Core Concepts - Zero Overhead Abstractions for Scalable Many-Core Data Analysis

Erik Zenker

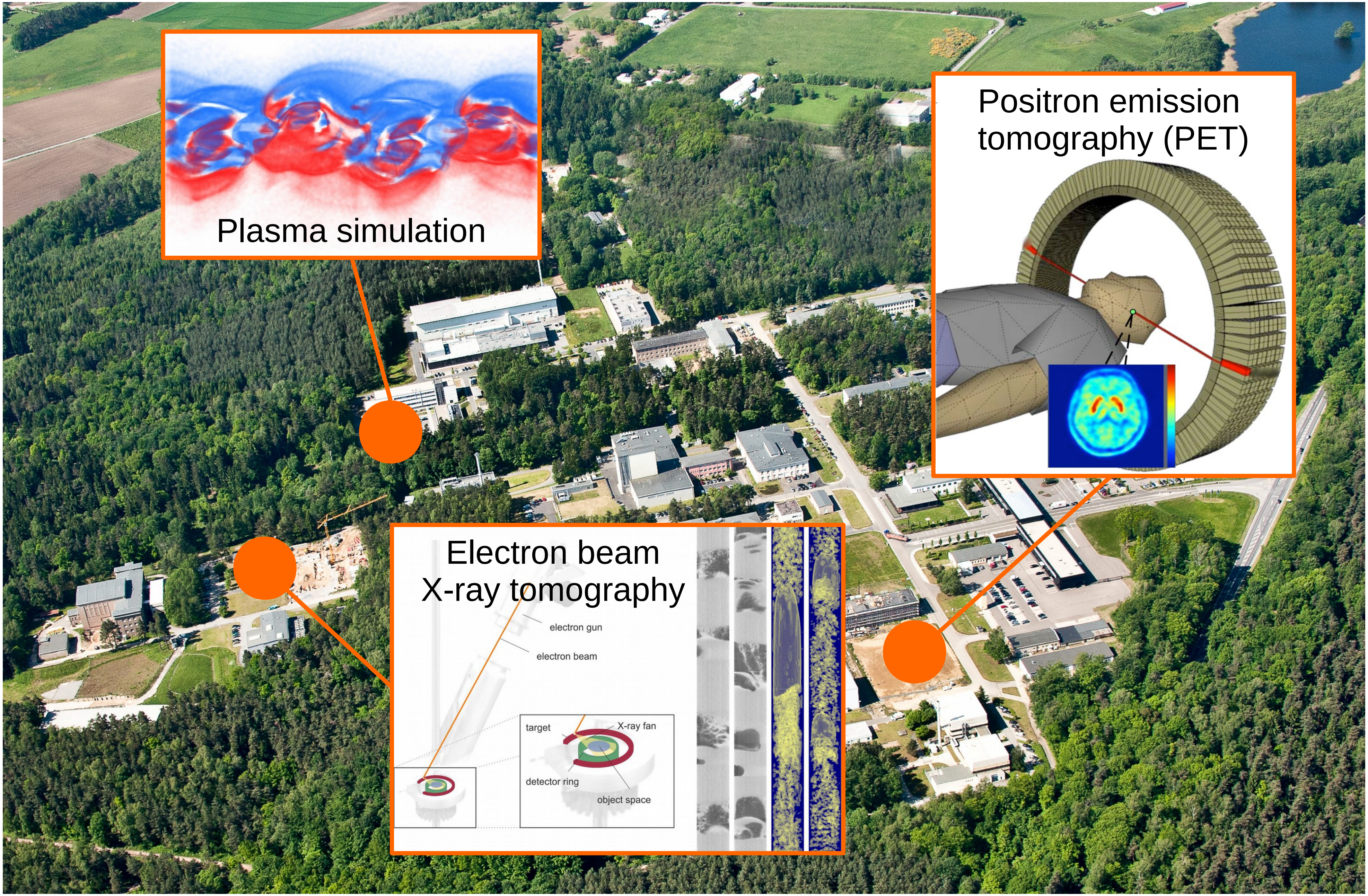
Computational Radiation Physics
Helmholtz-Center Dresden Rossendorf

TECHNISCHE UNIVERSITÄT DRESDEN

NVIDIA GPU CENTER OF EXCELLENCE

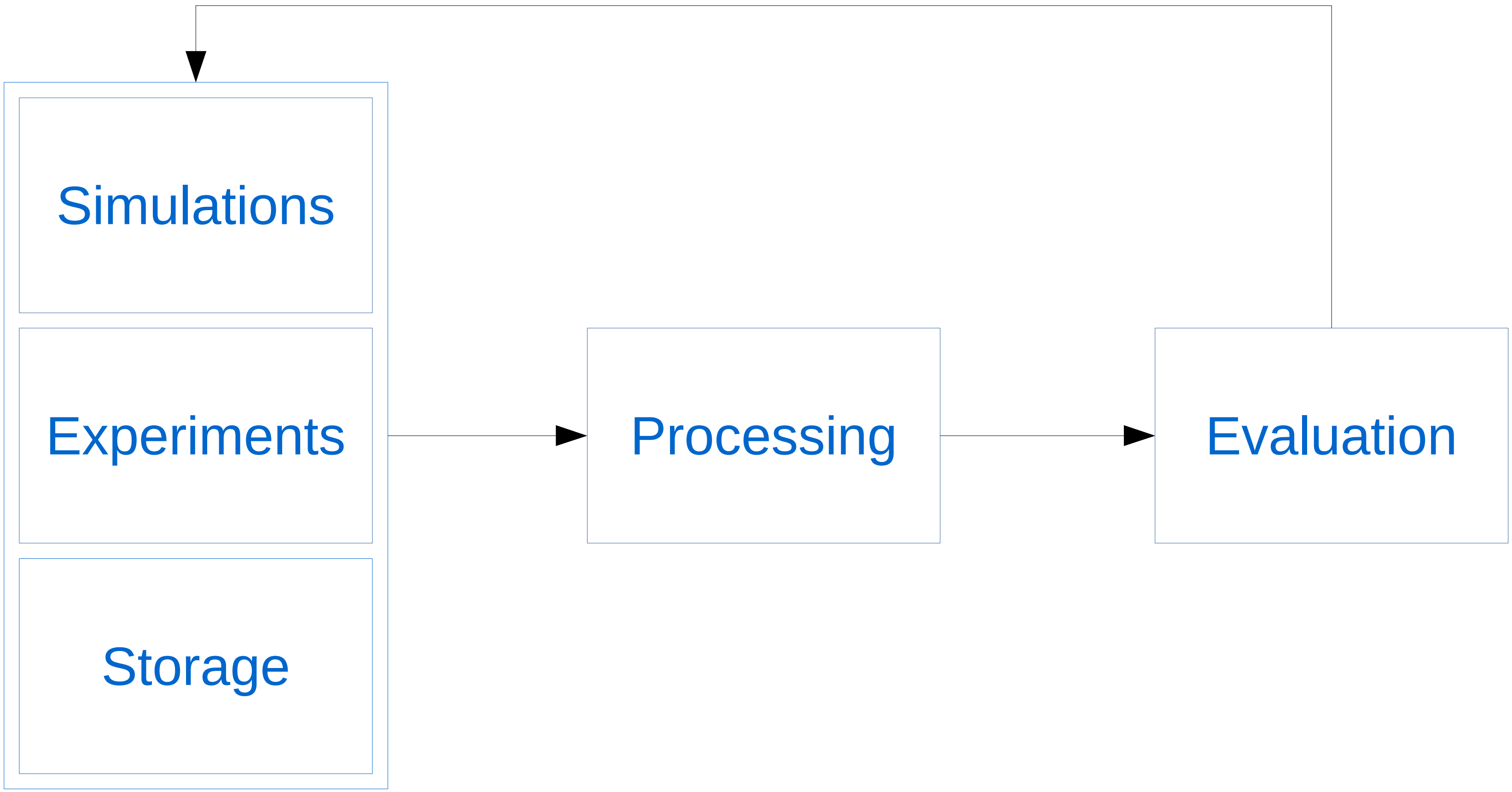
HZDR HELMHOLTZ ZENTRUM DRESDEN ROSSENDORF

Data is Everywhere



Erik Zenker · Computational Radiation Physics · www.hzdr.de/crp
e.zenker@hzdr.de

The Data Analysis Pipeline



The pipeline needs to scale with your data !

Erik Zenker · Computational Radiation Physics · www.hzdr.de/crp
e.zenker@hzdr.de

Problems to Solve

- Variable data rates
- Ondemand scaling
- Various communication protocols
- Heterogeneous cluster systems
  - GPU, MIC and CPU on the same node
  - Various network standards e.g. infiniband, ethernet
  - Hierarchical memory architecture
- Fast developing compute hardware and programming models

These Problems should not be solved all at once

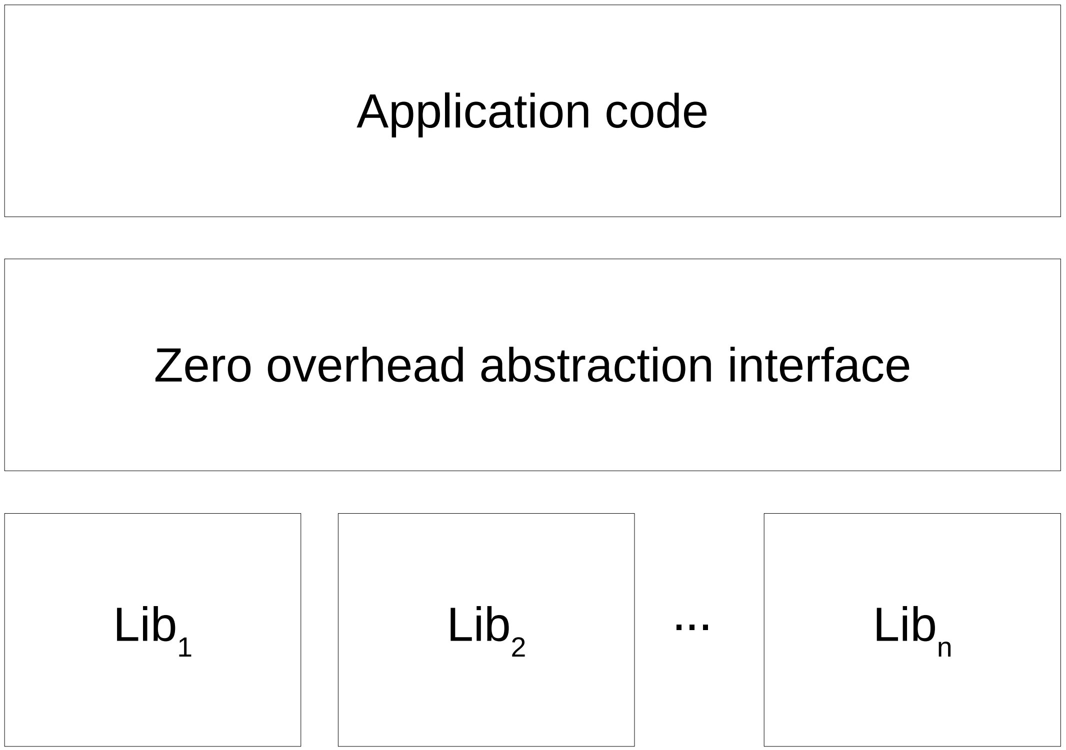
- Not maintainable, not scaleable, not sustainable
- Better solve them by independent building blocks

DRESDEN concept HZDR

Erik Zenker · Computational Radiation Physics · www.hzdr.de/crp
e.zenker@hzdr.de

For All That Already Exist Solutions

- We provide interfaces for existing solutions
- Each interface supports several solutions
- Each interface can be adapted to solutions
- Present today **Alpaka** **and** **Graybat**
- No overhead with zero overhead abstraction



