Benchmarking Celeritas

Peter Heywood, Research Software Engineer

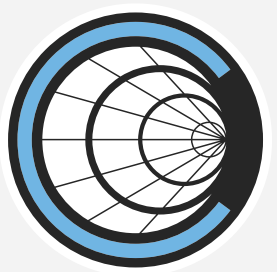
The University of Sheffield

2025-04-11

Celeritas

Celeritas

> The Celeritas project implements HEP detector physics on GPU accelerator hardware with the ultimate goal of supporting the massive computational requirements of the HL-LHC upgrade.



- NVIDIA GPUs via CUDA
- AMD GPUs via HIP
- 2 Geometry implementations
  - ORANGE (CUDA/HIP)
  - VecGeom (CUDA)
- Standalone executables
- Software library

More Information

"Accelerating detector simulations with Celeritas: performance improvements and new capabilities"

- CHEP 2024 presentation
- arXiv:2503.17608

github.com/celeritas-project/celeritas

celeritas-project/regression

a suite of test problems in Celeritas to track whether

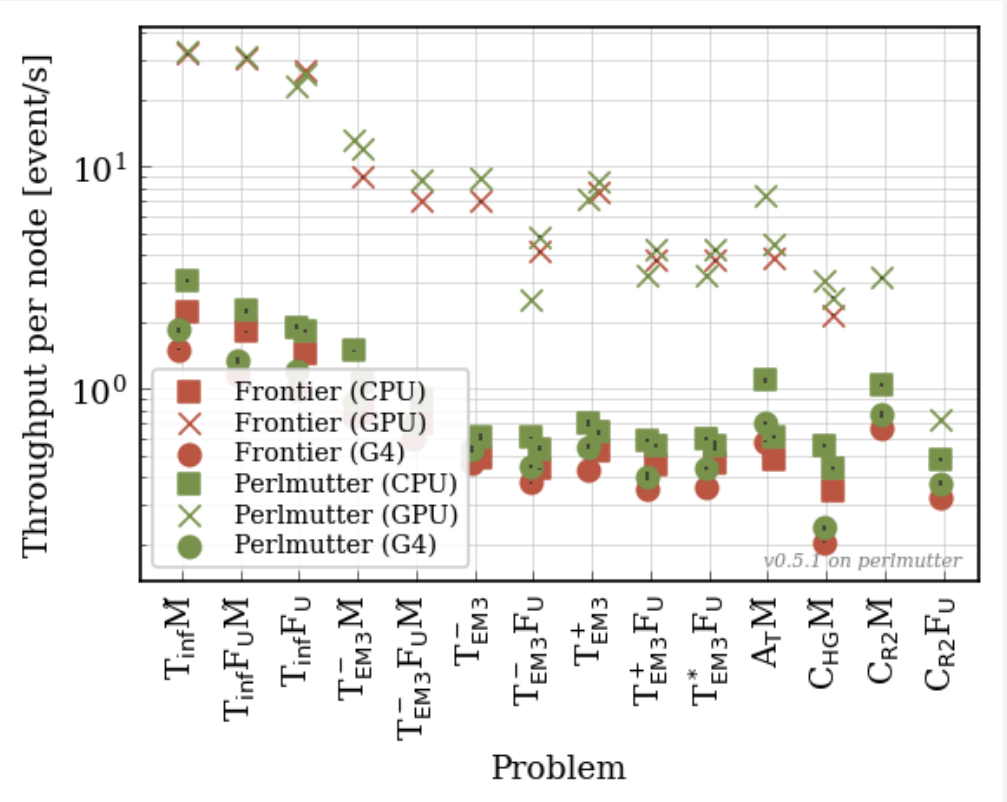
- the code is able to run to completion without hitting an assertion,
- how the code input options (and processed output) change over time,
- how the kernel occupancy requirements change in response to growing code complexity

- CPU and GPU runs
- Standalone: `celer-g4`, `celer-sim`
- Library: `geant4`
- GPU power usage monitoring
- Node-level benchmarking

- ~22 simulation inputs
  - 7 geometries
  - simulation options (msc, field)
  - orange vs vecgeom

github.com/celeritas-project/regression

celeritas-project/regression reference plots



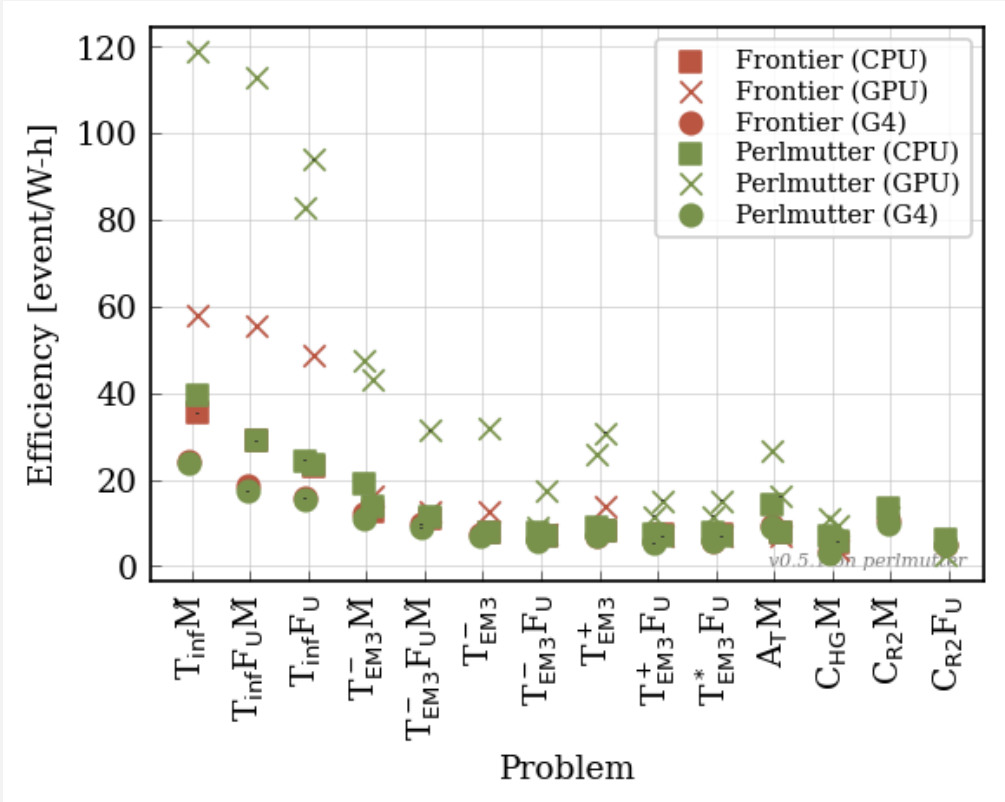


Figure 1: Per-node (a) throughput and (b) efficiency for Celeritas `v0.5.1` on Frontier & Perlmutter.
Generated using `update-plots.py` from commit d5b5c03.
Credit: celeritas-project/regression contributors

Hardware

Hardware

Machine	CPU per node	GPU per node
Frontier	1x AMD “Optimized 3rd Gen EPYC”	8x AMD MI250x
Perlmutter	1x AMD EPYC 7763	4x NVIDIA A100
JADE 2.5	2x AMD EPYC 9534	8x AMD MI300x
Bede GH200	1x NVIDIA Grace	1x NVIDIA GH200 480GB
3090 (TUoS)	1x Intel i7-5930k	2x NVIDIA RTX 3090

