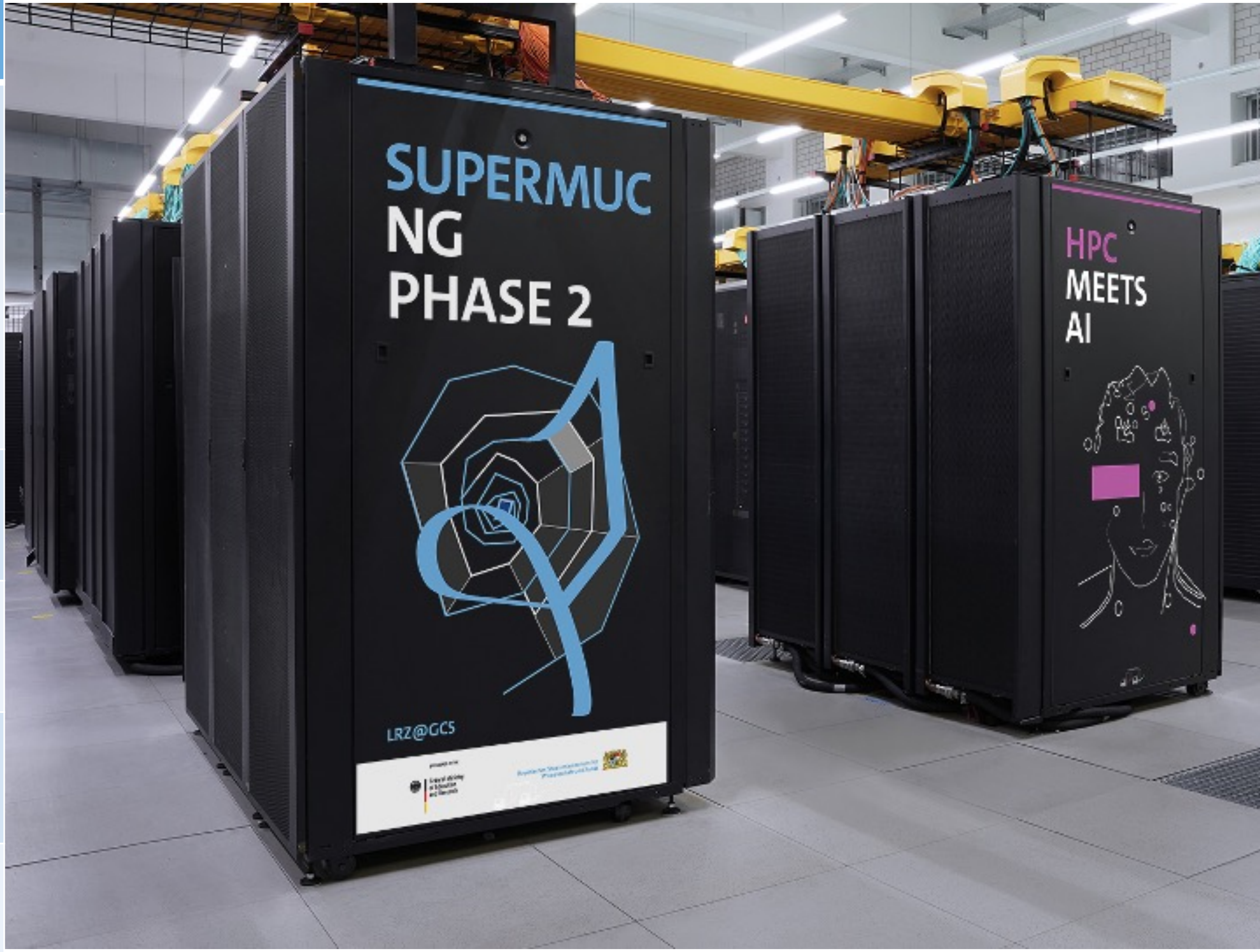
Node-Level Performance and Energy Characterization of Flagship Science Applications on SuperMUC-NG Phase 2

ISC 2026 IXPUG Workshop | June 26th 2026

Salvatore Cielo[1], Elmira Birang[1], Alexander Pöppl[2], Sajad Azizi[1],
Plamen Dobrev[1], Margarita Egelhofer[1], Ivan Pribec[1], Gerald Mathias[1]
[1)]Leibniz Supercomputing Center (LRZ), [2)] Intel Corporation

