



ENERGY EFFICIENCY IN HPC

UM6P-FZJ WORKSHOP

27 November 2025 | Andreas Herten | Forschungszentrum Jülich, Jülich Supercomputing Centre

Content

Before Exascale

JUPITER

JUPITER Booster

JUPITER Cluster

Exascale System

Comparison

JUPITER Results

TOP500

Pictures

Build-Up

Optimization Work

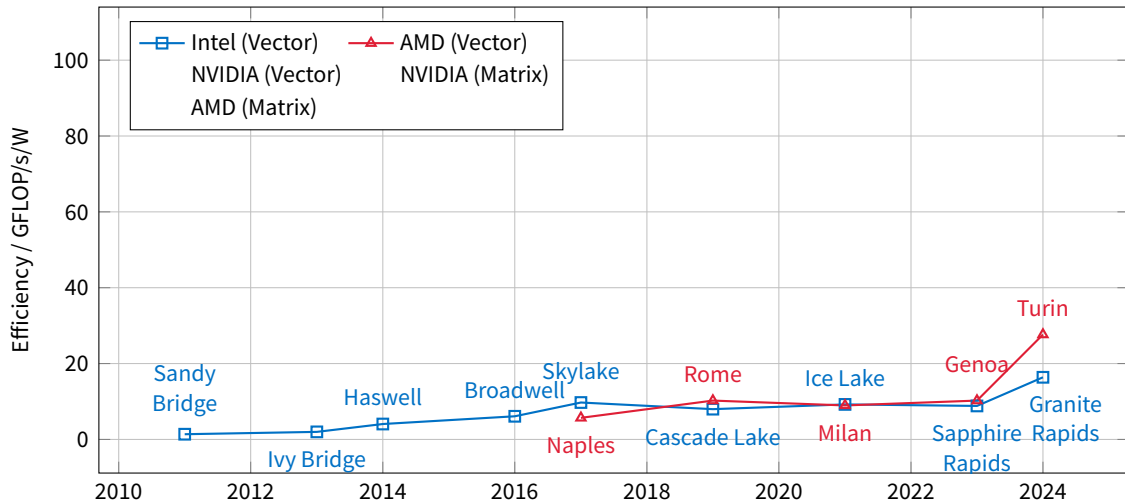
JUREAP & Early Users

Scaled Block Vecchia

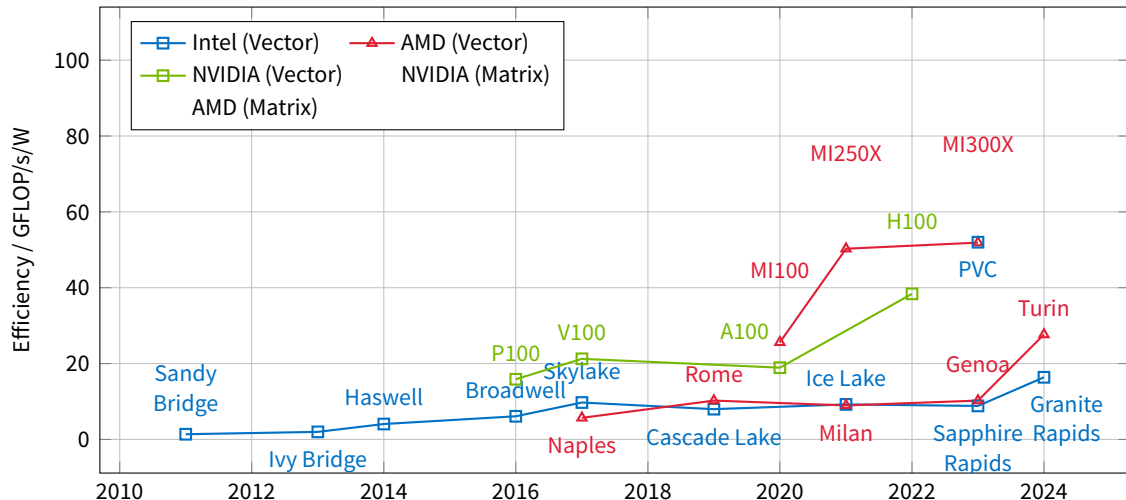
Previous Work

Hardware

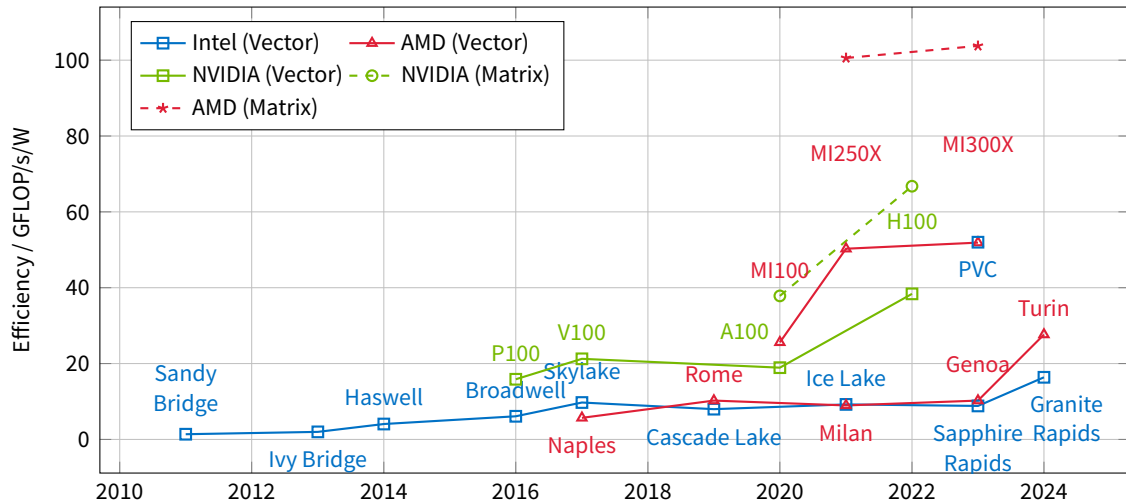
Energy Efficiency State-of-the-Art



Energy Efficiency State-of-the-Art



Energy Efficiency State-of-the-Art



Ways for Optimization

- Optimizing energy efficiency means balancing energy intake and compute time

Optimize Energy = Minimize(Energy Usage $\times$ Compute Time) = Minimize EDP

Ways for Optimization

- Optimizing energy efficiency means balancing energy intake and compute time
- But: Currently no incentive for users! They get core-h, not Wh.

Optimize Energy = Minimize(Energy Usage $\times$ Compute Time) = Minimize EDP

Ways for Optimization

- Optimizing energy efficiency means balancing energy intake and compute time
