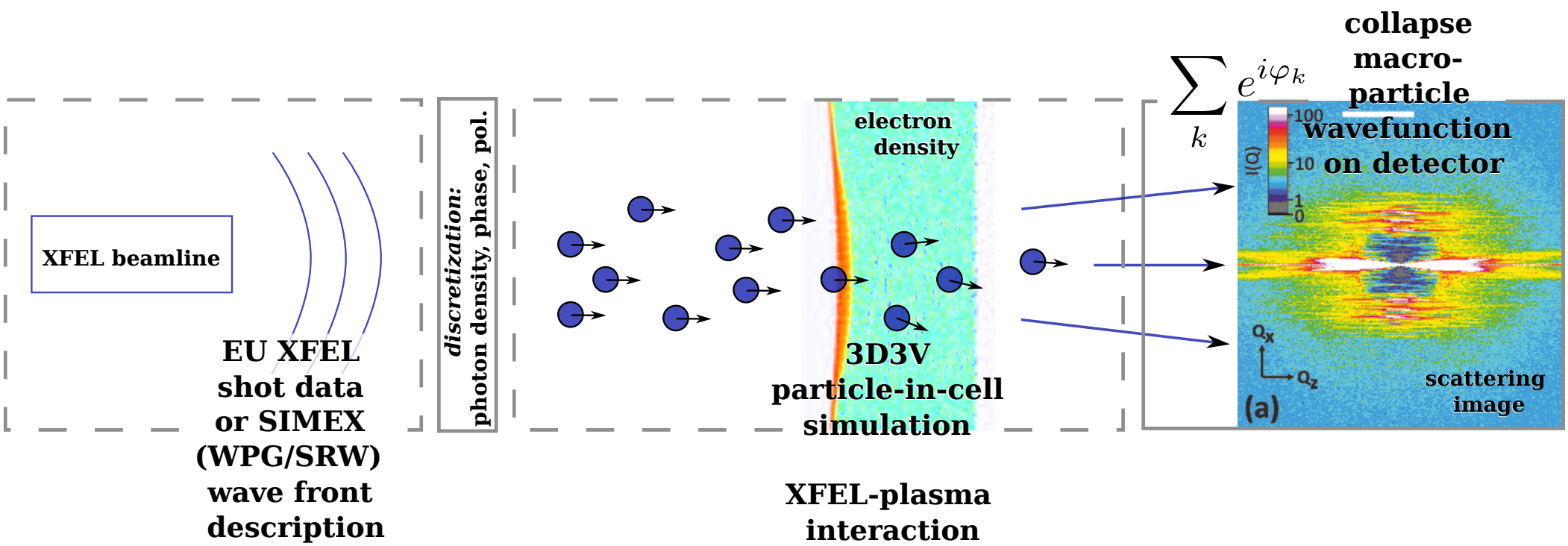
C. Fortmann-Grote, et al., Advances in X-ray Free-Electron Lasers Instrumentation **IV**, 10237 (2017)

SIMEX Platform
Start-to-End Modeling



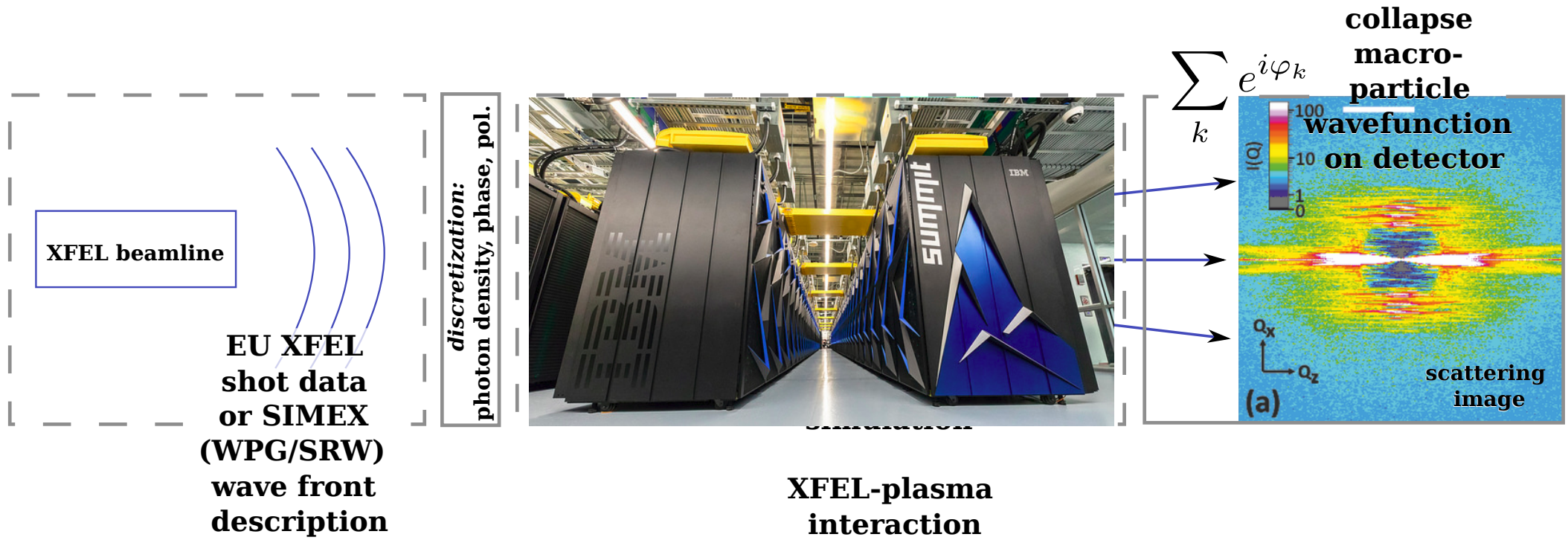
C. Fortmann-Grote, et al., Advances in X-ray Free-Electron Lasers Instrumentation **IV**, 10237 (2017)

SIMEX Platform
Start-to-End Modeling



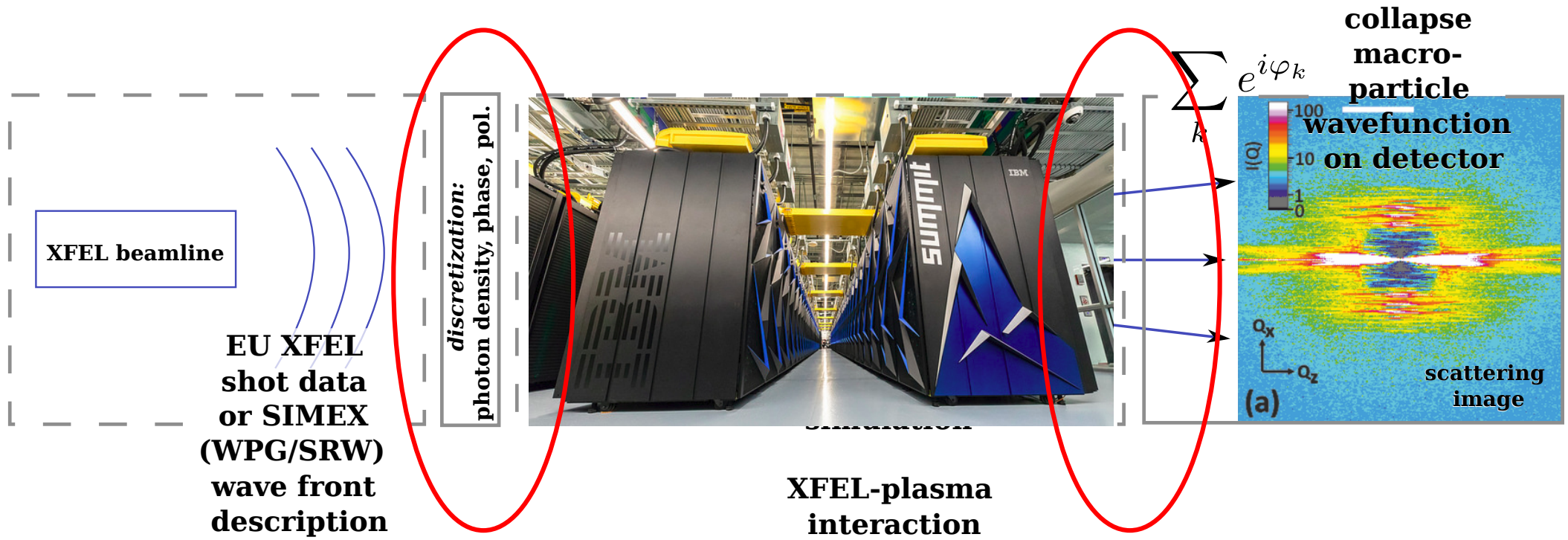
C. Fortmann-Grote, et al., Advances in X-ray Free-Electron Lasers Instrumentation **IV**, 10237 (2017)

