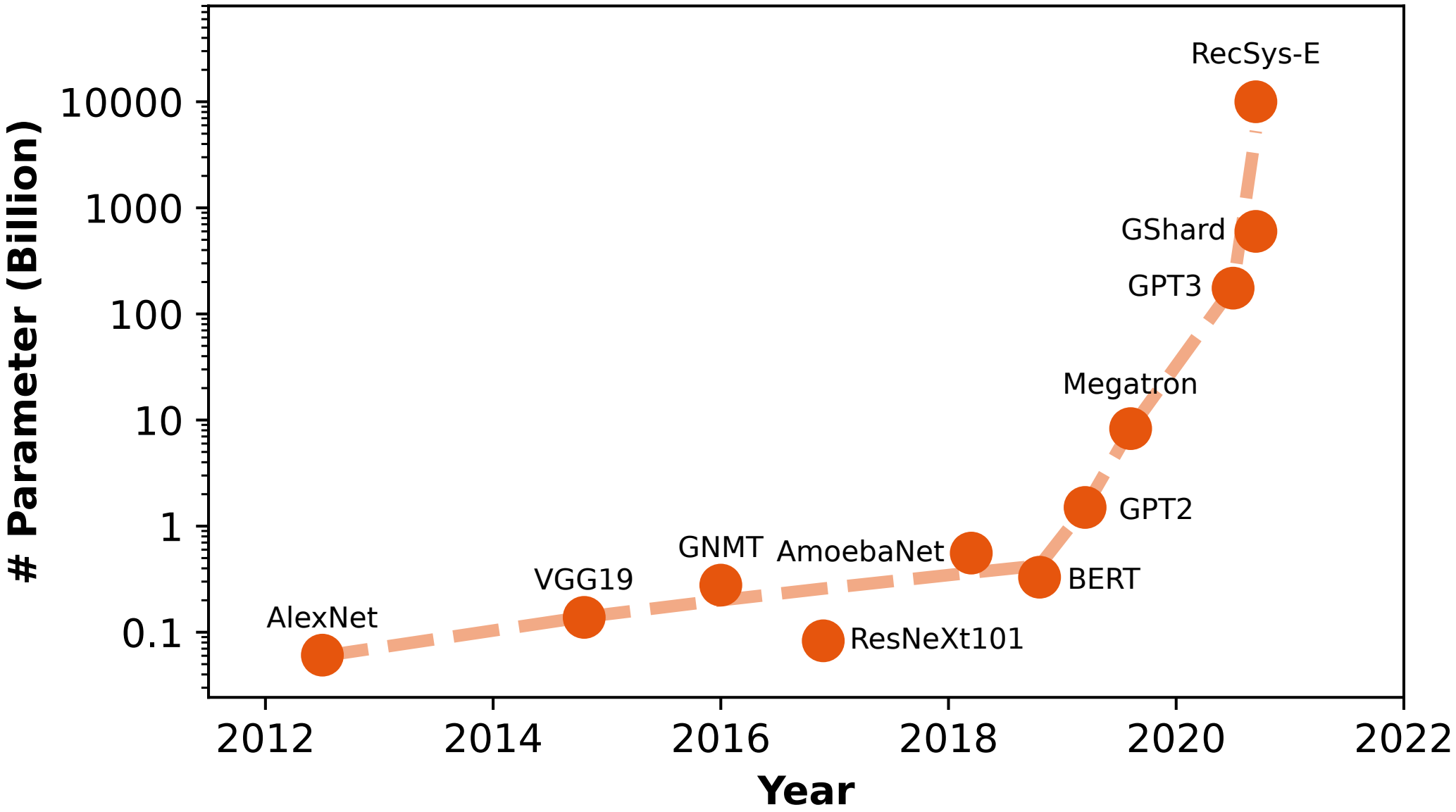
Harmony: Overcoming the Hurdles of GPU Memory Capacity to Train Massive DNN Models on Commodity Servers

Youjie Li

Amar Phanishayee Derek Murray Jakub Tarnawski Nam Sung Kim

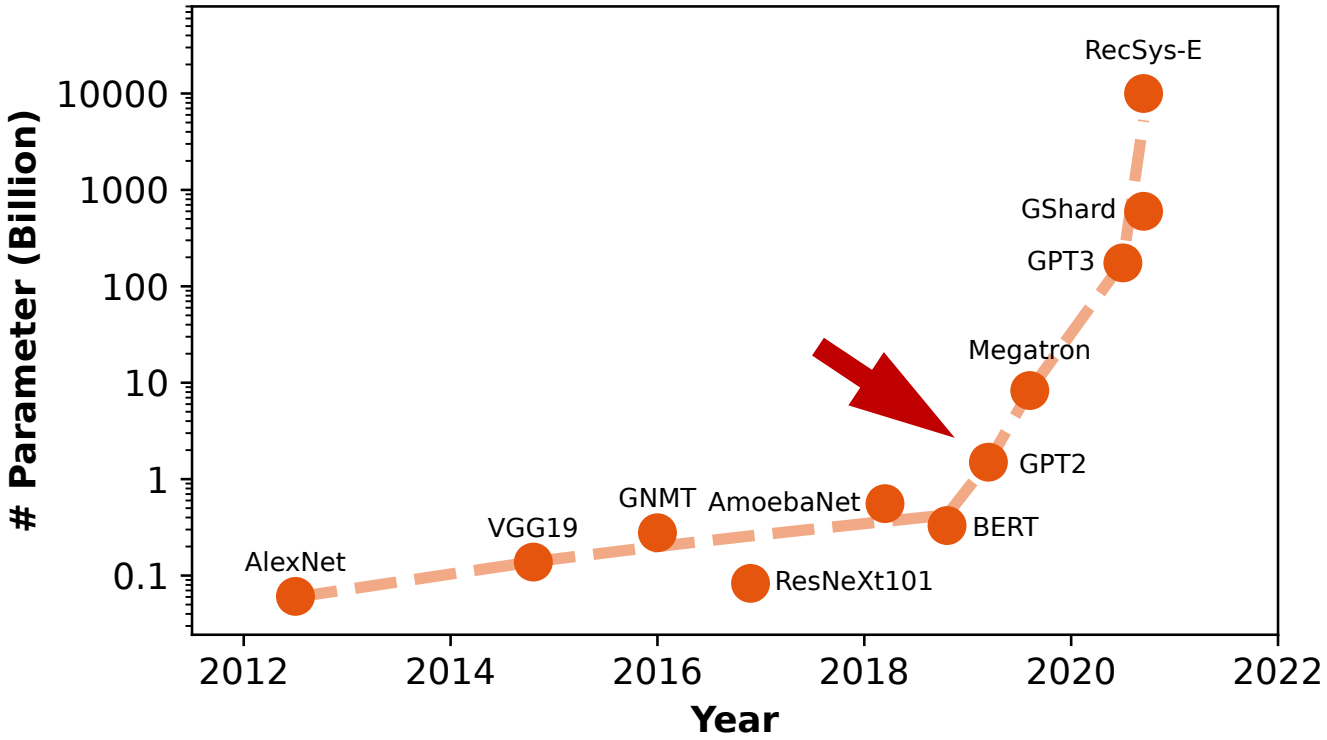
VLDB 2022

DNN Model Size are Growing Exponentially!



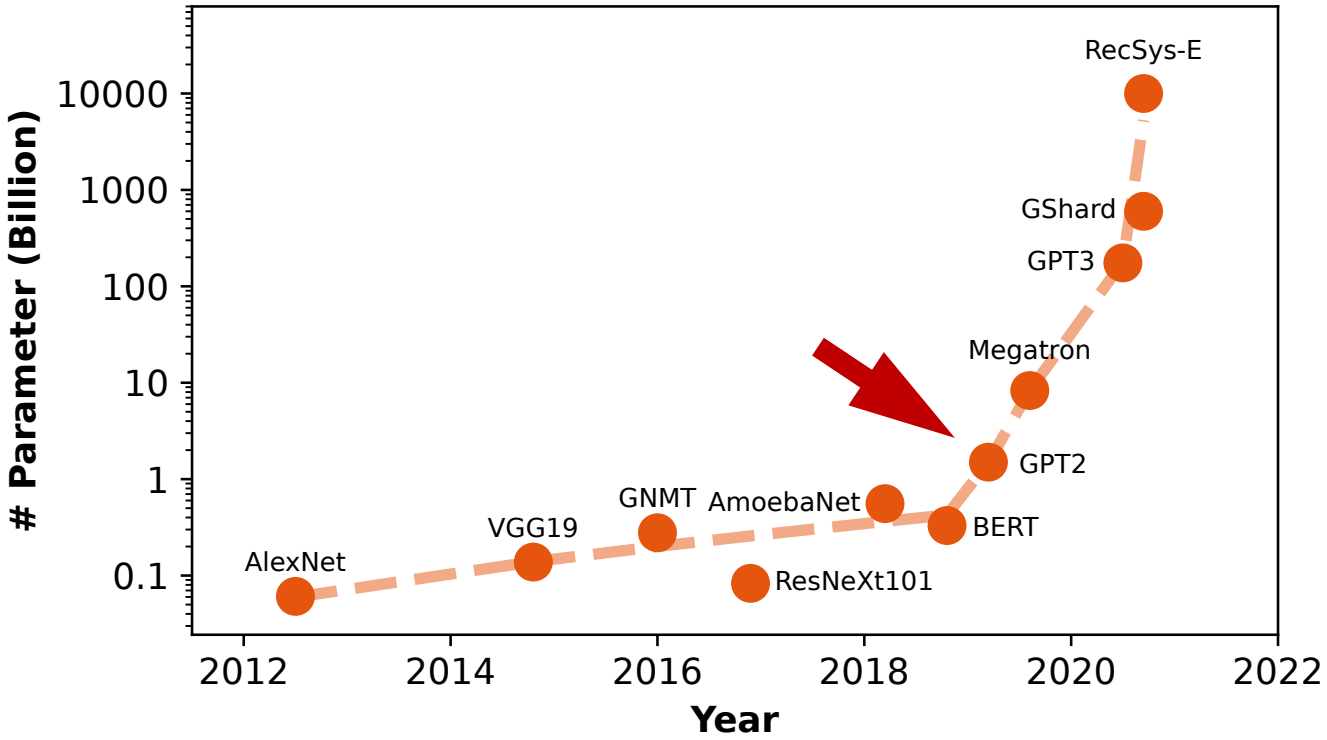
Training Requires Huge Amount of Memory

128 GB footprint for training GPT2*



*[GPT2, arXiv'19][Zero, SC'20]

Training Requires Huge Amount of Memory

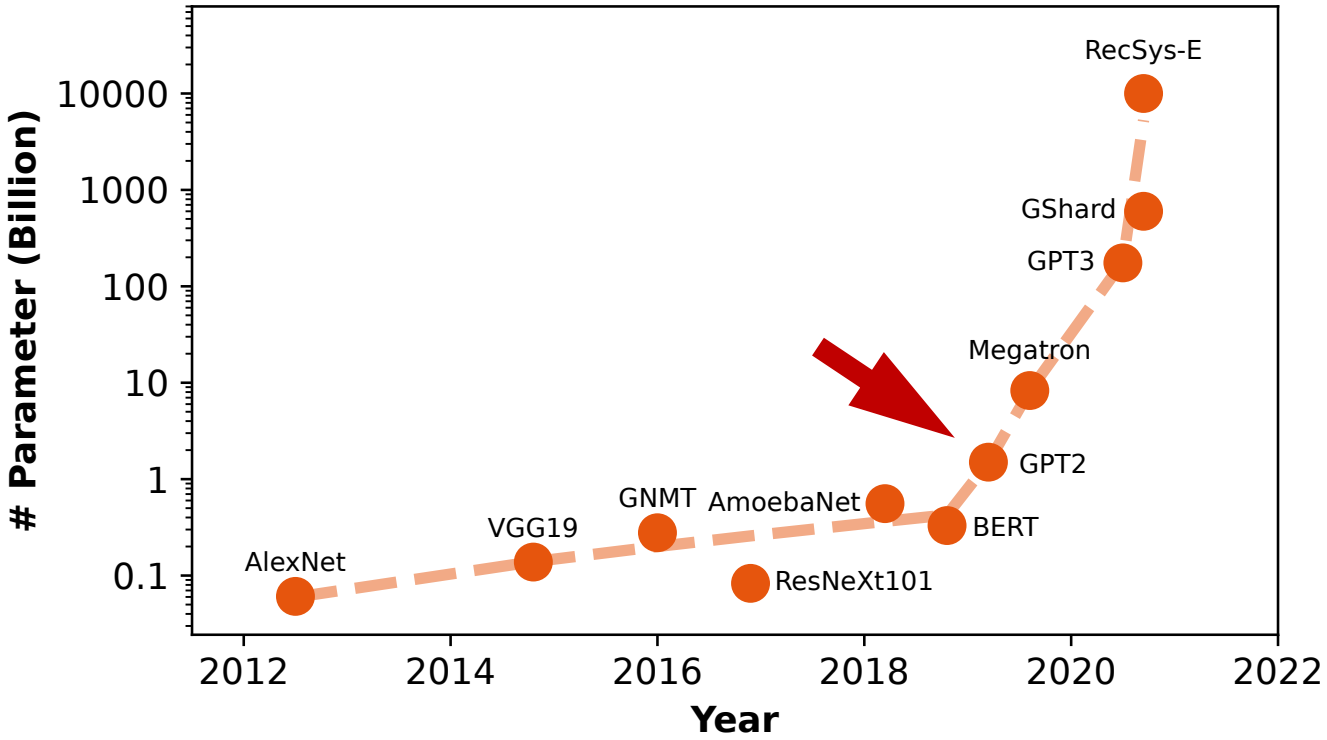


128 GB footprint for training GPT2*

Breakdown	Memory	Percent
Parameter	**3 GB**	**2.3%**
Gradient	3 GB	2.3%
Optimizer	18 GB	14.1%
Stashing	60 GB	46.9%
Buffer	6 GB	4.7%
Other	38 GB	29.7%
Total	128 GB	100%

*[GPT2, arXiv'19][Zero, SC'20]

Training Requires Huge Amount of Memory



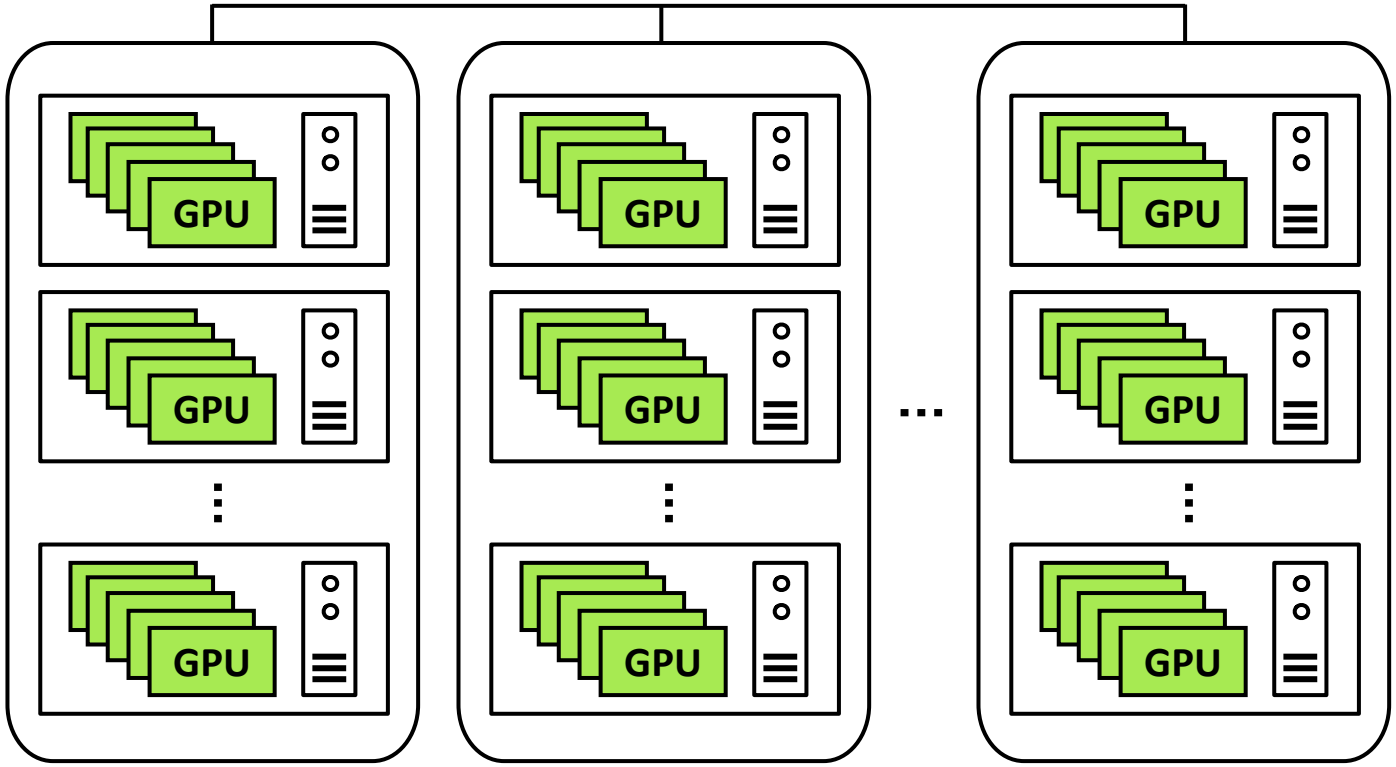
128 GB footprint for training GPT2*

Breakdown	Memory	Percent
Parameter	**3 GB**	**2.3%**
Stashing	60 GB	46.9%
Buffer	6 GB	4.7%
Other	38 GB	29.7%
Total	128 GB	100%

Parameter is a tiny fraction!

