HELMHOLTZ

MATTER AND TECHNOLOGIES

RESEARCH FOR GRAND CHALLENGES

LASER PARTICLE ACCELERATION NEEDS HETEROGENEOUS COMPUTING

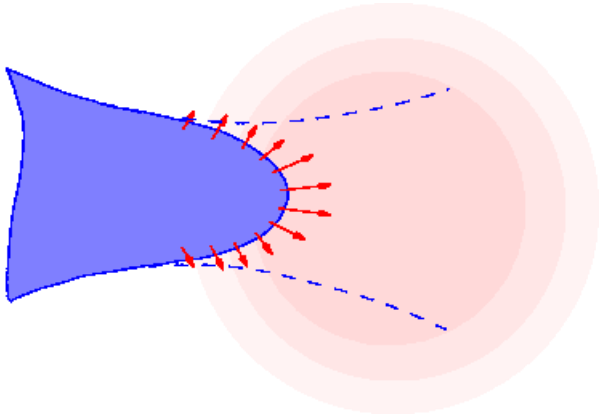
Michael Bussmann, Axel Hübl, René Widera, Alexander Matthes, Felix Meyer, Simeon Ehrig, Matthias Werner, Richard Pausch, Alexander Debus, Marco Garten, Thomas Kluge, Heiko Burau and many others!

HZDR

HELMHOLTZ ZENTRUM DRESDEN ROSSENDORF

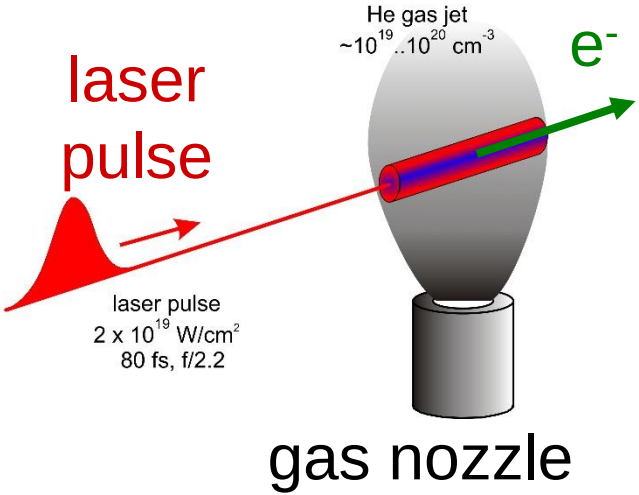
LASER PLASMA WAKEFIELD ACCELERATION (LWFA)

LASER WAKEFIELD ACCELERATION IN A NUTSHELL

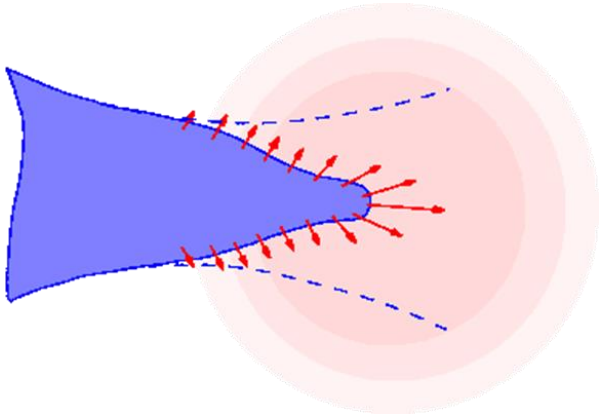


$$\omega_p^2 = \frac{e n_e}{\varepsilon_0 \gamma m_e}$$

$$n = \sqrt{1 - \frac{\omega_p^2}{\omega^2}}$$

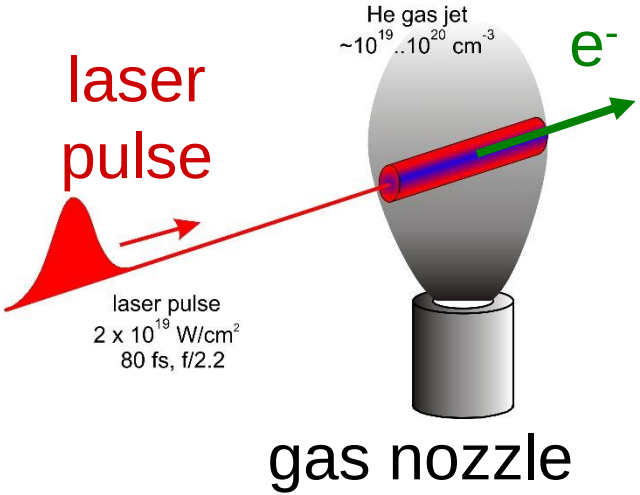


LASER WAKEFIELD ACCELERATION IN A NUTSHELL

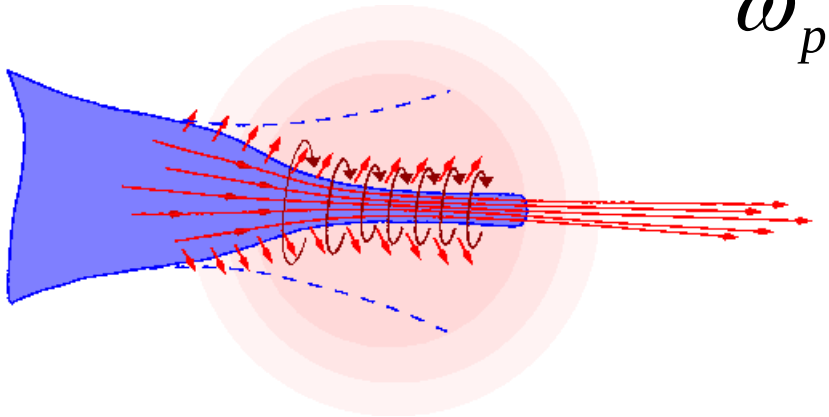


$$\omega_p^2 = \frac{e n_e}{\varepsilon_0 \gamma m_e}$$

$$n = \sqrt{1 - \frac{\omega_p^2}{\omega^2}}$$

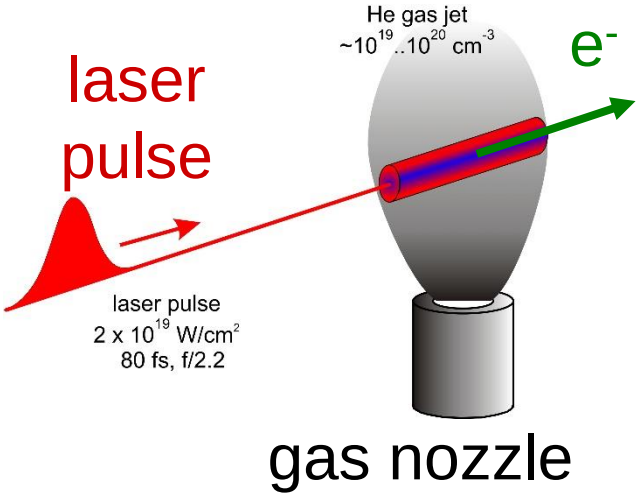


LASER WAKEFIELD ACCELERATION IN A NUTSHELL

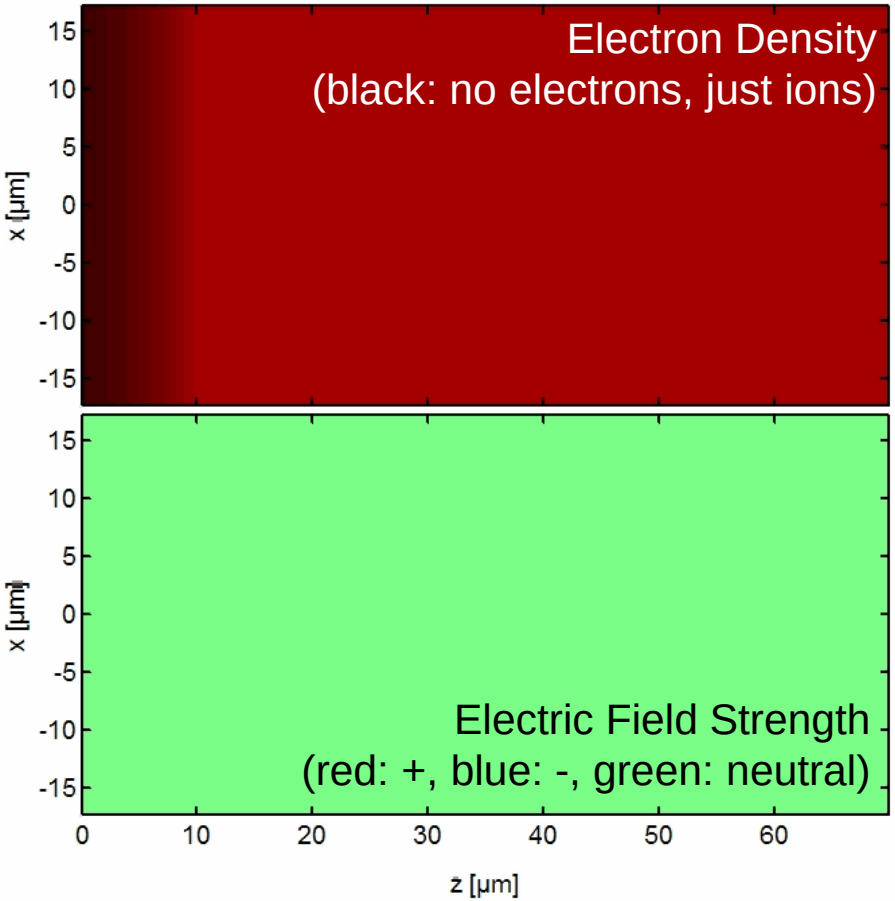


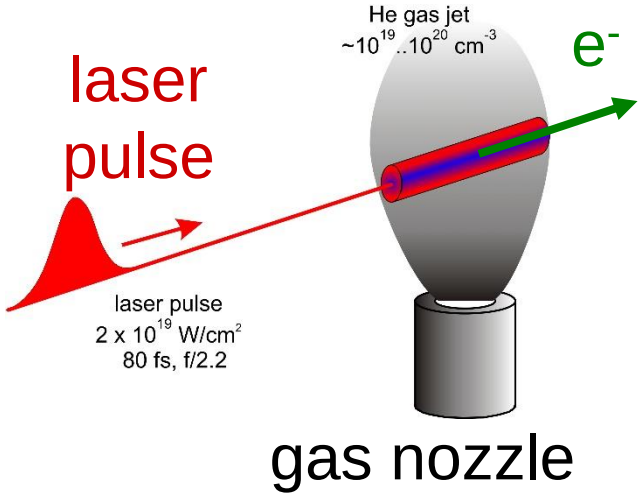
$$\omega_p^2 = \frac{e n_e}{\varepsilon_0 \gamma m_e}$$

