VECTOR INSTITUTE

Hugging Face

💥 BOOM: Scaling LLMs Training Across Public Supercomputers

Marco Ciccone

Distinguished Postdoctoral Fellow,
Vector Institute

NEURAL INFORMATION PROCESSING SYSTEMS 2025

Pre-training LLMs is a job for privileged researchers

Model Version	Largest Model Size (Parameters)	Training Tokens (Trillions)	GPU Type	GPU Count	Key Parallelism Strategies
Llama 1	65B	1.4	NVIDIA A100 (80GB)	2,048	Data & Model Parallelism (inferred)
Llama 2	70B	2.0	NVIDIA A100 (80GB)	Not Specified	FSDP, Tensor Parallelism
Llama 3.1	405B	15.6	NVIDIA H100 (80GB)	Up to 16,384	4D Parallelism (FSDP, TP, PP, CP)
Llama 4	400B (MoE, 17B active)	>30.0	Not Specified	Up to 32,000	Mixture-of-Experts (MoE), Multi-Dimensional Parallelism (inferred)

retrieved by Gemini

Clusters are getting larger and larger...

xAI Colossus cluster 200K GPUs

And it’s not going to stop!

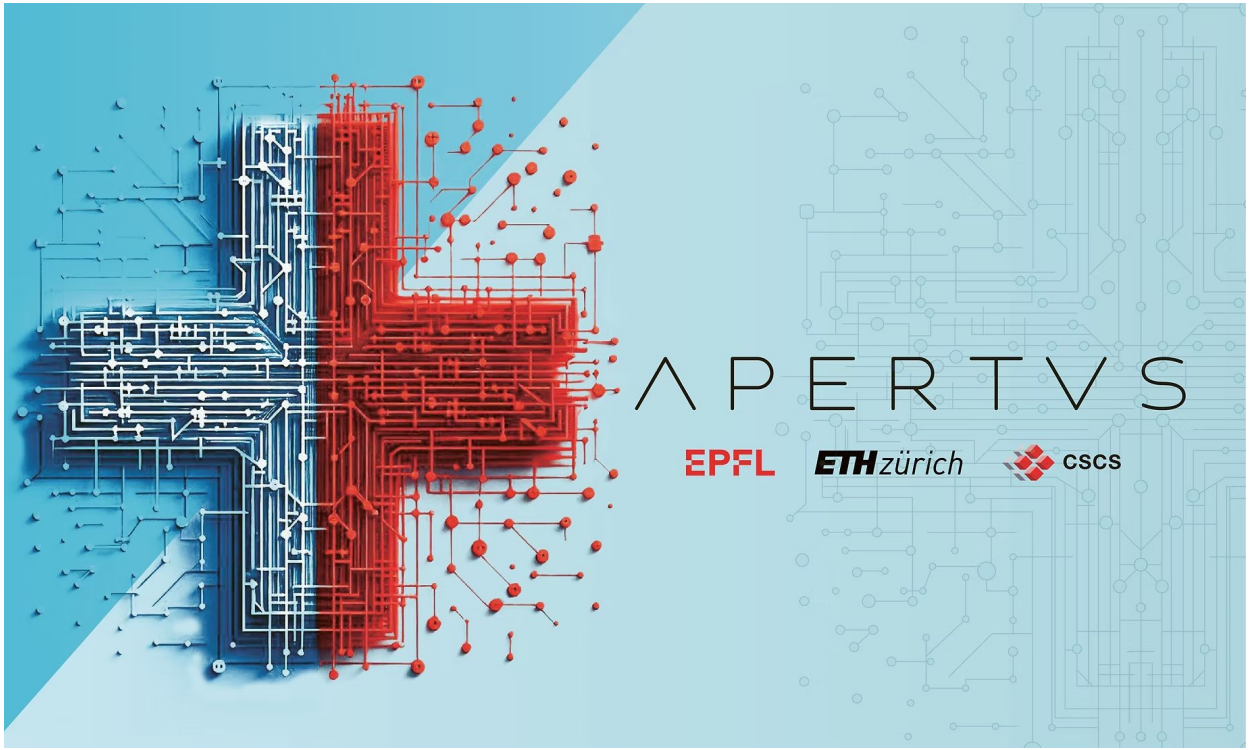
Largest AI Training Clusters By End of 2026			
AI Clusters	Anthropic - Project Rainier	OpenAI - Stargate	Meta - Prometheus
IT Power at YE2026	780 MW	880 MW	1,020 MW
Chip types	Trainium 2	GB200/300	GB200/300
# of Chips	800,000	400,000	500,000
Total TFLOPS	1,040,000,000	2,469,594,595	3,171,044,226
GPU/XPU Provider	AWS	Oracle	Meta

Source: SemiAnalysis Datacenter Model, SemiAnalysis Accelerator Model

Does it mean that only large companies can train frontier models?

Can Academia ever compete?

More and more examples of open LLMs trained by academic teams



8B, 70B multilingual LLMs (15 Trillion tokens)

Some numbers (approximately)

TOP-500

4th	🇩🇪 JUPITER	**24,000**	NVIDIA GH200	96GB
8th	🇨🇭 ALPS	**10,000**	NVIDIA GH200	96GB
9th	🇫🇮 LUMI	**12,000**	AMD MI250X	128GB
10th	🇮🇹 LEONARDO	**14,000**	NVIDIA A100	64GB
11th	🇬🇧 ISAMBARD	**5,448**	NVIDIA GH200	96GB
	🇪🇸 MARENOSTRUM 5	**4,480**	NVIDIA H100	64GB
	🇫🇷 JEAN ZAY	**1,500**	NVIDIA H100	80GB

From https://top500.org/lists/top500/2025/11/

Pretty good resources in isolation,
but in **aggregate** they are event better!

can we combine these resources?

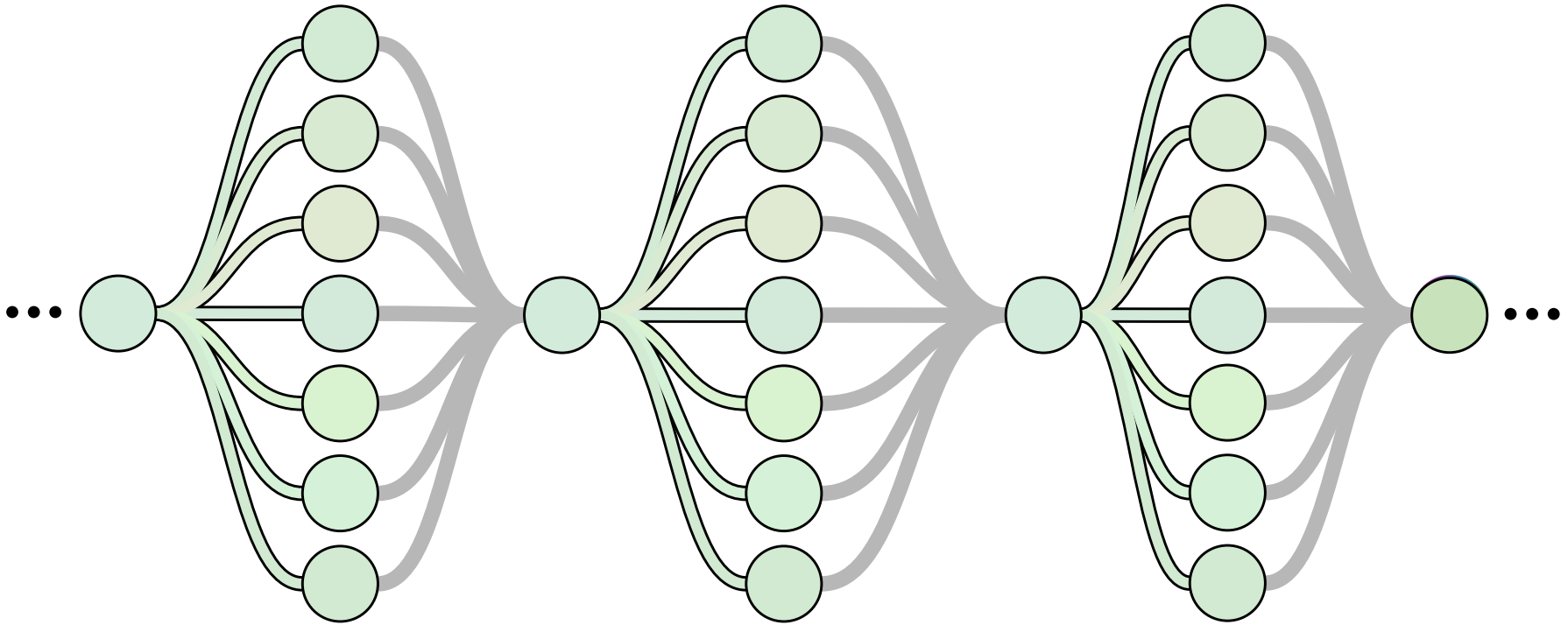
Collaborative Training of LLMs across public infrastructure