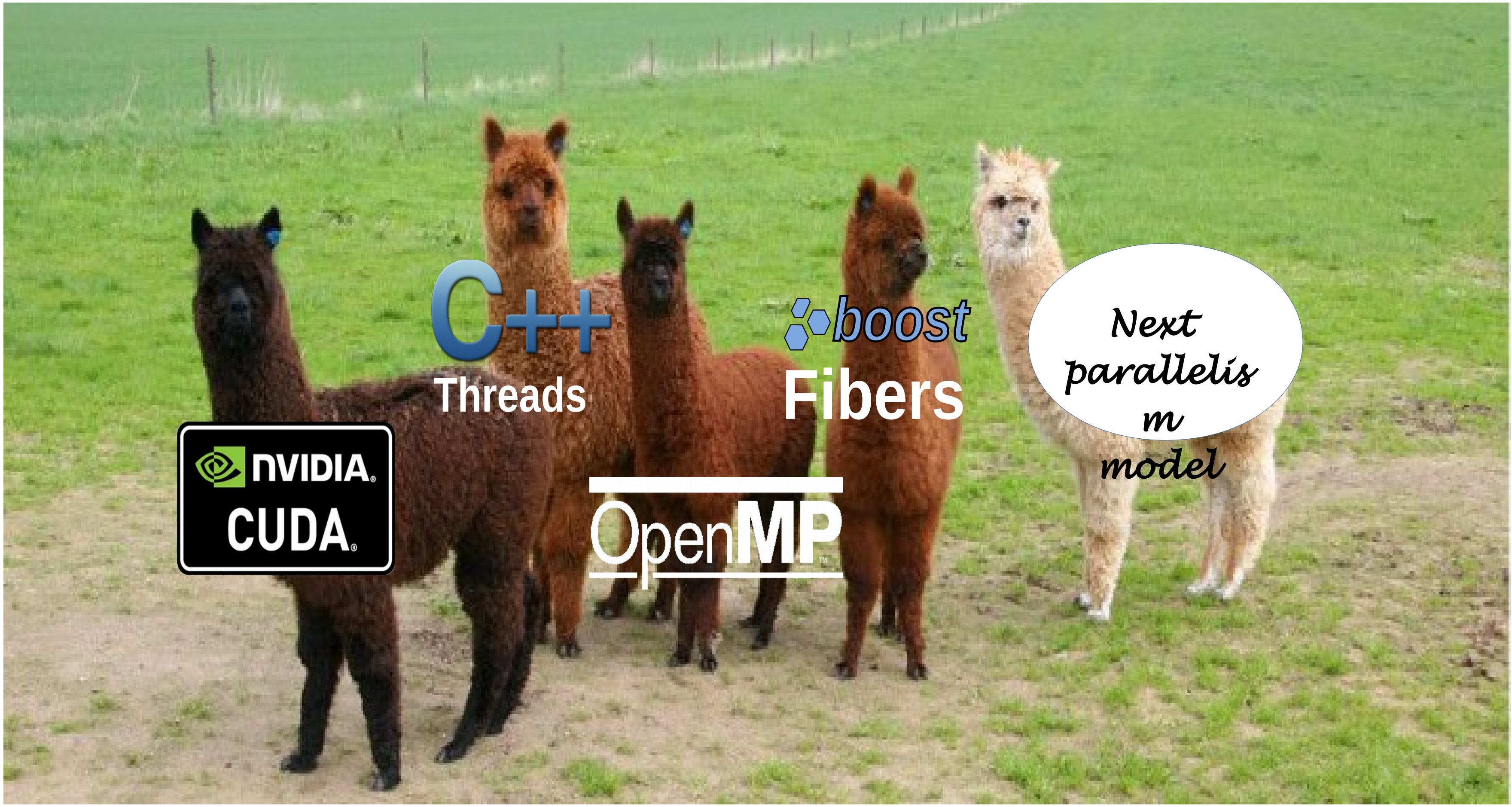
C++ compilers are able to almost completely remove abstraction layers

Erik Zenker · Computational Radiation Physics · www.hzdr.de/crp
e.zenker@hzdr.de

Alpaka

Erik Zenker · Computational Radiation Physics · www.hzdr.de/crp
e.zenker@hzdr.de

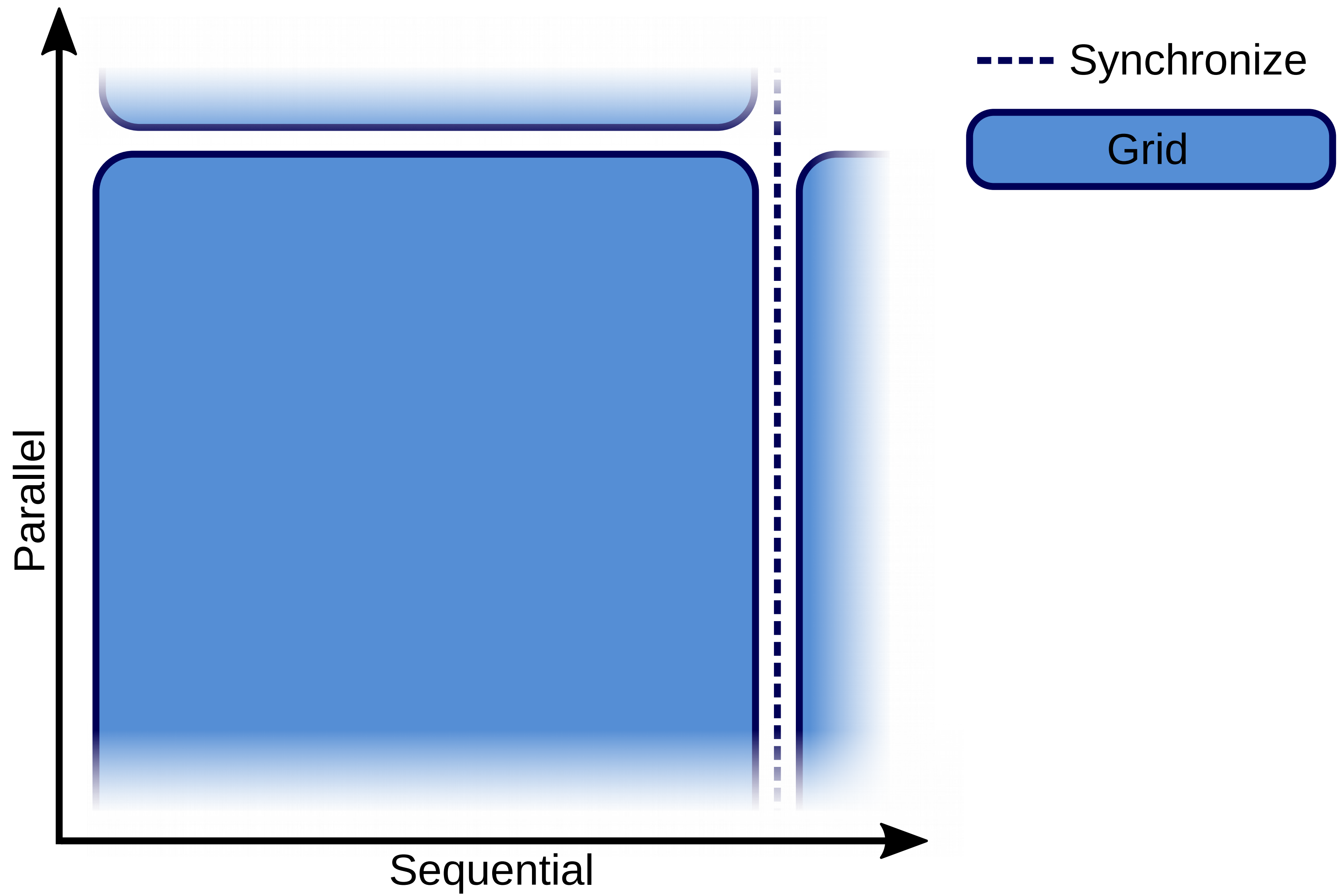
Alpaka: Parallel Kernel Execution



- Single abstract interface to existing parallelism models
- Single source C++ code using std::c++11
- Data-agnostic memory model

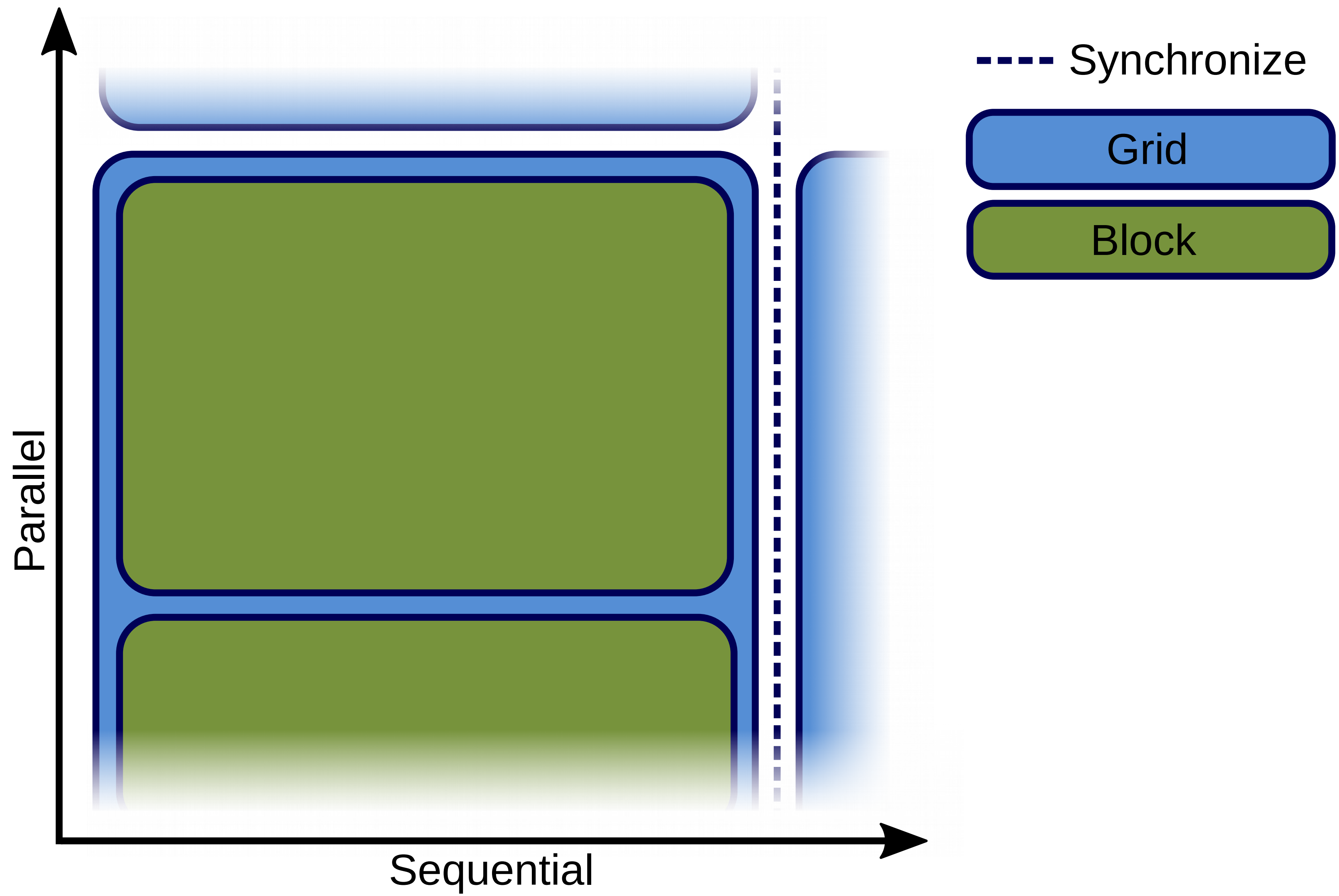
Erik Zenker · Computational Radiation Physics · www.hzdr.de/crp
e.zenker@hzdr.de

