11th International Computational Accelerator Physics Conference ICAP 2012 in Rostock

Simulating radiation from Laser-wakefield accelerators

Alexander Debus, Richard Pausch, René Widera,
Michael Bussmann, Ulrich Schramm

Acknowledgements

PIConGPU

Michael Bussmann[1], Florian Berninger[1], Heiko Burau[1], Thomas E. Cowan[1], Alexander Debus[1], Axel Hübl[1], Thomas Kluge[1] , Richard Pausch[1], Ulrich Schramm[1], René Widera[1]

Wolfgang Hönig[2], Guido Juckeland[2], Wolfgang Nagel[2], Felix Schmitt[2]

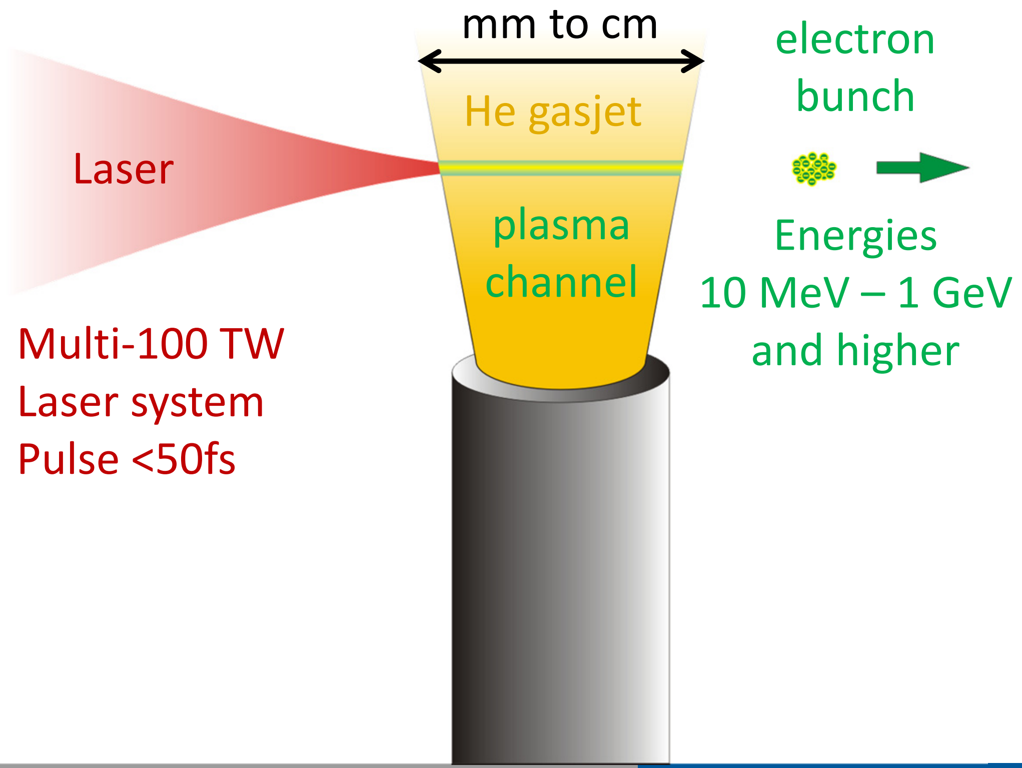
Laser wakefield experiments

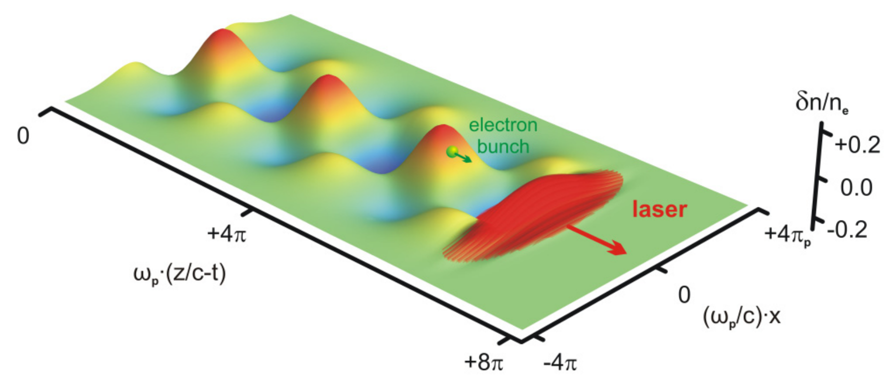
Arie Irman[1], Axel Jochmann[1], Jurjen Couperus[1]

[1] Helmholtz-Zentrum Dresden-Rossendorf
[2] Zentrum für Informationsdienste und Hochleistungsrechnen

Laser-wakefield accelerators can accelerate electrons to the GeV level and are promising for compact x-ray sources of high brilliance.

- High-power lasers generate plasma waves in underndense plasmas.
- Into these plasma waves electrons can be (self-) injected and over mm- distances accelerated to MeV to GeV energies.

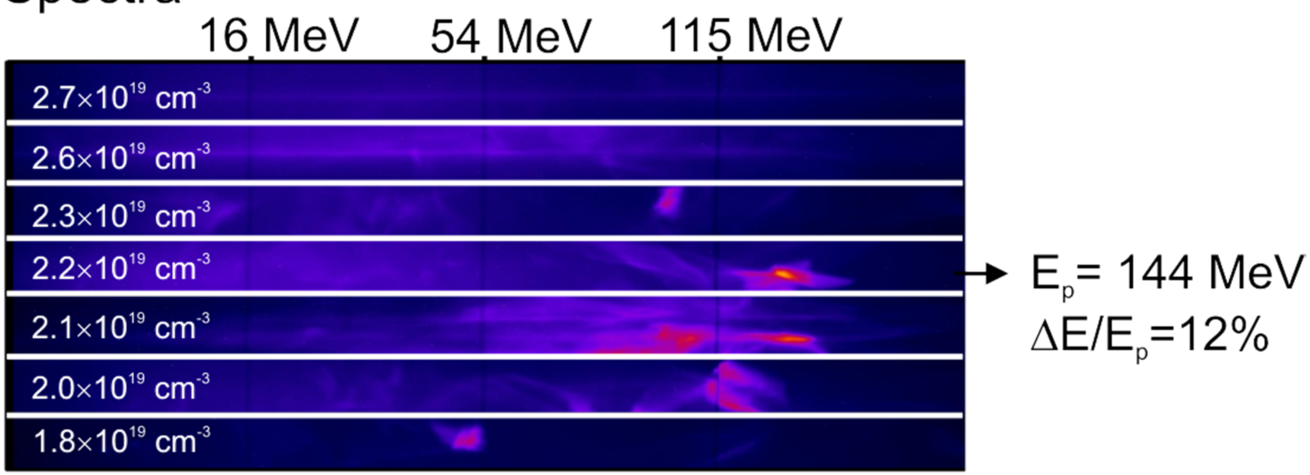




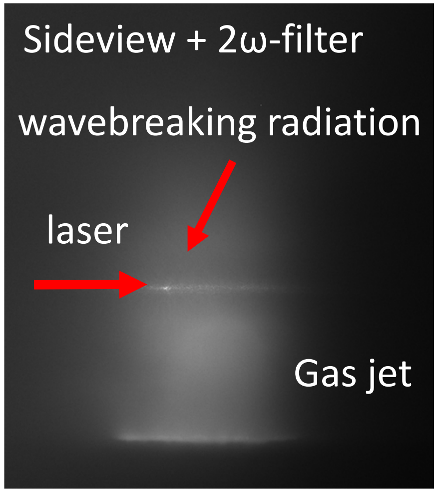
Plasma radiation has the potential of being a quantitative comparison between theory and experiment.

- Highly nonlinear processes make laser-accelerated electrons challenging to control.
- Especially diagnostics and control of the electron (self-)injection process is necessary.
- Electron spectra are measurements after the fact.
  (Currently best source for Theory vs. Experiment comparisons.)
- Radiation spectra tell us more on the processes during laser-plasma interaction.

Spectra



plasma self-emission



2.3 J, 25 fs, I_{peak} :2.4 x 10^{19} W cm^{-2} (a_0=3.4)

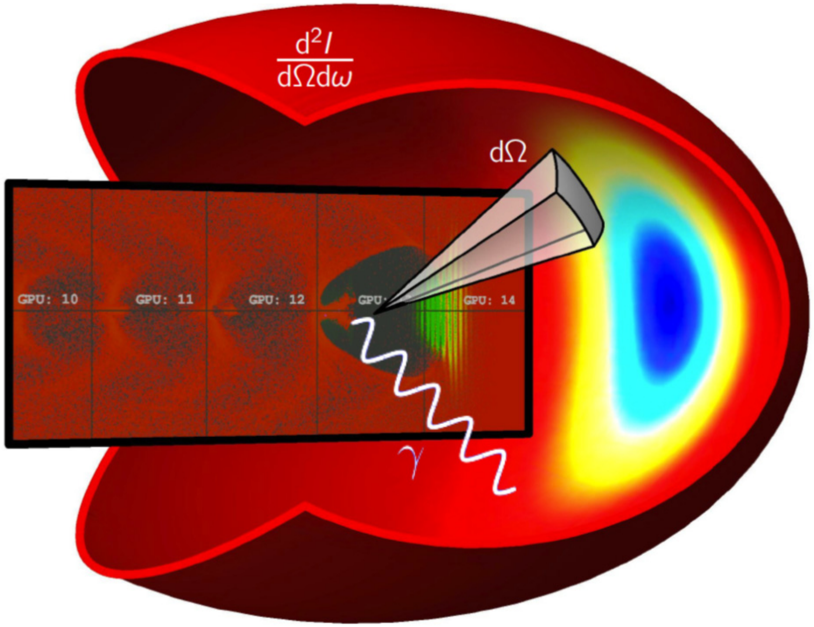
Detecting radiation is straightforward in experiment, but poses a challenge for computation.

- Physics: Calculate radiation and characterize signatures from electron injection and betatron oscillations.

- Goal: Calculate radiation from ALL electrons with full angular characteristics.

