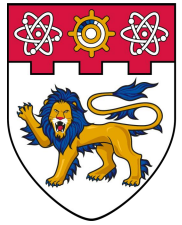


NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Di-PS: System-Algorithm Co-Design for Asynchronous and Heterogeneous Cross-cluster LLM Training at Scale

Shengwei Li*†, Qiaoling Chen*‡§, Zhiquan Lai†✉, Penglong Jiao‡, Wenwen Qu‡, Kun Cai‡, Jixing Li‡, Peng Sun‡, Xingcheng Zhang‡, Xiaoge Deng†, Dongsheng Li†, Kai Lu†, Tianwen Zhang§

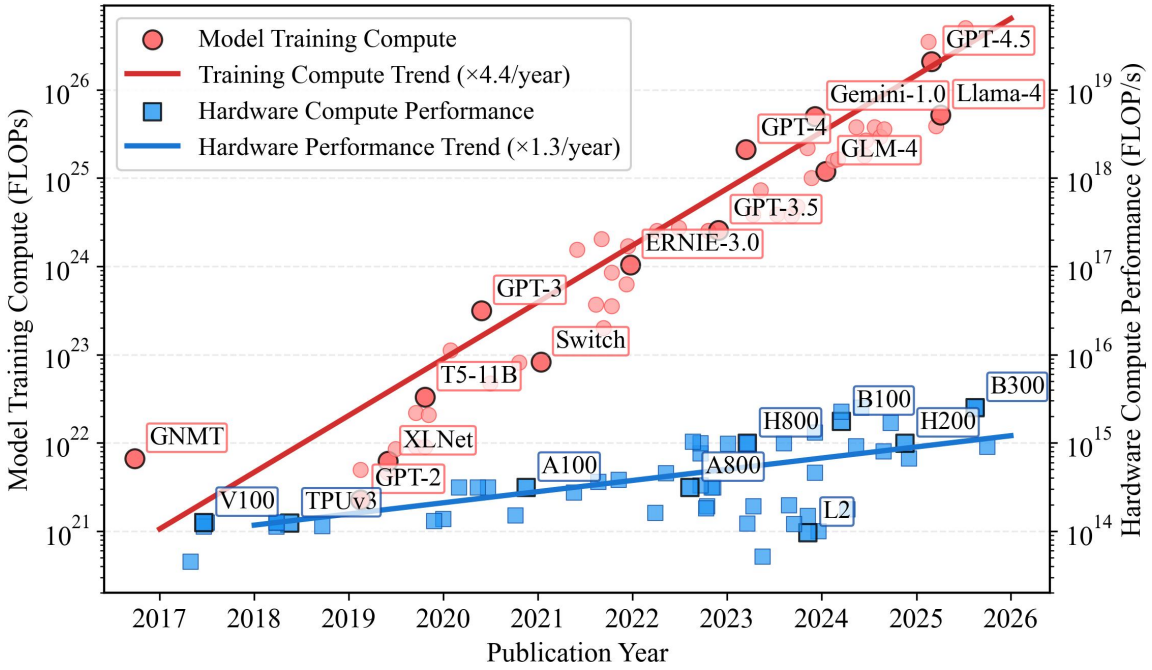
† National University of Defense Technology ‡ Shanghai AI Laboratory
§ Nanyang Technological University

*: Co-first Authors

LLM Training

Increasing LLM Training Scales

- While scaling laws lasts, exponentially increasing compute FLOP



Training compute of LLMs v.s. Hardware compute performance

Model	# of Accelerators
LLaMA-3.1	16,384
Grok-2	~20,000
LLaMA-4	32,000

LLM training scales

Data sources: https://epoch.ai/data-insights/compute-trend-post-2010

Cross-cluster LLM Training

Why Cross-cluster Training?

- Approximately 85% of clusters have NPUs less than 10K.

Cluster with 10,000 NPUs

NPU quantity on clusters

Data sources: https://epoch.ai/

Cross-cluster LLM Training

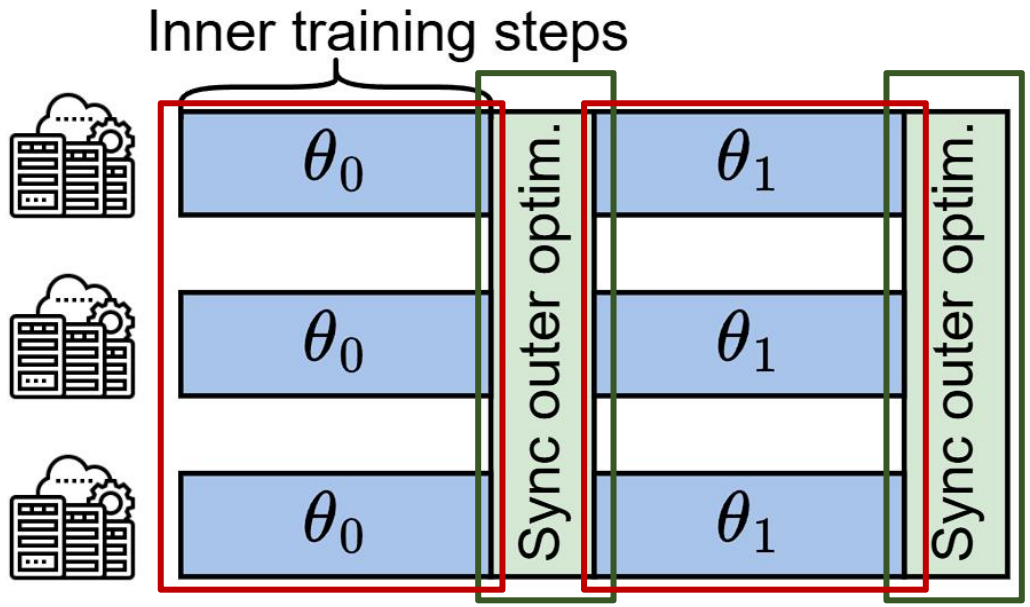
Why Cross-cluster Training?