Abstract Hierarchical Redundant Parallelism Model



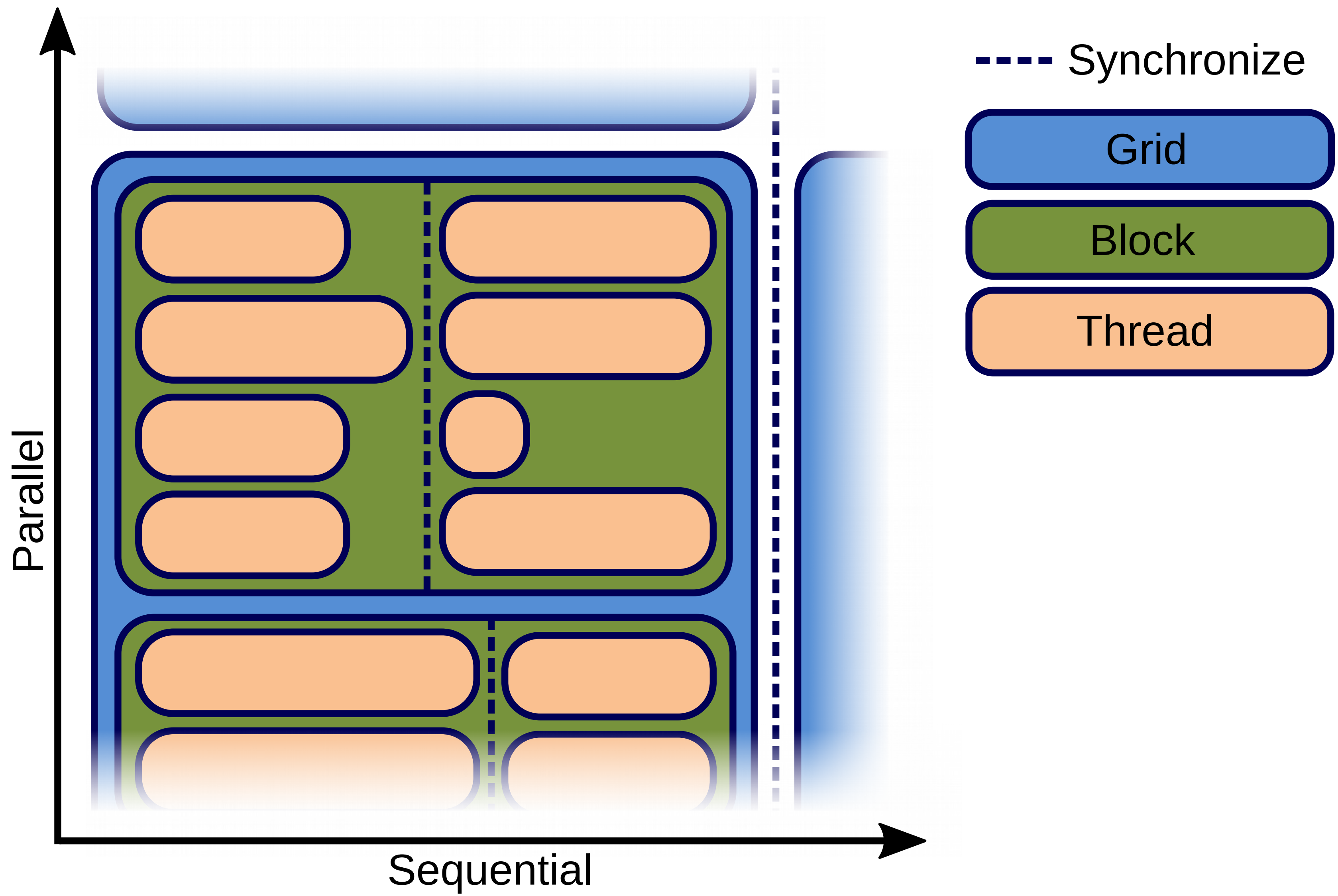
Erik Zenker · Computational Radiation Physics · www.hzdr.de/crp
e.zenker@hzdr.de

Abstract Hierarchical Redundant Parallelism Model



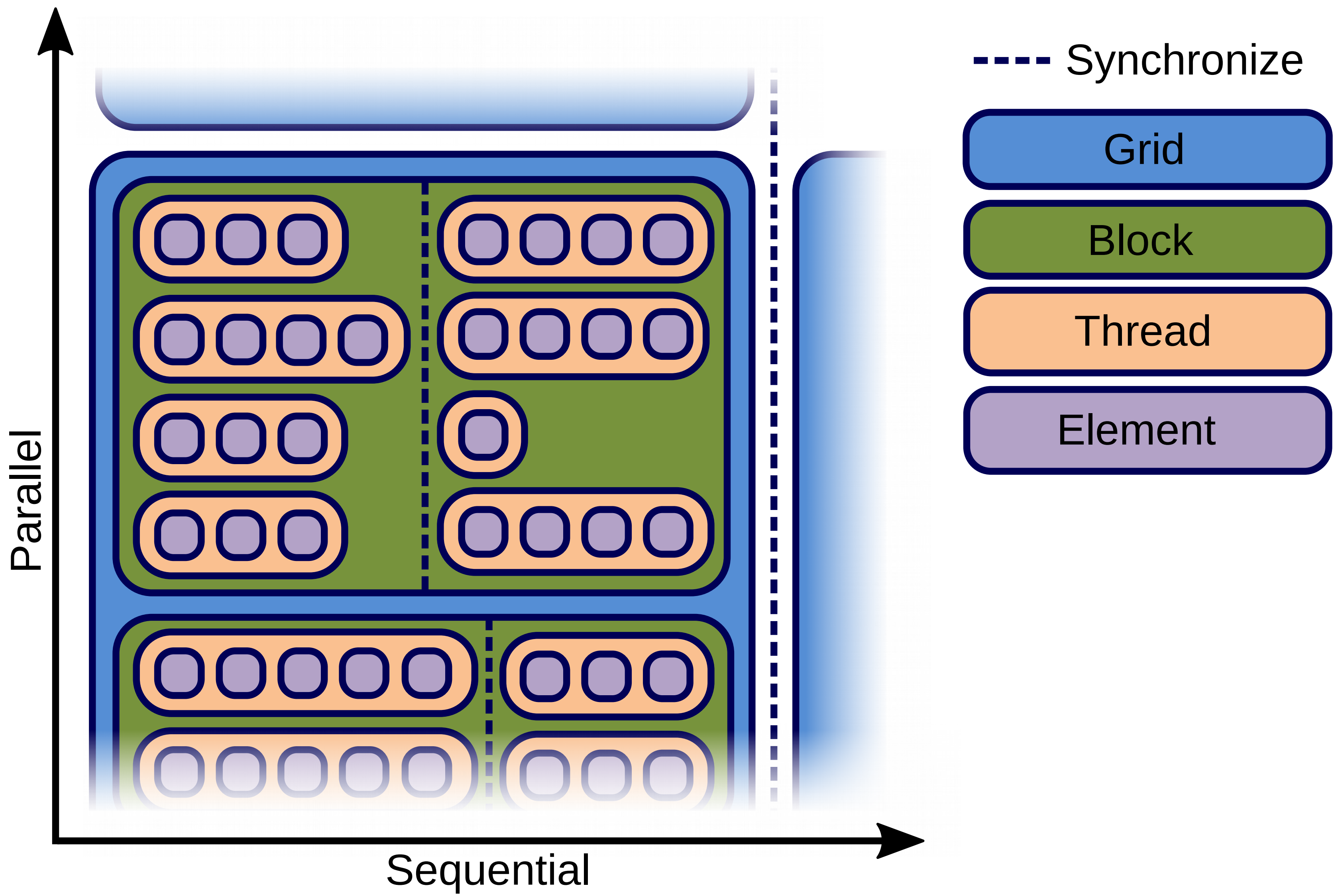
Erik Zenker · Computational Radiation Physics · www.hzdr.de/crp
e.zenker@hzdr.de

Abstract Hierarchical Redundant Parallelism Model



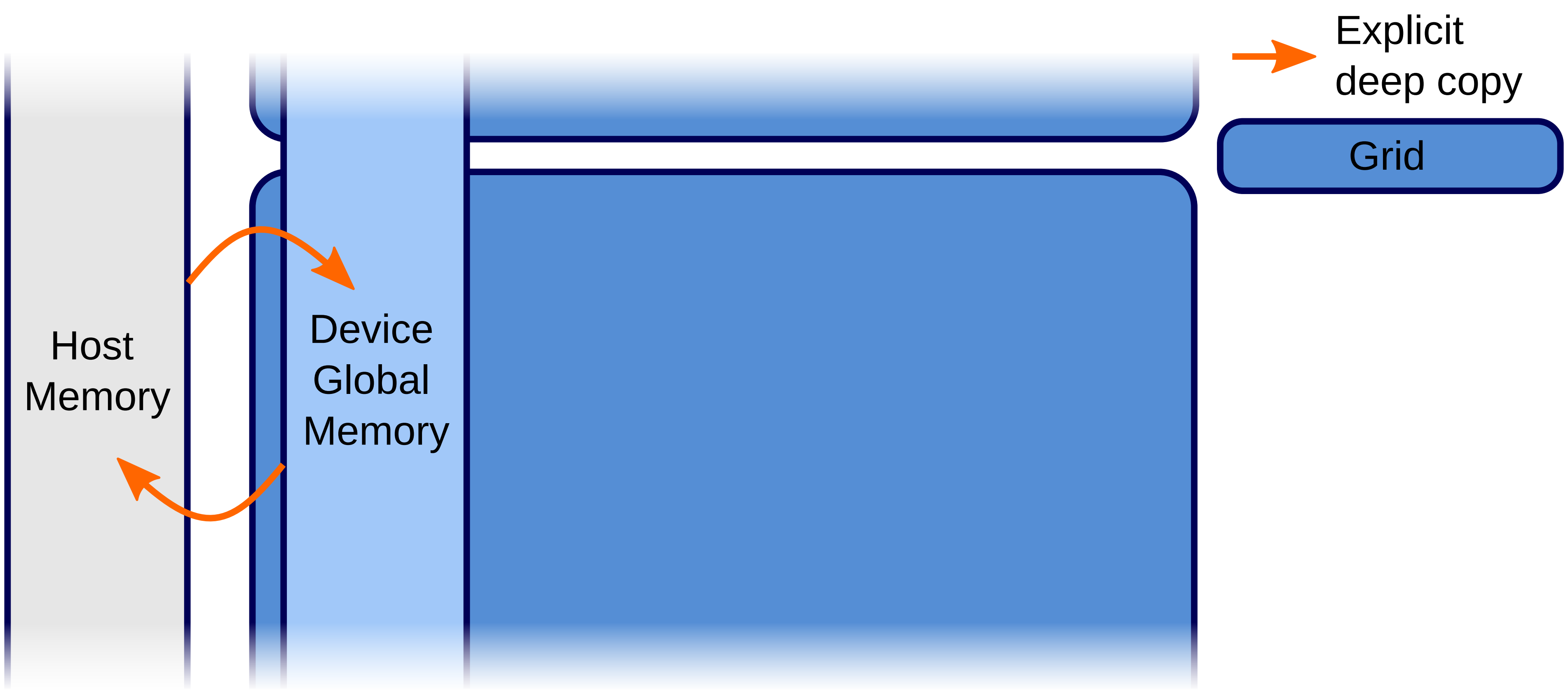
Erik Zenker · Computational Radiation Physics · www.hzdr.de/crp
e.zenker@hzdr.de