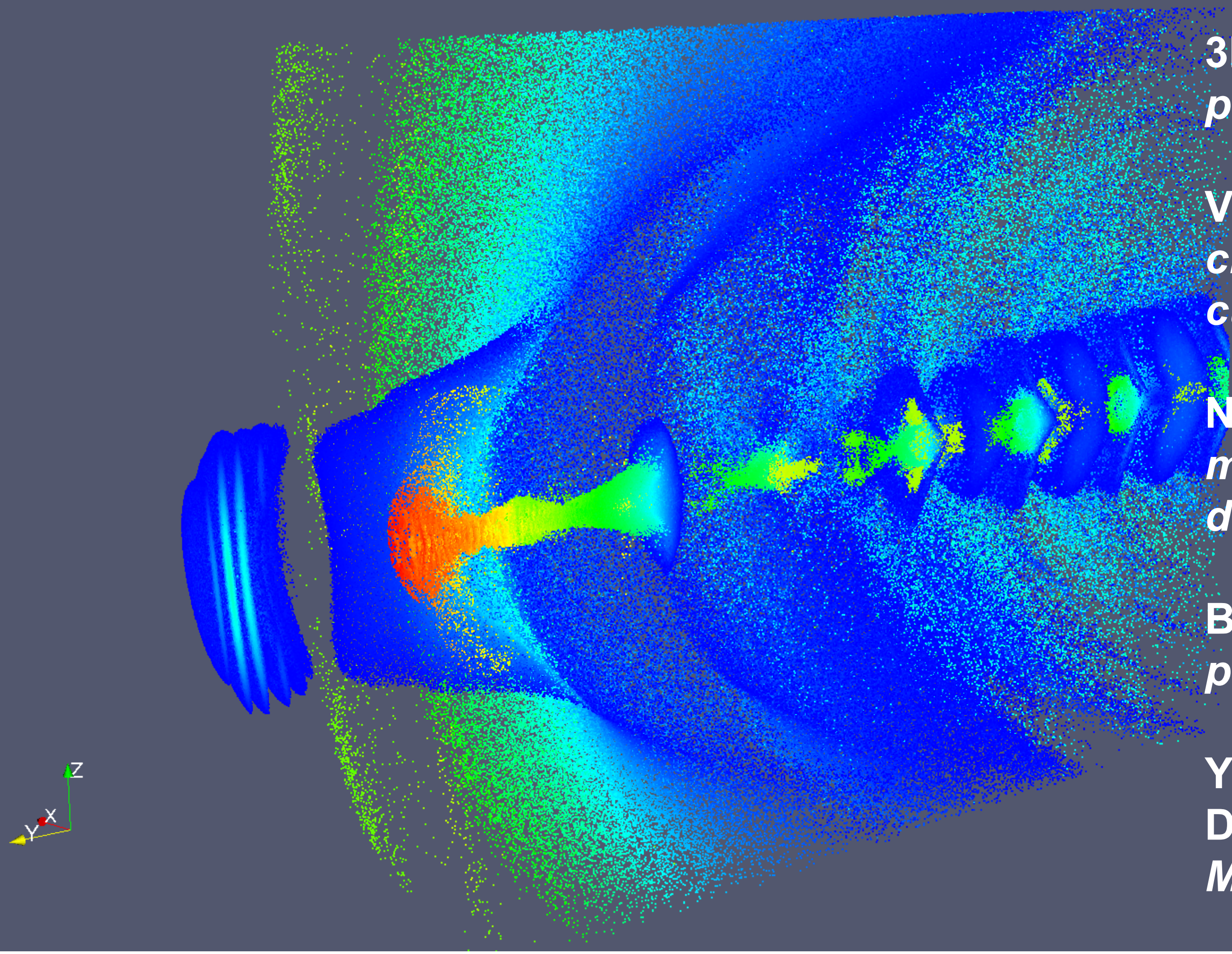
  Radiation codes as post-processing after PIC quickly become impracticable.

  CPU-PIC: 200 TB raw particle data within 3 weeks.



- EM fields from the PIC grid do not have enough spatial resolution for X-Ray radiation arising from large Doppler-shifts.
- Approach here: Liénard-Wiechert potentials → Calculate from trajectories.

PIConGPU - a GPU driven 3D-PIC code provides the basis.



3D3V relativistic
particle-in-cell code

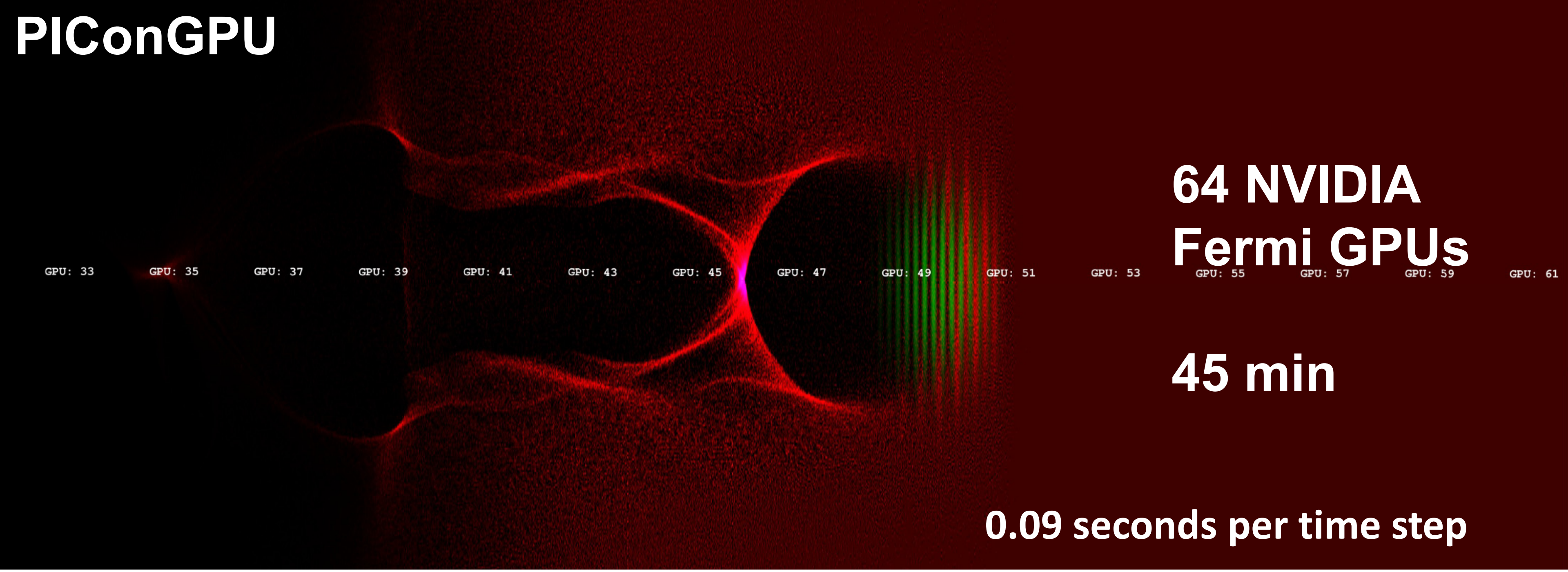
Villasenor-Buneman
charge conserving
current deposition

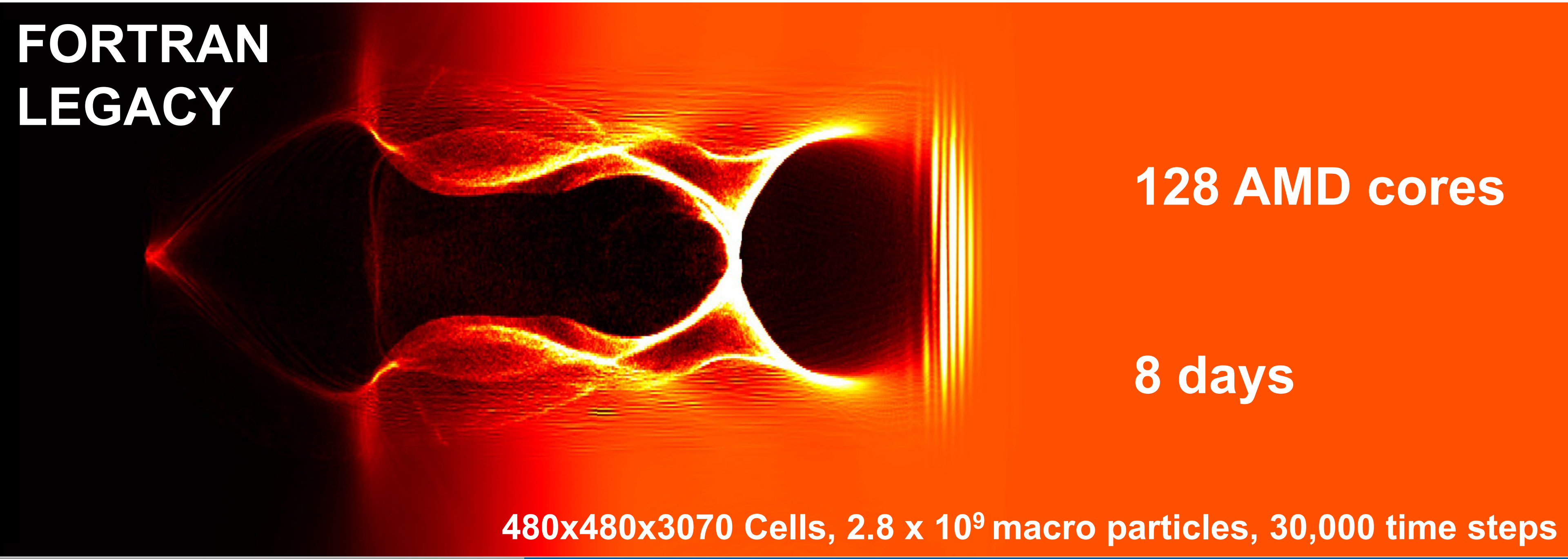
NGP / CIC / TSC
macro-particle
distribution functions