To connect clusters via the internet

we need to reduce the
communication cost of SGD

Federated Learning *(Local SGD) reduces communication cost by*

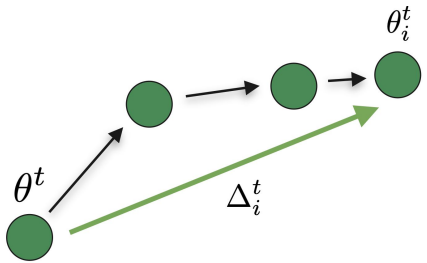
Independent training + sparse synchronization



From "Communication-Efficient Learning of Deep Networks from Decentralized Data" by McMahan et al. (2016)

Parameter Averaging *as core operation for FL*

$$\theta^{t+1} = \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\theta_i^t = \theta^t - \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\underbrace{\theta^t - \theta_i^t}_{\Delta_i^t}$$



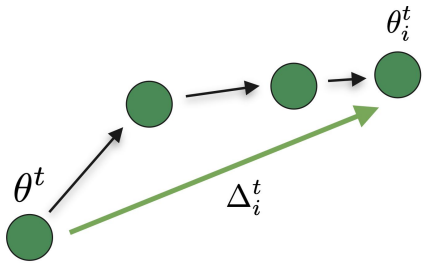
$$\theta^{t+1} = \theta^t - \eta\Delta^t$$

Equivalent to FedAVG or Local SGD when outer learning rate is 1

From "Adaptive Federated Optimization by Reddi et al. (2020)

Parameter Averaging *as core operation for FL*

$$\theta^{t+1} = \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\theta_i^t = \theta^t - \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\underbrace{\theta^t - \theta_i^t}_{\Delta_i^t}$$

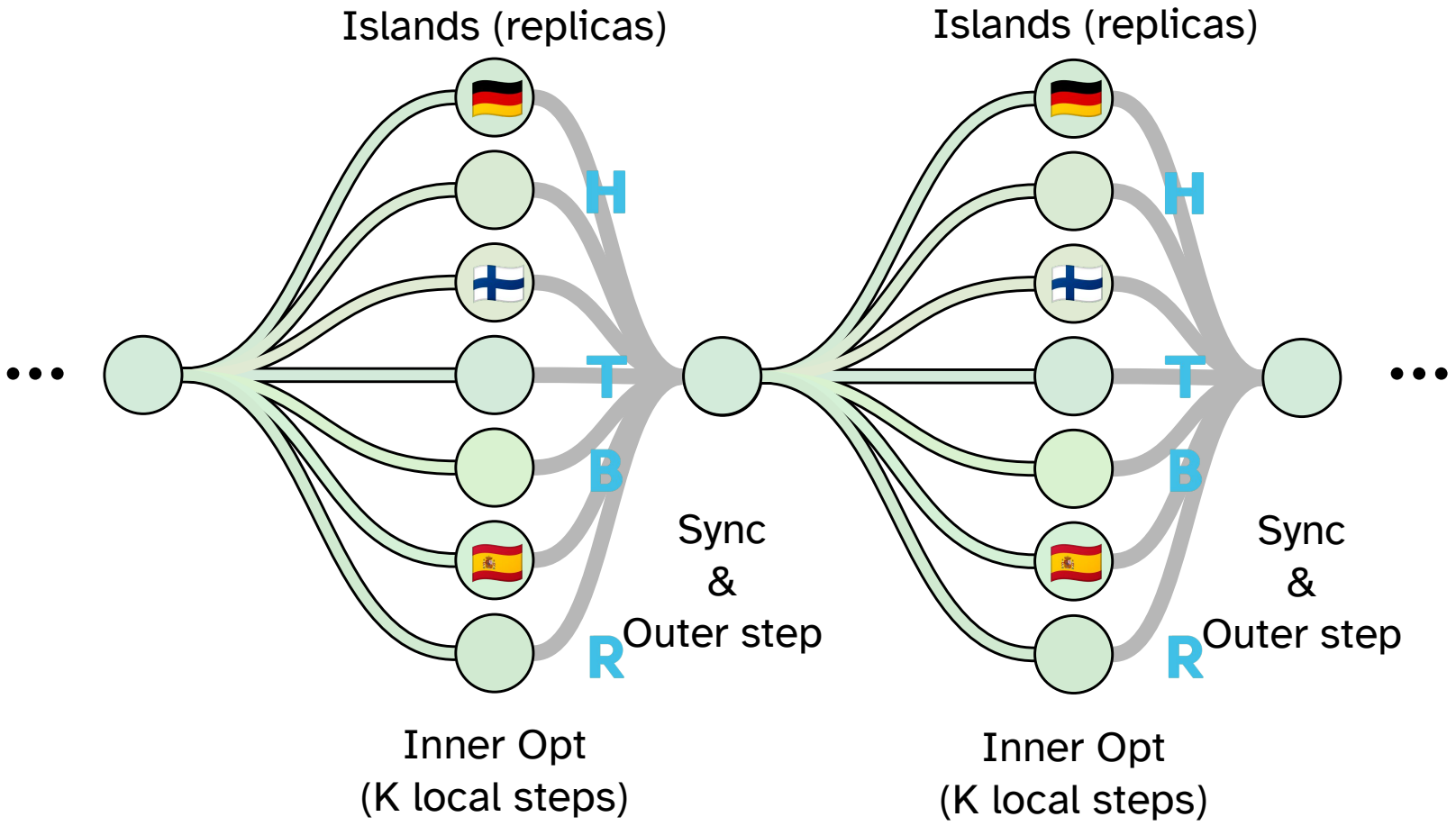


$$\theta^{t+1} = \textsc{ServerOpt}(\theta^t, -\Delta^t, \eta)$$

The displacement from initialization can be used as **pseudo-gradient for an outer/server optimizer**

From "Adaptive Federated Optimization by Reddi et al. (2020)

DILOCO = inner Adam + Outer Nesterov Momentum



Communication-Efficient Language Model Training Scales Reliably and Robustly:
Scaling Laws for DiLoCo

Zachary Charles[1*] Gabriel Teston[2] Lucio Dery[3] Keith Rush[1]
Nova Fallen[1] Zachary Garrett[1] Arthur Szlam[3] Arthur Douillard[3]

Harder: DiLoCo's hyperparameters are robust and predictable across model scales.