SIMEX Platform
Start-to-End Modeling



C. Fortmann-Grote, et al., Advances in X-ray Free-Electron Lasers Instrumentation **IV**, 10237 (2017)

SIMEX Platform
Start-to-End Modeling

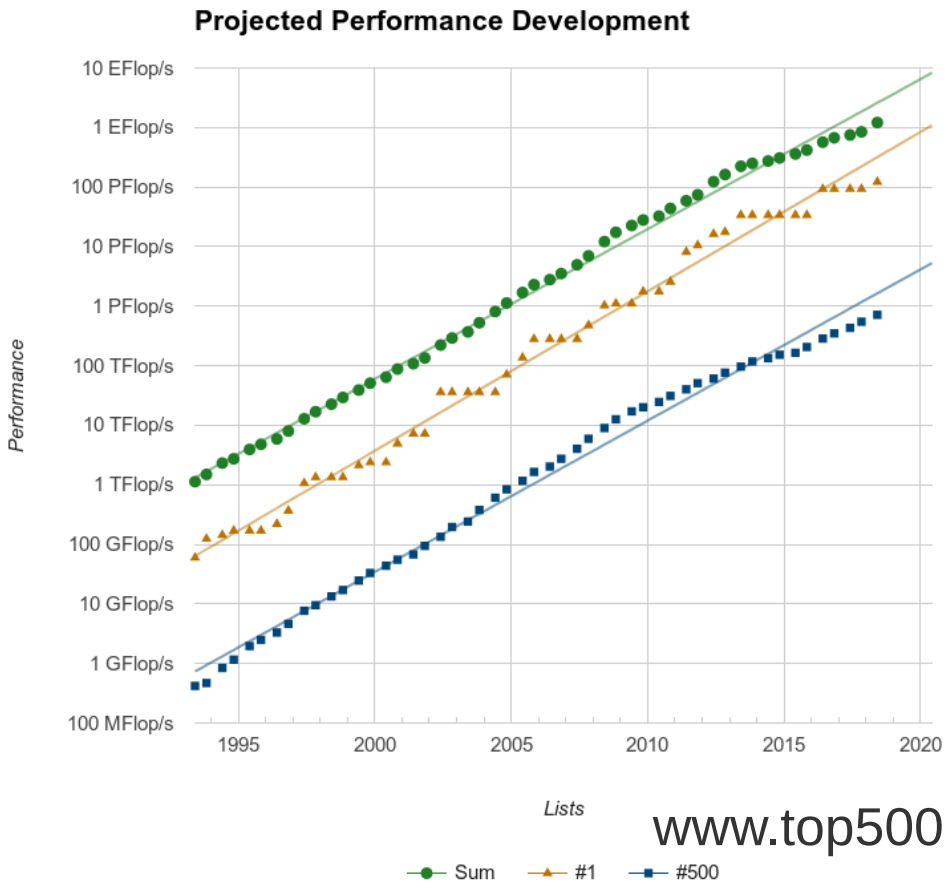


C. Fortmann-Grote, et al., Advances in X-ray Free-Electron Lasers Instrumentation **IV**, 10237 (2017)

HPC Means Massively Scalable
Challenges towards Exascale

Exascale Computing Challenges
Chances for simulations and data processing

- Today: 143 PFlop/s (Summit)
- 2020/'21: **1 EFlop/s**

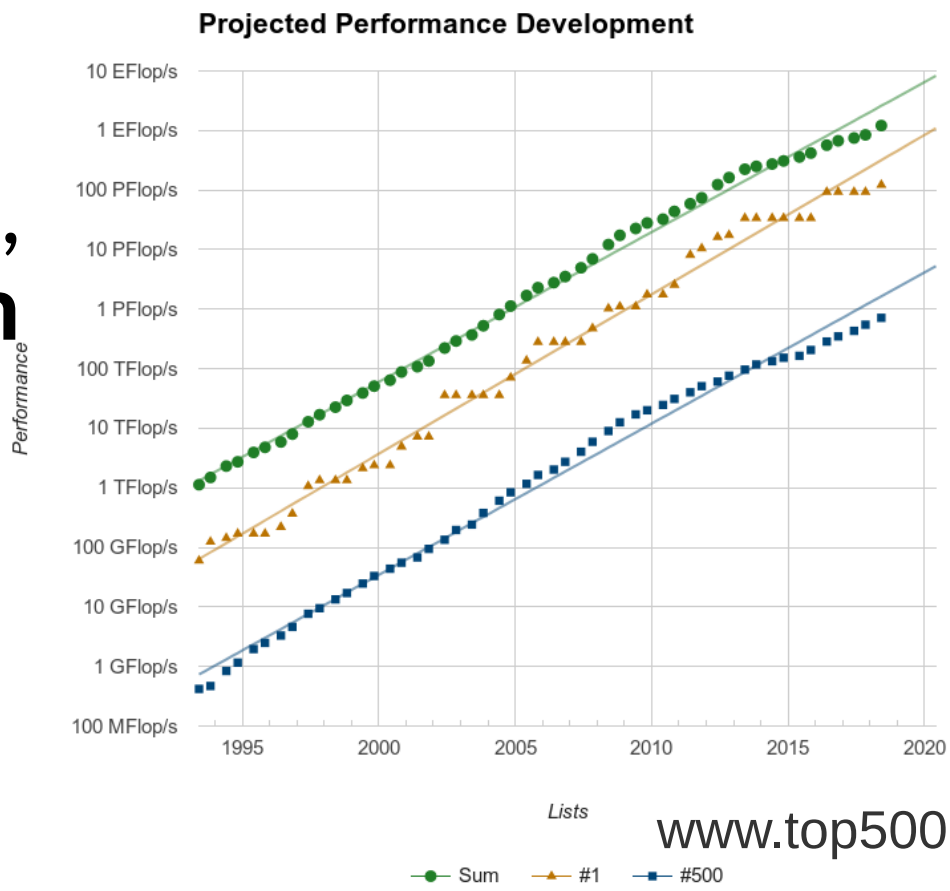


www.top500.org

Exascale Computing Challenges
Chances for simulations and data processing

- Today: 143 PFlop/s (Summit)
- 2020/'21: **1 EFlop/s**

- gains through massive, **multi-level parallelism**



www.top500.org

DRESDEN concept

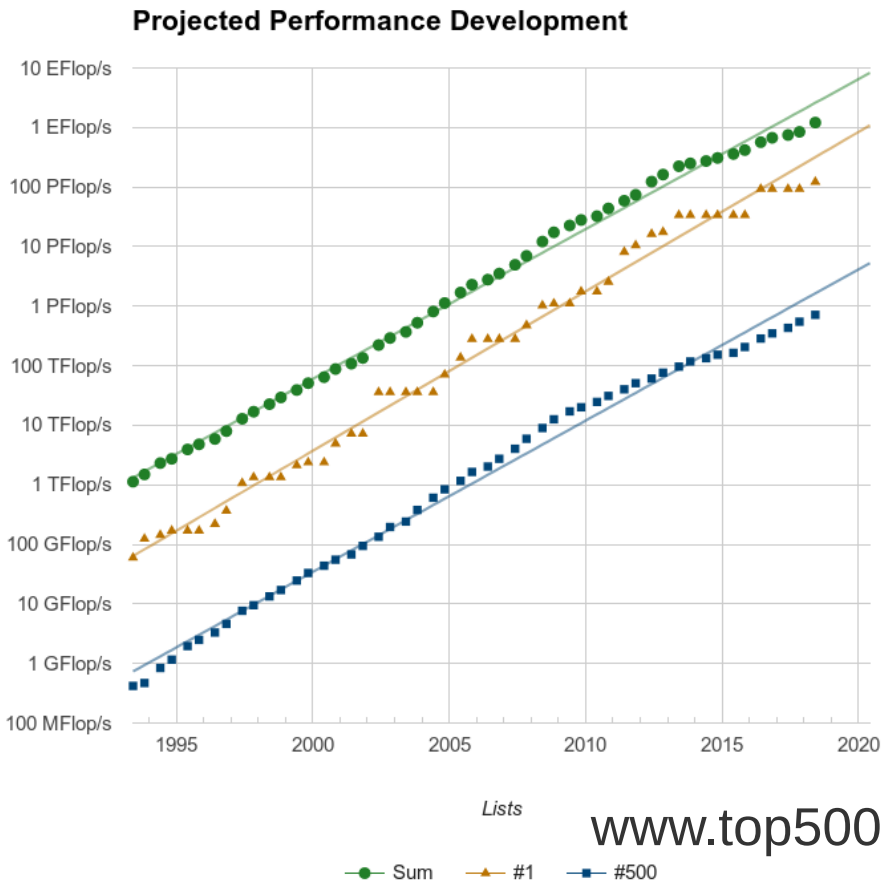
HZDR

Exascale Computing Challenges
Chances for simulations and data processing

- Today: 143 PFlop/s (Summit)
- 2020/'21: **1 EFlop/s**

- gains through massive, **multi-level parallelism**

- no *faster* solutions, but



www.top500.org

DRESDEN concept

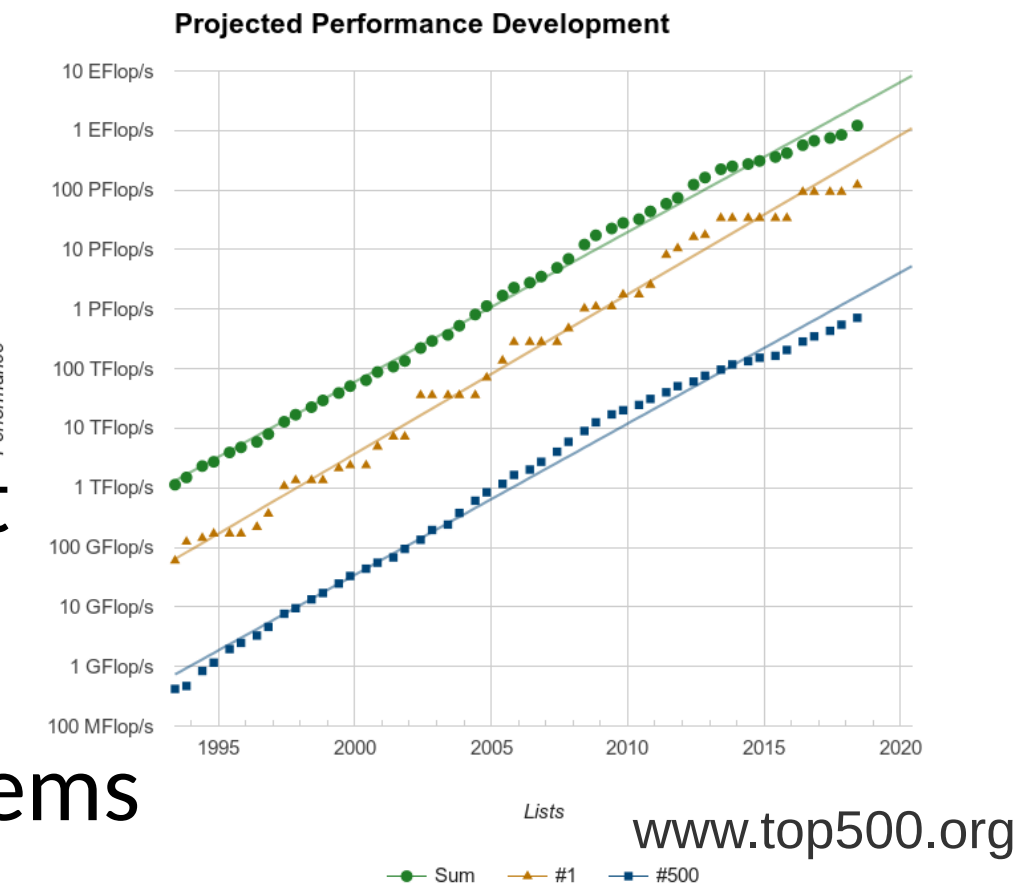
HZDR

Exascale Computing Challenges
Chances for simulations and data processing

- Today: 143 PFlop/s (Summit)
- 2020/'21: **1 EFlop/s**

- gains through massive, **multi-level parallelism**

- no *faster* solutions, but
  - larger problems
  - more problems
  - more complex problems



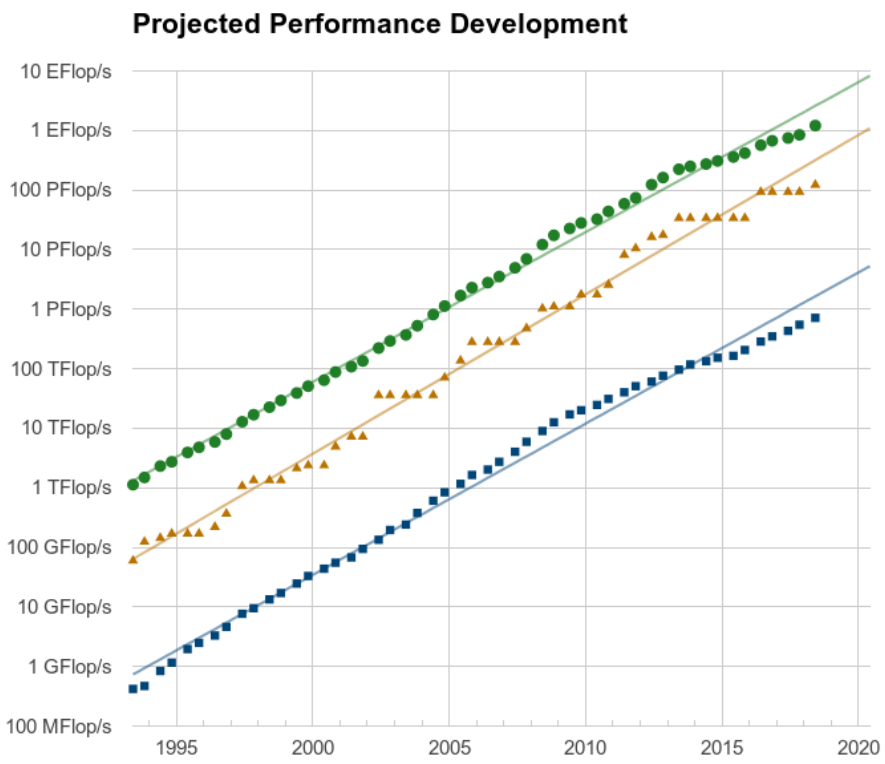
Exascale Computing Challenges
Chances for simulations and data processing

- Today: 143 PFlop/s (Summit)
- 2020/'21: **1 EFlop/s**

- gains through massive, **multi-level parallelism**

- no *faster* solutions, but
  - larger problems
  - more problems
  - more complex problems

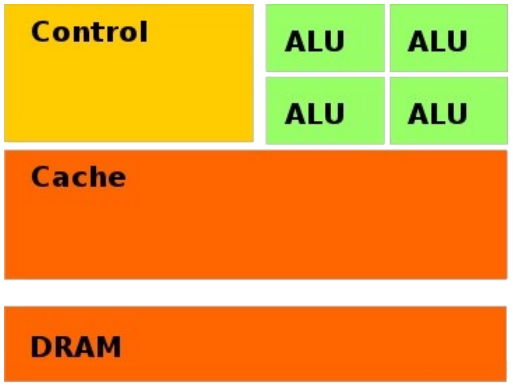
  in the ***same time to solution***



Parallel Architectures

Simplified, general purpose processors today

CPU

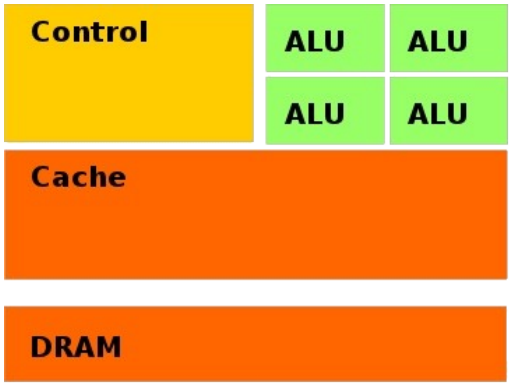


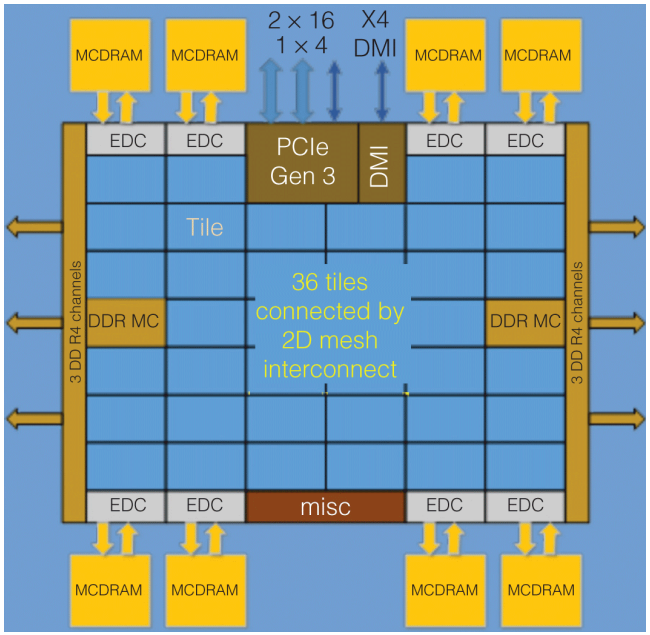
multi-core

Parallel Architectures

Simplified, general purpose processors today

CPU



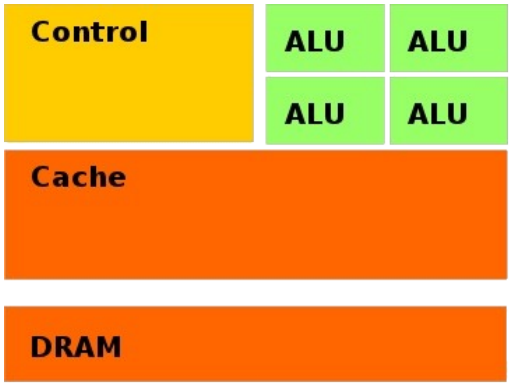


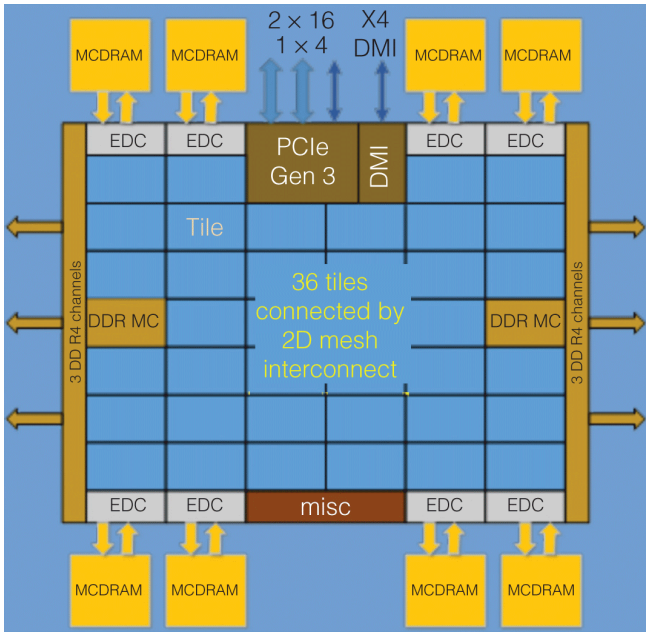
multi-core

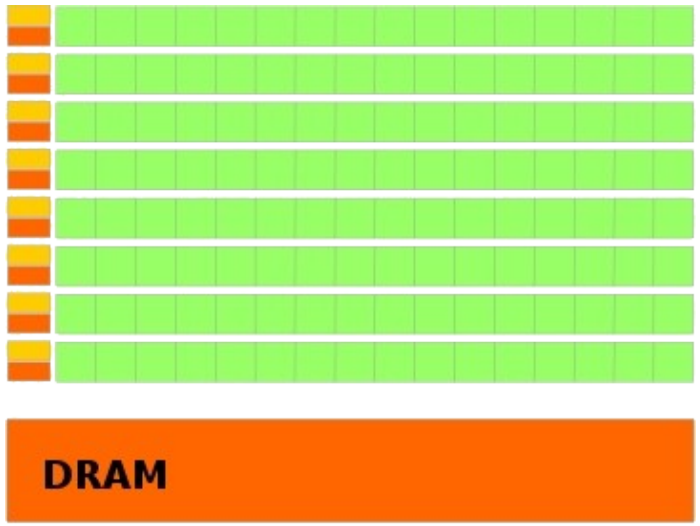
Parallel Architectures

Simplified, general purpose processors today

CPU → **GPGPU**



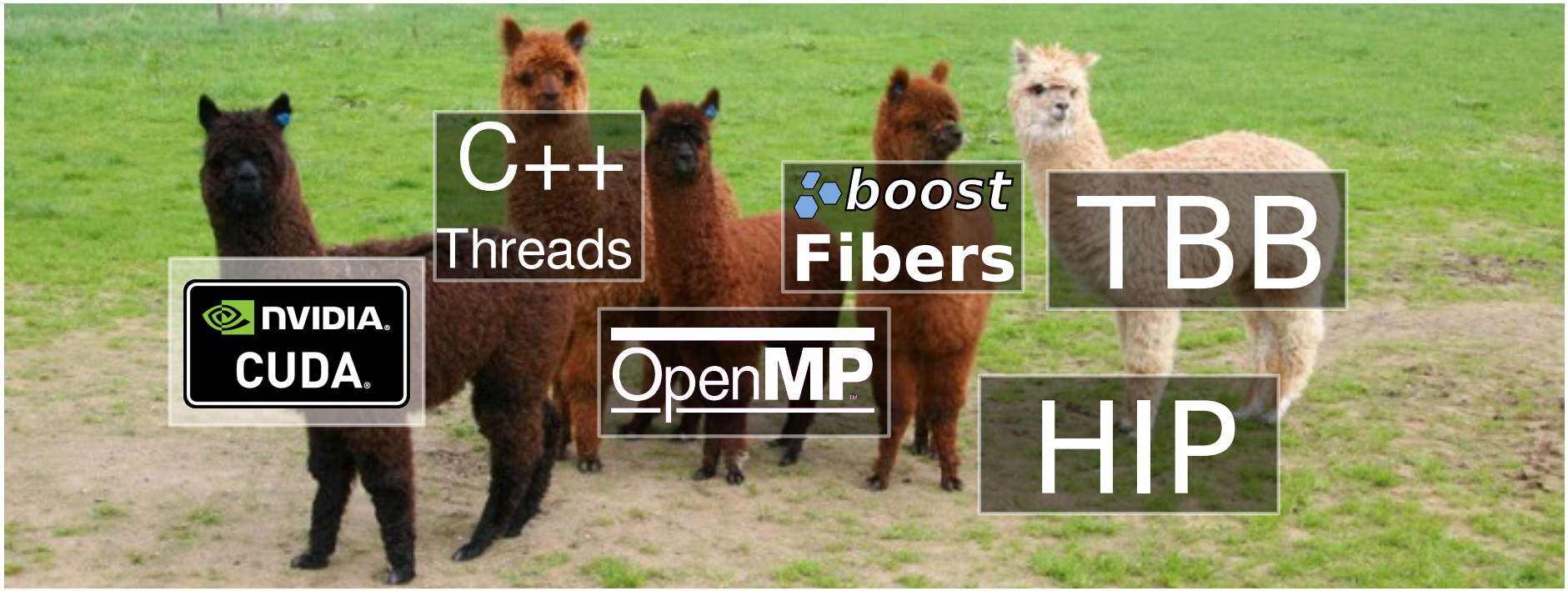




multi-core → many-core

Zoo of Programming Models

Each hardware has its own preference...



And even more: SYCL, HPX, ...