Boris Push
particle pusher

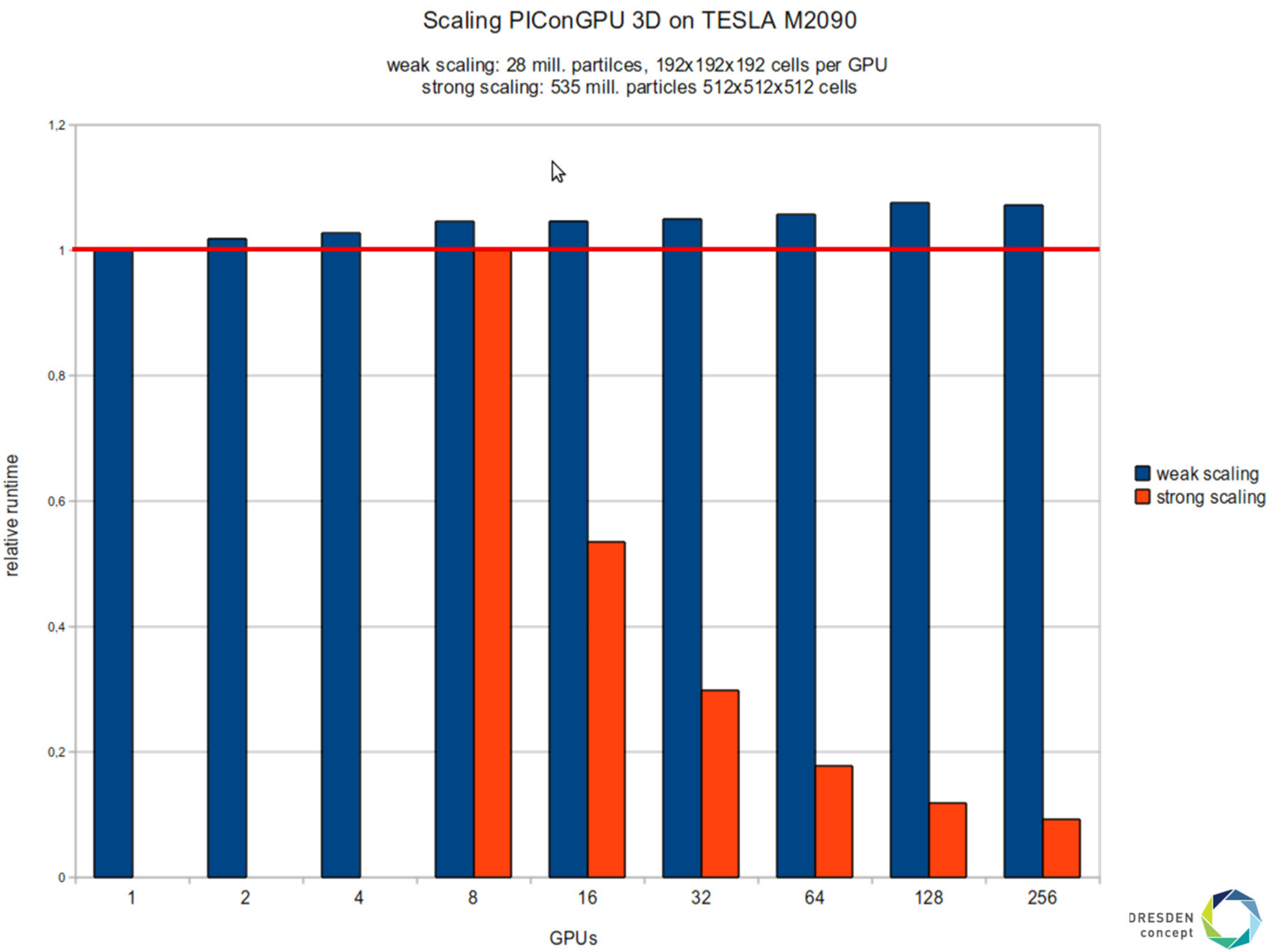
Yee-scheme /
Directional splitting
Maxwell-Solver

DRESDEN concept HZDR


PIConGPU
64 NVIDIA
Fermi GPUs
45 min
0.09 seconds per time step
GPU: 33
GPU: 35
GPU: 37
GPU: 39
GPU: 41
GPU: 43
GPU: 45
GPU: 47
GPU: 49
GPU: 51
GPU: 53
GPU: 55
GPU: 57
GPU: 59
GPU: 61


FORTRAN
LEGACY
128 AMD cores
8 days
480x480x3070 Cells, 2.8 x 10^9 macro particles, 30,000 time steps

Strong and Weak Scaling on NVIDIA TESLA M2090



Radiation calculations are highly parallel!

Spectral energy density

Blows up in ultrarelativistic regime

retarded time requires interpolation In FFT algorithms because of equidistant time steps

$$\frac{d^2 I}{d\omega\, d\Omega} = \frac{e^2}{4\pi^2 c} \left| \boldsymbol{P} \cdot \sum_j \int_{-\infty}^{+\infty} \frac{\boldsymbol{n} \times \left[(\boldsymbol{n} - \boldsymbol{\beta_j}) \times \dot{\boldsymbol{\beta}}_{\boldsymbol{j}} \right]}{(1 - \boldsymbol{\beta_j} \cdot \boldsymbol{n})^2} e^{i\omega(t - \boldsymbol{n} \cdot \boldsymbol{r_j}(t)/c)} dt \right|^2$$

Radiation amplitude for each **particle** at each **time** step, each observation **direction** and each **frequency**.

- n … normalized observation vector (Yes, this is a far field solution!)
- j … particle number
- β_j … normalized particle velocity
- r_j … particle position
- P … Polarisation filter vector ((1,1,1) accepts all polarisations)

Walter D. Jackson, *Classical Electrodynamics*, 3rd ed. (John Wiley & Sons, 1999),

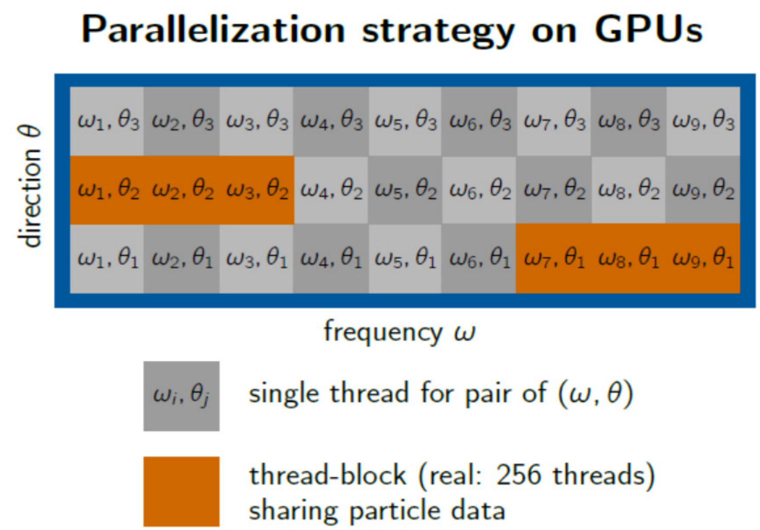
Discrete Fourier transform (DFT) is a good choice for GPU implementations.

- For FFT the trajectories of the particles need to be kept in memory, but not for DFT.

- DFT does not require interpolation of velocities and accelerations to equisdistant retarded times.

- From the physics side DFT is more flexible, because frequencies can be chosen arbitrarily.

  → Good for small-bandwidth „sky surveys“ for radiation localisation calculations or logarithmic spectra from X-ray to IR wavelengths.

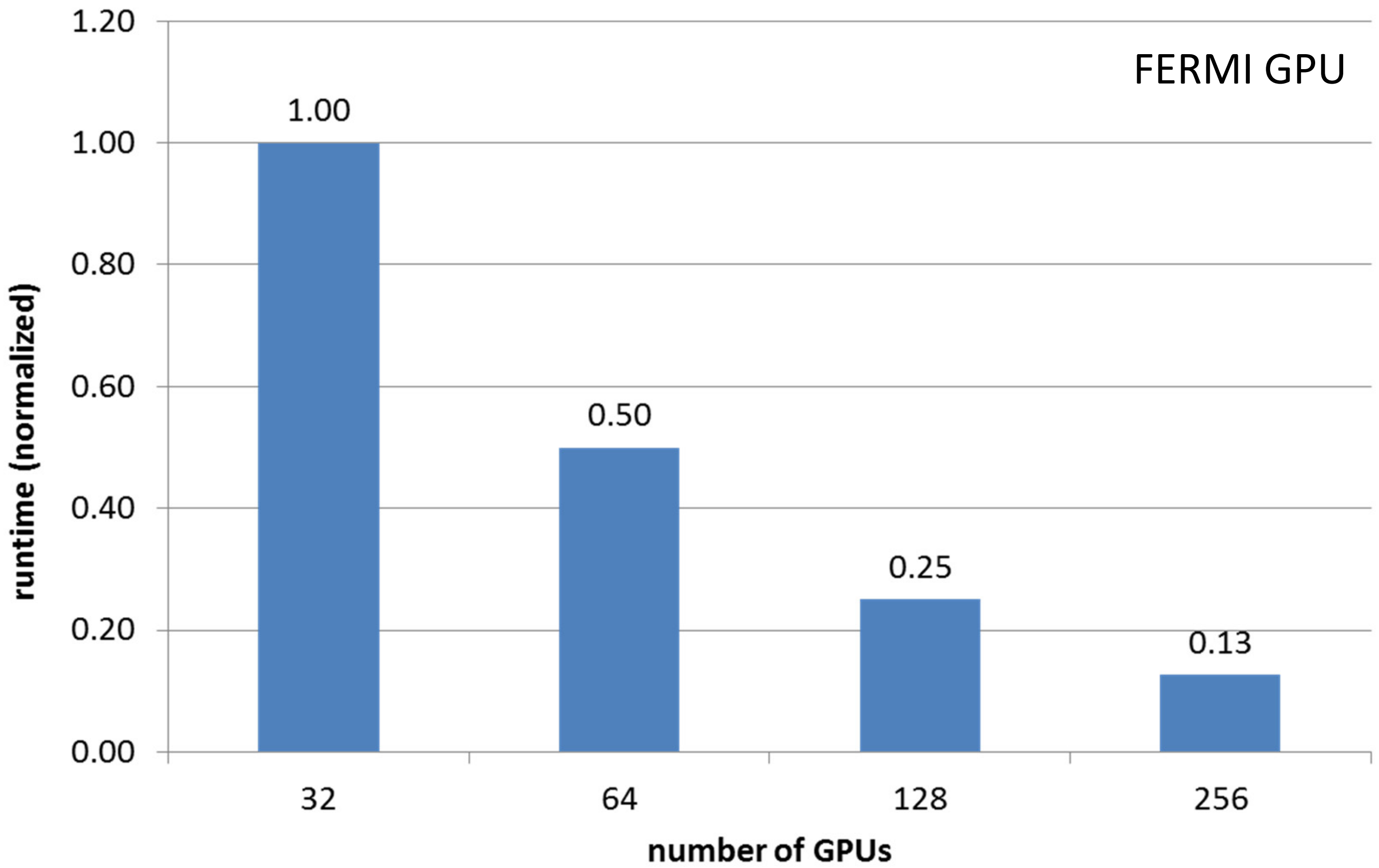


- However: Since resulting spectra at different observation directions have to be stored in CPU memory anyway, FFT is still interesting in a GPU-CPU-hybrid implementation. (GPU: Amplitudes, interpolation and superposition

  CPU: Calculating FFTs.)