JADE 2.5 / JADE@ARC

- Joint Aceademic Data Science Endeavour 2.5
- UK Tier-2 technology pilot resource
  - Funded by EPSRC
  - Hosted by the University of Oxford
- 3 Lenovo ThinkSystem SR685a V3 Nodes
  - 2 AMD EPYC 9534 64-Core CPUs @ 280W
  - 8 AMD MI300X GPUs @ 750W
- Currently in early accesss / beta phase

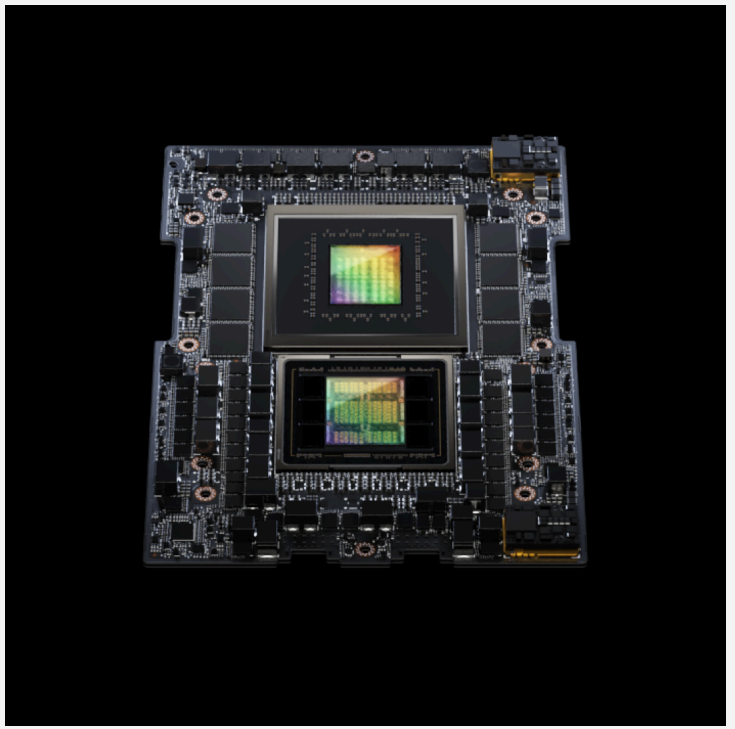
JADE 2.5 Technology Pilot

Source: jade-arc-docs.readthedocs.io

Bede GH200 Pilot

- N8 CIR Bede Grace-Hopper Pilot
- UK Tier-2 HPC resource
  - Originally funded by EPSRC
  - Hosted by Durham University
  - Extended by N8 partners for 1 year
- 6x NVIDIA GH200 480GB nodes
  - 1 NVIDIA Grace 72-core ARM CPU @ 100W
  - 1 96GB Hopper GPU @ 900W
  - NVLink-C2C host-device interconnect

N8 Bede
COMPUTATIONALLY INTENSIVE RESEARCH



NVIDIA Grace Hopper Superchip.

Source: NVIDIA

2x 3090 Workstation

- Headless Workstation @ TUoS RSE
- 1x Intel i7-5930k (6c 12t) @ 140W
- 2x NVIDIA RTX 5090 @ 370W
  - Consumer Ampere `SM_86`
  - Limited FP64 hardware (`1:64`)
  - ~*Equivalent* to A40 / RTX A5500 / RTX A6000

Not an ideal benchmark machine

- Originally built in ~2015 as a HEDT workstation
- GPUs upgraded in 2021
- **Biased towards GPUs**

Benchmarking

Setup `celeritas-project/regression`

```
# Assuming a working install of celeritas
git clone git@github.com:celeritas-project/regression
cd regression
git-lfs install
git-lfs pull
```

run-problems.py

```
class JadeARC(System):
    build_dirs = {
        "orange": Path("/path/to/celeritas/build-ndebug"),
    }
    name = "jadearc"
    num_jobs = 8 # 8 MI300X per node
    gpu_per_job = 1
    cpu_per_job = 16 # 128 core per node
    # ...

# ...

async def main():
  # ...
    _systems = {S.name: S for S in [Frontier, Perlmutter, Wildstyle, JadeARC]}
```

Setup `celeritas-project/regression` cont.

analyze.py

```
CPU_POWER_PER_TASK= {
    "frontier": 225 / 8,
    "perlmutter": 280 / 4,
    "jadearc": 280 / 4,
    "bede": 100 / 1,
    "waimea": 140 / 2,
}
GPU_POWER_PER_TASK = {
  # ...
}
CPU_PER_TASK = {
  # ...
}
TASK_PER_NODE = {
  # ...
}
```

update-plots.py

```
system_color = {
    "frontier": "#BC5544",
    "perlmutter": "#7A954F",
    "jadearc": "#E7298A",
    "bede": "#1B9E77",
    "waimea": "#666666",
}
# ...
def main():
    analyses["frontier"] = plot_minimal("frontier"
    analyses["perlmutter"] = plot_like = plot_all(
    analyses["jadearc"] = plot_minimal("jadearc")
    analyses["bede"] = plot_minimal("bede")
    analyses["waimea"] = plot_minimal("waimea")
    # ...

```

Running celeritas-project/regression

run-jadearc.sh

```
#!/bin/bash -e
#SBATCH -A jade-beta
#SBATCH -p medium
#SBATCH -t 2:59:59
#SBATCH --nodes=1
#SBATCH --ntasks=8
#SBATCH --gres=gpu:8
#SBATCH --cpus-per-gpu=16

# Load modules + activate spack environment
source path/to/jadearc.sh 2> /dev/null

echo "Running on $HOSTNAME at $(date)"
python3 run-problems.py jadearc
echo "Completed at $(date)"
exit 0
```