- Consolidating existing smaller clusters is practical, compared to building single monolithic cluster.
  - Δ **Lower cost:** Fewer switches and lower network infrastructure cost.
  - Δ **Easier deployment:** Better fits datacenter power, cooling, and space limits.
  - Δ **Better resource match:** Large homogeneous clusters are scarce, while most LLM jobs need fewer than 100 NPUs.
  - Δ **Higher practicality:** Very large clusters are hard to utilize efficiently; smaller clusters are easier to operate and scale.

Cross-cluster LLM Training

Current practice of cross-cluster LLM Training

- Inter-clusters network is much slower than intra-cluster.
- The two-stage optimization strategy:
  - Δ Each cluster runs multiple inner training steps with an inner optimizer.
  - Δ An outer optimizer updates and synchronizes models across clusters at a lower frequency.

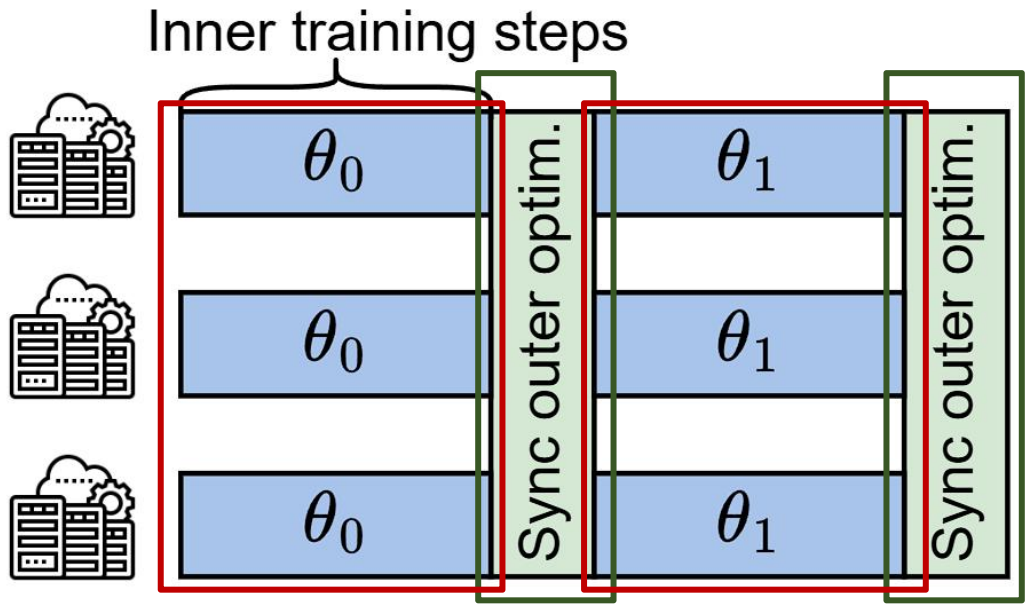


Cross-cluster LLM Training

Current practice of cross-cluster LLM Training

- Inter-clusters network is much slower than intra-cluster.
- The two-stage optimization strategy:
  - Δ Each cluster runs multiple inner training steps with an inner optimizer.
  - Δ An outer optimizer updates and synchronizes models across clusters at a lower frequency.
- Implementation (DiLoCo):
  Decentralized architecture.

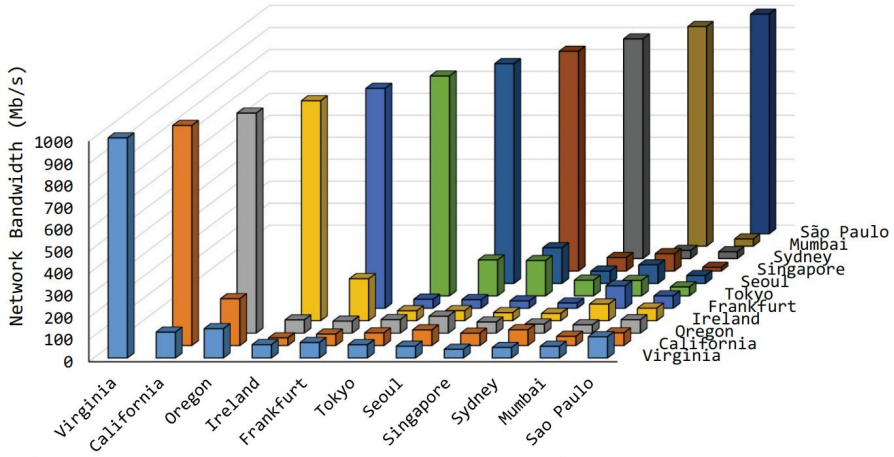
  Every cluster has an outer optimizer replica and synchronizes them with collective communications.