Maintainability

Heterogeneous Computing: Performance Portability

```
#ifdef CUDA_ENABLE
    // CUDA Kernel implementation
    // …




#endif
```

E. Zenker et al., ISC (2016), 10.1007/978-3-319-46079-6_21
A. Matthes et al., ISC (2017), 10.1007/978-3-319-67630-2_36

Maintainability

Heterogeneous Computing: Performance Portability

```
#ifdef CUDA_ENABLE
    // CUDA Kernel implementation
    // …

#elif OPENMP_ENABLE
    // OpenMP implementation
    // …




#endif
```

E. Zenker et al., ISC (2016), 10.1007/978-3-319-46079-6_21
A. Matthes et al., ISC (2017), 10.1007/978-3-319-67630-2_36

Maintainability

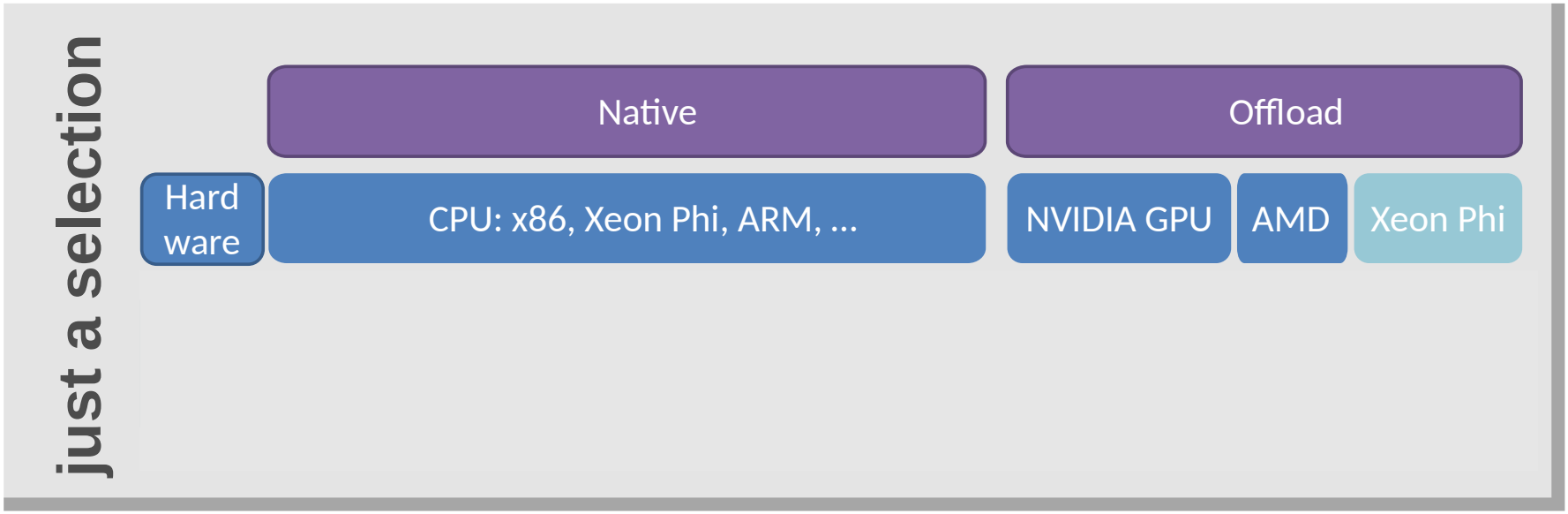
Heterogeneous Computing: Performance Portability

```
#ifdef CUDA_ENABLE
    // CUDA Kernel implementation
    // …

#elif OPENMP_ENABLE
    // OpenMP implementation
    // …

#elif TBB_ENABLE
    // TBB CPU implementation
    // …

#endif
```

E. Zenker et al., ISC (2016), 10.1007/978-3-319-46079-6_21
A. Matthes et al., ISC (2017), 10.1007/978-3-319-67630-2_36

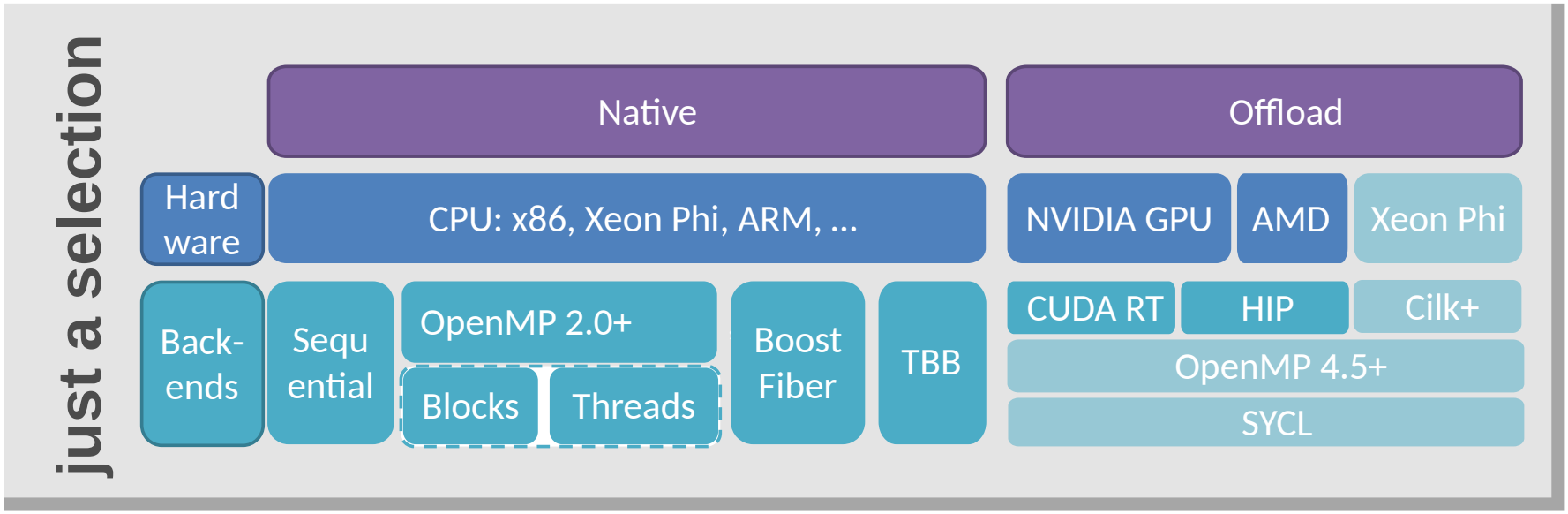
Maintainability

Performance Portability



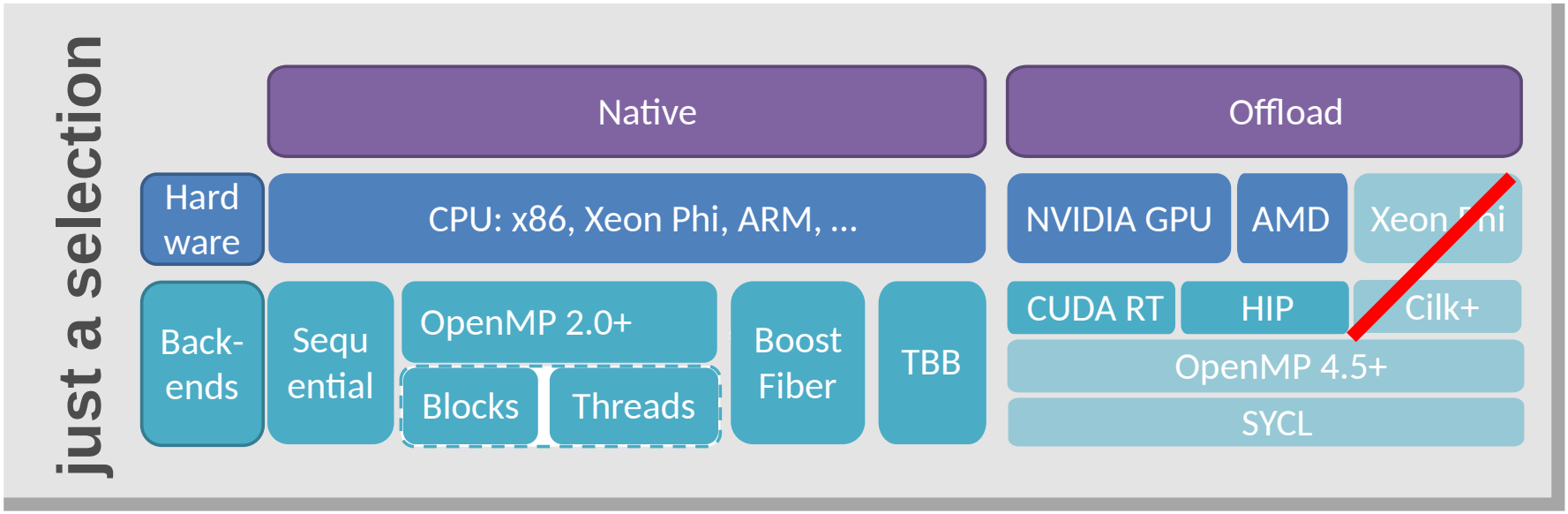
E. Zenker et al., ISC (2016), 10.1007/978-3-319-46079-6_21
A. Matthes et al., ISC (2017), 10.1007/978-3-319-67630-2_36

Maintainability
Performance Portability



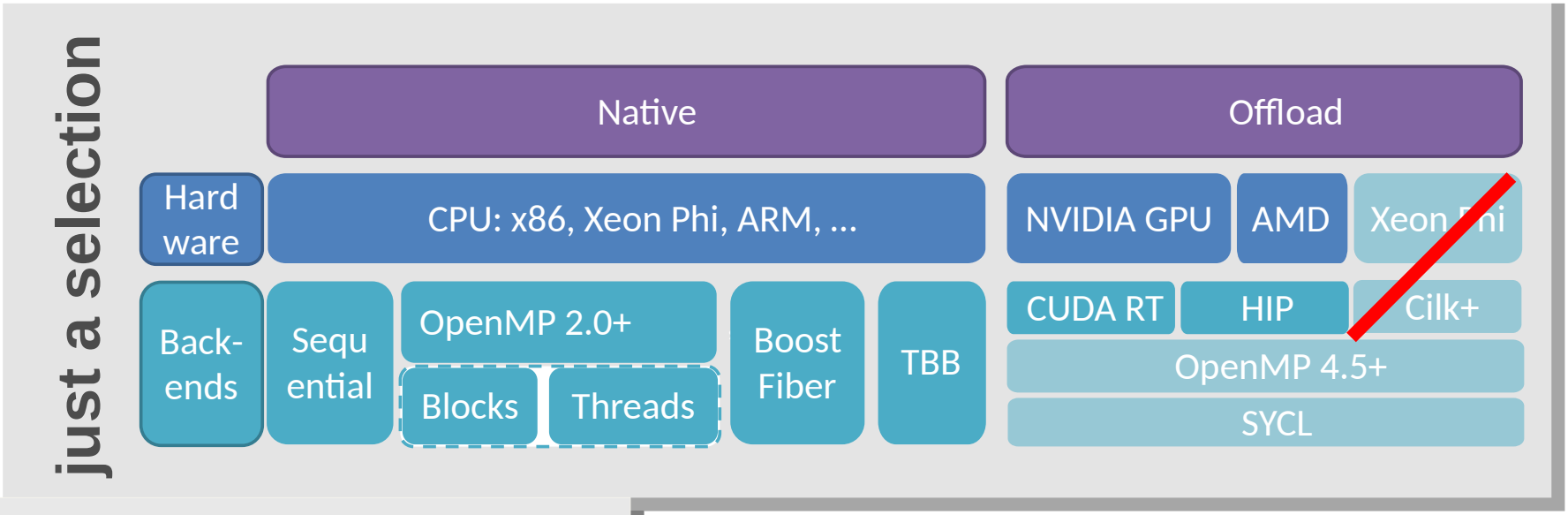
E. Zenker et al., ISC (2016), 10.1007/978-3-319-46079-6_21
A. Matthes et al., ISC (2017), 10.1007/978-3-319-67630-2_36

Maintainability
Performance Portability



E. Zenker et al., ISC (2016), 10.1007/978-3-319-46079-6_21
A. Matthes et al., ISC (2017), 10.1007/978-3-319-67630-2_36

Maintainability
Performance Portability

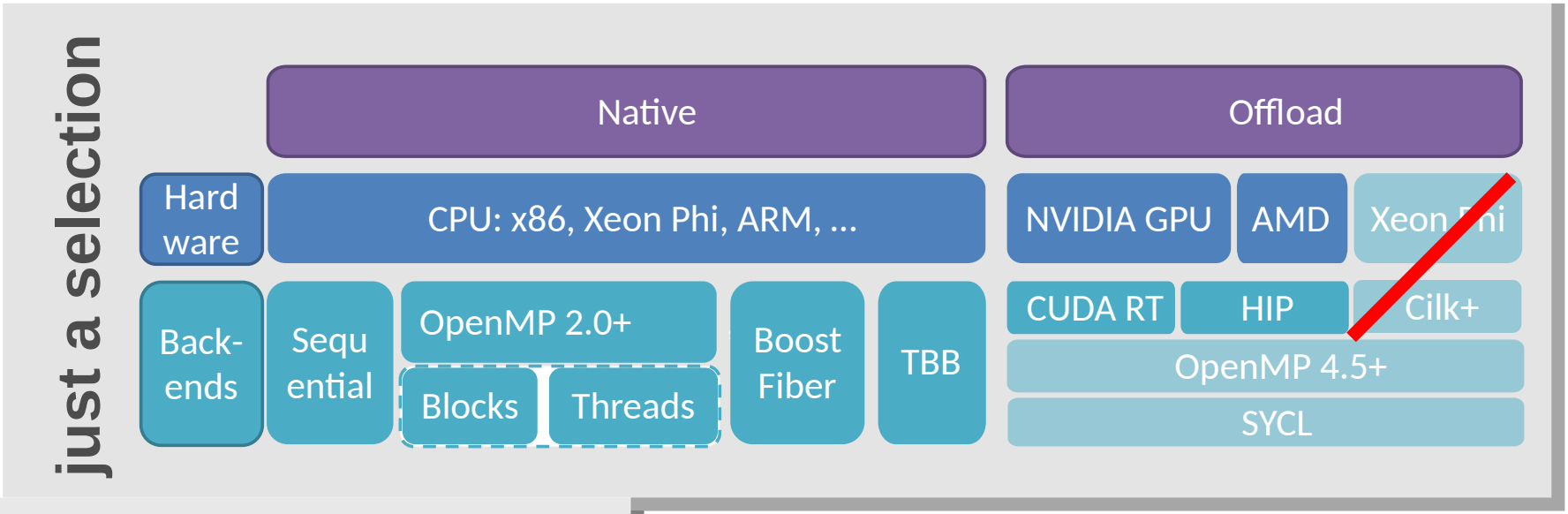



```
template< typename T_Acc >
ALPAKA_FN_ACC void operator()(
    T_Acc const & acc,
    // ...
) const
{
    // ...
}
```

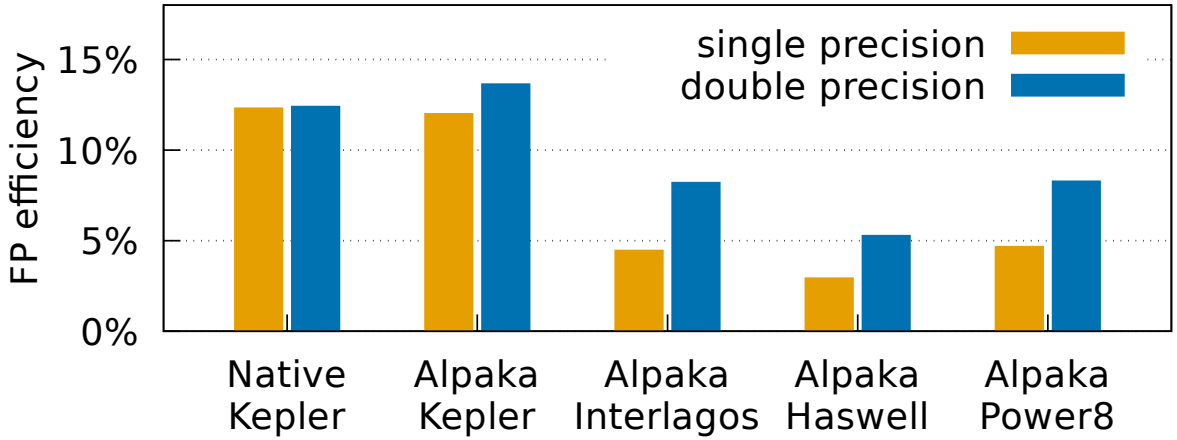

E. Zenker et al., ISC (2016), 10.1007/978-3-319-46079-6_21
A. Matthes et al., ISC (2017), 10.1007/978-3-319-67630-2_36

Maintainability

Performance Portability



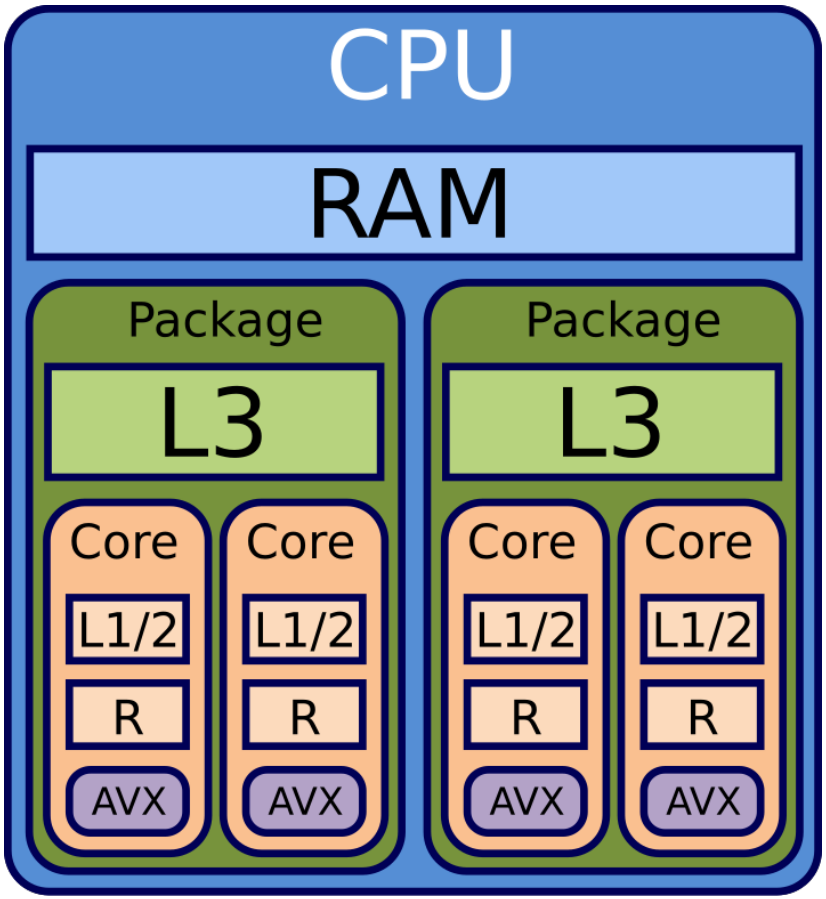

```
template< typename T_Acc >
ALPAKA_FN_ACC void operator()(
    T_Acc const & acc,
    // ...
) const
{
    // ...
}
```