- But: Currently no incentive for users! They get core-h, not Wh.
- Approximation: Energy $\approx$ Performance

Optimize Energy = Minimize(Energy Usage $\times$ Compute Time) = Minimize EDP
 $\approx$ Optimize Performance

Ways for Optimization

- Optimizing energy efficiency means balancing energy intake and compute time
- But: Currently no incentive for users! They get core-h, not Wh.
- Approximation: Energy $\approx$ Performance
- Add also data center and SOTA perspective

Optimize Energy = Minimize(Energy Usage $\times$ Compute Time) = Minimize EDP
 $\approx$ Optimize Performance
 $\rightsquigarrow$ Optimize Performance $\otimes$ Utilize Best-Fitting Hardware

JUPITER System Overview

- ParTec/Eviden consortium
- Implementing Modular Supercomputing Architecture



JUPITER System Overview

- ParTec/Eviden consortium
- Implementing Modular Supercomputing Architecture
- JUPITER **Booster**: High scalability, 1 EFLOP/s HPL, > 40 EFLOP/s FP8
≈ 6000 nodes: 4× Grace-Hopper superchip, 4× network
- JUPITER **Cluster**: High versatility, high memory-to-compute ratio
≈ 1300 nodes: 2× SiPearl Rhea1 (HBM), 1× network