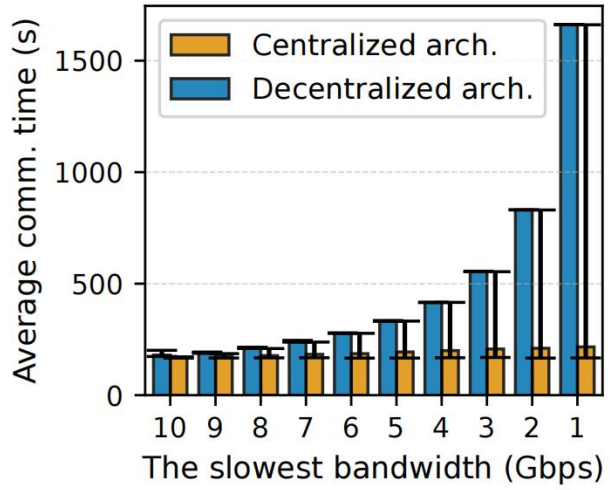
Challenges

Heterogeous inter-cluster network and computation

- Inefficient cross-cluster communication:
  Decentralized communication performs poorly under heterogeneous network bandwidth.



Inter-cluster bandwidth varies



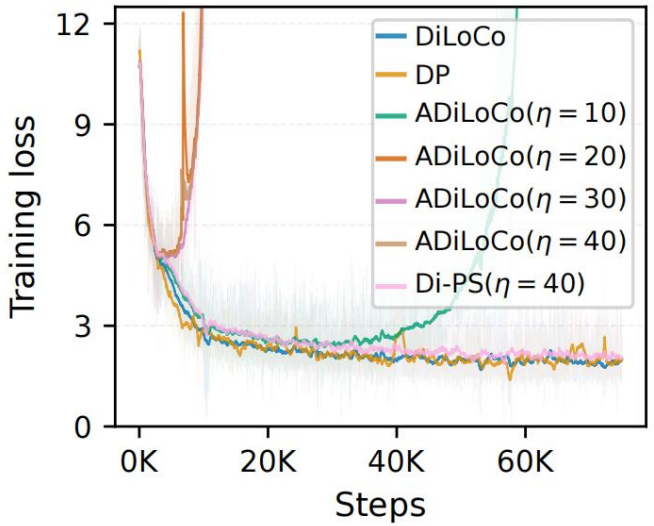
Network heterogeneity slows decentralized communication

Figure sources: https://www.usenix.org/system/files/conference/nsdi17/nsdi17-hsieh.pdf

Challenges

Heterogeous inter-cluster network and computation

- Inefficient cross-cluster communication.
- Unstable asynchronous convergence:
  Heterogeneous computation requires asynchrous training, but naive asynchronous DiLoCo (ADiLoCo) may fail to converge.



Challenges

Heterogeous inter-cluster network and computation

- Inefficient cross-cluster communication.
- Unstable asynchronous convergence.
- Dynamic and unreliable clusters:
  Failure rates and resource availability vary across clusters, and cluster management overhead is high in decentralized implementation.

Cluster	Amount	#NPU	Total PFLOPS (FP16)	#Failures	MTBF (Days)
A	5	1024	378.9	3	55.0
B	1	896	331.5	1	33.0
C	1	1472	329.5	4	7.3
D	1	448	229.4	2	3.0
E	1	2176	479.2	31	0.9

Failure rates and resource availability in a 33-day production training with 9 clusters

Observation and Requirement

Centralized PS for Cross-cluster Training

Challenges

- Inefficient cross-cluster communication.
- Unstable asynchronous convergence.
- Dynamic and unreliable clusters

Opportunities in centralized parameter server (PS) architecture

- Better use of cross-cluster bandwidth.
  The centralized PS can fully utilize high-bandwidth links.
- Global optimizer history.
  The centralized PS can maintain complete historical records.
- Localized cluster failures.
  Cluster failures only need to be handled by the centralized PS.

