DORA: A Scalable Asynchronous Reinforcement Learning System for Language Model Training

Tianhao Hu , Xiangcheng Liu, Yuchun Miao, Youshao Xiao, Hongyu Zang, Yang Zheng, Xuan Huang, Jinrui Ding, Yufei Zhang, Yu Yang, YiKai Zhang, Yueqing Sun, Chengcheng Han, Xiandi Ma, Qi Gu, Wei Wang, Yerui Sun, Yuchen Xie, Xunliang Cai

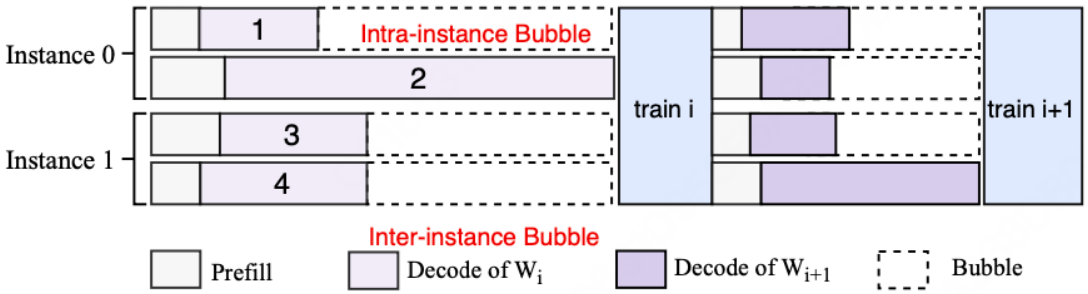
LongCat

Project Leader: Youshao Xiao

Team Leader: Xiandi Ma, Qi Gu, Wei Wang

NEURAL INFORMATION PROCESSING SYSTEMS

Practical Problems in Long-context RL



Long-tail rollout bottleneck! Very Low Efficiency!

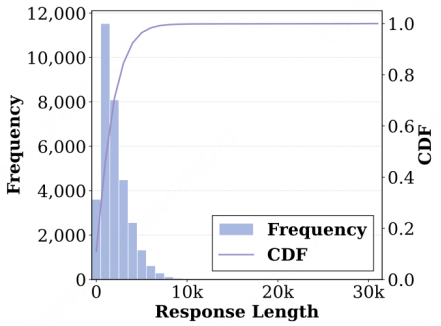


Figure 1: Response length distribution on the DAPO-Math-17K dataset.

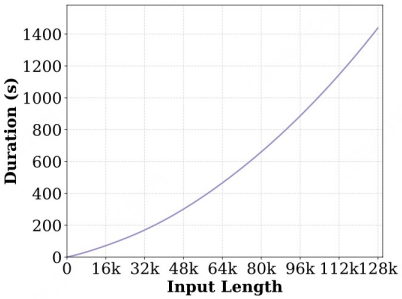
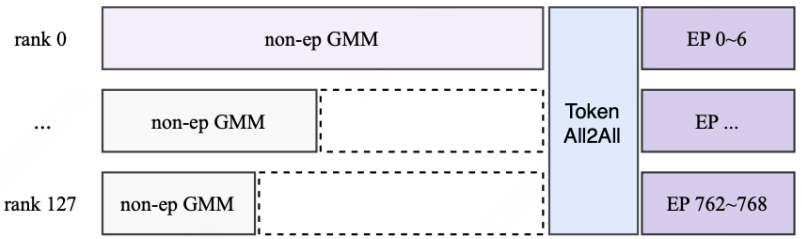


Figure 3: Prefill duration of a ~500B MoE model with expert parallelism size 128 on non-CUDA accelerators.

Mainstream methods

- **Synchronous:** Slowest trajectory barriers the whole step.
- **Partial Rollout:** Removes the rollout barrier by recomputing the long traj under latest policy model — at the cost of multiple policy versions within a trajectory, and the Re-prefill overhead.