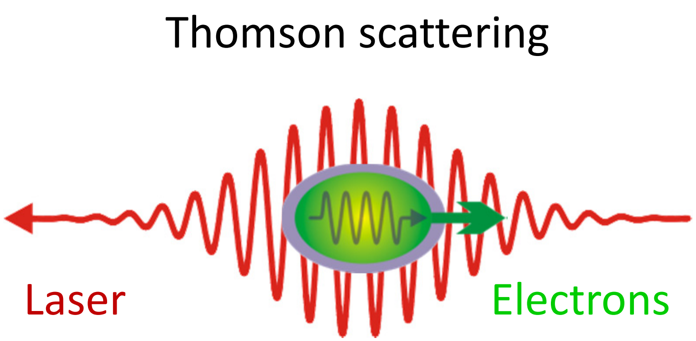
Strong scaling: radiation code becomes the main activity.

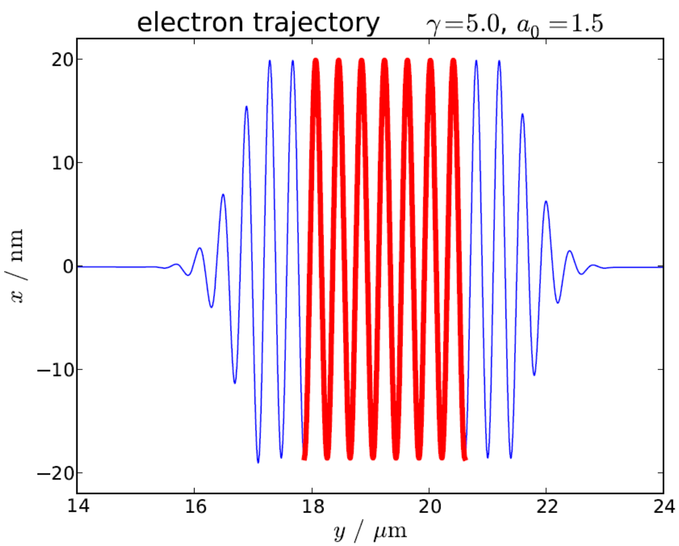
- Strong scaling is near ideal (overhead < 1%)
- 128 spectra à 2048 frequencies ; $1.1 \cdot 10^9$ particles in sim.; 70,000 part./s/GPU



The code is validated using analytic results of Thomson scattering.

- (Nonlinear) Thomson scattering from is a test for relativistic particles in high-intensity laser fields against analytic results.
- The core implementation of radiation code with the numerics was validated independently from the PIC-Code using predefined trajectories.
- The following tests (single particles and plasma) target the integration into PIConGPU, correct laser initialisation and numeric issues of the PIC code rather than the radiation code.

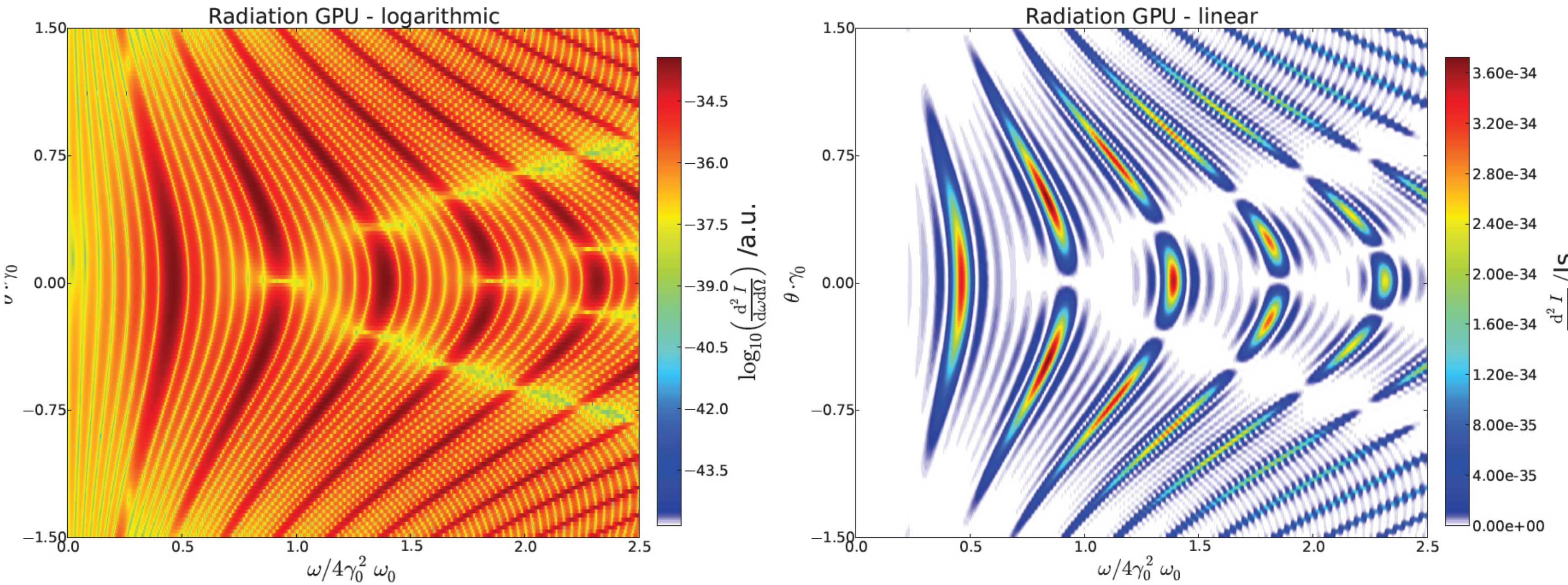




Nonlinear Thomson scattering with relativistic single particles

Analytic solution

- Peak laser intensity $I_0 = 4.9 \cdot 10^{18}$ W/cm²   ($a_0 = 1.5$)



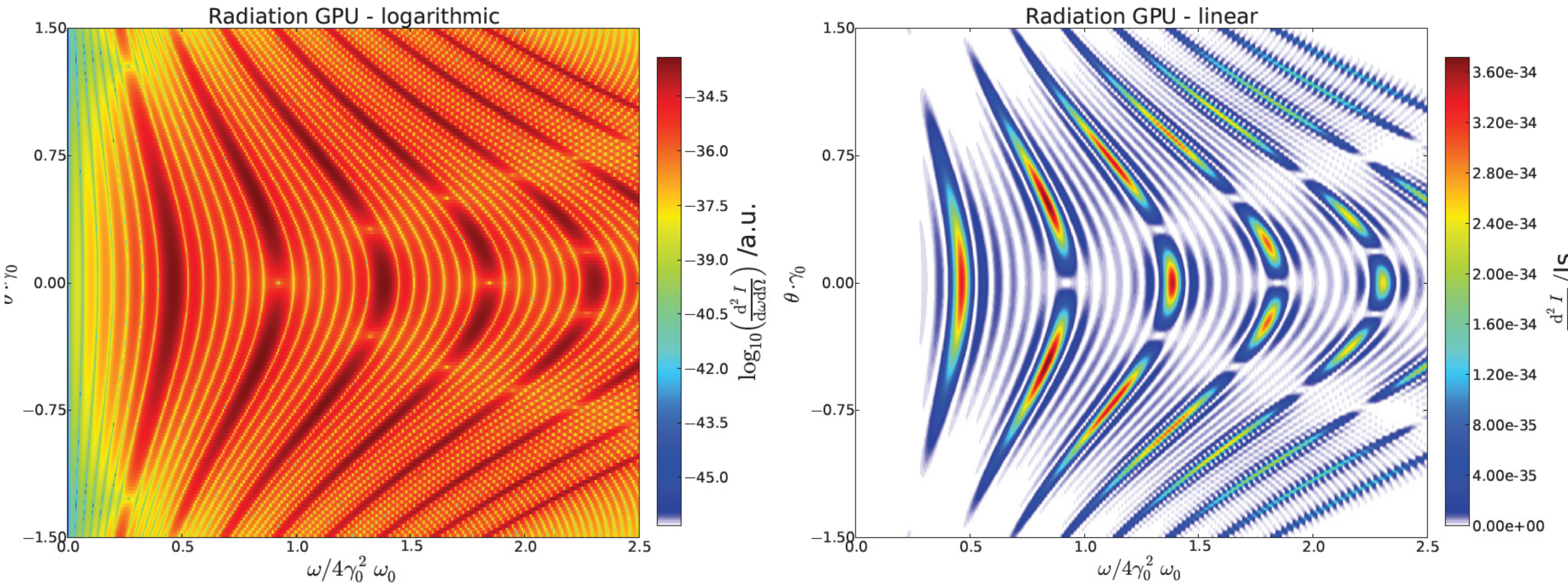
Theory:

Esarey *et al.*, Nonlinear Thomson scattering of intense laser pulses from beams and plasmas
Phys. Rev. E **48** (4), pp. 3003-3021, (1993).