*[GPT2, arXiv'19][Zero, SC'20]

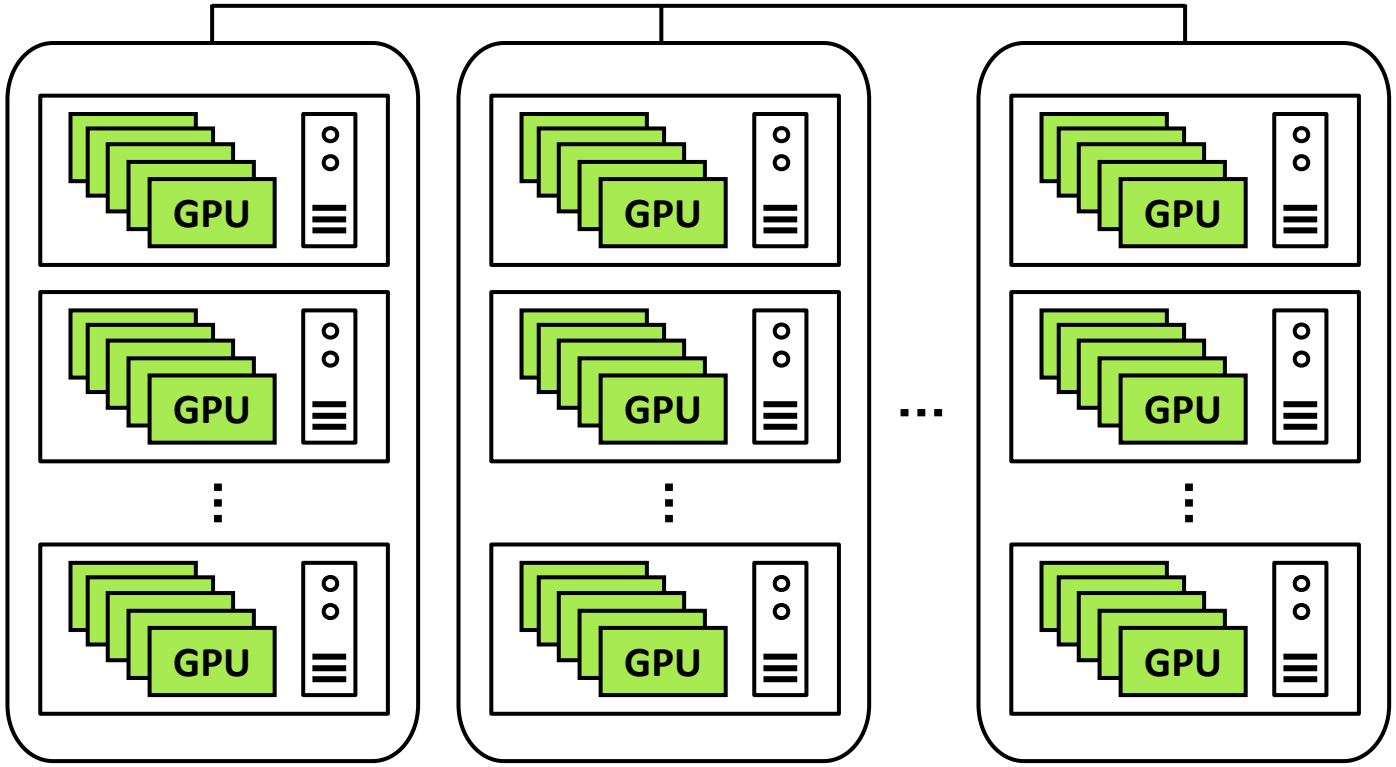
Only “Elites” can Train Massive Models



A GPU Cluster of the Elites

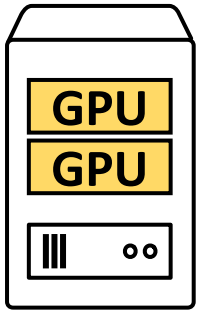
(e.g., Thousands of GPUs to Train GPT-3)

Only "Elites" can Train Massive Models



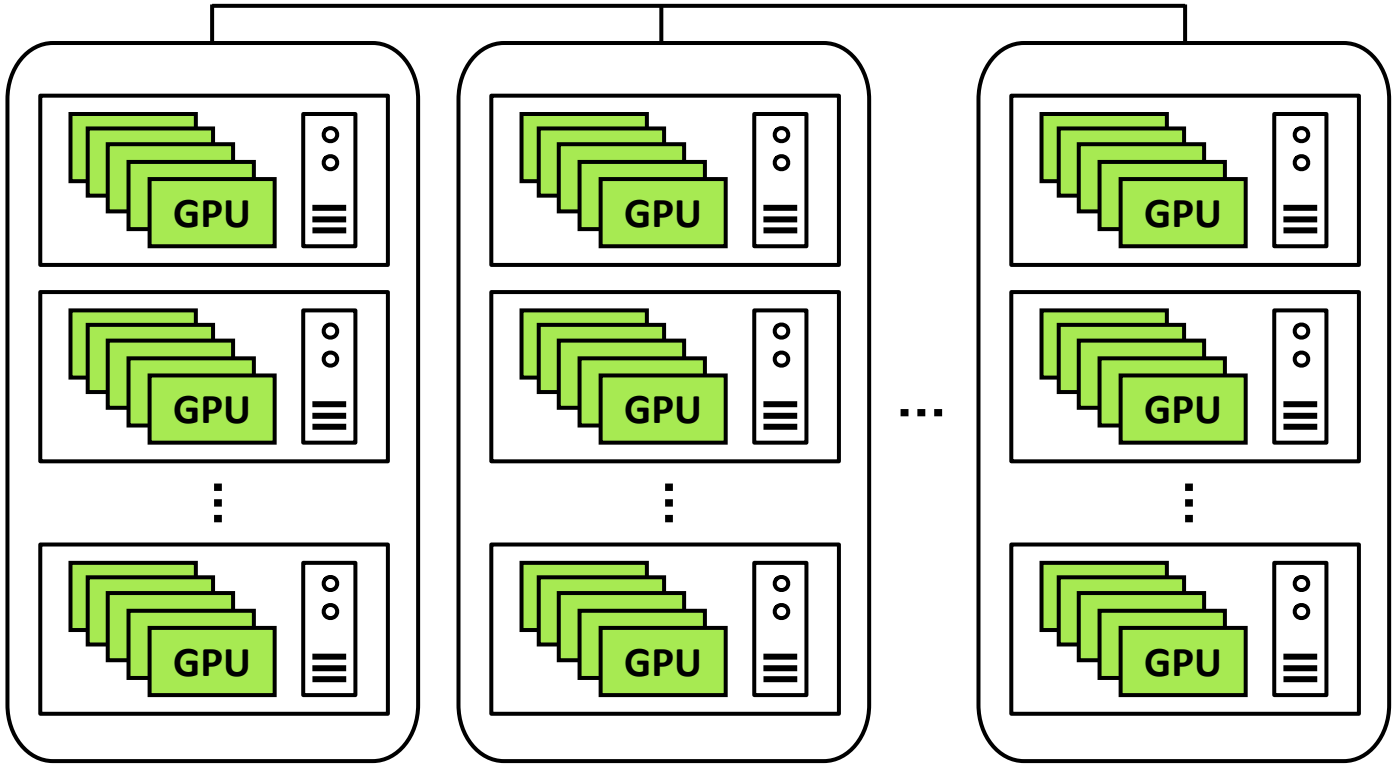
A GPU Cluster of the Elites

(e.g., Thousands of GPUs to Train GPT-3)



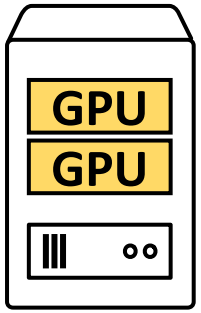
A Commodity Server of the Masses

Only "Elites" can Train Massive Models



A GPU Cluster of the Elites

(e.g., Thousands of GPUs to Train GPT-3)



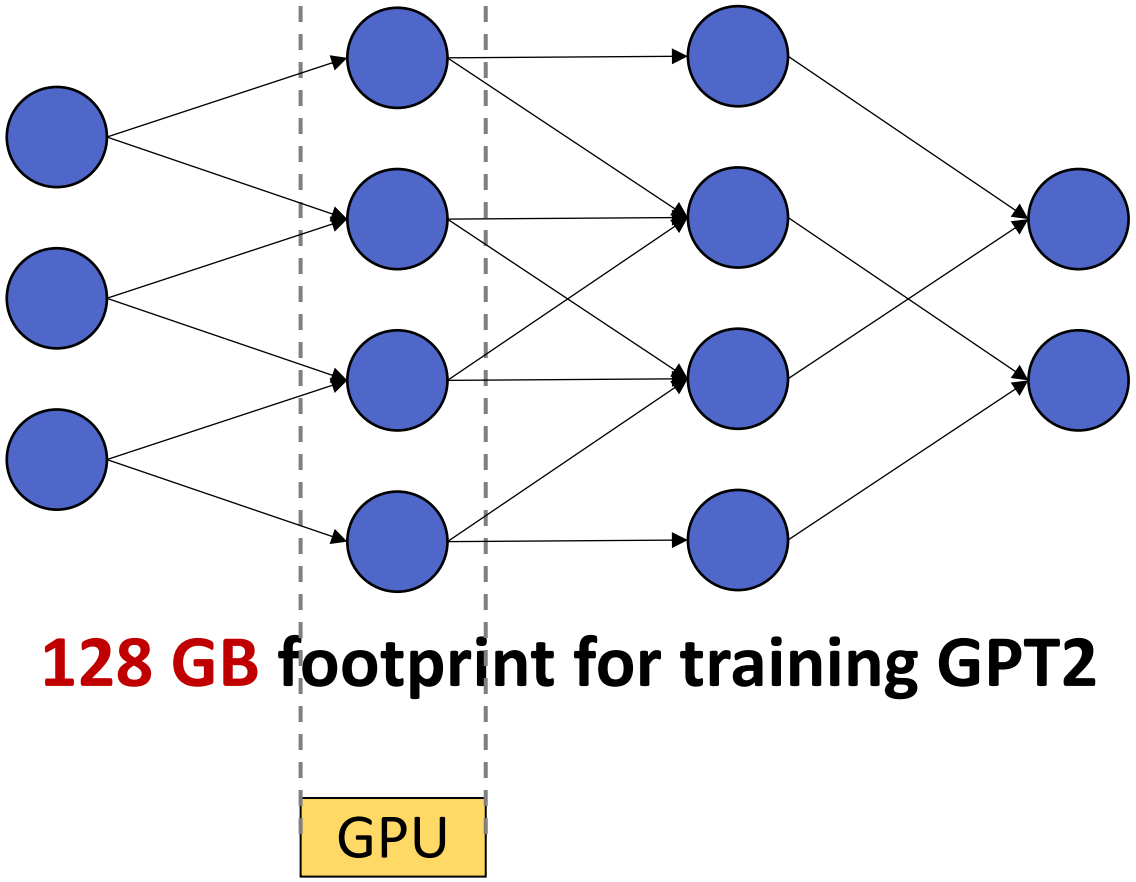
A Commodity Server of the Masses

Do the Masses Stand a Chance?

How to Train Massive Models on Commodity Servers? Efficiently?