$$n = \sqrt{1 - \frac{\omega_p^2}{\omega^2}}$$



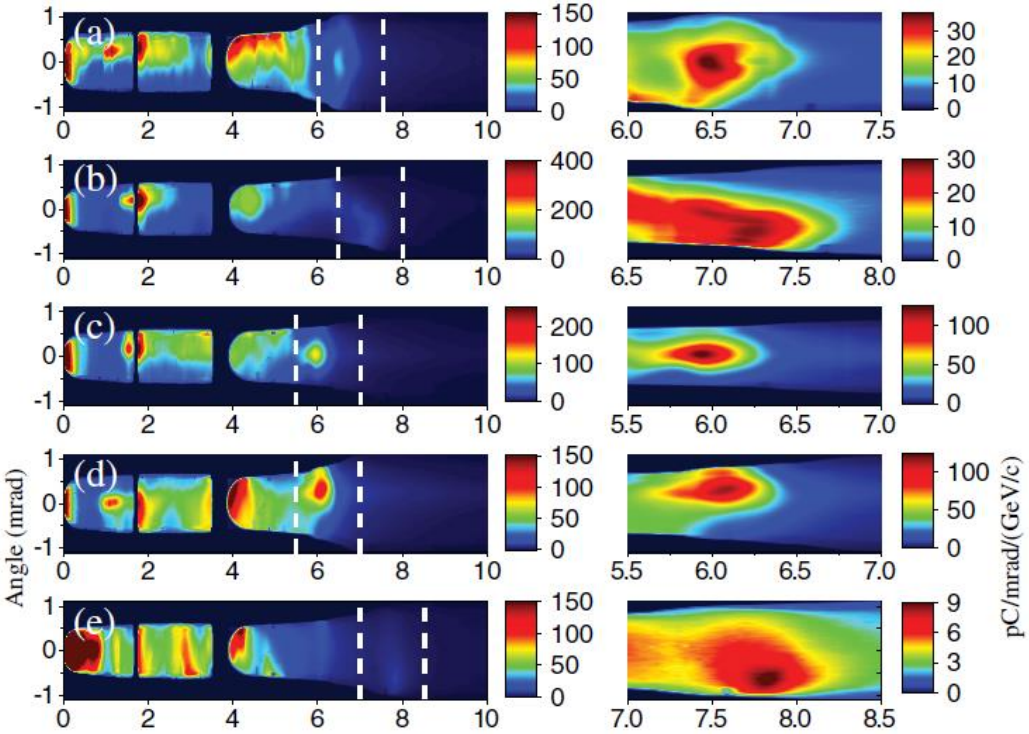
LASER WAKEFIELD ACCELERATION IN A NUTSHELL

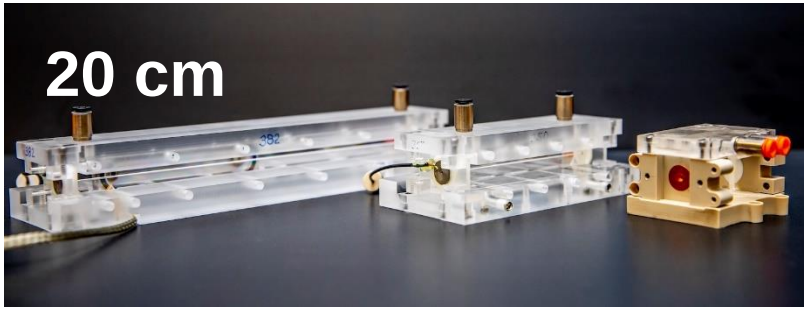


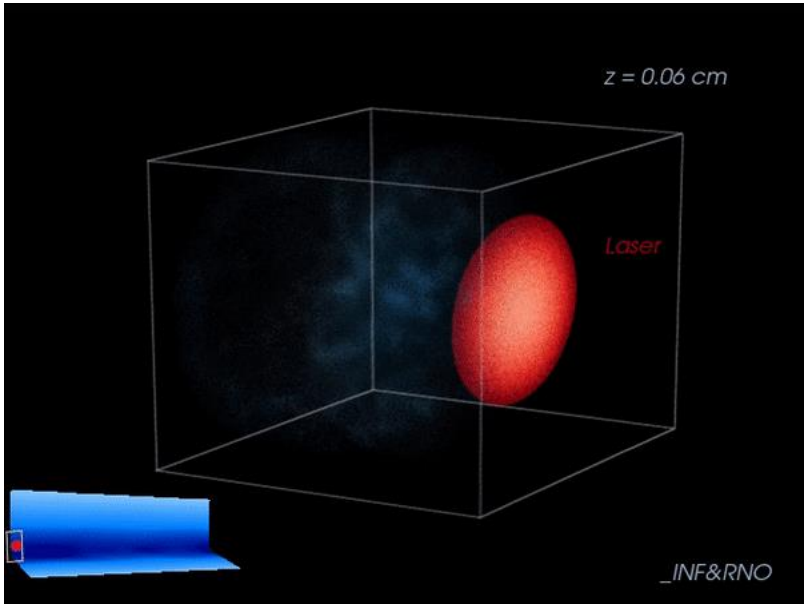


LWFA OF ELECTRONS

PRL 122, 084801 (2019)







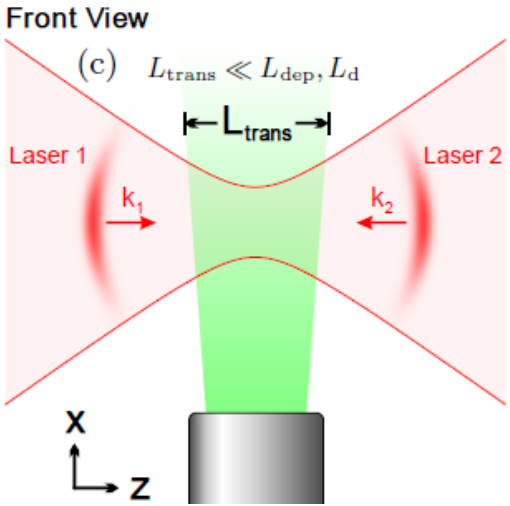
PLASMA „BUBBLE“ VS. ACCELERATOR CAVITY

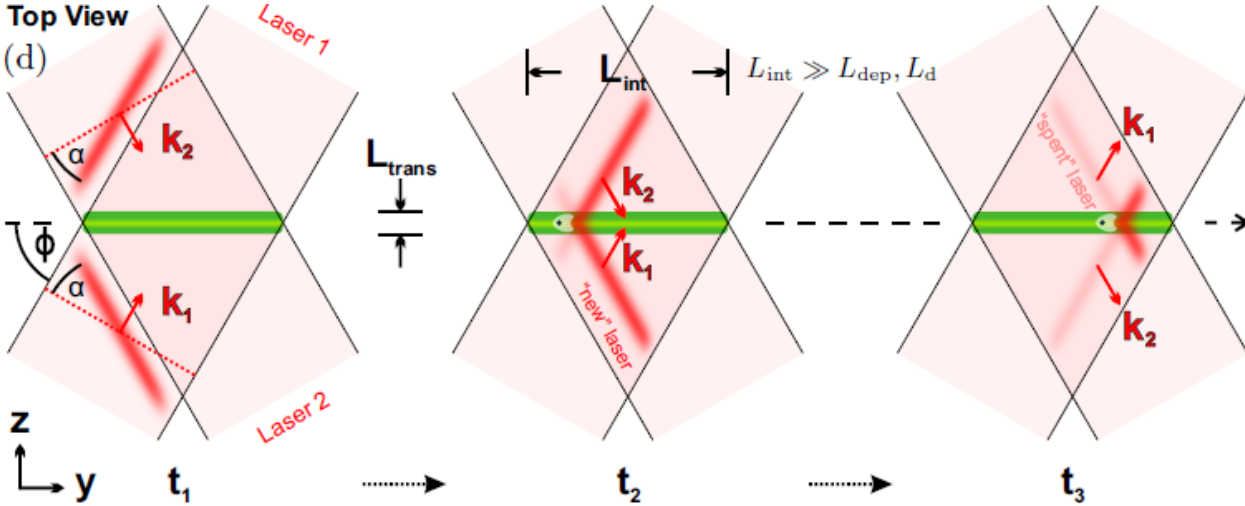
$$\frac{7.8\ GeV}{20\ cm} \approx 40\,000\,\frac{MeV}{m} \text{ vs. } 200\,\frac{MeV}{m}$$

