



JUPITER: STATUS, DEVELOPMENTS, AND EXPERIENCES OF EUROPE'S FIRST EXASCALE SYSTEM

12 March 2025 | Andreas Herten (with support by many) | Forschungszentrum Jülich, Jülich Supercomputing Centre

Content

JUPITER

- System Overview
- Procurement
- Framework

Benchmark Suite

- Benchmarks Overview
- Infrastructure
- Benchmark Details
 - The Grand Table
 - TCO Results
 - High-Scaling Results

First JUPITER Results

- GH200 Memory Comparison
- Hopper Power Analysis
- Conclusion

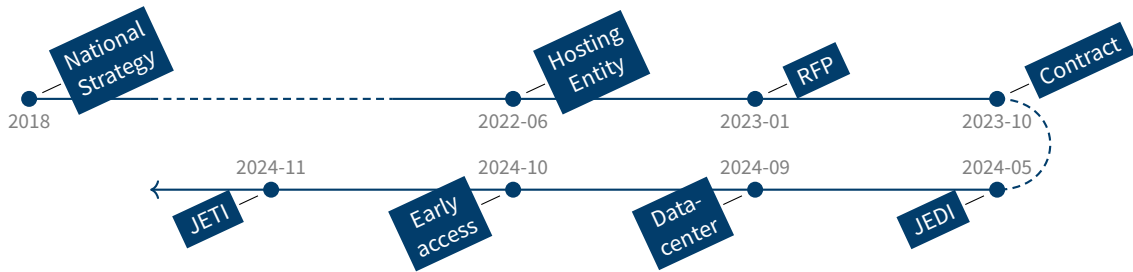
JUPITER

⚡ About JUPITER

- **JUPITER:** First European exascale supercomputer (HPL: 1 EFLOP/s)
 - Procured by EuroHPC JU, BMBF (Federal Ministry of Education and Research), MKW (NRW Ministry of Culture and Science)
 - Hosting entity: Forschungszentrum Jülich / Jülich Supercomputing Centre for Gauß Center for Supercomputing (GCS)

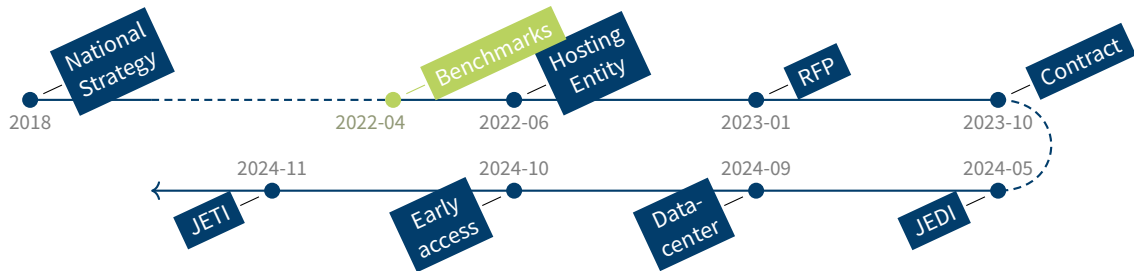
⚡ About JUPITER

- **JUPITER**: First European exascale supercomputer (HPL: 1 EFLOP/s)
 - Procured by EuroHPC JU, BMBF (Federal Ministry of Education and Research), MKW (NRW Ministry of Culture and Science)
 - Hosting entity: Forschungszentrum Jülich / Jülich Supercomputing Centre for Gauß Center for Supercomputing (GCS)



⚡ About JUPITER

- **JUPITER**: First European exascale supercomputer (HPL: 1 EFLOP/s)
 - Procured by EuroHPC JU, BMBF (Federal Ministry of Education and Research), MKW (NRW Ministry of Culture and Science)
 - Hosting entity: Forschungszentrum Jülich / Jülich Supercomputing Centre for Gauß Center for Supercomputing (GCS)



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, > 35 EFLOP/s FP8
≈ 6000 nodes: 4× Grace-Hopper superchip, 4× network
- JUPITER **Cluster**: High versatility, 0.5 B/FLOP balance
≈ 1300 nodes: 2× SiPearl Rhea1 (HBM), 1× network