Observation and Requirement

Centralized PS for Cross-cluster Training

Challenges

- Inefficient cross-cluster communication.
- Unstable asynchronous convergence.
- Dynamic and unreliable clusters

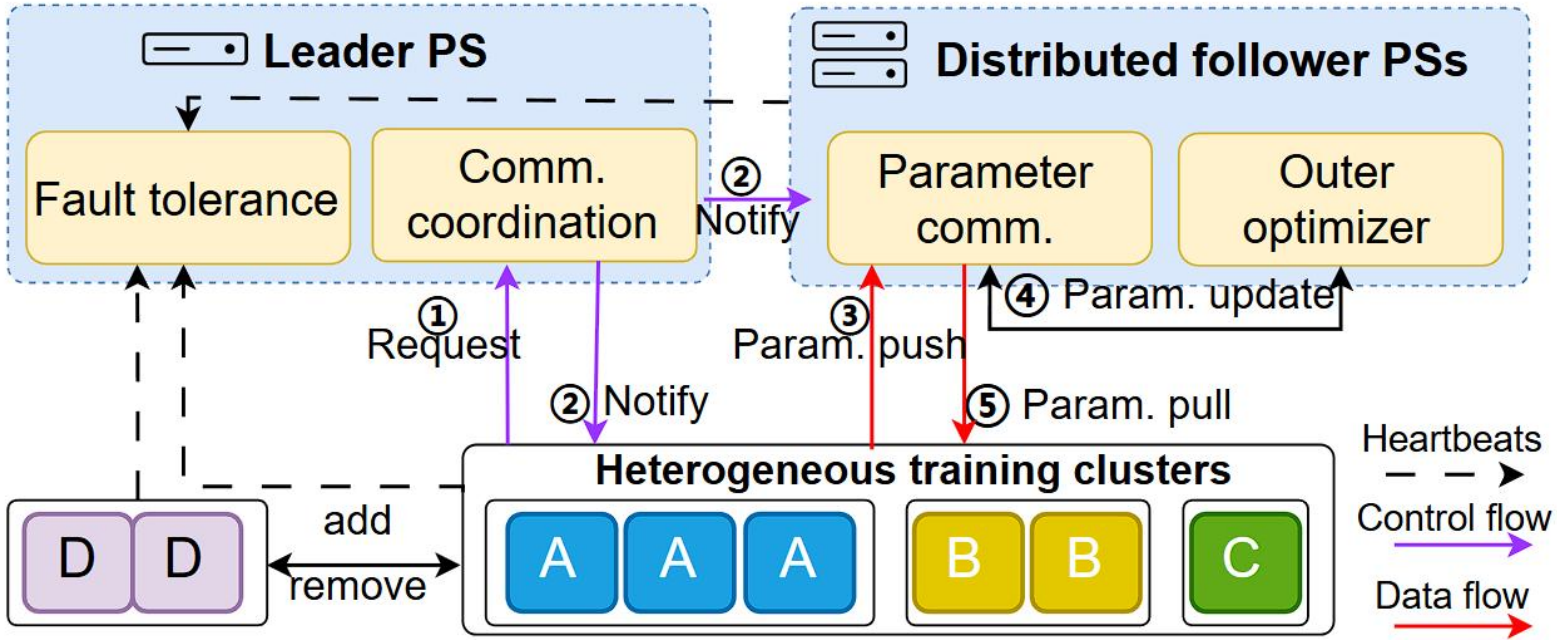
Requirements in centralized PS for cross-cluster training

- **Scalable efficiency.**
  Efficient parameter exchange over heterogeneous cross-cluster links.
- **Stable convergence.**
  Detect stale or abnormals and stabilize asynchronous training.
- **Resilience.**
  Isolate cluster failures, support elastic cluster join/leave.

Di-PS Design

Overview

- A Leader-Follower PS Architecture:
  - Distributed and scalable follower PSs shard the outer optimizer states.
  - The leader PS handles scheduling and management.
  - A dual-workflow design to relieve communication contention.



Di-PS Design

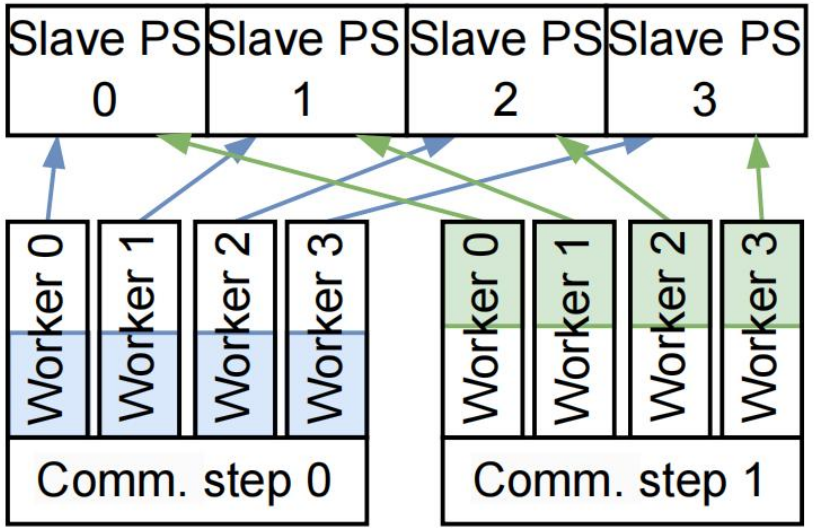
Scalable and efficienct PS for LLM

- Cross-Cluster Communication Scheduling:

  - Follower PSs and training clusters communicate in a many-to-many pattern.