PLASMA „BUBBLE“ VS. ACCELERATOR CAVITY

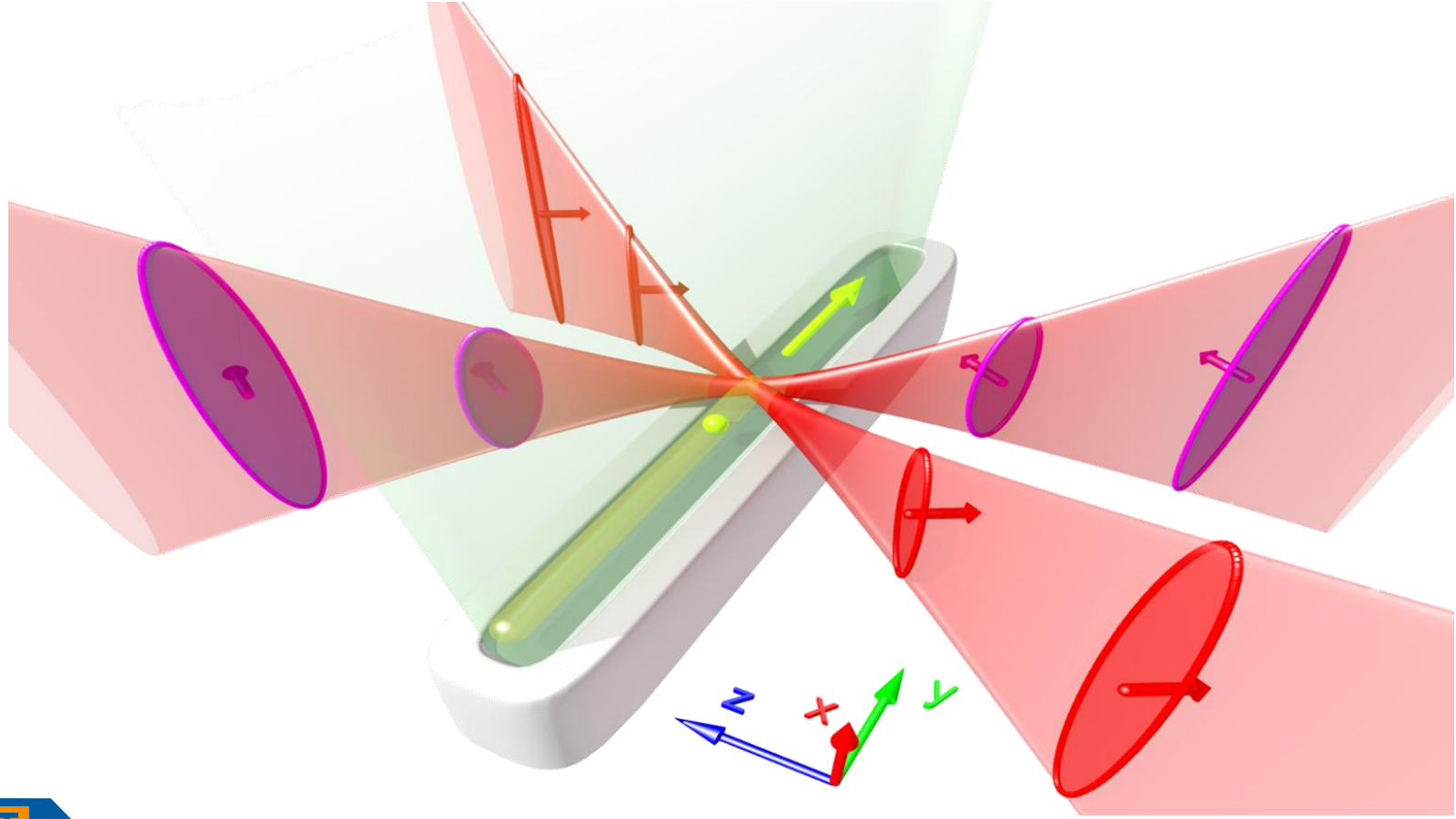
- Dephasing — Electrons outrun laser (plasma, not vacuum!)
- Guiding — Laser needs to be focused by plasma (capillary!)
- Staging — Hard to couple-in second laser along beam path
- Modulation — Laser modulates Plasma modulates Laser
- Self Injection — Electrons can be self-injected into the plasma

TRAVELLING-WAVE ELECTRON ACCELERATION (TWEAC)

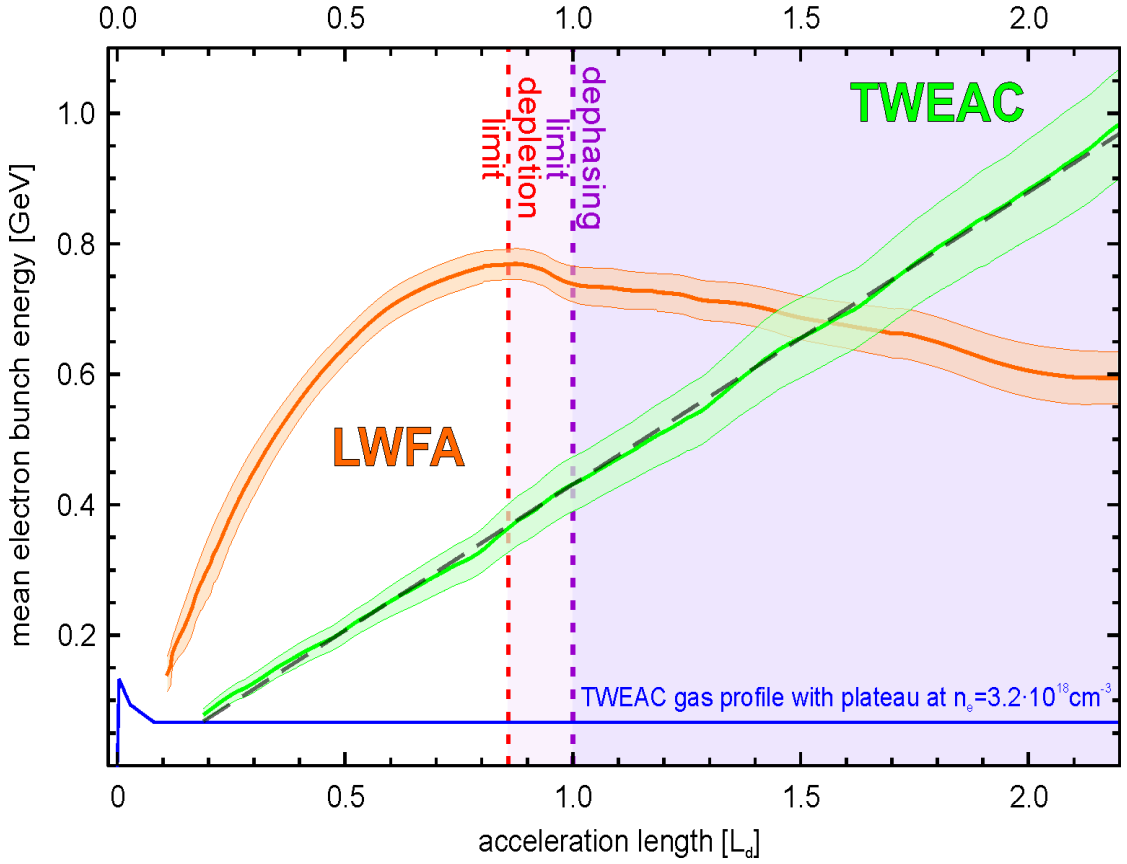




TWEAC DOES NOT NEED GUIDING, LASER NOT MODULATED

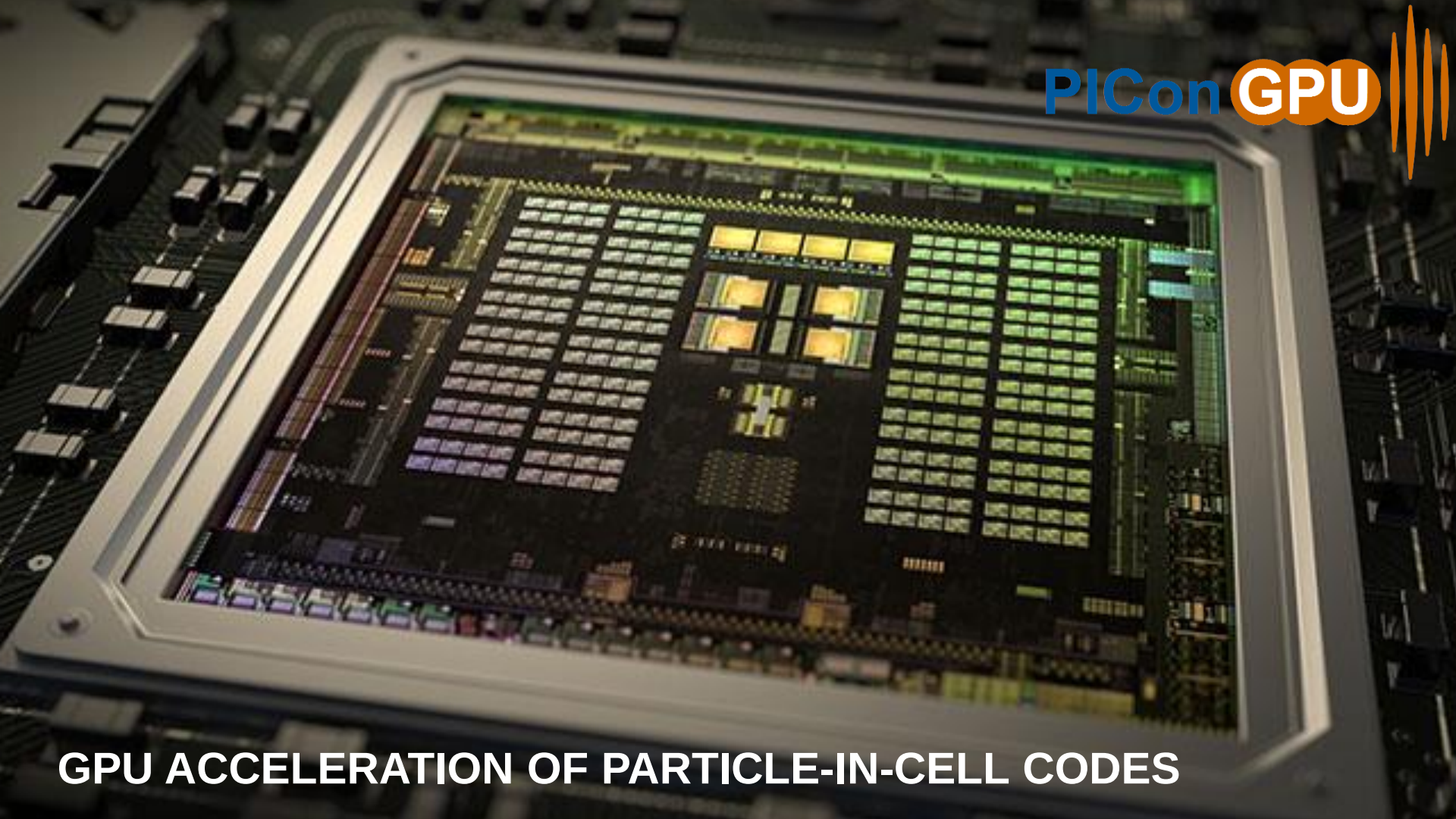


TWEAC SHOWS NO DEPHASING (OR DEPLETION)



TWEAC VS. LWFA

- ~~Dephasing — Electrons outrun laser (plasma, not vacuum!)~~
- ~~Guiding — Laser needs to be focused by plasma (capillary!)~~
- ~~Staging — Hard to couple-in second laser along beam path~~
- ~~Modulation — Laser modulates Plasma modulates Laser~~
- ~~Self Injection — Electrons can be self-injected into the plasma~~