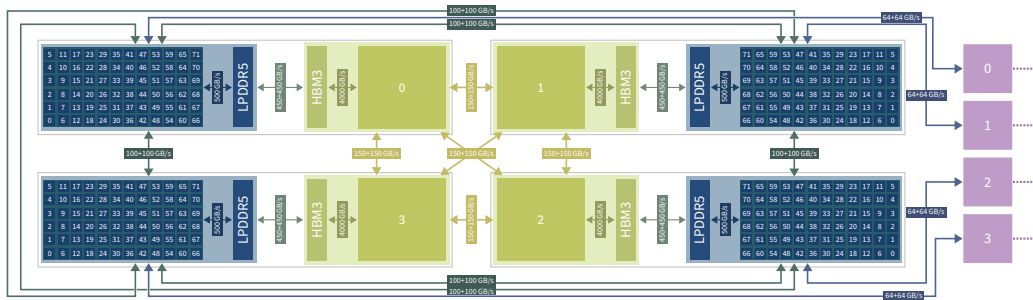


JUPITER System Overview

- ParTec/Eviden consortium
- Implementing Modular Supercomputing Architecture
- JUPITER **Booster**: High scalability, 1 EFLOP/s HPL, > 40 EFLOP/s FP8
≈ 6000 nodes: 4× Grace-Hopper superchip, 4× network
- JUPITER **Cluster**: High versatility, high memory-to-compute ratio
≈ 1300 nodes: 2× SiPearl Rhea1 (HBM), 1× network
- Network: 200/400 Gbit/s NVIDIA InfiniBand NDR (DragonFly+)
- Storage: 29 PB flash, 310 PB HDD, 370 PB tape
- Energy: 17 MW limit (HPL); direct liquid-cooled, energy re-use
- Modular data center of containers 



JUPITER Booster Node Design



4 × GH200 superchip (CG4)

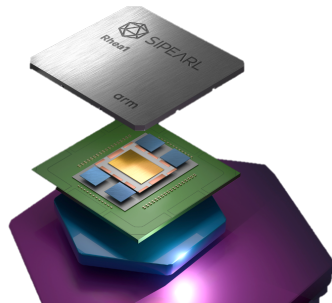
- Hopper: 132 SMs
- Grace: 72 Arm Neoverse-V2 cores
- 900 GB/s NVLink C2C
- 120 GB LPDDR5X memory, 500 GB/s (CPU)
- 96 GB HBM3 memory, 4000 GB/s (GPU)

TDP superchip: 680 W

- GPU-to-GPU: NVLink 4, 300 GB/s per pair
- CPU-to-CPU: cNVLink, 200 GB/s per pair
- Cache coherence between all GPUs and CPUs
- 4 × NDR200 InfiniBand (200 Gbit/s)

JUPITER Cluster Node Design

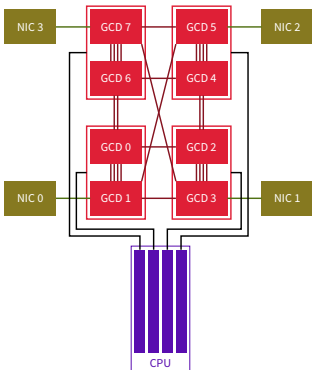
- CPU-based based design (no accelerators)
- Nodes with $2 \times$ SiPearl Rhea1 CPU, each
 - 80 Neoverse-V1 cores
 - 256 GB DDR5 memory
 - 64 GB HBM memory
- Origins in European HPC ecosystem: EPI, EUPEX, and further projects
- To be deployed soon



Other Exascale Systems

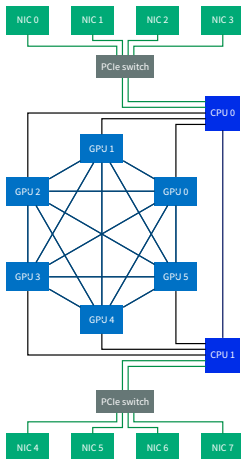
Frontier

ORNL, USA: 1.2 EFLOP/s



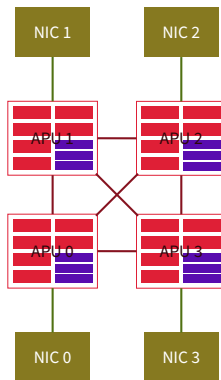
Aurora

ANL, USA: 1 EFLOP/s



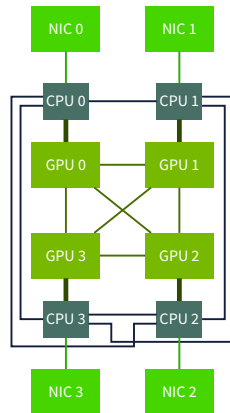
El Capitan

LLNL, USA: 1.7 EFLOP/s



JUPITER

JSC, DE: 1.0 EFLOP/s



Overview Exascale Systems

Unique features

Frontier AMD GPUs in GCD design (two chip dies), few cores

Aurora 6 Intel GPUs, HBM-equipped Intel CPUs

El Capitan AMD APU design, no *host* memory

JUPITER Superchip design, tight combination between CPU-GPU

Overview

System	Nodes	per Node					per GPU		
		TFLOP/s [§]	GPUs	CPUs	NICs	Cores	TFLOP/s	Cores	NICs
Frontier	9604	141	4	1	4	64	35	16	1
Aurora	10 624	95	6	2	8	112	16	19	1.3
El Capitan	11 136	156	4	4	4	96	39	24	1
JUPITER	5884	175*	4	4	4	288	43	72	1

