

BEST PRACTICES FOR TRAINING LARGE LANGUAGE MODELS ON HPC SYSTEMS

Carolín Penke · Chelsea Maria John · Jan Ebert · Stefan Kesselheim · Andreas Herten

Large-scale LLM training is a full-stack engineering problem.
 Reliable performance requires addressing software, hardware, data, and operations.

- 1

Build on existing work. Keep up with new developments but stay focused.
- 2

Acknowledge the engineering effort. This is a software engineering endeavor that demands careful system integration and infrastructure development.
- 3

Understand your HPC system. Get to know your hardware and software setup, collaborate with sys-admins, run benchmarks, monitor your runs.
- 4

Document and communicate. Clear documentation, reproducible setups, and shared knowledge bases are essential.

Evaluating 3D Parallelism

Training parallelization

along 3+ dimensions:

- Data parallelism (DP)
- Pipeline parallelism (PP)
- Tensor parallelism (TP)

#GPUs = DP × PP × TP

Heuristic: Favor DP when the whole model fits into one GPU's memory; increase PP and/or TP to make it fit.

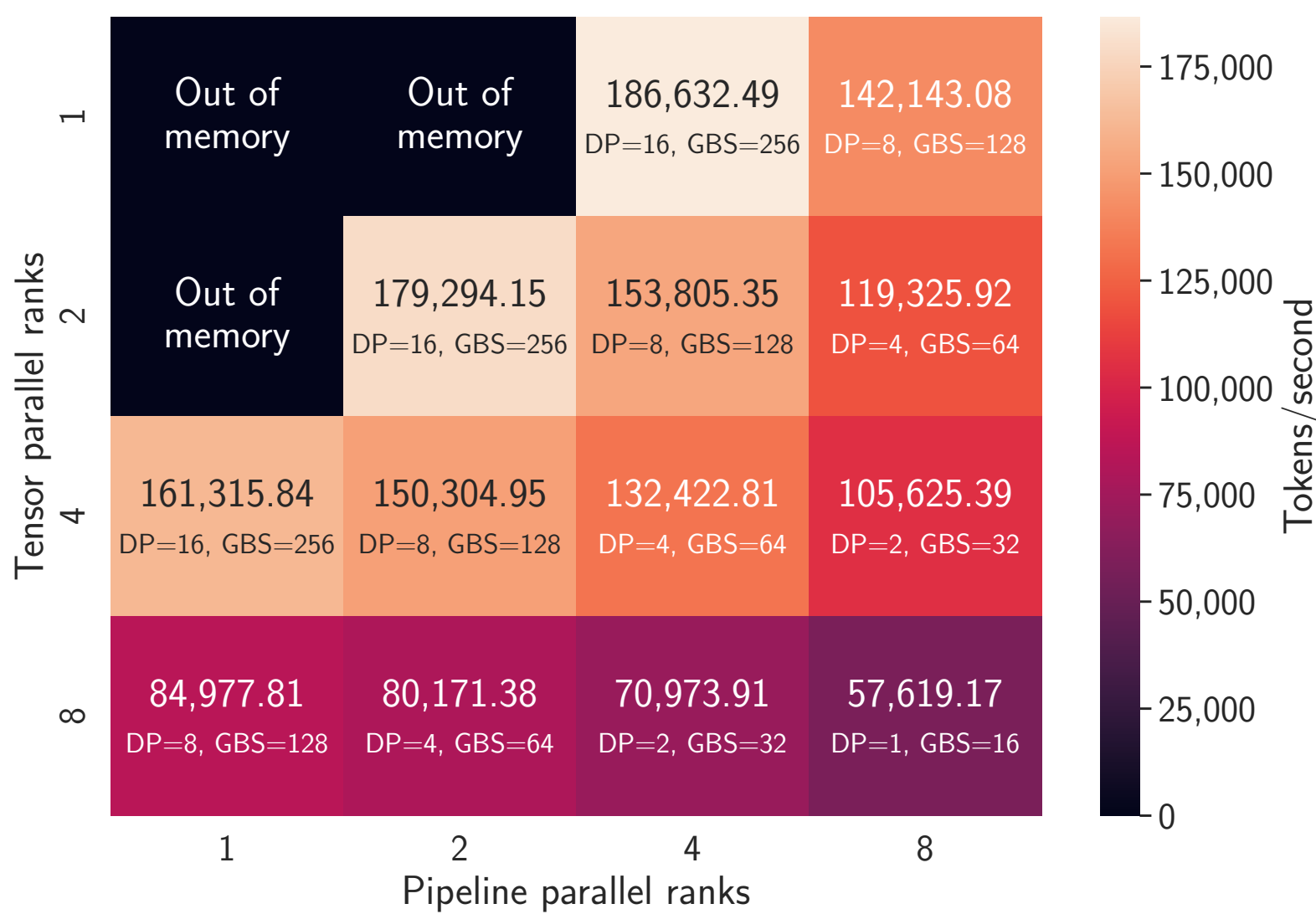


Figure 1: Throughput when training a 6.6B transformer model on 16 × 4 A100 GPUs (40 GB HBM2e) on JURECA. TP and PP were varied; DP = #GPUs/TP/PP.