GPU ACCELERATION OF PARTICLE-IN-CELL CODES

THE PARTICLE-IN-CELL METHODS FOR PLASMA SIMULATIONS

Compute the Lorentz Force

$$\vec{F} = q\left(\vec{E} + \vec{v} \times \vec{B}\right)$$

Integrate the Equation of Motion

$$\frac{d\vec{p}}{dt} = \vec{F}$$

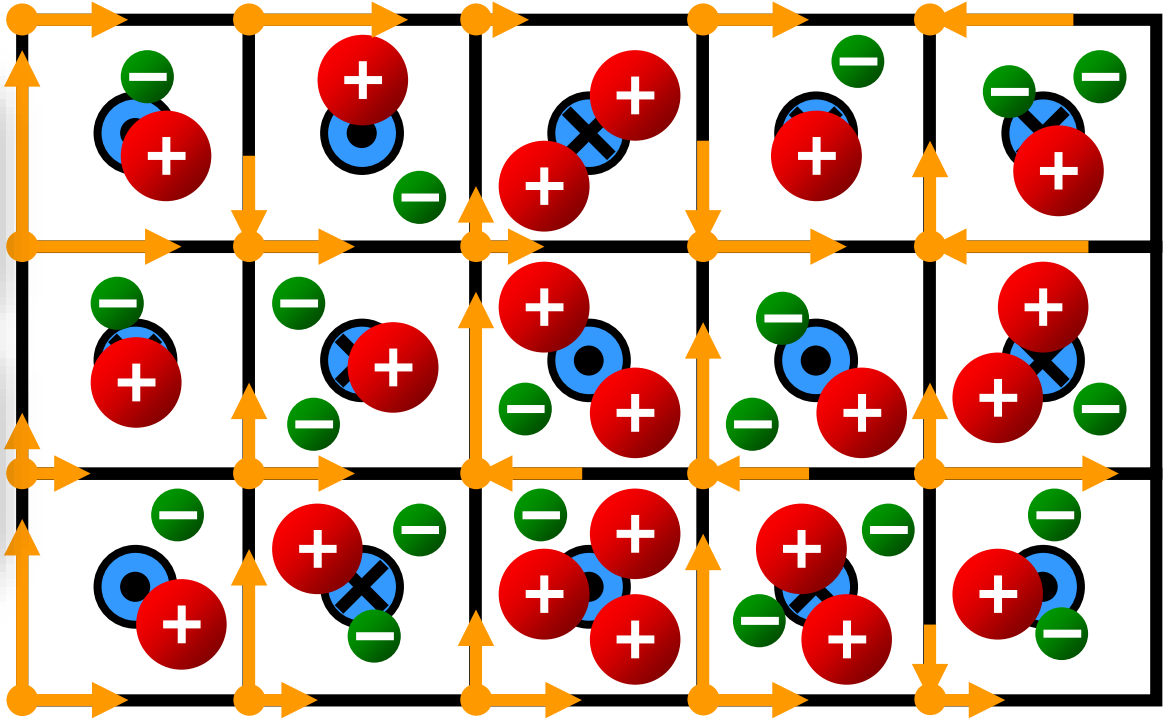
Compute Currents

$$\vec{J} = \int \vec{u} f(t, \vec{r}, \vec{v}) d\vec{v}$$

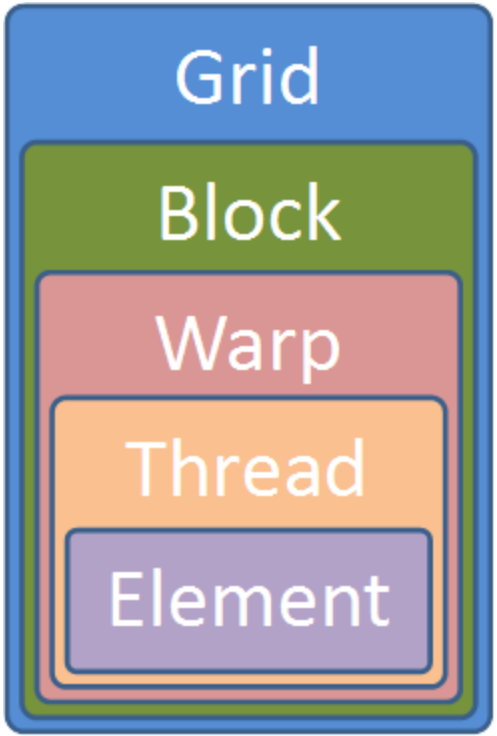
Compute new Fields

$$\frac{\partial \vec{E}}{\partial t} = \frac{1}{c^2} \nabla \times \vec{B} - \frac{\vec{J}}{\varepsilon_0}$$

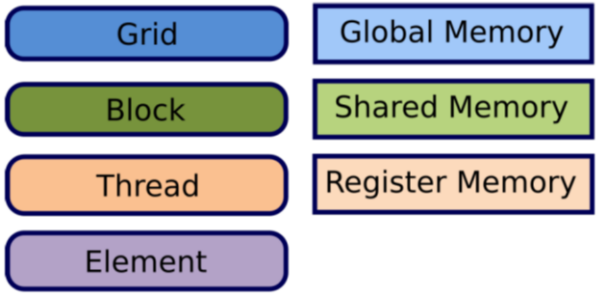
$$\frac{\partial \vec{B}}{\partial t} = -\nabla \times \vec{E}$$