§: HPL value divided by number of nodes; for theoretical values, see [appendix](#)

*: For TOP500-reported HPL run with 5735 nodes

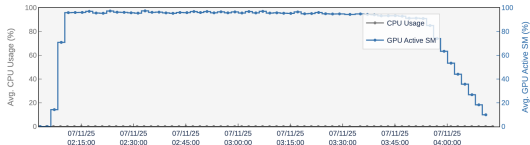
JUPITER TOP500

- TOP500 Nov 2025: #4 world, #1 EU
- 1000.184 PFLOP/s HPL
1 JUGENE (2008) over 1 EFLOP/s

Average
CPU Usage



Job-Usage Overview



Average
GPU Usage



LLview job report of HPL job

JOINING FORCES



jupiter.fz-juelich.de



Certificate from first listing in June 2025

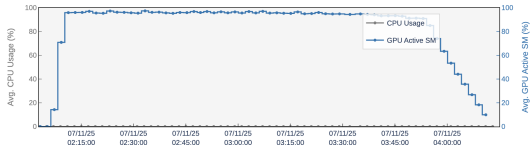
JUPITER TOP500

- TOP500 Nov 2025: #4 world, #1 EU
- 1000.184 PFLOP/s HPL
1 JUGENE (2008) over 1 EFLOP/s
- Green500 Nov. 2025: JUPITER #14
(63.3 GFLOP/(s W) no dedicated run)
Jun. 2025: JEDI #1 (72.7 GFLOP/(s W), many tricks)

Average
CPU Usage



Job-Usage Overview



Average
GPU Usage



LLview job report of HPL job

JOINING FORCES



jupiter.fz-juelich.de



Certificate from first listing in June 2025

MDC Concrete Foundation; Andreas for scale



Delivery of first entry hall containers



Delivery of data hall



Delivery of
IT Module 2



Photo by Herwig Zilkner

*Installation of
Compute Blades
in IT Module 1*



Photo by Herwig Zilken / FZJ



MDC bird's eye view

Photo by Sascha Kieklau / FZJ



110V2

ABB

Inside of IT module

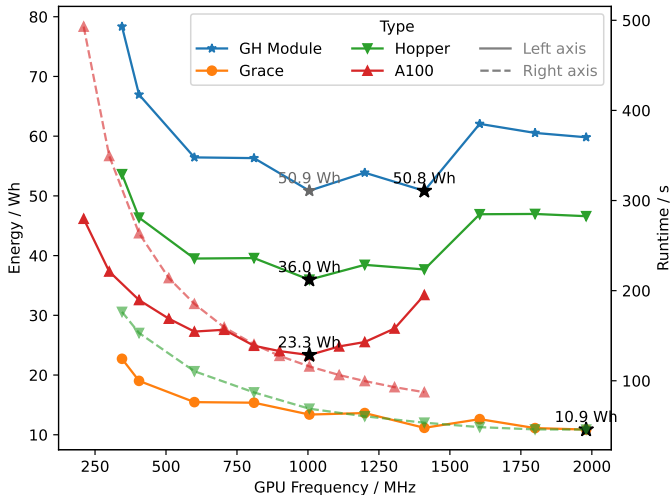
Photo by Sascha Kieklau / FZJ

Optimization Work

GH200/A100 Application Performance/Efficiency

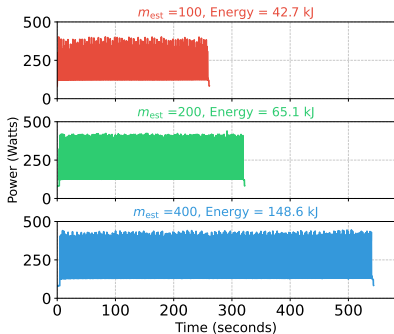
- Test application: **MPTRAC**
- Part of **JUREAP**, JUPITER Research & Early Access Program
- Lagrangian transport of particles
- Analysis of Energy-to-Solution for Grace, Hopper, GH200; A100
- Horizontal: Different GPU clock frequencies