Versions & Limitations

- Celeritas `v0.5.1`
- CUDA `12.6`
- ROCm `6.2.1`

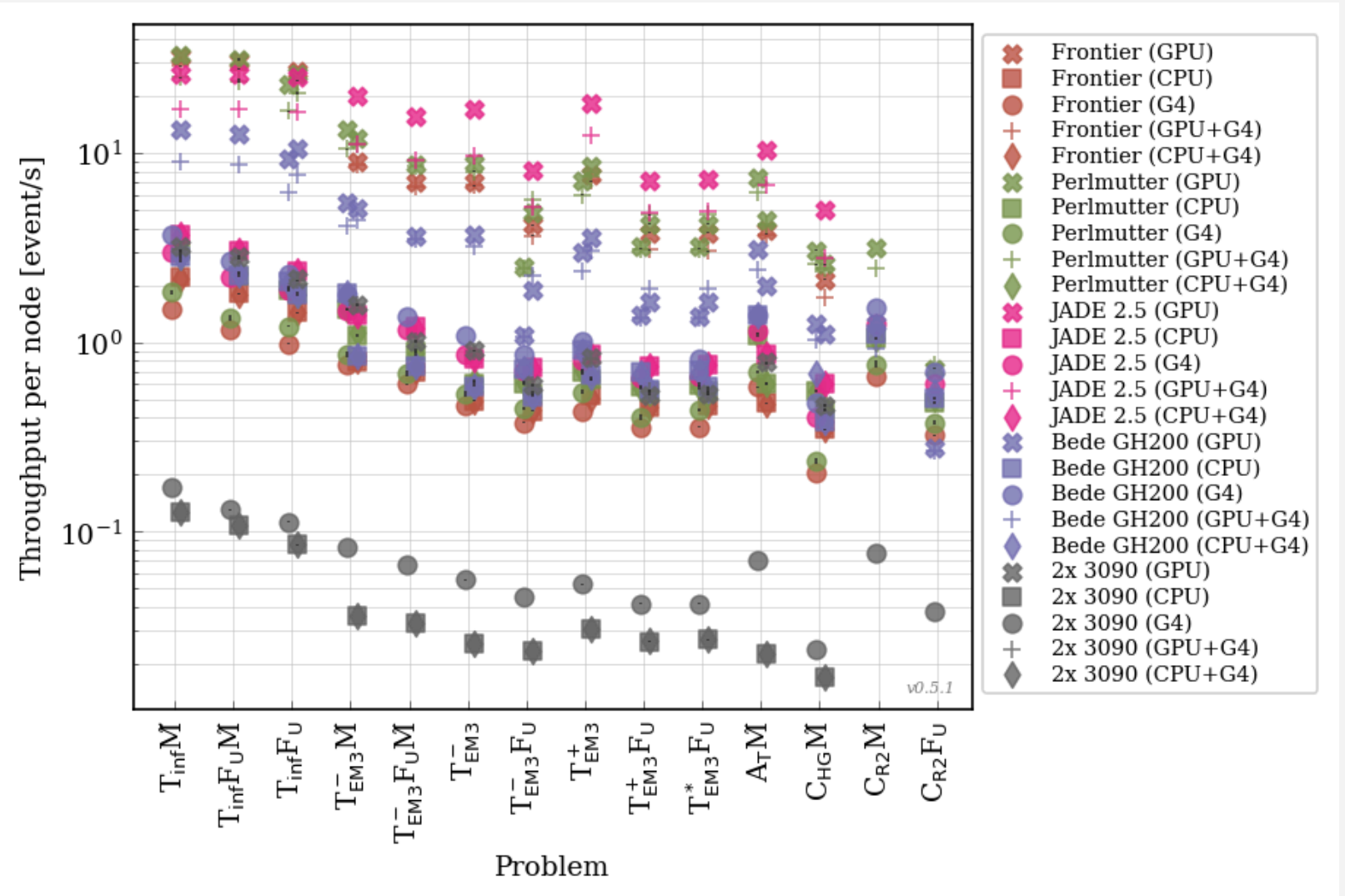
- JADE 2.5
  - Power monitoring not implemented (`amd-smi`)
  - Single-GPU run with 16 CPU cores only
    - Per-node results scaled up x8
  - Manually installed dependencies (spack unhappy)

- 2x 3090
  - No VecGeom results due to Ubuntu/VecGeom link failures
  - Some CPU runs timedout due to old CPU.

- Bede GH200
  - Power monitoring errors
  - G4 GPU offload errors with CUDA OOM
    - 72 CPU cores per GPU is not typical
  - Some G4 failures with OK looking stdout/stderr

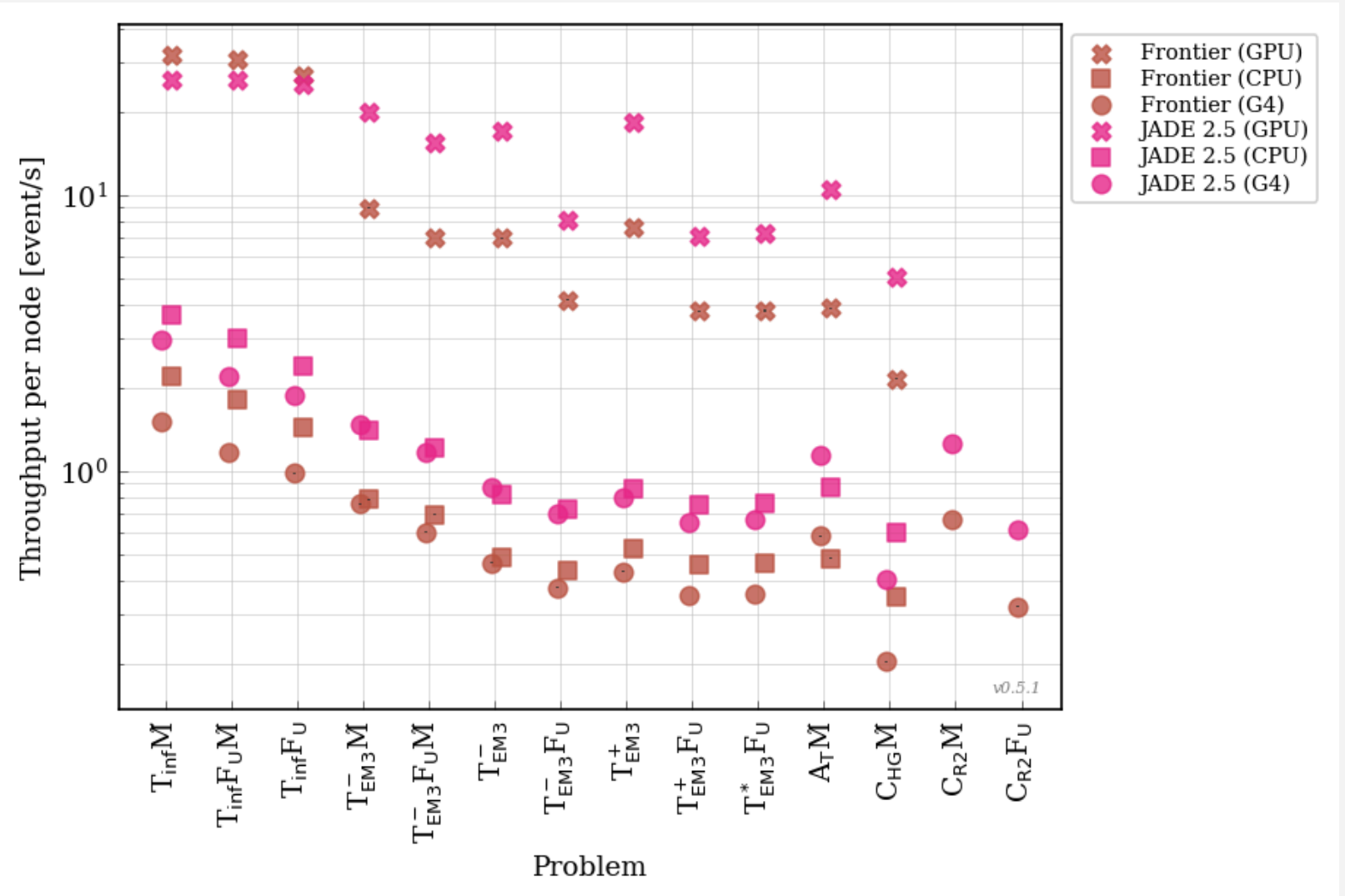
Results

Results: Per-node throughput (all)