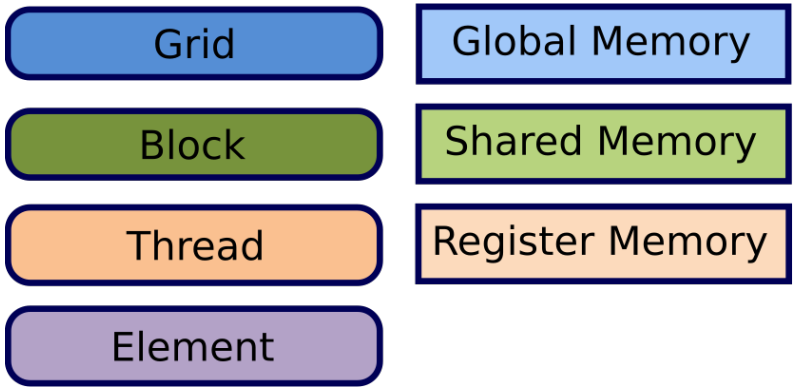


E. Zenker et al., ISC (2016), 10.1007/978-3-319-46079-6_21
A. Matthes et al., ISC (2017), 10.1007/978-3-319-67630-2_36

Hardware Mapping

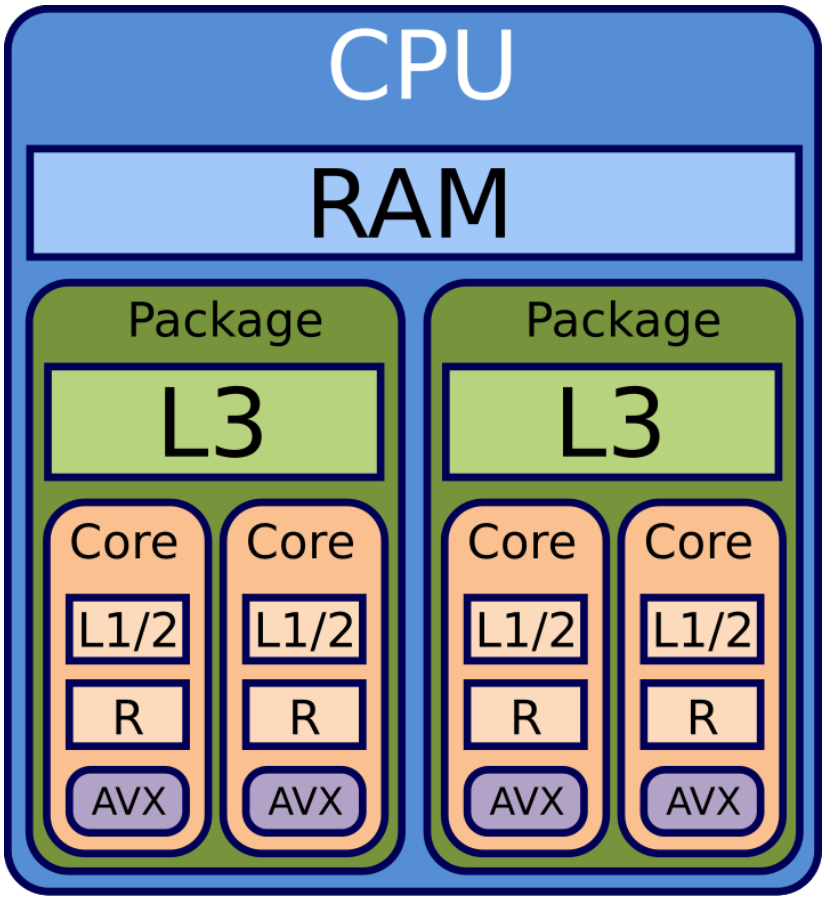
Alpaka on CPU, GPU, ...

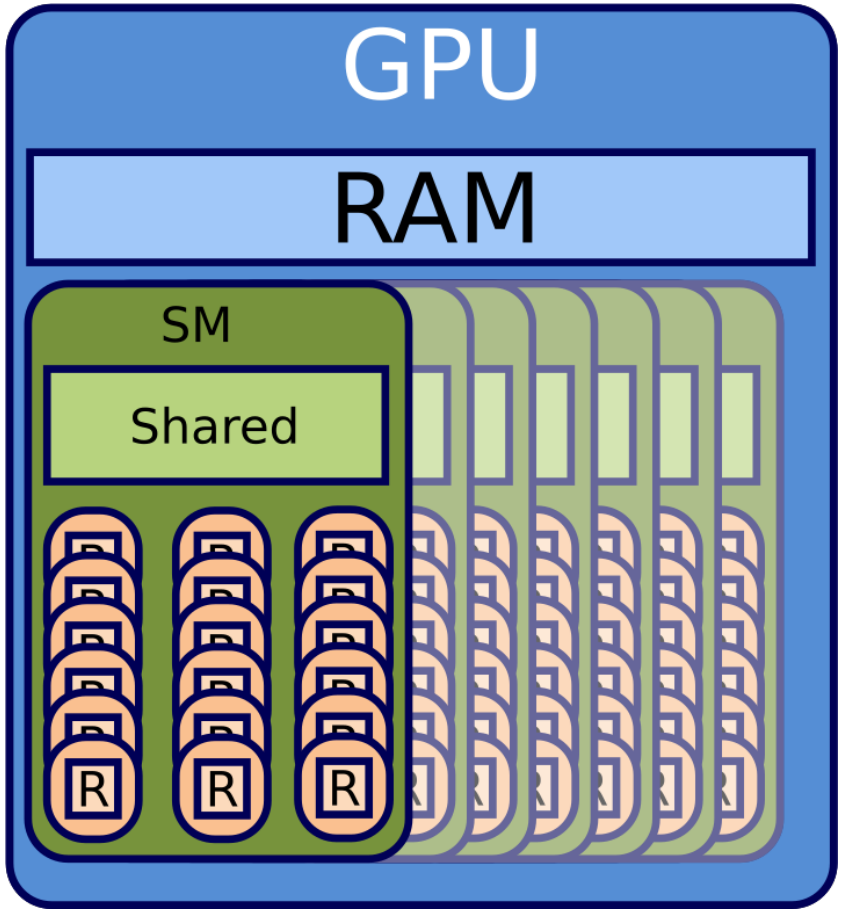


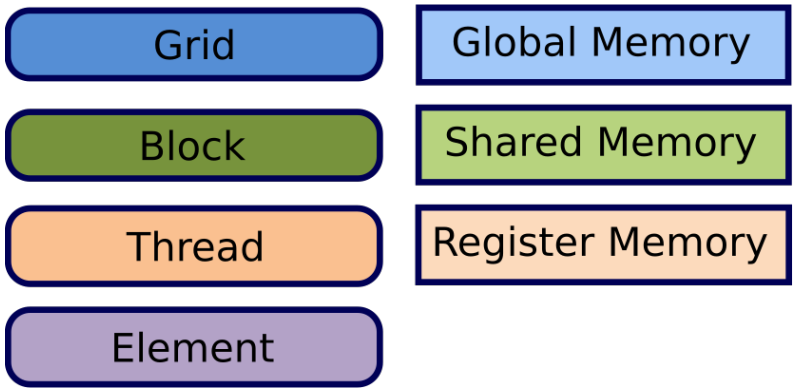


Hardware Mapping

Alpaka on CPU, GPU, ...