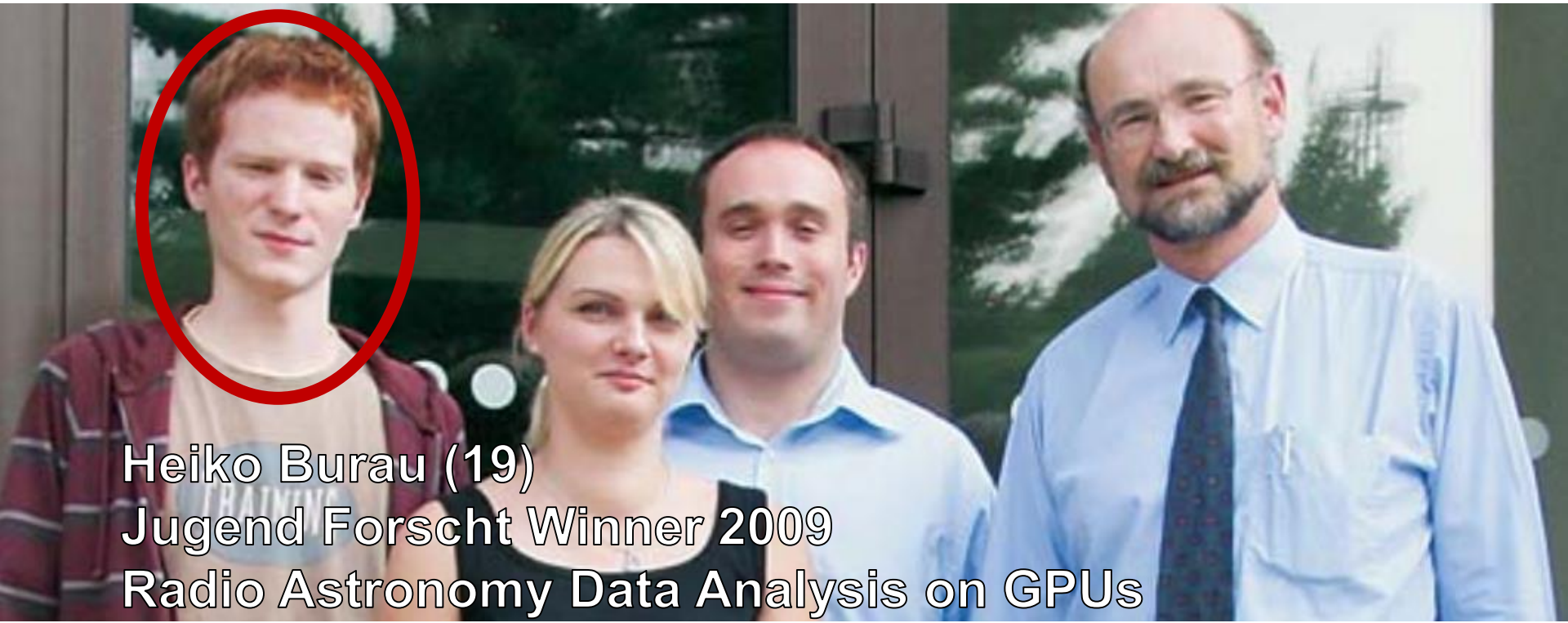
DATA PARALLELISM IN MODERN HARDWARE



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

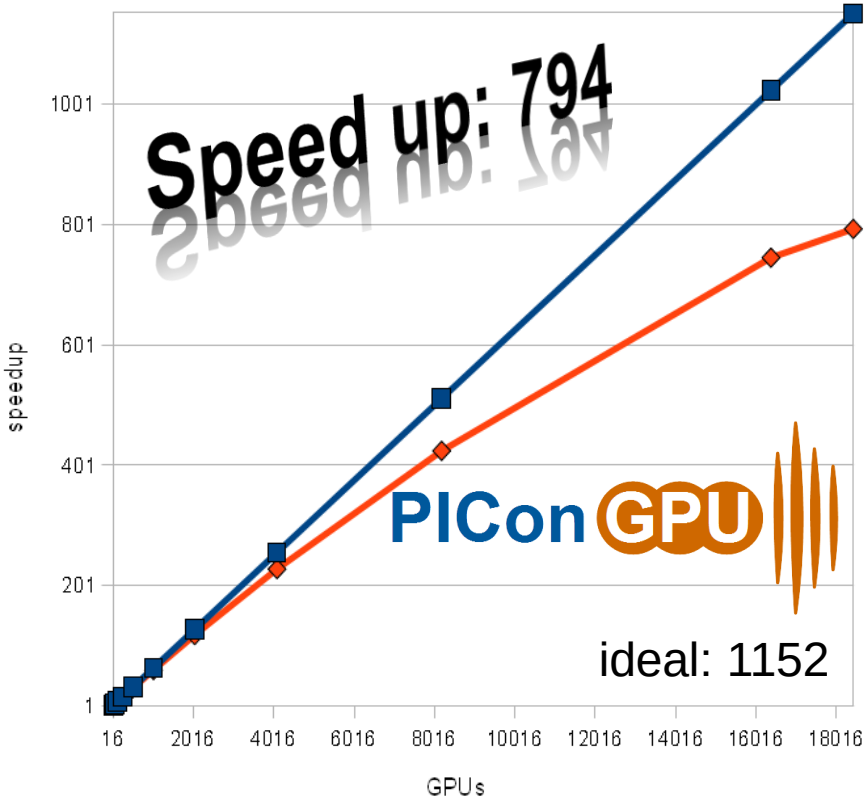


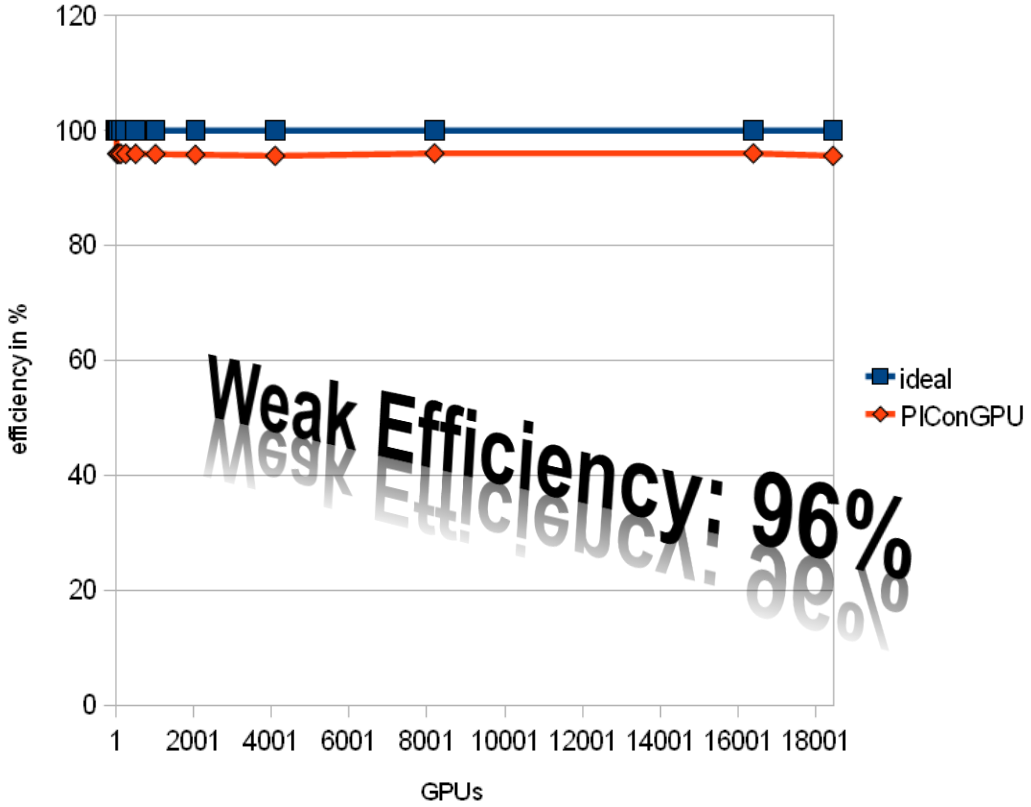
ALL-STUDENT DEVELOPMENT STARTED IN 2009



STILL ALL-STUDENT DEVELOPMENT IN 2013

RESULTS FROM 2013 (TITAN @ ORNL, 17.59 PFLOPS/S)





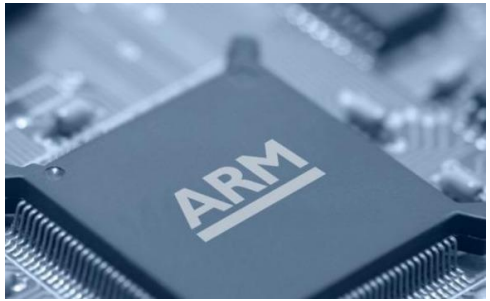
7.176 PFLOP/s (double precision)

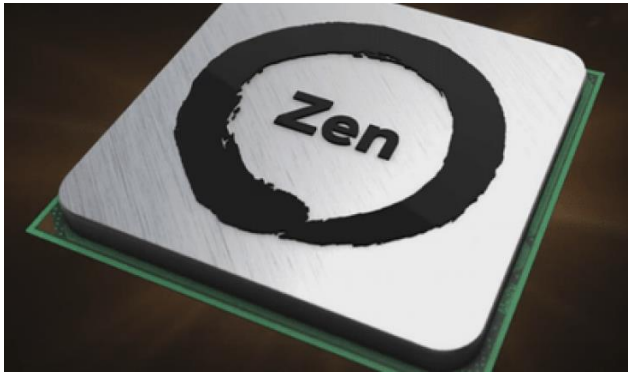
RUN ON ALL THE MACHINES!

Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
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	2,397,824	143,500.0	200,794.9	9,783
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	125,712.0	7,438
3	Sunway TaihuLight - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway , NRCPC National Supercomputing Center in Wuxi China	10,649,600	93,014.6	125,435.9	15,371
4	Tianhe-2A - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000 , NUDT National Super Computer Center in Guangzhou China	4,981,760	61,444.5	100,678.7	18,482
5	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	27,154.3	2,384
6	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	41,461.2	7,578
7	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	32,576.6	1,649
8	SuperMUC-NG - ThinkSystem SD530, Xeon Platinum 8174 24C 3.1GHz, Intel Omni-Path , Lenovo Leibniz Rechenzentrum Germany	305,856	19,476.6	26,873.9	
9	Titan - Cray XK7, Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x , Cray Inc. DOE/SC/Oak Ridge National Laboratory United States	560,640	17,590.0	27,112.5	8,209
10	Sequoia - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom , IBM DOE/NNSA/LLNL United States	1,572,864	17,173.2	20,132.7	7,890

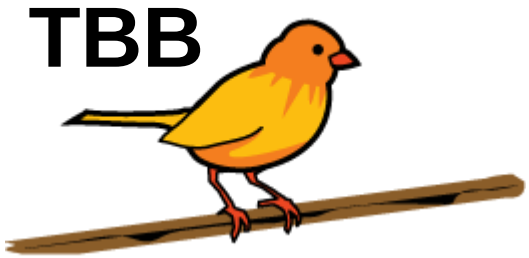
```
#ifdef CUDA_ENABLE
  // CUDA implementation

#elif OpenMP_ENABLE
  // OpenMP implementation

#elif OpenACC_ENABLE
  // OpenACC implementation

#elif …

#else
 // Serial implementation

#endif
```

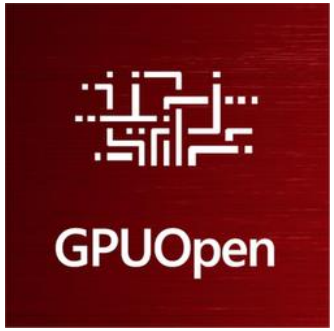
A ZOO OF HARDWARE

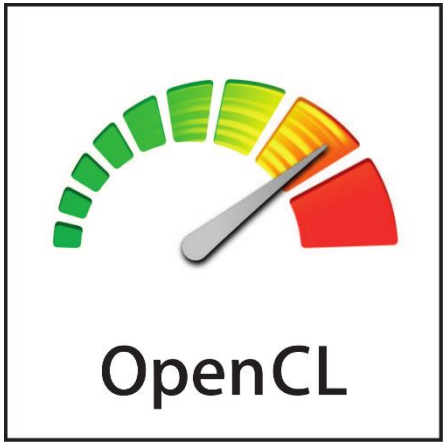




A ZOO OF PROGRAMMING MODELS



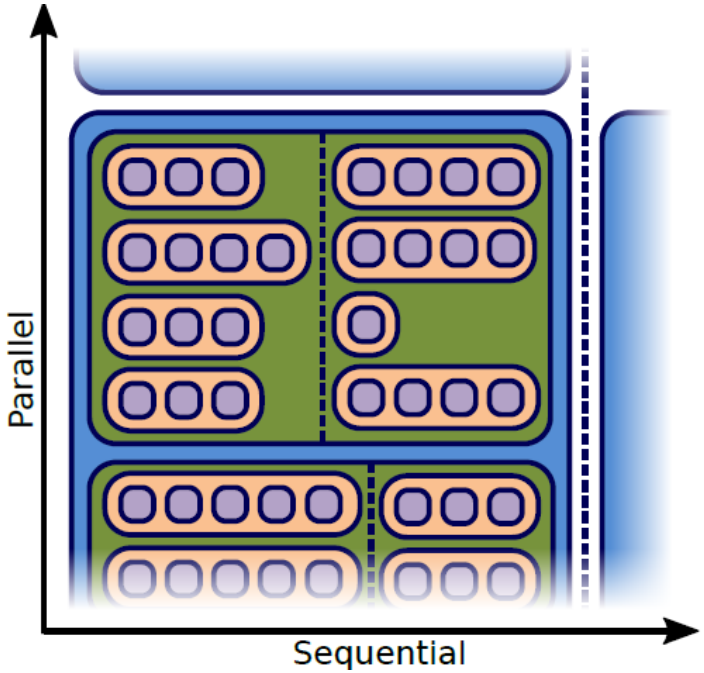




ALPAKA

- Provide **performance portability** across accelerators through abstraction of the underlying levels of parallelism.
- **Not hiding** the abstraction, but making it explicit
- **C++11 header-only** library
- Write algorithms **once**
- Platform decision at compile-time
- Aim for **zero overhead** abstraction
- Data are plain pointers (parallel data structures are hard)

DATA PARALLELISM IN MODERN HARDWARE



Grid

Block

Thread

Element

---- Synchronize

Hierarchy Level	Parallelism	Synchronizable
grid	sequential / parallel	✗ / ✓
block	parallel	✗
warp	parallel	✓
thread	parallel / lock-step	✓
element	sequential	✗