All per-node throughput results

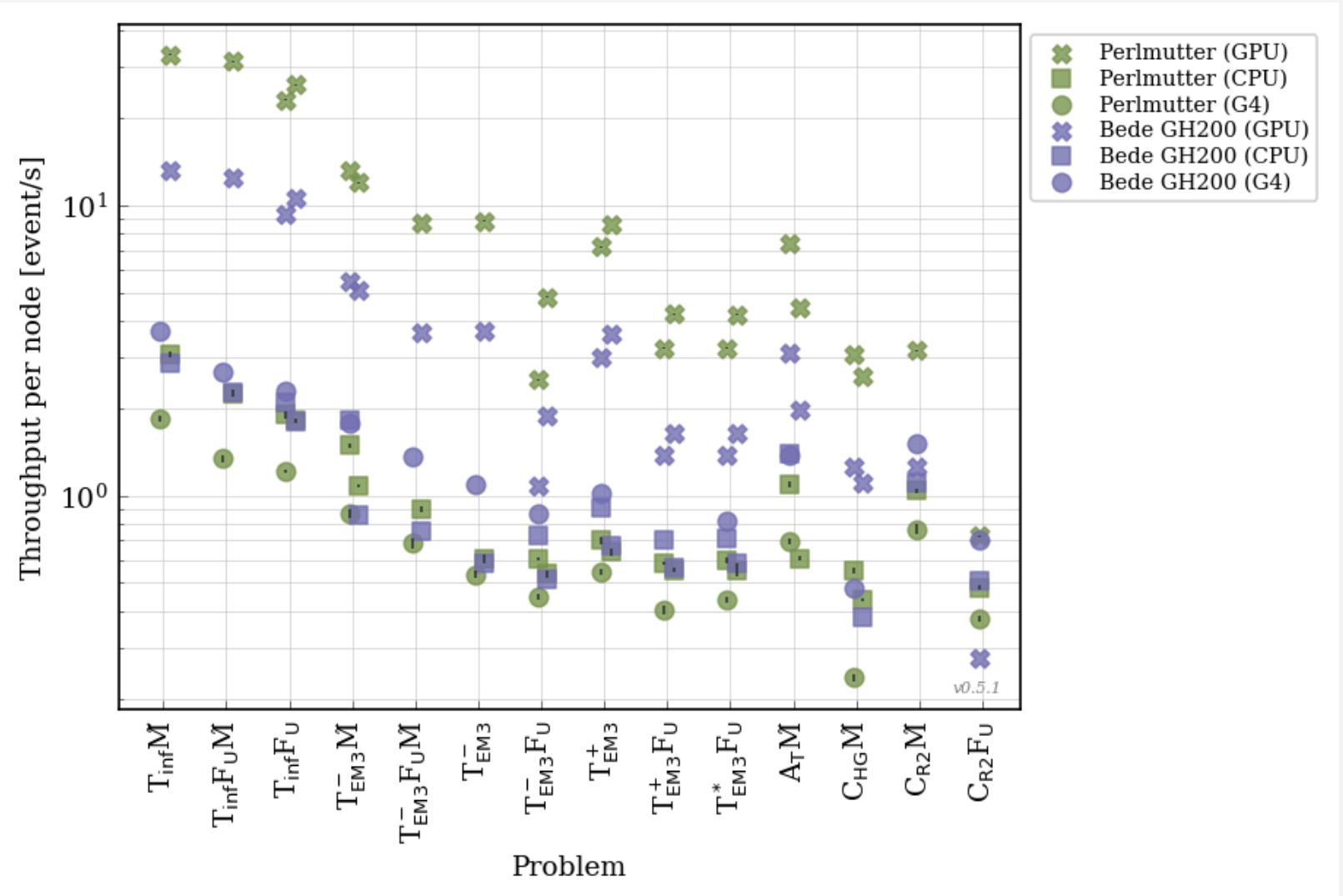
Results: Per-node throughput (AMD)



Per-node throughput for AMD systems (Frontier & JADE 2.5)

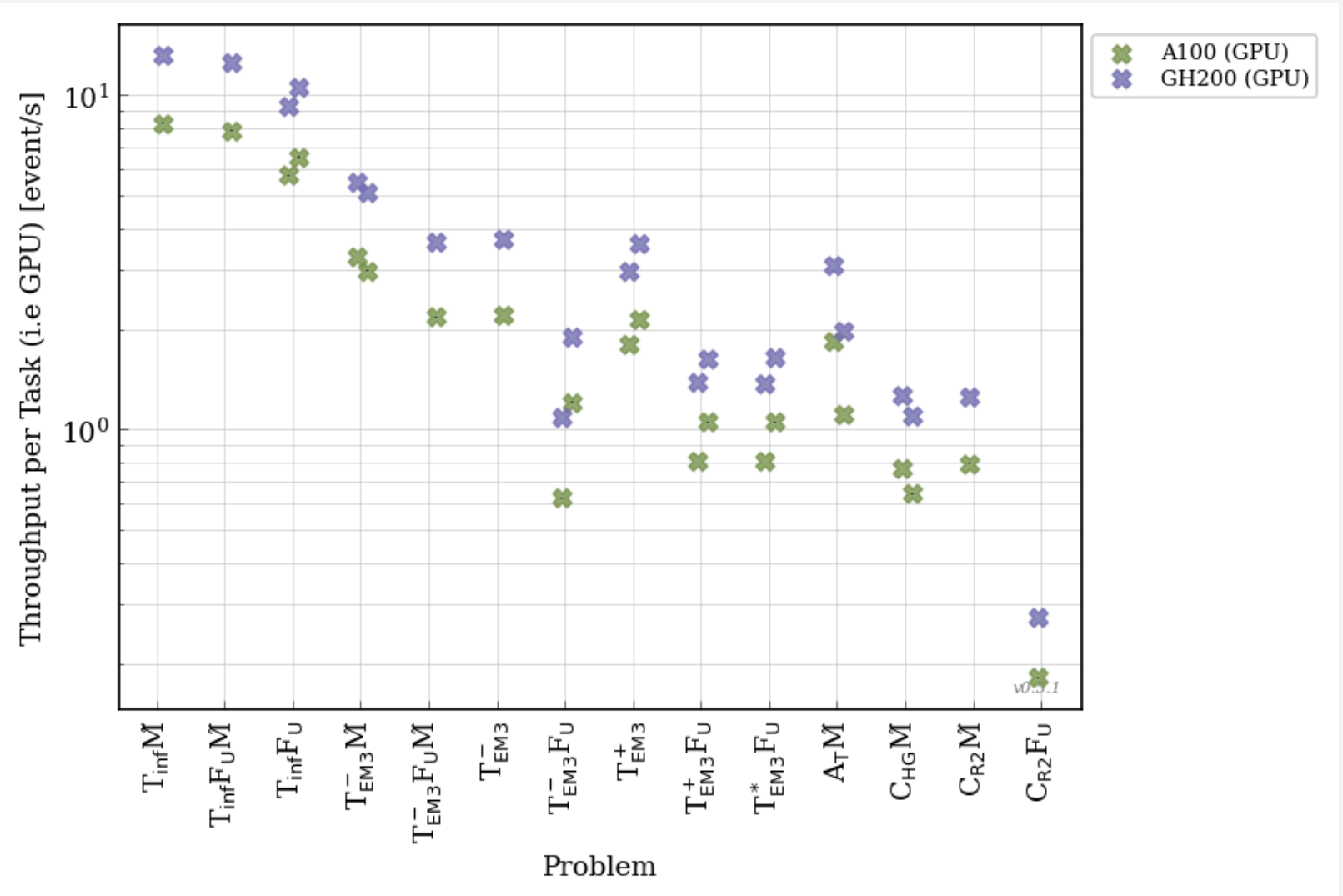
Note: JADE@ARC per-node is estimated from a single GPU run.