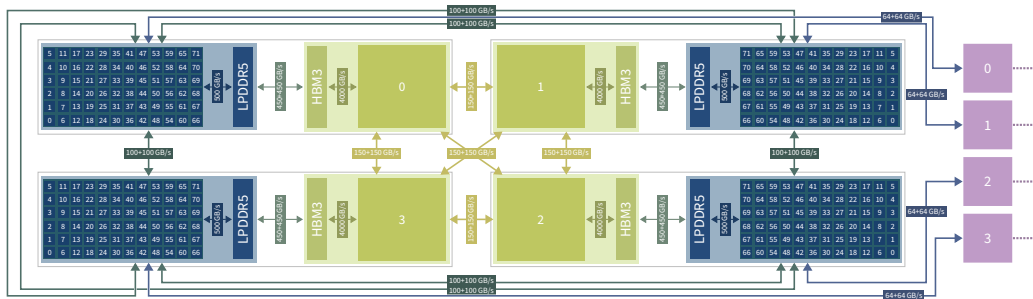


JUPITER System Overview

- ParTec/Eviden consortium
- Implementing Modular Supercomputing Architecture
- JUPITER **Booster**: High scalability, 1 EFLOP/s HPL, > 35 EFLOP/s FP8
 ≈ 6000 nodes: $4 \times$ Grace-Hopper superchip, $4 \times$ network
- JUPITER **Cluster**: High versatility, 0.5 B/FLOP balance
 ≈ 1300 nodes: $2 \times$ SiPearl Rhea1 (HBM), $1 \times$ 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)

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

Delivery of
IT Module 4



Preparations

JEDI: Preparation system

- 48 nodes
($\frac{1}{5}$ DragonFly group)
- May 2024: #1 Green500
(72.7 GFLOP/(s W))



JETI: Staging system

- 480 nodes
(2 DragonFly groups)
- Nov 2024: #18 Top500
(83 PFLOP/s)



JUREAP: Research & Early Access Program

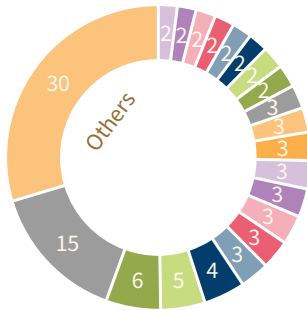
- > 130 participants
- 33 Lighthouse Applications selected (EU+DE)



But how did we get there?

JSC Workload

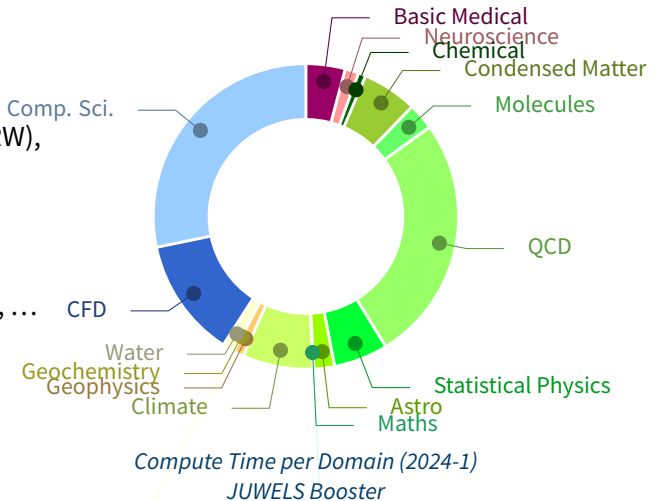
- JSC: HPC resources for Forschungszentrum campus, state (NRW), Germany, Europe
- Compute time through peer-review
- Heterogeneous workload
- Physics, climate, biology, chemistry, AI, ...
- **Goal:** Respect current & anticipated workload in procurements of new systems; incl. domains, methods, programming languages, profiles