Abstract Hierarchical Redundant Parallelism Model



Erik Zenker · Computational Radiation Physics · www.hzdr.de/crp
e.zenker@hzdr.de

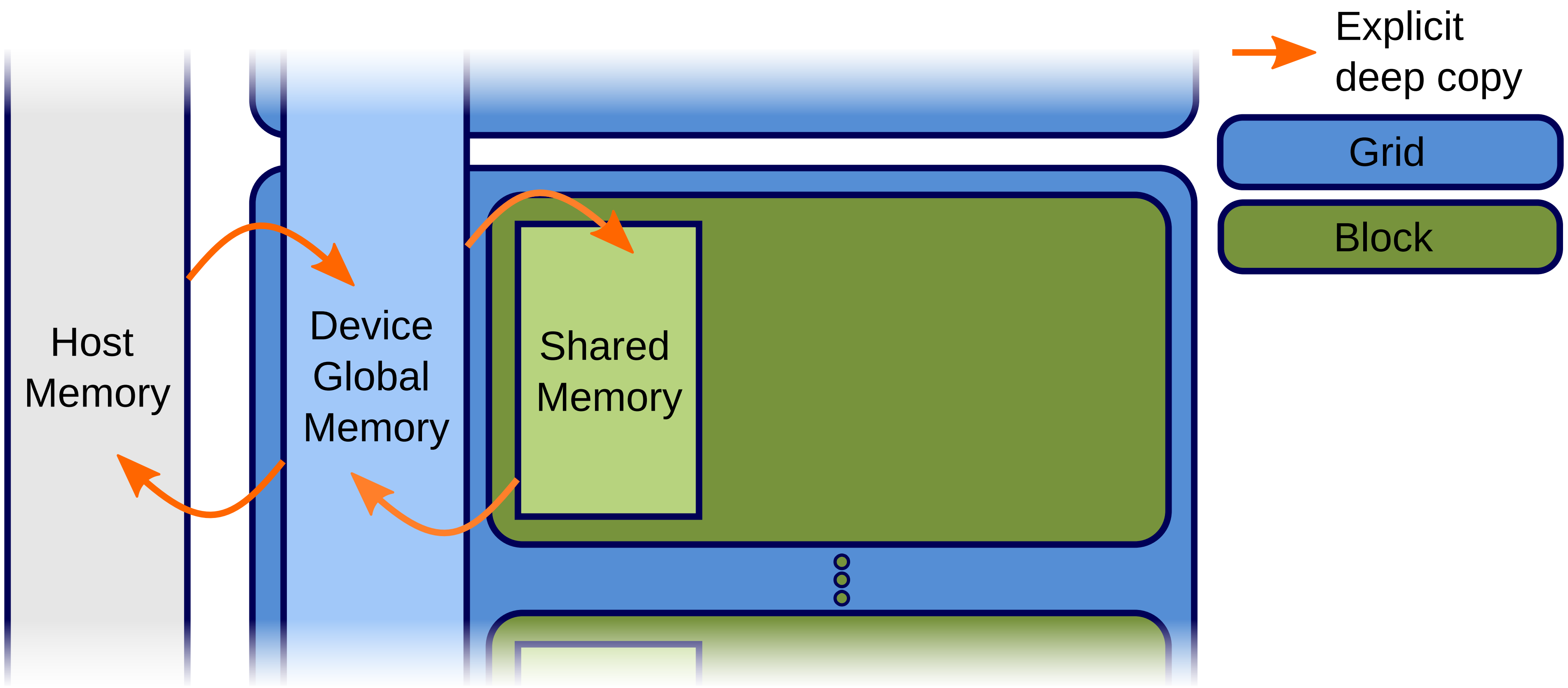
Data Structure Agnostic Memory Model



- Explicit deep copies between memory levels

Erik Zenker · Computational Radiation Physics · www.hzdr.de/crp
e.zenker@hzdr.de

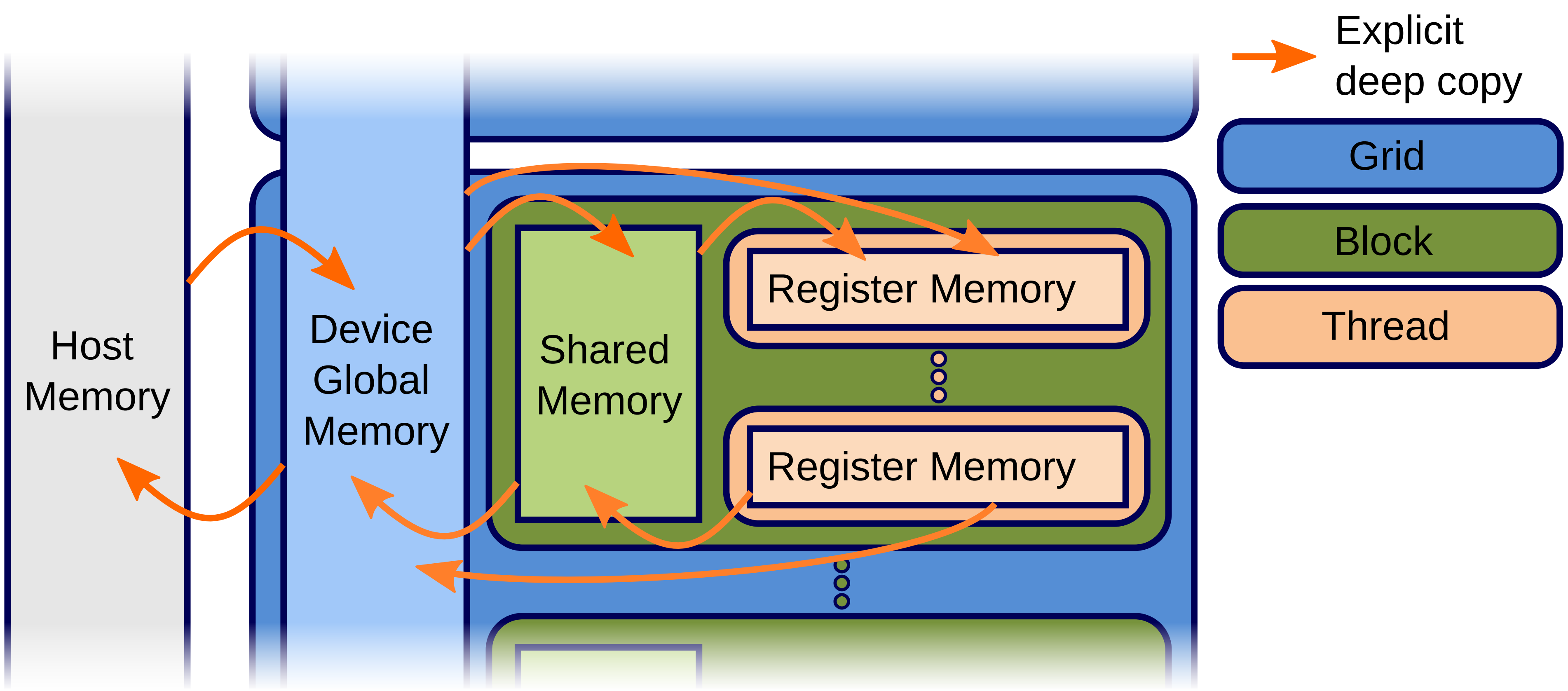
Data Structure Agnostic Memory Model



- Explicit deep copies between memory levels

Erik Zenker · Computational Radiation Physics · www.hzdr.de/crp
e.zenker@hzdr.de

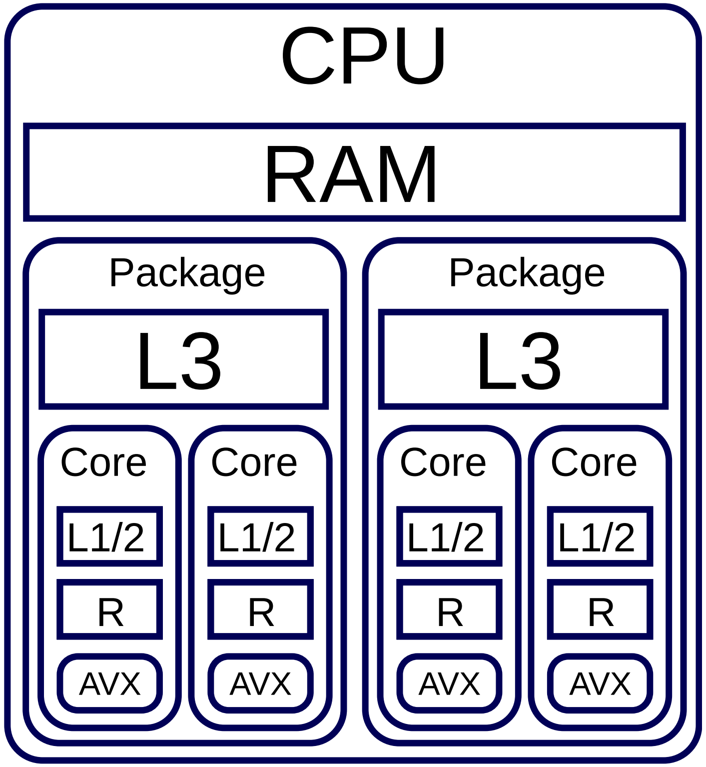
Data Structure Agnostic Memory Model



- Explicit deep copies between memory levels

Map the Abstraction Model to the Hardware

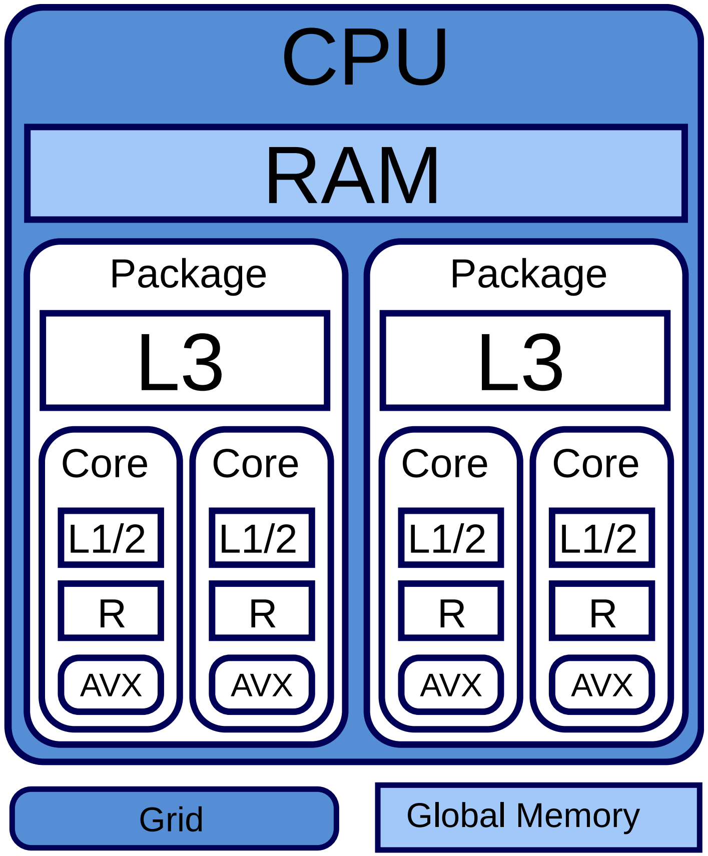
- Explicit mapping of parallelization levels to hardware