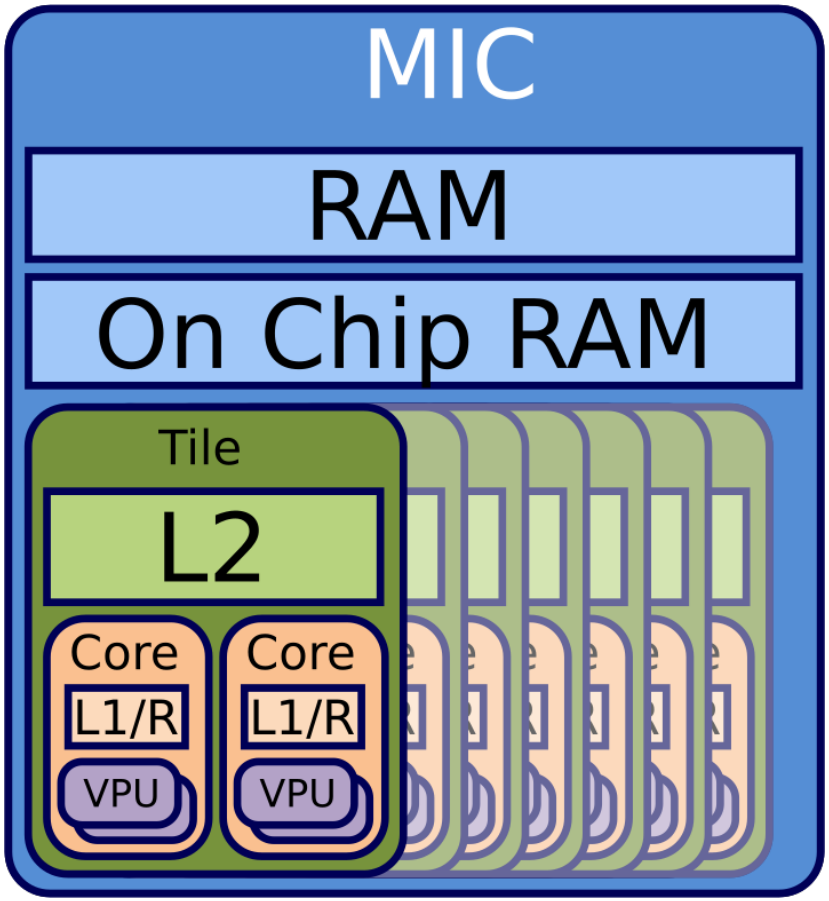


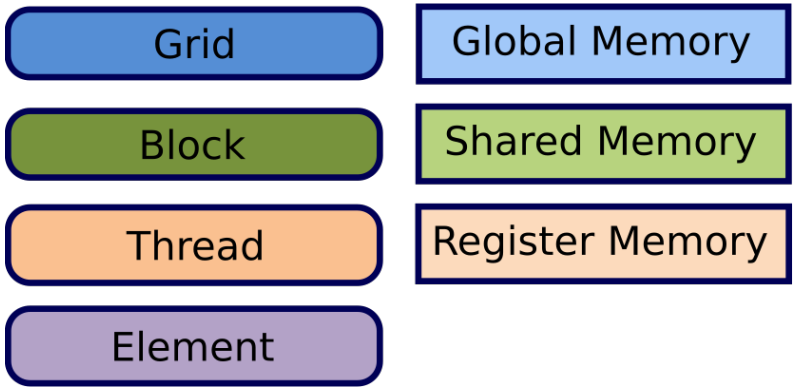




Hardware Mapping

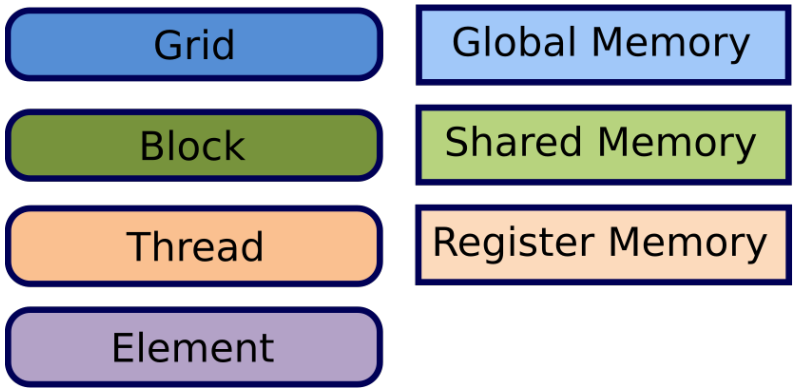
…, MIC, Power9, …

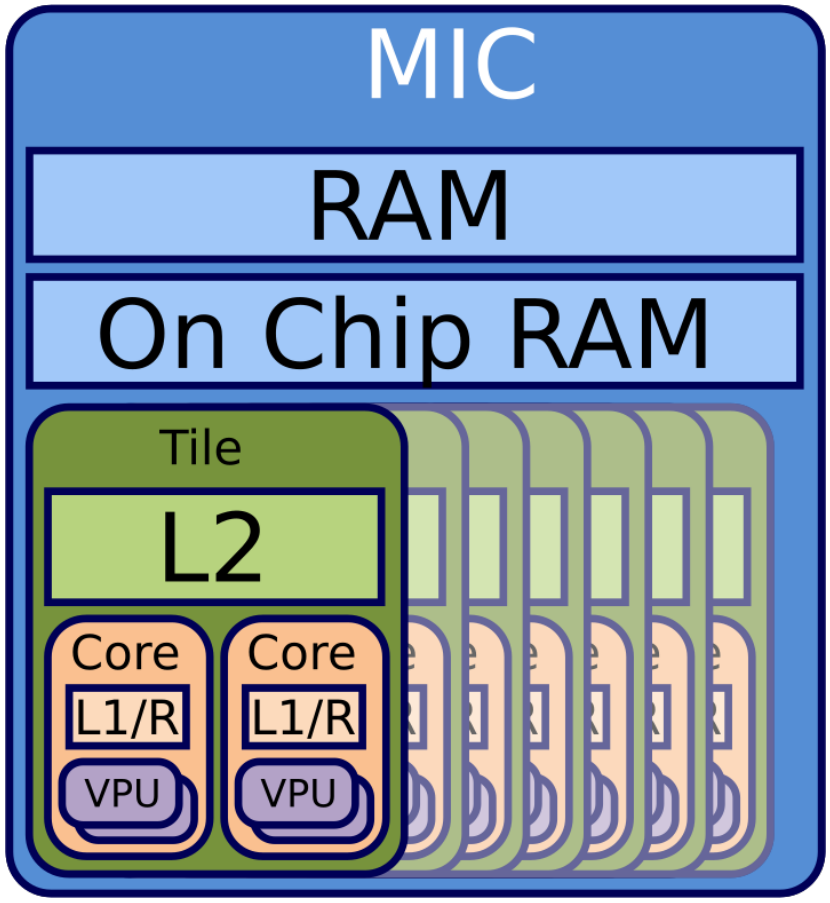


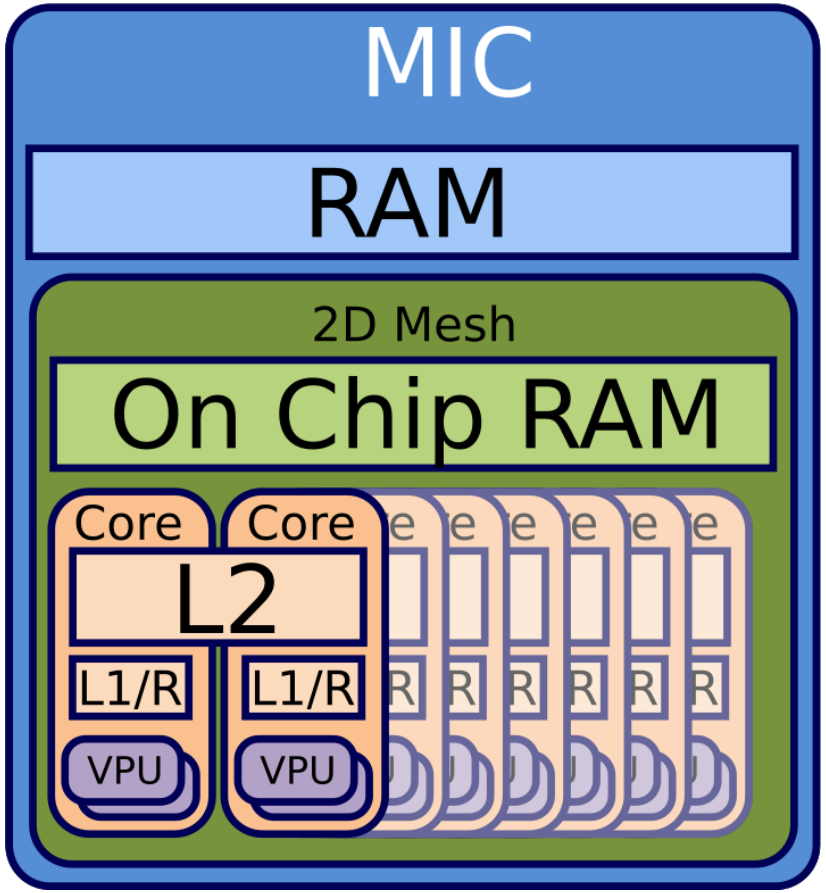


Hardware Mapping

…, MIC, Power9, …

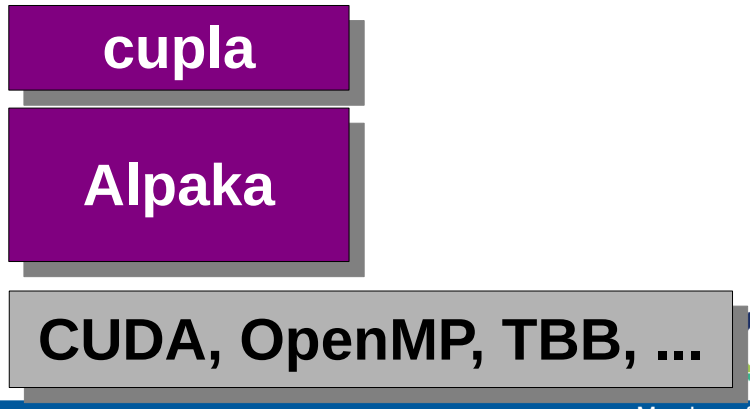






Exascale Software Stack

Further Aspects & FOSS Libraries of Ours



Exascale Software Stack

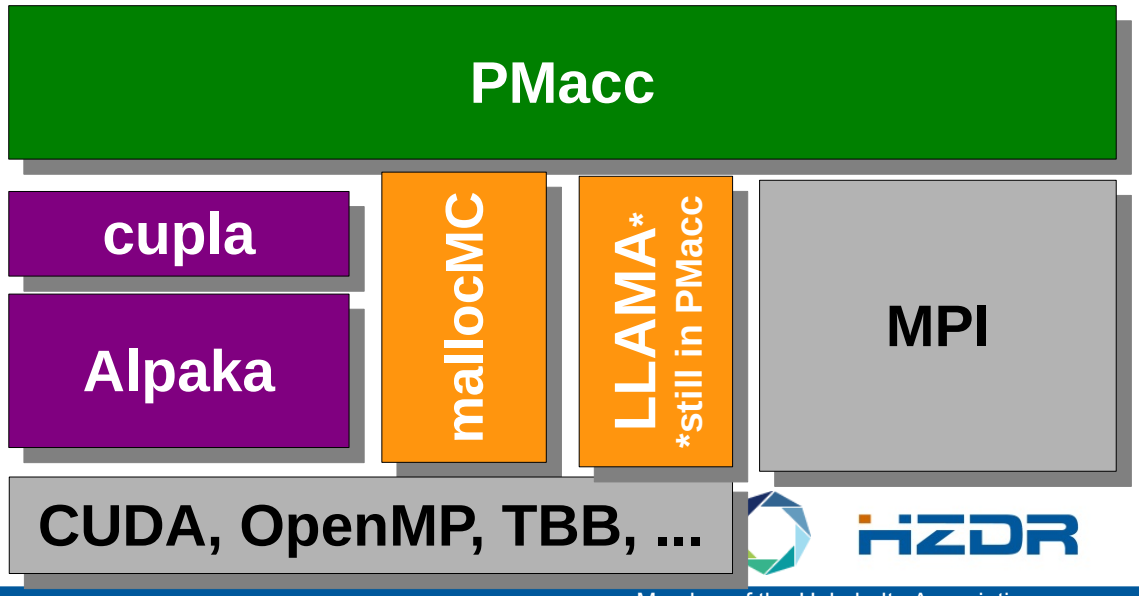
Further Aspects & FOSS Libraries of Ours

- massively **parallel heap**
- generalize SoA/AoS

Exascale Software Stack

Further Aspects & FOSS Libraries of Ours

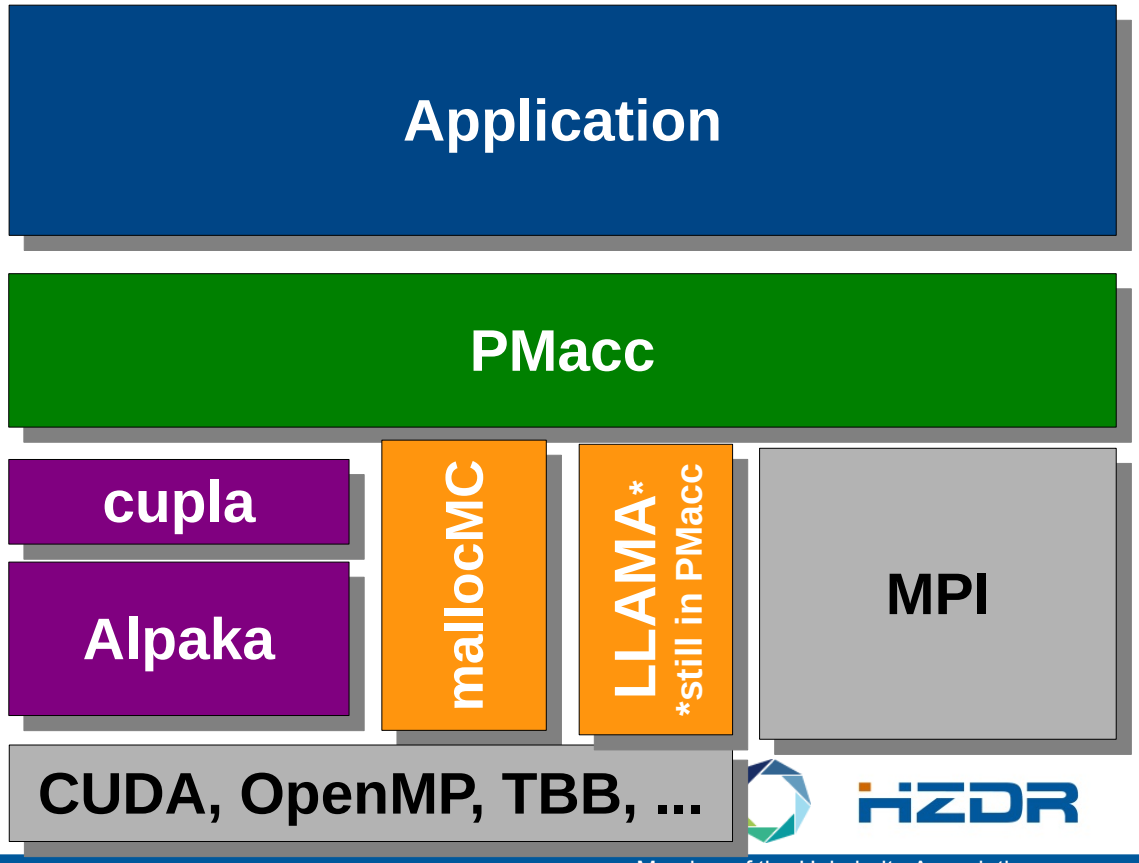
- **multi-node containers** for structured and unstructured 3D data
- massively **parallel heap**
- generalize SoA/AoS



Exascale Software Stack

Further Aspects & FOSS Libraries of Ours

- **multi-node containers** for structured and unstructured 3D data
- massively **parallel heap**
- generalize SoA/AoS
- asynchronous **in-node scheduler**

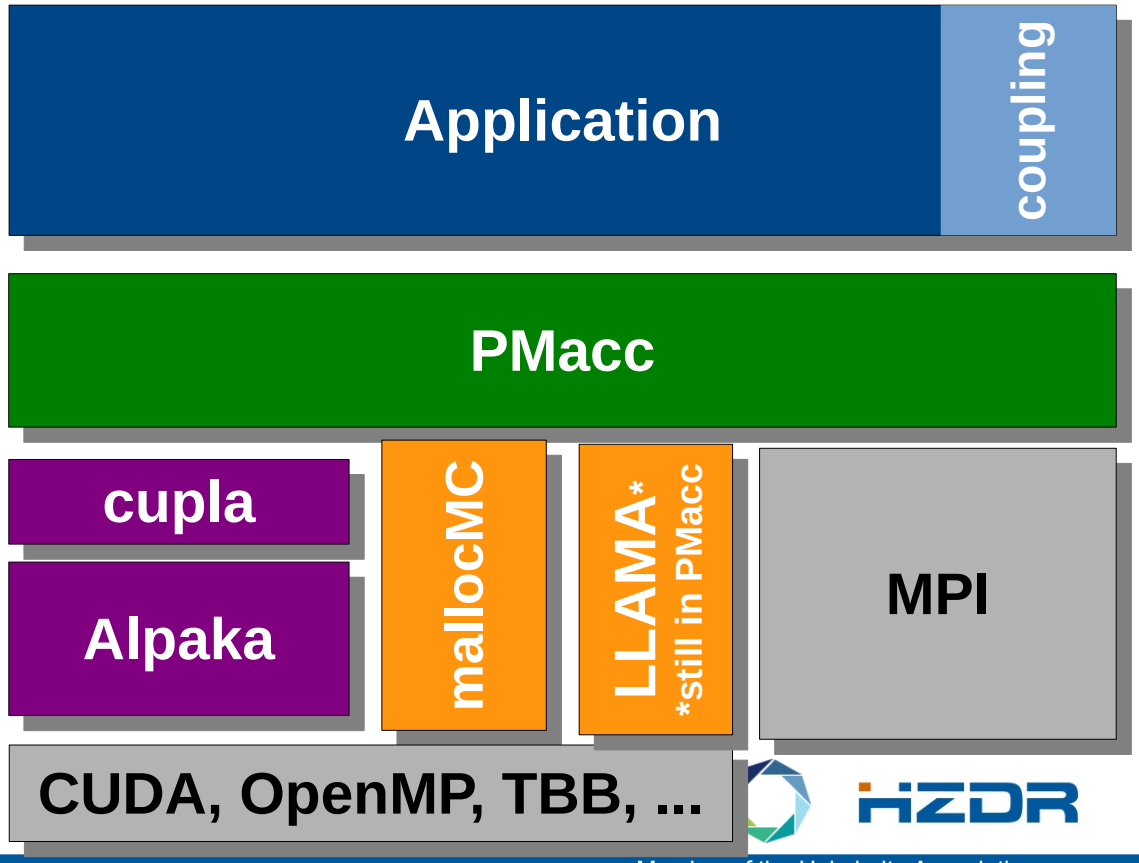


Exascale Software Stack

Further Aspects & FOSS Libraries of Ours

- **multi-node containers** for structured and unstructured 3D data
- massively **parallel heap**
- generalize SoA/AoS

- asynchronous **in-node scheduler**
- multi-node, **in-memory** visualization



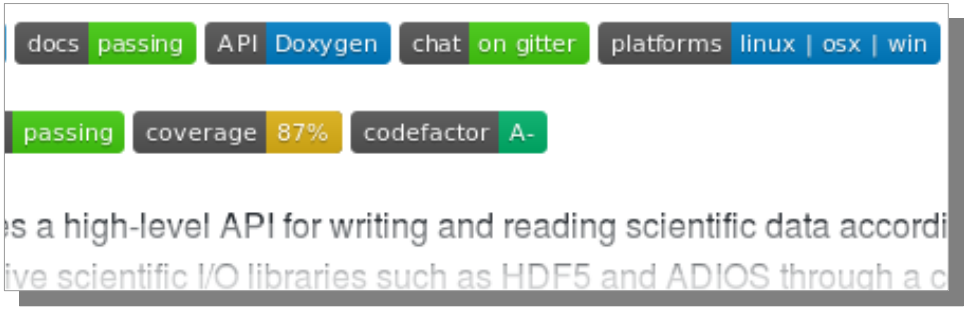
Collaboration through Modularity

Zero-Overhead Abstractions

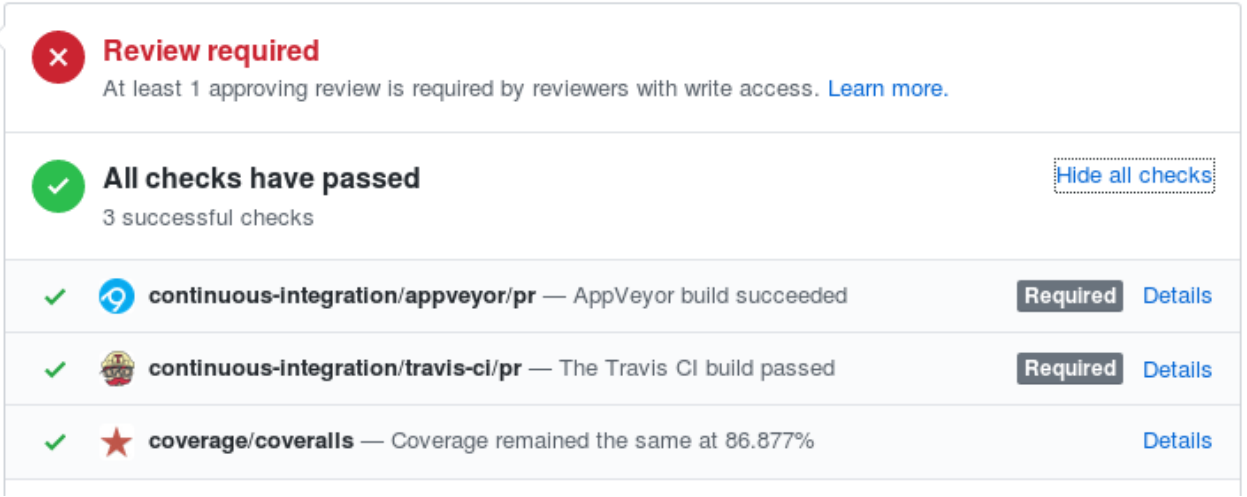
- **Maintainability** through
  - **modularity**

Collaboration through Modularity

Zero-Overhead Abstractions

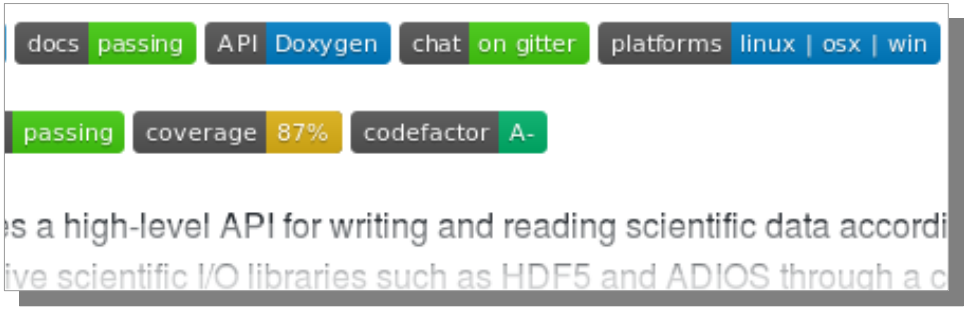


- **Maintainability** through
  - **modularity**
  - single-source programming models
  - **fully open** + CI, CD & contin. docs