  - Scheduling goals:
    - High utilization of cross-cluster bandwidth.
    - Support communication requests from other clusters during communication.

  - Greedy mapping strategy:
    - Transfers model parameters with high concurrency.
    - Scales well. Schedules can be generated independently for each cluster.

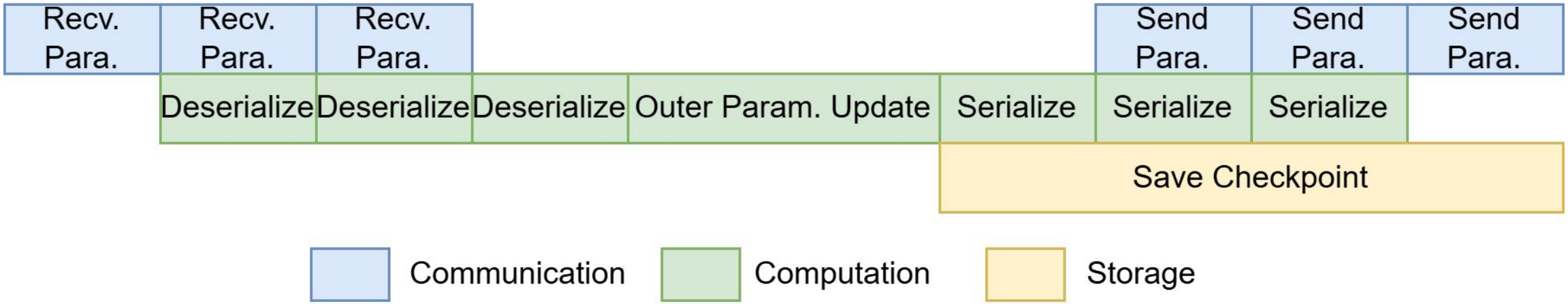


Example of communication schedule

Di-PS Design

Scalable and efficienct PS for LLM

- Operation Overlapping in Follows PSs:
  - Δ Follower PSs perform outer optimization operations.
  - Δ Pipelining communication, computation, and storage operations on follower PSs.



Di-PS Design

Stable Asynchronous Training

- Pseudo-Gradient Correction
  - Detect abnormal gradients from update history to improve asynchronous training stability.
  - Correction strategy:
    1. Compute the norm of each cluster's pseudo-gradient.
    2. Score gradient quality using the norm and update history, and filter out low-quality gradients.
    3. Average accepted gradients based on consumed training data.

$$G_t^j = \sum_{i=1}^{n} \|\Delta_t^{i,j}\|_2,$$

$$E_t = \frac{G_t - \mu_t}{\sigma_t},$$

$$\mu_{t+1} = \alpha G_t + (1-\alpha)\mu_t,$$

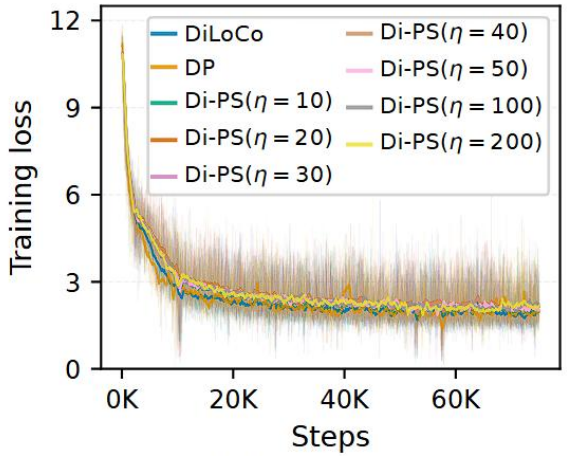
$$\sigma_{t+1} = \sqrt{(1-\alpha)(\sigma_t)^2 + \alpha(G_t - \mu_{t+1})^2},$$

$$\delta_t^i = \frac{\sum_{j \in c} T_j \Delta_t^{i,j}}{\sum_{j \in c} T_j},$$

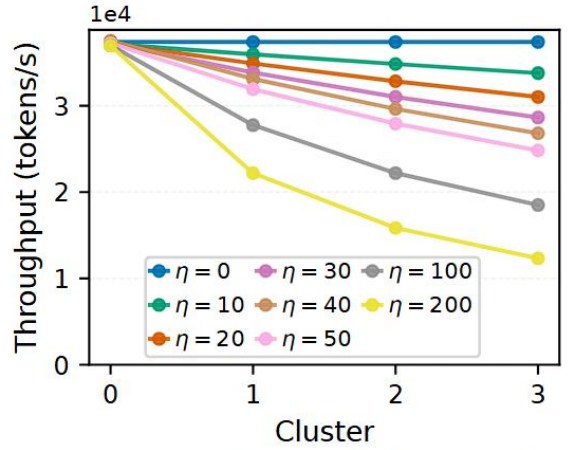
Di-PS Design

Stable Asynchronous Training

- Effect of Pseudo-Gradient Correction
  - **Achieves convergence** comparable to synchronous methods under different heterogeneous settings.
  - The trained model shows nearly identical downstream task performance to synchronous methods.



(a) The training loss of asynchronous cross-cluster training with Di-PS.