The system overhead is amplified by MoE structure and on our domestic accelerators in production



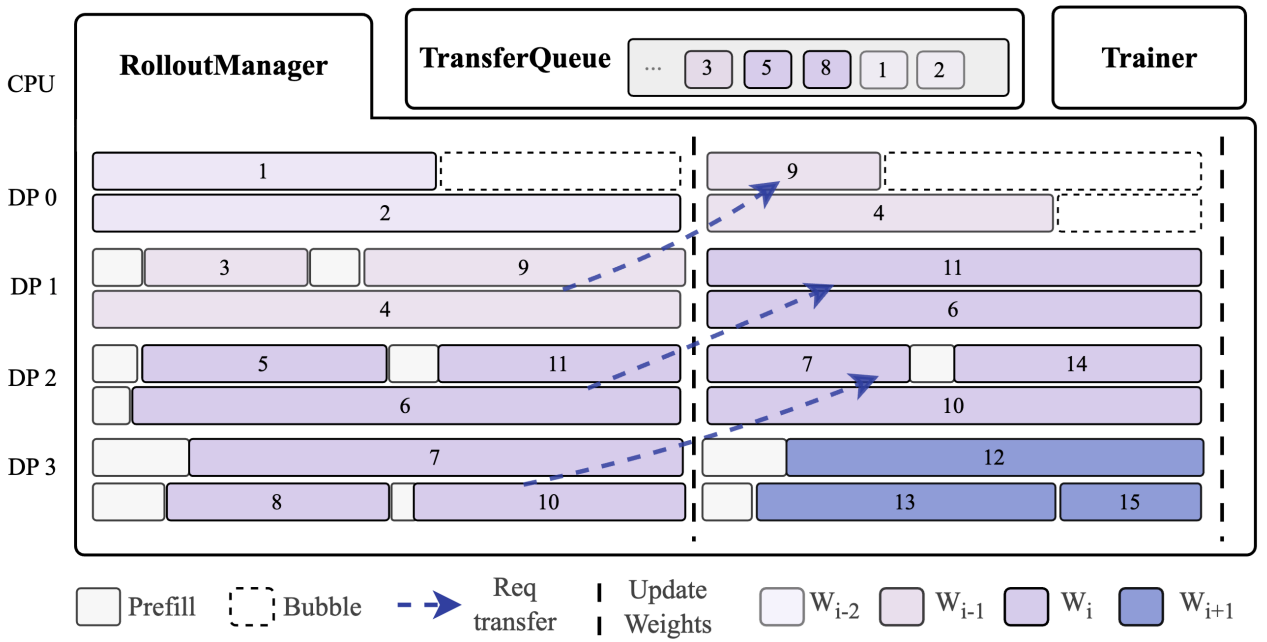
Key Insights

Multi-version streaming keeps long trajectories alive under the original policy while the completed trajectories **stream** into training as completed.

Example- GRPO with latest policy version 3 and group 2

Partial rollout:
Traj 1: 11111 | 222 | 33333333
Traj2: 333333

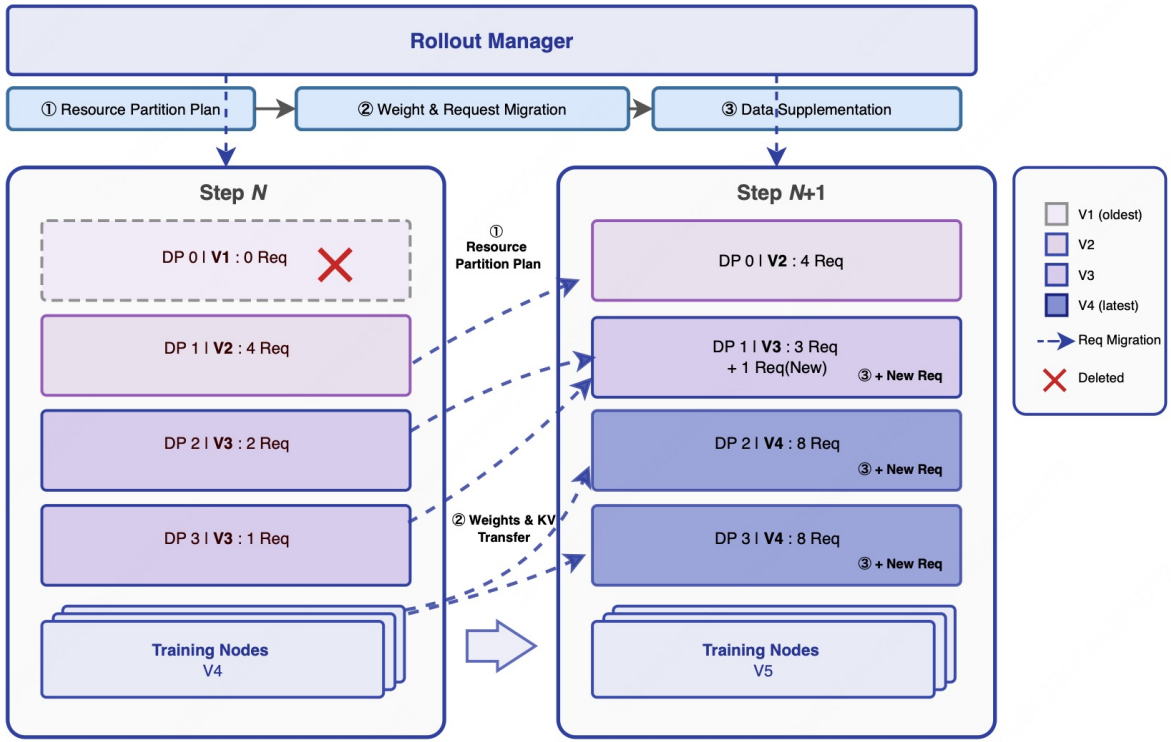
DORA:
Traj 1: 1111111111111111111
Traj 2: 11111111