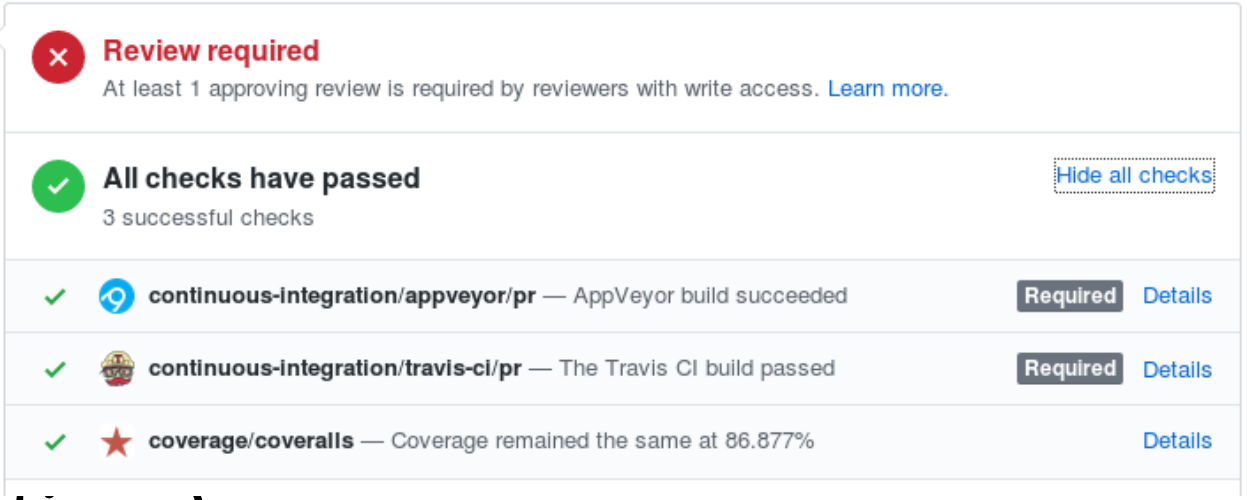


Collaboration through Modularity

Zero-Overhead Abstractions



- **Maintainability** through
  - **modularity**
  - single-source programming models
  - **fully open** + CI, CD & contin. docs



- **Abstraction**
  - **zero-cost** (runtime)
  - specializable

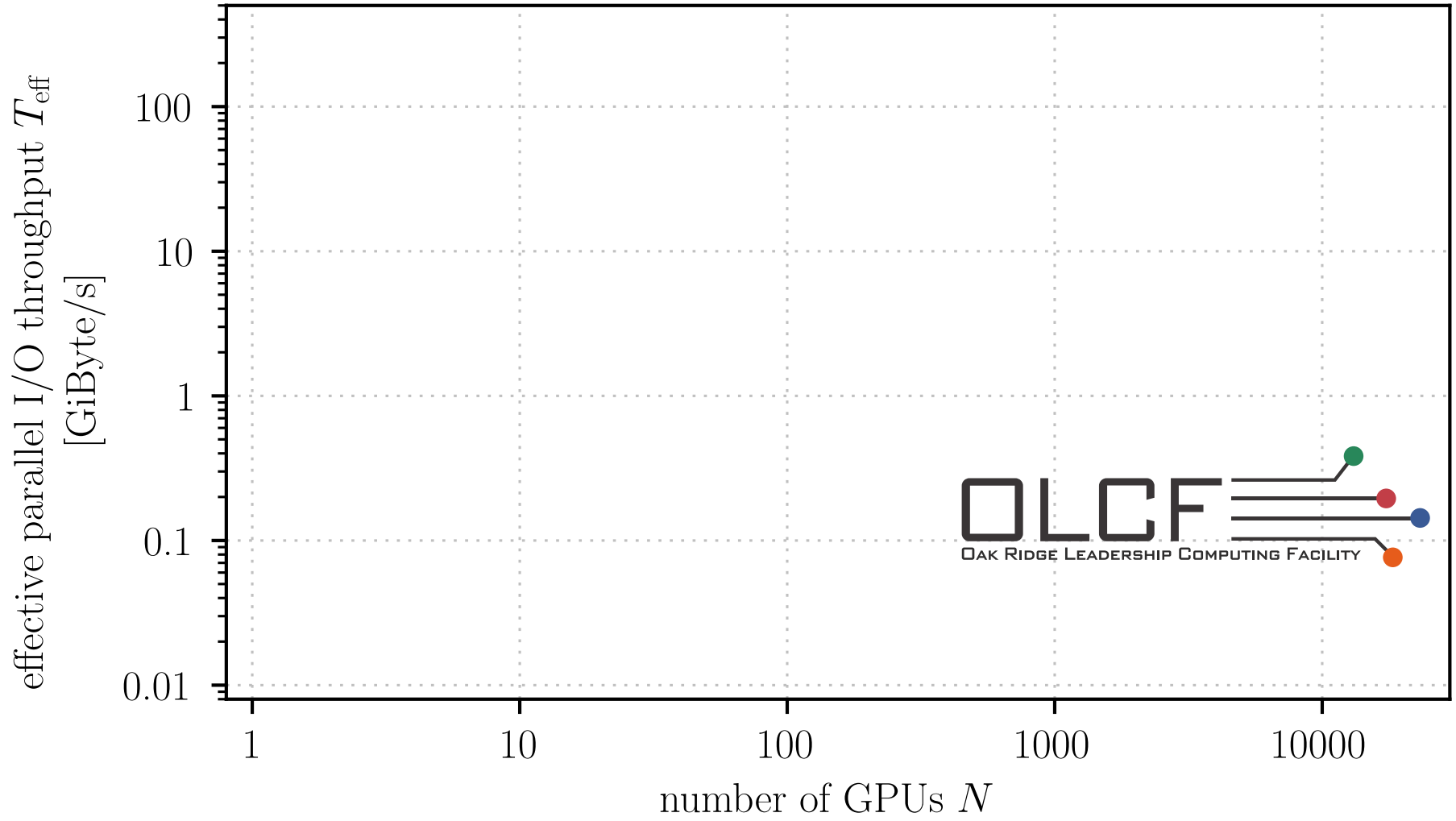
Data-Driven Science

A modern use-case for digital twins

(Cold) Data

I/O died in 2013. Long live I/O!

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



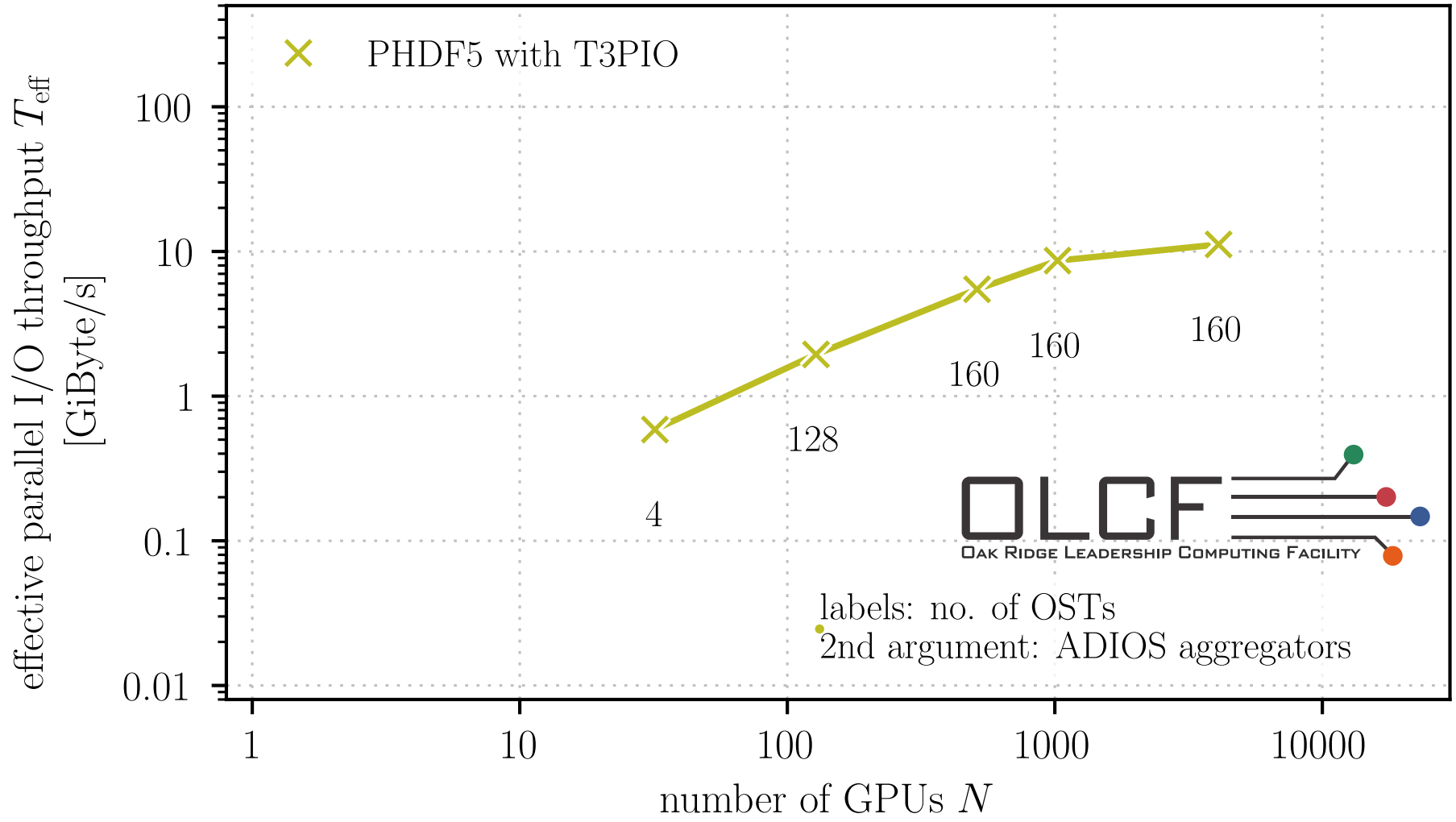
www.olcf.ornl.gov/summit

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

(Cold) Data

I/O died in 2013. Long live I/O!

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

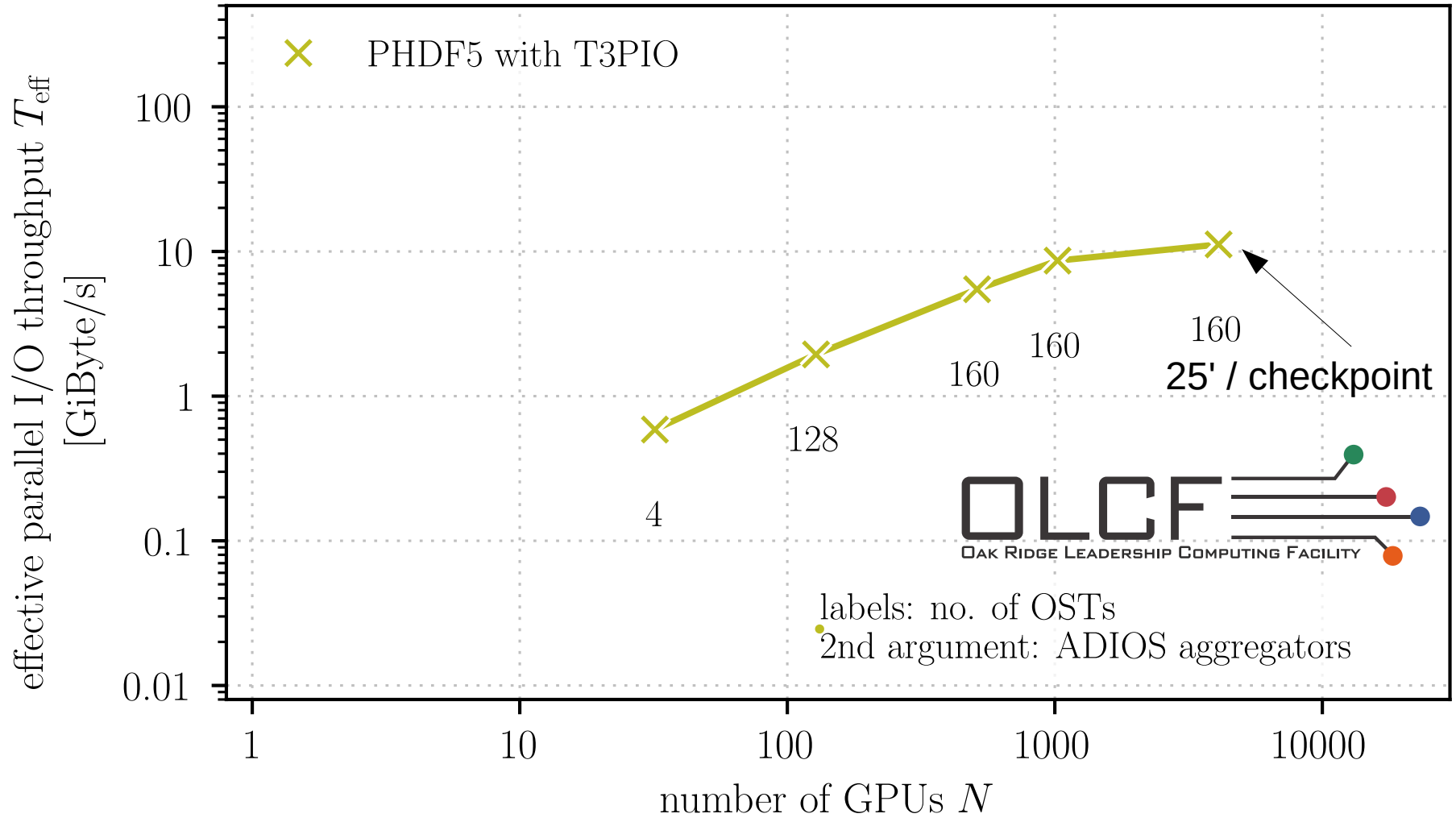


www.olcf.ornl.gov/summit

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

(Cold) Data
I/O died in 2013. Long live I/O!

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



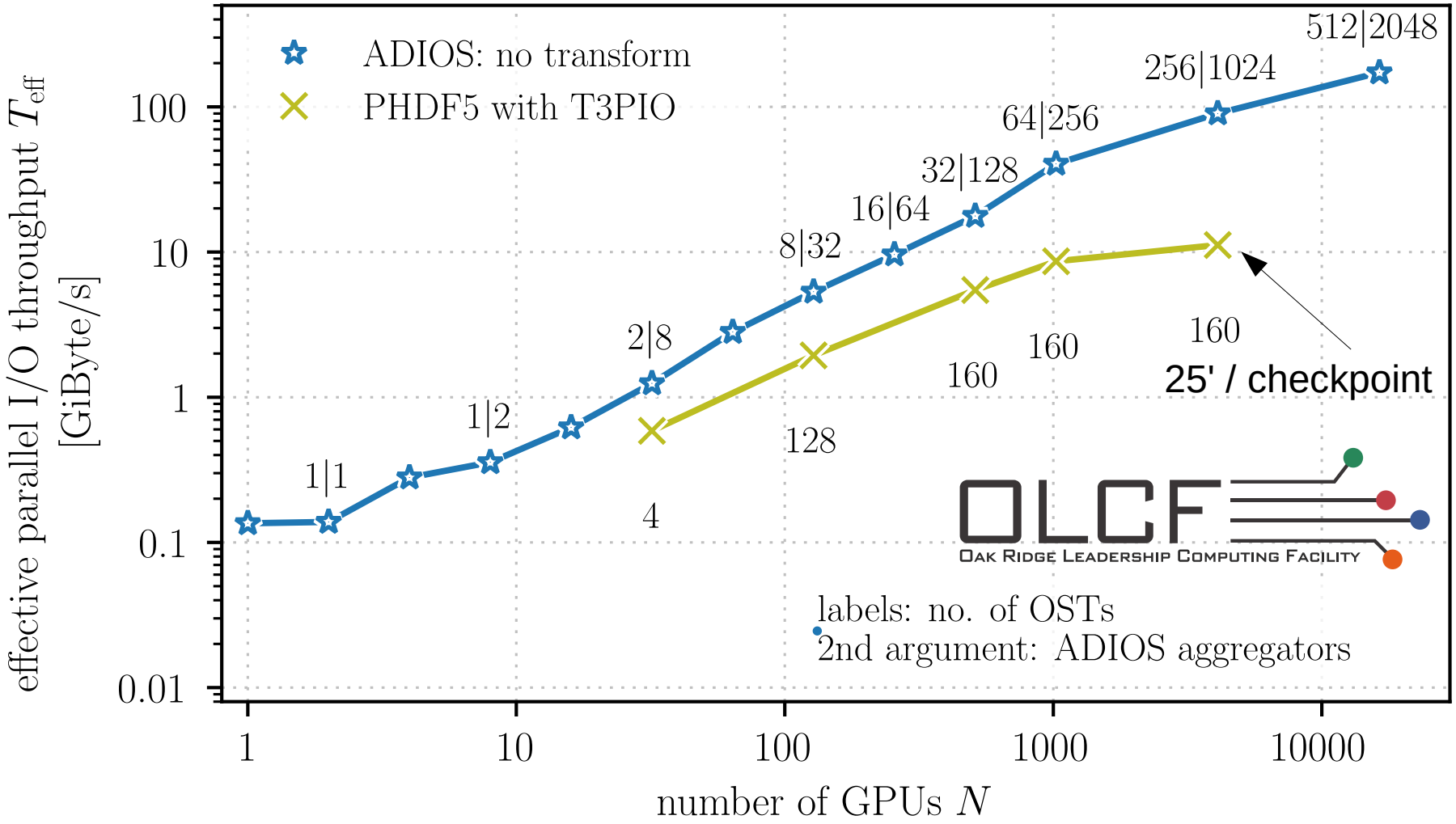
www.olcf.ornl.gov/summit

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

DRESDEN concept HZDR

(Cold) Data
I/O died in 2013. Long live I/O!