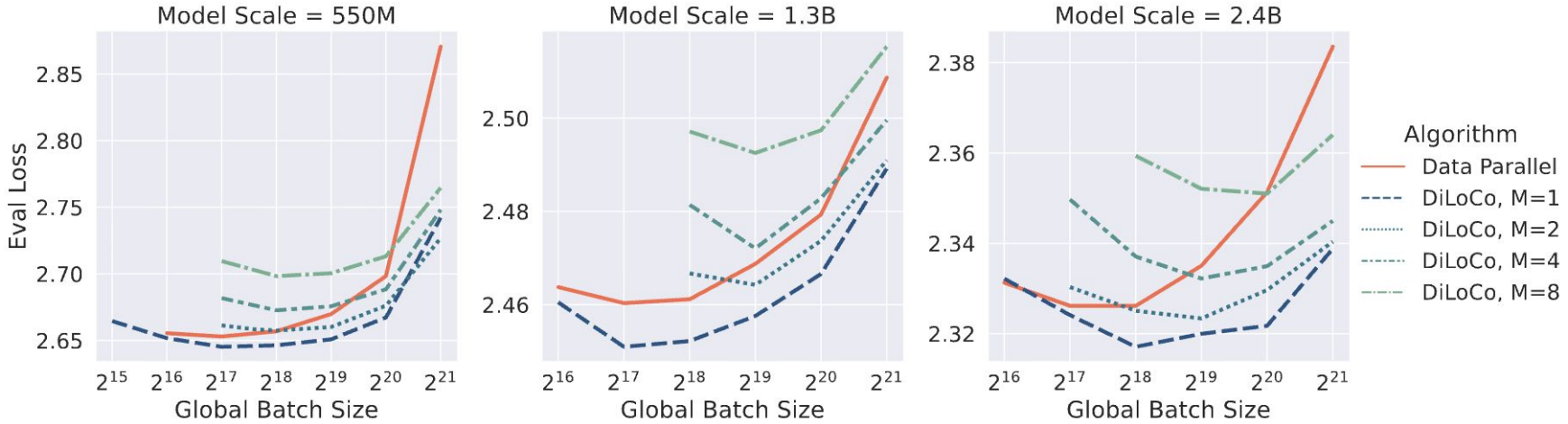
Better: DiLoCo further improves over data-parallel training as model size increases.

Faster: DiLoCo uses orders of magnitude less bandwidth than data-parallel training.

Stronger: DiLoCo tolerates a significantly larger batch size than data-parallel training.

And it works well!

Finding 3: Optimal batch size. DiLoCo increases the optimal batch size and moreover, the optimal global batch size increases with M (see Figures 4 and 5). This means that DiLoCo improves horizontal scalability relative to Data-Parallel (see Figure 6).



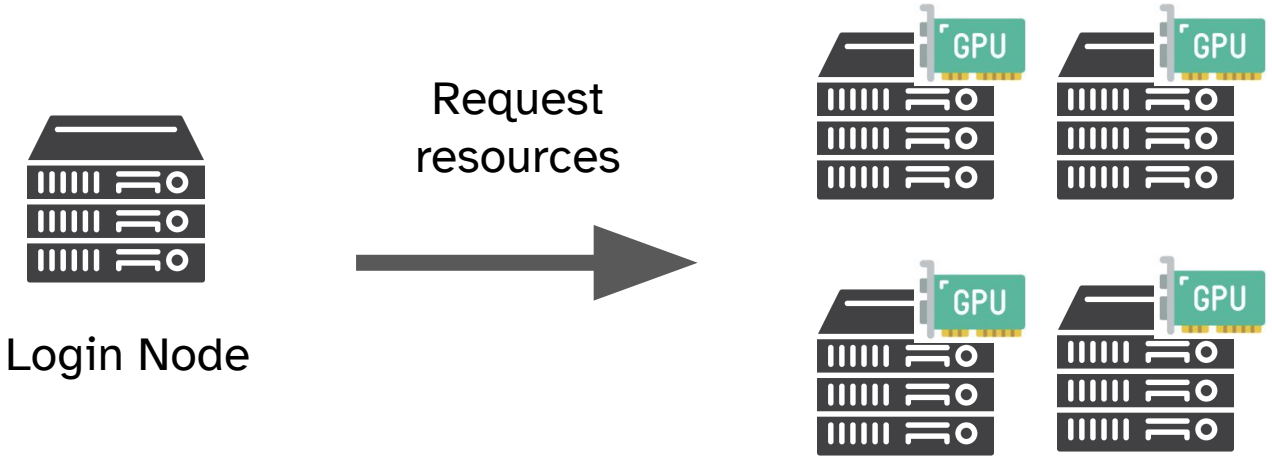
INTELLECT-1: 10B model, trained on 1 trillion tokens

From "INTELLECT-1 Technical Report" by Jaghouar, Sami, et al.

Complications with public infrastructures

slurm
workload manager

SLURM based clusters



SLURM based clusters

⚖️ Fairshare queues
(users competing for resources)

⏰ Max walltime
(24 to 72 hours)

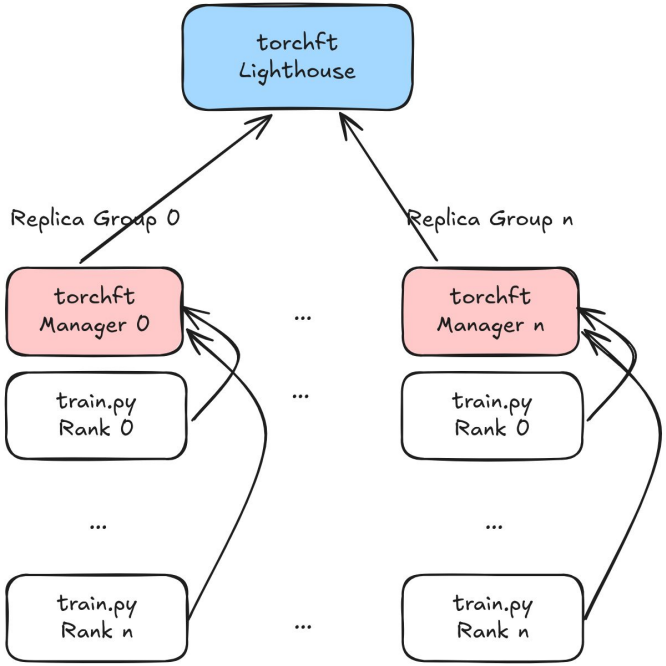
Very low uptime
Frequent maintenance

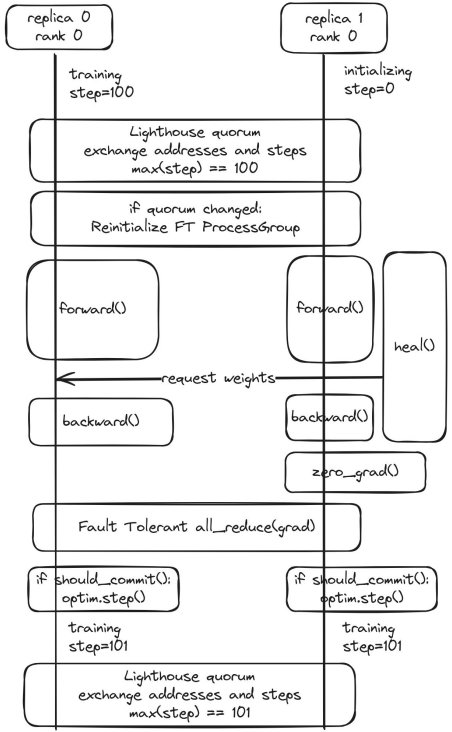
(Complex systems, understaffed teams)
They are doing their best!

