MoC-System: Efficient Fault Tolerance for Sparse Mixture-of-Experts Model Training

Weilin Cai, Le Qin, Jiayi Huang
{wcai738, lqin674}@connect.hkust-gz.edu.cn, hjy@hkust-gz.edu.cn
The Hong Kong University of Science and Technology (Guangzhou)

ASPLOS 2025
Wed. April 2
17:30-17:50 (6-C)

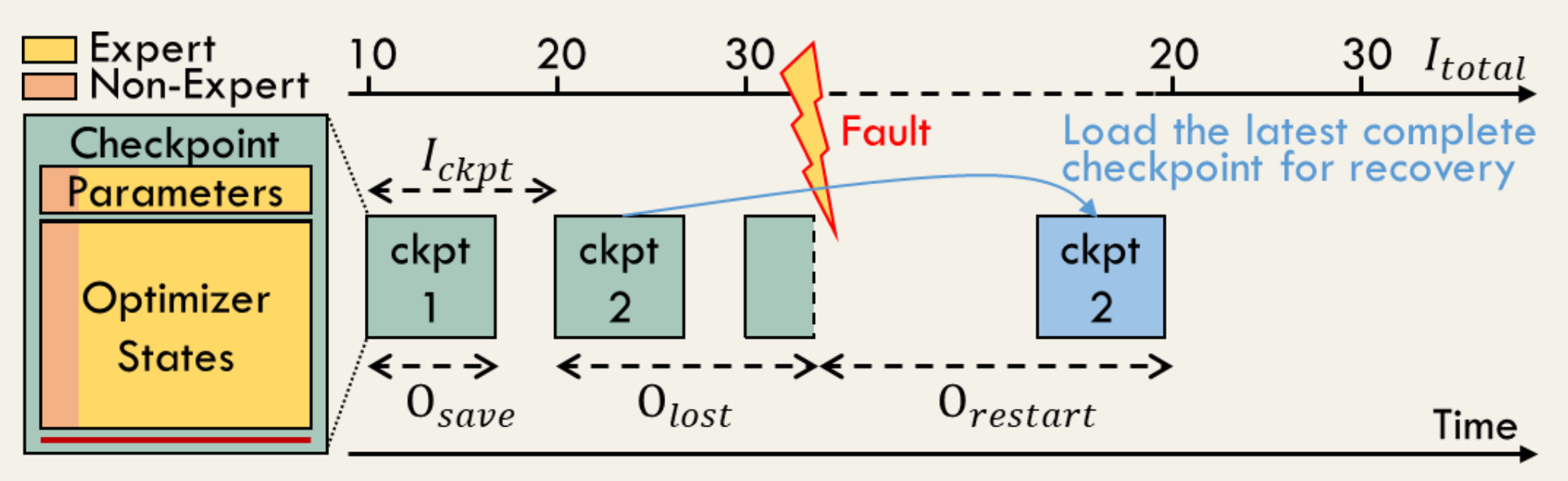
Introduction

❖ **Background:** As large language models continue to scale up, distributed training systems have expanded beyond 10k nodes, intensifying the importance of fault tolerance [1]. Although numerous studies have effectively addressed fault tolerance for dense (non-MoE) models through periodical checkpoints, the distinctive characteristics of MoE models necessitate specialized strategies to assure their reliable and efficient fault-tolerant training [2].

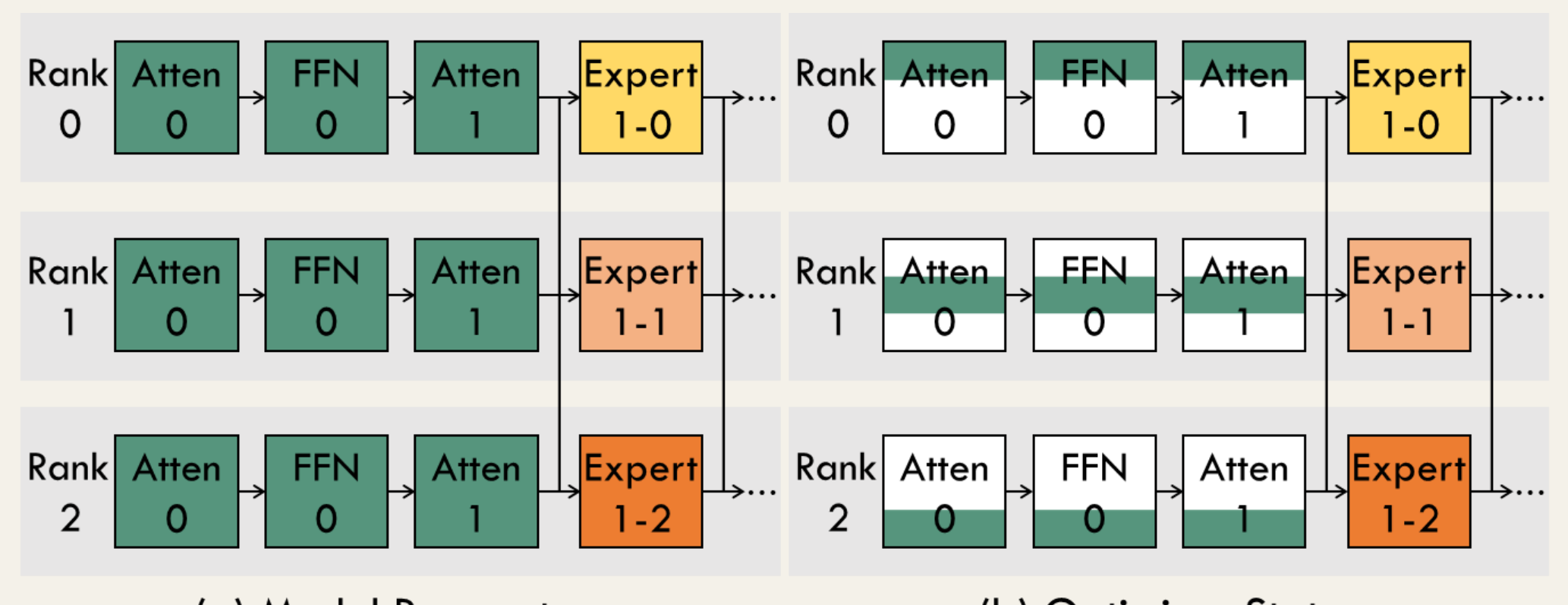
■ **Unprecedented Model Size:** It results in the substantial increase in checkpoint size, which poses a storage burden that distributed filesystems struggle to handle efficiently.