DRESDEN concept HZDR

Nonlinear Thomson scattering with relativistic single particles

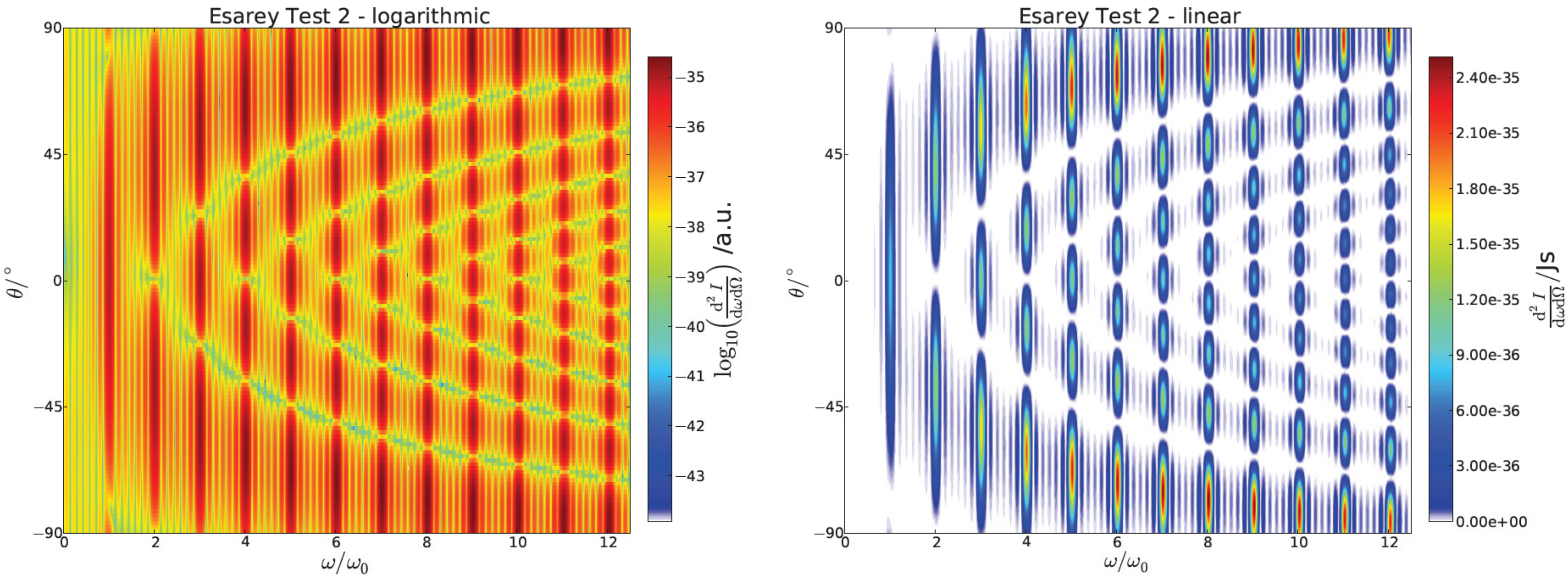
Numeric solution with PIConGPU



Nonlinear Thomson scattering in plasma

Analytic solution

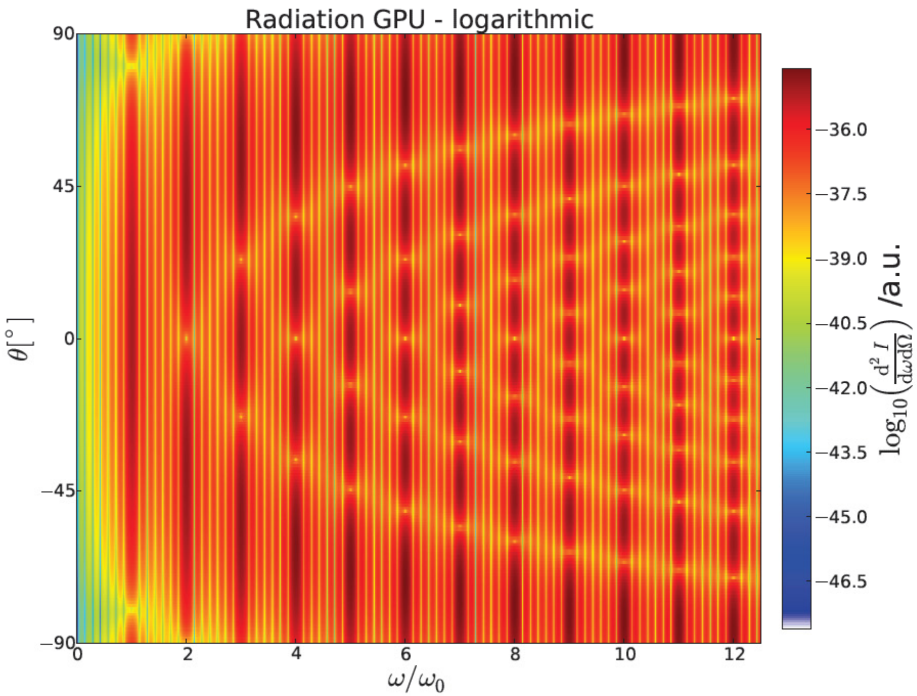
- Peak laser intensity $I_0 = 8.7 \cdot 10^{18}$ W/cm² ($a_0 = 2.0$)

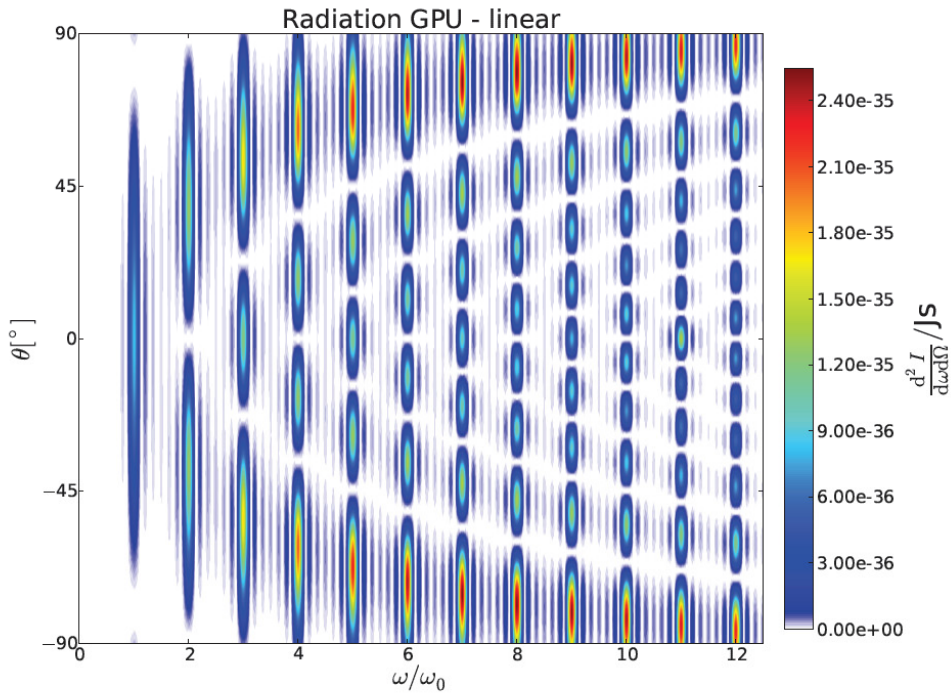


Theory:
Esarey *et al.*, Nonlinear Thomson scattering of intense laser pulses from beams and plasmas
Phys. Rev. E **48** (4), pp. 3003-3021, (1993).

Nonlinear Thomson Scattering in plasma

Numeric solution with PIConGPU

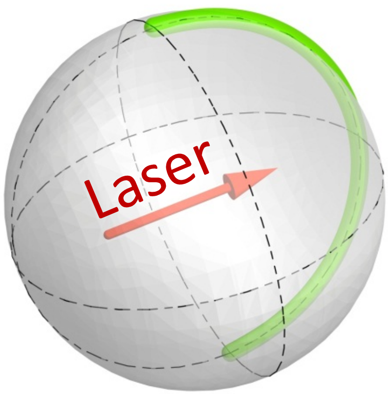


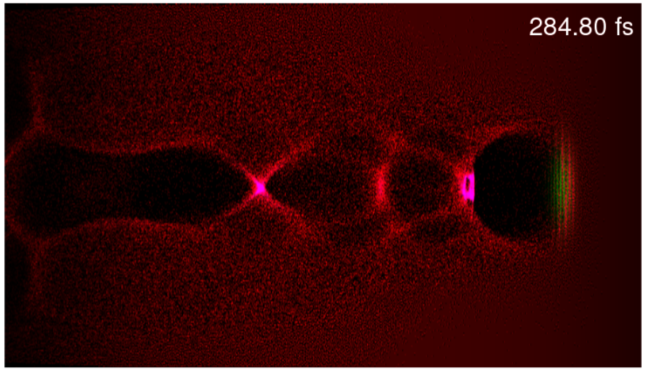


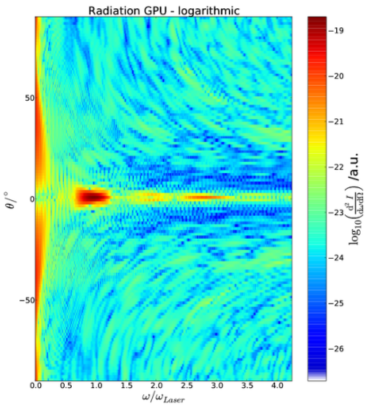
Very preliminary results: Radiation from Laser-wakefield accelerators

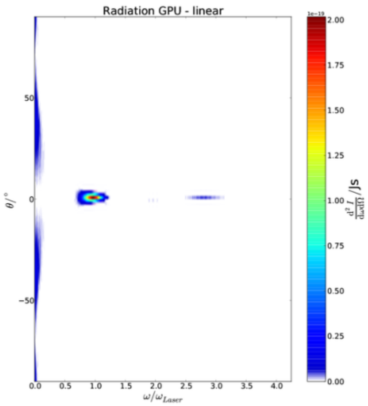
- 30 min without radiation → 23h 15min with 128 spectra à 2048 frequencies
- 66 Mio particles on 8 GPU; 10% of all macro particles radiate