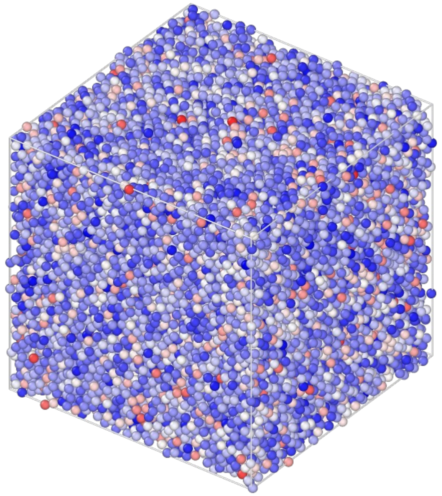
SuperMUC-NG Phase II

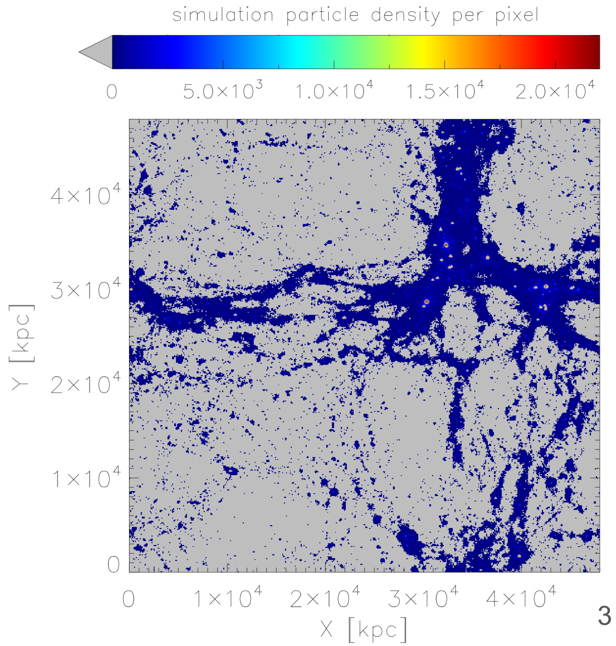
	SuperMUC-NG Phase II
Number of compute nodes	240
Architecture	56 core Intel Xeon Platinum 8470 ("Sapphire Rapids") Intel Data Center GPU Max 1550 ("Ponte Vecchio") 2 CPUs and 4 GPUs per node
Performance	28 PFLOPs (Peak) 17.2 PFLOPs (HPL)
Main memory	159 TB DDR5 + 123 TB HBM
Storage	1 PB DAOS
Throughput memory to disk	750 GB/s



Motivation

- Energy costs are a significant chunk of overall costs of an HPC system
- Thus, throughput is complemented by energy efficiency as a metric for system evaluation
- List of evaluated applications
  - AthenaK (Astrophysics/Cosmology, Kokkos)
  - OpenGadget (Astrophysics/Cosmology, OpenMP Offloading)
  - GROMACS (Molecular Dynamics, SYCL/OpenCL)
  - Deal.ii (PDE/FEM, Kokkos/SYCL/TinyTensorCompiler)
  - LAMMPS (Molecular Dynamics, Kokkos)

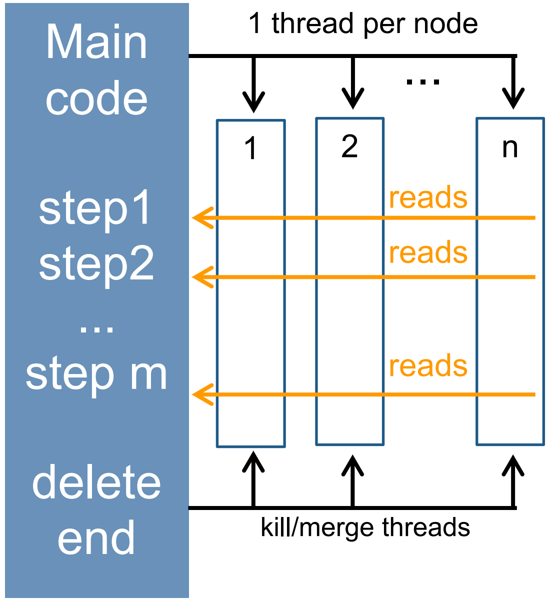




The energy meter p3em: p^3 stands for...