■ **Sparsely-Activated Expert Computation:** The enlarged checkpointing duration cannot be fully overlapped with the training process, as the computational workload does not scale proportionally with the enlarged model size.

• **Checkpoint in LLMs:**

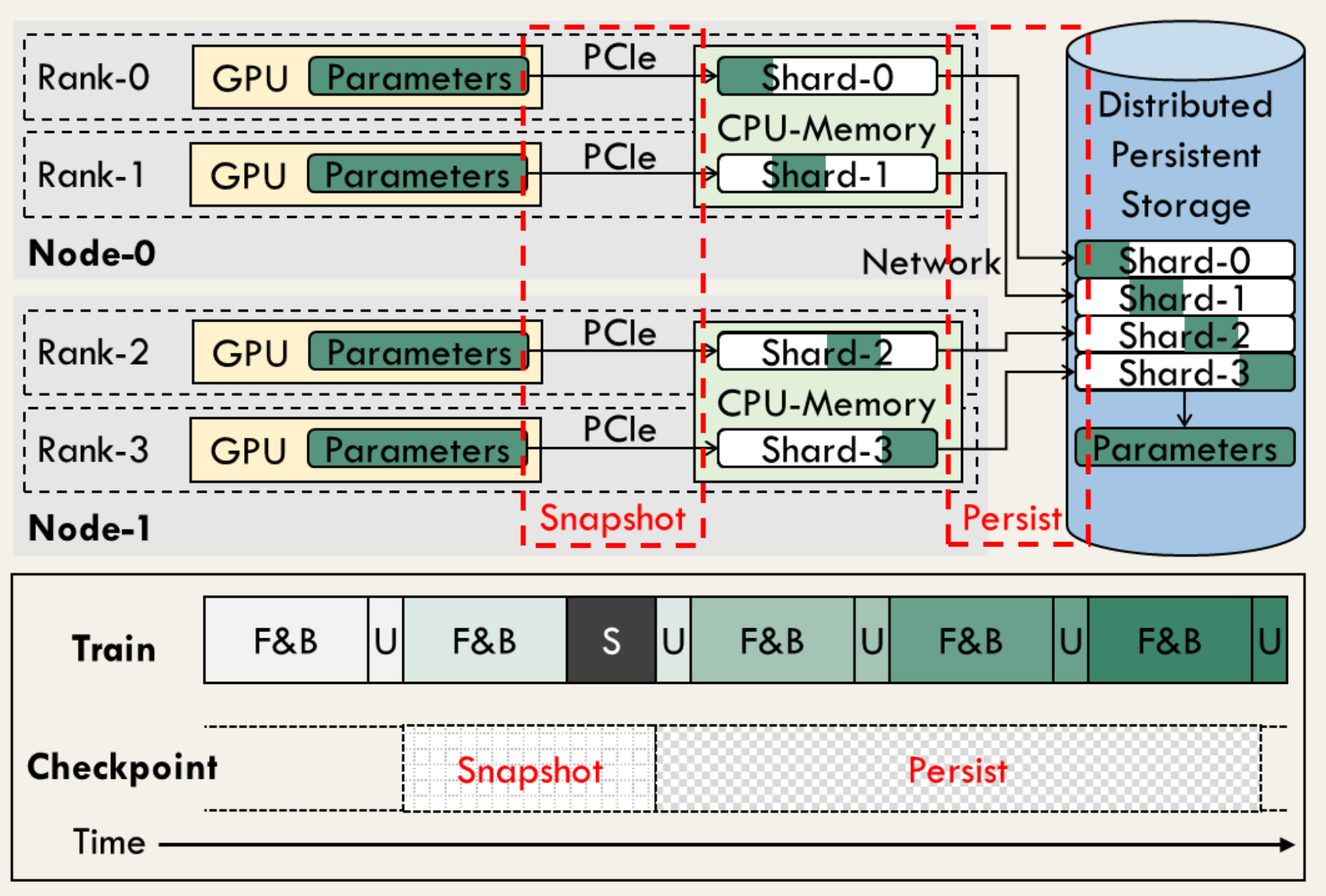


• **Model States Distributed with ZeRO-2 DP + EP:**



(a) Model Parameters (b) Optimizer States

• **Checkpointing Workflow:**



❖ **Inspiration:** Expert parameters are less sensitive to a limited number of training updates compared to non-expert parameters [3].

❖ **Our Contributions:** We propose the Mixture-of-Checkpoint System (MoC-System) to orchestrate the vast array of checkpoint shards produced in distributed training systems.