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

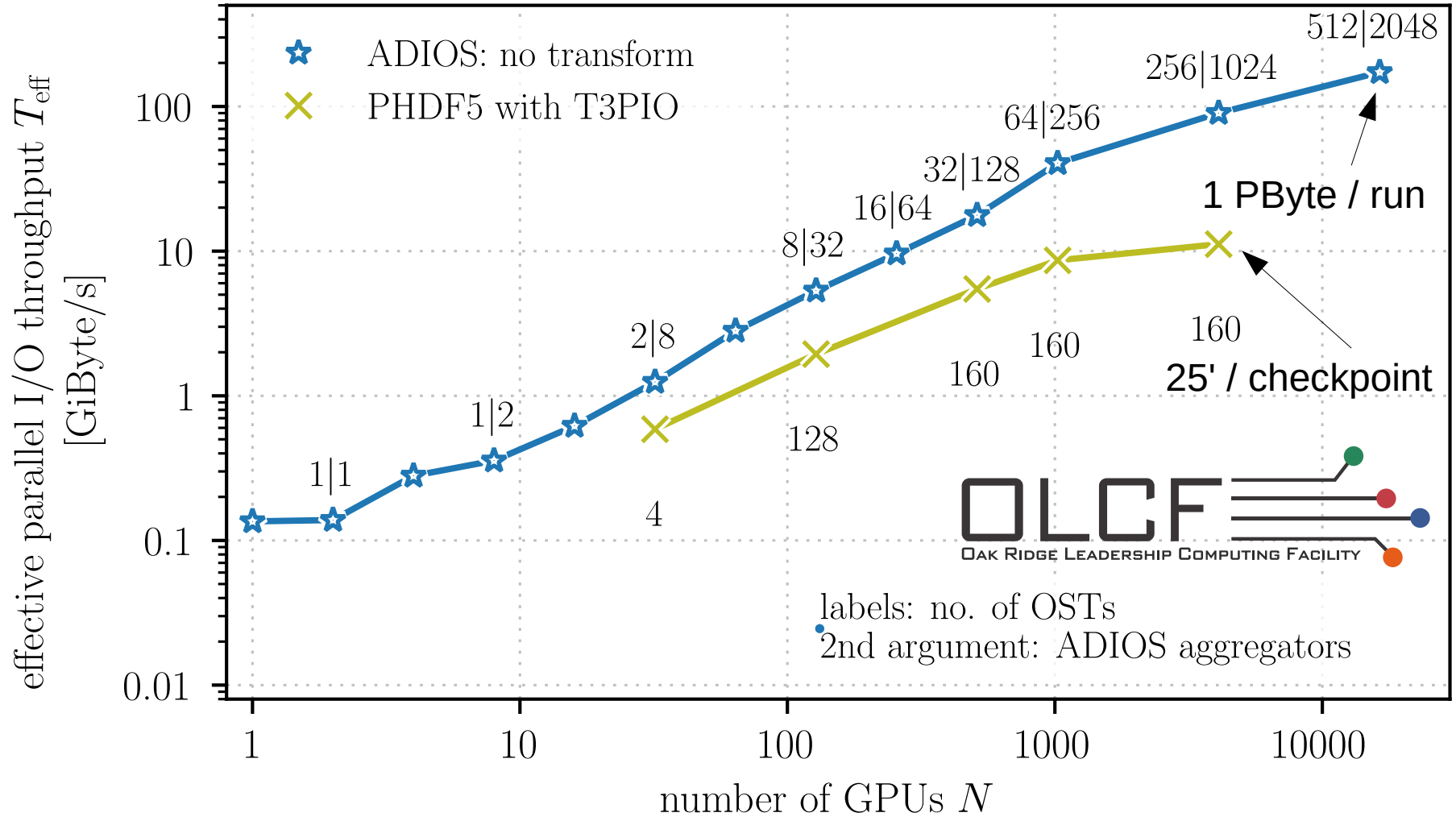


www.olcf.ornl.gov/summit

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

(Cold) Data
I/O died in 2013. Long live I/O!

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



www.olcf.ornl.gov/summit

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

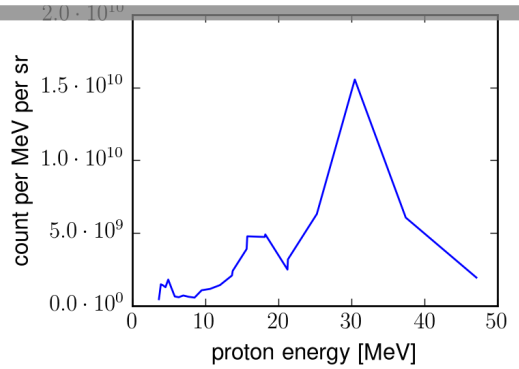
DRESDEN concept HZDR

~~(Cold)~~ Hot Data Analysis

In Situ: Invert Workflows and Repeat

in situ solutions:
openPMD + ADIOS,
"plugins": ISAAC, ...

Binning of a **spectrogram**
Creation of a **phase space image**



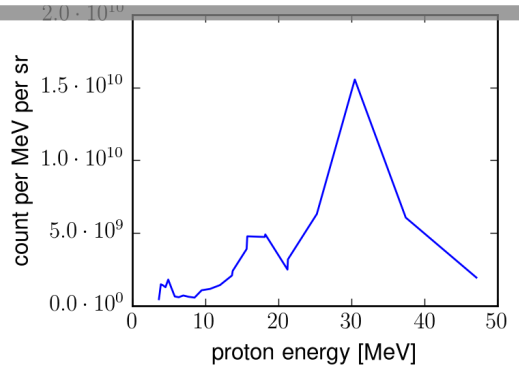
A. Huebl et al. (2014), DOI:10.1109/TPS.2014.2327392
R. Pausch et al. (2017), DOI:10.1103/PhysRevE.96.013316
A. Matthes, A. Huebl et al., ISC'16 (2016), DOI:10.14529/jsfi160403
A. Huebl et al., ISC'17 (2017), DOI:10.1007/978-3-319-67630-2_2

~~(Cold)~~ Hot Data Analysis

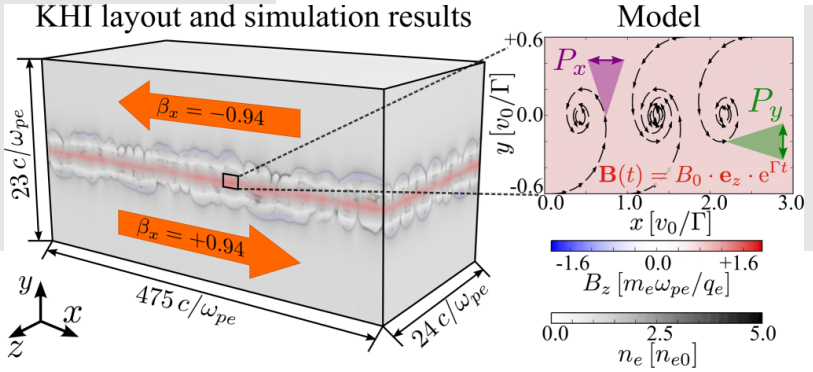
In Situ: Invert Workflows and Repeat

in situ solutions:
openPMD + ADIOS,
"plugins": ISAAC, ...

Binning of a **spectrogram**
Creation of a **phase space image**



In situ **radiation diagnostics**



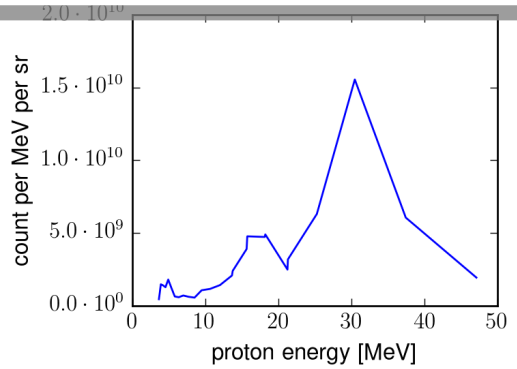
A. Huebl et al. (2014), DOI:10.1109/TPS.2014.2327392
R. Pausch et al. (2017), DOI:10.1103/PhysRevE.96.013316
A. Matthes, A. Huebl et al., ISC'16 (2016), DOI:10.14529/jsfi160403
A. Huebl et al., ISC'17 (2017), DOI:10.1007/978-3-319-67630-2_2

~~(Cold)~~ Hot Data Analysis

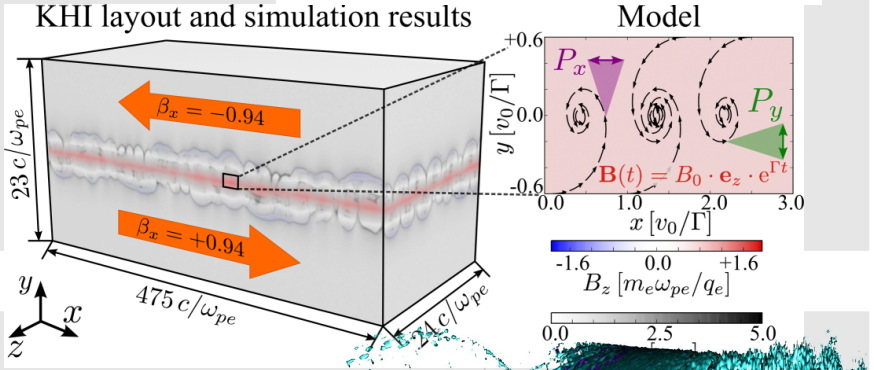
In Situ: Invert Workflows and Repeat

in situ solutions:
openPMD + ADIOS,
"plugins": ISAAC, ...

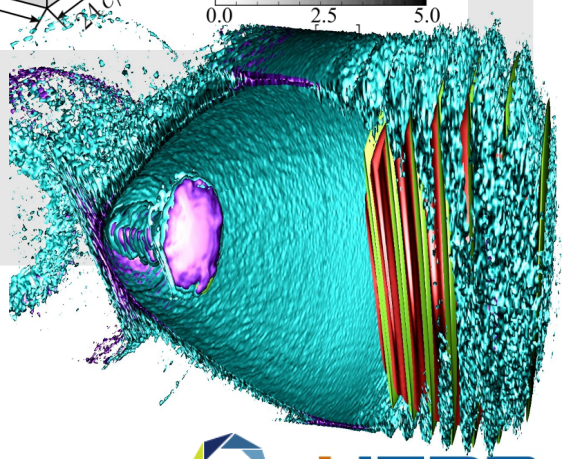
Binning of a **spectrogram**
Creation of a **phase space image**



In situ **radiation diagnostics**



Ray-cast or photo-realistic ray-trace
Lossy data **compression**



A. Huebl et al. (2014), DOI:10.1109/TPS.2014.2327392
R. Pausch et al. (2017), DOI:10.1103/PhysRevE.96.013316
A. Matthes, A. Huebl et al., ISC'16 (2016), DOI:10.14529/jsfi160403
A. Huebl et al., ISC'17 (2017), DOI:10.1007/978-3-319-67630-2_2

DRESDEN concept

HZDR

~~(Cold)~~ Hot Data Analysis

In Situ: Invert Workflows and Repeat

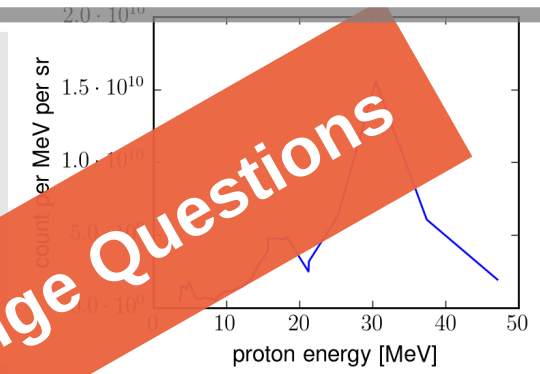
in situ solutions:
openPMD + ADIOS,
"plugins": ISAAC, ...

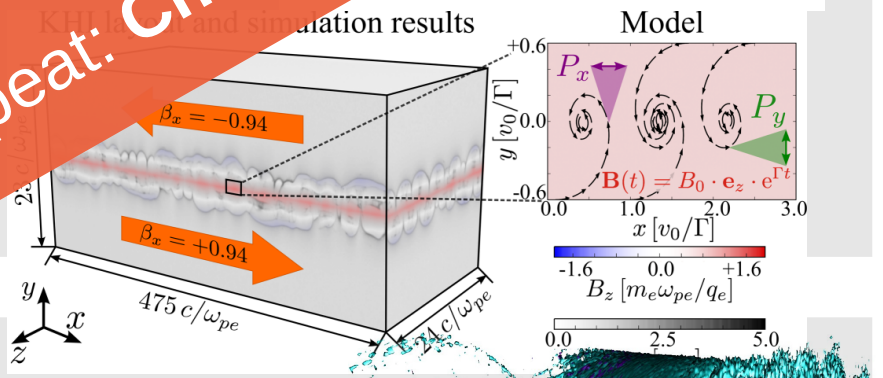
Binning of a **spectrogram**
Creation of a **phase space image**

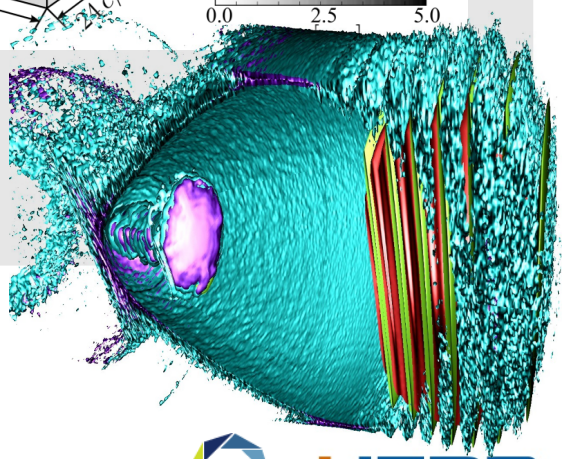
In situ **radiation diagnostics**

Ray-cast or photo-realistic ray-trace
Lossy data **compression**

Observe, Correlate & Repeat: Change Questions







A. Huebl et al. (2014), DOI:10.1109/TPS.2014.2327392
R. Pausch et al. (2017), DOI:10.1103/PhysRevE.96.013316
A. Matthes, A. Huebl et al., ISC'16 (2016), DOI:10.14529/jsfi160403
A. Huebl et al., ISC'17 (2017), DOI:10.1007/978-3-319-67630-2_2

DRESDEN concept
HZDR

Data-Driven, Interactive Exploration

Usability

- **data** (reduced or not)
  open meta standard

Data-Driven, Interactive Exploration

Usability

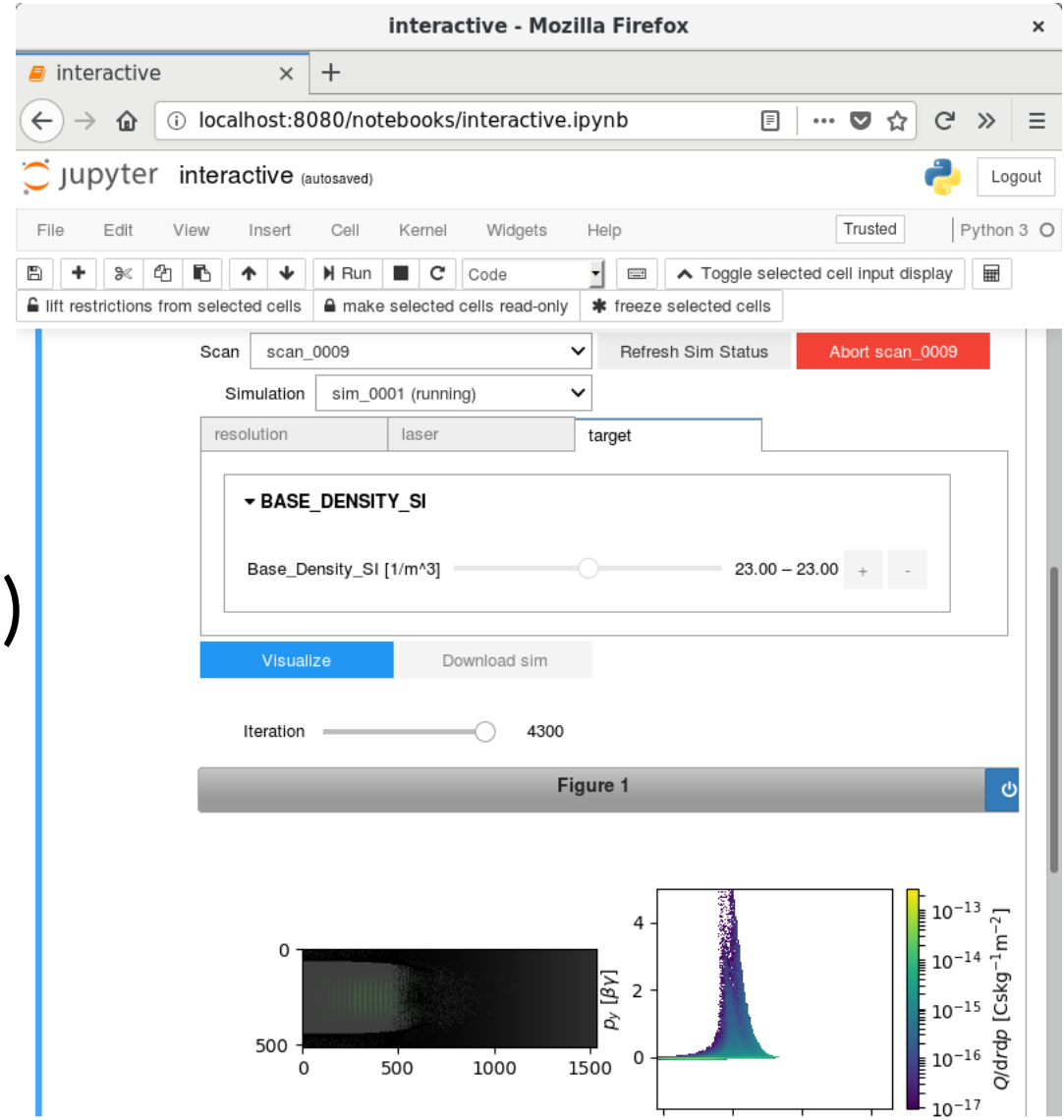
- **data** (reduced or not)
  open meta standard

- **installability**
  containerization
  package manager

Data-Driven, Interactive Exploration

Usability

- **data** (reduced or not)
  open meta standard
- **scientific questions**
  python & GUI scripts
  transparent (hackable)
  Jupyter integration
- **installability**
  containerization
  package manager



Summary

Our Existing Open Source & OpenData Projects

Our Scientific FOSS Projects

Try, Reuse, Complain ;-) and Improve

ComputationalRadiationPhysics/

PIConGPU

root-project/cling: -x CUDA

mallocMC **PMacc**

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

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 has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 654220.