*Programs / GPUh (2020)
JUWELS Booster*

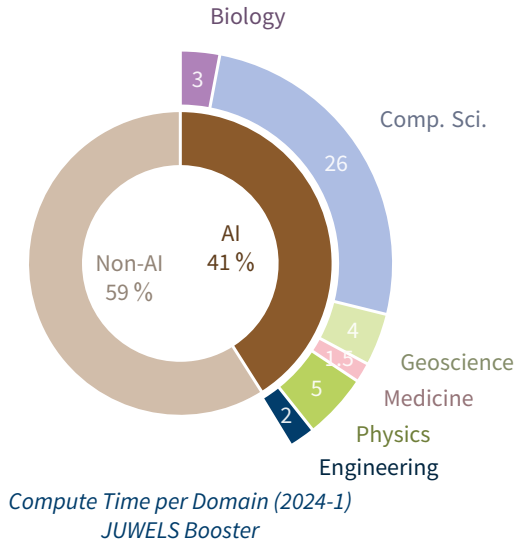
JSC Workload

- JSC: HPC resources for Forschungszentrum campus, state (NRW), Germany, Europe
- Compute time through peer-review
- Heterogeneous workload
- Physics, climate, biology, chemistry, AI, ...
- **Goal:** Respect current & anticipated workload in procurements of new systems; incl. domains, methods, programming languages, profiles



JSC Workload

- JSC: HPC resources for Forschungszentrum campus, state (NRW), Germany, Europe
- Compute time through peer-review
- Heterogeneous workload
- Physics, climate, biology, chemistry, AI, ...
- **Goal:** Respect current & anticipated workload in procurements of new systems; incl. domains, methods, programming languages, profiles



Framework

- Procuring entity: EuroHPC JU
 - Hosting entity: JSC
 - Procurement: JSC, with support from EuroHPC, national ministries
 - Compute time allocations: GCS (Germany), JU (Europe)
 - 500 M€ project budget
- Strong investment from all sides → replicability, reproducibility, reusability R^3
 - Benchmarks
 - **Applications** (Base TCO, High-Scaling, MSA)
 - Synthetic
 - JUPITER Cluster  and Booster 

Framework

- Procuring entity: EuroHPC JU
 - Hosting entity: JSC
 - Procurement: JSC, with support from EuroHPC, national ministries
 - Compute time allocations: GCS (Germany), JU (Europe)
 - 500 M€ project budget
- Strong investment from all sides → replicability, reproducibility, reusability R^3
 - Benchmarks
 - **Applications** (Base TCO, High-Scaling, MSA)
 - Synthetic
 - JUPITER Cluster  and Booster 

Evaluation Target

Mainly: Total Cost of Ownership (TCO)

- Proposals ranked by *workload intensity (how much workload over system lifespan)*
- Also energy consumption respected (simplified)
- Master formula: calculate value for ranking
- Normalized metric/FOM: runtime
- Applications-based

Evaluation Target

Mainly: Total Cost of Ownership (TCO)

- Proposals ranked by *workload intensity (how much workload over system lifespan)*
- Also energy consumption respected (simplified)
- Master formula: calculate value for ranking
- Normalized metric/FOM: runtime
- Applications-based

New: High-Scaling Benchmarks

- For Exascale procurement → respect large-scaleness of system
- Run dedicated workload on entire system
- Applications-based

Evaluation Target

Mainly: Total Cost of Ownership (TCO)

- Proposals ranked by *workload intensity (how much workload over system lifespan)*
- Also energy consumption respected (simplified)
- Master formula: calculate value for ranking
- Normalized metric/FOM: runtime
- Applications-based

New: High-Scaling Benchmarks

- For Exascale procurement → respect large-scaleness of system
- Run dedicated workload on entire system
- Applications-based
- Method:

Full JUWELS Booster Workload



20× workload on full JUPITER Booster

- *Full*: 50 PFLOP/s vs. 1000 PFLOP/s th. peak
- Instructions/rules to determine workload
- Memory variants: (tiny,) small, medium, large



Evaluation Target

Mainly: Total Cost of Ownership (TCO)

- Proposals ranked by *workload intensity (how much workload over system lifespan)*
- Also energy consumption respected (simplified)
- Master formula: calculate value for ranking
- Normalized metric/FOM: runtime
- Applications-based

Also: Synthetic benchmarks

New: High-Scaling Benchmarks

- For Exascale procurement → respect large-scaleness of system
- Run dedicated workload on entire system
- Applications-based
- Method:

Full JUWELS Booster Workload



20× workload on full JUPITER Booster

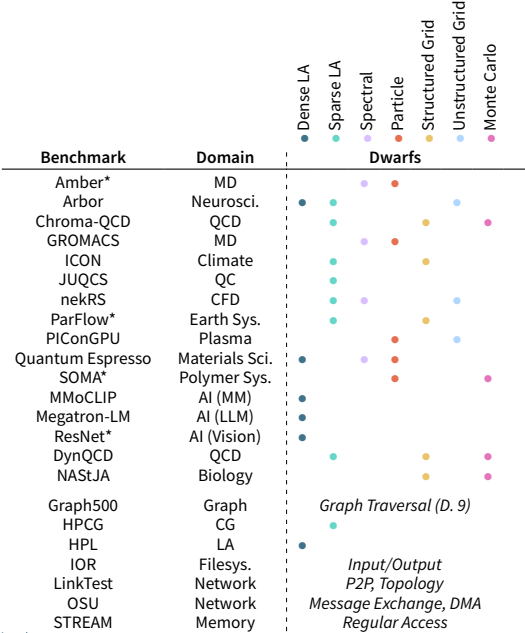
- *Full*: 50 PFLOP/s vs. 1000 PFLOP/s th. peak
- Instructions/rules to determine workload
- Memory variants: (tiny,) small, medium, large



Benchmark Suite

Benchmarks Overview

- 16 application benchmarks (4 de-selected for actual procurement)
Cross-section of domains and methods, 3× AI
- 7 synthetic benchmarks
- Extensive [descriptions](#)
- Right: Patterns of 7 Dwarfs



Infrastructure

- Preparation system: **JUWELS Booster** (73 PFLOP/s peak, 44 PFLOP/s HPL), JURECA-DC (15 PFLOP/s, 9 PFLOP/s)
Dependencies: EasyBuild; versioned environment modules
- **JUBE** workflow environment for every benchmark, similar structure; implicit documentation; platform-independence through inheritance
→ Continuous Benchmarking with *exaCB*
- Extensive **description** (incl. guidelines, rules), similar structure
- **Git**, Git submodules for sources (if possible)
- **Management**: Teams with captains and domain scientists, meetings every 2 weeks, hackathons, scale days, *11 step program*, Gitlab issues

```
- name: systemParameter
  init_with: platform.xml
  parameter:
    - name: preprocess
      _: $modules
    - name: executable
      _: myapp
    - name: args_exec
      _: input.json
    - name: queue
      tag: "baseline|scaling_base|scaling"
      _: booster
    - name: queue
      tag: "exa_tiny|exa_small|exa_medium"
      _: largebooster
```

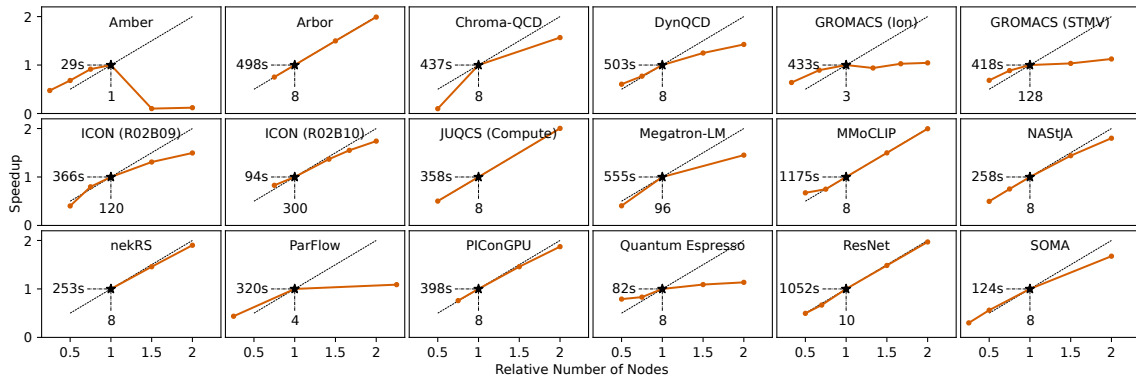
JUBE example

GrandTable Benchmark Details

- Languages, models, libraries
- Licenses
- References nodes Base, High-Scaling
- Memory variants
- Execution targets