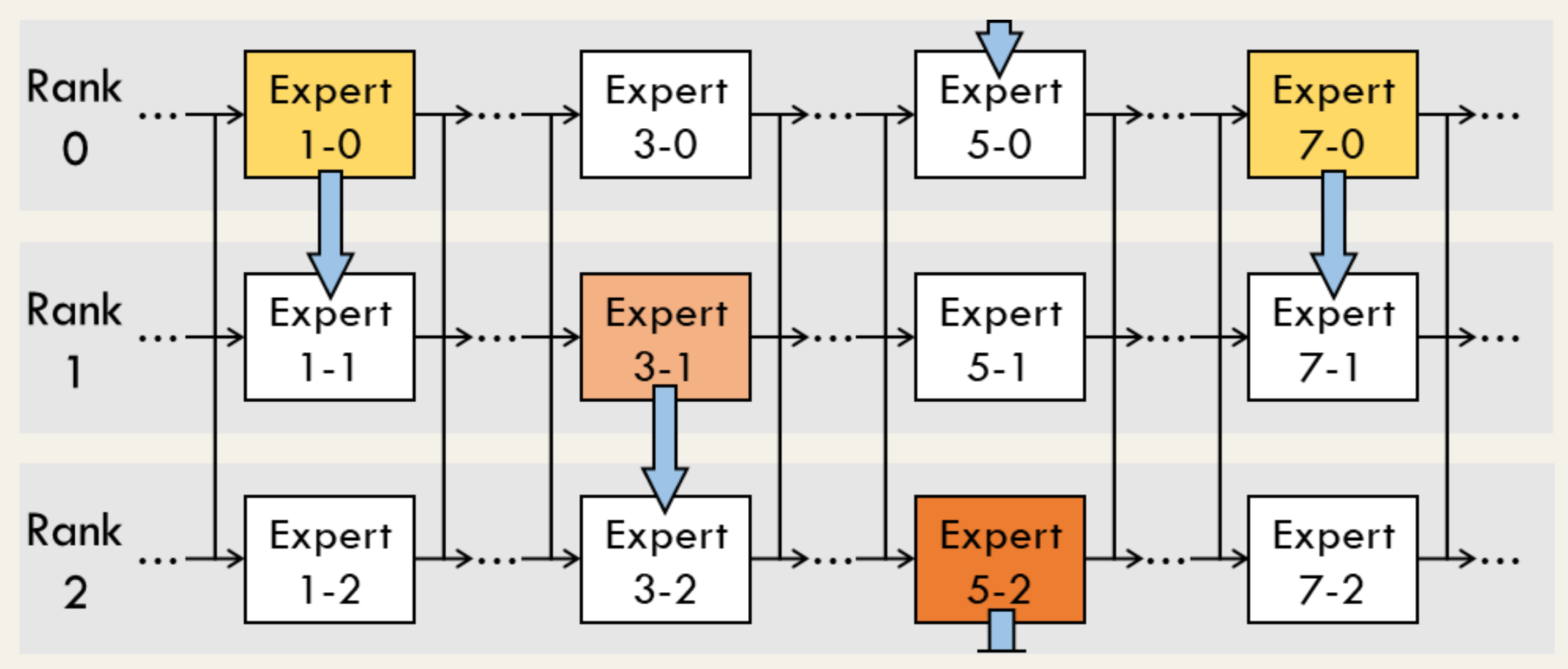
■ **Partial Experts Checkpointing:** MoC-System features a novel Partial Experts Checkpointing (PEC) mechanism, an algorithm-system co-design that strategically saves a selected subset of experts, effectively reducing the MoE checkpoint size to levels comparable with dense models.

■ **Fully Sharded Checkpointing:** Incorporating hybrid parallel strategies, MoC-System involves fully sharded checkpointing strategies to evenly distribute the workload across distributed ranks.

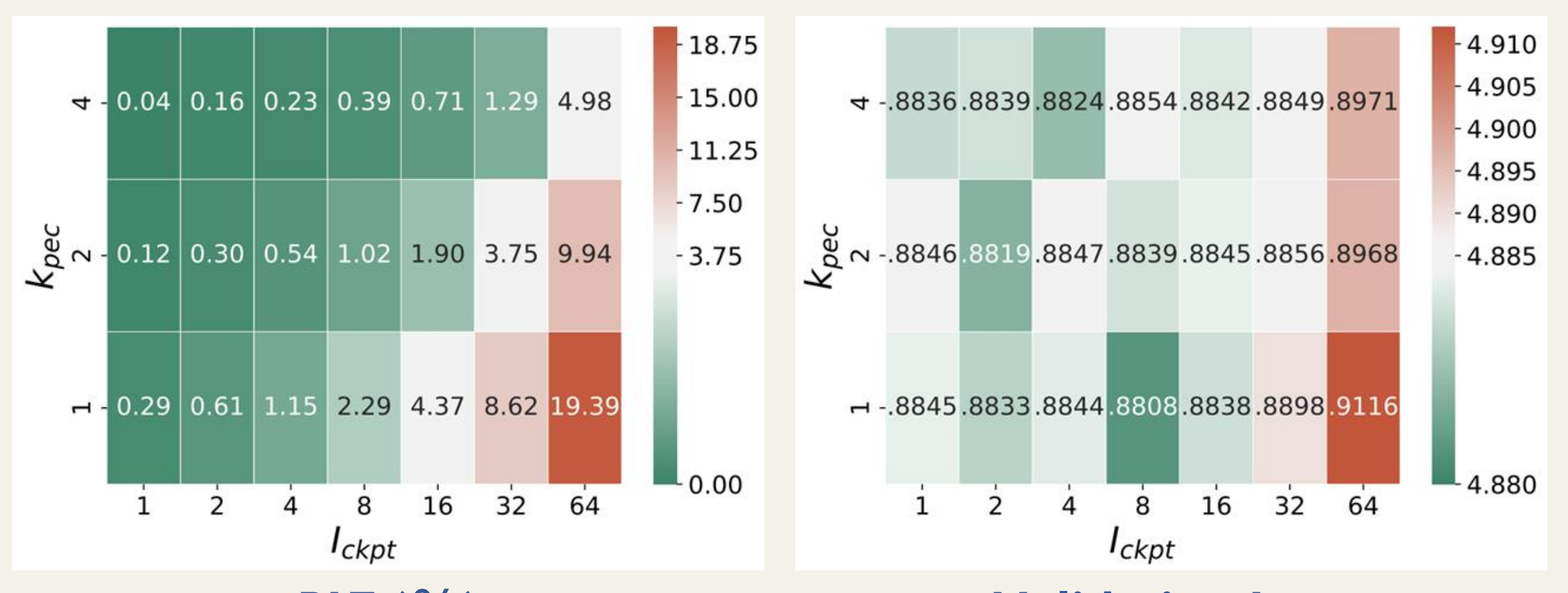
■ **Two-Level Checkpointing Management:** Furthermore, MoC-System introduces a two-level checkpointing management method that asynchronously handles in-memory snapshots and persistence processes.

Partial Experts Checkpointing

In light of the substantial increase in checkpoint size predominantly attributed to the multiplicity of FFN experts within the MoE model, we introduce the concept of Partial Experts Checkpointing (PEC) to significantly downsize the checkpoint. PEC selectively saves K_{pec} experts per MoE layer during each checkpointing, while fully saving the non-expert parameters of the model.