PyTorch

Fault-tolerance (TorchFT)

islands could join and leave without disrupting the training





https://github.com/meta-pytorch/torchft

Compute nodes generally don’t have internet connection (for security reasons)

Some they don’t have it even on the login nodes

NO pip, uv, S3 …
only rsync

Nothing much we can do here

Negotiation with sysadmin and leadership

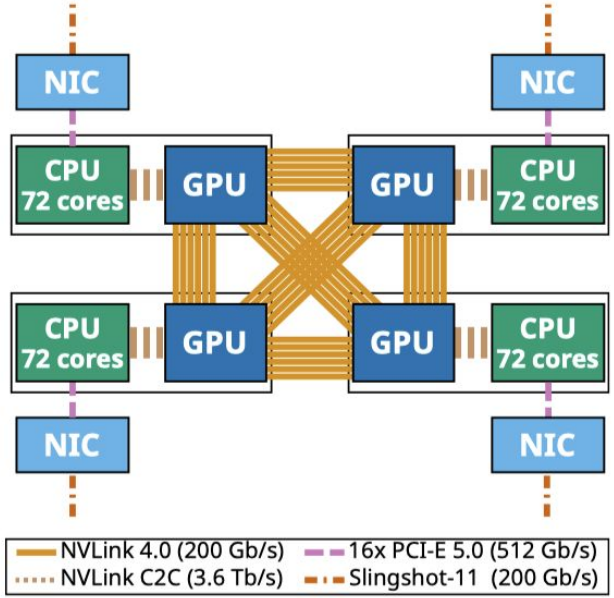
Heterogeneous Hardware and custom configurations

Different GPU generations
(NVIDIA A100, H100, GH200, AMD MIX250)

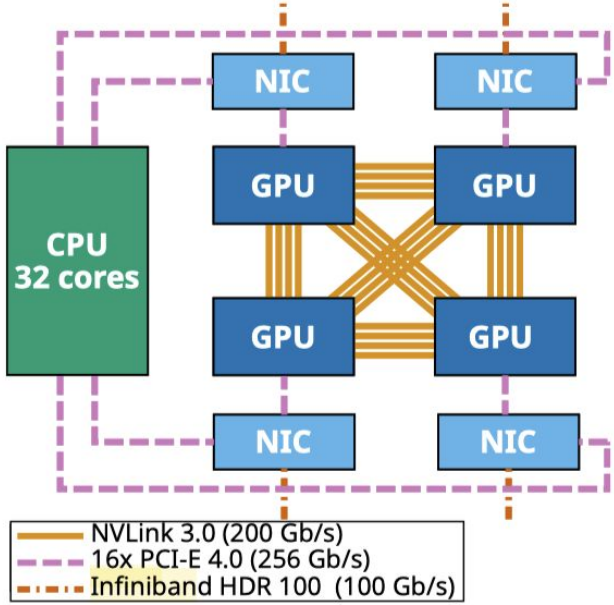
64GB, 96GB, 128GB

Inter-node vs Intra-node connectivity

NVLink (intra-node communication) is faster than inter-node (Infiniband, Slingshot)



(a) Alps.

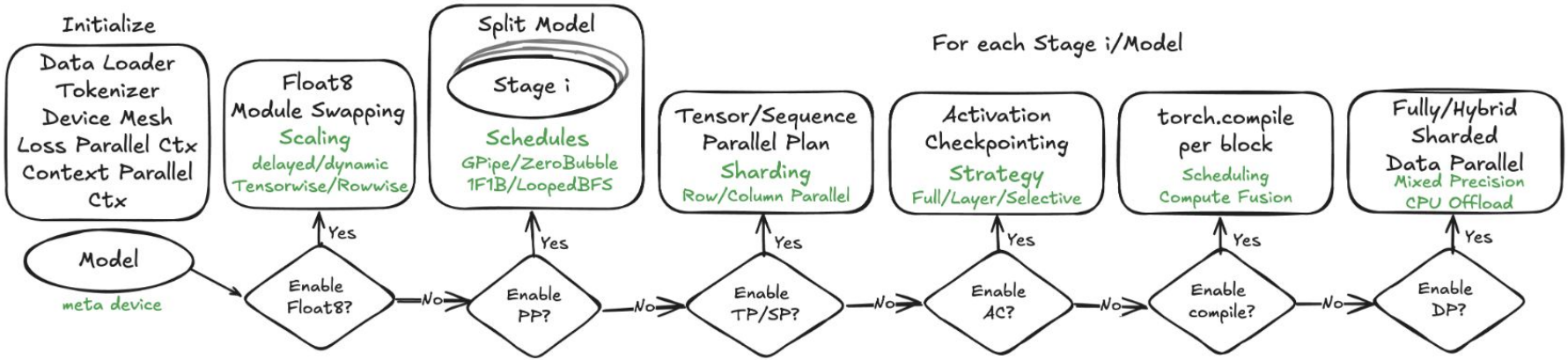


(b) Leonardo.

PyTorch

TorchTitan

composable, pytorch native framework for pre-training
Integration with TorchFT (LocalSGD, DILOCO)



Hybrid Sharded Data Parallel (HSDP)

2D parallelism, less communication across gpus as DDP communicates only gradients (all-reduce).

- FSDP within each ***island*** of GPUs
- DDP ***across N islands (DP replicas)***

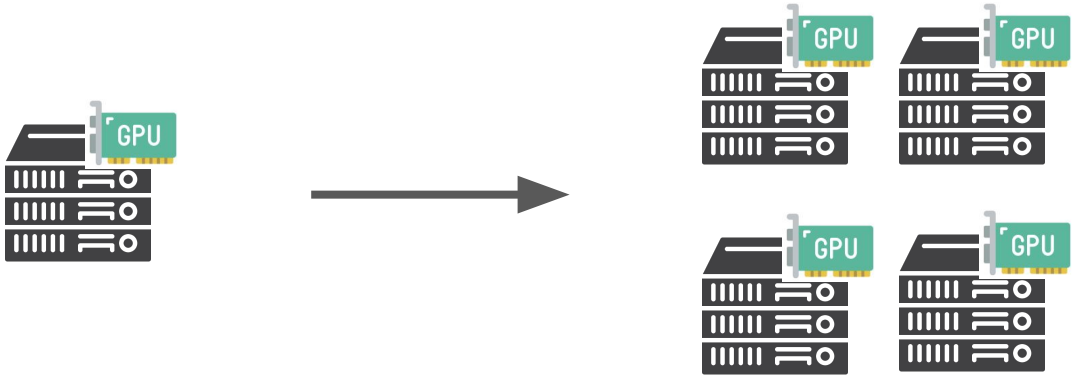
384 gpus = 96 nodes x 4

dp_replicates	memory per gpu	tps	tflops	mfu
1	39.71GiB(62.61%)	2,464	229.82	23.24%
4	40.32GiB(63.57%)	3,389	316.11	31.96%
8	49.51GiB(78.05%)	3,484	325.03	32.86%
12	50.60GiB(79.77%)	**3,605**	336.29	**34.00%**