	Application Features			Execution Targets		
	Benchmark Name	Progr. Language, [Libraries,]Prog. Models	Licence	Nodes Base	Nodes High-Scale N _{Mem Vars}	Module/Device B _G B _S C _E M _{GC}
Application	Amber*	Fortran, CUDA	Custom	1		✓
	Arbor	C++, CUDA/HIP	BSD-3-Clause	8	642 ^{T,S,M,L}	✓
	Chroma-QCD	C++, QUDA, CUDA/HIP	JLab	8	512 ^{S,M,L}	✓
	GROMACS	C++, CUDA/SYCL	LGPLv2.1	3/128		✓
	ICON	Fortran/C, OpenACC/CUDA/HIP	BSD-3-Clause	120/300		✓
	JUQCS	Fortran, CUDA/OpenMP	None	8	512 ^{S,L}	✓
	nekRS	C++/C, OCCA, CUDA/HIP/SYCL	BSD-3-Clause	8	642 ^{S,M,L}	✓
	ParFlow*	C, Hypre, CUDA/HIP	LGPL	4		✓
	PICongPU	C++, Alpaka, CUDA/HIP	GPLv3+	8	640 ^{S,M,L}	✓
	Quantum Espresso	Fortran, ELPA, OpenACC/CUF	GPL	8		✓
	SOMA*	C, OpenACC	LGPL	8		✓
	MMoCLIP	Python, PyTorch, CUDA/ROCm	MIT	8		✓
	Megatron-LM	Python, PyTorch/Apex, CUDA/ROCm	BSD-3-Clause	96		✓
	ResNet*	Python, TensorFlow, CUDA/ROCm	Apache-2.0	10		✓
	DynQCD	C, OpenMP	None	8		✓
Synthetic	NASTJA	C++, MPI	MPL-2.0	8		✓
	Graph500	C, MPI	MIT	4/16/all		✓
	HPCG	C++, OpenMP, CUDA/HIP	BSD-3-Clause	1/4/all		✓
	HPL	C, BLAS, OpenMP, CUDA/HIP	BSD-4-Clause	1/16/all		✓
	IOR	C, MPI	GPLv2	-/ > 64		✓
	LinkTest	C++, MPI/SIONlib	BSD-4-Clause+	all		✓
	OSU	C, MPI, CUDA	BSD	1/2		✓
	STREAM	C, CUDA/ROCm/OpenACC	Custom	1		✓
						✓
						✓

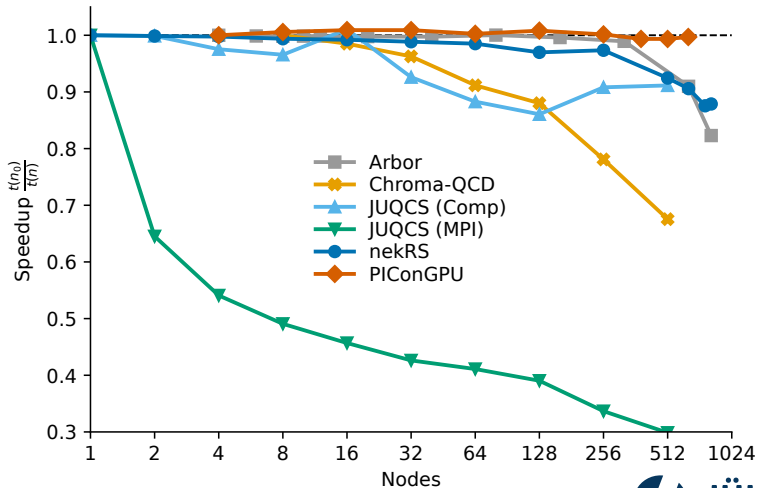
TCO Base Result Grid



Execution on reference number of nodes ($x = 1$) resulting in reference timing ($y = 1$); absolute numbers super-imposed; strong scaling to $0.5 \times - 2 \times$

High-Scaling Results

Weak-scaling
relative to
reference up to
642 nodes
(50 PFLOP/s th.)