Portable

- Spawns one thread per node
- reads cumulant node energy into main code
- API in C C++ Fortran Python



Permitted

Each thread reads any user-decided app/script.

- ➢ No access rights to counters
- ➢ build your own meters...
- ➢ ... or use what you have on the system:

Ships plug&play samples

```
perf, likwid, hwmon,
xpu- nvidia- amd- smi
...
```

Parsable

One measure per timestep, same as time-to-solution.

- no second backbone for energy
- no new set of parsing tools

Swap 🕘⟹🔌⚡ in the code's internal timers and use your existing parsing workflow. E.g.:

std::chrono::system_clock::now()

⇓

p3emChrono::system_clock::now()

Energy Measurements — EAR & p3em

p3em	EAR
GitHub: svtcli/p3em	GitHub: eas4dc/EAR
Lightweight power measurement API that is easy to integrate for non-admin users	Energy management framework for super-computers: provides optimization, accounting & management
provides very simple API in C, C++, Python, Fortran	EAR daemon for node-monitoring, accounting & reporting to DB
Wraps any monotonic command-line energy reading tool	EAR library for energy-aware intercept of MPI calls, amongst others
Command line call must be provided by library user	Requires installation by cluster administrators.

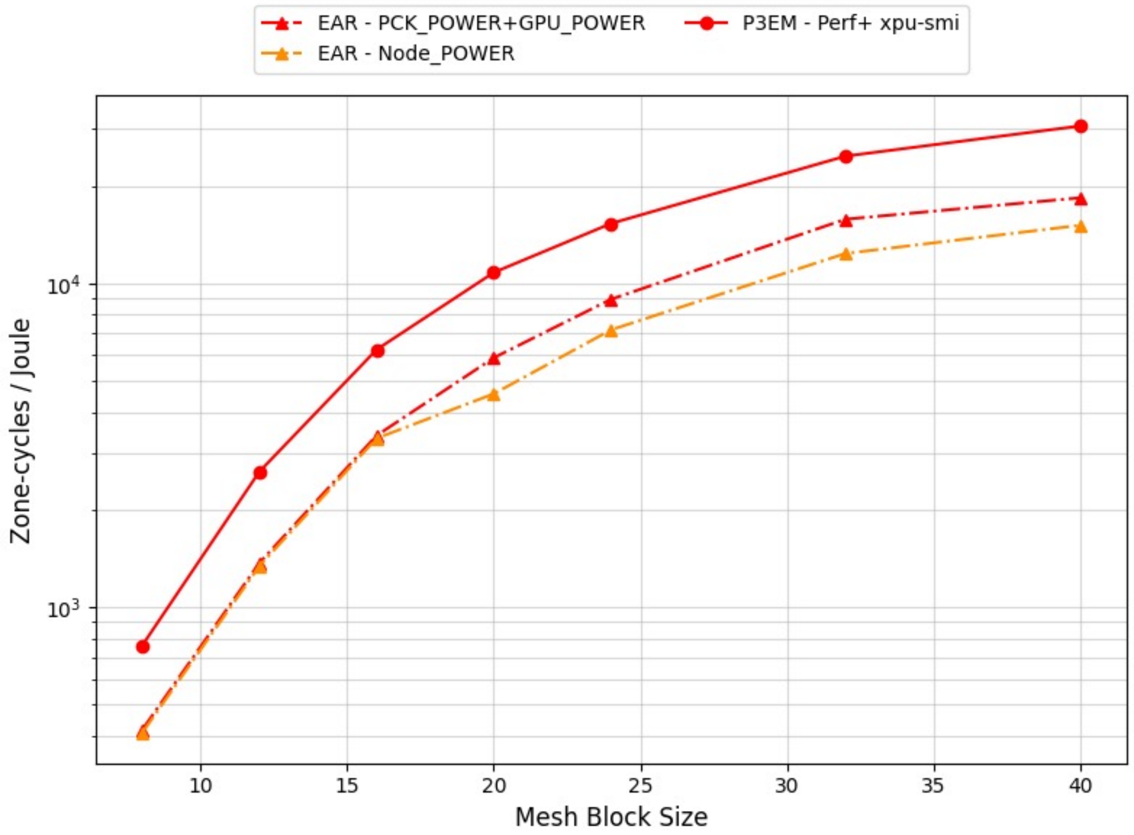
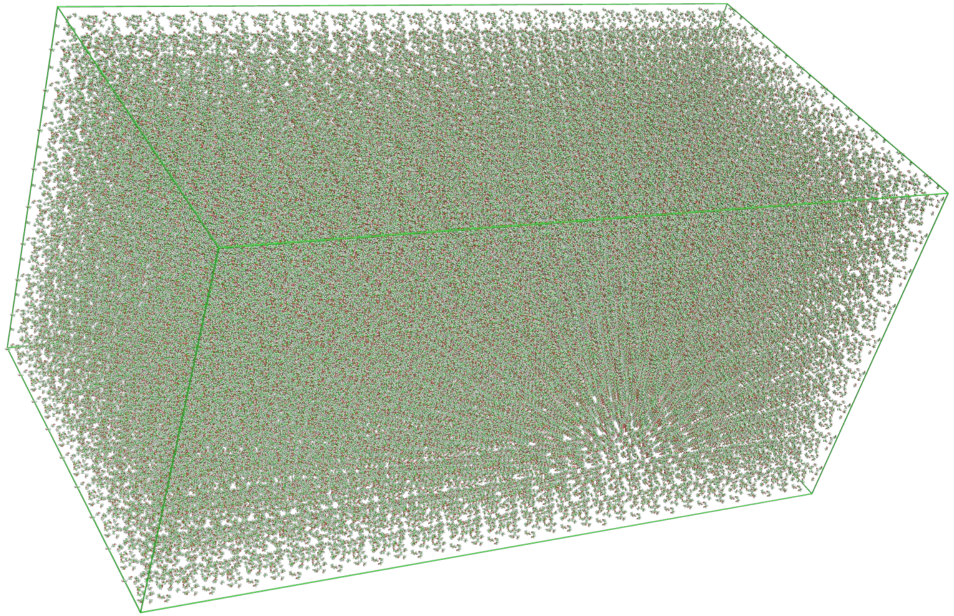
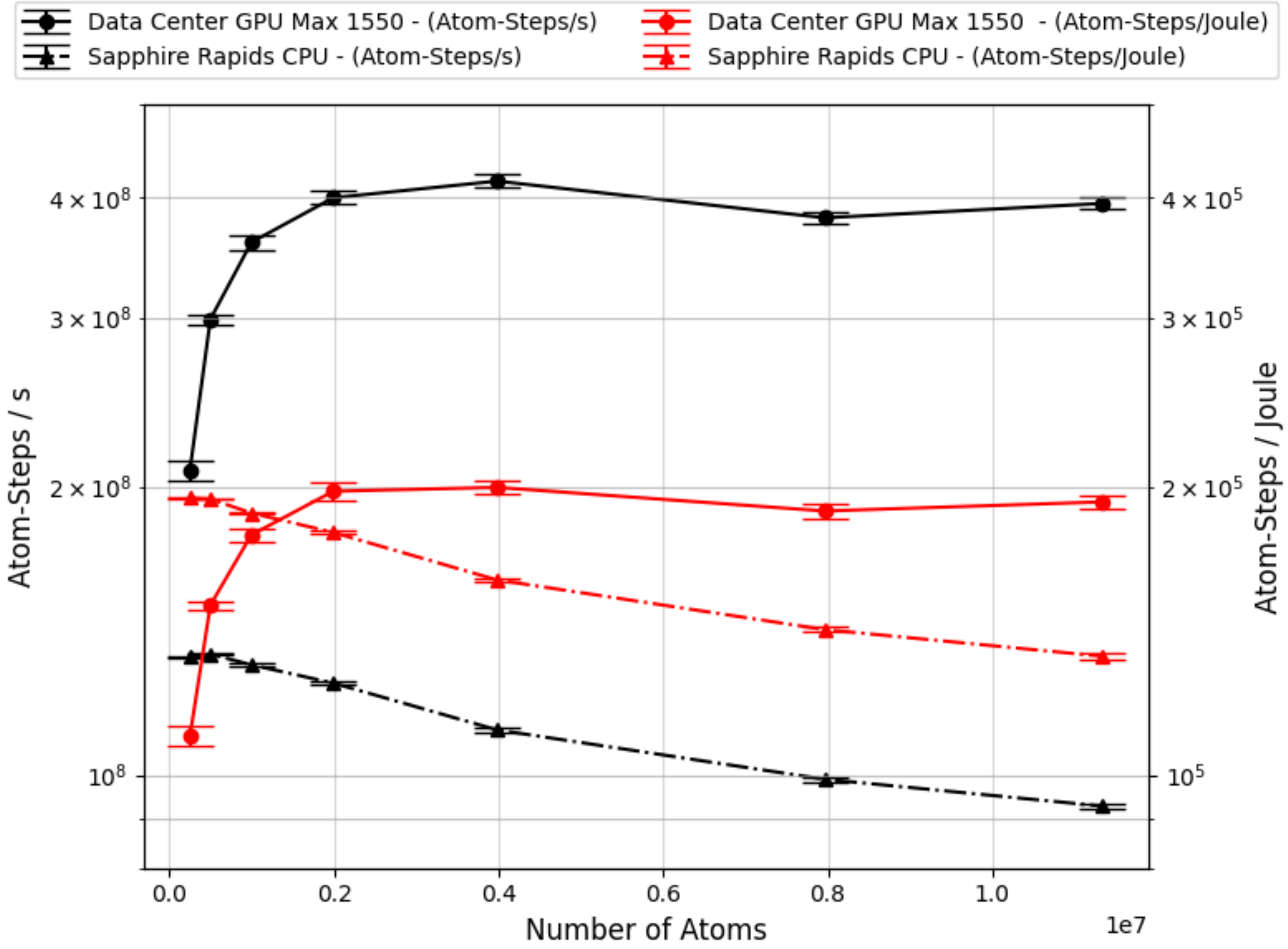