Training

250K H100 hours euroHPC grant on BSC MareNostrum 5 🇪🇸

250K H100 hours euroHPC grant on BSC MareNostrum 5 🇪🇸
🚫📶 *no internet, and allocation only for 3 months*

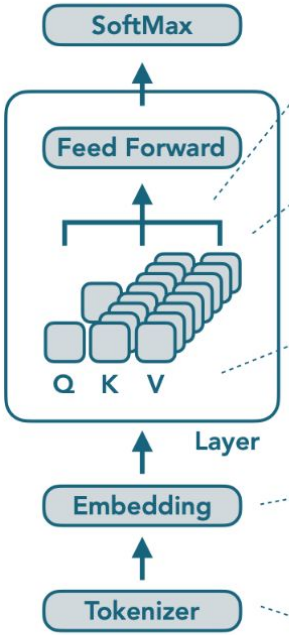


1) single-cluster warmup

2) multi-cluster diloco

🌱 *We can use this run as centralized baseline to derisk diloco*

Architecture and training details



Grouped Query Attention

The 16 attention heads share 4 query which maintains the performance of full multi-head attention and saves memory during inference.

Intra-Document Masking

Attention masking ensures that tokens from different documents in the same training sequence don't attend to each other which helps with long context training.

NoPE

We use NoPE which selectively removing rotary position embeddings from every 4th layer improving long context performance.

No Weight Decay in Embeddings

We remove weight decay from embedding layers to improve training stability as embedding norms naturally stabilize at healthier values.

Multilingual Tokenizer

We use the Llama 3.2 tokenizer which covers all languages used for pretraining.

14B parameters (dense model)

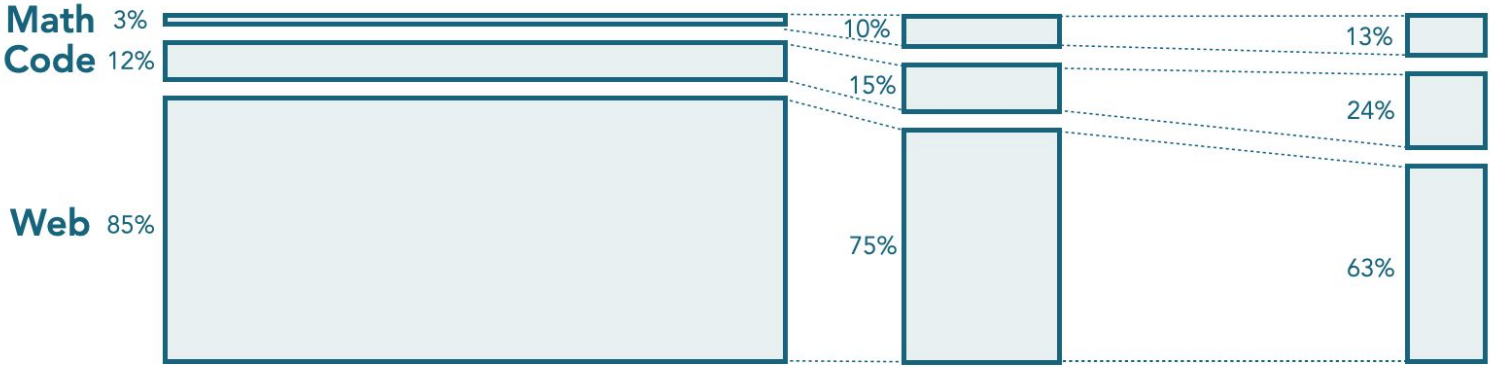
40 Layers
40 Attention heads
8 shared QV

QK-norm
z-loss

4096 Seq Length
13M Batch size
WSD schedule

Precision: bf16

Pre-training data 11T tokens *(from Smollm 3B)*



Phase I

Description: Base training

Duration: 8T tokens

Datasets: Base mix for pretraining

Web: FineWeb-Edu, DCLM, FineWeb2, FineWeb2-HQ

Code: The Stack v2 (16 langs), StarCoder2 PRs,

Code: Jupyter/Kaggle NBs, GH issues, StackExchange

Math: FineMath3+ | InfiWebMath3+

Phase II

Description: High quality injection

Duration: 2T tokens

Datasets: Adding Stack-Edu, FineMath4+, InfiWebMath4+, MegaMath (incl. Qwen Q&A, Pro synthetic rewrites, and text code interleaved blocks)

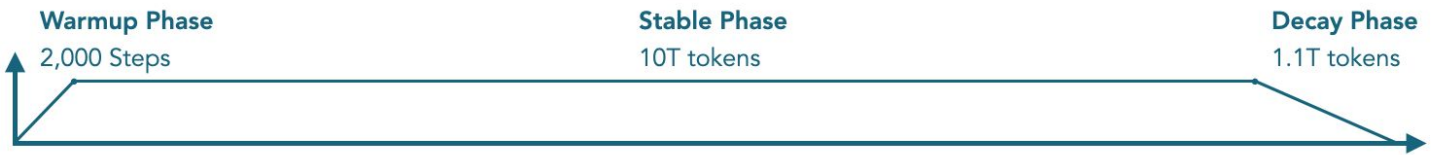
Phase III

Description: LR Decay

Duration: 1.1T tokens

Datasets: Upsampling high quality code/math datasets and adding instruction/reasoning data such as OpenMathReasoning