First JUPITER Results

GH200 Memory Comparison

- STREAM Triad benchmark

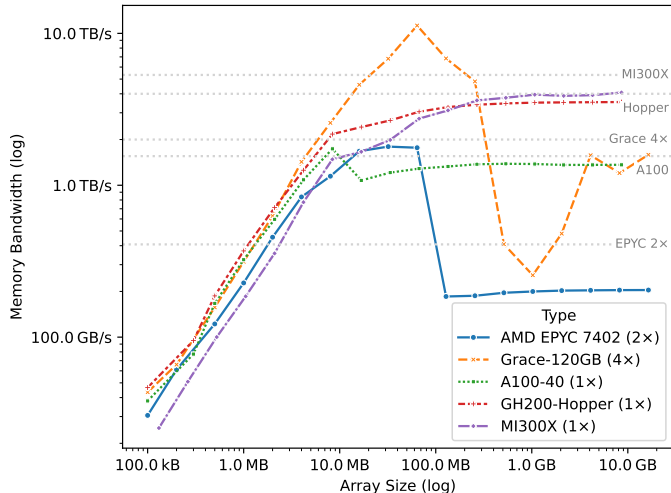
$$\vec{z} = a * \vec{x} + \vec{y}$$

- Horizontal: Scan over memory sizes

- Systems under test:

- JUWELS Booster CPU *AMD EPYC 7402* & GPU *NVIDIA A100-40*
- JUPITER Booster CPU *NVIDIA Grace* & GPU *NVIDIA Hopper*
- AMD Instinct MI300X

→ Great memory bandwidth, also on CPU

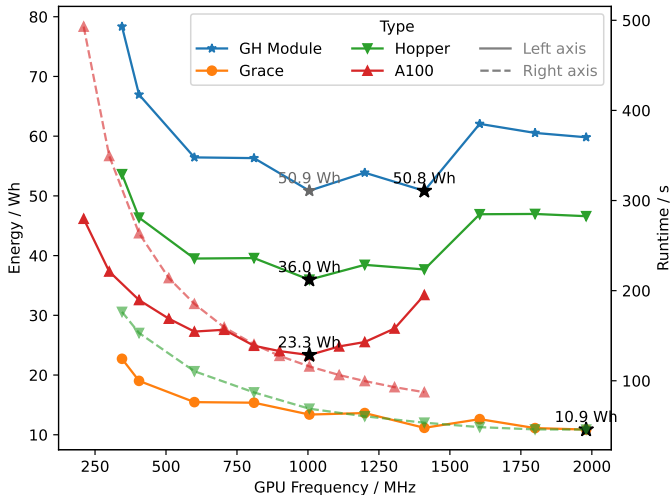


Preliminary data, in publication

Application Performance/Efficiency JUREAP

- JUREAP applications: MPTRAC
- Lagrangian transport of particles
- Analysis of Energy-to-Solution for Grace, Hopper, GH200; A100
- Horizontal: Different GPU clock frequencies

➔ Energy-saving potential



Preliminary data, in publication

Conclusion

Conclusions

- ⚡ JUPITER: First European exascale system (EuroHPC JU, BMBF, MKW; hosted at JSC); currently being built 🧱🧱🧱
- **JUPITER Benchmark Suite:** Benchmark suite for JUPITER procurement *and beyond*
- Application workloads from variety of domains, balanced profiles
- Publication:
 - Paper: [10.1109/SC41406.2024.00038](https://arxiv.org/abs/10.1109/SC41406.2024.00038)
 - Benchmark repos: github.com/FZJ-JSC/jubench
- Extensive effort by whole institute
- Encouraging results from first tests and applications

Conclusions

- ⚡ JUPITER: First European exascale system (EuroHPC JU, BMBF, MKW; hosted at JSC); currently being built 🧱🧱🧱
- **JUPITER Benchmark Suite:** Benchmark suite for JUPITER procurement *and beyond*
- Application workloads from variety of domains, balanced profiles
- Publication:
 - Paper: [10.1109/SC41406.2024.00038](https://arxiv.org/abs/10.1109/SC41406.2024.00038)
 - Benchmark repos: github.com/FZJ-JSC/jubench
- Extensive effort by whole institute
- Encouraging results from first tests and applications