Figure: Comparison of EAR vs. p3em (with xpu-smi) using AthenaK as a benchmark

GROMACS

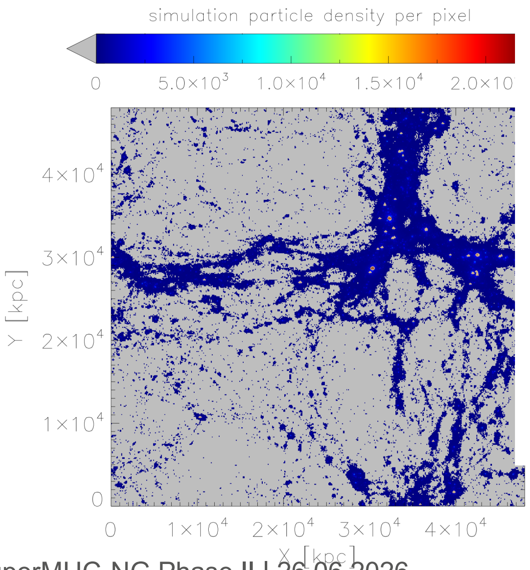
- Suite for classical molecular dynamics
- In GPU runs, most work executes on the GPU: seven tiles perform short-range interactions, and one tile handles long-range interactions via particle-mesh Ewald (PME)