Erik Zenker · Computational Radiation Physics · www.hzdr.de/crp
e.zenker@hzdr.de

Map the Abstraction Model to the Hardware

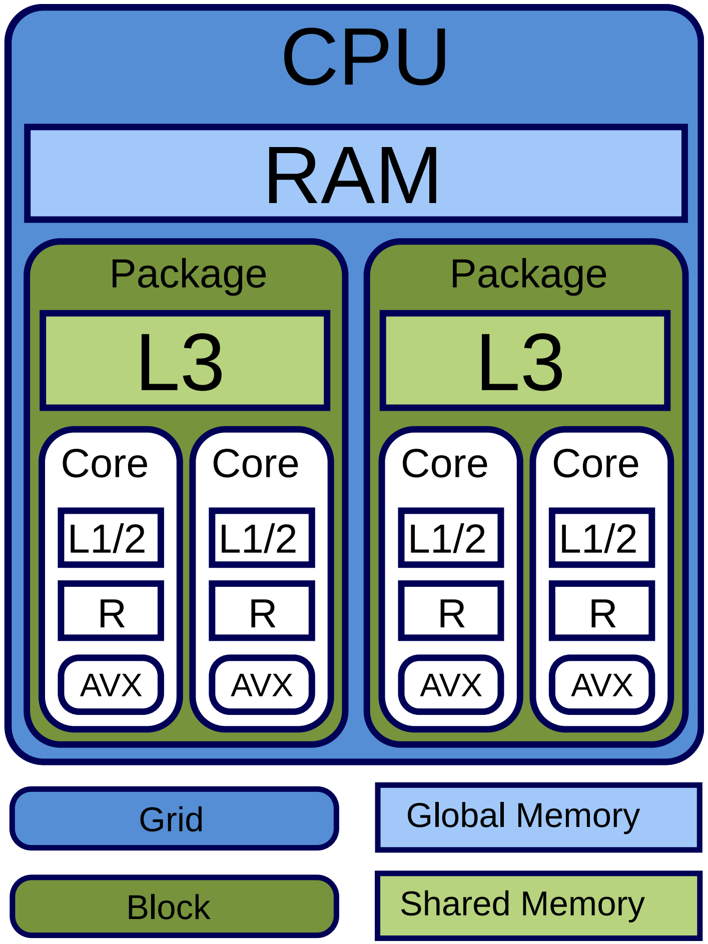
- Explicit mapping of parallelization levels to hardware



Erik Zenker · Computational Radiation Physics · www.hzdr.de/crp
e.zenker@hzdr.de

Map the Abstraction Model to the Hardware

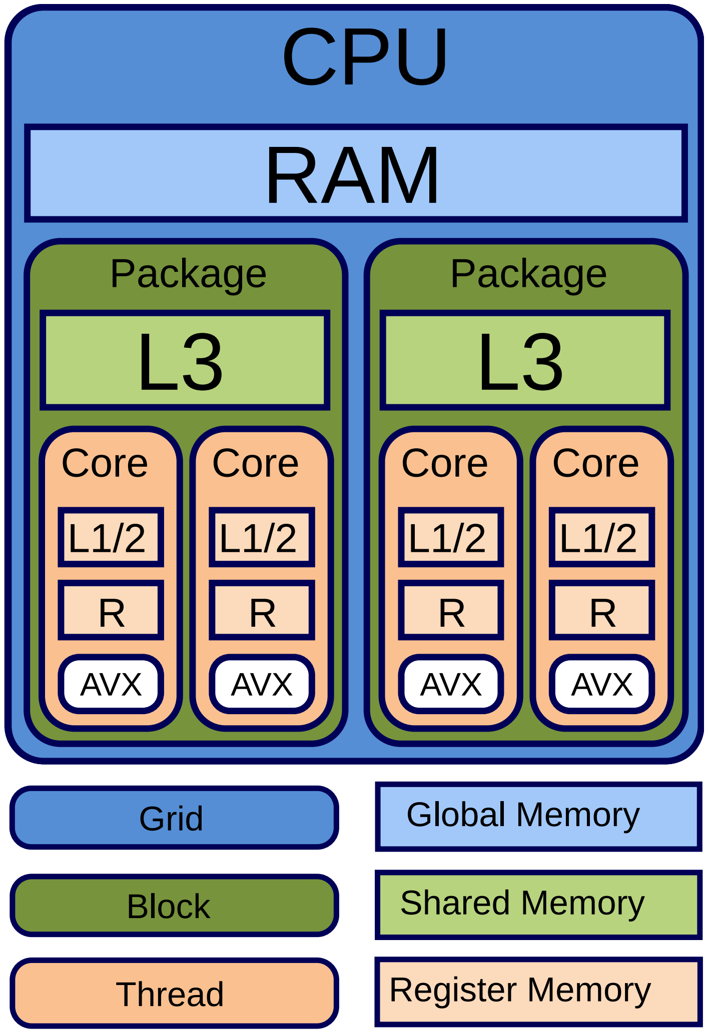
- Explicit mapping of parallelization levels to hardware



Erik Zenker · Computational Radiation Physics · www.hzdr.de/crp
e.zenker@hzdr.de

Map the Abstraction Model to the Hardware

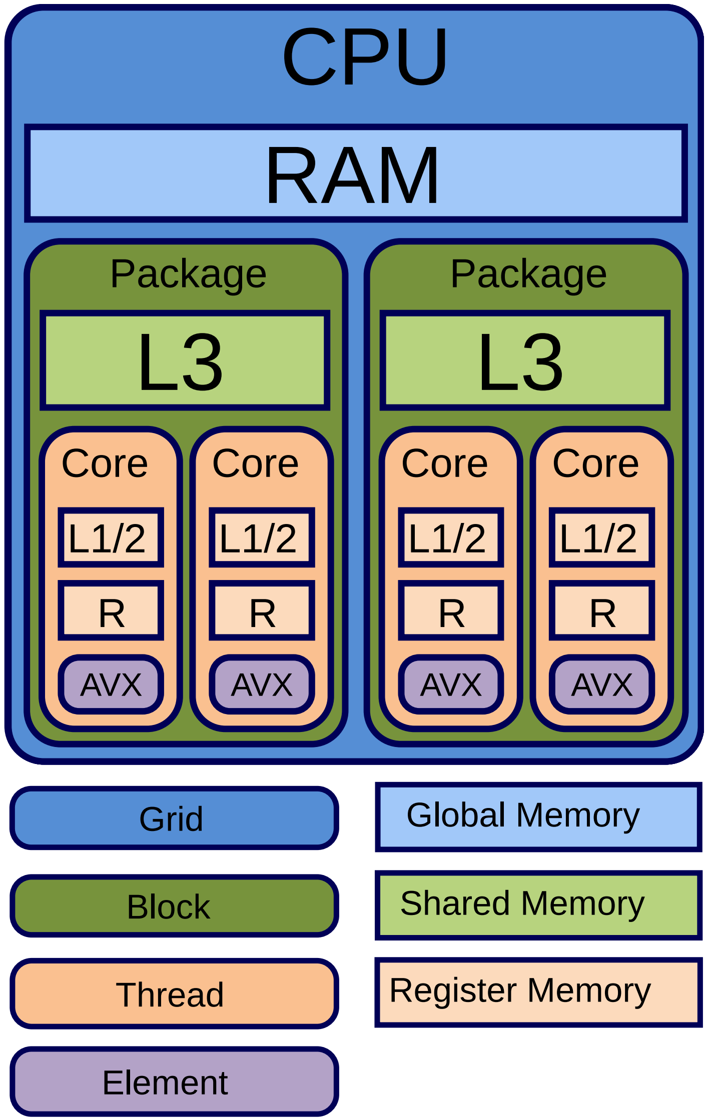
- Explicit mapping of parallelization levels to hardware



Erik Zenker · Computational Radiation Physics · www.hzdr.de/crp
e.zenker@hzdr.de

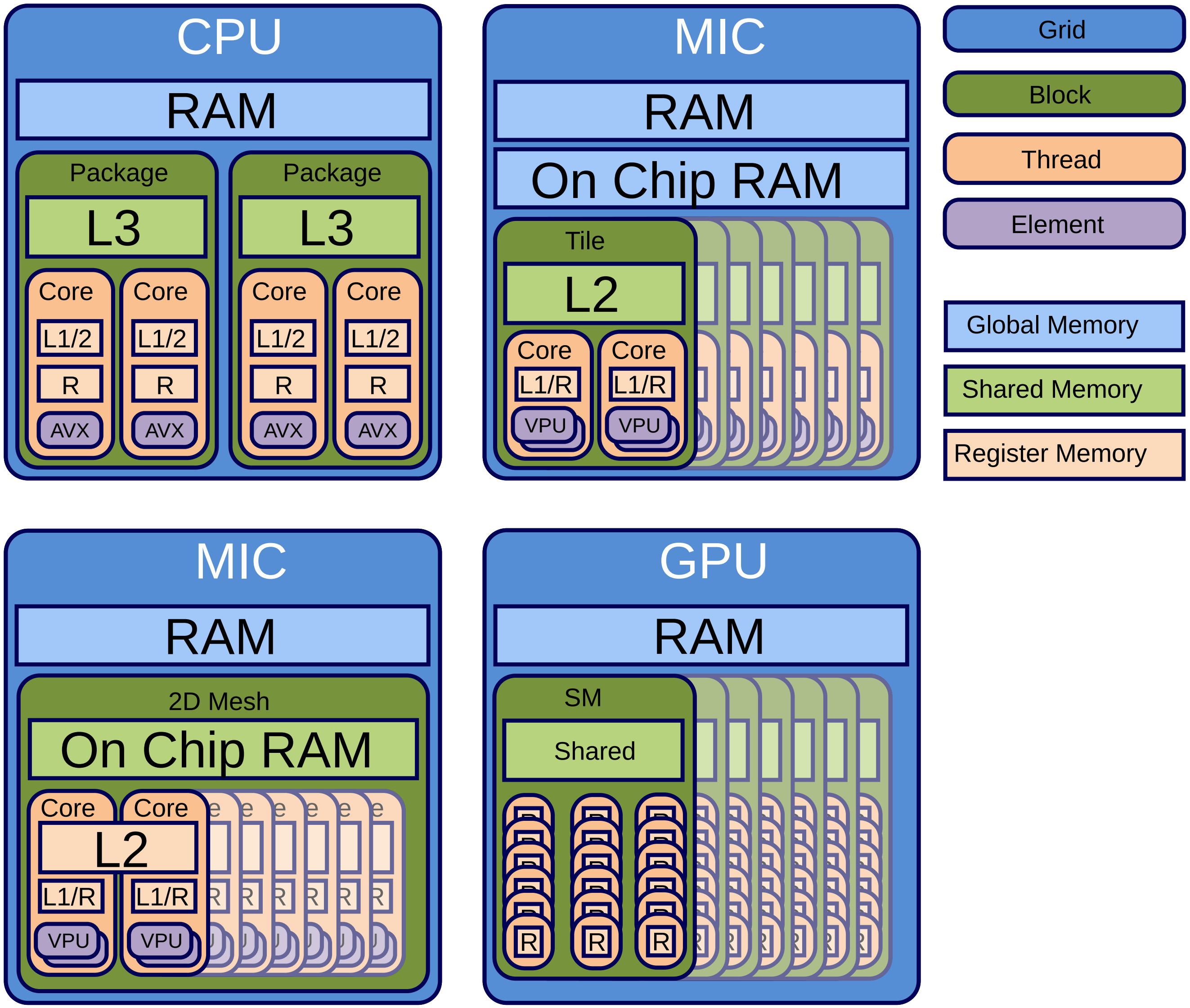
Map the Abstraction Model to the Hardware

- Explicit mapping of parallelization levels to hardware



Erik Zenker · Computational Radiation Physics · www.hzdr.de/crp
e.zenker@hzdr.de

Mapping Allows to Go for Various Architectures



Erik Zenker · Computational Radiation Physics · www.hzdr.de/crp
e.zenker@hzdr.de

How Does it Feel ?

```
struct Kernel
{
    template< typename TAcc>
    ALPAKA_FN_ACC auto operator()(TAcc const & acc) const
    -> void
    {
        // Do your parallel calculation here
    }
};
```

```
using Acc = alpaka::acc::AccGpuCudaRt<Dim, Size>;
```

```
WorkDiv workdiv(alpaka::workdiv::WorkDivMembers<Dim, Size>(blocksPerGrid,
                                                           threadsPerBlock,
                                                           elementsPerThread));
```

```
Kernel kernel;

auto const exec(alpaka::exec::create<Acc>(
    workDiv,
    kernel);

alpaka::stream::enqueue(stream, exec);
```