Algorithm bonus: No extra algorithmic correction for version coexistence within a trajectory, preserving theoretical convergence guarantees of traditional asynchronous RL.

System bonus: Higher efficiency in long-tailed scenarios, with nearly zero re-prefill overhead.

System Overview



Dynamic Resource Orchestration

- Make the partition plan of the inference instances according to the tasks workloads by the model version
- Make the request migration using the KV-Cache reuse
- Staleness-ware data supplementation

KV-Cache Reuse

- **Locality-ware scheduling.** Prioritize reassigning requests back to the original DP rank to avoid recompute.
- **Hierarchical memory management.** Use the host memory to temporarily offload/onload the KV-cache in the long-context training.

Convergence Guarantees

LongCat

Staleness Control

- **Sliding-window Staleness control**: Active versions are managed through a sliding window to control the maximum staleness.
- **Reproducibility:** DORA's multi-version asynchrony can be faithfully emulated in a synchronous trainer using a replay buffer, enabling controlled studies of policy staleness. **Matters in large-scale RL!!!**

Policy improvement guarantee

$$\underbrace{J(\pi_\theta) - \sum_i \alpha_i J(\pi_{w_i})}_{\text{Return improvement over the weighted average of past policies}} \geq \underbrace{\sum_i \alpha_i L_i(\pi_\theta)}_{\text{Surrogate objective improvement}} - \underbrace{\frac{2\gamma K \delta}{(1-\gamma)^2} \max_i C_{\theta,i}}_{\text{Upper bound on the distribution mismatch penalty}}$$

K is the staleness bound, δ is per-update policy drift upper-bound, and $C_{\theta,i}$ bounds the magnitude of expected advantage.

DORA controls staleness to bound distribution mismatch.

Experiment Results

LongCat

Comparison across the training paradigms using the **in-house framework**

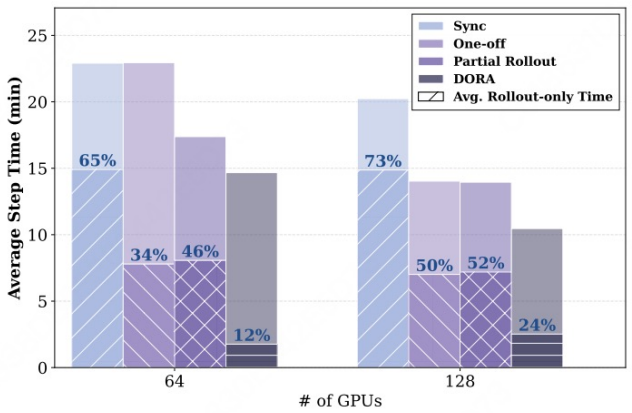


Figure 8: Average RL step time for Dense-32B across training paradigms at 64 and 128 GPUs.