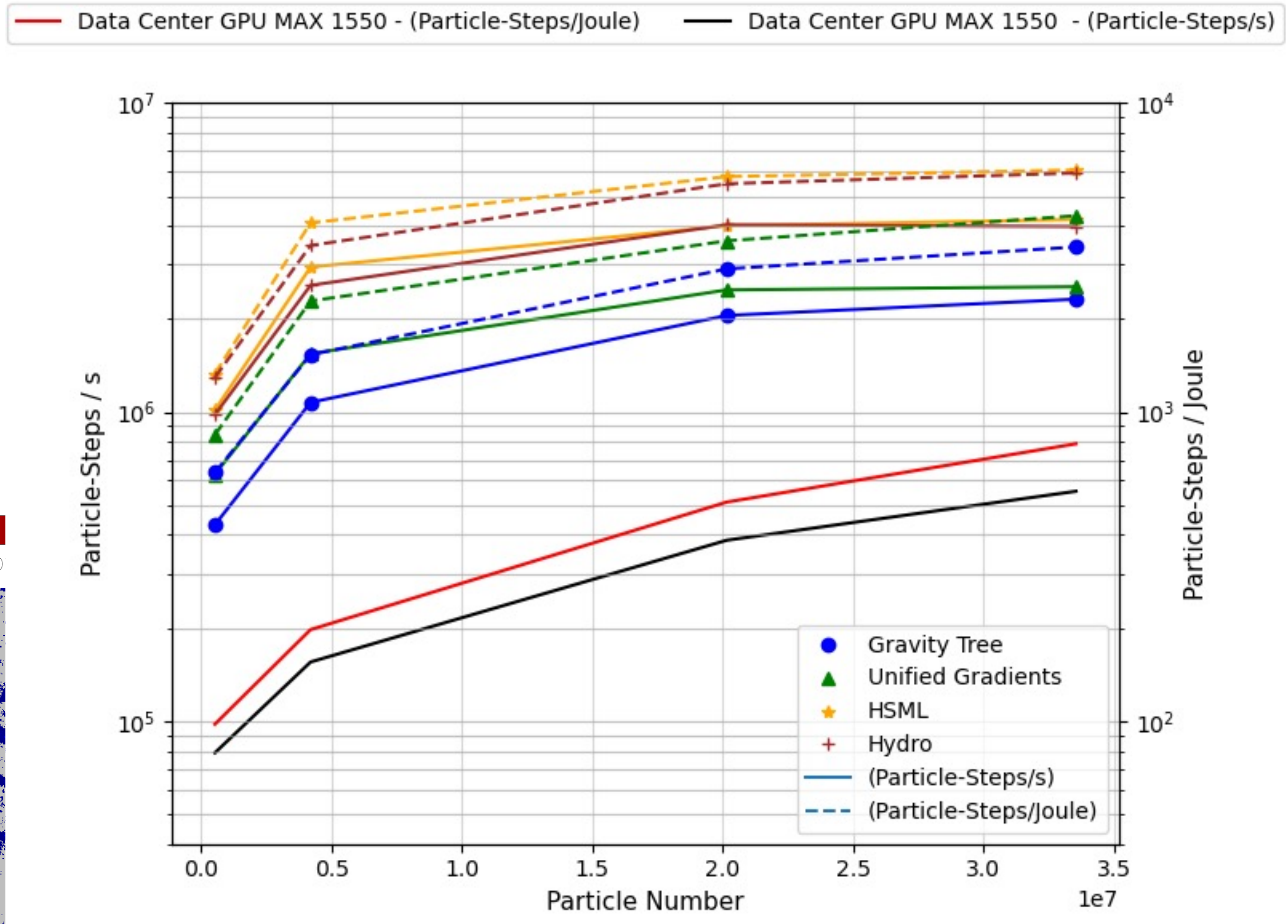




OpenGadget

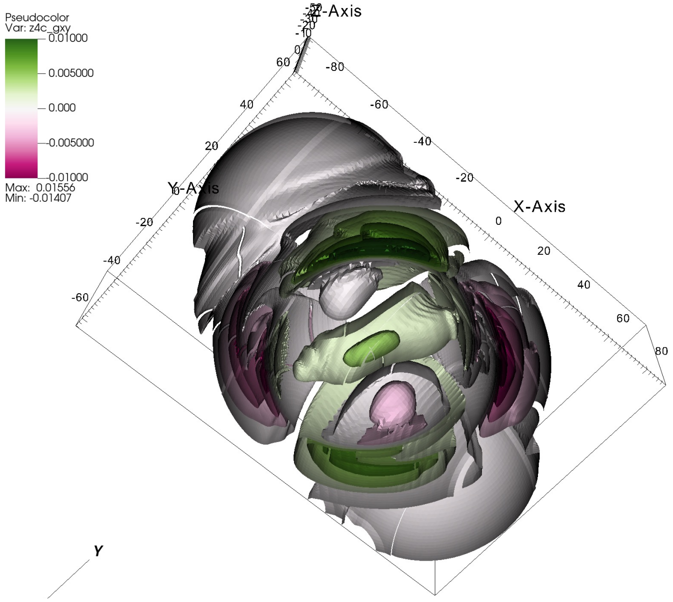
- OpenGadget3 is a large-scale cosmological hydrodynamics and N-body simulation code
- Code partially runs on the GPU: gravitational tree, smoothing lengths (HSML), conduction, hydrodynamics, and density, while the remaining computations stay on the CPU.
- Timings and energy use collected using a hierarchical timer structure

