MAELSTROM

Empowering weather & climate forecast:

ML Apps & Datasets

ML Workflow Tools

Hardware Systems

Andreas Herten, Stepan Nassyr
Jülich Supercomputing Centre

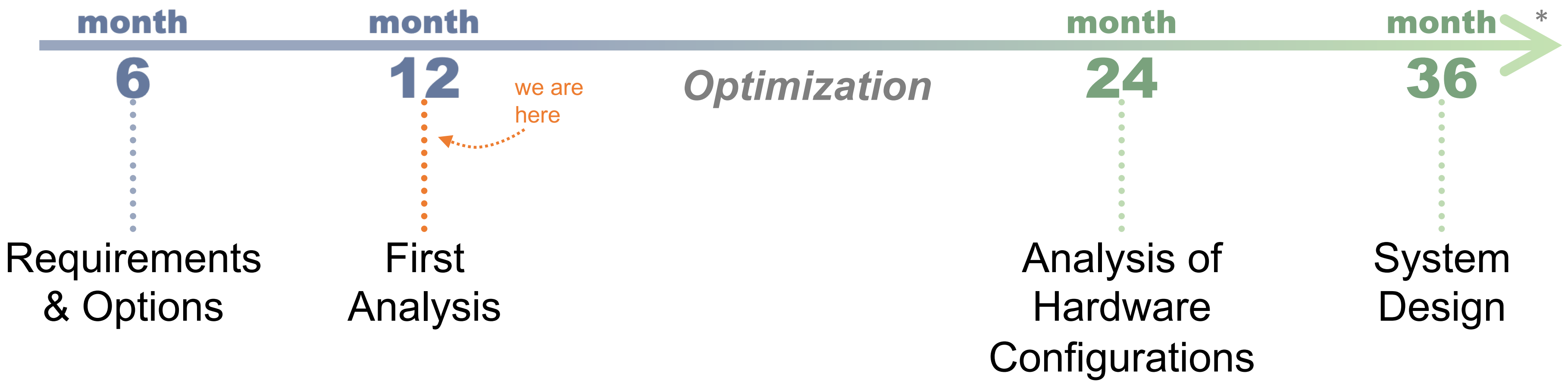
"The MAELSTROM project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955513. The JU receives support from the European Union's Horizon 2020 research and innovation programme and United Kingdom, Germany, Italy, Luxembourg, Switzerland, Norway".

EuroHPC Joint Undertaking

Sponsored by the Federal Ministry of Education and Research

Co-ordinated by ECMWF

About



month 6 — Requirements & Options

month 12 — First Analysis

we are here

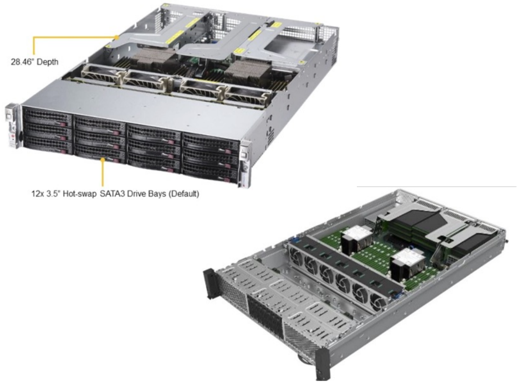
Optimization

month 24 — Analysis of Hardware Configurations

month 36* — System Design

Dedicated versions of MAELSTROM apps benchmarked on:

Jülich system: JUWELS (mostly Booster, but also Cluster)



E4 system: *Lido Adriano* system

Benchmarks run by application owners, guided by WP3 systems staff

Metrics selected in cooperation

Objective: Assess status, identify points of improvement, study hardware

Final goal: Provide bespoke W&C ML system design; fitting W&C ML applications

*: Workpackage timeline not to scale; mileage may vary

Image: Forschungszentrum Jülich (left), E4 (right)

Metrics

Time-related

- Total runtime
- Total training time
- Training time per epoch (avg, min, max)
- Training time per iteration (avg, min, max)
- Training time of first epoch
- Model saving time

Learning-related

- Final loss (training, validation)