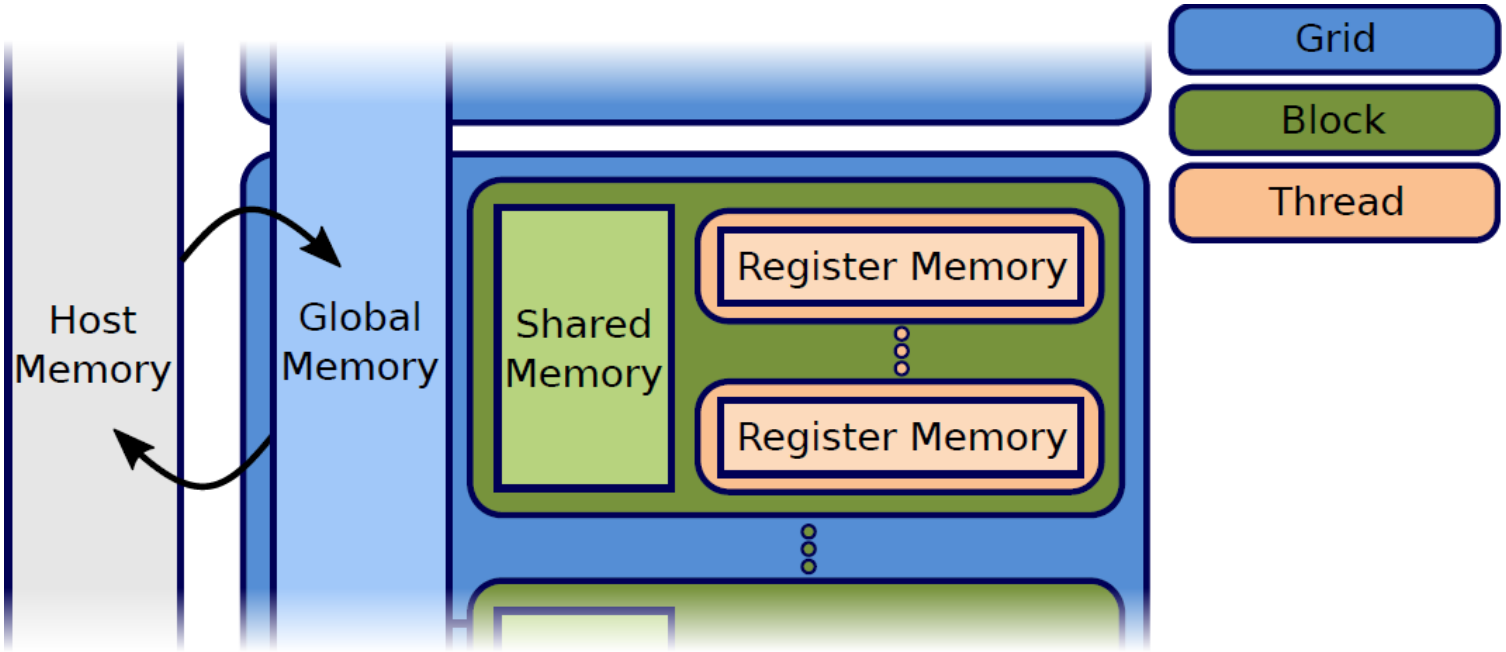
FUTURE:

VC GSI

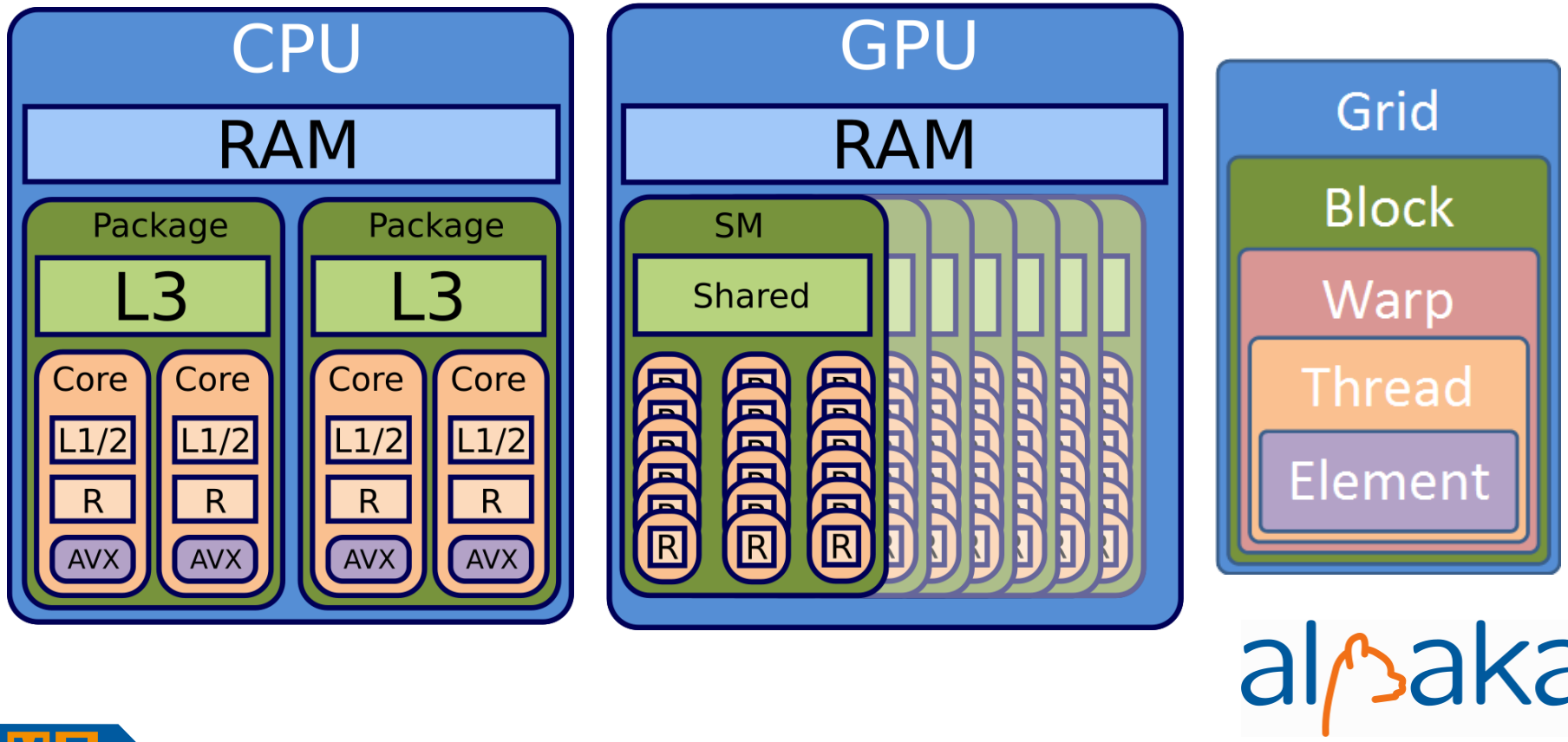
C++ SIMD
Vectorization &
Packing

alpaka

NEED TO COVER MEMORY LEVELS AS WELL



ALPAKA — ONE C++ INTERFACE TO RULE THEM ALL

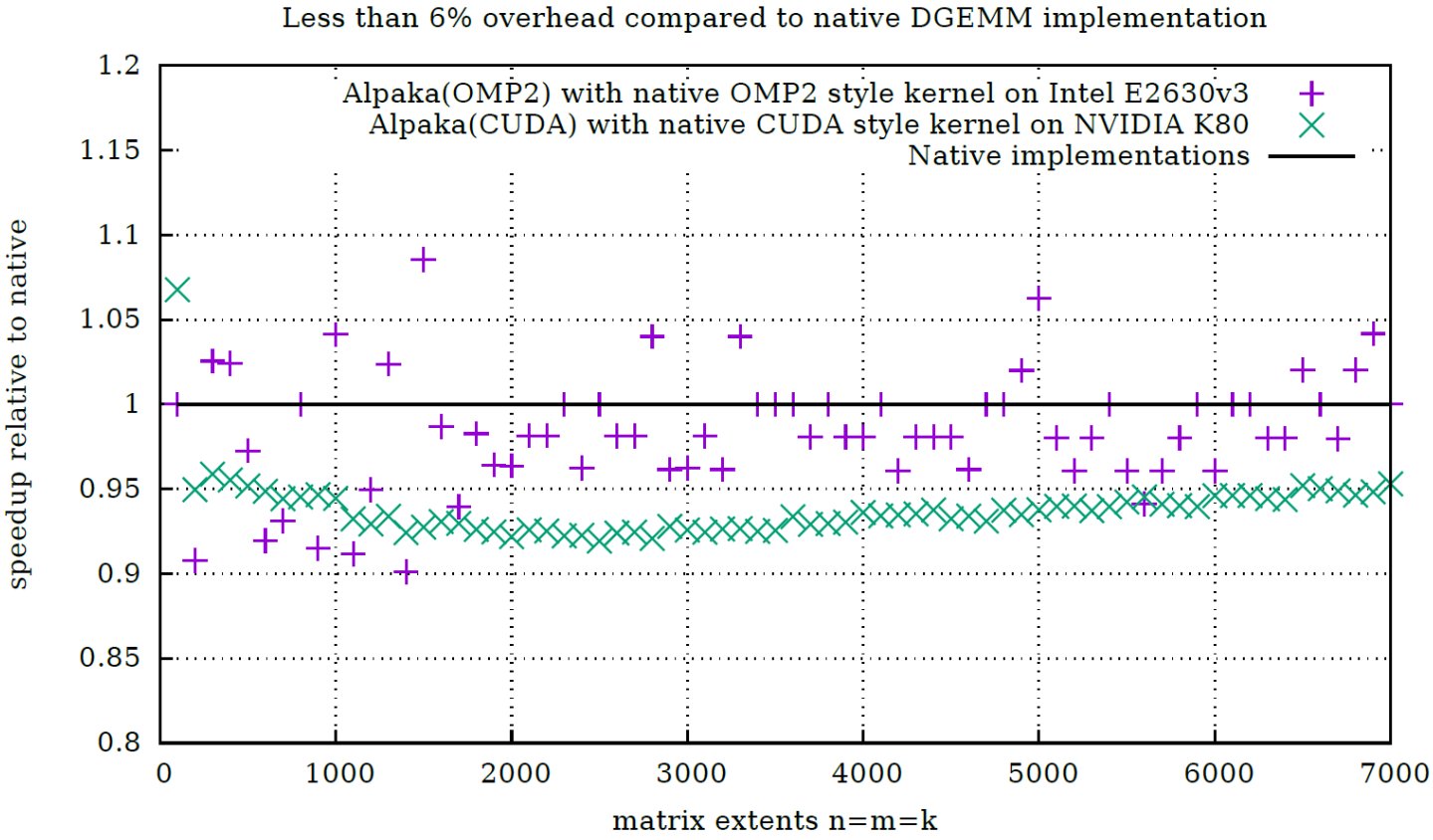


ALPAKA – DOUBLE PRECISION Y = A * X + Y

```
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] = alpha * X[i] + Y[i];
    }
};
```