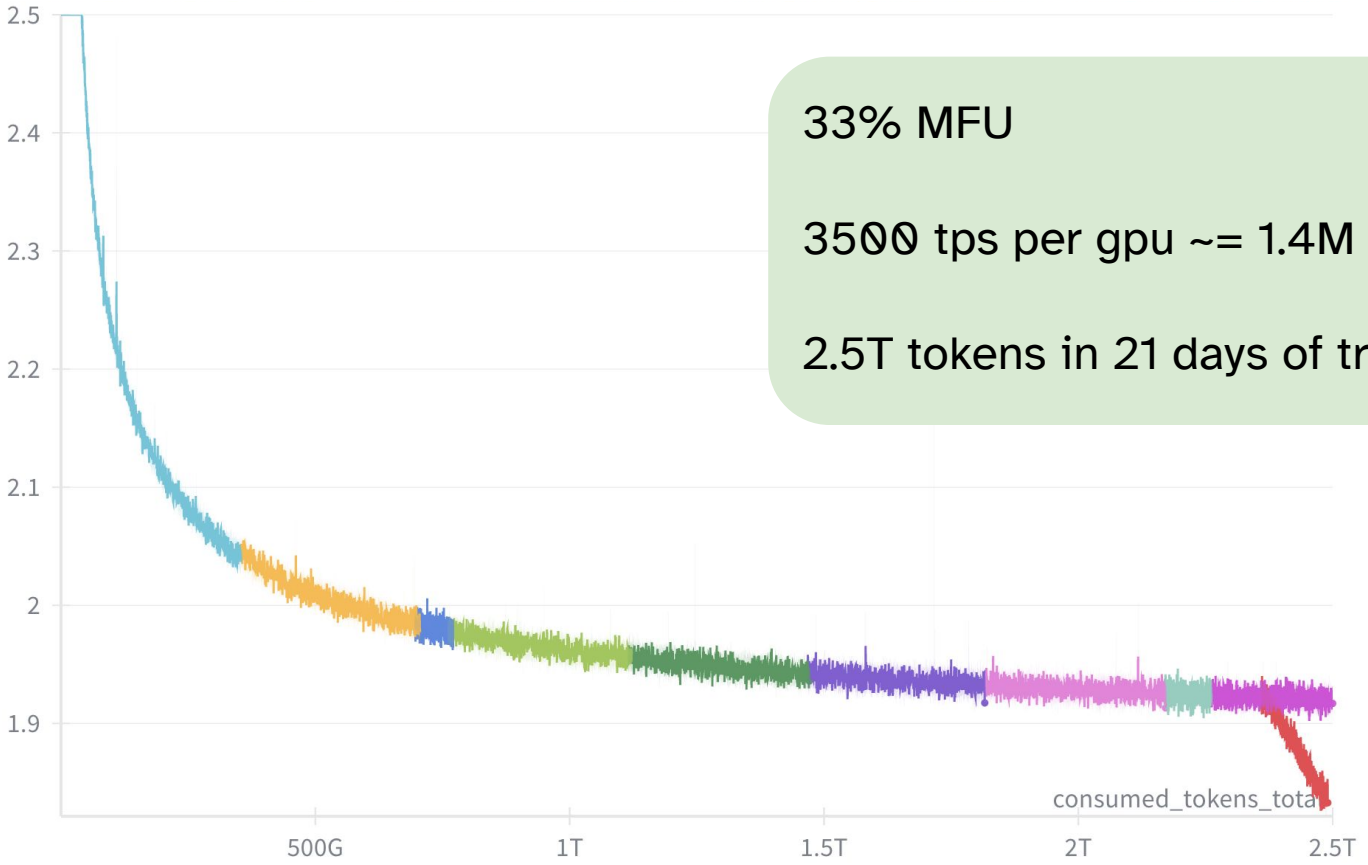
LR Schedule



Final warmup setup

- **400 GPUS** = 100 nodes (max per job) x 4 gpus
- HSDP with 10 dp_replicas x 40 gpus (fsdp)
- Full Activation Checkpoint
- per-gpu batch size: 8
- global batch size: 3200 x 4096 (13M tokens)

