Estonian language support in open-source large language models

Kairit Sirts, Aleksei Dorkin, Martin Vainikko

LUMI hackathon

Oslo, May 2025

Overall project goal

Continue training of pretrained open-source LLMs (like LLaMa) on Estonian data to improve:

- Estonian language capabilities and fluency
- Knowledge of Estonian language and culture

Our goals for the hackathon

Figure out how to continue training a 70B parameter model from the LLaMa-3.1-70B checkpoint.

- Use the latest rocm-megatron-lm code

- Figure out the optimal configuration of model parallelism required to fit the 70B model into GPU memory

- Scale up data parallelism by trying with a larger number of nodes

- Make sure that the loss of the same model matches when loaded into HF Accelerate and Megatron-LM

Goal: use the latest rocm-megatron-lm code

What we wanted

- Start using the latest rocm-megatron-lm code

- Convert the HF checkpoints to megatron-lm

- But: we got a converted TP8xPP8 checkpoint from Luoni Juoma

What we achieved

- Done ✅

- Partially achieved: 🟨
    - LlaMa3.1-8B with TP2xPP2 ✅
    - LlaMa3.1-8B with TP1xPP1(runs OOM when loading) ❌
    - LlaMa3.1-70B with TP8xPP4 (runs OOM when converting) ❌

- We were able to load it to megatron-lm and start training ✅

Goal: figure out model parallelism configuration

What we wanted

- Figure out how to fit 70B model into GPU memory

TODO:

- Check what is more optimal in terms of throughput:
  - TP8 x PP4 or
  - TP8 x PP8

What we achieved

- Done ✅

- The 70B model fits into 4 nodes with:
  - tensor parallelism TP = 8
  - pipeline parallelism PP = 4

Goal: scale up data parallelism to more nodes

What we wanted

- Scale up data parallelism to 100+ nodes ✅
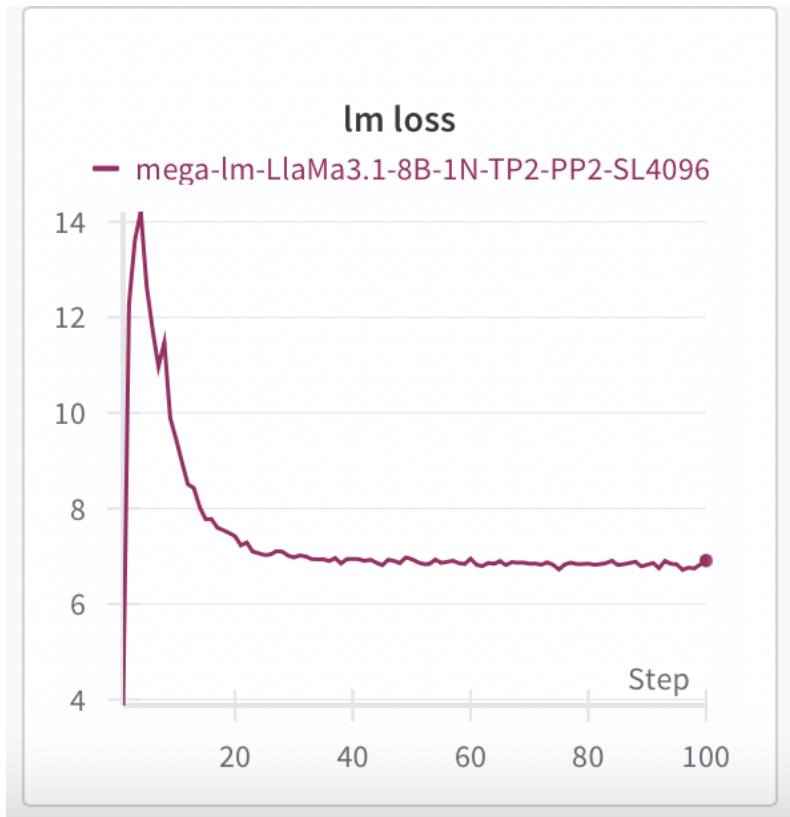
TODO:

- Scale up to 128-256 nodes

What we achieved

- Partially achieved 🟨
  - Managed succesfully to scale to 16 nodes (on Wednesday) ✅
  - On Thursday, was able to run max on 12 nodes, any runs using more nodes got distributed timeout error (maybe something in the system) ❌
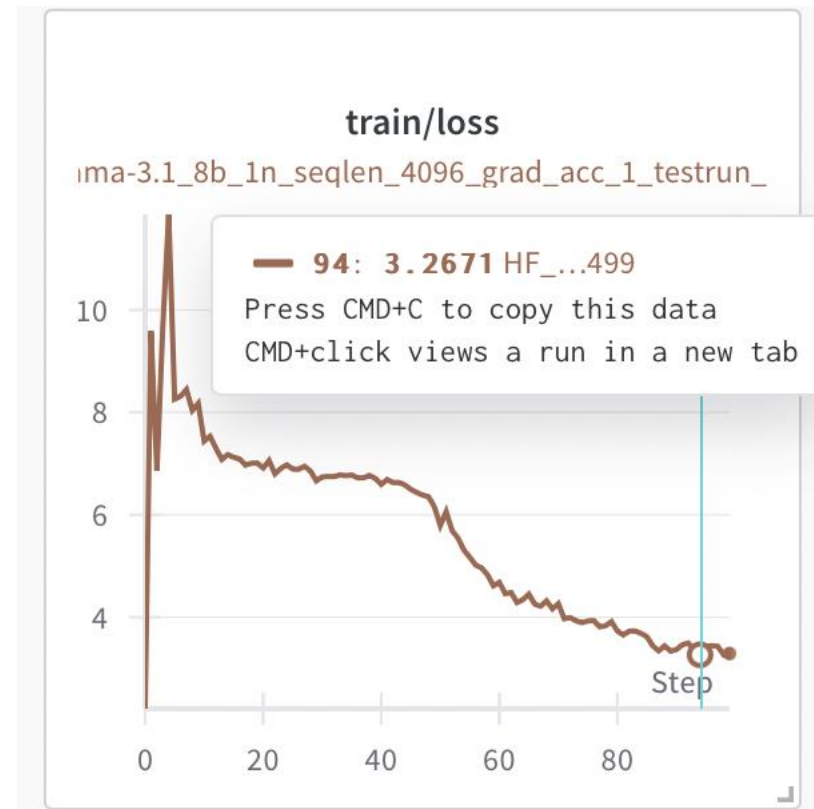
Goal: make sure the losses match