Talk by **A. Huebl** (HZDR), a.huebl@hzdr.de

DRESDEN concept HZDR

Backup Slides

Comparison to X
Performance Portability

Framework / API	Open-Source	Free	Single-Source C++	Port-ability	Hetero-genity	Maintain-ability	Test-ability	Optimiz-ability	Data structure agnostic
CUDA	✗	✓	✓	✗	✗	✗	✗	✓	✓
PGI CUDA-x86	✗	✗	✓	✓	❍	✓	✓	✗	✓
GPU Ocelot	✓	✓	✓	✓	❍	✓	✓	✗	✓
OpenMP	✓	✓	✓	✓	✓	✓	✓	✗	✓
OpenACC	✓	✓	✓	✓	✓	✓	✓	✗	✓
OpenCL	✓	✓	✗	✓	✓	✓	✓	✗	✓
SYCL	✓	(✓)	✓	✓	✓	✓	✓	(✓)	✓
C++AMP	✓	✓	✓	(✓)	✓	✓	✓	✗	✓
KOKKOS	✓	✓	✓	✓	❍	✓	✓	✗	❍
Thrust	✓	✓	✓	✓	❍	✓	✓	✗	✗

New since then: RAJA, Kokkos, HPX, ...

E. Zenker et al., IPDPSW (2016), DOI: 10.1109/IPDPSW.2016.50
E. Zenker et al., ISC (2016), 10.1007/978-3-319-46079-6_21
A. Matthes et al., ISC (2017), 10.1007/978-3-319-67630-2_36

Scalability

Algorithmic Challenges & Memory Bounds

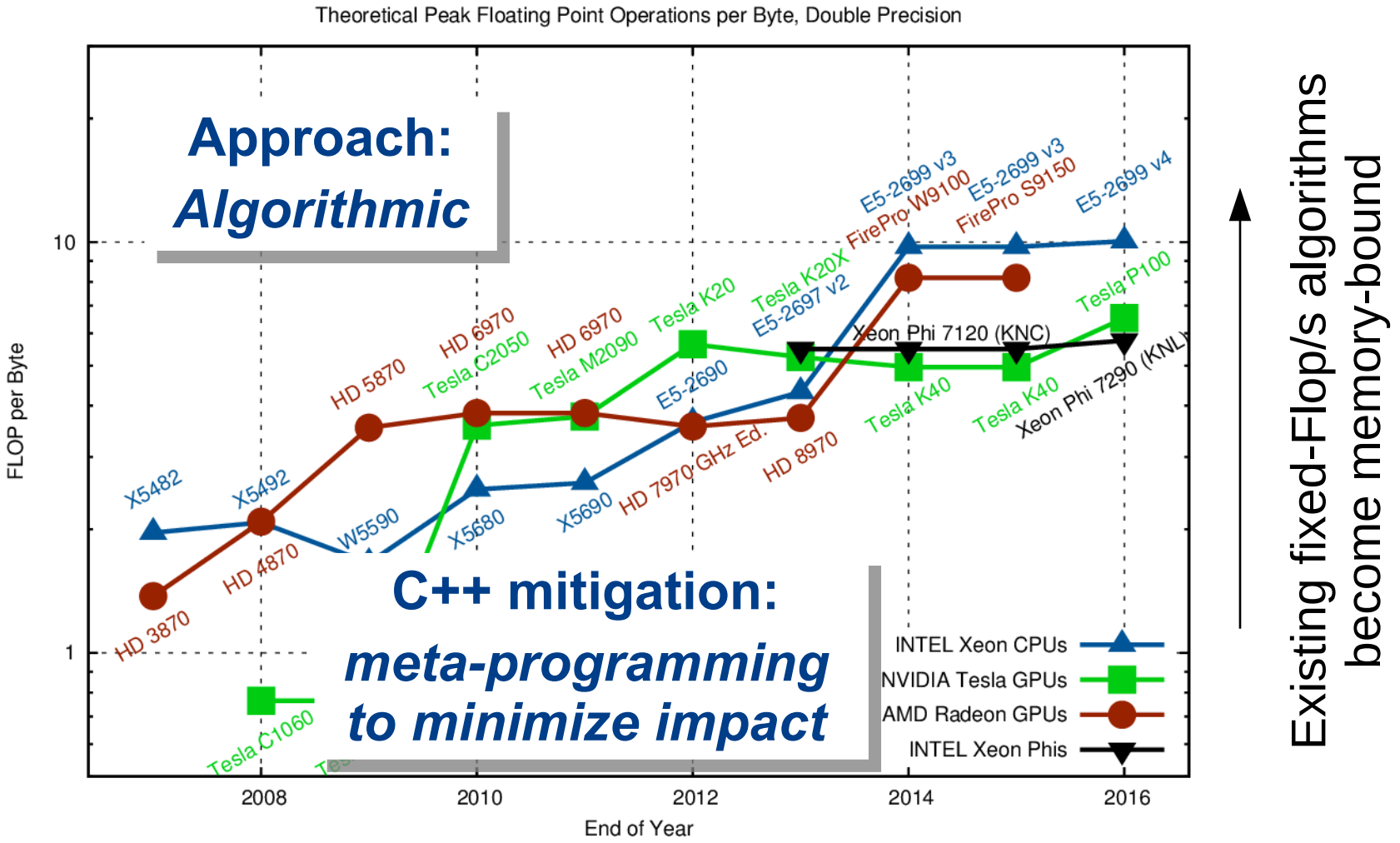


Image: K. Rupp. *CPU, GPU and MIC Hardware Characteristics over Time*, on karlrupp.net
Also: S. Williams. *Roofline: an insightful vis. perf. model for multicore architectures*, 10.1145/1498765.1498785 (2009)

~~(Cold)~~ Hot Data Analysis

In Situ Analysis: Reproducibility

+ review
+ eductation

- Mandates **open science approach**
- **Repeat** (simulation, analysis) with:
  - newly measured data / computing time
  - new question (query)
- Imagine LHC High-Luminosity upgrade:
  - submit in situ **“queries”** before beam-time
  - just as in software: **review, improve,** … queries
  - repeat **modified** queries in next beam-time

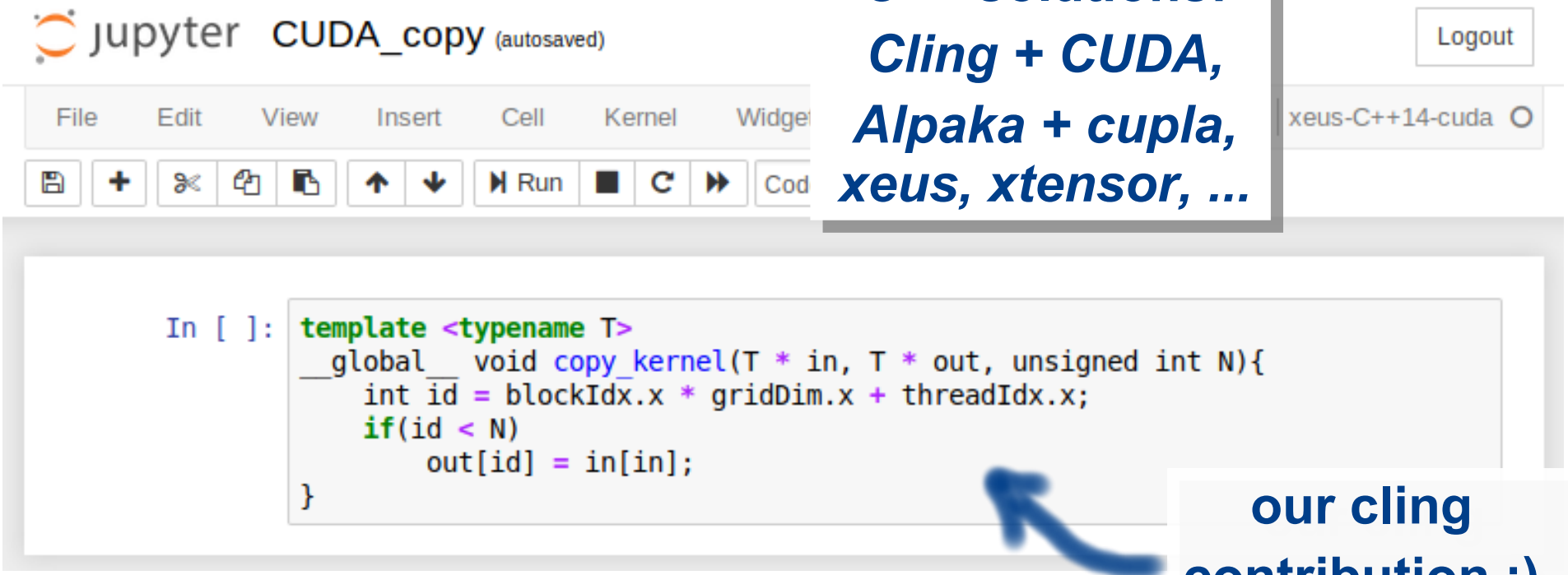
explorative science!

DRESDEN concept HZDR

Expressive, Productive Programming

Interactive, Modern, JITed C++

C++ solutions:
Cling + CUDA,
Alpaka + cupla,
xeus, xtensor, ...



our cling contribution :)

- towards full **Alpaka** support → template change: CPU/GPU/...

Cling CUDA: S. Ehrig (HZDR, TU Dresden), Diploma Thesis (2018)

Application Software Stack

Let's Draft an Ideal World

Application

helper

Containers and Algorithms

In-Node Acceleration

Message-Passing

DRESDEN concept

HZDR

Application C++ Software Stack
PIConGPU

PIConGPU

Plugins

Boost

PMacc

cupla

Alpaka

mallocMC

LLAMA*
*still in PMacc

MPI

CUDA, OpenMP, TBB, ...

DRESDEN concept

HZDR

Code Example: OpenMP

Mix of algorithm/data & parallelization strategy

- Pragma-based approaches, e.g. **OpenMP**:

```
auto axpy(
    double const & alpha,
    double const * const X,
    double * const Y,
    std::size_t const & n
) const
    -> void
{
    #pragma omp parallel for
    for( std::size_t i = 0u; i < n; ++i )
    {
        Y[i] = alpha * X[i] + Y[i];
    }
}
```

Code Example: Alpaka

Zero-Overhead, User-Specializable

- Parallelized depending on Hardware `TAcc const & acc`

```
struct DaxpyKernel
{
    template< typename T_Acc >
    ALPAKA_FN_ACC void operator()(
        T_Acc const & acc,
        double const & alpha,
        double const * const X,
        double * const Y,
        int const & numElements
    ) const
    {
        using alpaka;
        auto const i = alpaka::idx::getIdx< Grid, Threads >( acc )[0];
        Y[i] = alpha * X[i] + Y[i];
    }
};
```

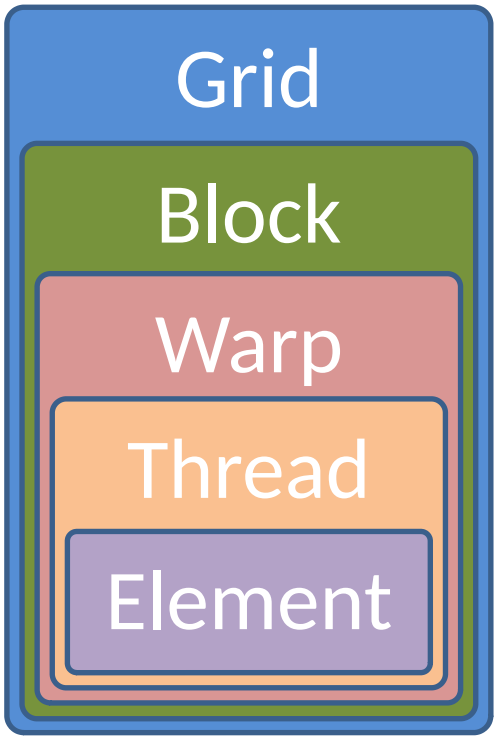
Code Example: Alpaka II

SIMD Optimized

```
struct DaxpyKernel
{
    template< typename T_Acc >
    ALPAKA_FN_ACC void operator()(
        T_Acc const & acc,
        double const & alpha,
        double const * const X,
        double * const Y,
        int const & numElements
    ) const
    {
        using alpaka;
        auto const globalIdx = idx::getIdx< Grid, Threads >( acc )[0u];
        auto const elemCount = workdiv::getWorkDiv< Thread, Elems >( acc )[0u];

        auto const begin = globalIdx * elemCount;
        auto const end = min( begin + elemCount, numElements );

        for( TSize i = begin; i < end; i++ )
            Y[i] = X[i] + Y[i]; // Note difference between worker and data index
    }
};
```

Explicit Programming Model

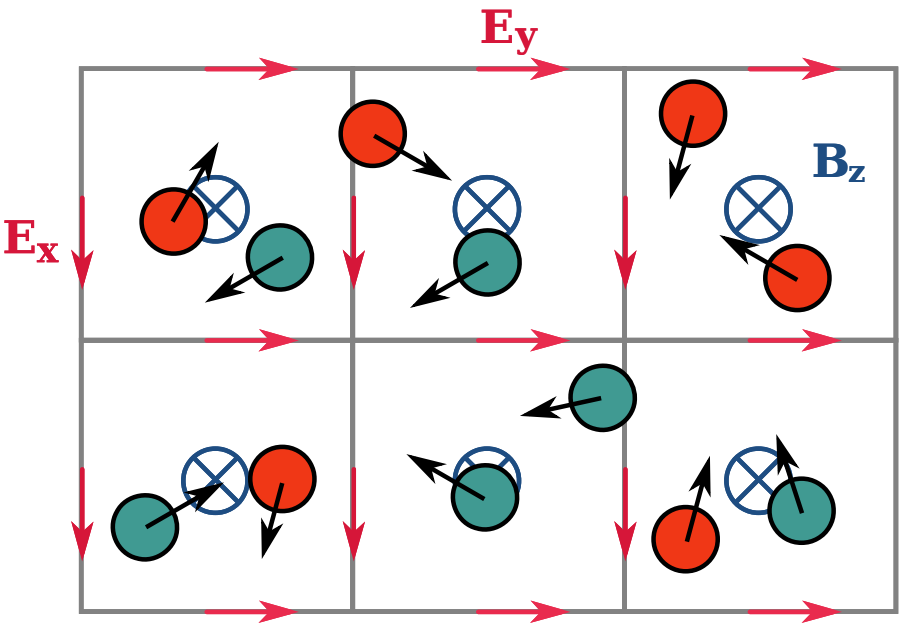
Alpaka



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

PIConGPU

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



7.2 PFlop/s (DP)
+ 1.4 PFlop/s (SP)

96% weak scaling efficiency

M. Bussmann et al., SC'13, DOI:10.1145/2503210.2504564