Observation directions
around laser polarization





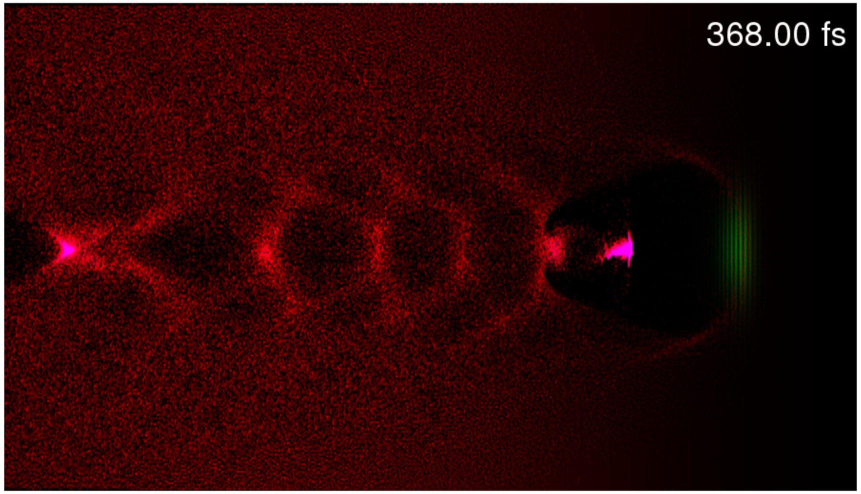


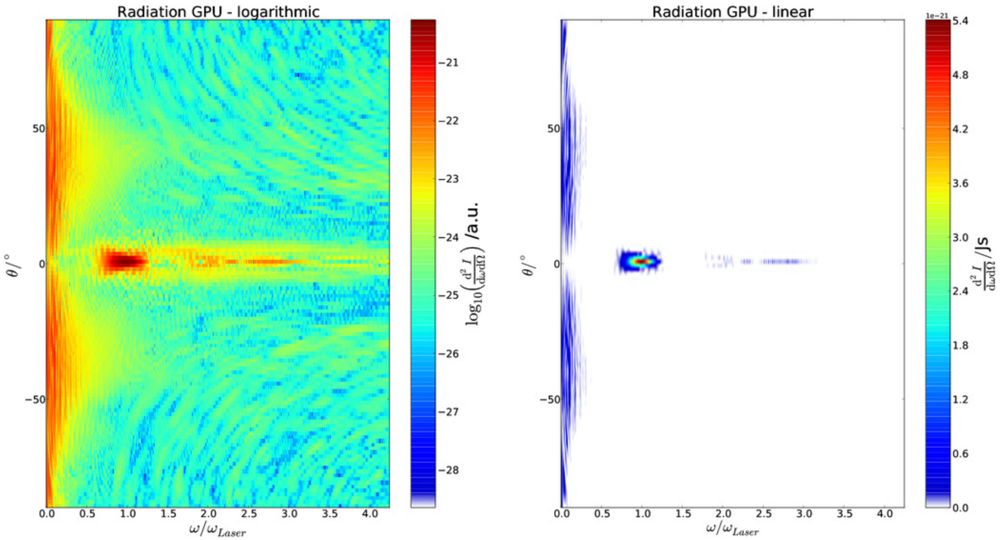


Disclaimer:
Wrong resolution due to lack of GPUs,
This is a test, not physics!

For full resolution ~1000 GPUs are required.

Very preliminary results: Laser-wakefield accelerator





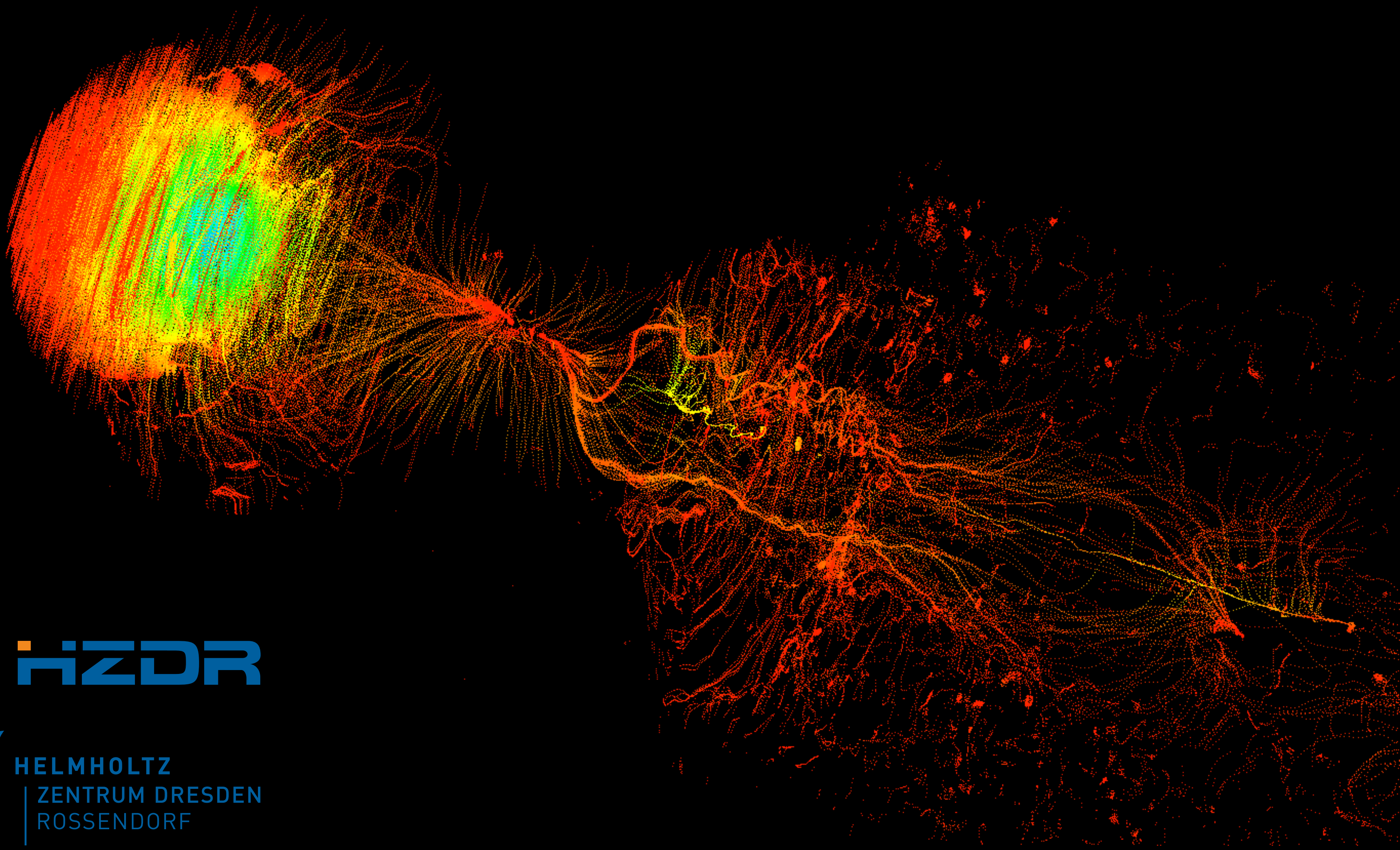
Conclusions

- The GPU-based 3D-PIC code PIConGPU provides the basis for full-scale radiation calculations using all simulated particles.
- For computing the radiation on GPU minimal memory footprint a discrete Fourier approach has been implemented.
- Benchmarks for strong scaling and physics tests for the integration of the radiation code into PIConGPU have been shown.
- First Laser-wakefield radiation tests provide *proof-of-principle*.

Near future:

- Full scale simulation LWFA PIC-simulation with more GPUs.
- FFT-based GPU-CPU Hybrid code (CPU provides memory and FFT) uniformly spaced spectral data in many directions over the full solid angle.

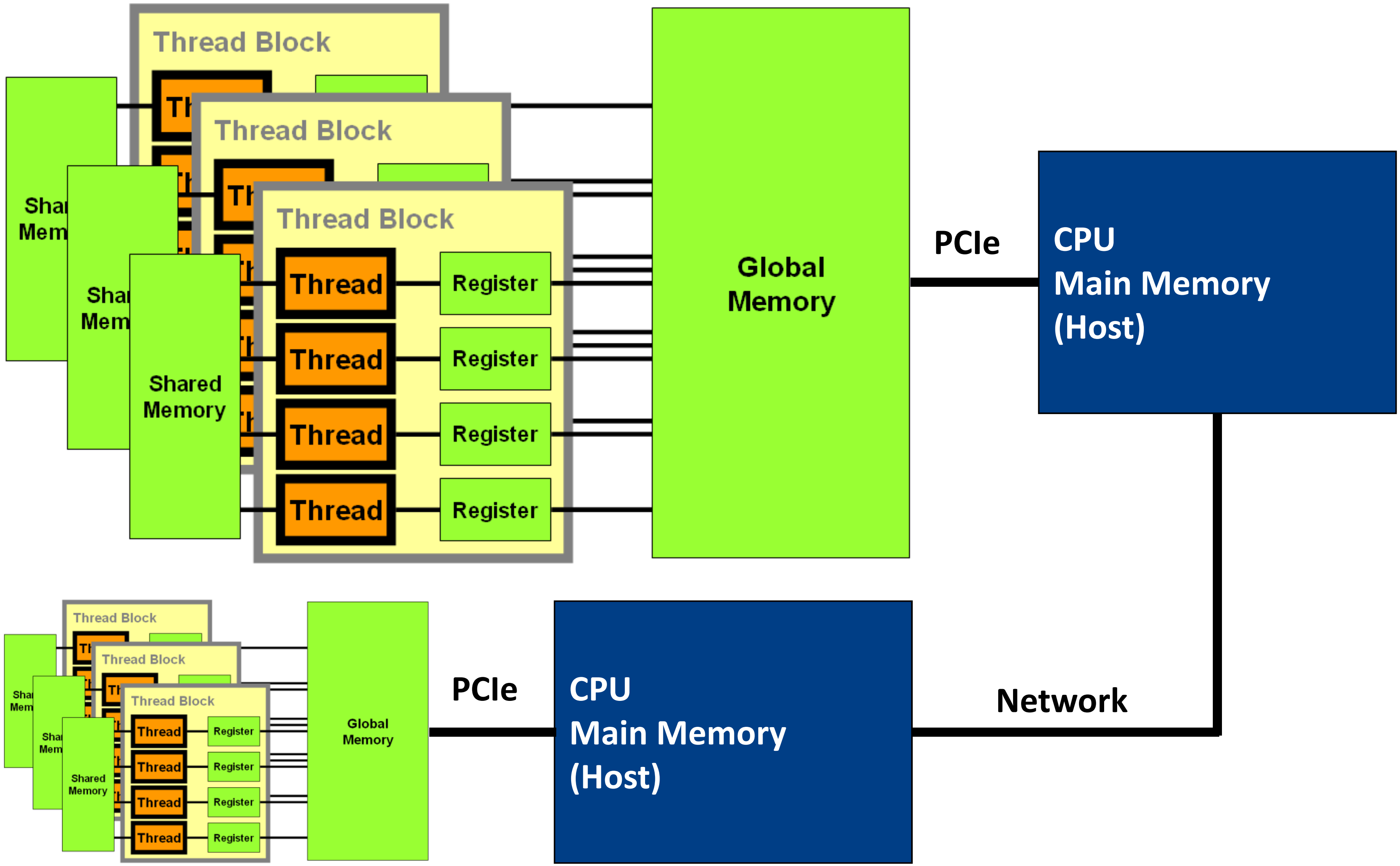
Thank you!