Energy-related

- GPU power draw (max)
- Energy consumption (GPU, node)

Image: Metrics by SBTS (up), Energy by Srinivas Agra (down) (both from NounProject.com)

Result Highlights: AP1

JUWELS Booster

- 10 experiments
- 350 s per experiment; ⅓ training, ⅔ data loading, 2% other

JUWELS Cluster

- 3 experiments
- 700 s per experiment, similar distribution

Lido Adriano

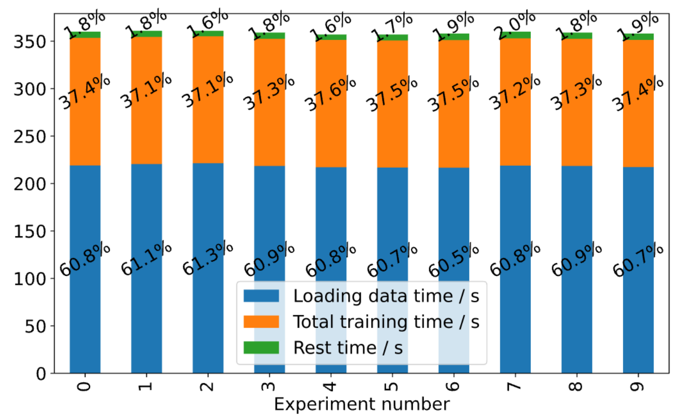
- 5 experiments
- 450 s per experiment, 28% training, 72 % data loading, 1% other

Mostly stable results over various experiments; first epoch always ~30% (JUWELS) / 2× (E4) slower

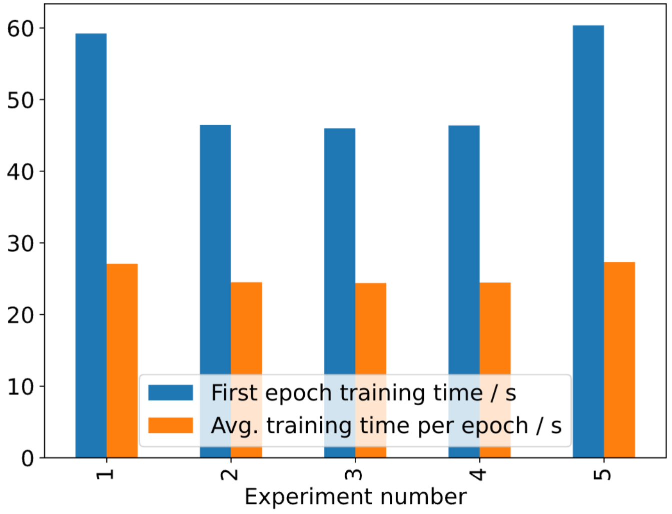
Summary:

- Bound by filesytem, not using GPUs very efficiently
- GPFS > NFS
- E4-A100 slower than JSC-A100

JUWELS Booster: Total Time Split



E4: Epoch Comparison



Result Highlights: AP3

JUWELS Booster

- 440 s runtime; 98% training time; largely stable over 3 repetitions
- Experiments with various configurations: synthetic data; disabled cache in Tensorflow; different GPU number (1 or 2), different batch size (512 or 1024)
    - Disable cache: runtime increase 20%
    - GPU+batch size: runtime decrease 25%
- Energy: 9.25 Wh/GPU (2-1024) vs 12.32 Wh/GPU (1-512)