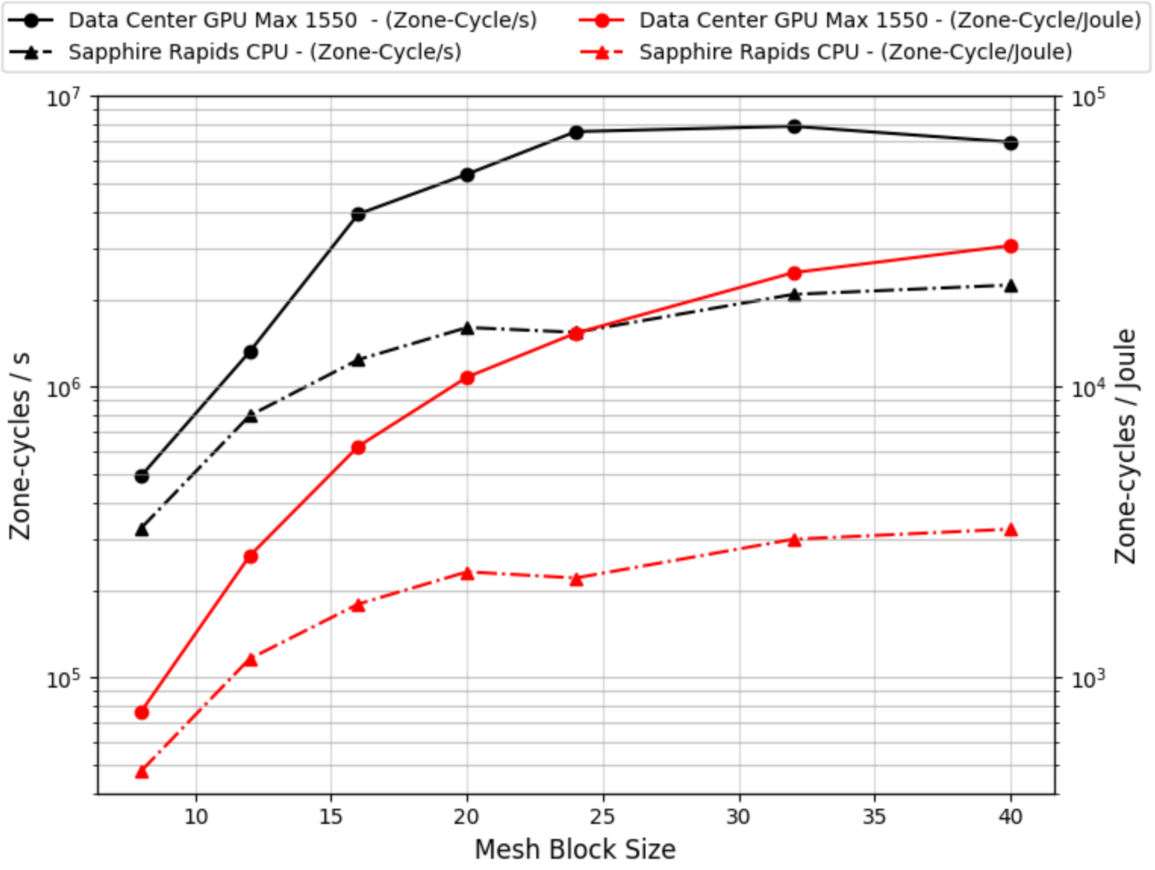




AthenaK

- AthenaK is an astrophysical MHD code built with Kokkos
- The benchmark uses the GW150914 binary black hole merger problem to evaluate performance.
- Mesh-block edge length determines the work granularity assigned to each GPU tile.
- Requires mesh-block sizes ≥ 10 to efficiently use GPU





Summary and Outlook

- Wide Range of applications have been benchmarked and are running on SuperMUC-NG Phase II
- Performance-portability frameworks such as Kokkos or widely used parallelization frameworks like OpenMP Offloading provide an easy on-ramp for running codes on Intel GPUs.
- High utilization is important for energy-efficiency
- Efficiency often correlates with throughput, not always identical.

Get P3em
on GitHub