The Main Challenge:

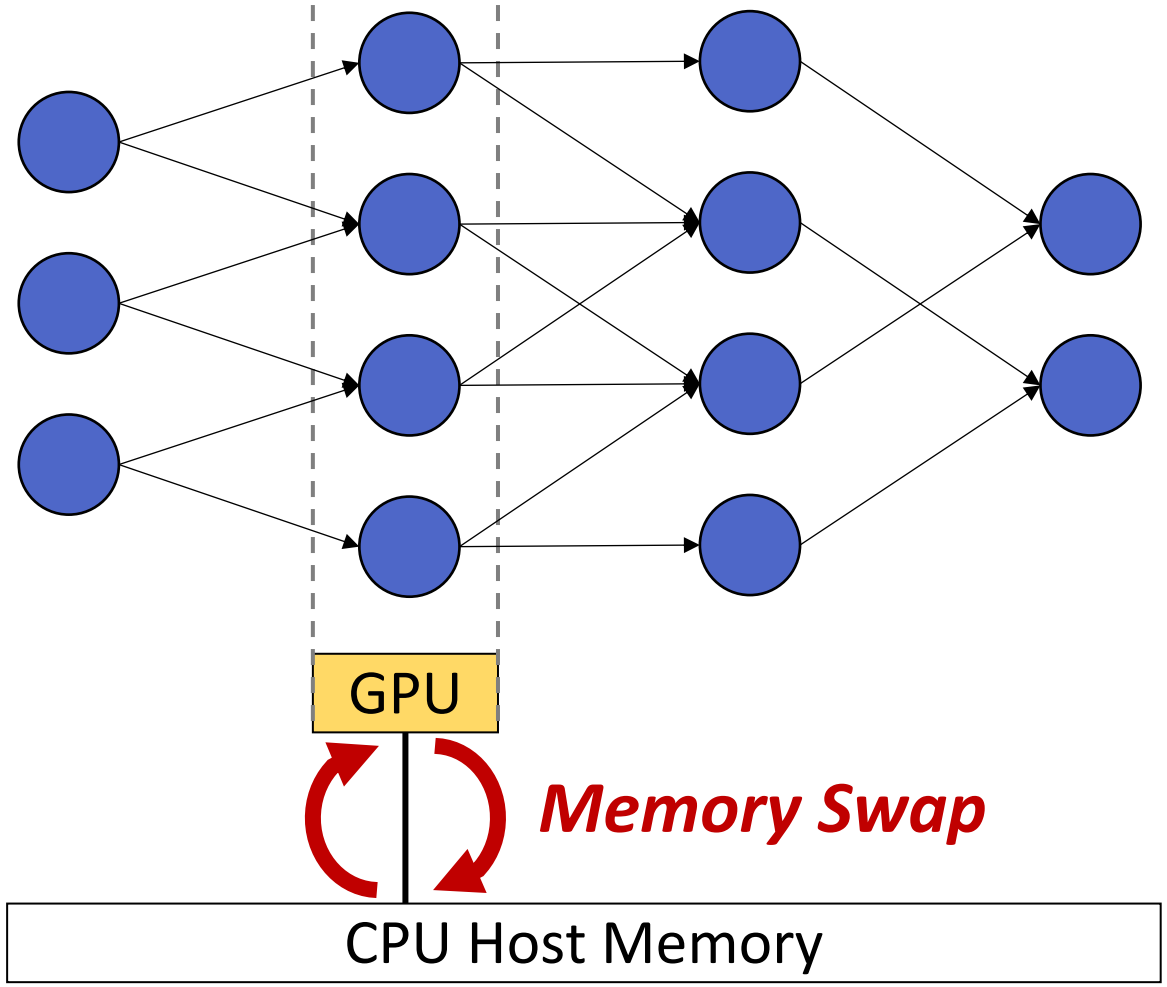
Training Memory Footprint >> GPU Memory Capacity



128 GB footprint for training GPT2

16 GB V100

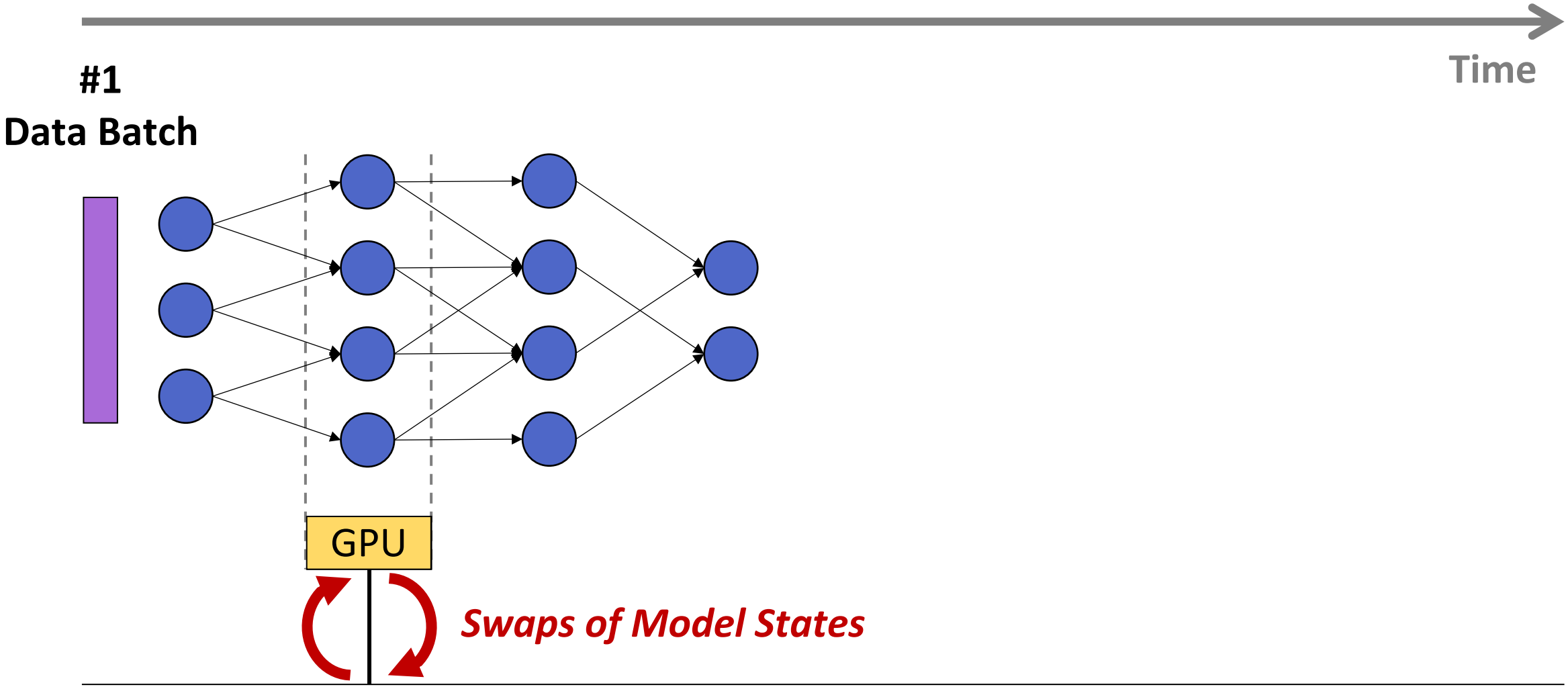
Promising Technique #1: *Single-GPU Memory Virtualization**



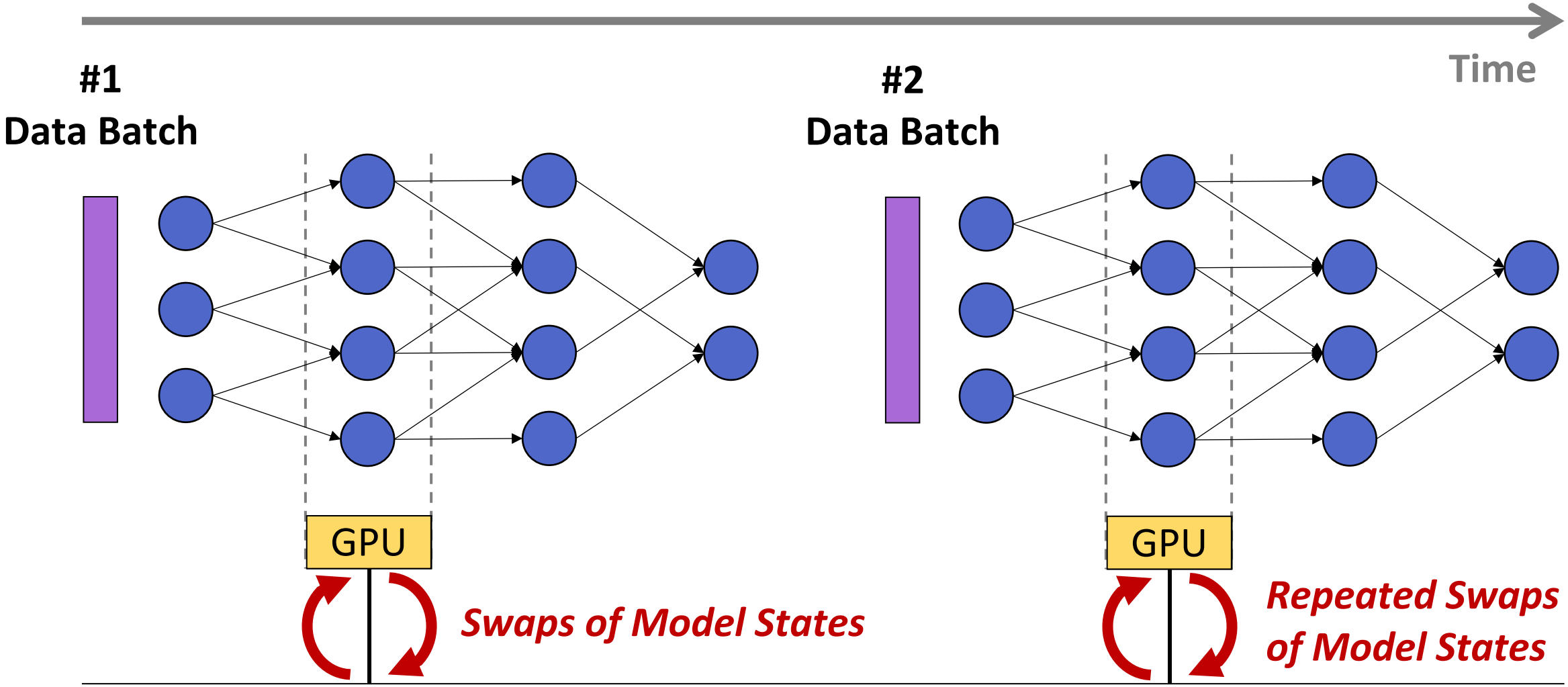
*[vDNN, MICRO'16]
[LMS, SysML'18]
[SwpAdv, ASPLOS'20]
[Sentinel, HPCA'21]

Excessive Swap Overhead:

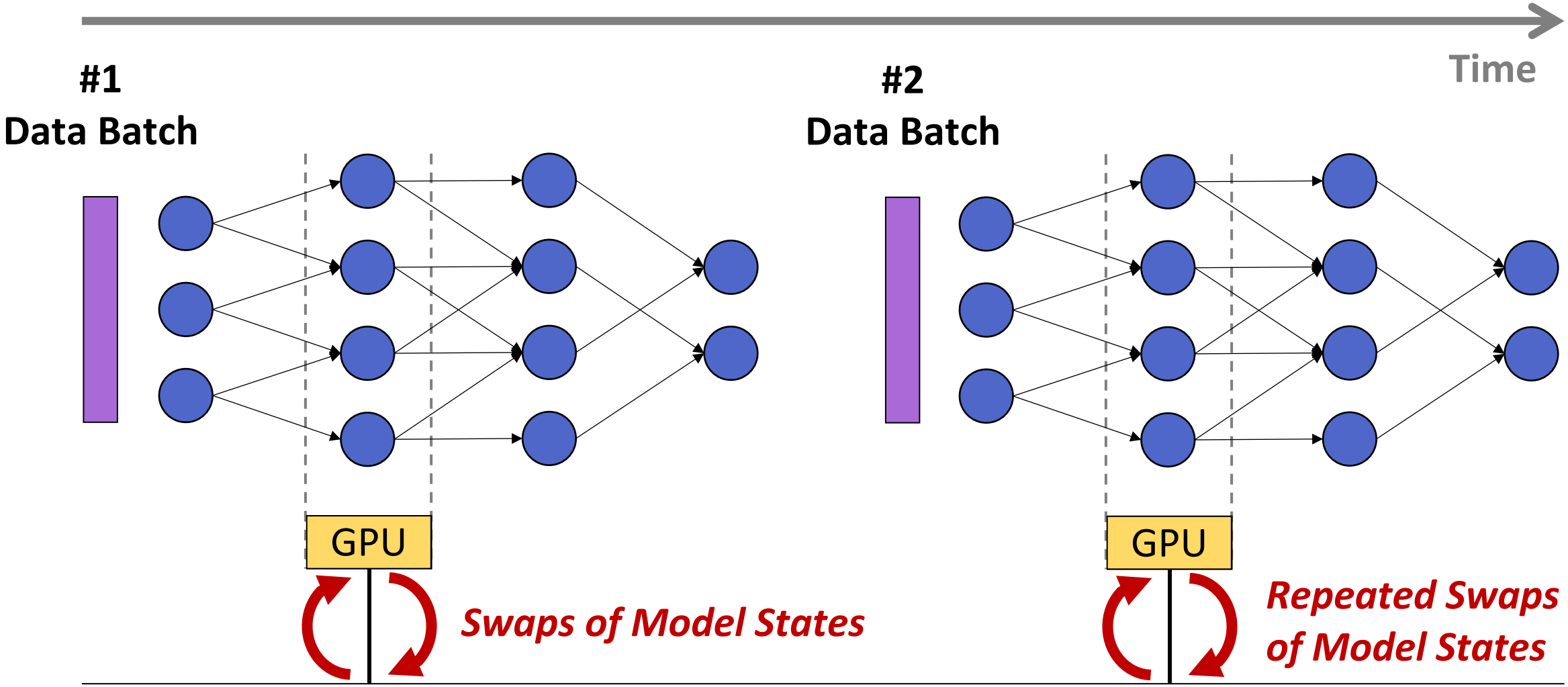
Excessive Swap Overhead:



Excessive Swap Overhead:

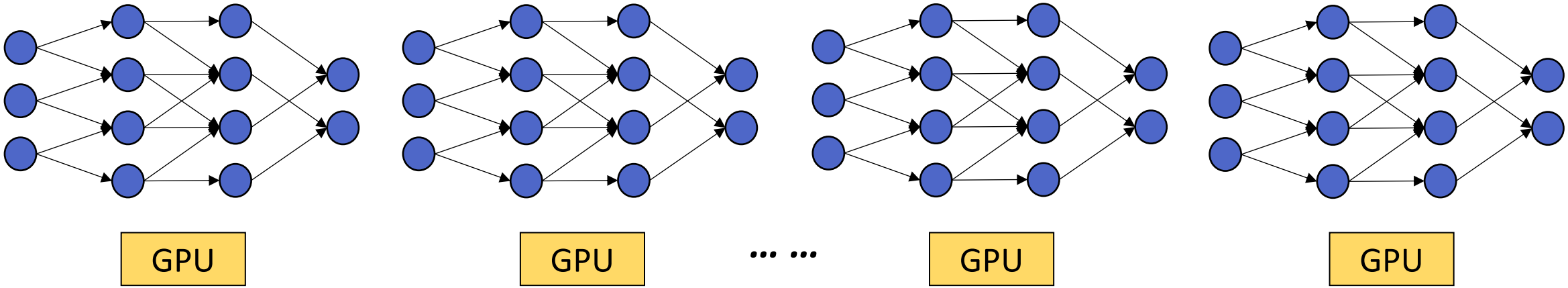


Excessive Swap Overhead: Repeated Swaps across Data Batches

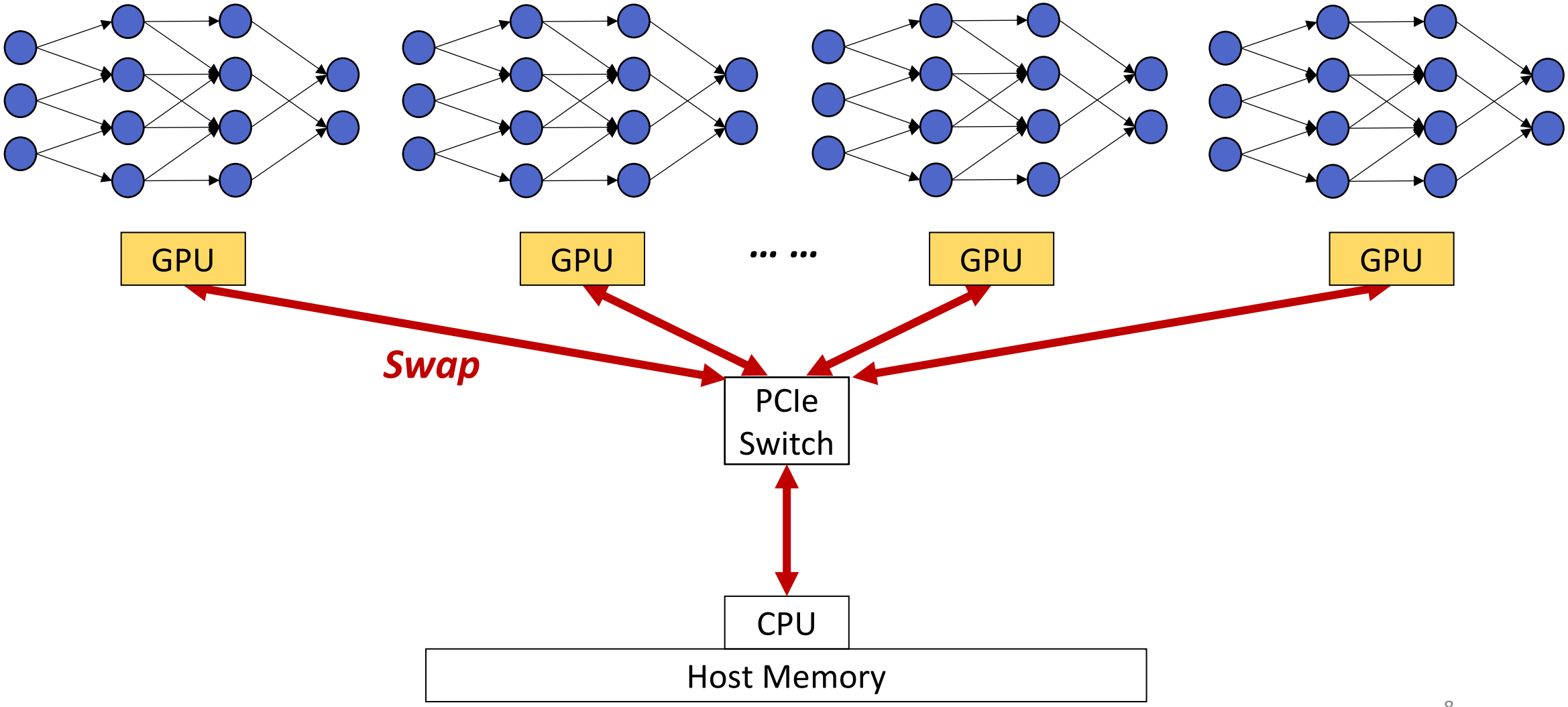


CPU-GPU Swap Bottleneck

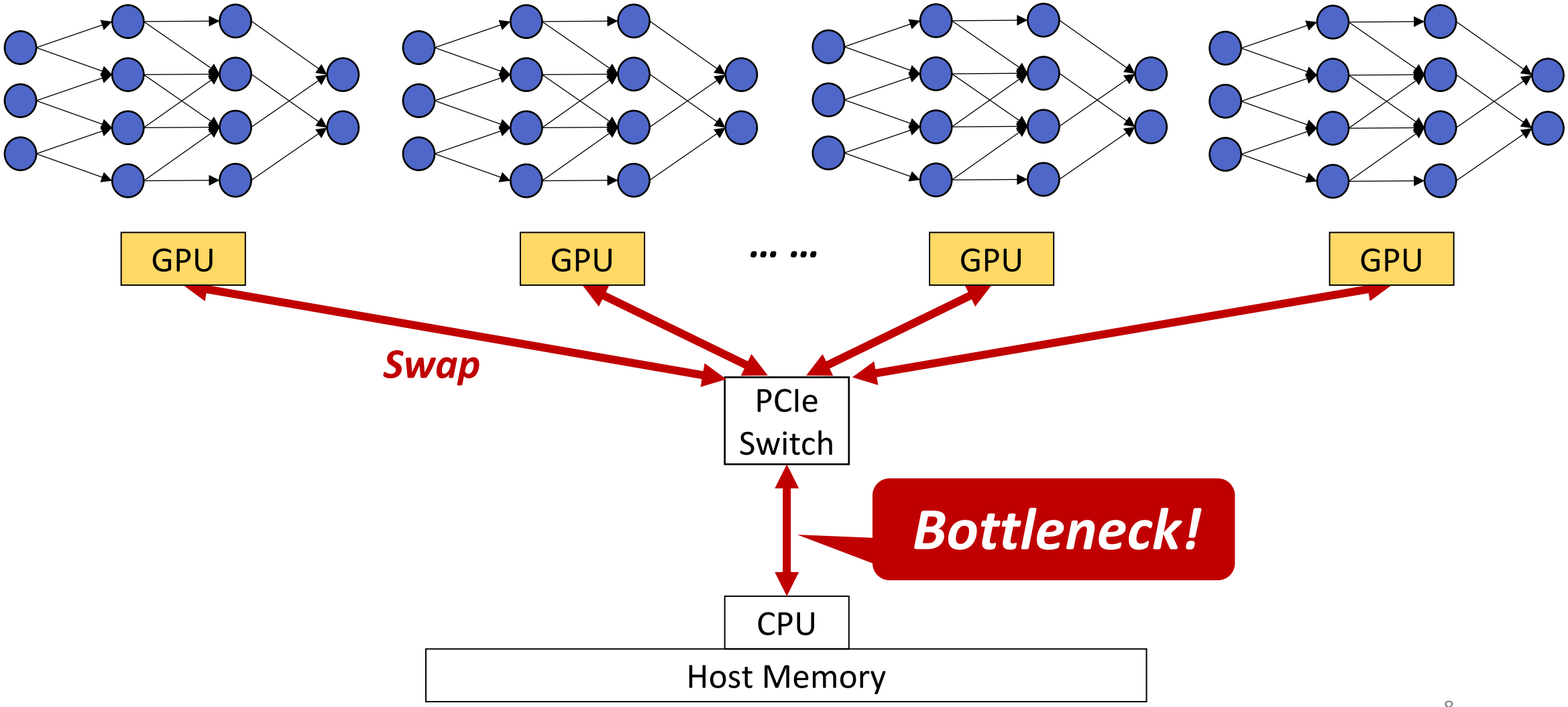
CPU-GPU Swap Bottleneck



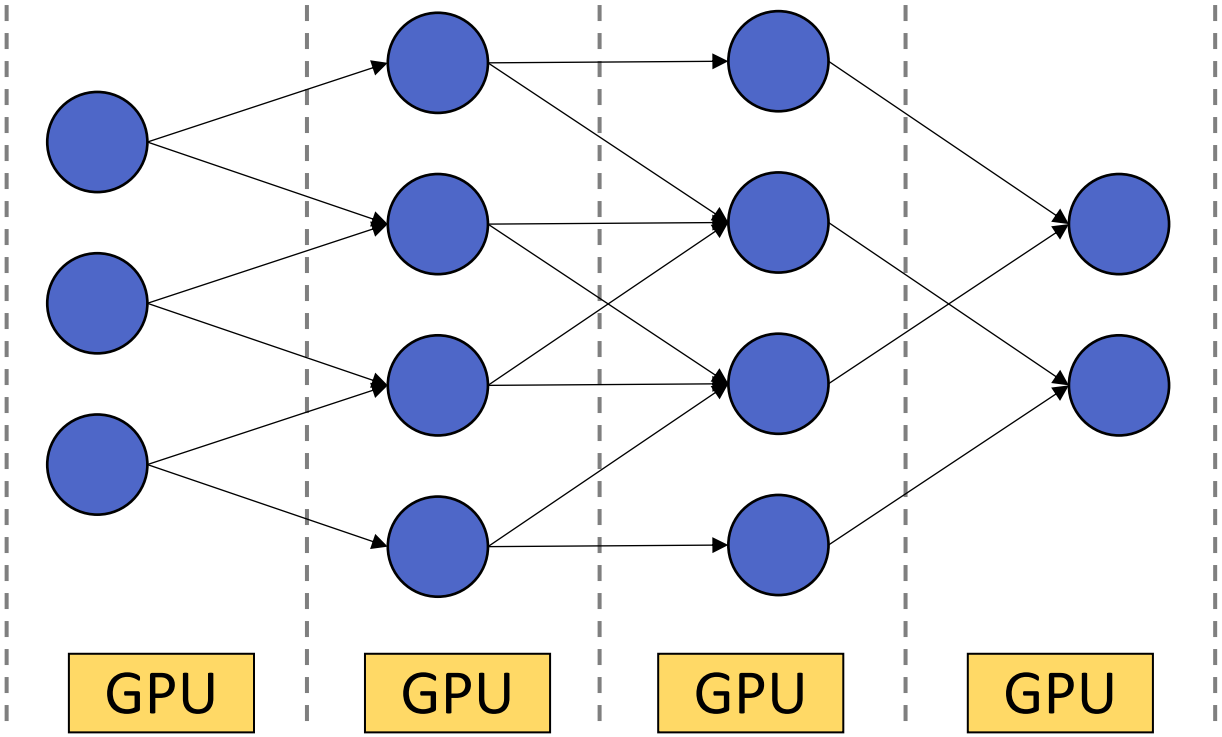
CPU-GPU Swap Bottleneck



CPU-GPU Swap Bottleneck

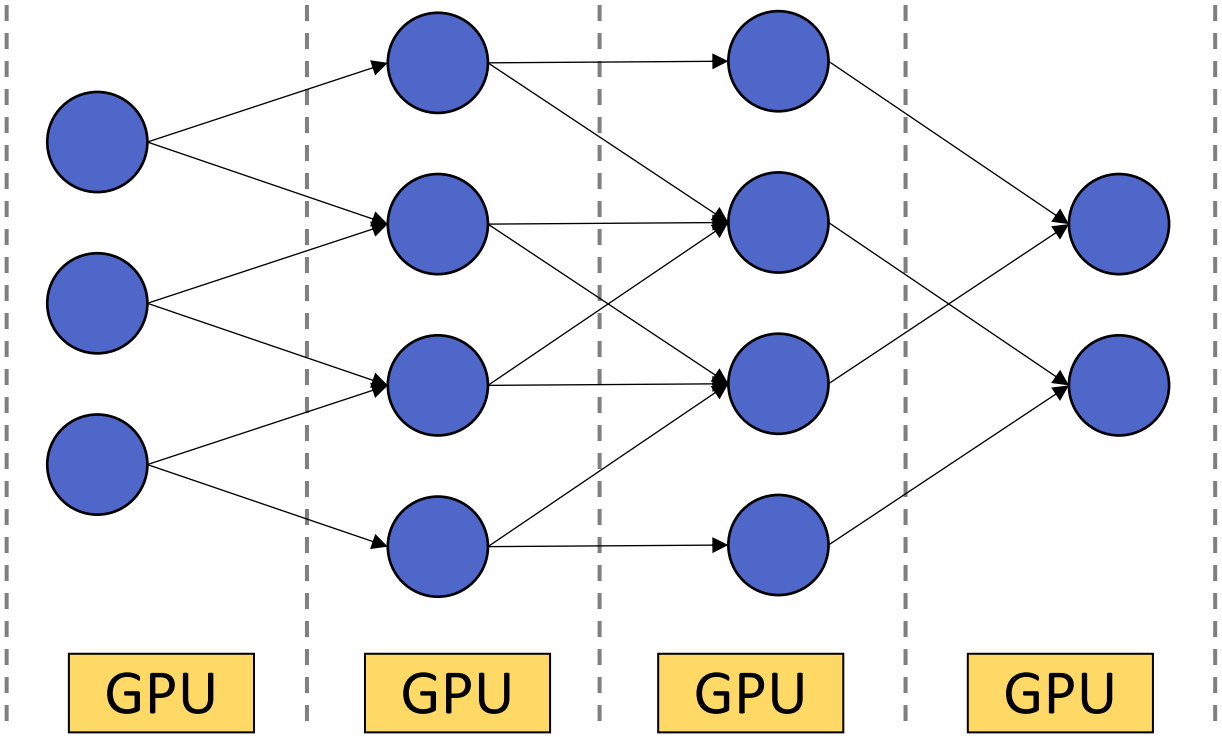


Promising Technique #2: *Model/Pipeline Parallel Training*



- **Model Parallelism** [Megatron, arXiv'19]
- **Pipeline Parallelism** [GPipe, NeurIPS'19] [PipeDream, SOSP'19]

Promising Technique #2: *Model/Pipeline Parallel Training*

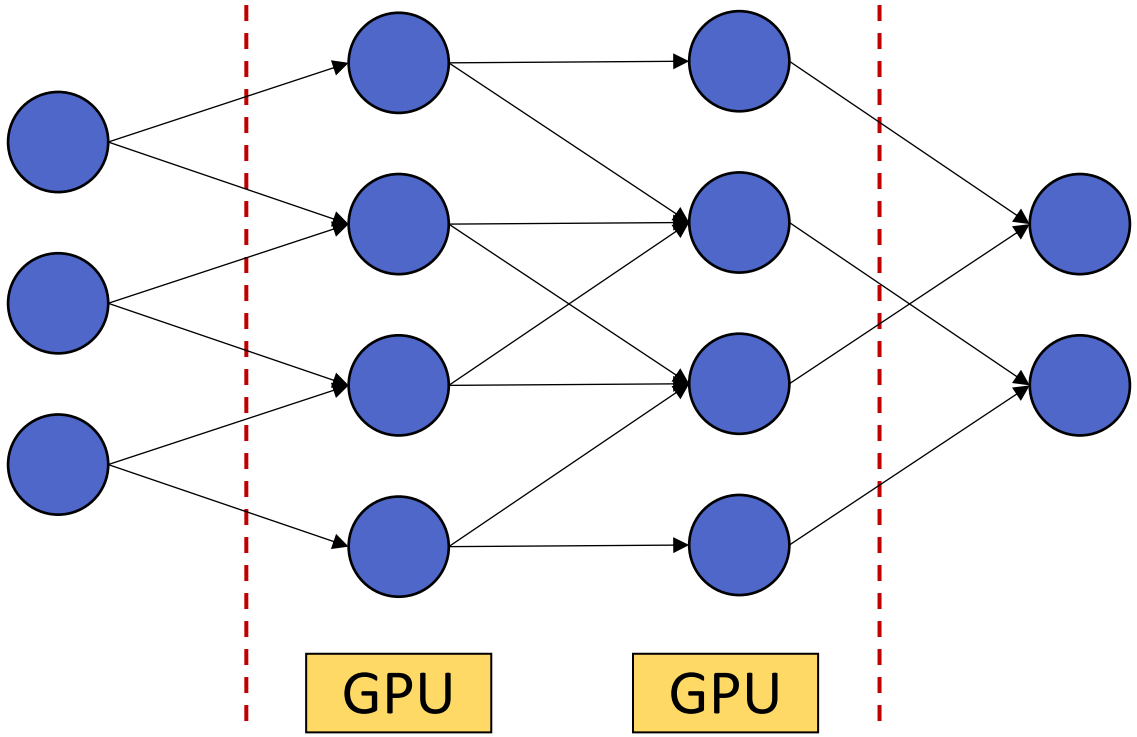


- **Model Parallelism** [Megatron, arXiv'19]
- **Pipeline Parallelism** [GPipe, NeurIPS'19] [PipeDream, SOSP'19]