Results: Per-node throughput (NVIDIA HPC)



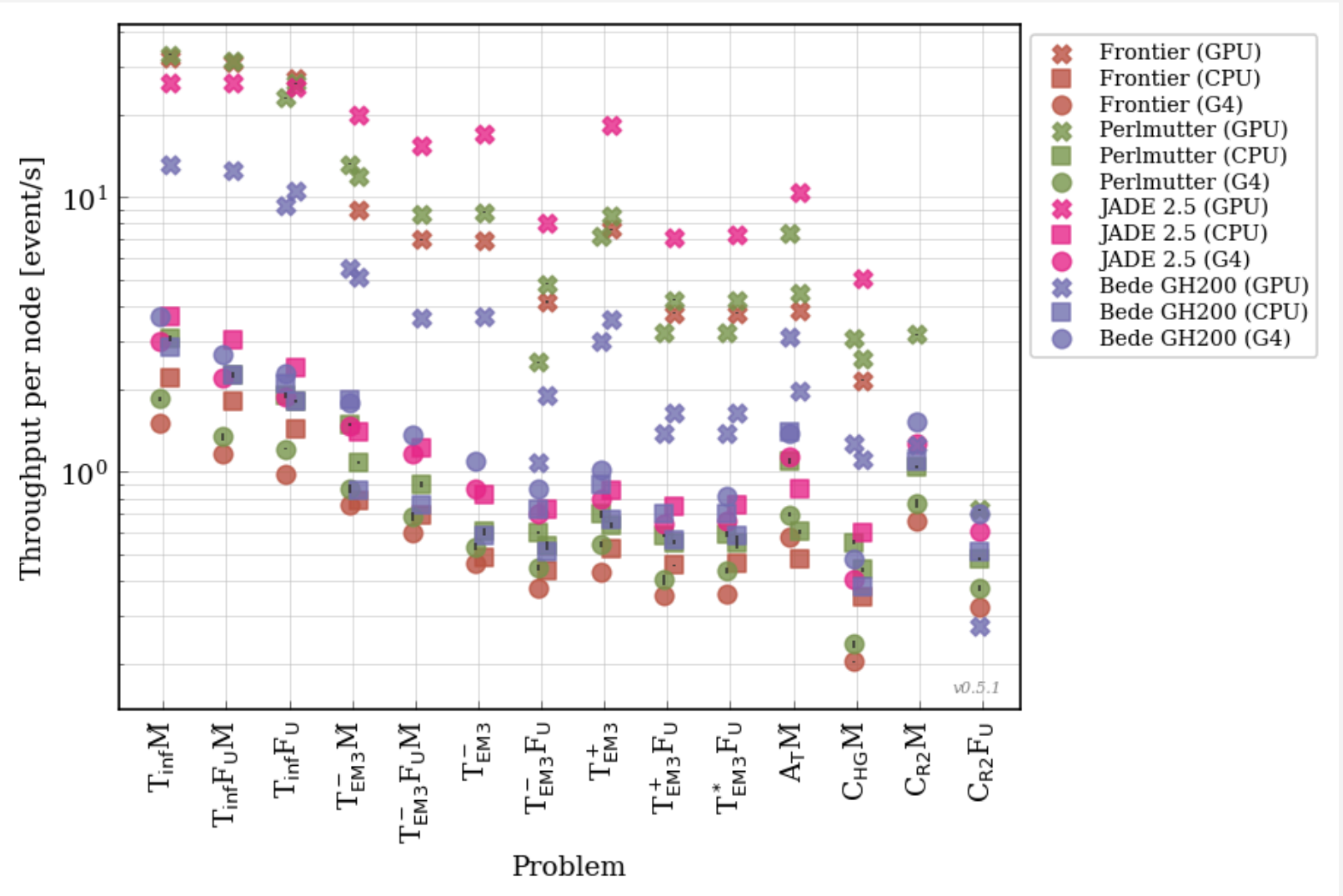
Per-node throughput for NVIDIA HPC systems (Perlmutter & Bede GH200).
GH200 node contains a 72-core CPU & 1 GPU

Results: Per-GPU throughput (NVIDIA HPC)



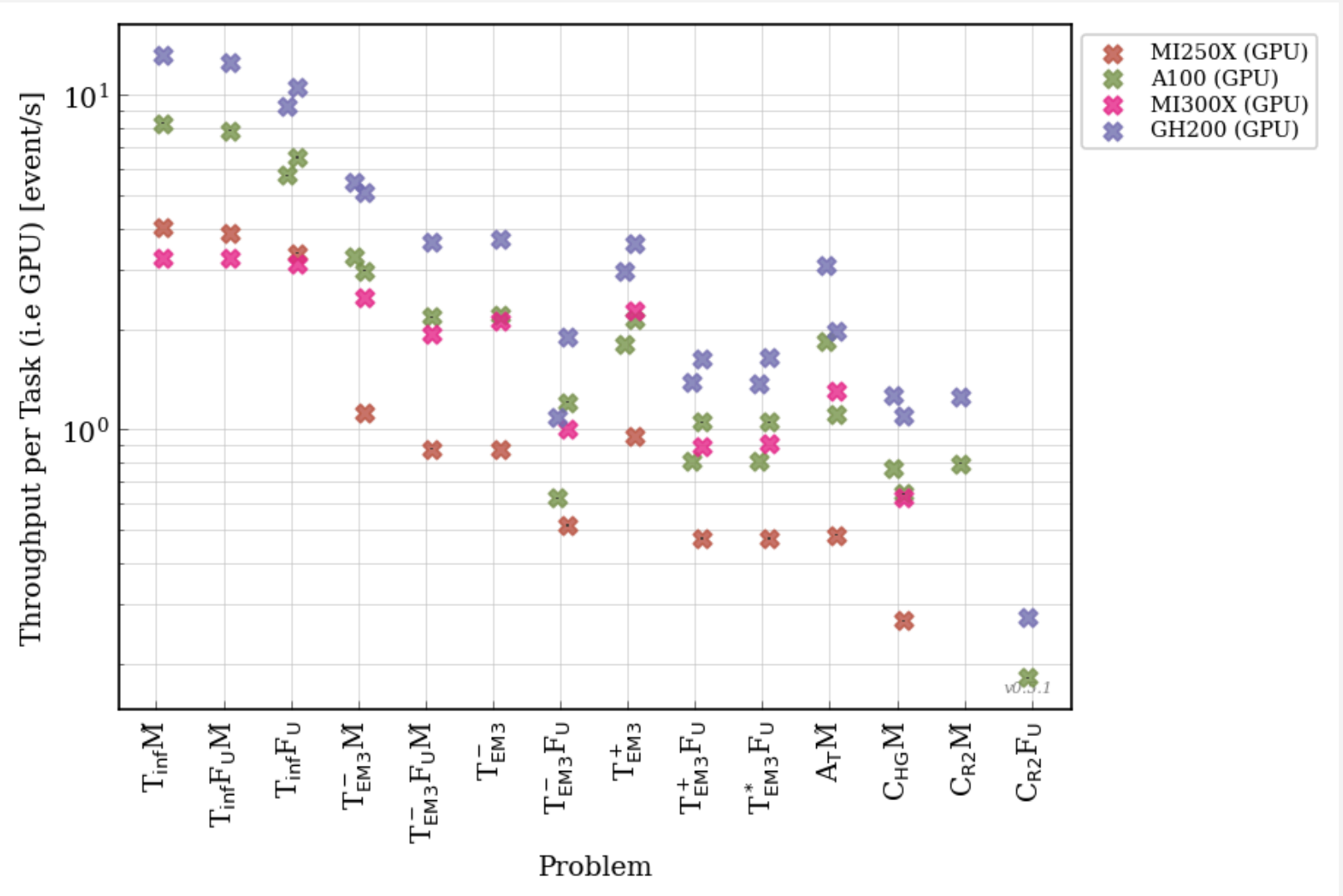
Per-**GPU** throughput for NVIDIA HPC systems (Perlmutter & Bede GH200).
GH200 node contains a 72-core CPU & 1 GPU