Impact of Features on Performance and Memory

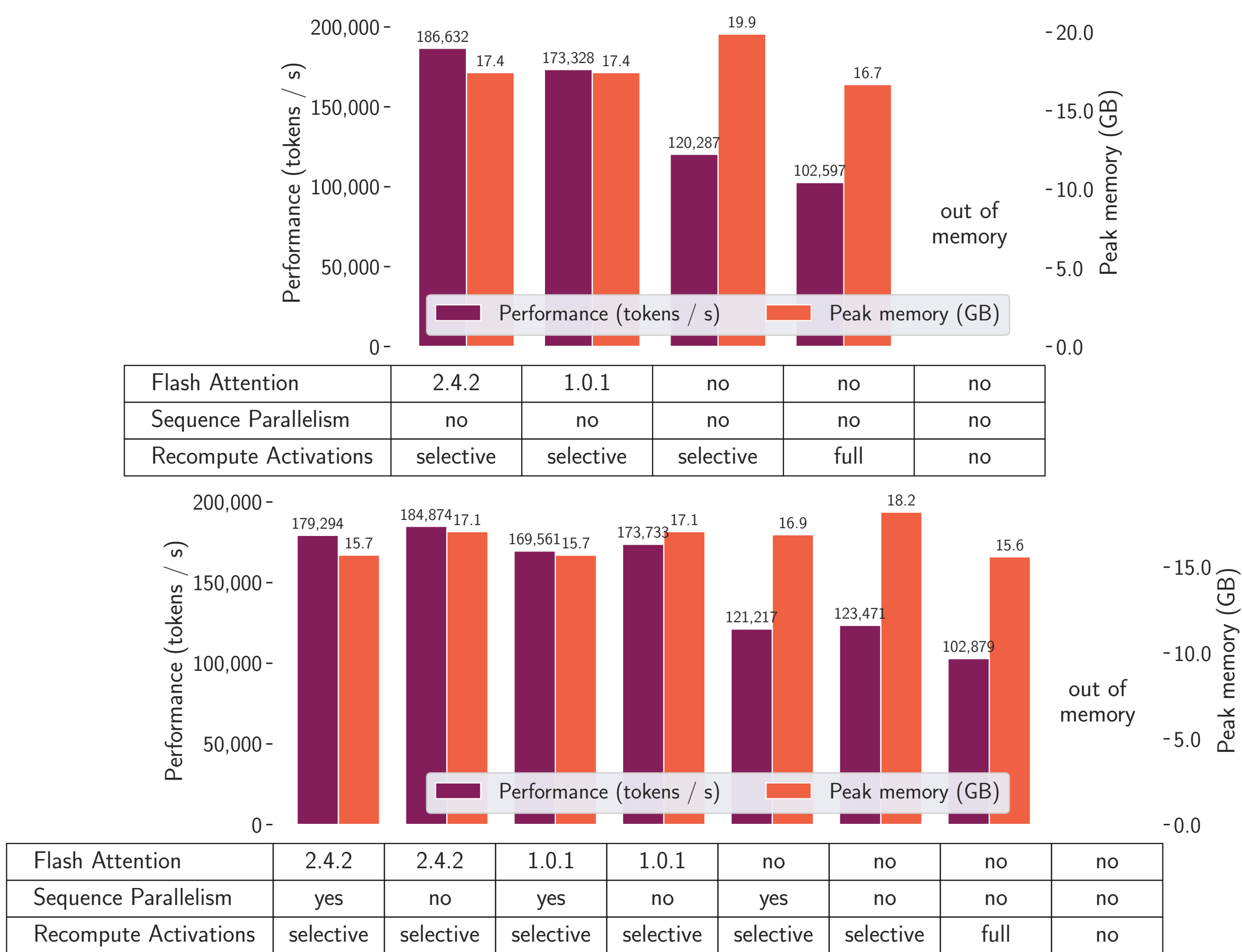


Figure 3: Running the training configuration from Figure 1, TP=1, PP=4 (top), TP=2, PP=2 (bottom) with specific features enabled.

- **Flash Attention improves throughput drastically.**
- **Activation recomputation may be needed to fit model into memory.**

References and Further Reading

Penke et al. (2025), *Training LLMs on HPC Systems: Best Practices from the OpenGPT-X Project*.

John et al. (2024), *Performance and Power: Systematic Evaluation of AI Workloads on Accelerators with CARAML*.

Hugging Face (2025), *The Ultra-Scale Playbook: Training LLMs on GPU Clusters*.

Comparing Accelerators with CARAML

CARAML (**C**ompact **A**utomated **R**e reproducible **A**ssessment of **M**achine **L**earning) is a benchmark framework for assessing AI workloads on accelerators.