Conclusions

- ⚡ JUPITER: First European exascale system (EuroHPC JU, BMBF, MKW; hosted at JSC); currently being built 🧱🧱🧱
- **JUPITER Benchmark Suite:** Benchmark suite for JUPITER procurement *and beyond*
- Application workloads, balanced profiles
- Publication:
 - Paper: [10.1109/SC46470.2021.9788211](https://doi.org/10.1109/SC46470.2021.9788211)
 - Benchmark repos: github.com/FZJ-JSC/jubench
- Extensive effort by whole institute
- Encouraging results from first tests and applications

Thank you
for your attention!

a.herten@fz-juelich.de

JUPITER

The Arrival of
Exascale in Europe

fz-juelich.de/jupiter | [#exa_jupiter](https://twitter.com/#!/exa_jupiter)



Funding Agencies:



Ministry of Culture and Science
of the State of
North Rhine-Westphalia



JOINING FORCES



fz-juelich.de/jupiter

Appendix

Application Descriptions

TCO

- Amber** Molecular dynamics; STMV use-case; single node
- ParFlow** Hydrology; ClayL use-case
- SOMA** Polymer simulation; Monte-Carlo
- ResNet** AI: Computer vision; TensorFlow
- DynQCD** Particle physics; CPU-only
- GROMACS** Molecular dynamics: GluCL, STMV use-cases; multi-node
- ICON** Climate simulation; atmosphere with R02B09, R02B10; many nodes
- Megatron-LM** AI: LLM; PyTorch
- MMoCLIP** AI: Mixed-Modal (text, image); PyTorch
- Quantum ESPRESSO** Electronic structure; Car-Parrinello test-case
- NASStJA** Biology; CPU-only, MPI-only

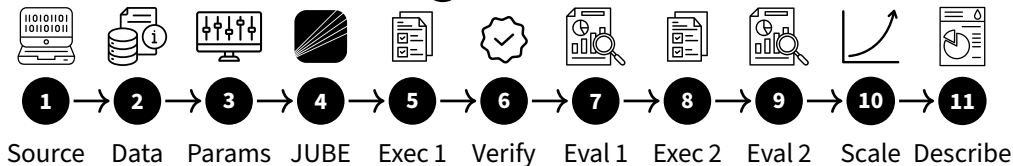
TCO & High-Scaling

- Arbor** Neuroscience; busyring benchmark
- Chroma** Particle physics; hybrid-Monte-Carlo test; QUDA with JIT; max 512 nodes
- JUQCS** Quantum computer simulator; gate-based simulation; communication-heavy; max 512 nodes; Cluster-Booster version (MSA)
- nekRS** Fluid dynamics; Rayleigh-Bénard convection use-case
- PICongPU** Plasma physics; Kelvin-Helmholtz instability use-case

JUPITER Application Benchmarks

- JUPITER: Largest procurement to date
- >18 months of work
- >30 people involved
- 1(-3) associated people (*captains*) per benchmark
- Meetings every two weeks
- Gitlab issue tracker, status tracker (11 points)

1. Source code available
2. Input data available
4. JUBE integration
11. Description, documentation



Lessons Learned

Performance models useful, even if simple

Domain decomposition scripts/rules for unknown system makeup

Intensive feedback for app devs

Verification is hard



Applications

Suite: resource-intensive
→ aim for short turn-around times during dev

Artificially limit benchmarks on prep system to mimic future system

Extensive, balanced execution rules/guidelines



Benchmarks

Multi-system procurement
→ benchmark balance 🤖

Collaboration, tools

Bias towards incremental update

Limiter: time, on *all* sides
→ reuse!



Procurement

Benchmark Suite Availability

- All benchmark workflows, descriptions, data released online
- **Code** GitHub jubench meta-repository, Zenodo meta-record
Individual repos: Arbor Amber Chroma-LQCD GROMACS ICON JUQCS Megatron-LM
MMoCLIP nekRS ParFlow PConGPU Quantum ESPRESSO ResNet SOMA DynQCD
(CPU) NASTJA (GPU) Graph500 HPCG HPL IOR LinkTest OSU Micro-Benchmarks
STREAM STREAM (GPU)
- **Paper** SC24 Proceedings, arXiv:2408.17211
Including extensive SC reproducibility appendix



GitHub



Proceedings