■ **Impact on Model Accuracy:** To quantitatively assess the potential impact on accuracy attributed to PEC, we introduce a novel metric, the Proportion of Lost Tokens (PLT). The PLT metric is designed to quantify the average proportion of tokens lost across all the MoE layers throughout the training. The following figures show that PEC can minimize checkpoint size without harming model accuracy in the case of limited PLT.

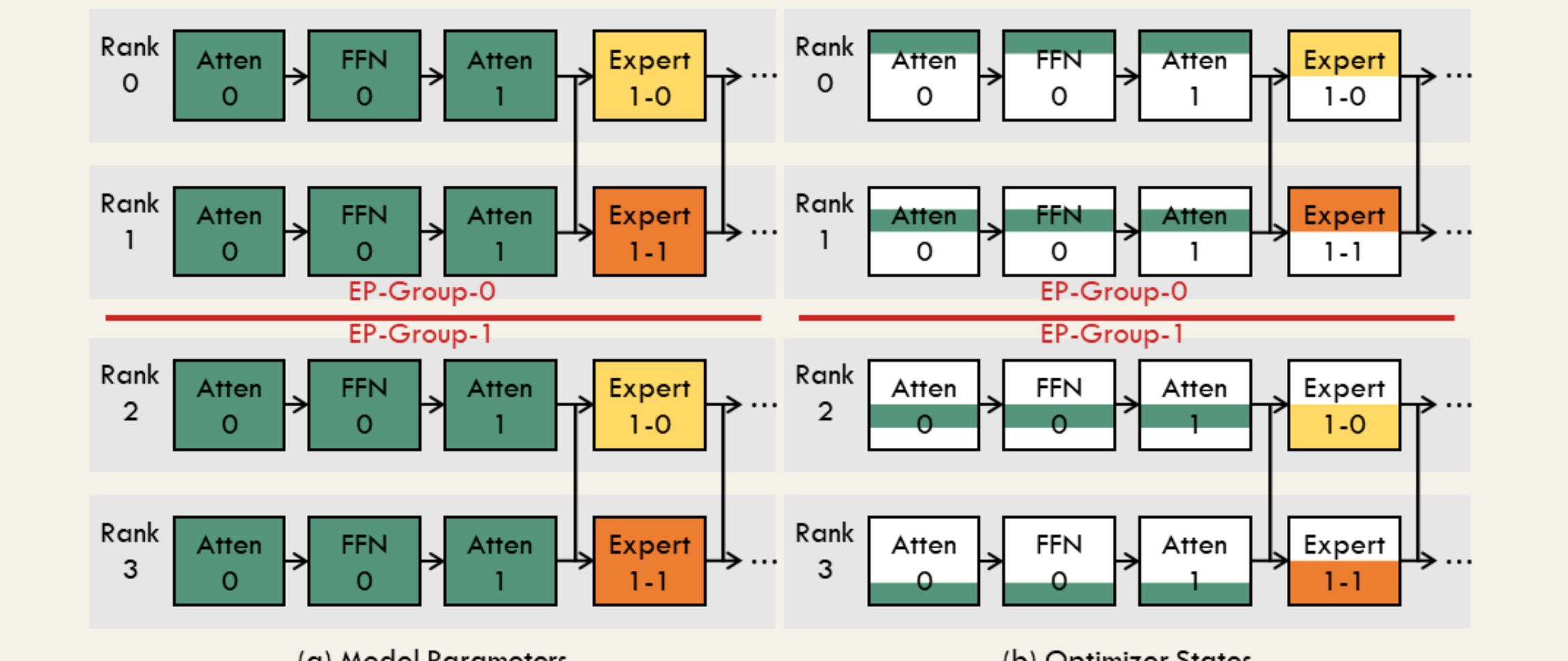


PLT (%) Validation Loss

Fully Sharded Checkpointing

Existing distributed training frameworks lack an efficient data-parallel sharding strategy for MoE model training. In contrast, we implement fully sharded checkpointing for MoE model training and further introduce an adaptive sharding strategy with our PEC mechanism, outperforming the commonly used equal sharding strategy.

■ **Replicated Model Parameters across Ranks:**

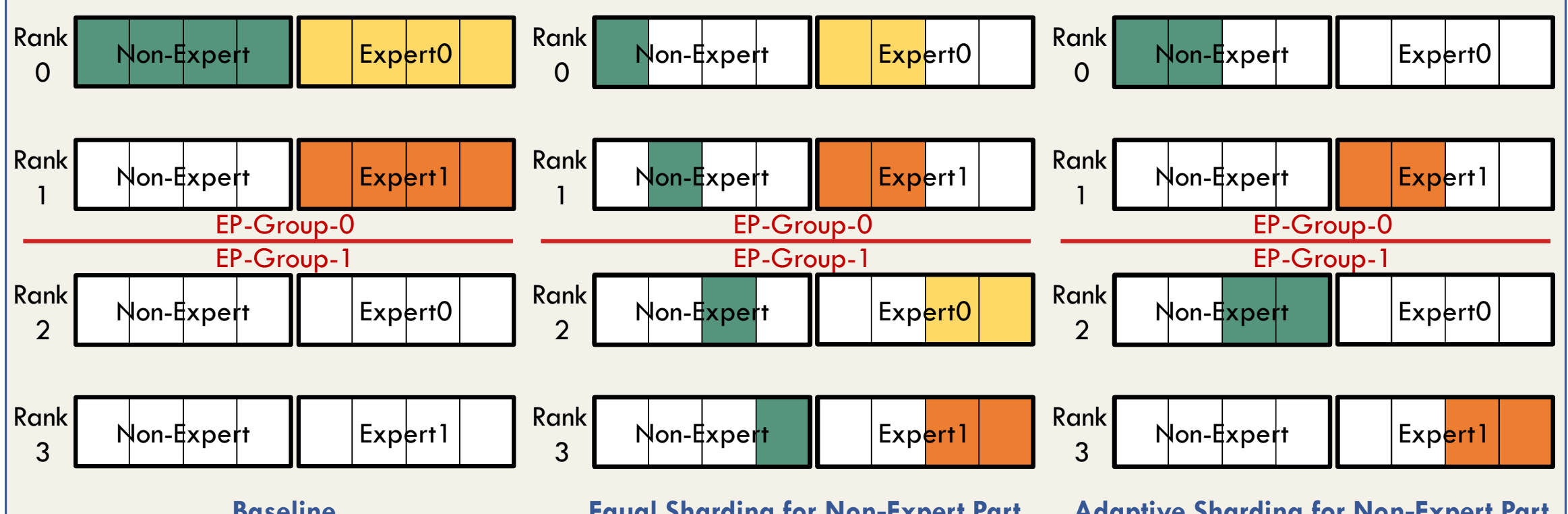


(a) Model Parameters (b) Optimizer States

☐ **Equal Sharding for Expert Part:** Employs each expert as the smallest unit for distribution across various EP groups.