Erik Zenker · Computational Radiation Physics · www.hzdr.de/crp
e.zenker@hzdr.de

Compile to Almost Same PTX Code (DAXPY)

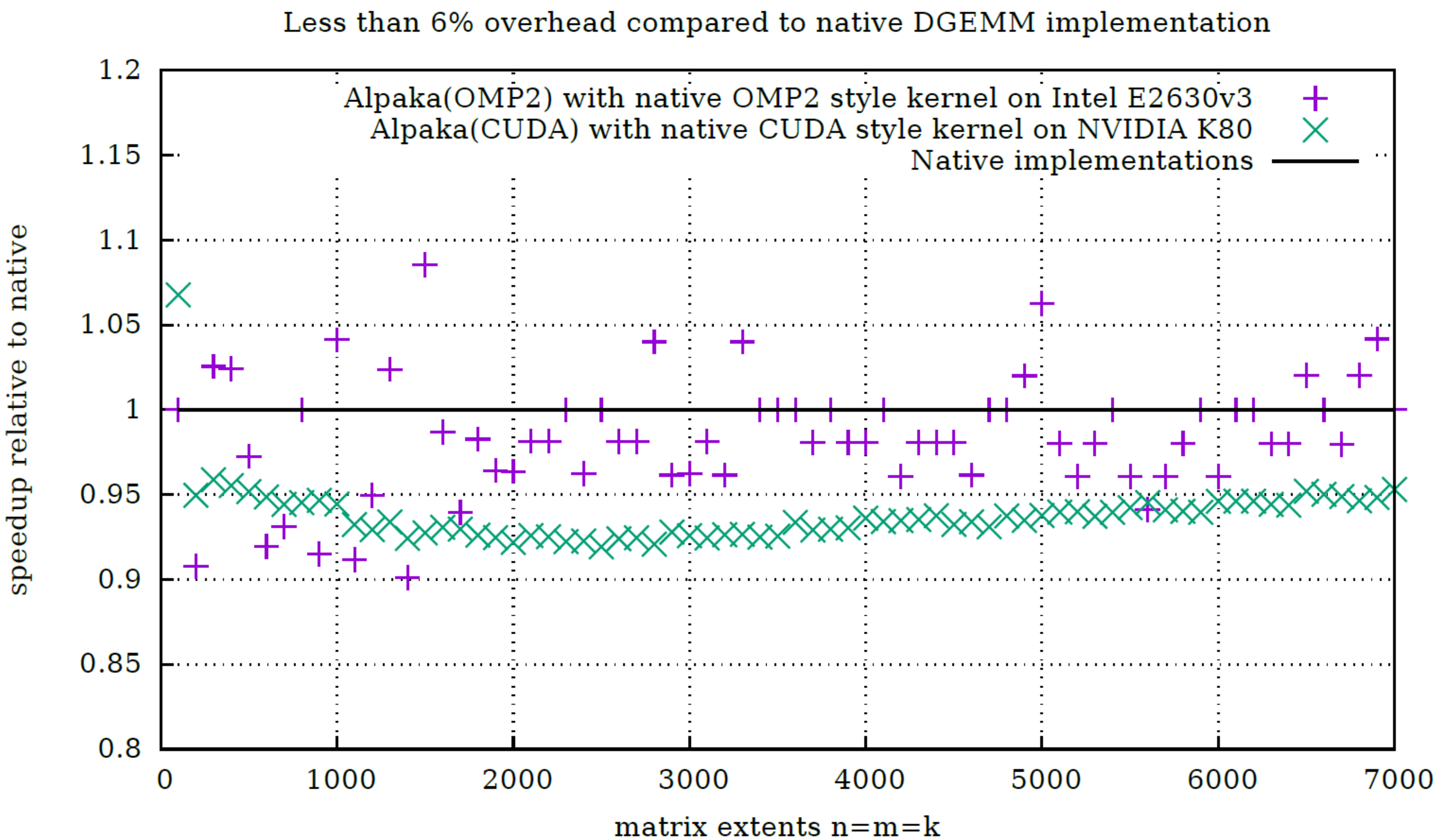
Alpaka CUDA PTX

```
mov.u32      %r3, %ctaid.x;
mov.u32      %r4, %ntid.x;
mov.u32      %r5, %tid.x;
mad.lo.s32   %r1, %r4, %r3, %r5;
setp.ge.s32 %p1, %r1, %r2;
@%p1 bra   BB6_2;


cvta.to.global.u64   %rd3, %rd2;
cvta.to.global.u64   %rd4, %rd1;
mul.wide.s32         %rd5, %r1, 8;
add.s64              %rd6, %rd4, %rd5;
ld.global.f64        %fd2, [%rd6];
add.s64              %rd7, %rd3, %rd5;
ld.global.f64        %fd3, [%rd7];
fma.rn.f64           %fd4, %fd2, %fd1, %fd3;
st.global.f64       [%rd7], %fd4;
```

Native CUDA PTX

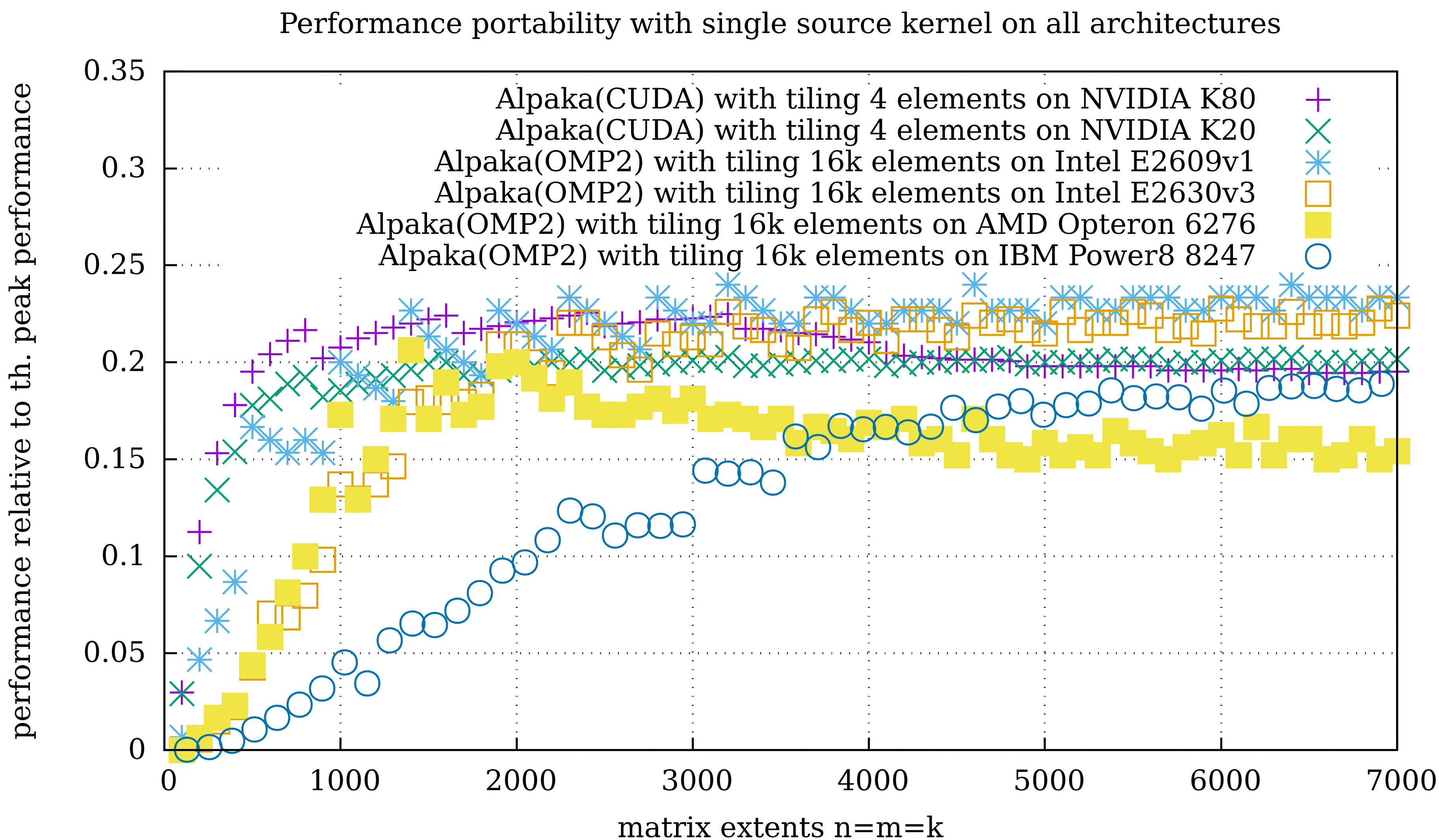
```
mov.u32      %r3, %ctaid.x;
mov.u32      %r4, %ntid.x;
mov.u32      %r5, %tid.x;
mad.lo.s32   %r1, %r4, %r3, %r5;
setp.ge.s32 %p1, %r1, %r2;
@%p1 bra   BB6_2;


cvta.to.global.u64   %rd3, %rd2;
cvta.to.global.u64   %rd4, %rd1;
mul.wide.s32         %rd5, %r1, 8;
add.s64              %rd6, %rd4, %rd5;
ld.global.nc.f64     %fd2, [%rd6];
add.s64              %rd7, %rd3, %rd5;
ld.global.f64        %fd3, [%rd7];
fma.rn.f64           %fd4, %fd2, %fd1, %fd3;
st.global.f64       [%rd7], %fd4;
```

Almost Zero Overhead (DGEMM)



Erik Zenker · Computational Radiation Physics · www.hzdr.de/crp
e.zenker@hzdr.de

Performance Portable (DGEMM)