Results: Per-node throughput (AMD & NVIDIA HPC)



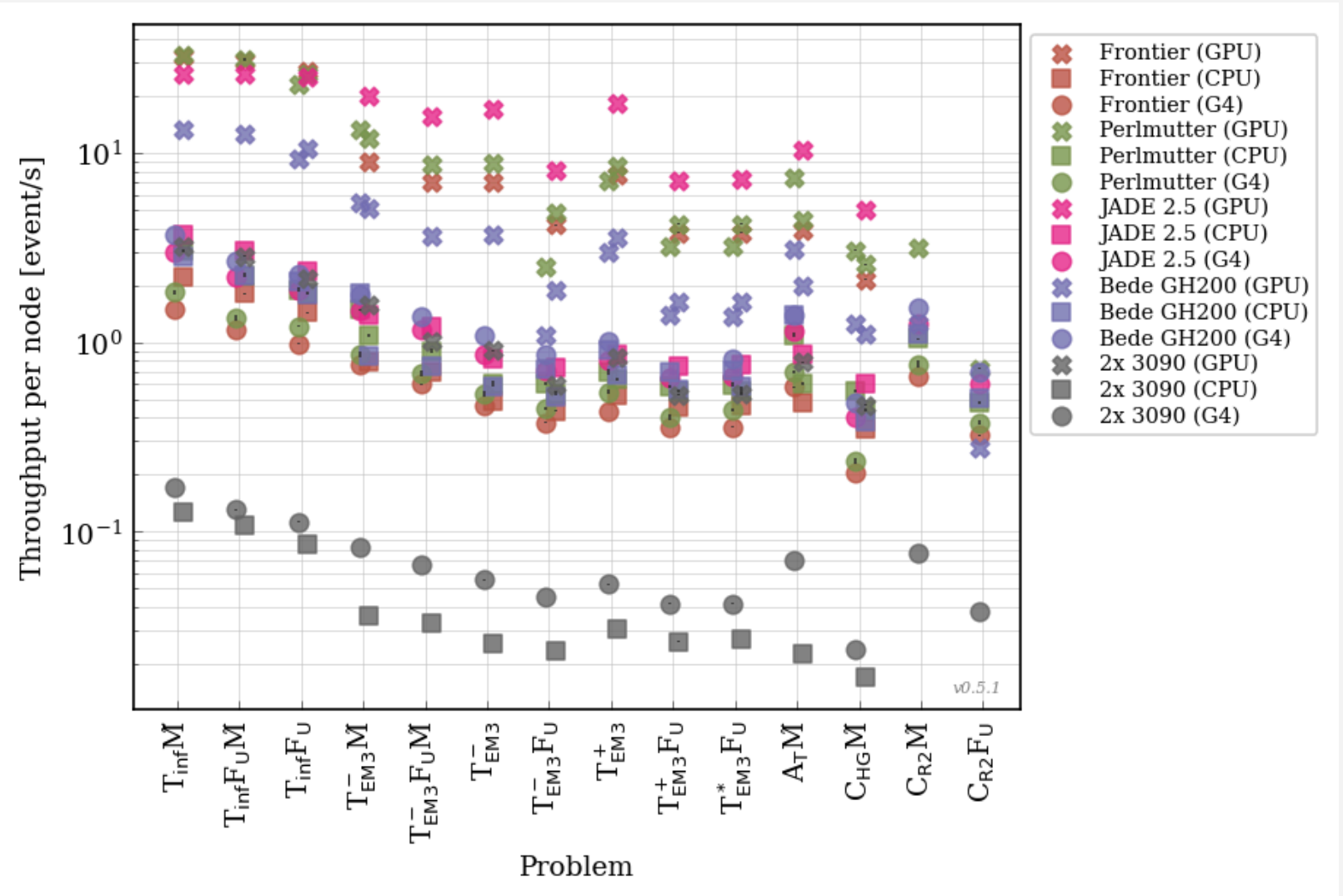
Per-node throughput for AMD & NVIDIA HPC systems (Frontier, Perlmutter, JADE 2.5 & Bede GH200)

Results: Per-GPU throughput (AMD & NVIDIA HPC)



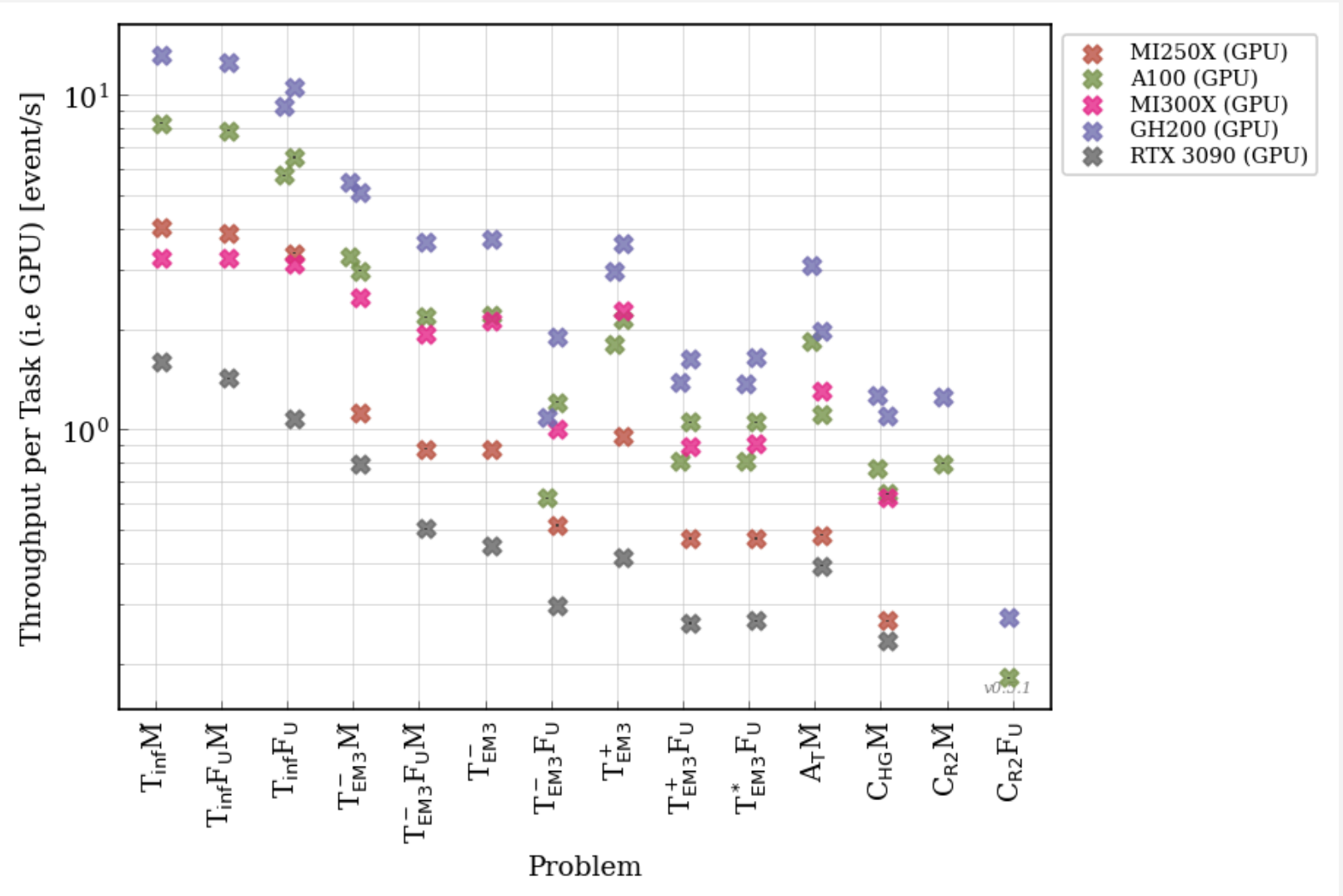
Per-GPU throughput for AMD & NVIDIA HPC systems (Frontier, Perlmutter, JADE 2.5 & Bede GH200)