☐ **Equal Sharding for Non-Expert Part:** Evenly distribute the workload of checkpointing non-expert layers across all DP ranks.

☐ **Adaptive Sharding for Non-Expert Part:** PEC may lead to an imbalanced checkpointing workload for the expert part. To leverage the spare capacity across ranks, we introduce the adaptive sharding to allocates non-expert parts based on the selection pattern of PEC.

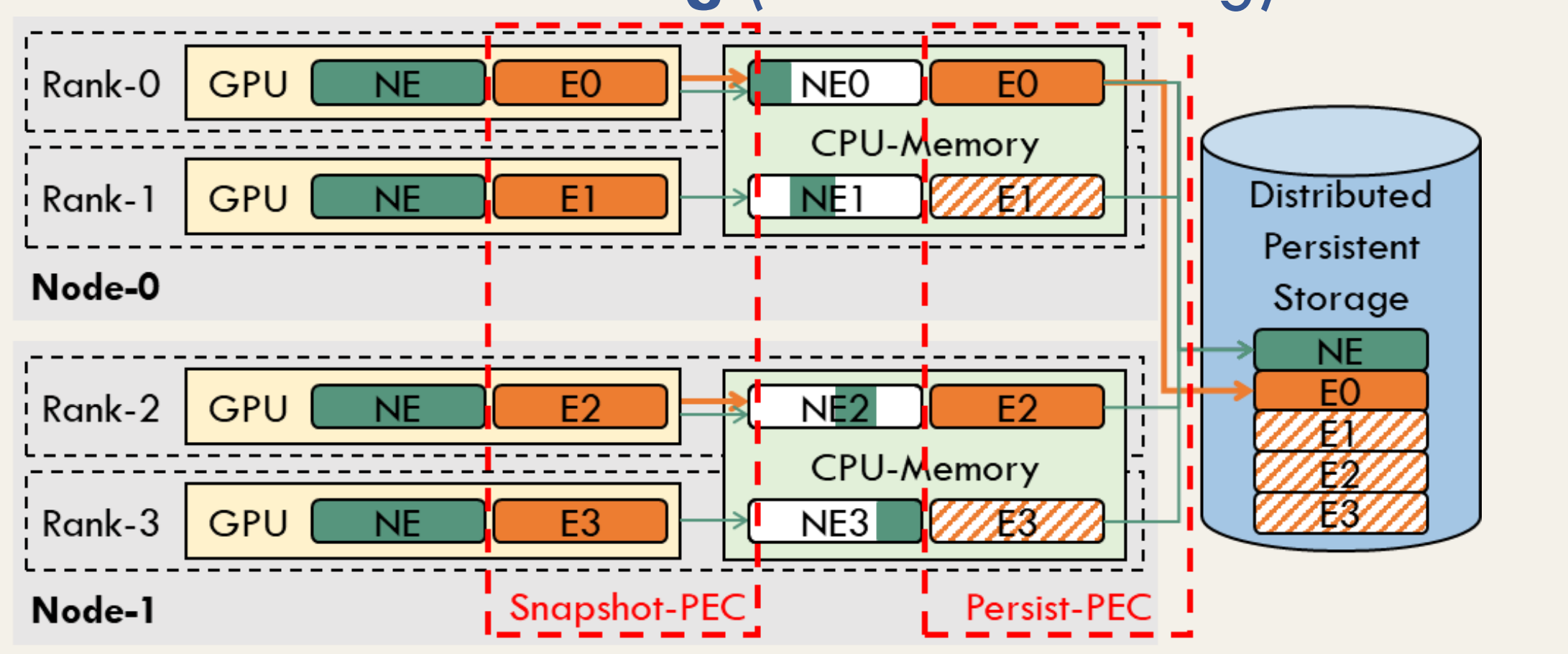


Baseline Equal Sharding for Non-Expert Part Adaptive Sharding for Non-Expert Part