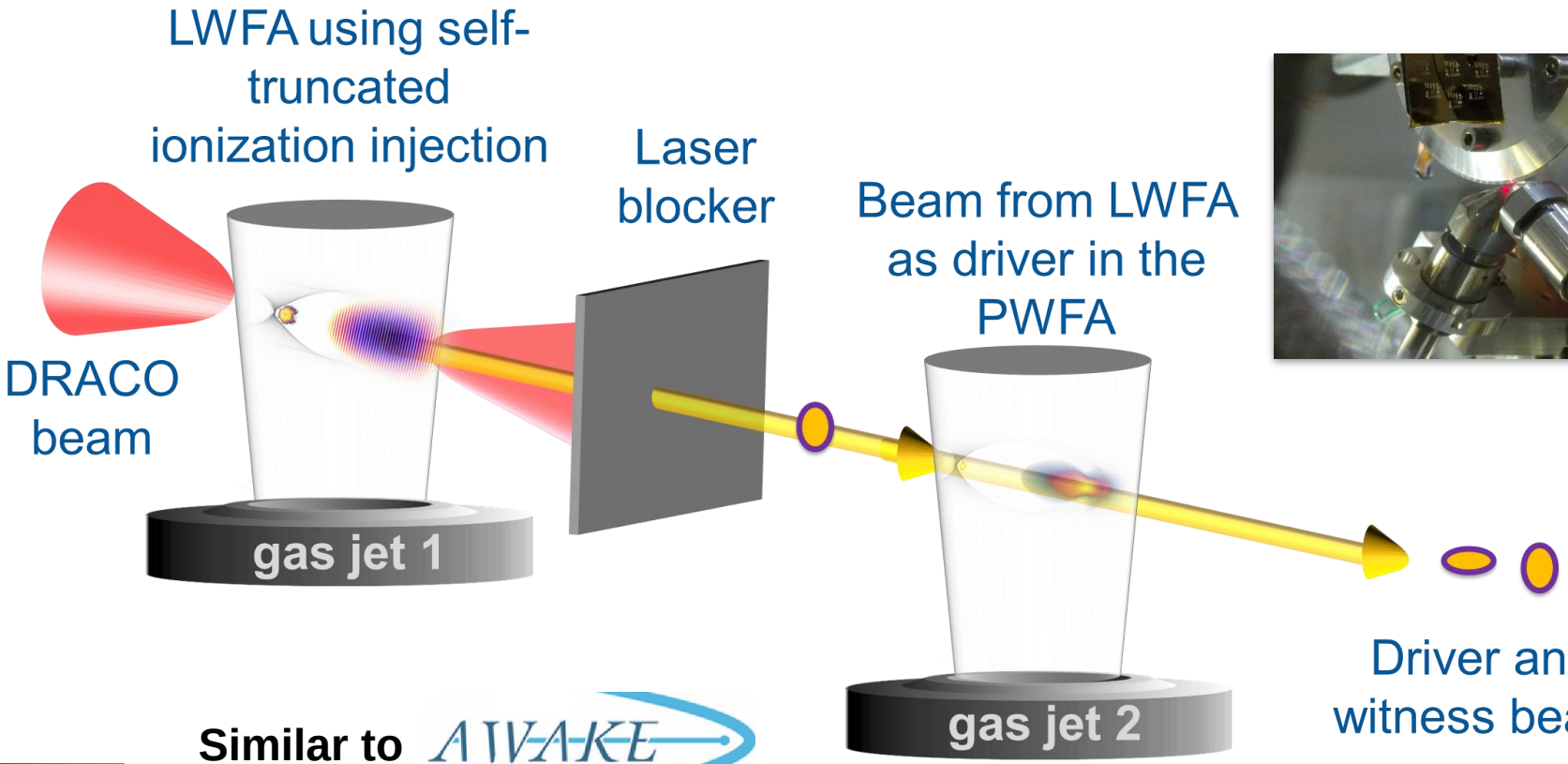
Use CMAKE to change Accelerators or common header

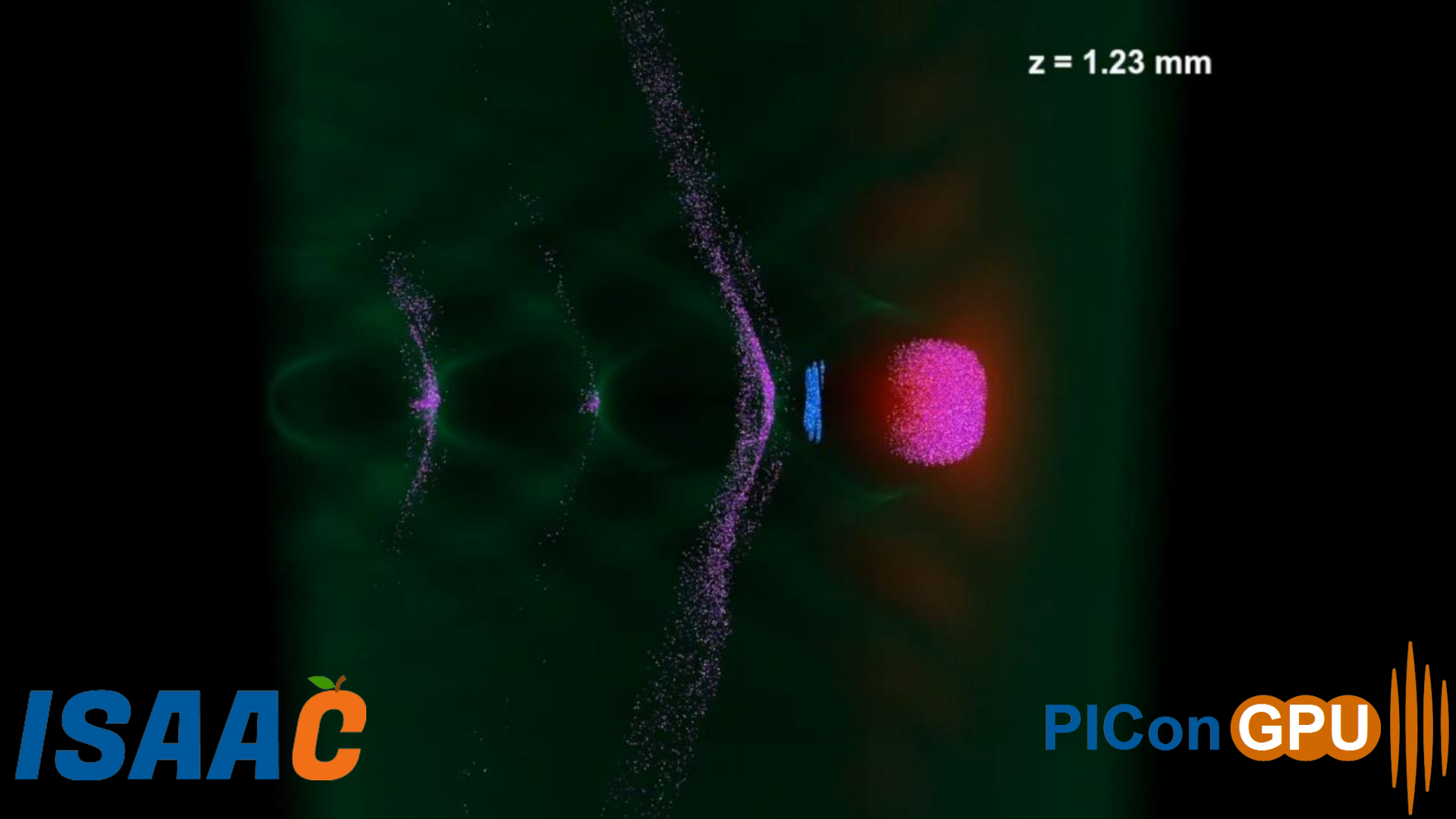
ALPAKA – NEGLIGIBLE OVERHEAD (DGEMM EXAMPLE)