Results: Per-node throughput (2x 3090)



Throughput per node including the 6-core i7 + 2x 3090 workstation.
This is an unfair comparison

Results: Per-GPU throughput (2x 3090)



Per-GPU throughput for GPU runs only

Todo / What's next

- Fix power monitoring/extraction on Jade and Bede
- Full-node mi300x run to check for thermal throttling
- Investigate failures on Bede/GH200
  - OOM, CPU assertions, silent G4 failures
  - Check / update / pin dependencies
- Tidy up into a branch on my fork of `regression` for future re-runs
- Re-run with Celertias 0.6?
- Try Dual 144GB GH200 node when delivered to Bede

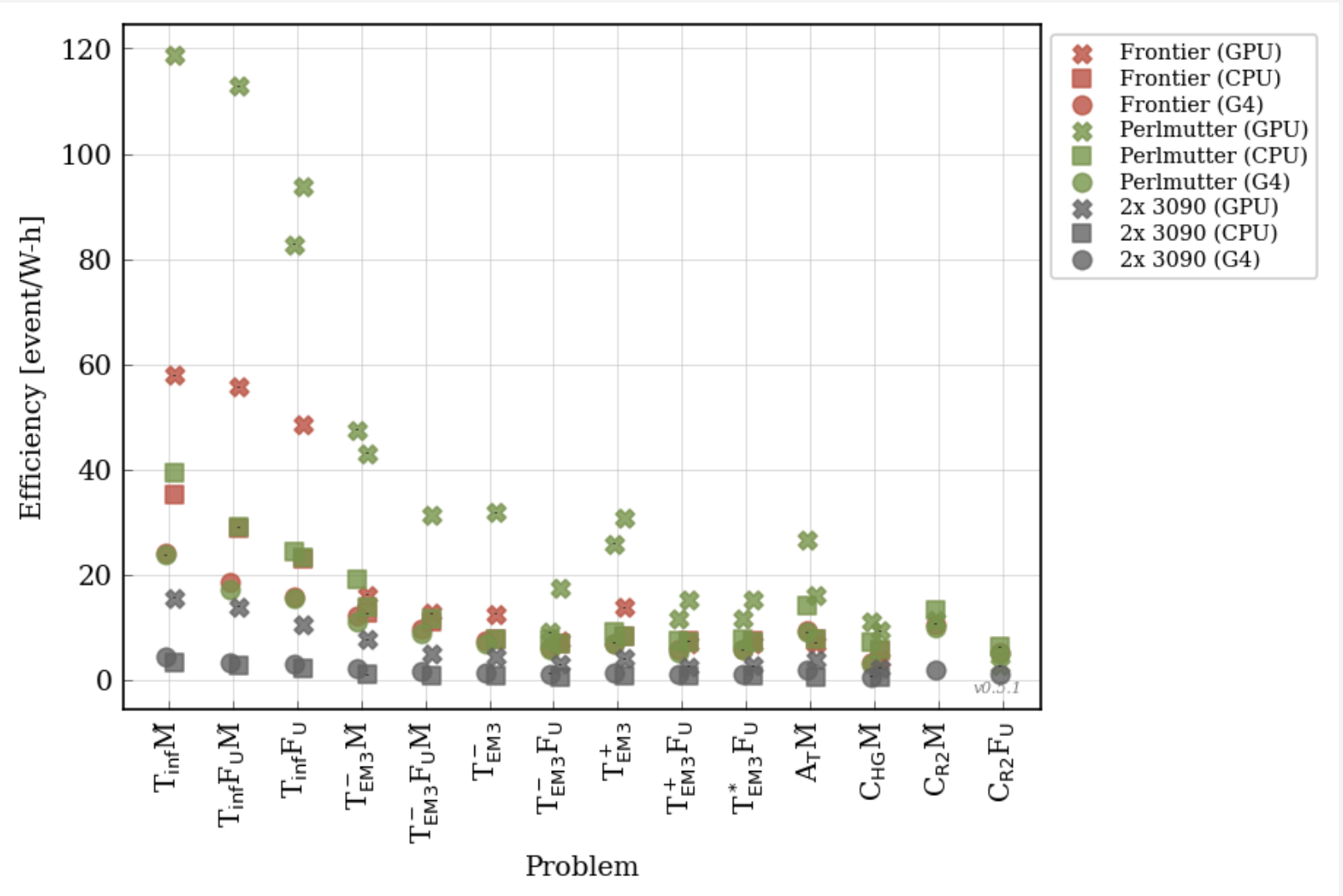
Thank you

Acknowledgements

- `celeritas-project/celeritas` developers
- `celeritas-project/regression` developers
  - Seth R. Johnson @sethrj
  - Julien Esseiva @esseivaju
  - Amanda Lund @amandalund