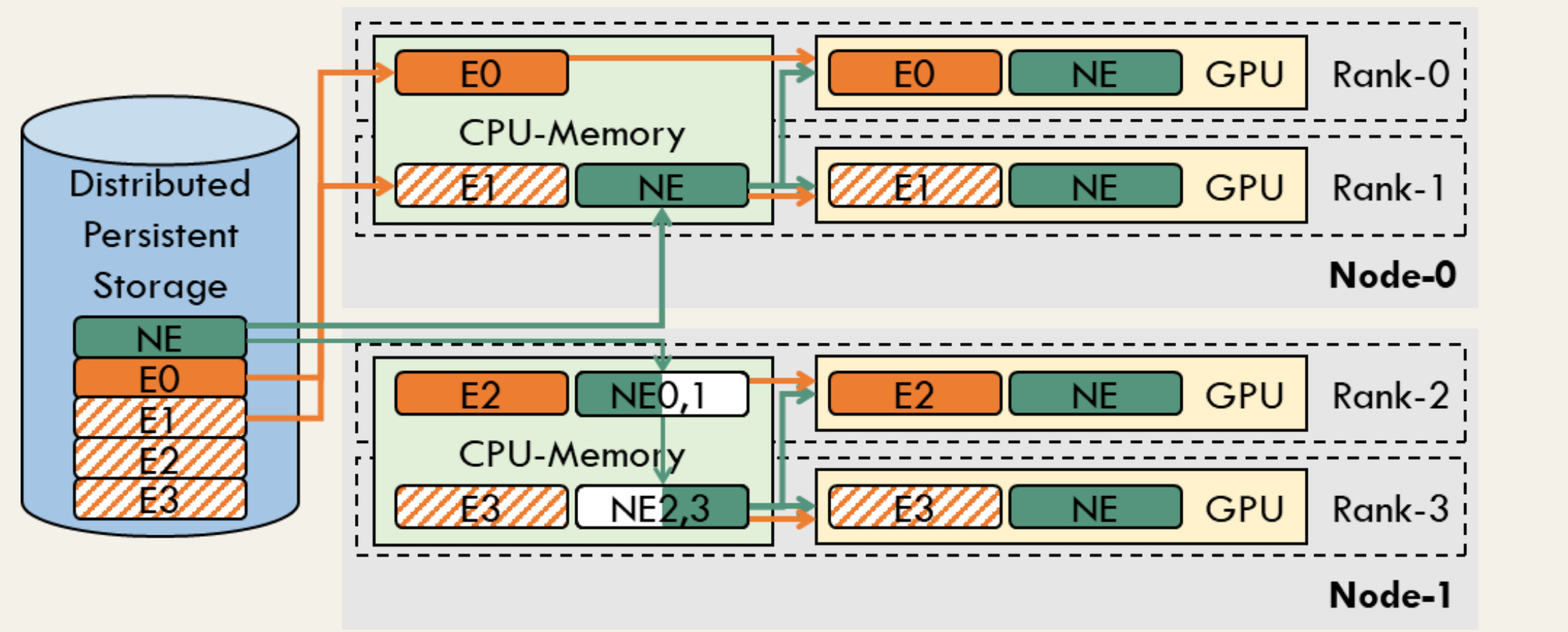
Two-Level Checkpointing Management

To maximize the benefits of hierarchical storage, we propose a two-level checkpointing management into our MoC system, comprising (1) in-memory snapshot and (2) persist, coupled with a suite of optimization techniques.

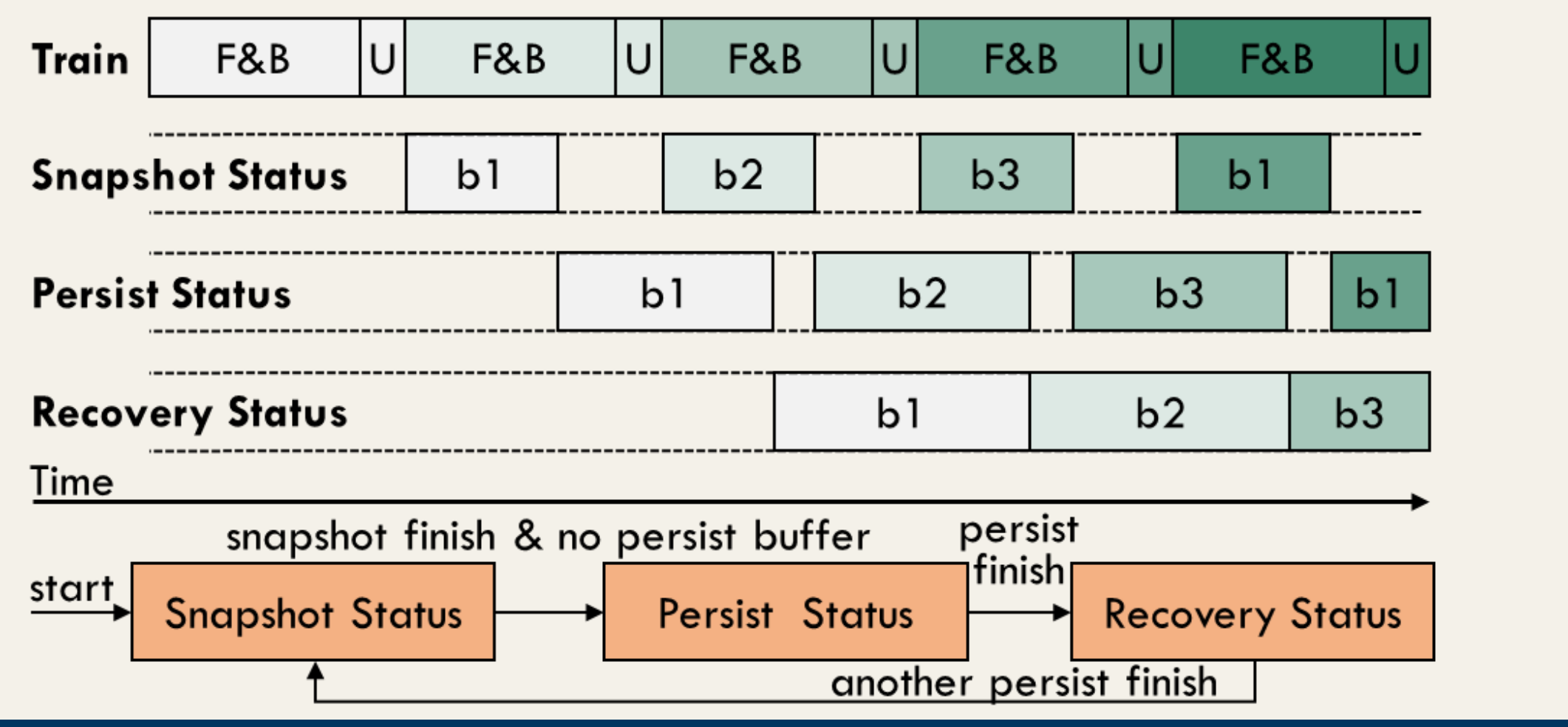
☐ **Two-level PEC Saving** (Normal Training):



☐ **Two-level PEC Recovery** (Node-0 Fault):