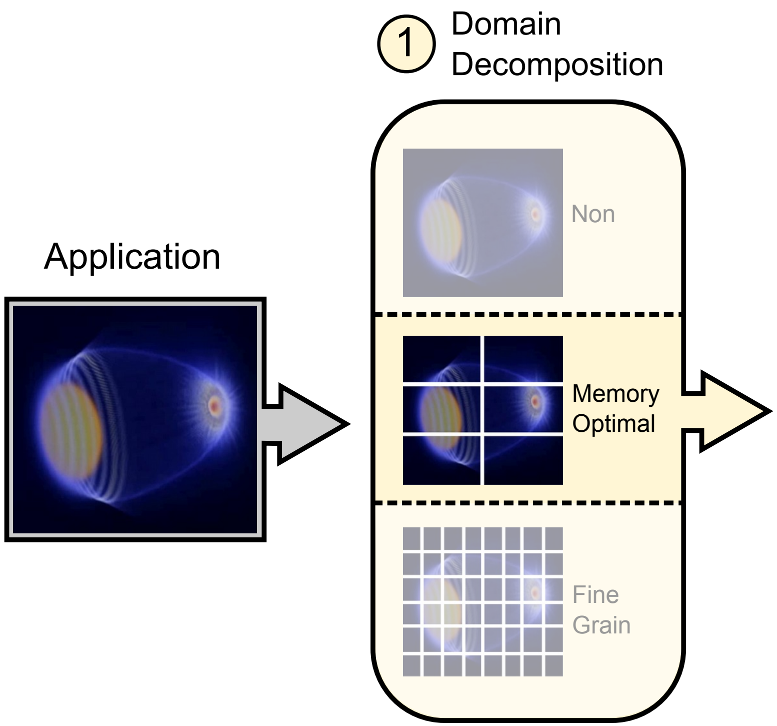
DRESDEN concept

Erik Zenker · Computational Radiation Physics · www.hzdr.de/crp
e.zenker@hzdr.de

GrayBat

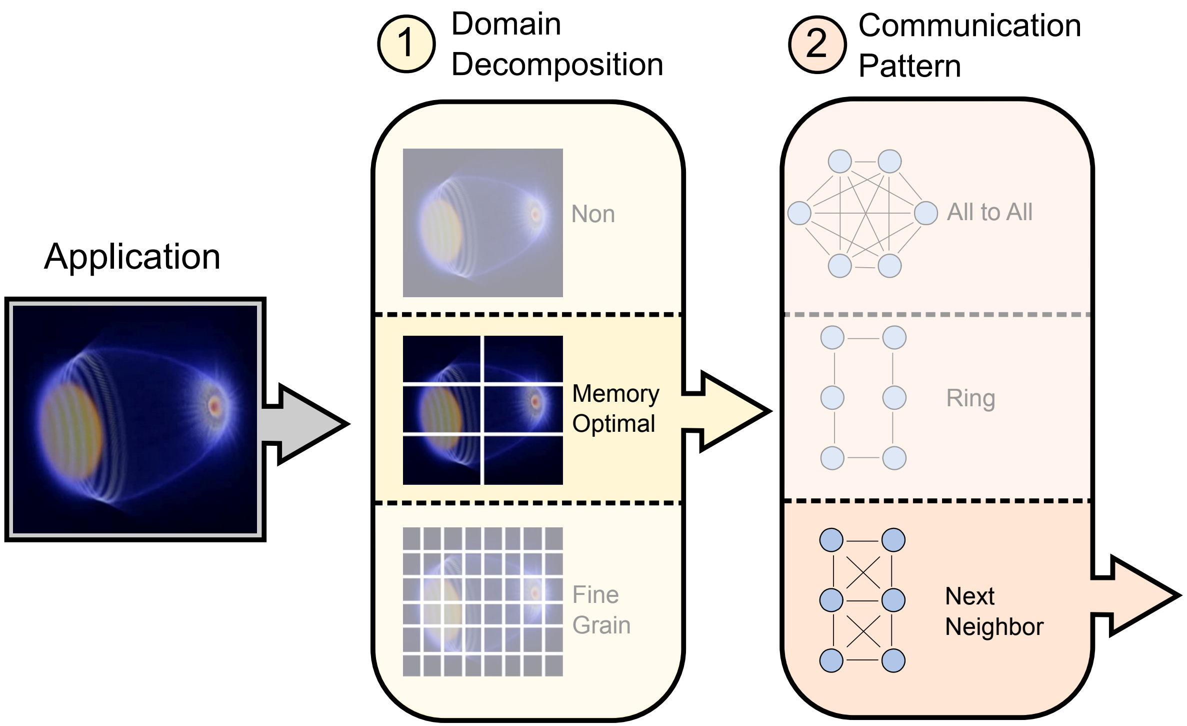
e.zenker@hzdr.de

GrayBat: Graph-Based Communication Approach



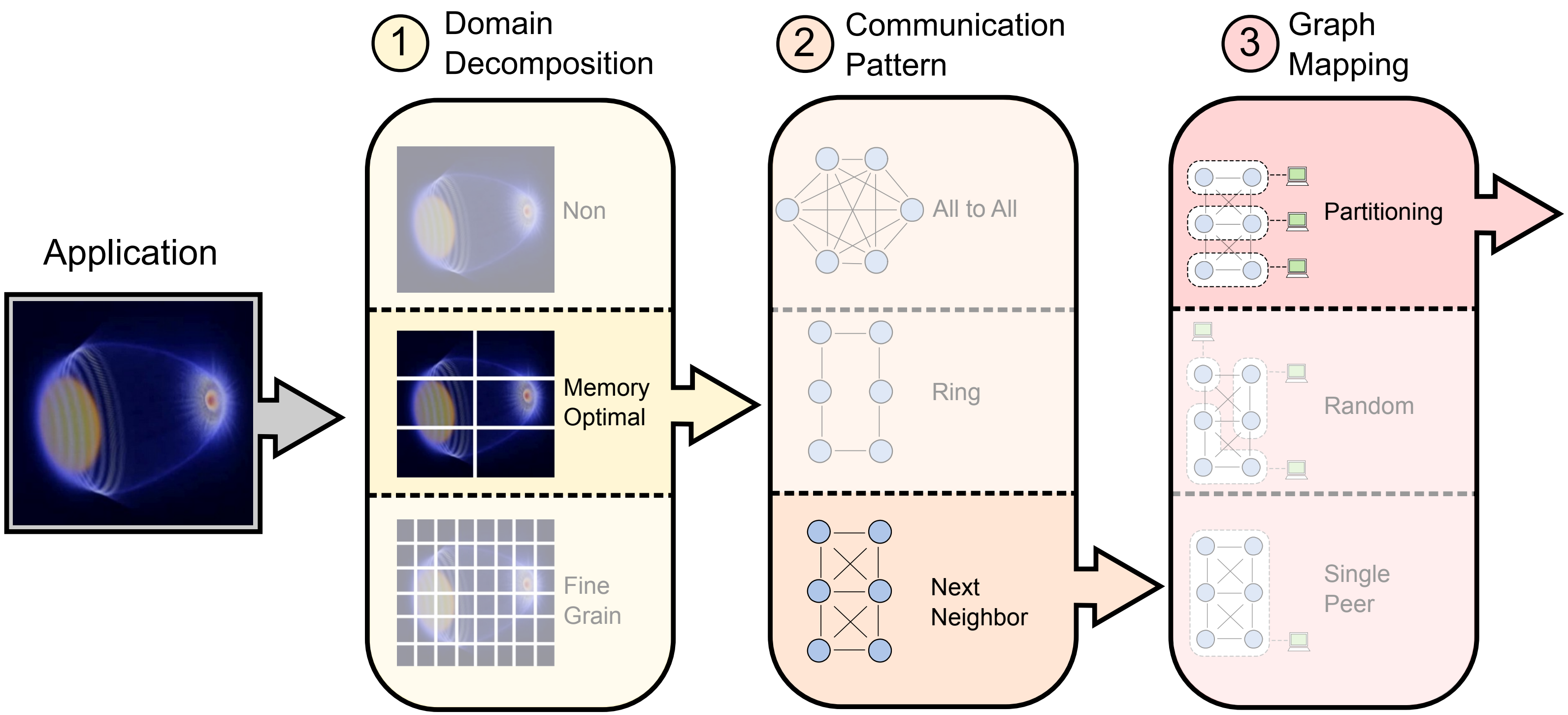
Erik Zenker · Computational Radiation Physics · www.hzdr.de/crp
e.zenker@hzdr.de

GrayBat: Graph-Based Communication Approach

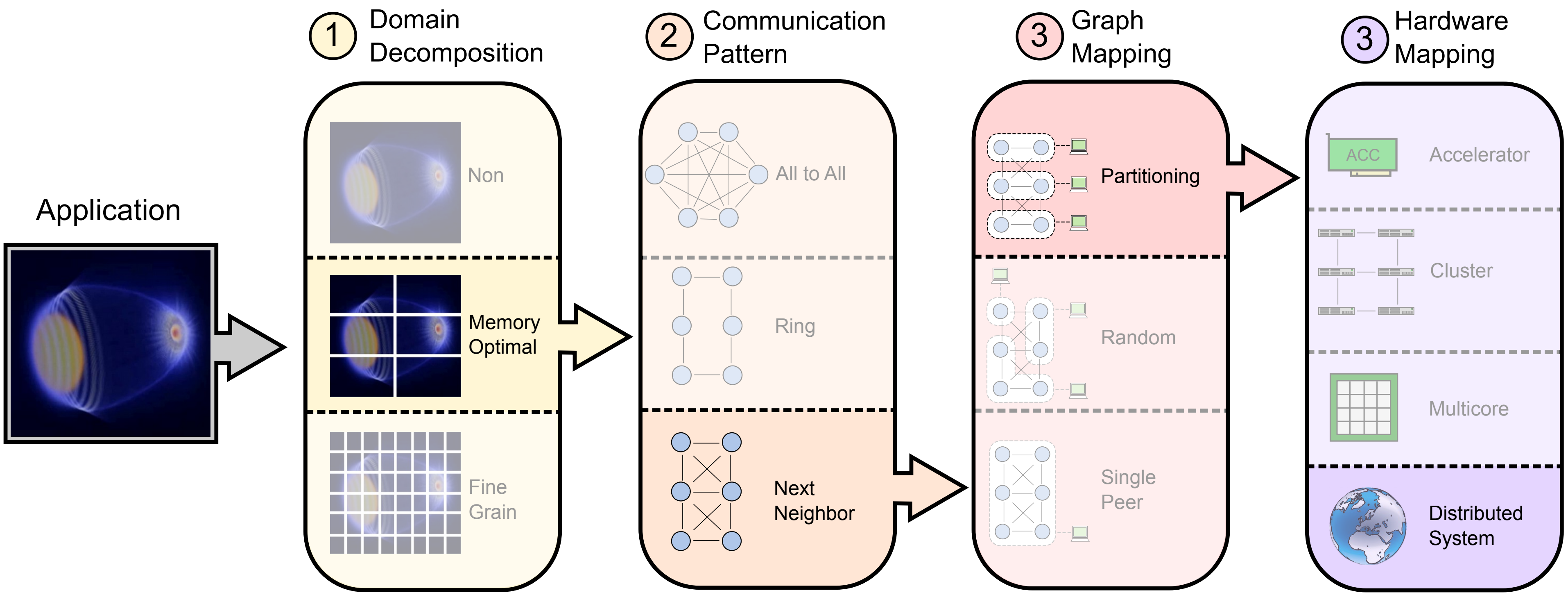


Erik Zenker · Computational Radiation Physics · www.hzdr.de/crp
e.zenker@hzdr.de

GrayBat: Graph-Based Communication Approach



GrayBat: Graph-Based Communication Approach

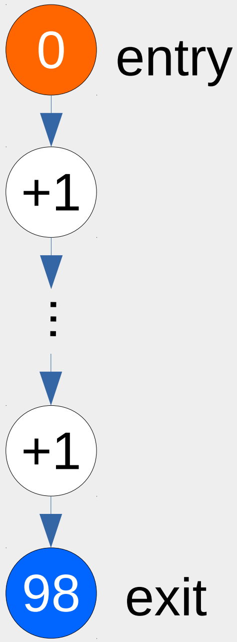


Lets Have a Short Look Into the Code

```
using CP   = graybat::communicationPolicy::BMPI;
using GP   = graybat::graphPolicy::BGL<Property>;
using Cage = graybat::Cage<CP, GP> Cage;
```