Great for models that fit collective GPU memory capacity

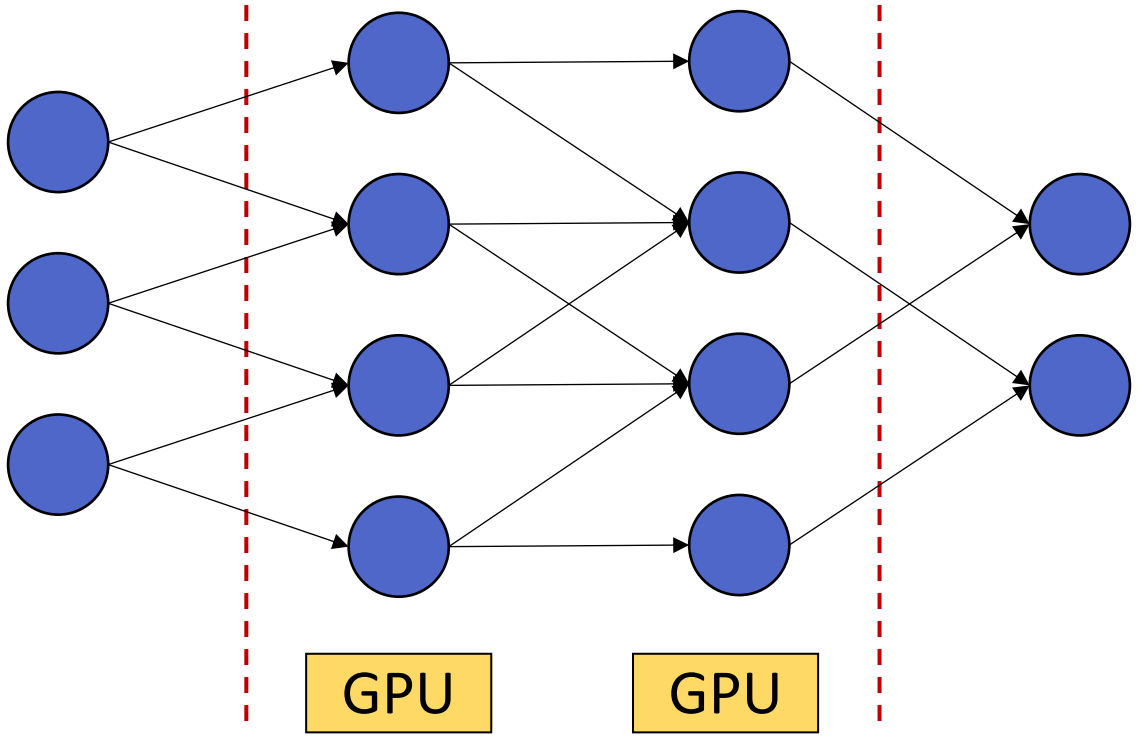
But on Commodity Servers:

Training Memory Footprint > Collective GPU Memory Capacity



But on Commodity Servers:

Training Memory Footprint > Collective GPU Memory Capacity



Large Model Size | **Large Data Size** | **Low-End GPUs** | **Insufficient # of GPUs**

Excessive Swap Overhead

CPU-GPU Swap Bottleneck

Unbalanced Swaps in Pipeline Parallelism

Excessive Swap Overhead

CPU-GPU Swap Bottleneck

Unbalanced Swaps in Pipeline Parallelism

Reason

The Fundamental Limitations in Current ML Frameworks

Excessive Swap Overhead

CPU-GPU Swap Bottleneck

Unbalanced Swaps in Pipeline Parallelism

Reason

The Fundamental Limitations in Current ML Frameworks

Coarse-Granularity Scheduling

Excessive Swap Overhead

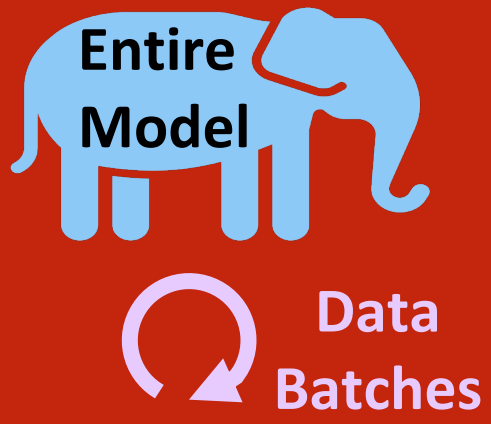
CPU-GPU Swap Bottleneck

Unbalanced Swaps in Pipeline Parallelism

Reason

The Fundamental Limitations in Current ML Frameworks

Coarse-Granularity Scheduling



Excessive Swap Overhead

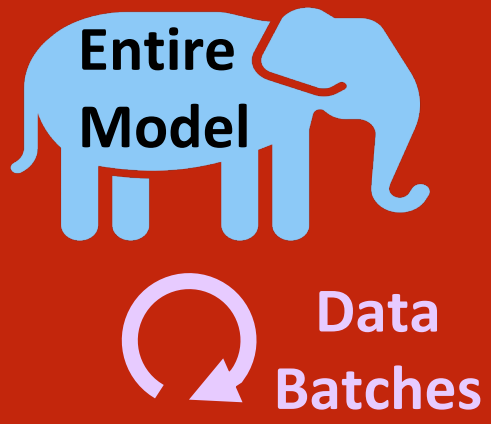
CPU-GPU Swap Bottleneck

Unbalanced Swaps in Pipeline Parallelism

Reason

The Fundamental Limitations in Current ML Frameworks

Coarse-Granularity Scheduling



Early Binding to Devices

Excessive Swap Overhead

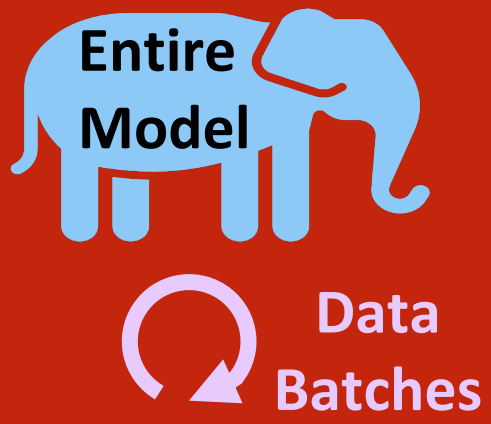
CPU-GPU Swap Bottleneck

Unbalanced Swaps in Pipeline Parallelism

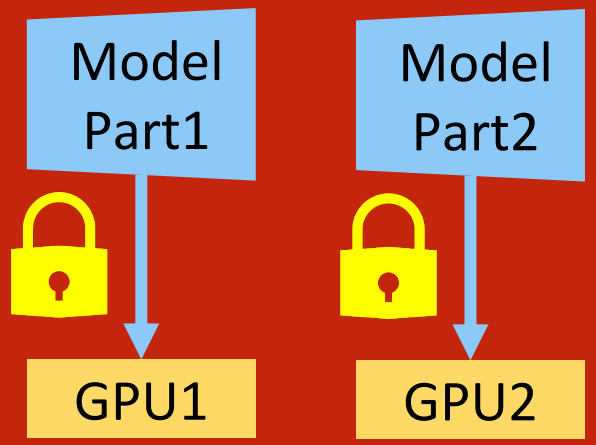
Reason

The Fundamental Limitations in Current ML Frameworks

Coarse-Granularity Scheduling

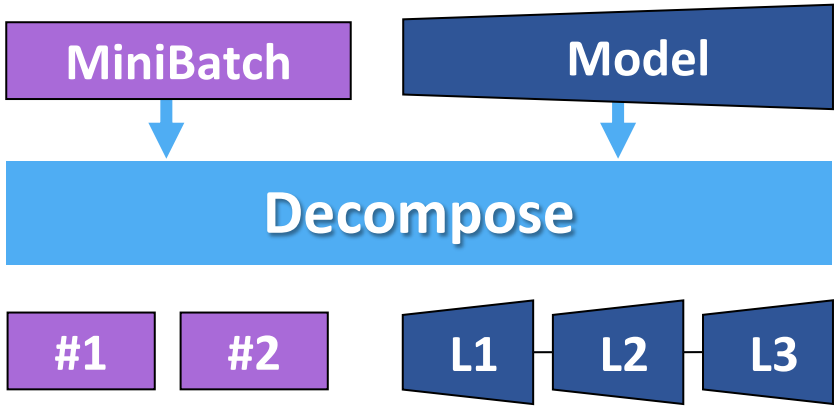


Early Binding to Devices

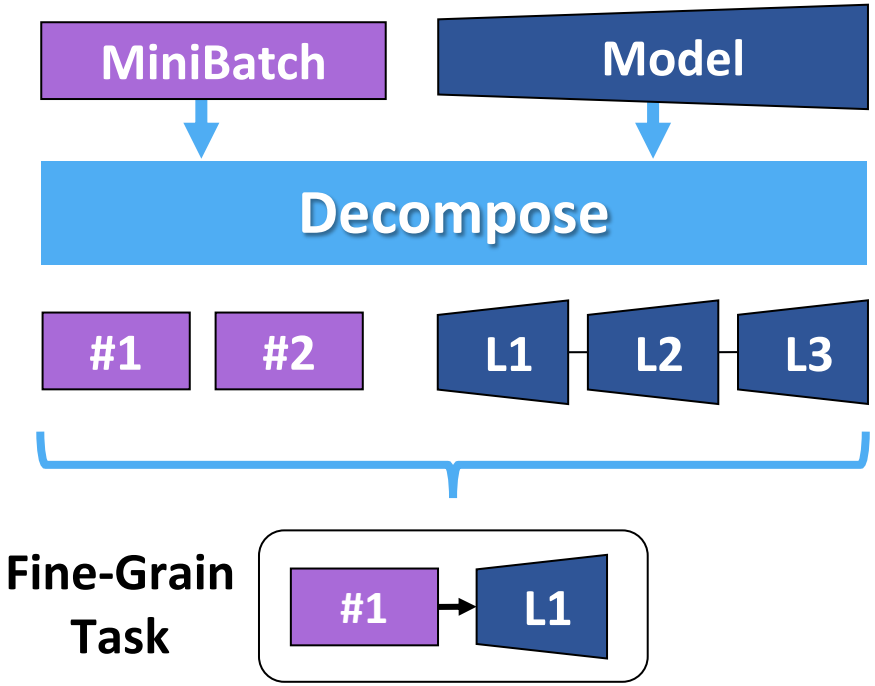


Harmony: A New Framework for Massive Model Training

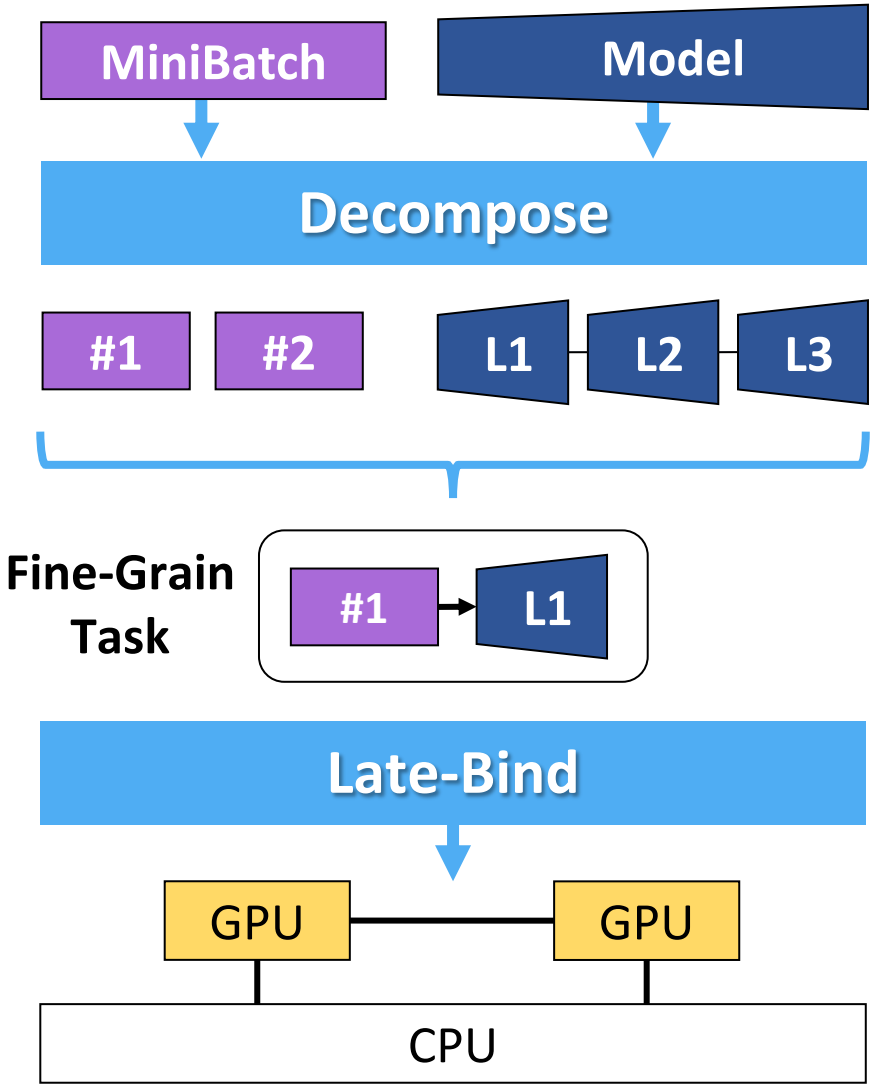
Harmony: A New Framework for Massive Model Training



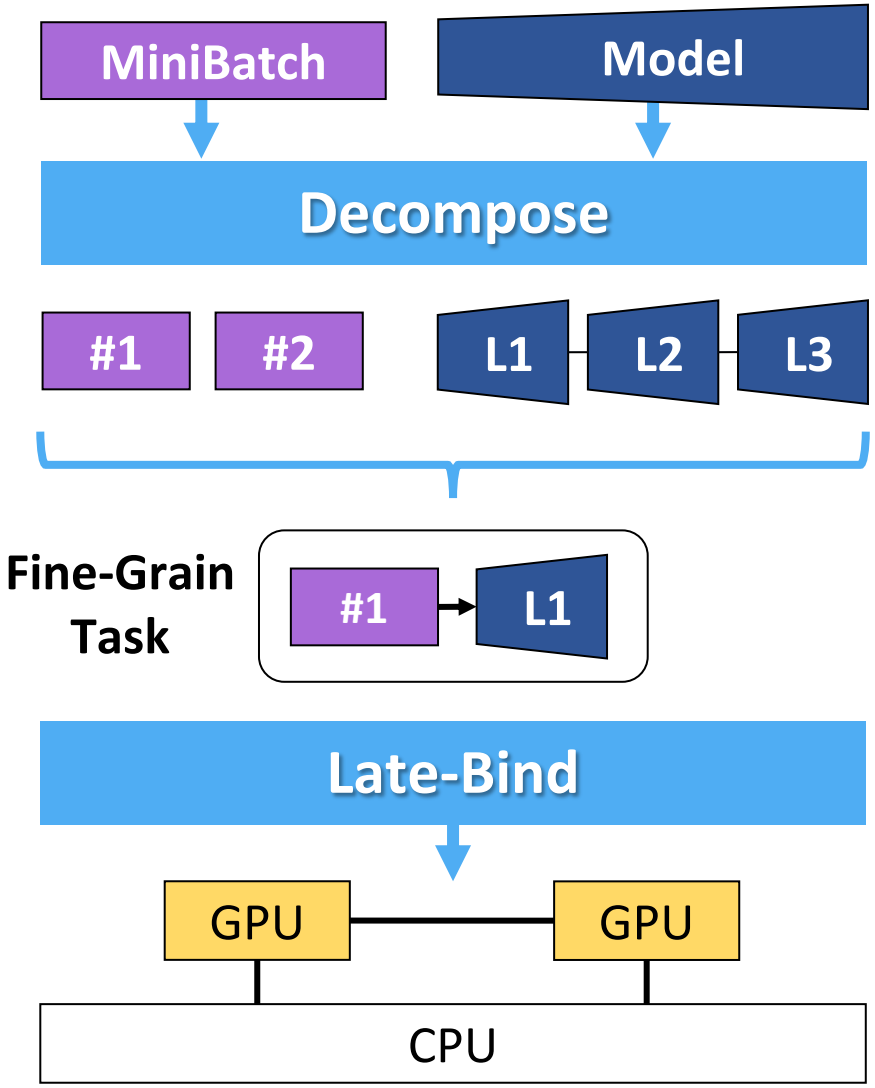
Harmony: A New Framework for Massive Model Training