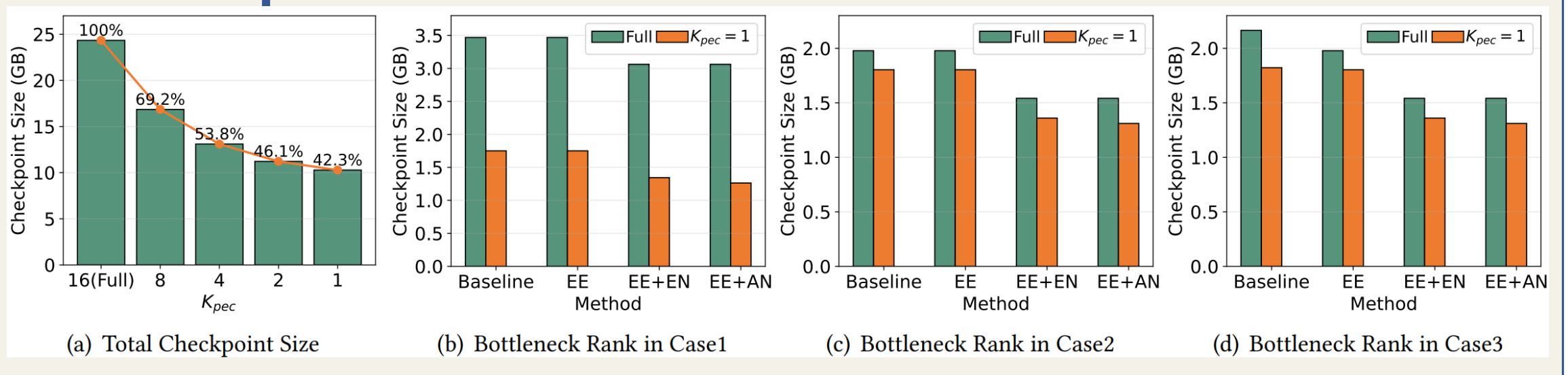
☐ **Asynchronous Checkpointing & Triple Buffer:**



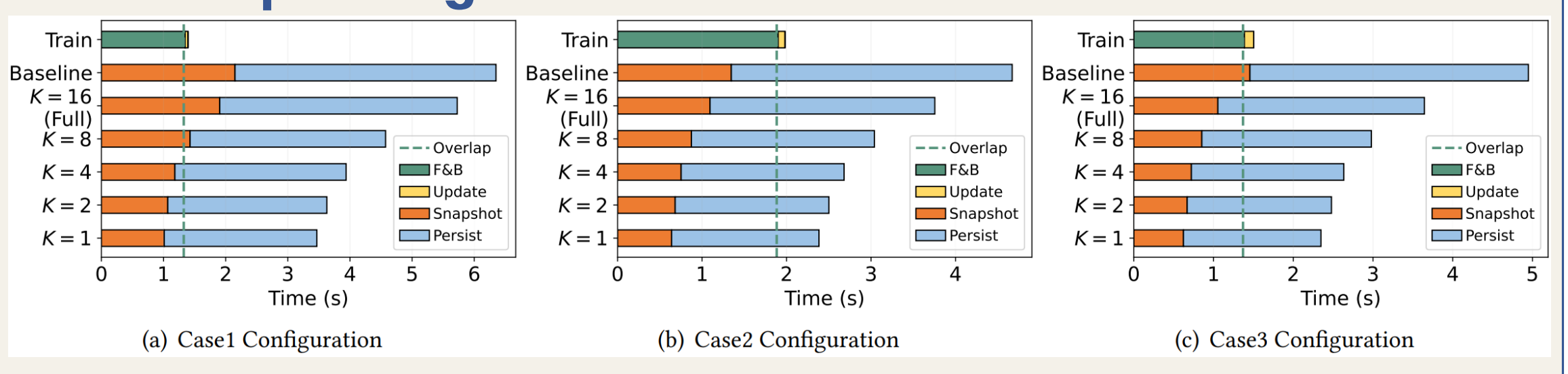
Evaluation

We implement our proposed MoC-System and conduct experiments upon the Megatron-DeepSpeed [4]. Our experiments include both language (GPT-MoE) and vision (Swin-MoE) models, utilizing three distributed training configurations of ZeRO-2 Data Parallelism and Expert Parallelism.