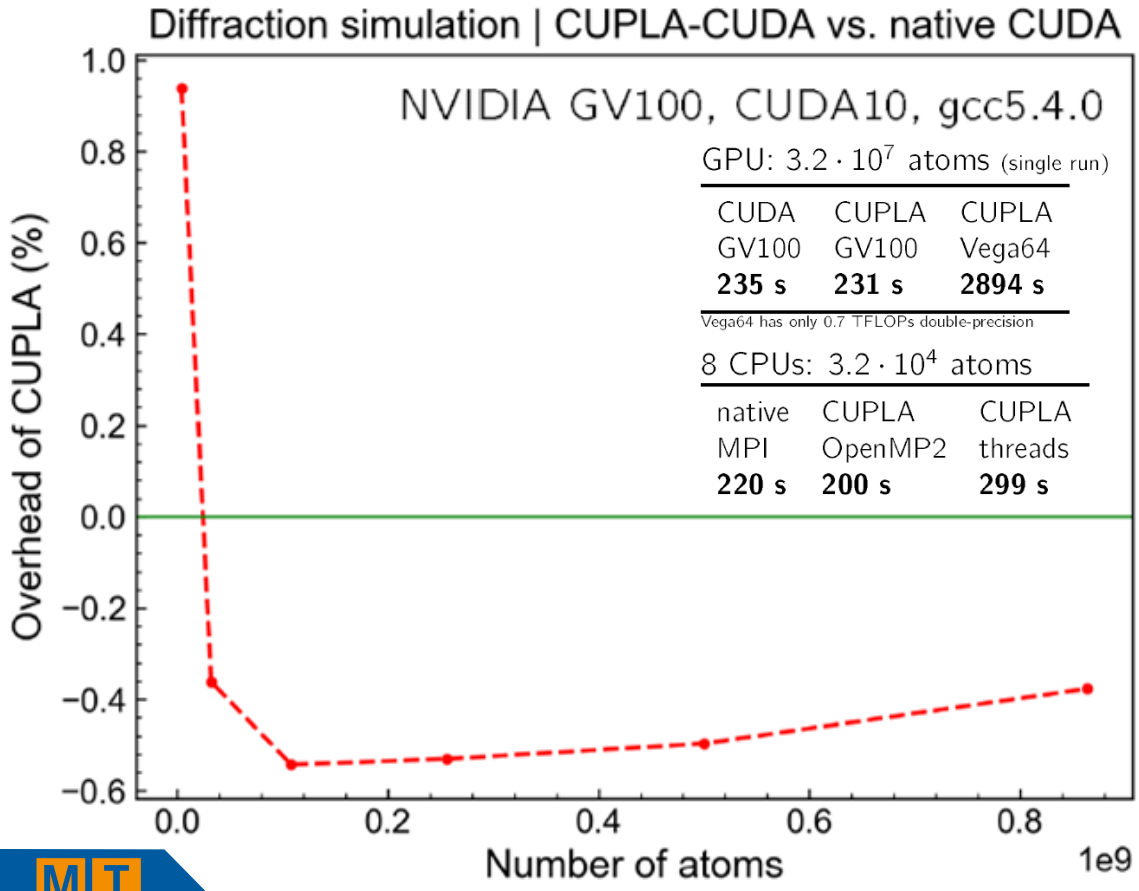
alpaka

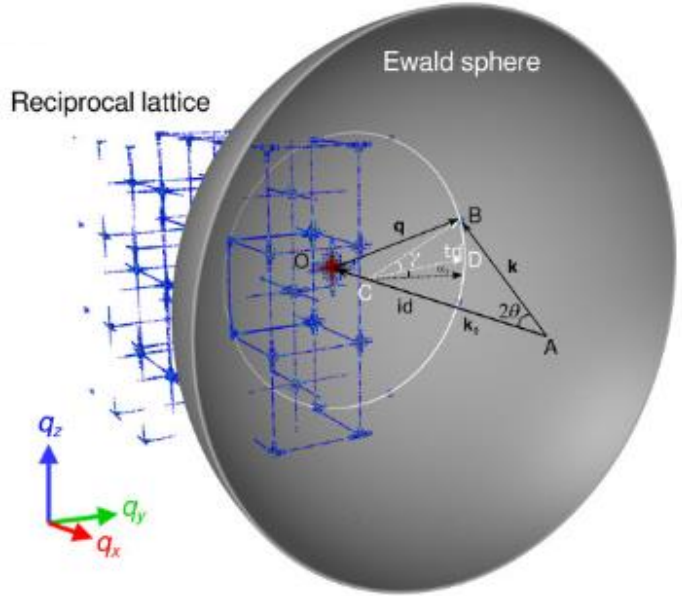
FROM LASERS TO ELECTRON BUNCHES AS DRIVERS (PWFA)



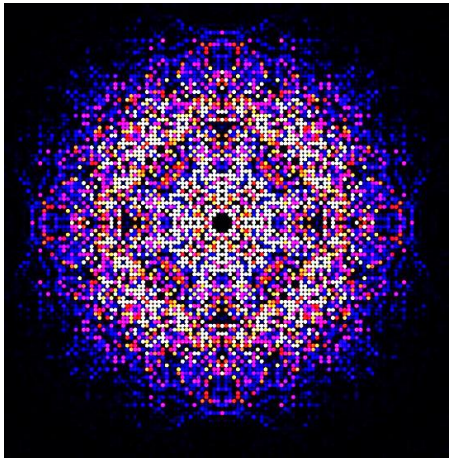

z = 1.23 mm
ISAAC
PIConGPU

ALPAKA – GAPD DIFFRACTION SIMULATION

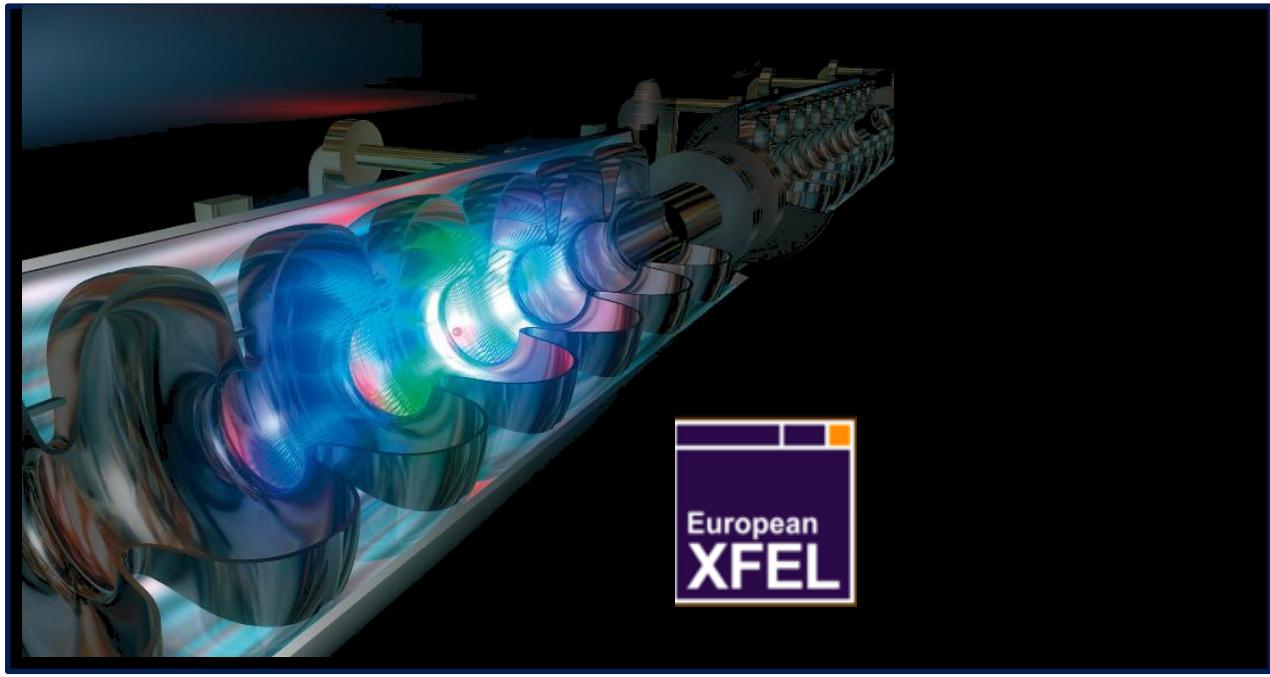




ALPAKA – NEXT UP: PHOTON SCIENCE



Trypanosoma brucei cathepsin B scattering pattern



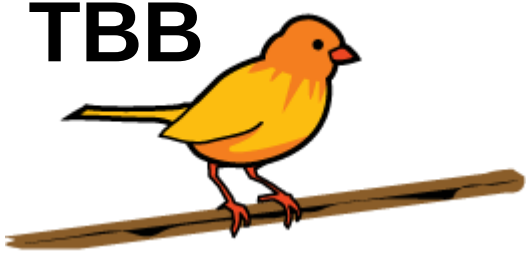
alpaka

ALPAKA BACKENDS & ARCHITECTURES

NVIDIA CUDA®

OpenMP®

TBB



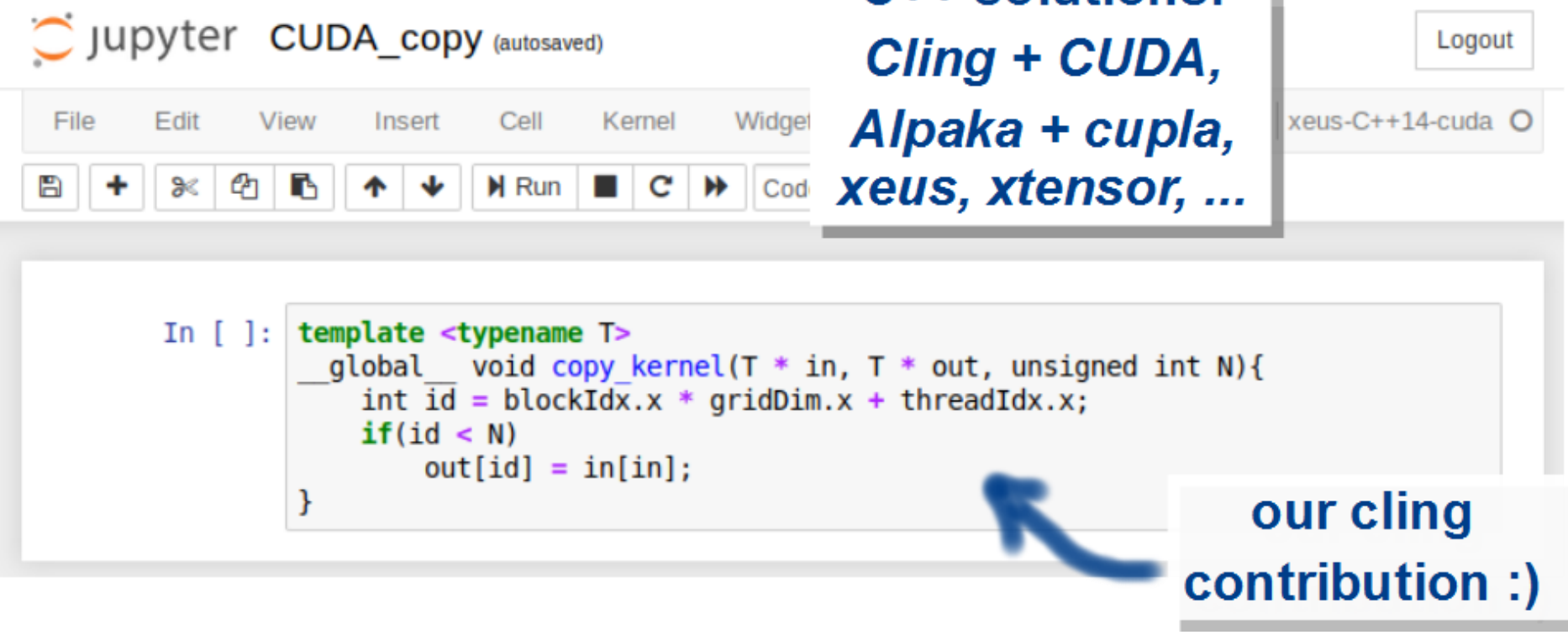
GPUOpen

SYCL™

- ✓ INTEL
- ✓ OpenPOWER (IBM)
- ✓ AMD (CPUs + GPUs)
- ✓ ARM
- ✓ nVIDIA GPUs
- ? FPGAs

alpaka

C++ CUDA JIT COMPILATION IN JUPYTER NOTEBOOKS



THANK YOU FOR INVITING US!

https://github.com/ComputationalRadiationPhysics