Harmony: A New Framework for Massive Model Training



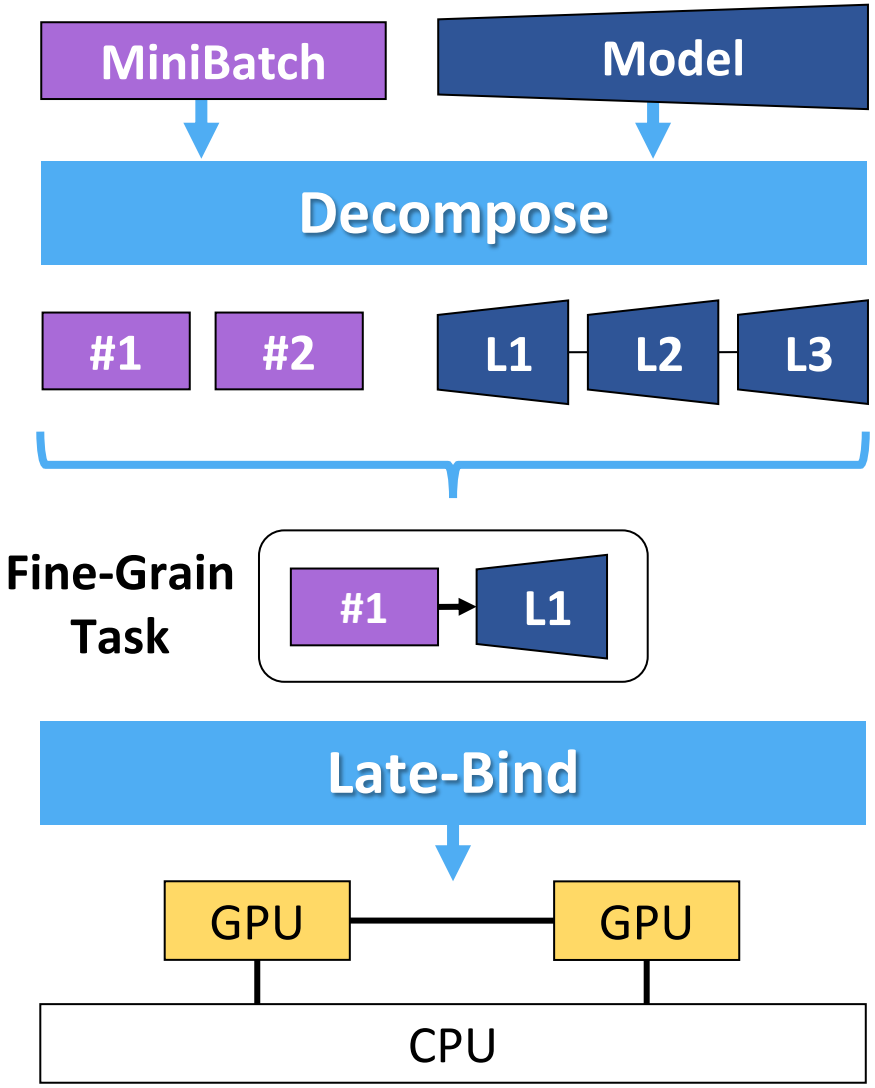
Harmony: A New Framework for Massive Model Training



Performance

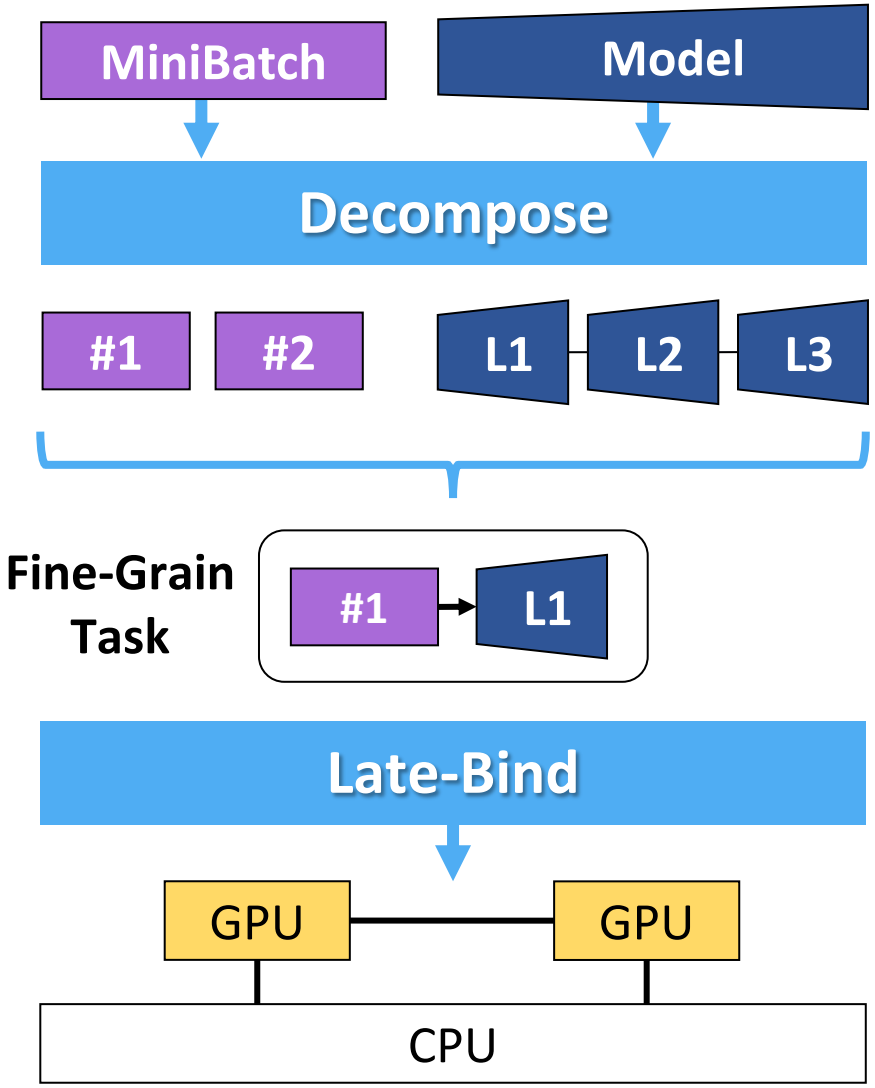
Maximize swap efficiency

Harmony: A New Framework for Massive Model Training



Performance	**Maximize swap efficiency**
Programmability	**Single-GPU abstraction**

Harmony: A New Framework for Massive Model Training

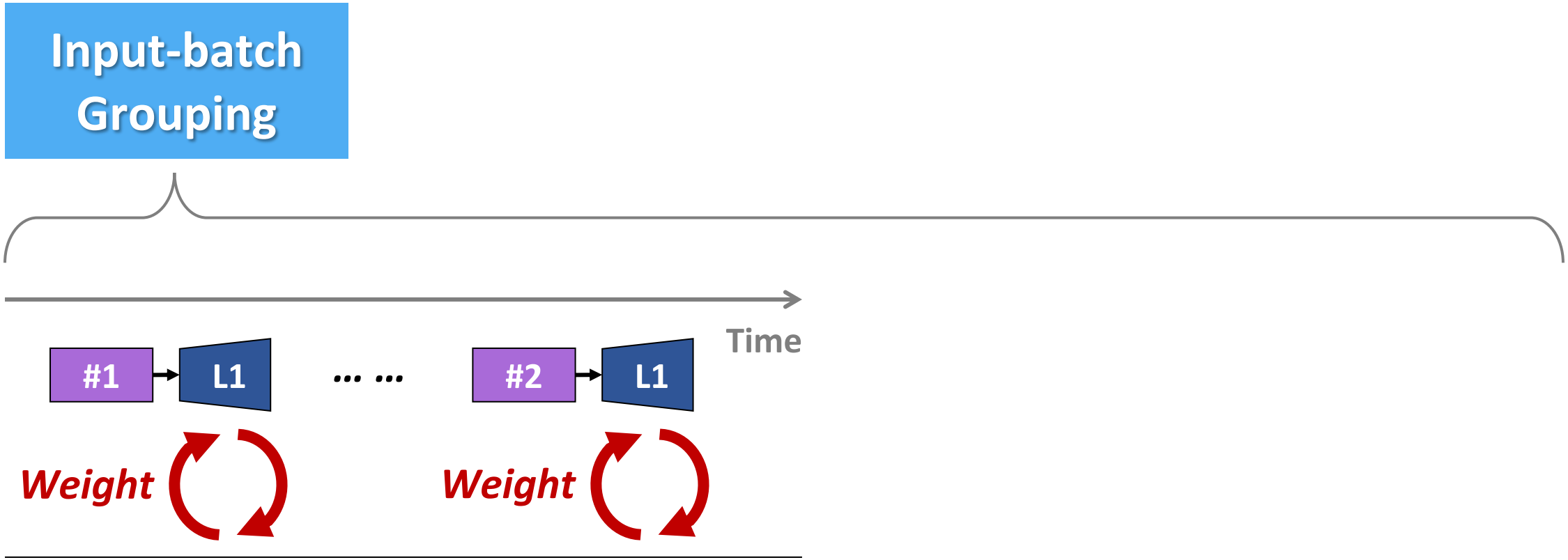


Performance	Maximize swap efficiency
Programmability	Single-GPU abstraction
Flexibility	Reconfigure parallel training