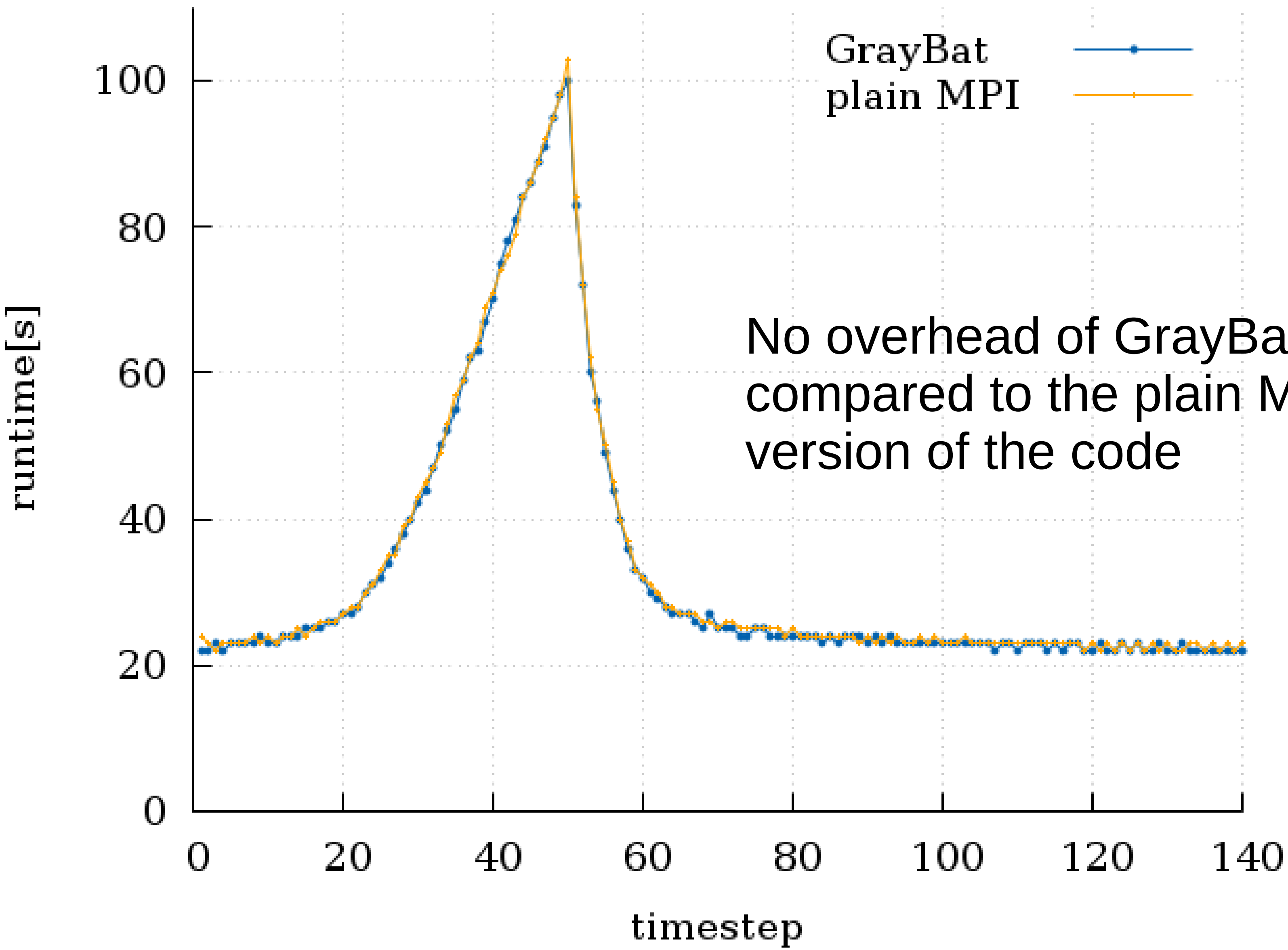
```
Cage cage;
cage.setGraph  (graybat::pattern::Chain<GP>(100));
cage.distribute(graybat::mapping::Consecutive());
```

```
std::array<unsigned, 1> input, output, intermediate;

const Vertex entry = cage.getVertex(0);
const Vertex exit  = cage.getVertex(99);

for(Vertex v : cage.hostedVertices){

    if(v == entry)
        v.spread(input, events);

    if(v != entry and v != exit)
        v.forward(intermediate, std::plus<int>());

    if(v == exit)
        v.collect(output);
}
```

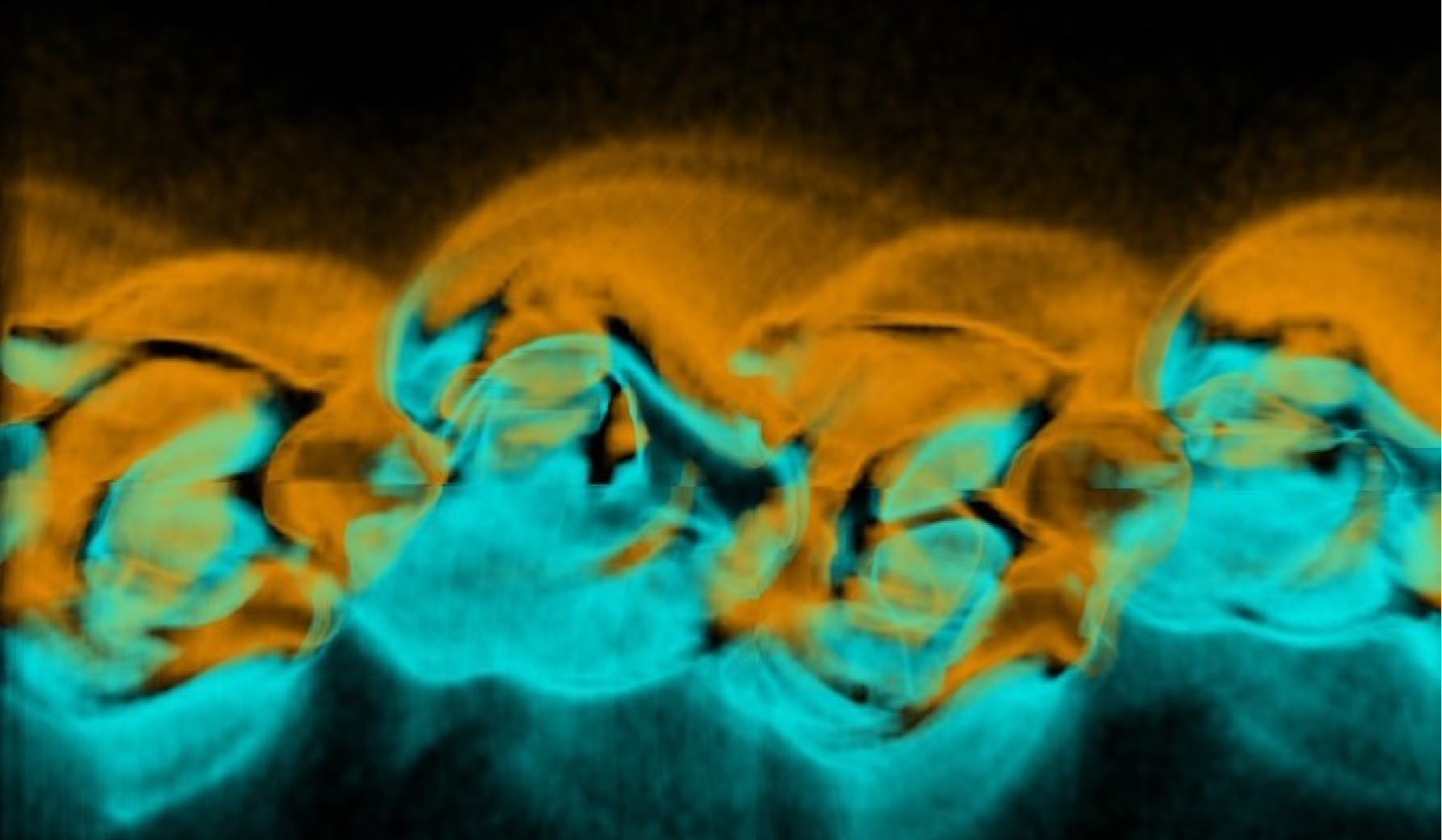


Erik Zenker · Computational Radiation Physics · www.hzdr.de/crp
e.zenker@hzdr.de

Usage in a Real World Code (HASEonGPU)



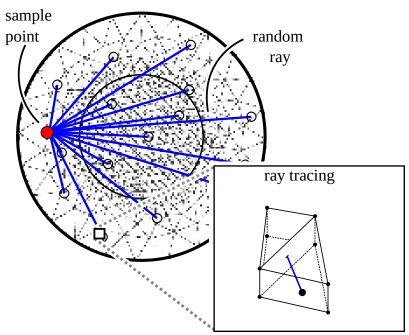
No overhead of GrayBat compared to the plain MPI version of the code

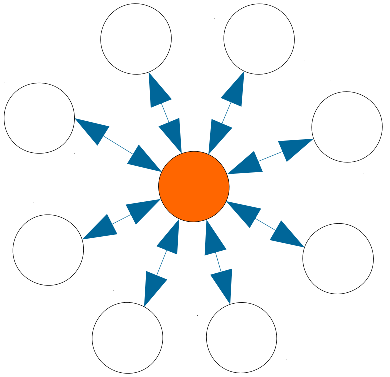


We are Using these Methods Successfully

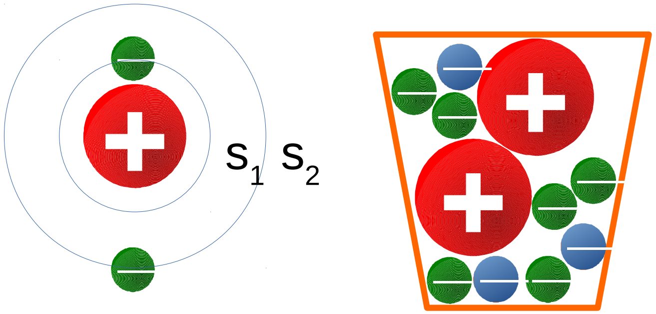
Sucessful Utilized

- **HASEonGPU** – Multi Node Multi GPU ASE Simulation
  - Execution on GPU or CPU without source duplication by Alpaka
  - Dynamic work distribution based on MPI or ZeroMQ communicaiton by Graybat
  - Dynamic work distribution to CPU and GPU in the same application



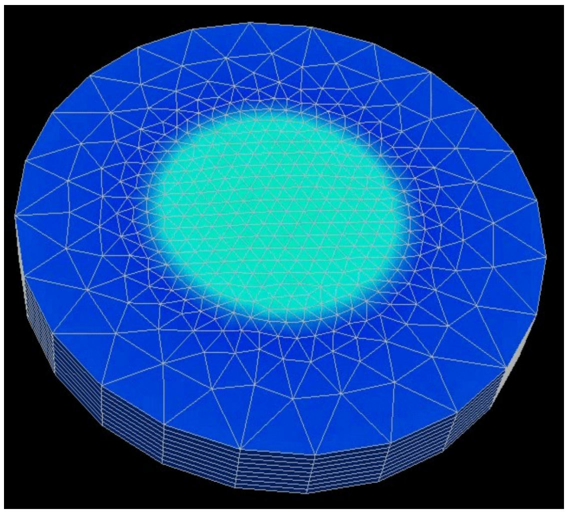


- **PIConGPU** – Many GPGPU Particle-In-Cell Code
  - Additional usage of CPUs on I/O intensive tasks by Alpaka
  - Fast dynamic memory allocation for multi species simulations by MallocMC
  - More complex physical simulations and better data analysis capabilities

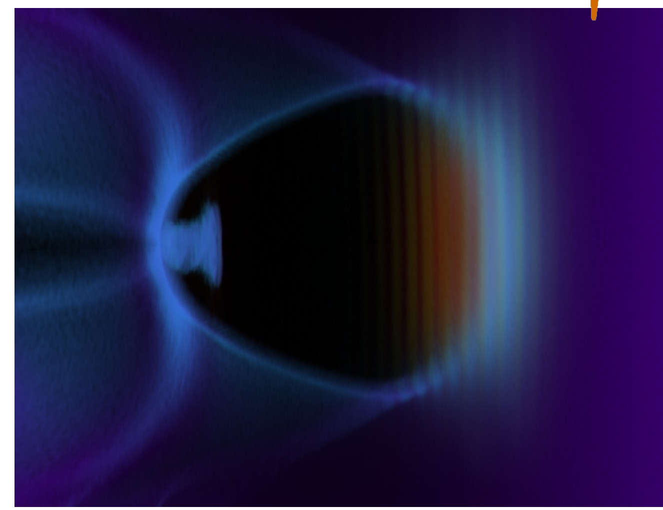


Successfully Utilized

HASEonGPU



PIConGPU



There is even more to explore

Abstract Libraries

- MallocMC
- Halt

Applications

- Imresh
- Craken (talk today)
- Isaac (talk today)

Erik Zenker · Computational Radiation Physics · www.hzdr.de/crp
e.zenker@hzdr.de

Clone us on GitHub

https://github.com/ComputationalRadiationPhysics

git clone https://github.com/ComputationalRadiationPhysics/alpaka

git clone https://github.com/ComputationalRadiationPhysics/graybat

DRESDEN concept HZDR

e.zenker@hzdr.de