Warmup training on MN5 *(2.5 T tokens)*

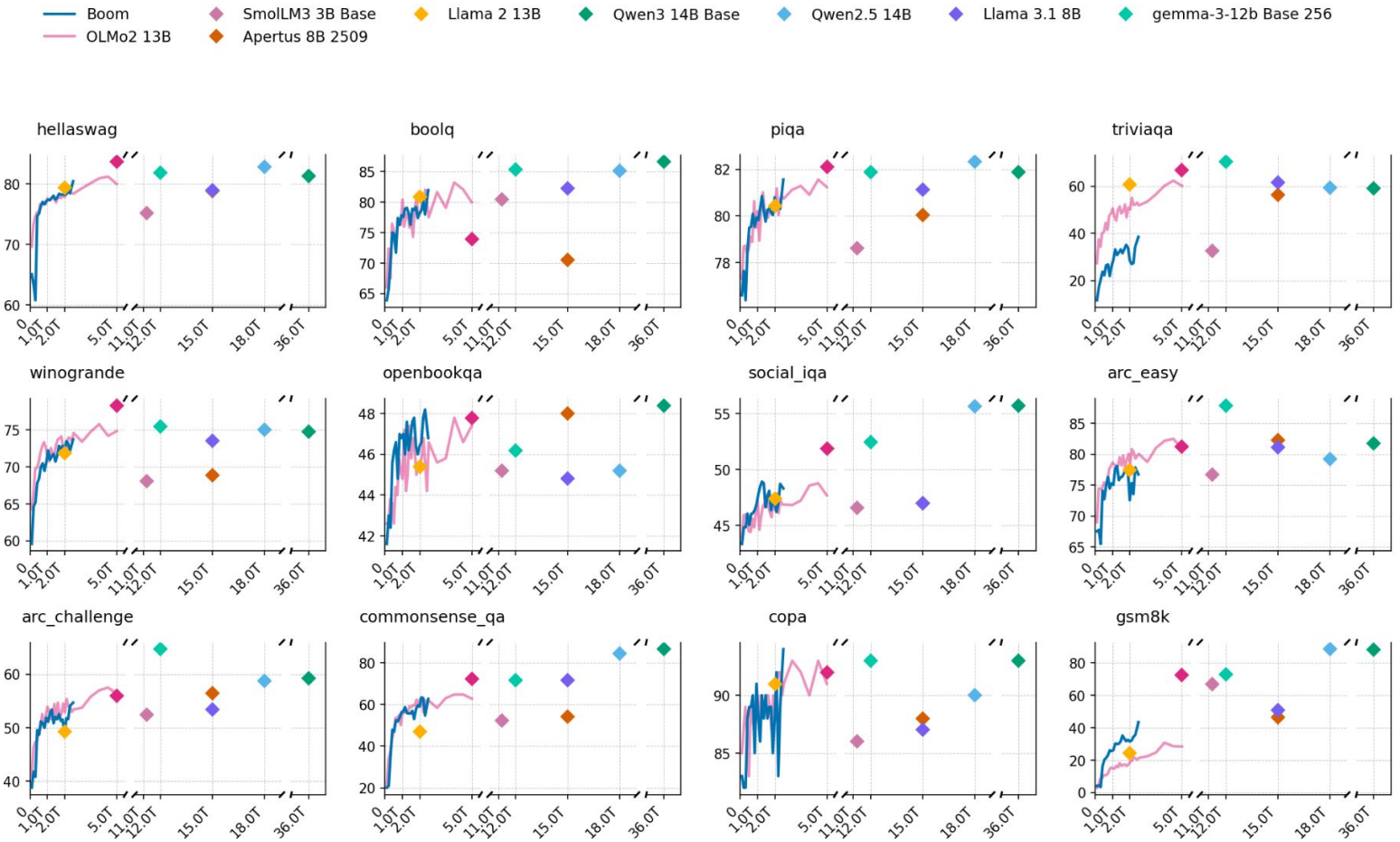


33% MFU

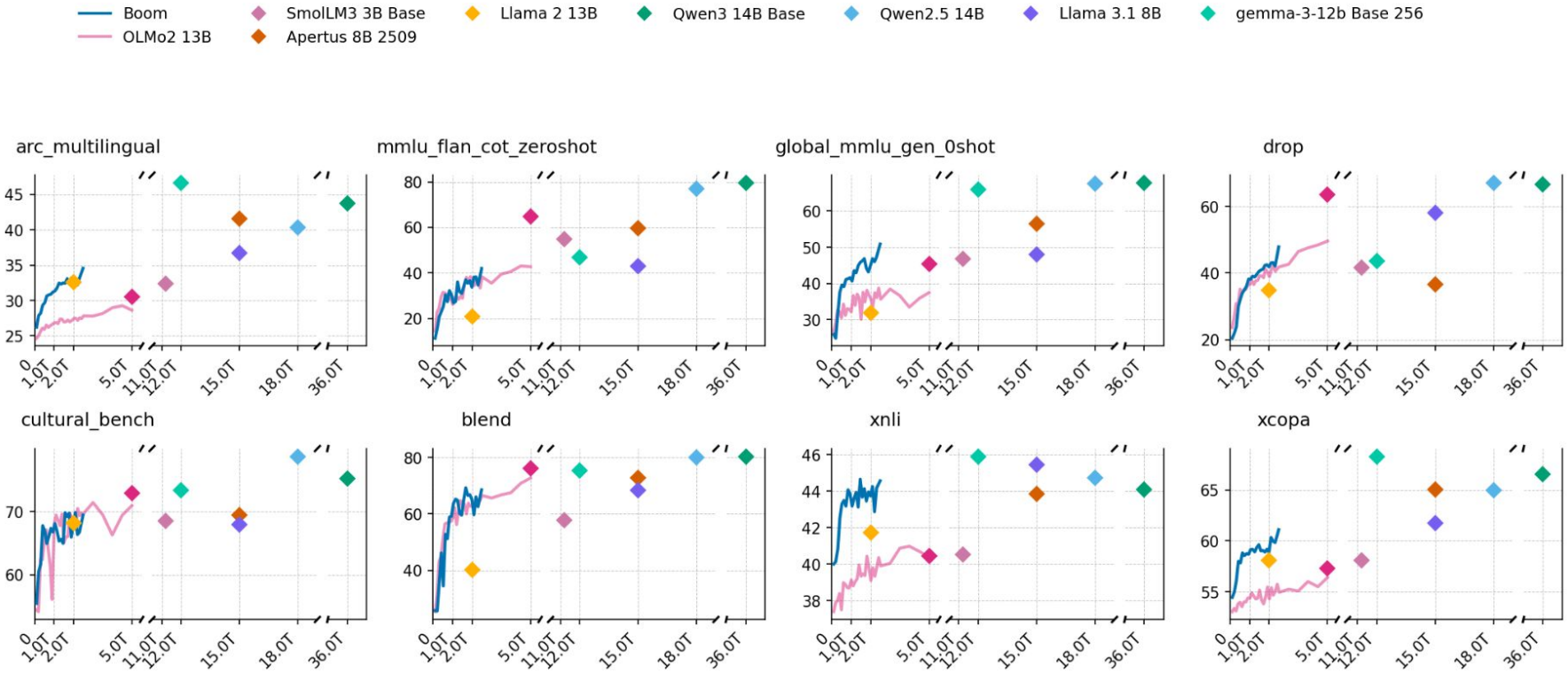
3500 tps per gpu ~= 1.4M tps

2.5T tokens in 21 days of training

Intermediate eval *(2.5 T tokens)*

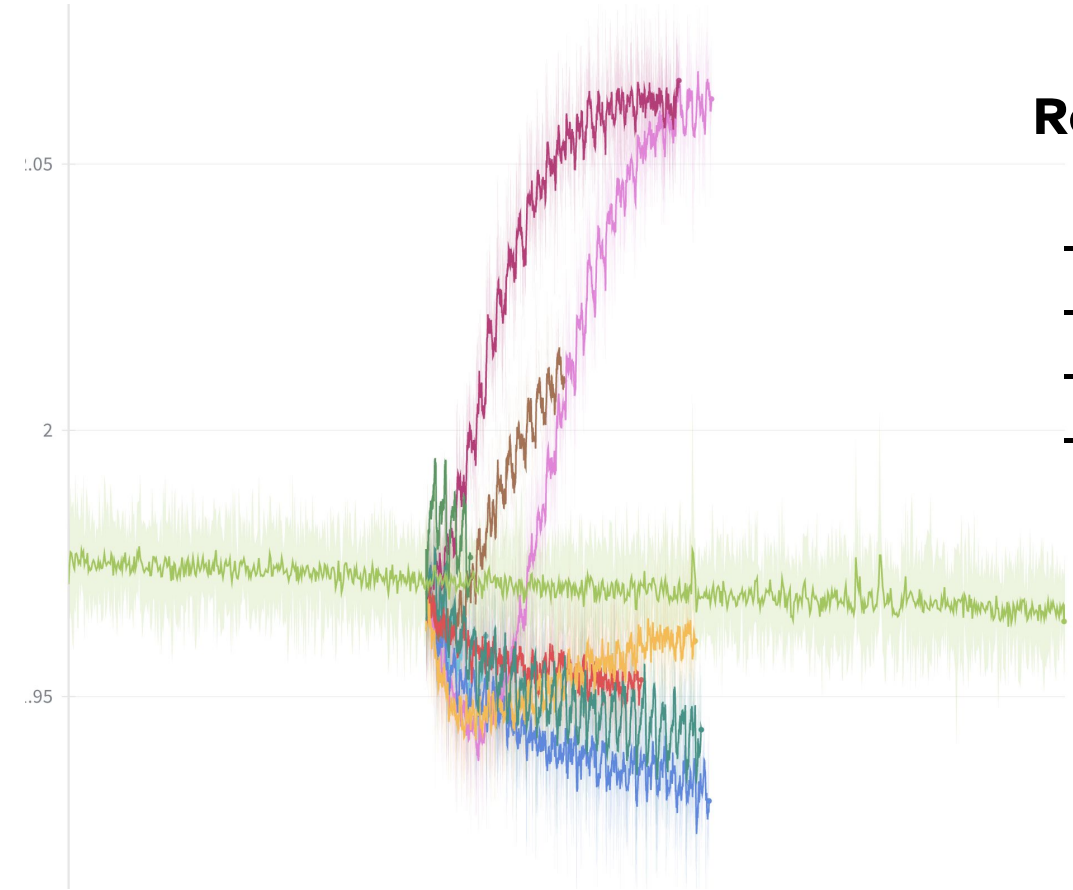


Intermediate eval *(2.5 T tokens)*



Diloco run

0.1) Testing Diloco

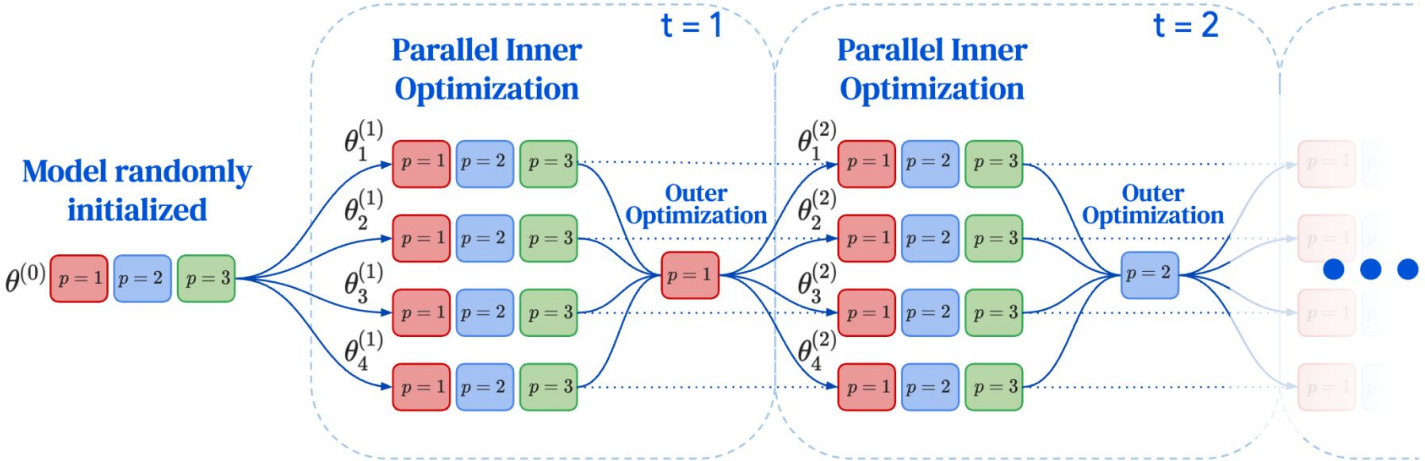


Restarting 1T ckp

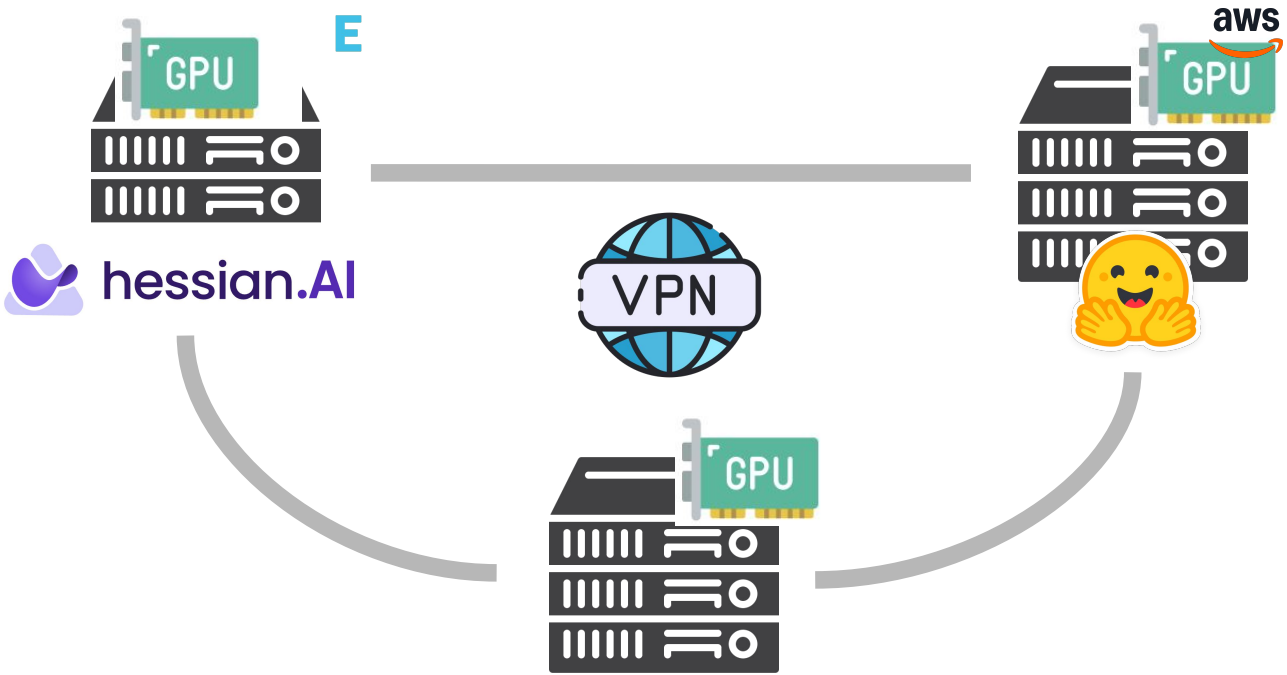
- split global BS across M replicas
- 100 inner steps
- adjust LR based on local BS
- warmup outer momentum

0.2) Testing Diloco

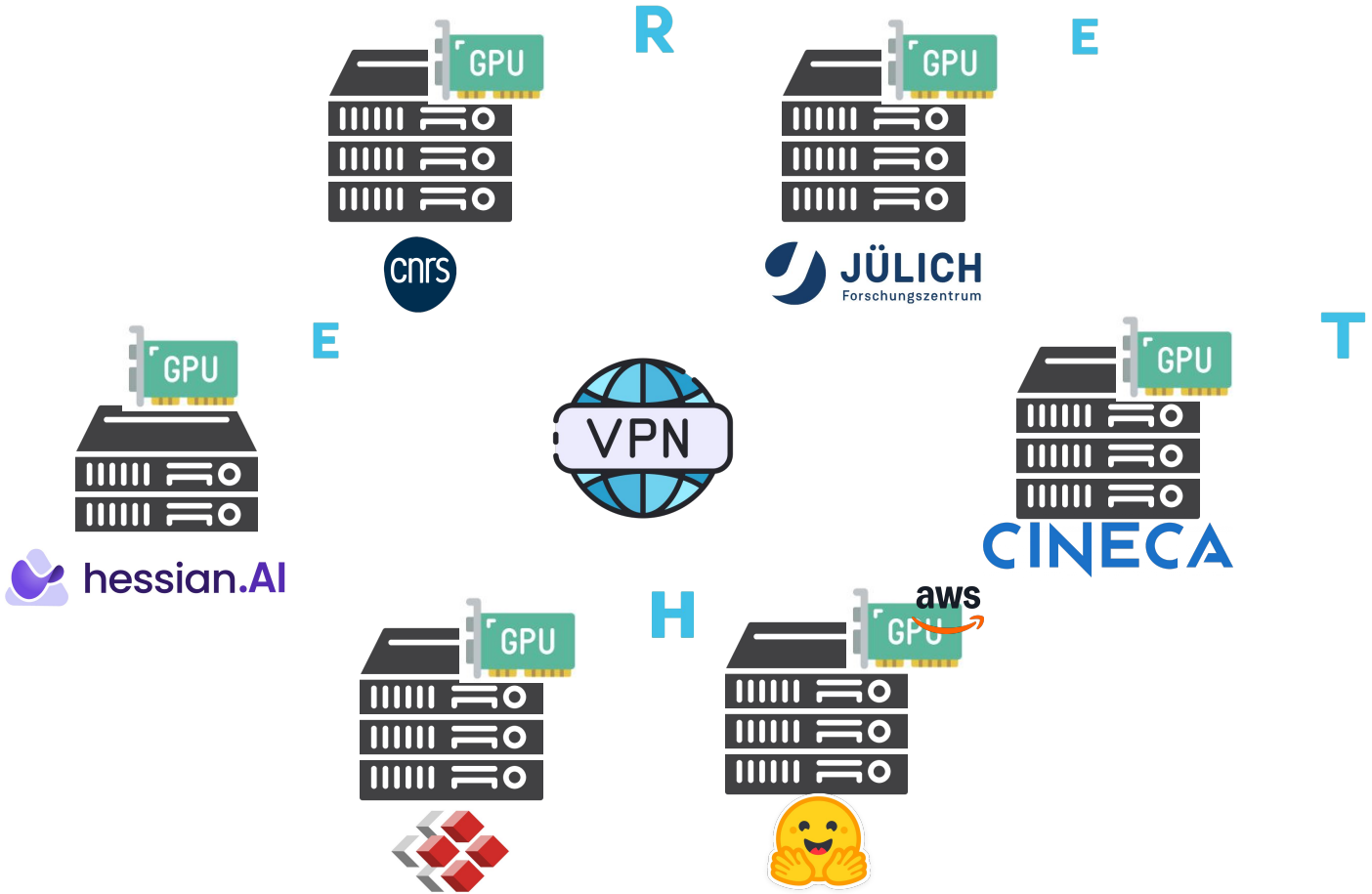
- Outer gradient quantization : 8-4 bits
- Streaming Diloco (to reduce peak communication)



Next steps: ***1) POC training***



Next steps: ***2) expand to more clusters***



Final remarks

Our goal is to demonstrate the viability of public infrastructure for

large-scale training via decentralized low-communication techniques

14B multilingual model on 11T tokens

Trained across (more than) 3 european institutions

Largest decentralized run across public compute

Elie 10:07 PM

so you tell me you are running 800 gpus right now?

On that day, at 10:07 pm, 19th August 2025,
I realized I was not gpu-poor anymore.

For now.

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

Hugging Face

Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación

Thank you!

If you are interested and want to contribute, please get in touch!

marco.ciccone@vectorinstitute.ai
craffel@gmail.com