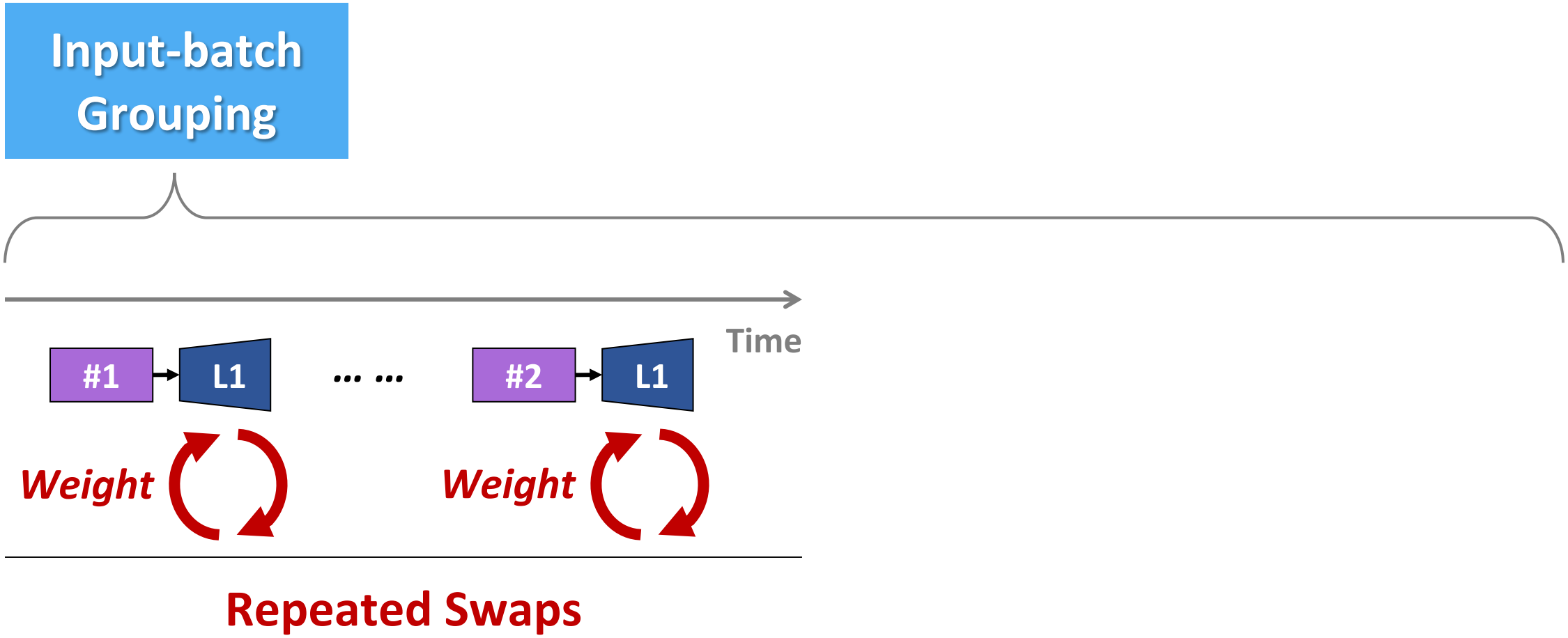
Harmony's Key Performance Optimizations

Input-batch Grouping

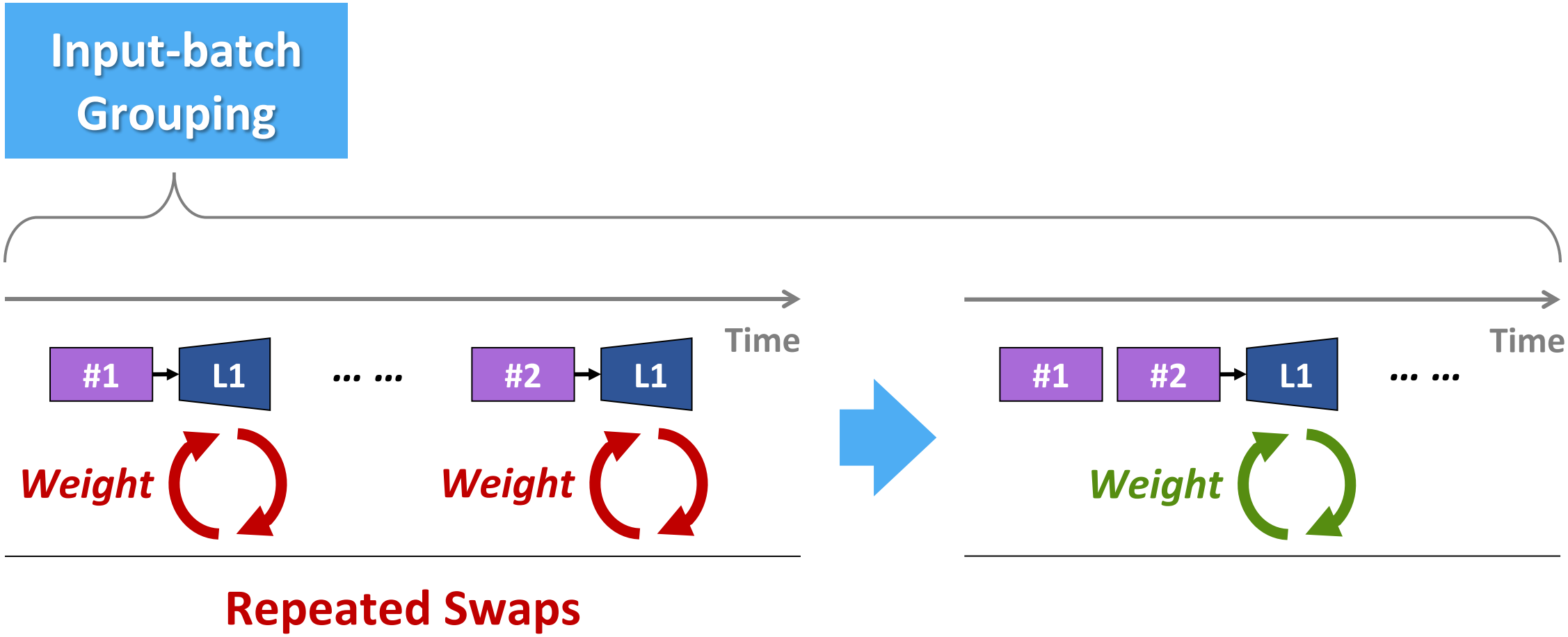
Harmony's Key Performance Optimizations



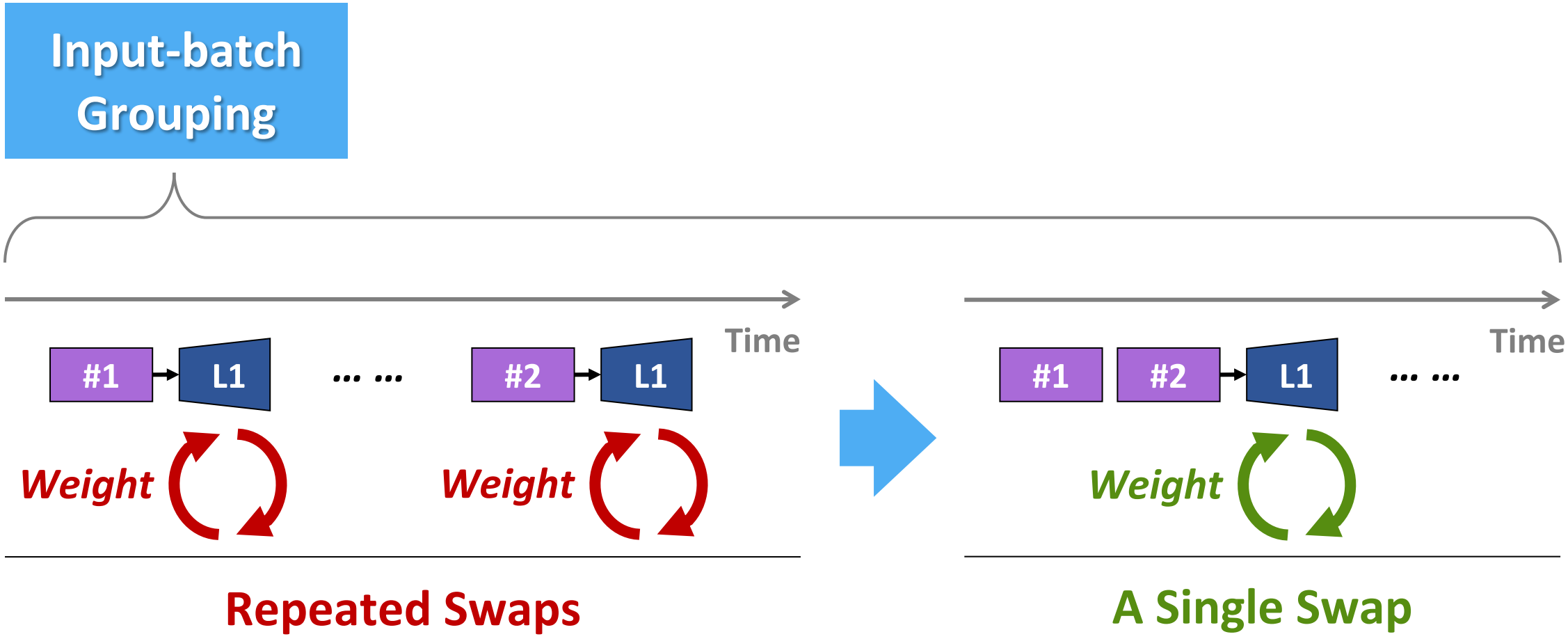
Harmony's Key Performance Optimizations



Harmony's Key Performance Optimizations



Harmony's Key Performance Optimizations



Harmony's Key Performance Optimizations

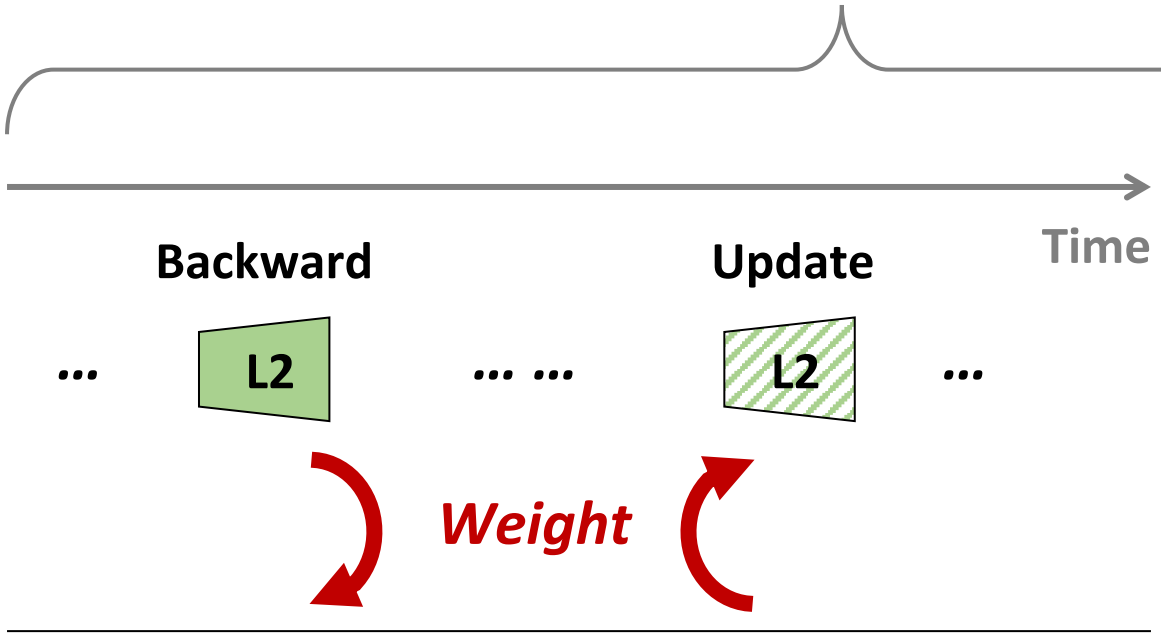
Input-batch Grouping

Just-In-Time Scheduling

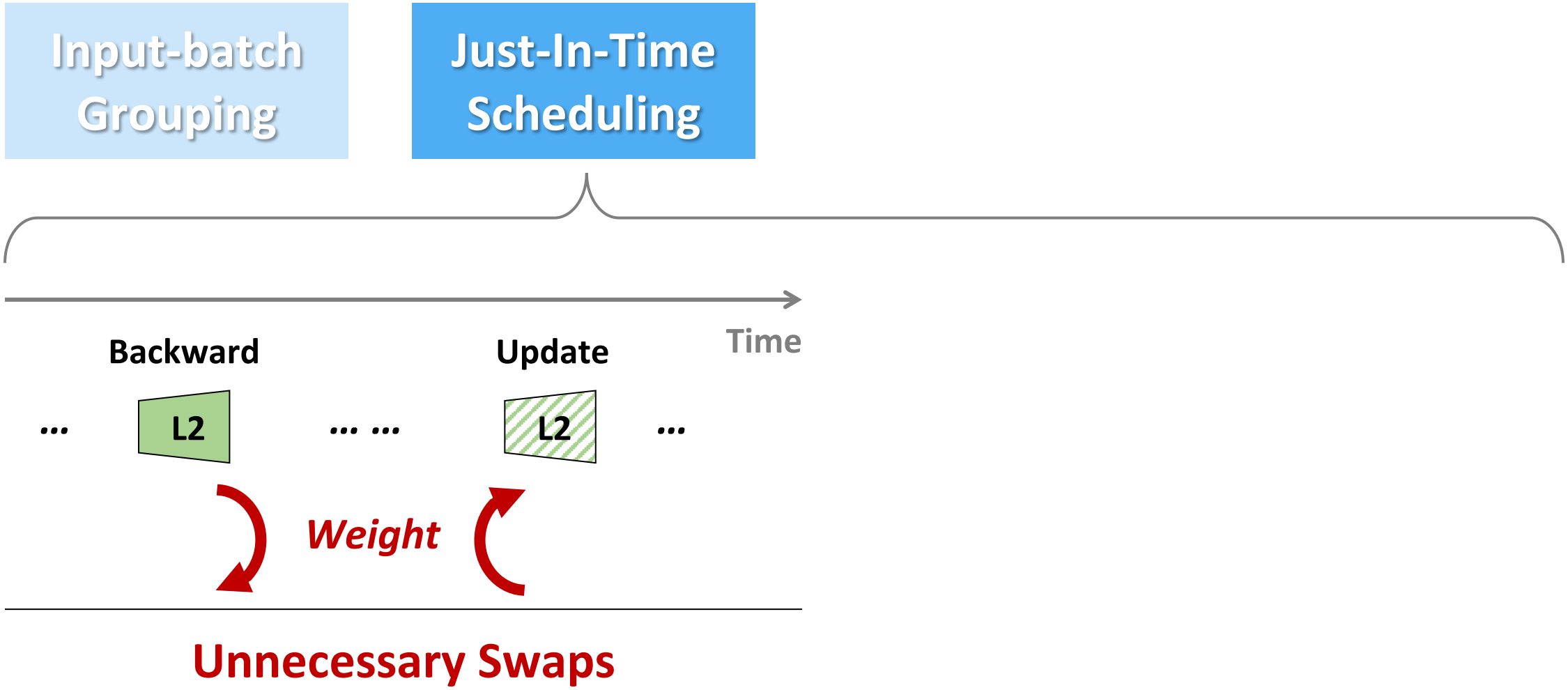
Harmony's Key Performance Optimizations

Input-batch Grouping

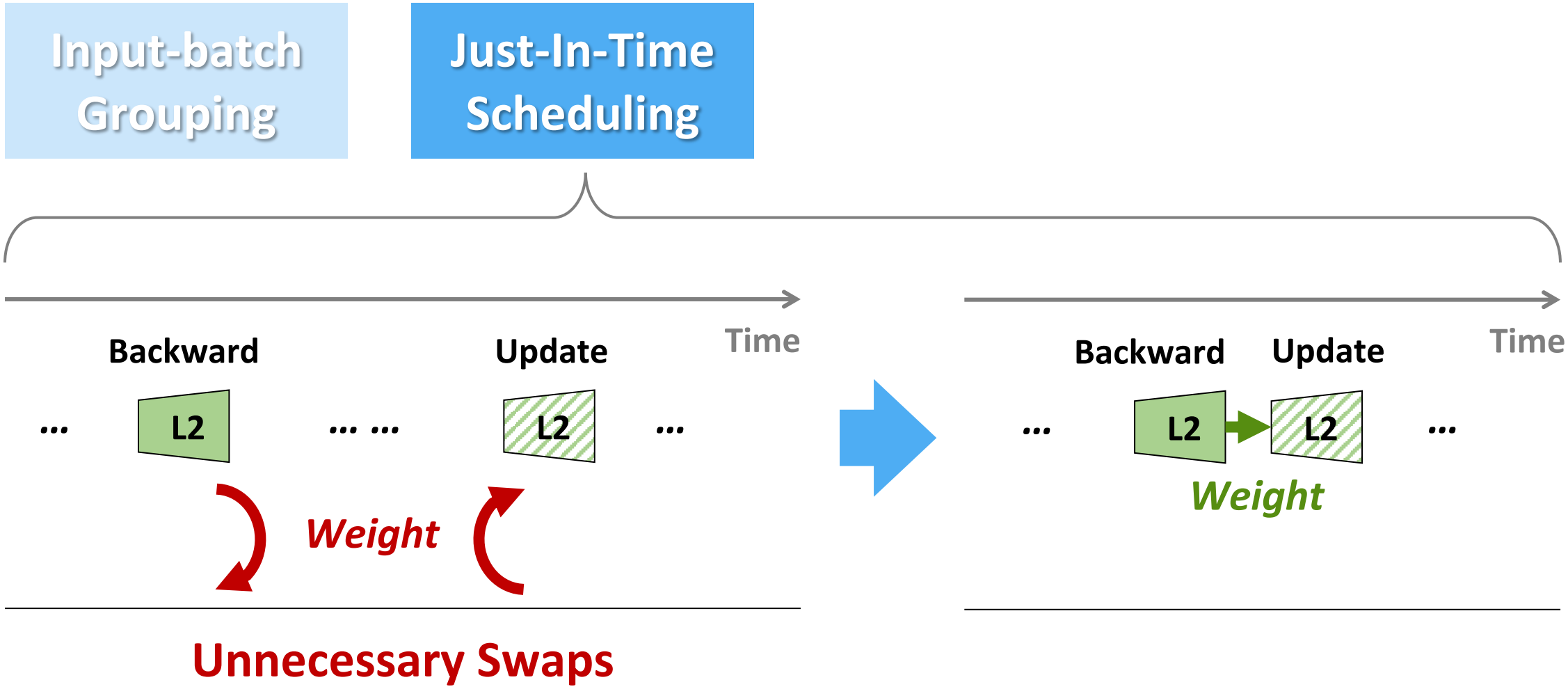
Just-In-Time Scheduling



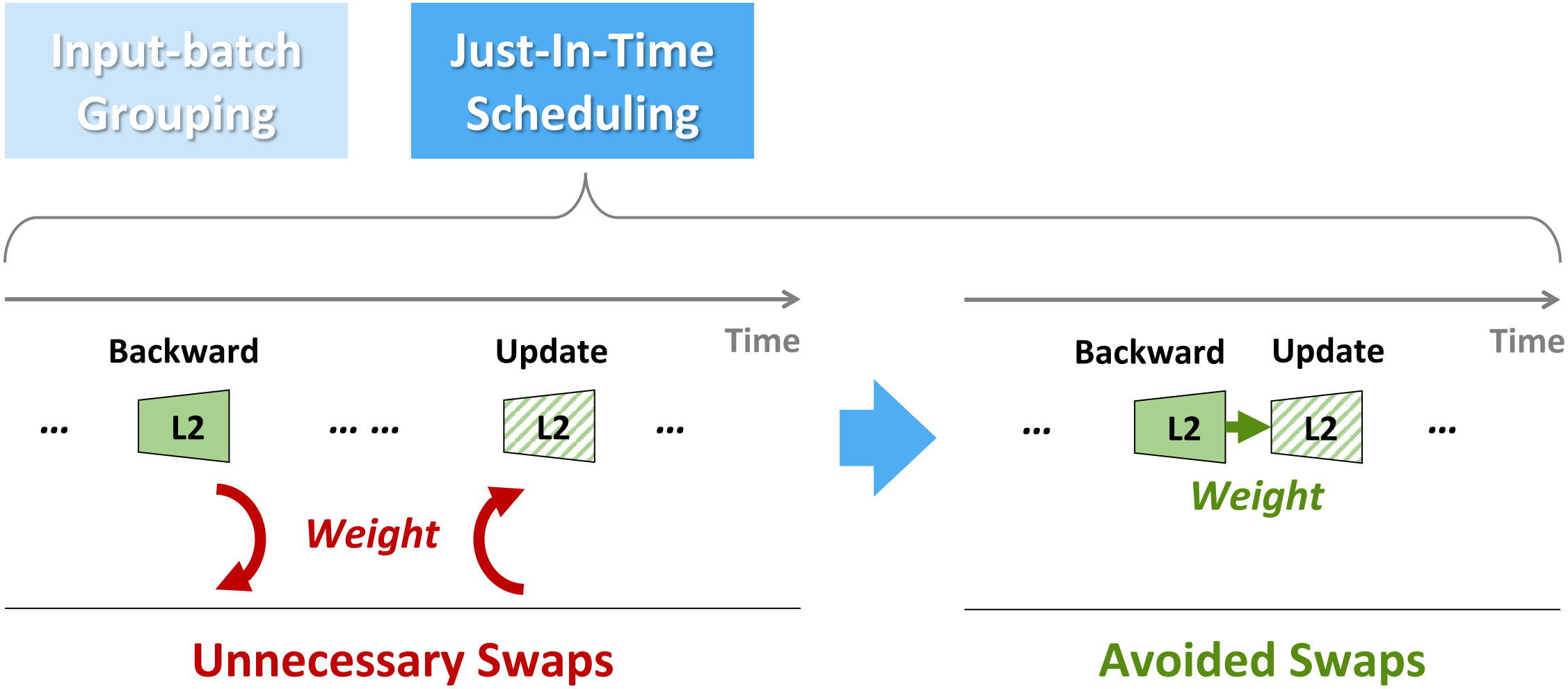
Harmony's Key Performance Optimizations



Harmony's Key Performance Optimizations



Harmony's Key Performance Optimizations



Harmony's Key Performance Optimizations

Input-batch Grouping

Just-In-Time Scheduling

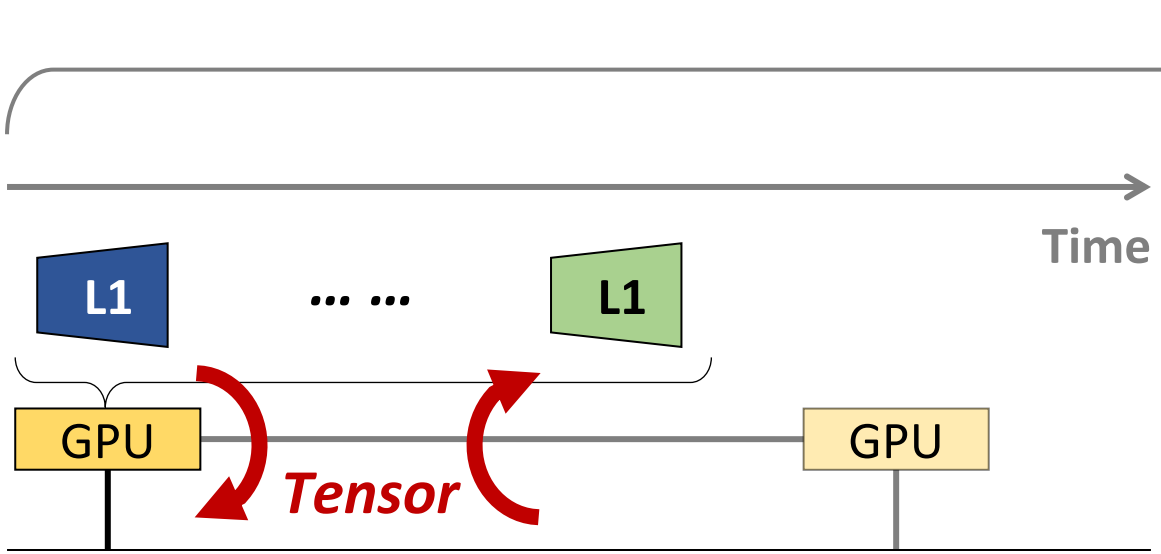
Generalized Peer2Peer Swap

Harmony's Key Performance Optimizations

Input-batch Grouping

Just-In-Time Scheduling

Generalized Peer2Peer Swap