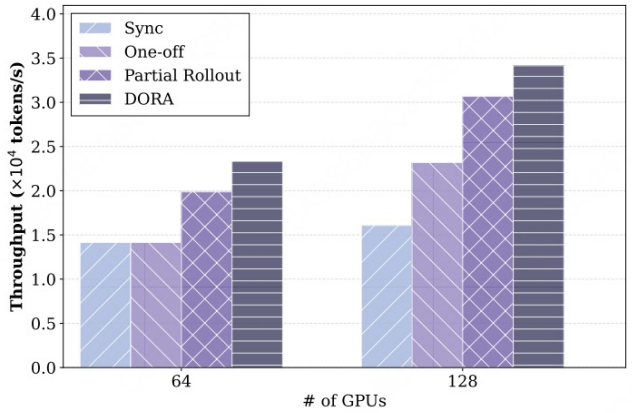


Figure 9: End-to-end throughput for Dense-32B across paradigms at 64 and 128 GPUs.

Comparison with the **open-source framework**

Table 2: Throughput comparison on Qwen2.5-Dense-32B under the same experimental configuration.

Framework	Paradigm	Throughput	DORA Speedup
Slime (v0.3.0)	Synchronous	13K tok/s	2.6×
Slime (v0.3.0)	Partial rollout	19.3K tok/s	1.76×
DORA (ours)	**Multi-version streaming**	**34K tok/s**	**– (ref.)**

Experiment Results

Model Convergence

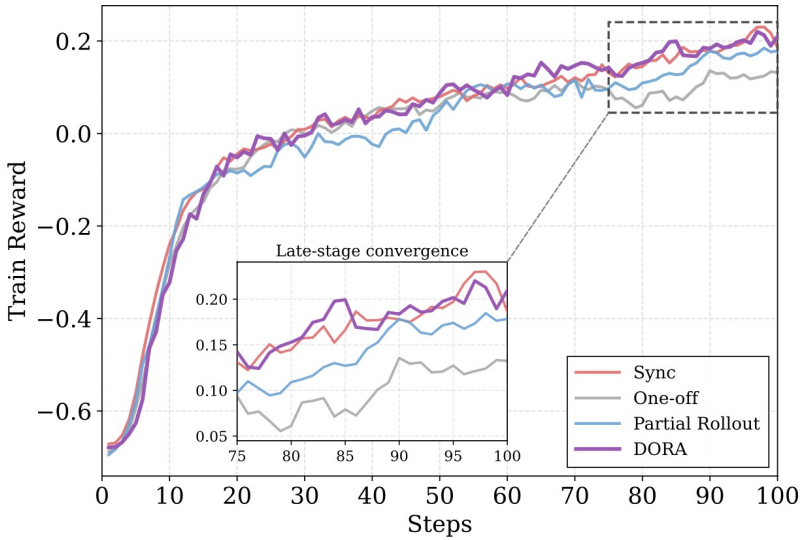


Figure 10: Training rewards for Dense-32B across training paradigms at 128 GPUs.

Production Results

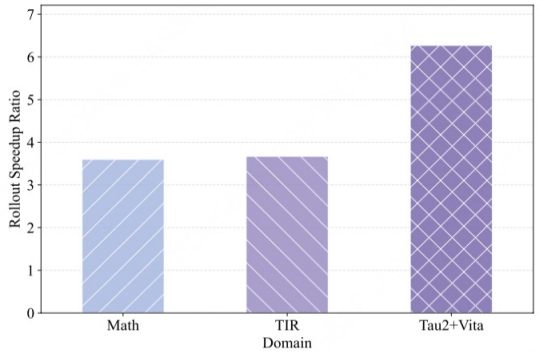


Figure 13: Production rollout time on a 560B MoE model (64K-token limit).

In the support the release of Longcat-Flash-Thinking and Longcat-Flash-Thinking-2601,

DORA is **3.6x to 6.2x** speedup over well-tuned synchronous training in Production and **2-4x speedup** in the End2end time using thousands of Accelerators.

Table 1: Task performance of DORA-based RL and frontier models on Tau2-bench.

Method	Tau2-Retail	Tau2-Airline	Tau2-Telecom
SFT (cold start)	70.7	72.75	47.6
+ DORA RL	**88.6**	**76.5**	**99.3**
Claude Opus 4.5 Thinking	88.9	–	98.2
GPT-5.2 Thinking xhigh	82.0	–	98.7