➔ Energy-saving potential

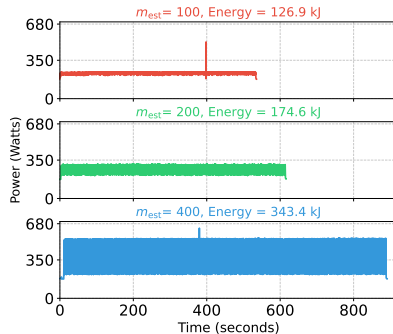


Scaled Block Vecchia Approximation

- Investigation by KAUST, supported by JSC
- New algorithm for approximate modelling of simulations with GPUs
- Extensive tests on JUPITER
- Focus point: GPU energy efficiency
- Preprint on [arXiv \(2504.12004\)](https://arxiv.org/abs/2504.12004)



A100: 2M points



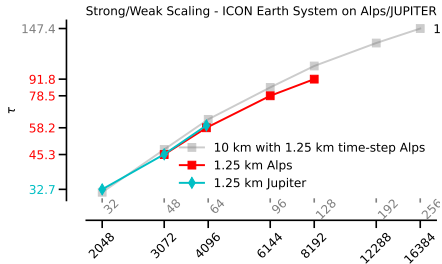
GH200: 5M points

Preliminary data, in publication

JUPITER Heterogeneous Science Highlights

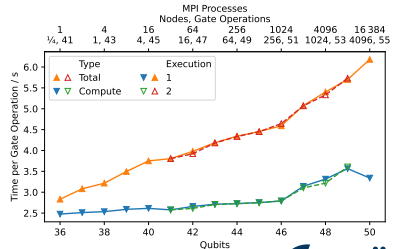
ICON: 1.25 km Simulation (arXiv 2511.02021)

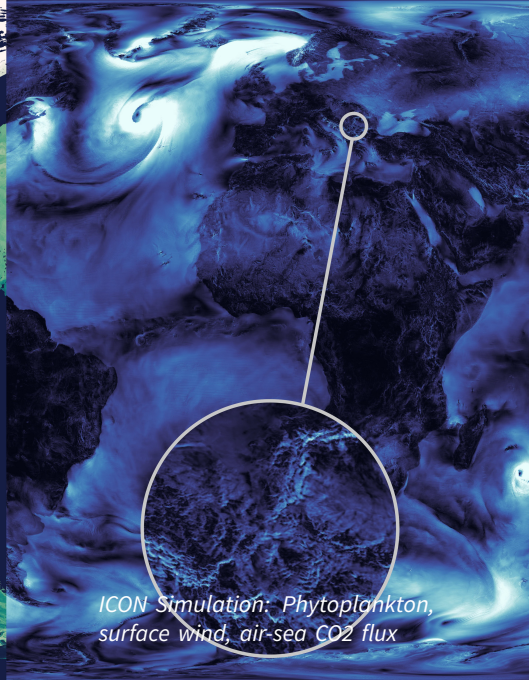
- Gordon Bell Climate Modelling: Computing the Full Earth System at 1 km Resolution
- 20 480 GH200 of JUPITER, using CPU (ocean, land) and GPU (atmosphere)
- Programming: Fortran, OpenACC
- Utilize full superchip



JUQCS: 50 Qubits (arXiv 2511.03359)

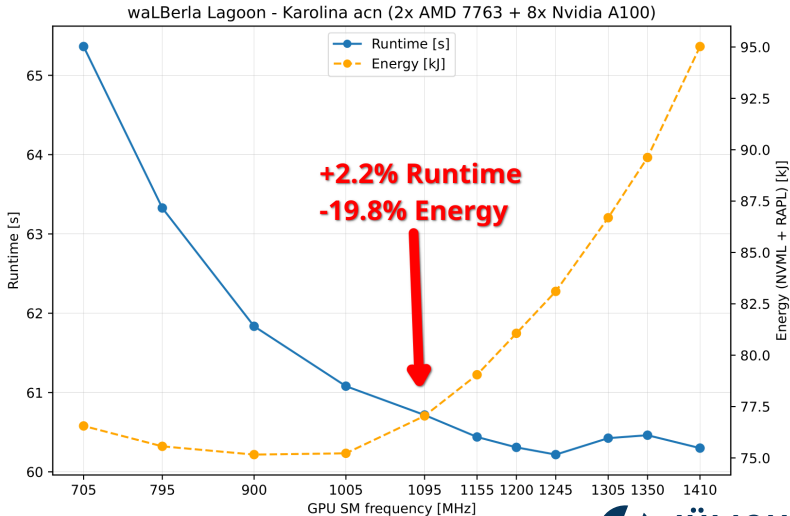
- Simulation of universal quantum computer; record: 50 qubits
- 16 384 GH200 of JUPITER, using GPU for simulation, but CPU+GPU memory at same time (2 PB)
- Programming: Fortran, CUDA Fortran