(b) The detailed training performance distribution of different heterogeneity values η.

Dataset	Metric	DP	DiLoCo	Di-PS						
				η =10	η =20	η =30	η =40	η =50	η =100	η =200
BBH	acc	31.11	29.52	29.23	29.46	30.09	30.47	29.45	28.94	29.67
MMLU	acc	24.38	24.16	24.18	24.94	24.61	24.21	24.66	24.76	26.34
DROP	acc	31.77	27.88	31.45	32.75	31.02	31.22	31.53	31.42	29.94

(c) Model evaluation results.

Di-PS Design

Resilience and Fault-tolerance Mechanism

- Failure Analysis
  - Training resources are dynamic, with frequent scale-in and scale-out events.
  - Intra-cluster failures can be handled by the cluster management system.
  - Need incorporating resilience against failures within PS.

Category	Reasons	Amount	Avg. Recovery Time (min)
Hardware	Network Interface	2	95.71
	Faulty NPUs	2	109.54
	HBM Overflow	5	88.21
	Storage Device	1	10.63
	Backplane	1	30.38
Software	Collective Failure	17	41.78
	Framework Issue	3	46.46
	User Code Bug	3	68.60
	Configuration Issue	2	92.73
	Management System	5	159.54
Di-PS	Leader PS Failure	1	44.43
	Follower PS Issue	3	3.05

- Heartbeat-Based Elasticity and Fault Tolerance
  - The leader PS monitors training clusters and follower PSs through heartbeats.
  - Failed clusters are removed from outer optimization; other clusters continue training.
  - Failed follower PSs are automatically restarted.

Evaluation

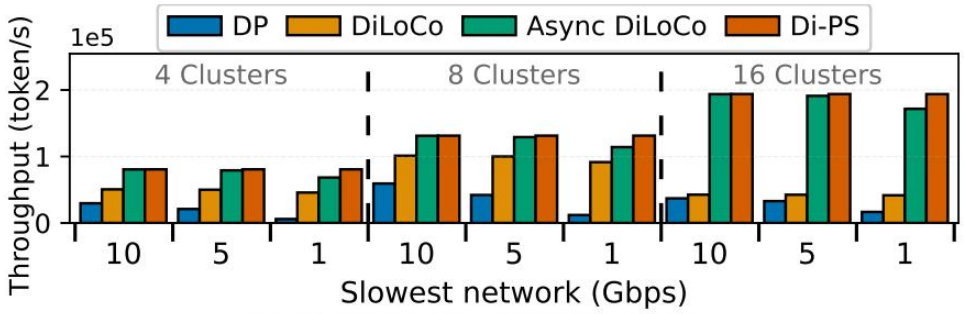
Experimental Settings

- Two cross-cluster environments:
  - 16 real heterogeneous clusters with up to 8.18x training performance gap;
  - 16 emulated clusters with injected delays to create 0-100% performance variation.
- Heterogeneous network setting:
  - One cluster has 10, 5, or 1 Gbps cross-cluster bandwidth, while others remain at 10 Gbps.
- Models: LLaMA3.2-1B and Qwen3-14B.
- Baselines：
  - DP: Synchronizes model gradients across all clusters at every training iteration.
  - DiLoCo: Decentralized synchronous two-stage optimization baseline.
  - Async DiLoCo: DiLoCo with asynchronous outer updates.

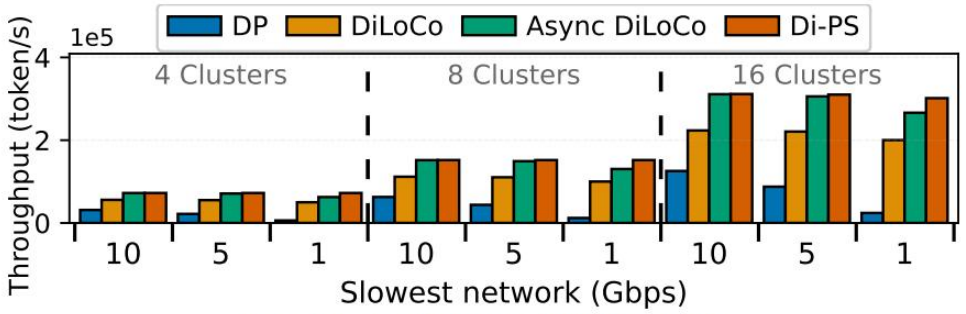
Evaluation

End-to-end Performance Comparison

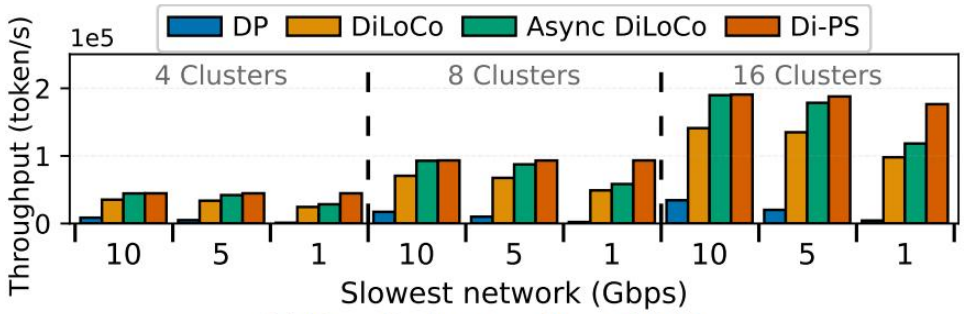
- Di-PS improves training performance by 1.27-4.67X over convergent DiLoCo and 1.00-1.60X over non-convergent Async DiLoCo.
- Model evaluation show that Di-PS achieves comparable quailty to synchronous methods.