HZDR

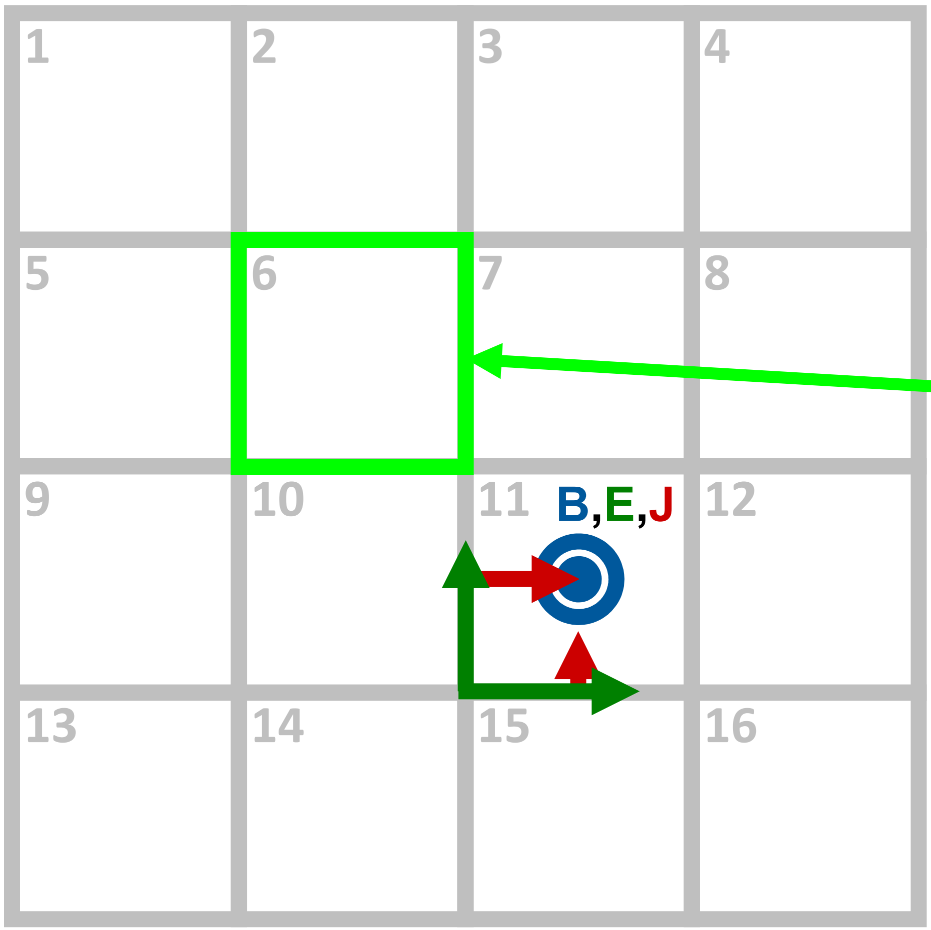
HELMHOLTZ
ZENTRUM DRESDEN
ROSSENDORF

"Almost all Programming can be viewed as an Exercise in Caching"

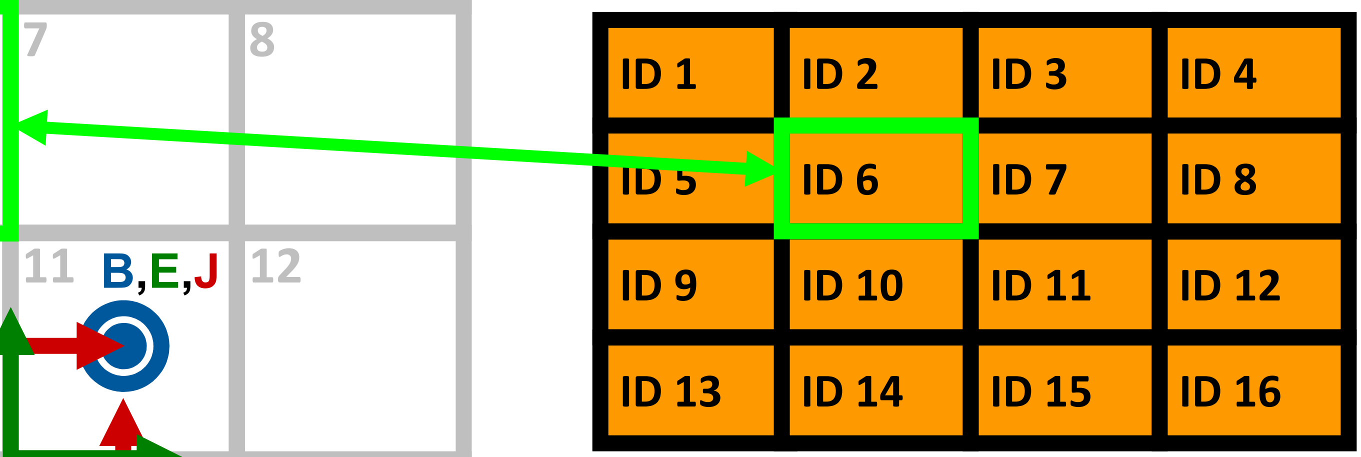


Everything depends on Data Structures *PART 1* — Cell-based Data

Super Cell

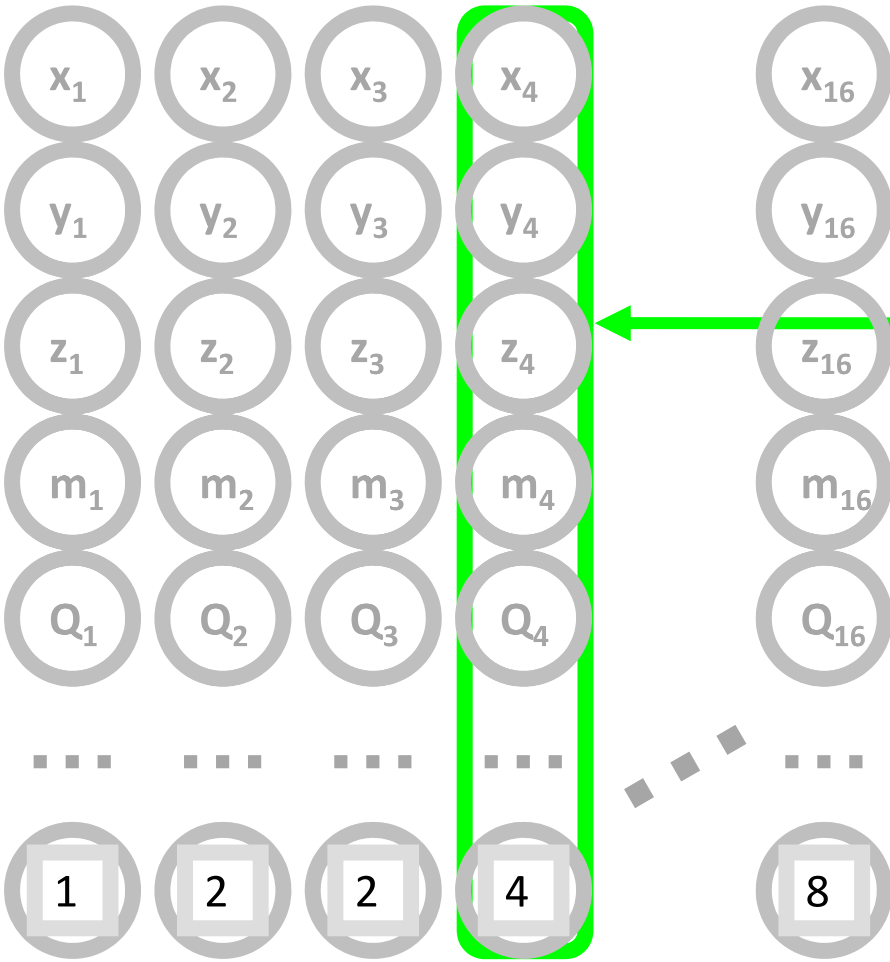


Each thread in a thread block corresponds to one cell in the super cell

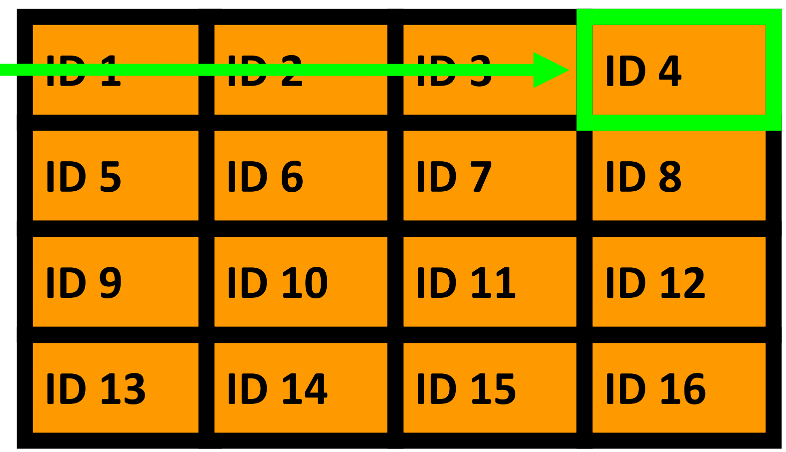


Everything depends on Data Structures *PART 2* — Particle Data

Attribute Frame (the data structure formerly known as „tile“, „pool“, ...)