SCALABLE: waLBerla

- SCALABLE project:
Optimize
Lattice-Boltzmann CFD
application
- Graph: Dynamic
frequency tuning leads
with MERIC leads to
optimal working point



The Hardware Perspective

- New CPUs, GPUs improve performance, also energy efficiency
- Well..., some new GPUs don't improve energy efficiency for FP64 workloads (NVIDIA / GFLOP/(s W): 97 (Hopper) $\longrightarrow$ 72 (Blackwell)) $\Rightarrow$ replace by surrogates and hybrids?
- But new hardware is coming!

The Hardware Perspective

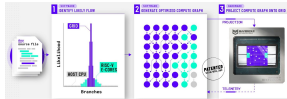
- New CPUs, GPUs improve performance, also energy efficiency
- **Well...**, some new GPUs don't improve energy efficiency for FP64 workloads (NVIDIA / GFLOP/(s W): 97 (Hopper) $\rightarrow$ 72 (Blackwell)) $\Rightarrow$ replace by surrogates and hybrids?
- But new hardware is coming!

RISC-V

- Many different variations of chips
- DARE: EU-funded project for 3 chips (CPU, VEC, AIPU)
- Tenstorrent: Massive array of simple tiny cores

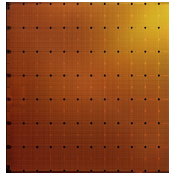
Dataflow

- Data moves through pipeline of operators
- NextSilicon: Maverick 2 chip



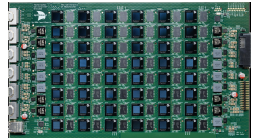
AI Chips

- Specialize to AI demands
- Cerebras: Wafer Scale Engine; 800 000 cores



Neuromorphic

- In-memory / near-memory compute; event-driven compute; ...
- SpiNNaker2: 152 independent cores (neurons/synapses)



Other Venues

- Programming models: Performance portability a driver for efficiency (→ Kokkos)
- Librarization for separation of concerns
- Hardware-software co-design
- Scheduler improvements (also with AI)
- Tight collaboration between operations, support, user
- Continuous, user-exposed monitoring with LLview
- Continuous benchmarking with exaCB
- Exposing tuning knobs to users

```
Kokkos::View<double * [3] view_type  
↪ a("A", N);
```

```
Kokkos::parallel_for("Init", N,  
    KOKKOS_LAMBDA(int i) {  
        a(i, 0) = 1.0 * i;  
    });  
double sum = 0;  
Kokkos::parallel_reduce("Red", N,  
    KOKKOS_LAMBDA(int i, int& lsum) {  
        lsum += i*i;  
    },  
    sum);
```

```
srun --comment="  
    CPU_POWER_CAP=100;  
    CPU_FREQUENCY_MAX=300;  
    NVIDIA_APPLICATIONS_CLOCKS=..."
```

Concluding

Optimize Energy $\approx$ Optimize Performance $\otimes$ Utilize Best-Fitting Hardware

- Modular Supercomputing for heterogeneous HPC
- JUPITER Cluster, Booster
- GH200 superchip
- Optimizing energy
- New hardware
- Other comprehensive, holistic points of attack

Concluding

Optimize Energy $\approx$ Optimize Performance $\otimes$ Utilize Best-Fitting Hardware

- Modular Supercomputing for heterogeneous HPC
- JUPITER Cluster, Booster
- GH200 superchip
- Optimizing energy
- New hardware
- Other comprehensive, holistic points of attack

**Thank you
for your attention!**
a.herten@fz-juelich.de