(a) Real clusters, LLaMA3.1-1B.



(b) Emu-S clusters, LLaMA3.1-1B.



(c) Emu-L clusters, Qwen3-14B.

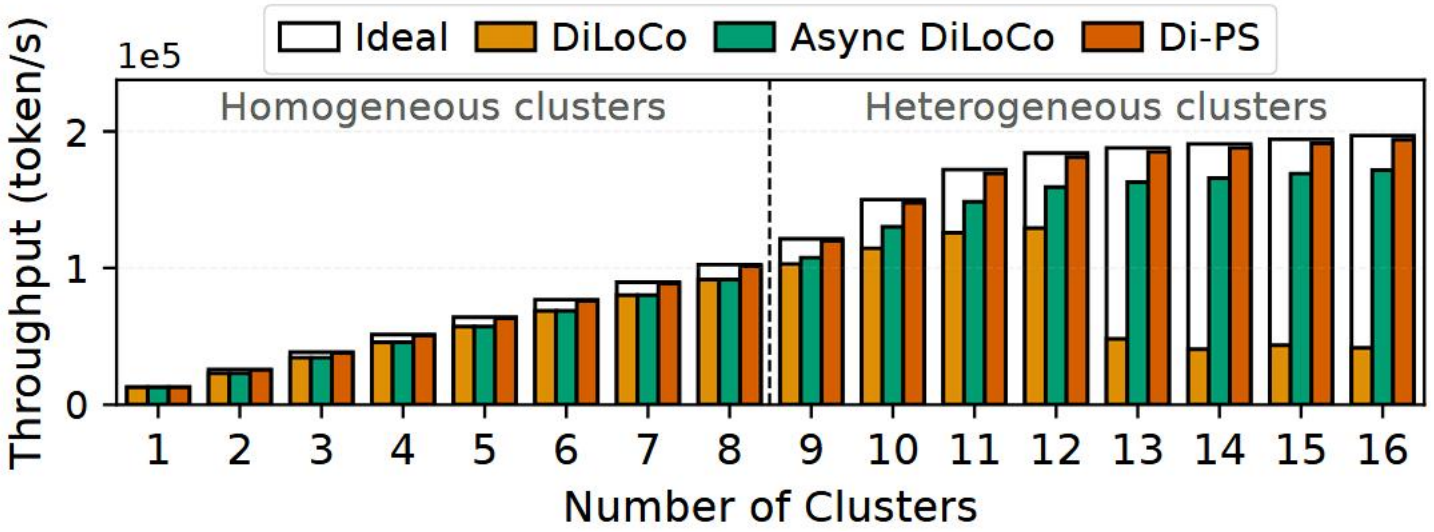
Dataset	Metric	LLaMA3.1-1B			Qwen3-14B		
		DP	DiLoCo	Di-PS	DP	DiLoCo	Di-PS
BBH	acc	31.24	31.13	31.35	69.44	68.41	72.61
MMLU	ppl	27.04	26.02	26.69	77.08	73.37	76.10
DROP	acc	31.77	31.11	31.42	70.34	64.58	68.17

(d) Evaluation results on models trained with 16 clusters.

Evaluation

Scalability on the Number of Training Cluster

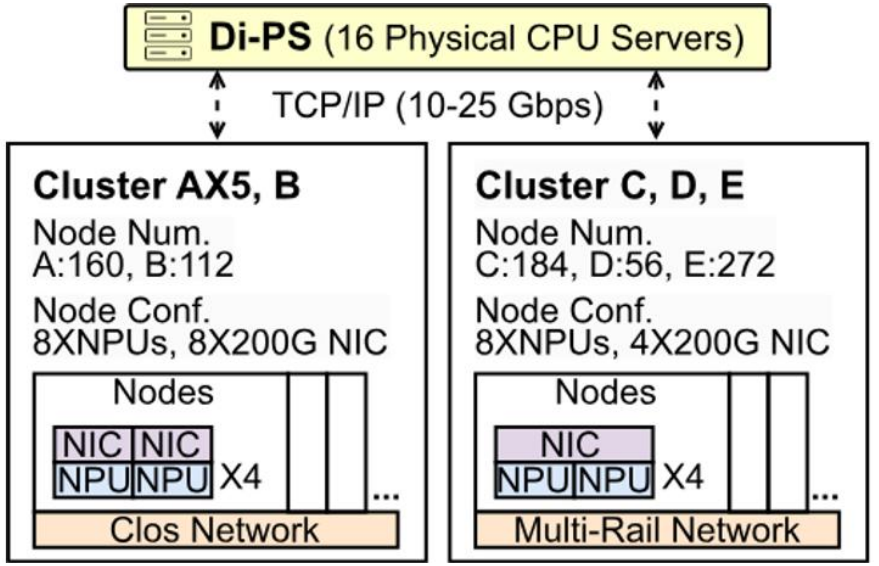
- Clusters are scaled first in homogeneous settings, then in heterogeneous settings.
- Di-PS reaches 98.3-98.8% of ideal performance..