- Comparing LlaMa3.1-8B models: initially
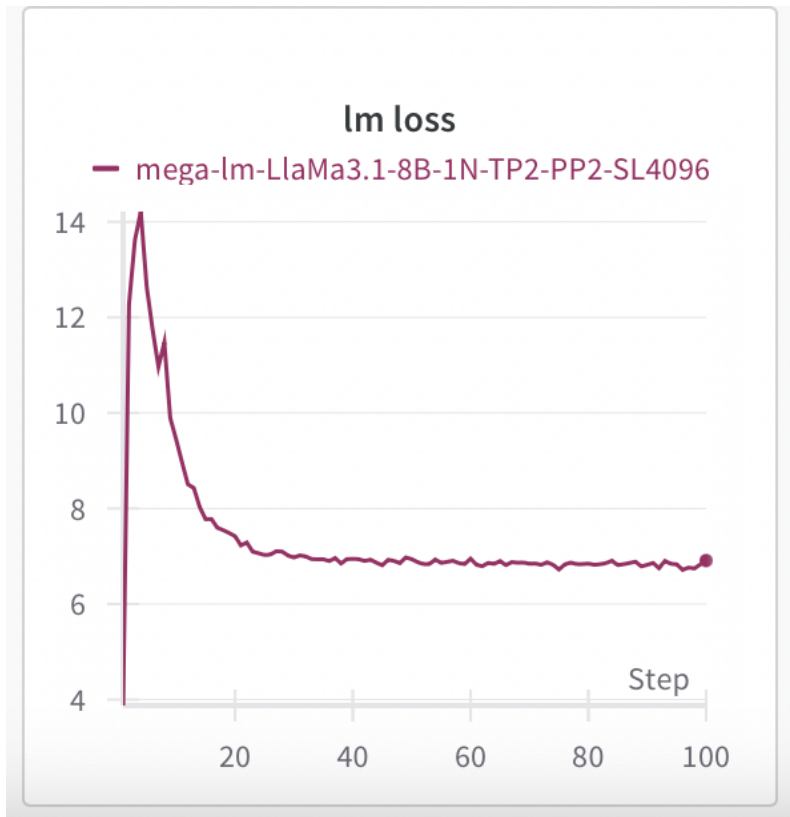
megatron-lm: PP2xTP2
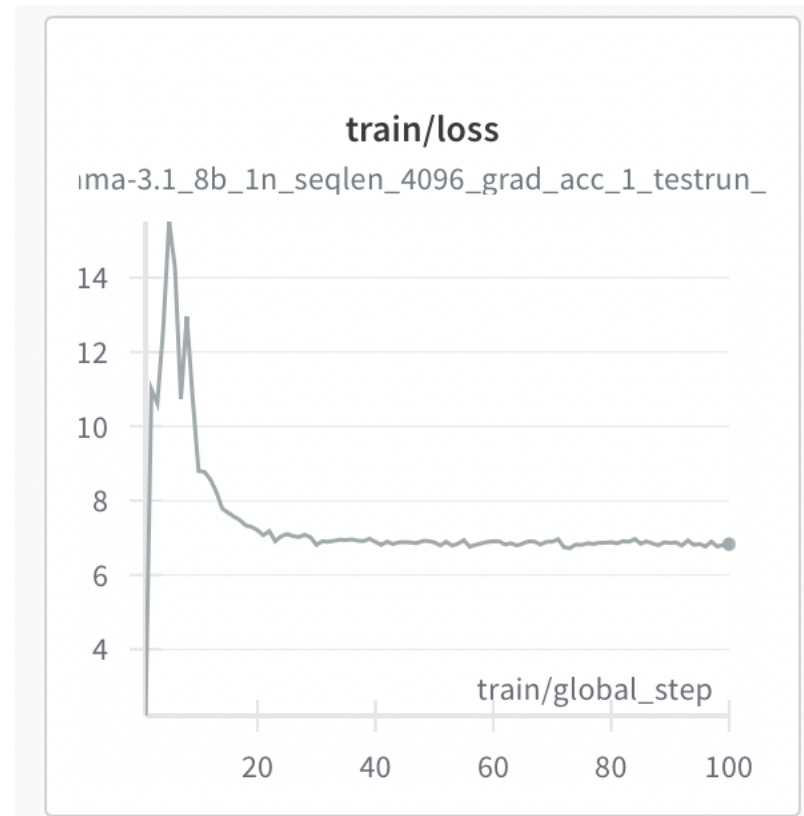
HF accelerate: no model parallelism

Goal: make sure the losses match

- After making sure the hyperparameters match: losses on LlaMa3.1-8B ✅

megatron-lm: PP2xTP2

HF accelerate: no model parallelism

Overall conclusions

We consider the hackathon a success:

- We learned a lot more about how to do distributed training in LUMI using the latest AMD-supported megatron-lm code

- We were partially able to achieve what we wanted → **we are in a good position to continue at home on our own**

- We were able to verify the equality of the losses under HF Accelerate and rocm-megatron LM → **we can be more confident in working towards starting large scale training with megatron-lm**

Next steps

- Figure out the model conversion from HF to megatron-lm
  - or use the converted checkpoints we received from Juoni Luoma
- Test which configuration is more optimal in terms of throughput:
  - TP8xPP4 or TP8xPP8
- Scale up to more nodes (both the HF Accelerate and megatron-lm training)
- Perform some hyperparameter search

- Start training the actual model on real data on the scale of 128-256 nodes