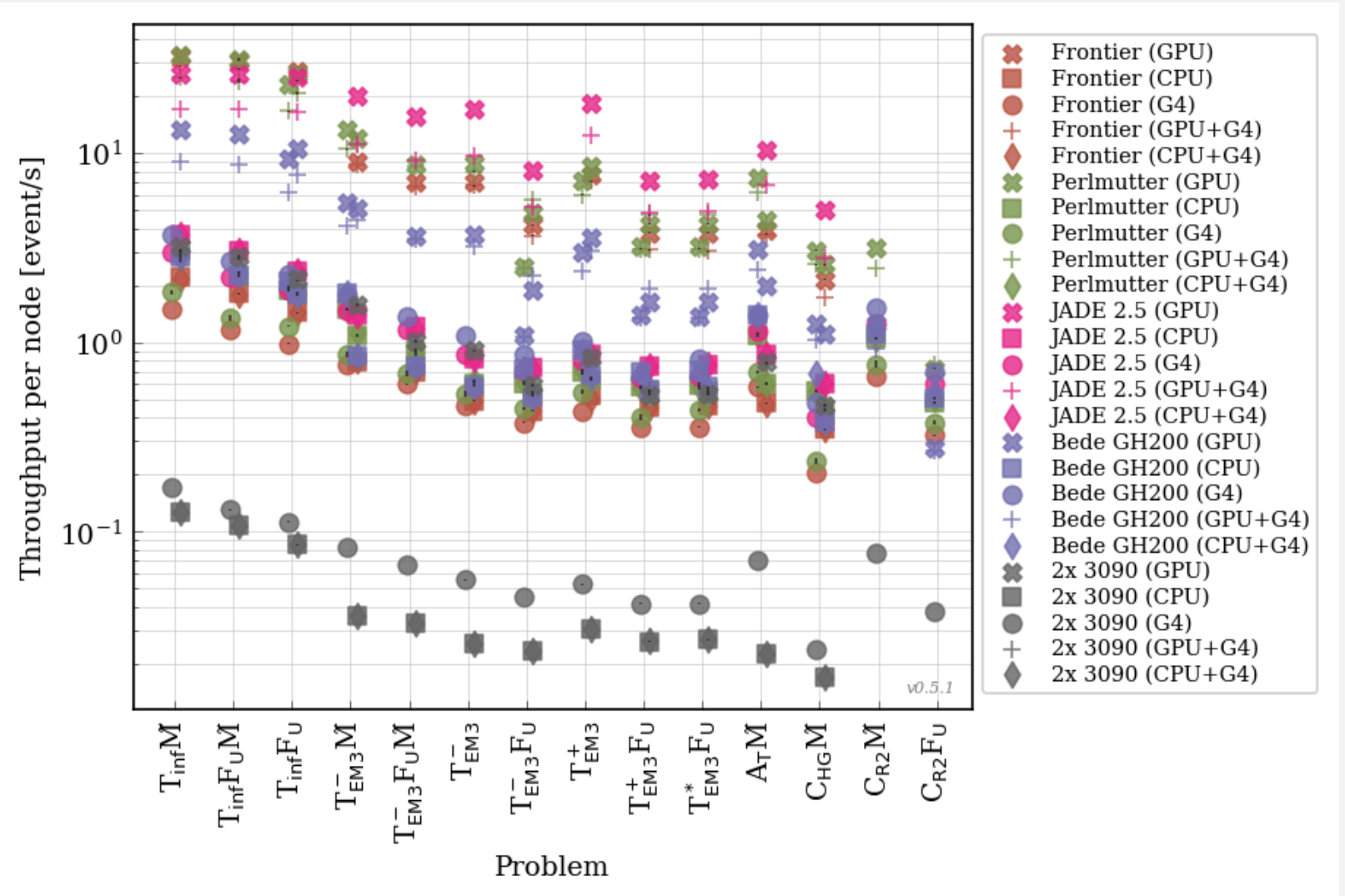
Additional Slides

Energy efficiency (partial)



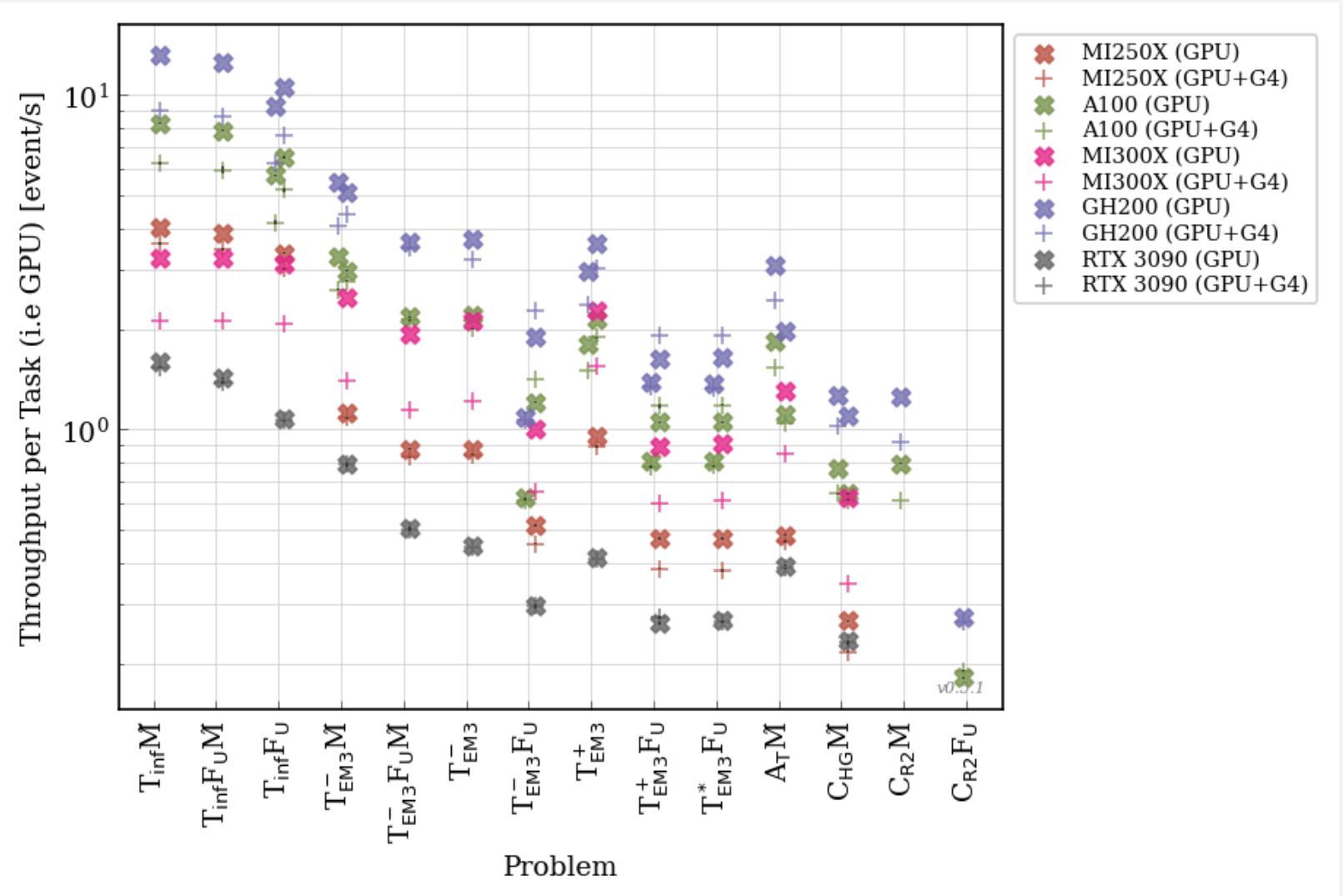
Energy Efficiency for Frontier, Perlmutter and 2x 3090 workstation.
Power consumption extraction not implemented / failed on JADE & Bede

Results: Per-node throughput (all)



All per-node throughput results

Results: Throughput per-GPU w/ G4+GPU



Per-GPU throughput for GPU and GPU+G4