Di-PS in Production Training

Production Training Setup

- Trained a **100B** LLaMA-based LLM.
- Used up to 9 heterogeneous training clusters, peaking at **10,112** compute devices in total.
- Deployed 16 follower PSs, each running on a separate CPU node.
- Training lasted **33** days and processed **2.3T** tokens.



Topology of the training clusters

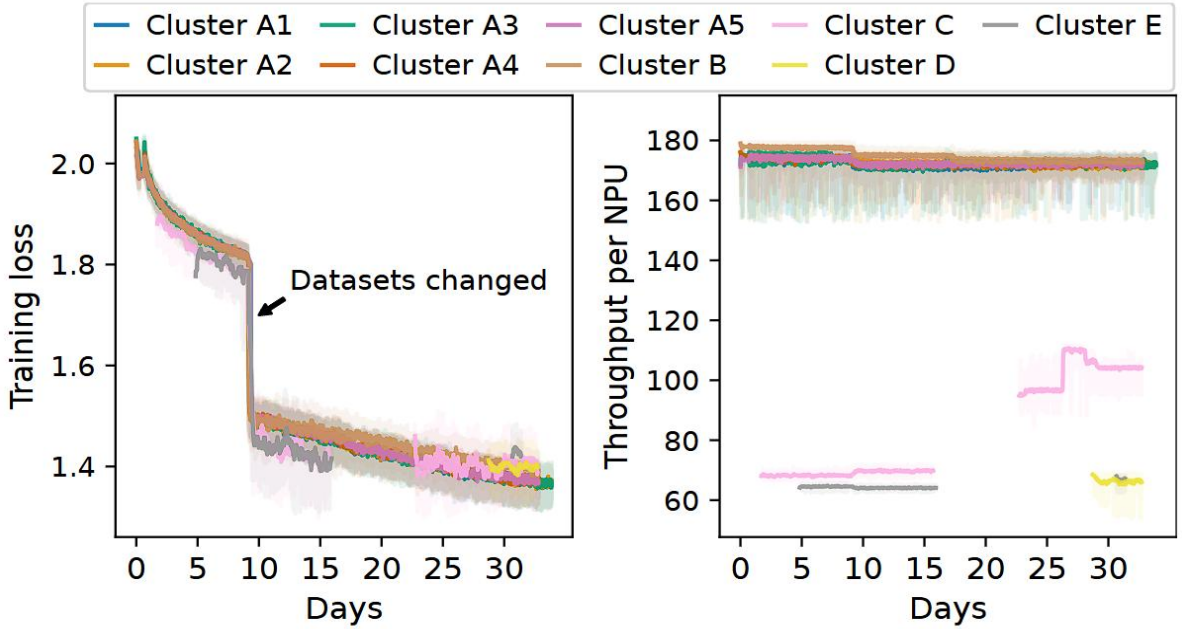
Cluster	Amount	#NPU	Total PFLOPS (FP16)
A	5	1024	378.9
B	1	896	331.5
C	1	1472	329.5
D	1	448	229.4
E	1	2176	479.2

Peak device count and BF16 capacity

Di-PS in Production Training

Model Training Results

- Training loss decreases steadily.
- Compute device performance remains stable.
- The trained model achieves expected performance.



Training loss and training performance

Dataset Metric	BBH acc	MMLU acc	CMMLU acc	DROP acc	MBPP score	GSM8K acc	HellaSwag acc
Ours	**83.4**	81.4	83.5	80.2	72.0	84.5	**93.2**
LLaMA3.1-70B	81.6	79.3	68.8	79.6	66.2	83.6	79.9
Qwen2.5-72B	79.8	**85.0**	**89.5**	80.6	**72.6**	**88.3**	84.8
LLaMA3.1-405B	82.9	84.4	73.7	**86.0**	68.4	83.5	89.2

Model evaluation result

Summary and Contributions

Problem

- Cross-cluster LLM training faces heterogeneous bandwidth, unstable asynchronous convergence, and frequent cluster failures.

Key Idea

- A centralized PS to coordinate efficient, stable, and fault-tolerant cross-cluster training.

Techniques in Di-PS

- A scalable and efficienct leader-follower PS architecture for LLM.
- Pseudo-gradient correction for stable asynchronous training.
- Heartbeat-based elasticity and fault tolerance.

Results

- Di-PS improves training throughput while preserving model quality.
- Di-PS scales to 10,112 devices and trains a 100B-parameter LLaMA-based model for 33 days.

Q & A

Thanks!