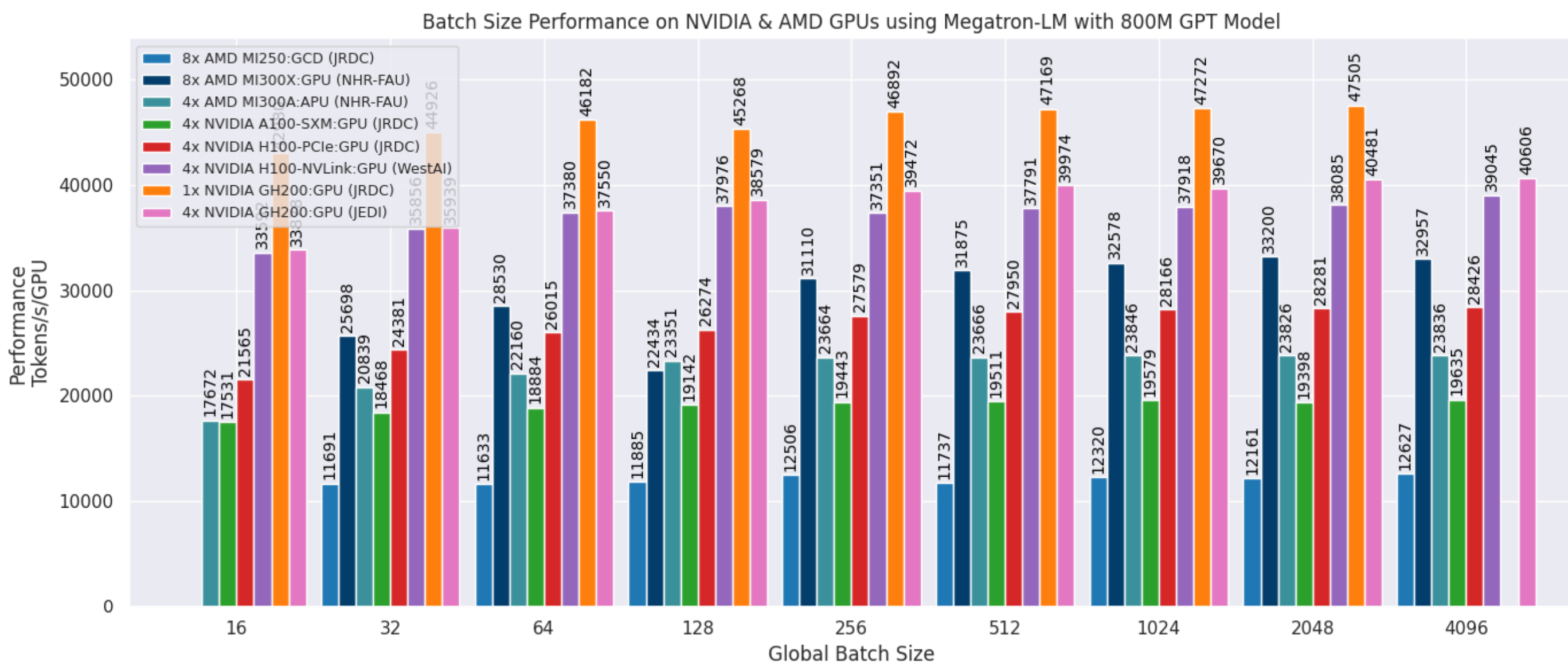
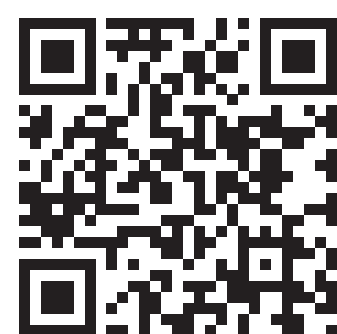


Figure 2: Throughput for training an 800M parameter model on various accelerator nodes.

- GBS=128 is a sweet spot in this setup.
- **Newer hardware improves throughput, but topology, node layout, and workload configuration matter as well.**

Challenges

```
jwb0694:13939:14102 [0] transport/net_ib.cc:94 NCCL WARN NET/IB : Got async event : port error
```

- **Flipping Links** mitigated by high verbosity error handling, checkpointing, ferrite beads on cables.
- Pinning behavior, container mismatches, race conditions, non-default hostnames

New Developments since 2025

- **New hardware:** Hopper, Blackwell, and larger NVLink domains (GB200 NVL72).
- **New LLM architectures:** Mixture-of-Experts and state-space models.
- **More ways to parallelize:** FSDP, FSDP2, HSDP, context parallelism, and expert parallelism.
- **Mixed-precision training** pushed further to MXFP8 and FP4.
- More **low-level optimizations:** asynchronicity via CUDA graphs, kernel fusions, fully asynchronous Tensor Cores, and softmax approximation.
- **Agentic coding** accelerates everything.