JUWELS Cluster

- 824 s runtime, ~86 % slower than A100

Lido Adriano

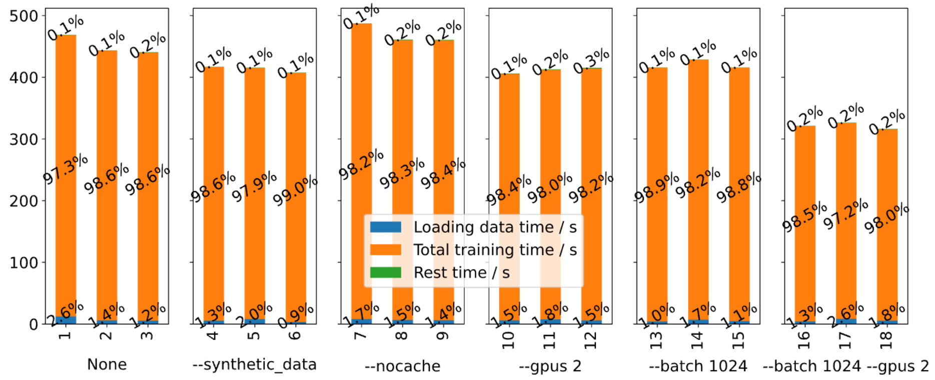
- Slightly faster: 390 s runtime
- Extra experiment: clear filesystem (NFS) cache by rebooting → 2.5 × slower; benefits from streaming data

Additional tests with inference on JUWELS Booster

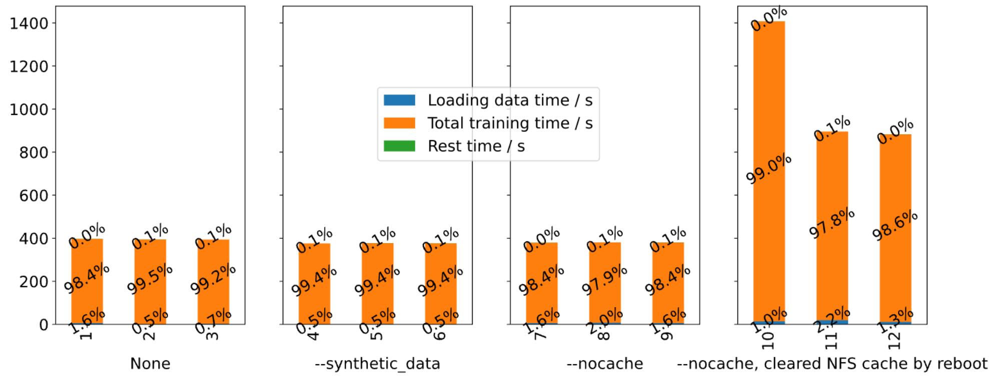
Summary:

- Compute-intensive application (little I/O impact)
- 2-GPU study (benefits from larger batch size)
- Caches used during streaming-in data important

JUWELS Booster: Experiment Overview



E4: Epoch Overview, NFS Cache



Result Highlights: AP4

JUWELS Booster

- 6400 s runtime; 70% training time, significant unaccounted time
- 150 Wh energy consumed, GPU max draw 400 W

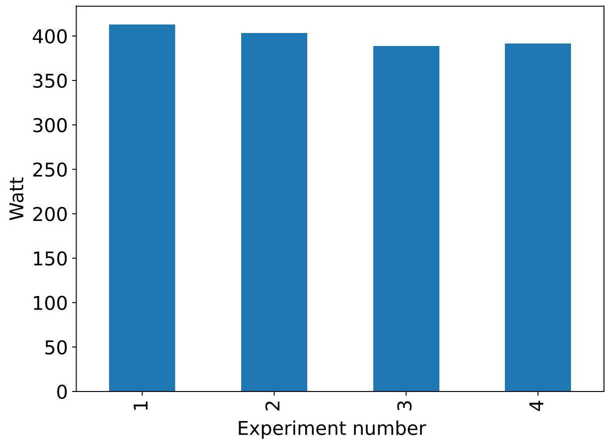
Lido Adriano

- Batch size: 2 (JUWELS Booster: 1)
- 25 729 s runtime; 88 % training time

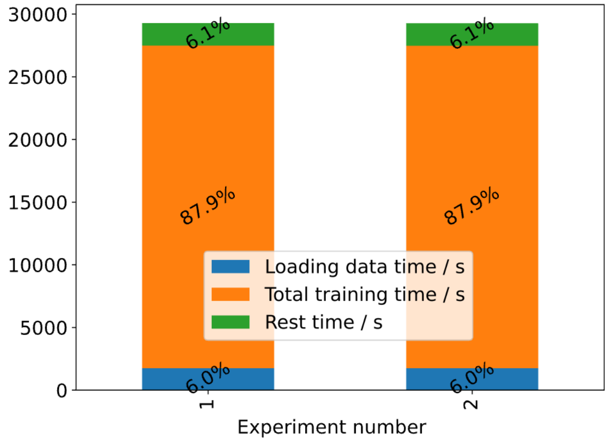
Summary:

- Long runtime for easy statistical measurements
- Good GPU usage
- Investigate run 1 outlier

JUWELS Booster: Max Power Draw



E4: Epoch Runtime Overview



Result Highlights: AP5

JUWELS Booster

- Small data set: 75 s runtime, 92 % training
- Large data set: 1500 s runtime, 98% training
- First epoch 1.75 × (large) / 20 × (small) slower
- 300 W max, 45 Wh consumed

JUWELS Cluster

- Large data set: 2700 s runtime
- 300 W max, 190 Wh

Lido Adriano

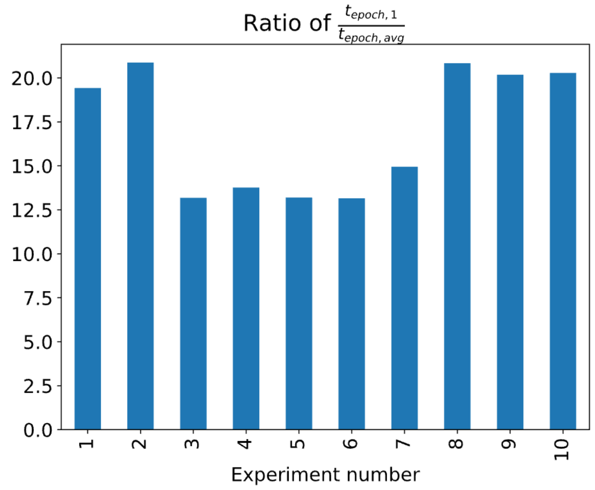
- Various experiments
- Large data set: 1600 s runtime, 94 % training

Additional tests with inference on JUWELS Booster

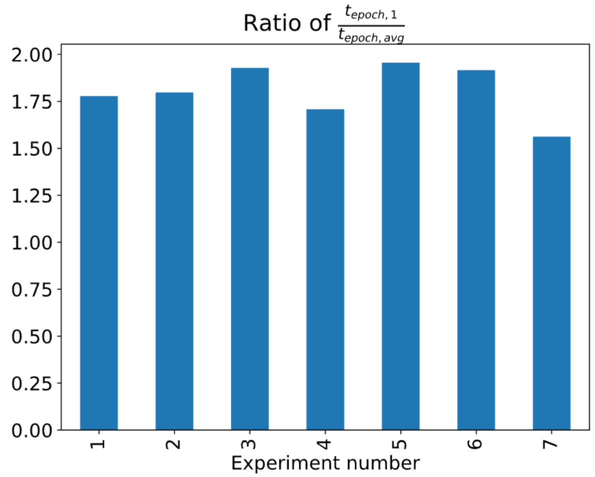
Summary:

- Small data set: Too short runtimes with curious behaviors
- Faster GPU, less energy

JUWELS Booster: Epoch Time Ratio (Small)



JUWELS Booster: Epoch Time Ratio (Large)



Conclusion

- Examples shown of selected MAELSTROM application benchmarks
- Applications × Configurations × Hardware = Many data points
- Investigation ongoing, already many specific (and interesting!) features identified
- Also spotted curiosities for further investigation

Much more data and results then presented here!

→ ***See maelstrom-eurohpc.eu website for D3.4, soon***

Thank you
for your attention!
a.herten@fz-juelich.de // @AndiH
s.nassyr@fz-juelich.de

MAELSTROM

Questions?

EuroHPC
Joint Undertaking

"The MAELSTROM project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955513. The JU receives support from the European Union's Horizon 2020 research and innovation programme and United Kingdom, Germany, Italy, Luxembourg, Switzerland, Norway".

SPONSORED BY THE
Federal Ministry
of Education
and Research

Co-ordinated by
ECMWF