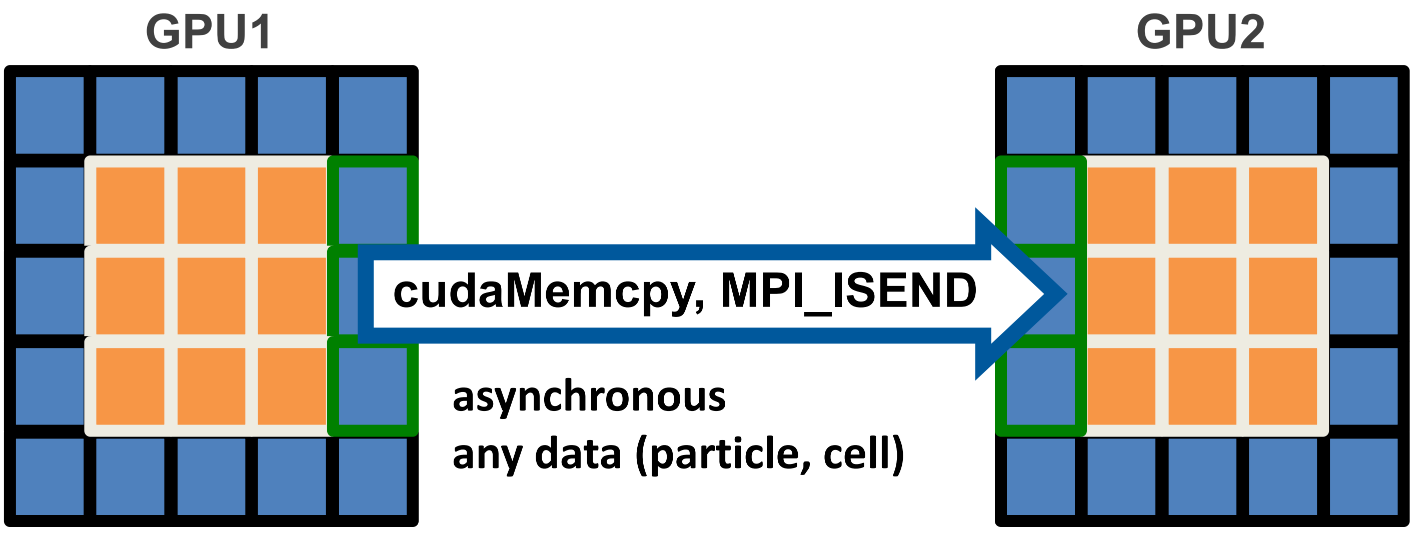


Each thread in a thread block is started for one particle in the frame



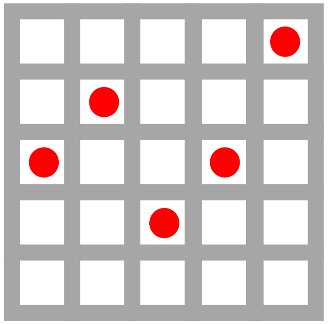
Cell index tells you the particle‘s local cell index inside the super cell

Encapsulation of all memory transfer functions



asynchronous
any data (particle, cell)

+



Local Domain Model
a thread can assume
that all the data it needs
is in its local memory
(*SISD*)

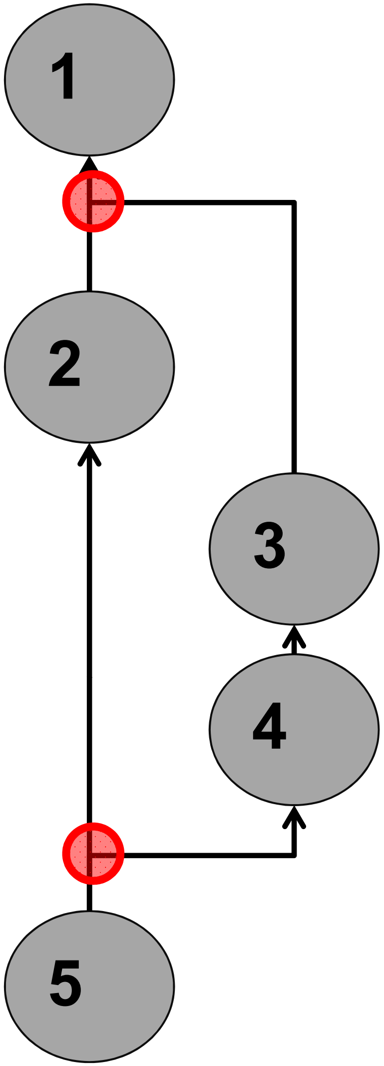
Events help to direct kernel execution, data exchange, etc.

EventTask tree

Parallel hierarchy types

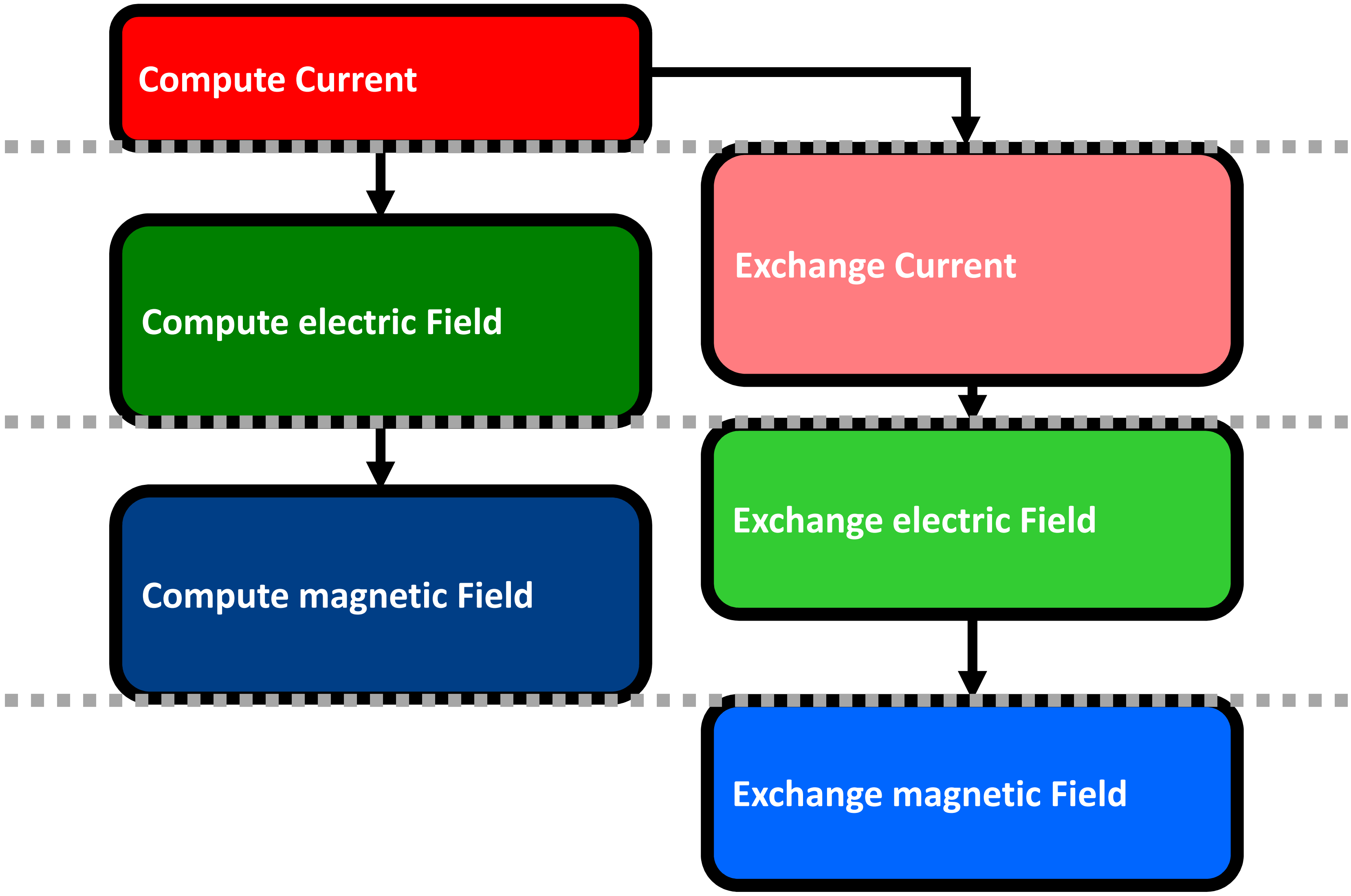
- TASK_HOST (not parallel, limitation of cuda runtime)
- TASK_CUDA (use streams for parallel work on gpu)
- TASK_MPI

Defining the algorithmic structure of a task graph




```
getSizeFromDevice();                       // memcpy from device

EventTask e_split = __getTransactionEvent();
__startTransaction( e_split );             // depend from e_split
cudaKernel_X();                            // create a kernel task
EventTask e1 = __endTransaction();

__startTransaction( e_split );
cudaKernel_Y();

synchronizeWithNeighbor();                 // memcpy+mpi task
EventTask e2 = __endTransaction();
__setTransactionEvent( e1+e2 );   // combine parallel work

cudaKernel_Z();
```

Communicate and compute concurrently using Tasks



Strong and Weak Scaling on NVIDIA FERMI