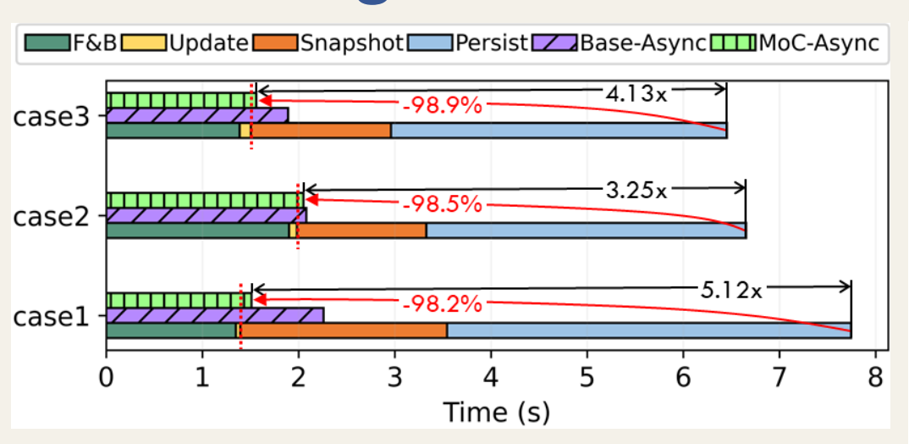
■ **Checkpoint Size:**

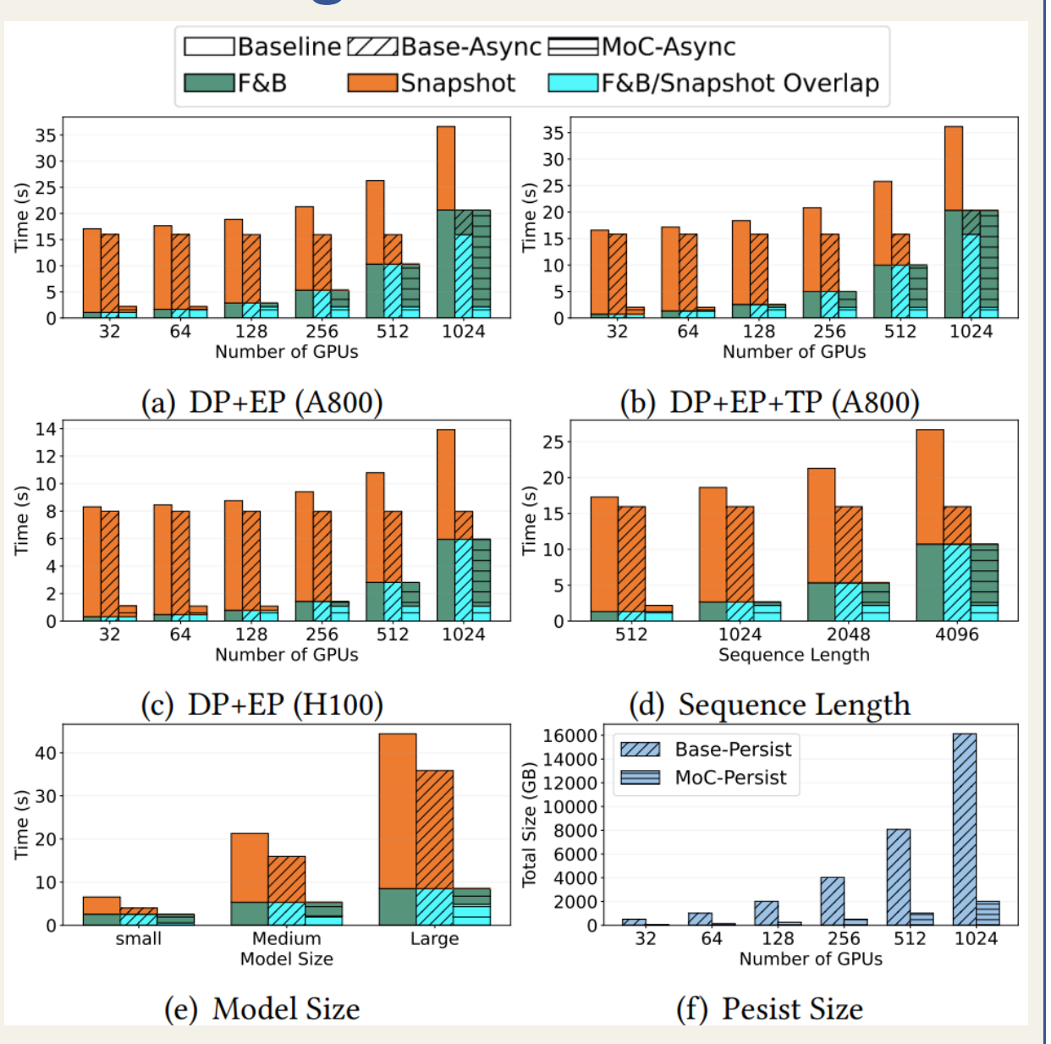


■ **Checkpointing Time:**

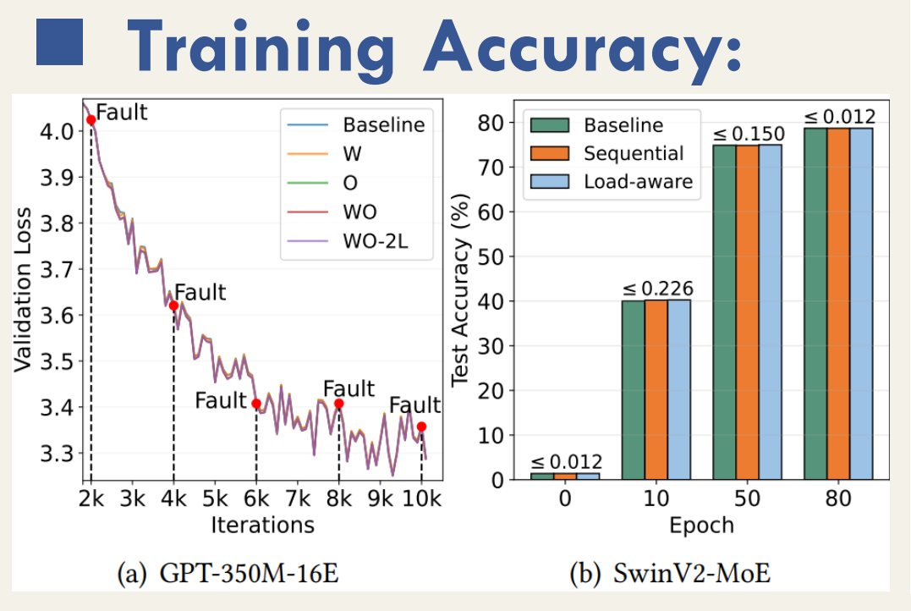


■ **Training Acceleration:** ■ **Scaling Simulation:**





■ **Training Accuracy:**



■ **Accuracy Impact on Downstream Tasks:**

Method	Ckpt	HellaSwag	PIQA	WinoGrande	BoolQ	ARC-E	OBQA	RACE	MathQA	Avg. (↑)
Baseline	1	26.85	58.22	49.09	54.77	36.83	13.00	24.21	20.54	35.44
W	0.88	26.92	58.16	49.72	57.52	**37.84**	12.80	24.69	**20.84**	36.06
O	0.54	26.93	58.00	48.54	61.28	37.21	**13.40**	**25.26**	19.97	36.32
WO	0.42	26.91	58.38	49.33	61.31	37.33	13.20	24.50	20.20	36.40
WO-2L	0.42	**26.96**	**58.49**	**50.12**	**61.74**	37.12	13.20	24.40	20.13	**36.52**
Deviation	-	(0.06, 0.11)	(-0.22, 0.27)	(-0.55, 1.03)	(2.75, 6.97)	(0.29, 1.01)	(-0.20, 0.40)	(0.19, 1.05)	(-0.57, 0.30)	(0.62, 1.08)

References

☐ [1] Jiang, Ziheng, et al. "MegaScale: Scaling large language model training to more than 10,000 GPUs." 21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24). 2024.

☐ [2] Mohan, Jayashree, Amar Phanishayee, and Vijay Chidambaram. "CheckFreq: Frequent,Fine-Grained DNN Checkpointing." 19th USENIX Conference on File and Storage Technologies (FAST 21). 2021.

☐ [3] Zoph, Barret, et al. "St-moe: Designing stable and transferable sparse expert models." arXiv preprint arXiv:2202.08906 (2022).

☐ [4] Microsoft. 2022. Megatron-DeepSpeed. https://github.com/microsoft/Megatron-DeepSpeed