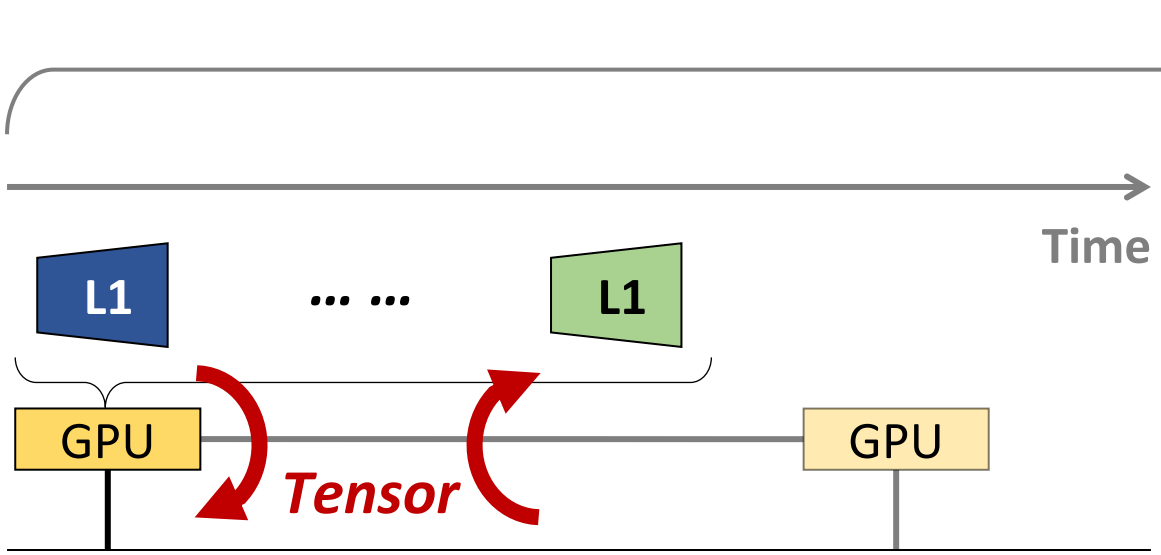


Harmony's Key Performance Optimizations

Input-batch Grouping

Just-In-Time Scheduling

Generalized Peer2Peer Swap



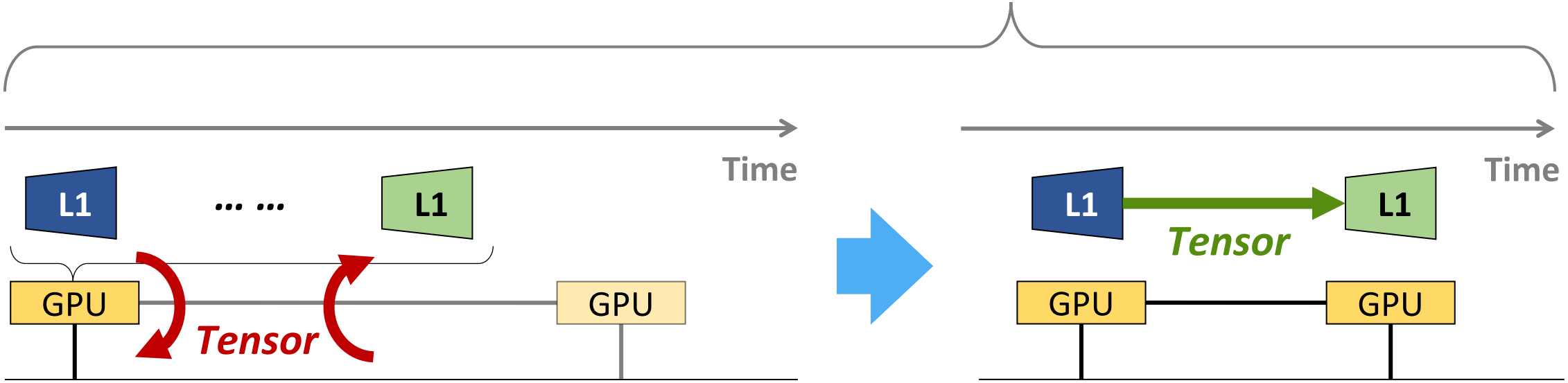
CPU-GPU Swap Bottleneck

Harmony's Key Performance Optimizations

Input-batch Grouping

Just-In-Time Scheduling

Generalized Peer2Peer Swap



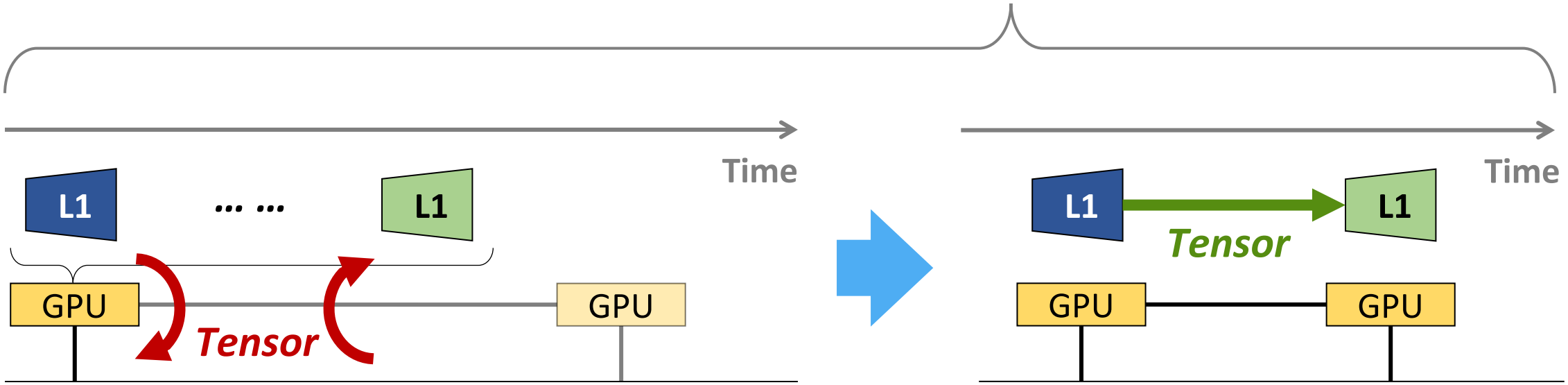
CPU-GPU Swap Bottleneck

Harmony's Key Performance Optimizations

Input-batch Grouping

Just-In-Time Scheduling

Generalized Peer2Peer Swap



CPU-GPU Swap Bottleneck

No CPU-GPU Swaps

Example: Training 6-layer DNN on 2 GPUs with Harmony

Example: Training 6-layer DNN on 2 GPUs with Harmony

Forward Pass	Backward Pass	Weight Update
Microbatch Index Layer IDs	Microbatch Index Layer IDs	Layer IDs

Example: Training 6-layer DNN on 2 GPUs with Harmony

Forward Pass

Microbatch Index

Layer IDs

Input-Batch Grouping

Backward Pass

Microbatch Index

Layer IDs

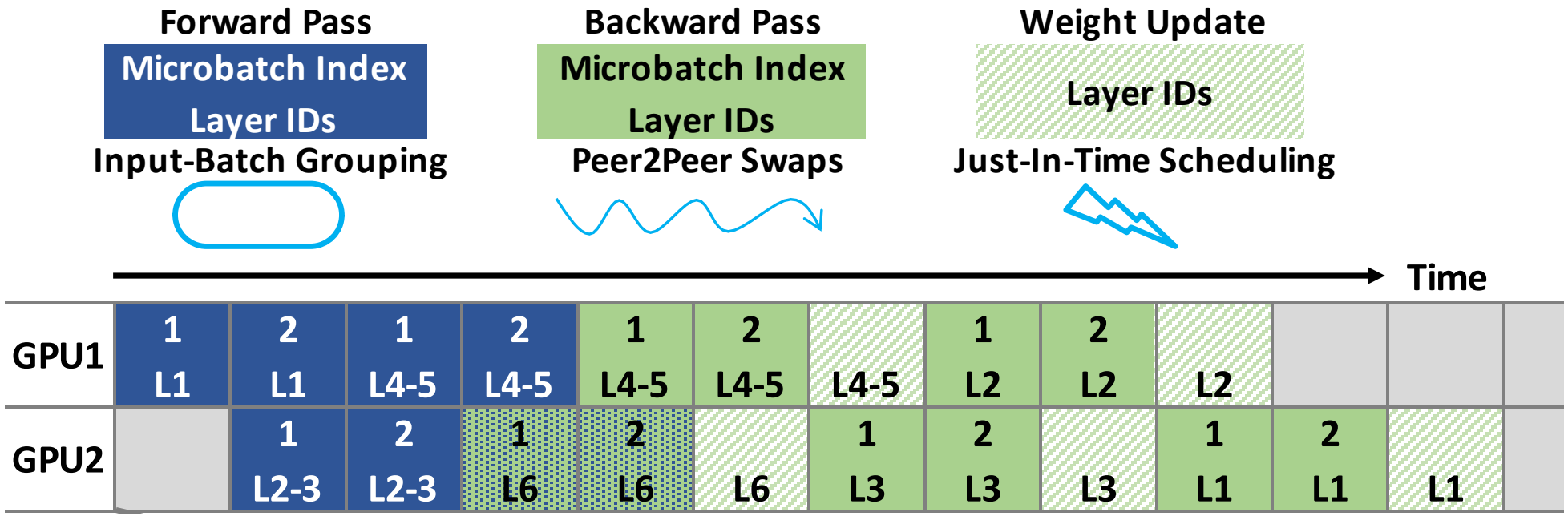
Peer2Peer Swaps

Weight Update

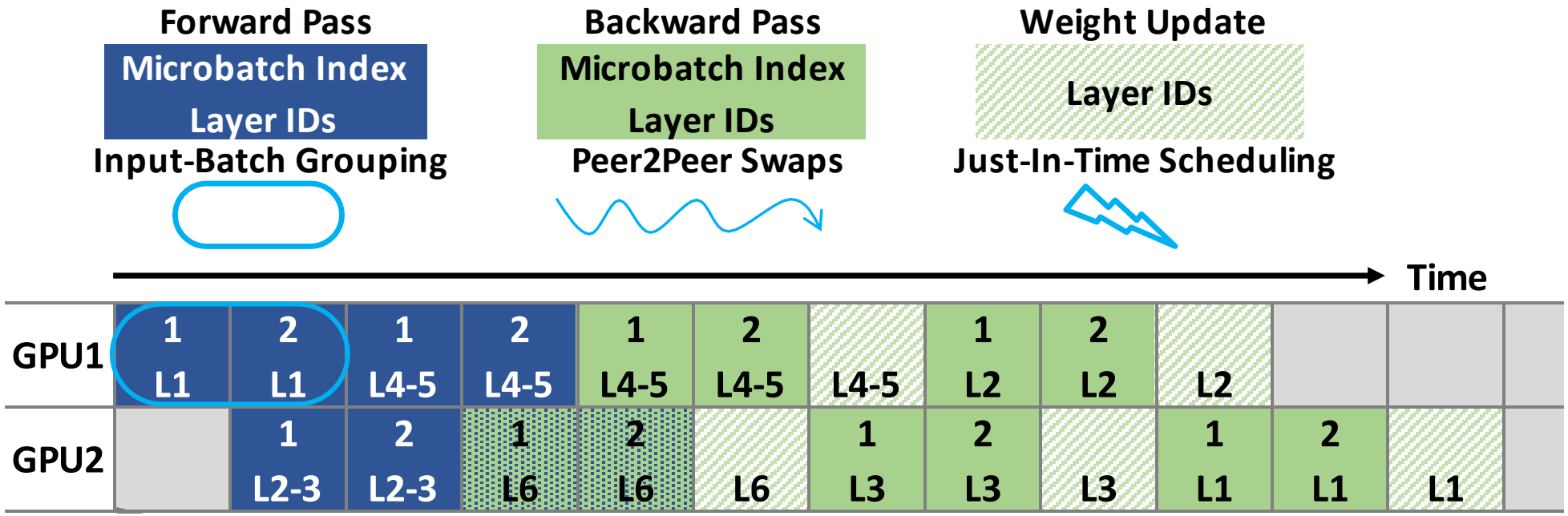
Layer IDs

Just-In-Time Scheduling

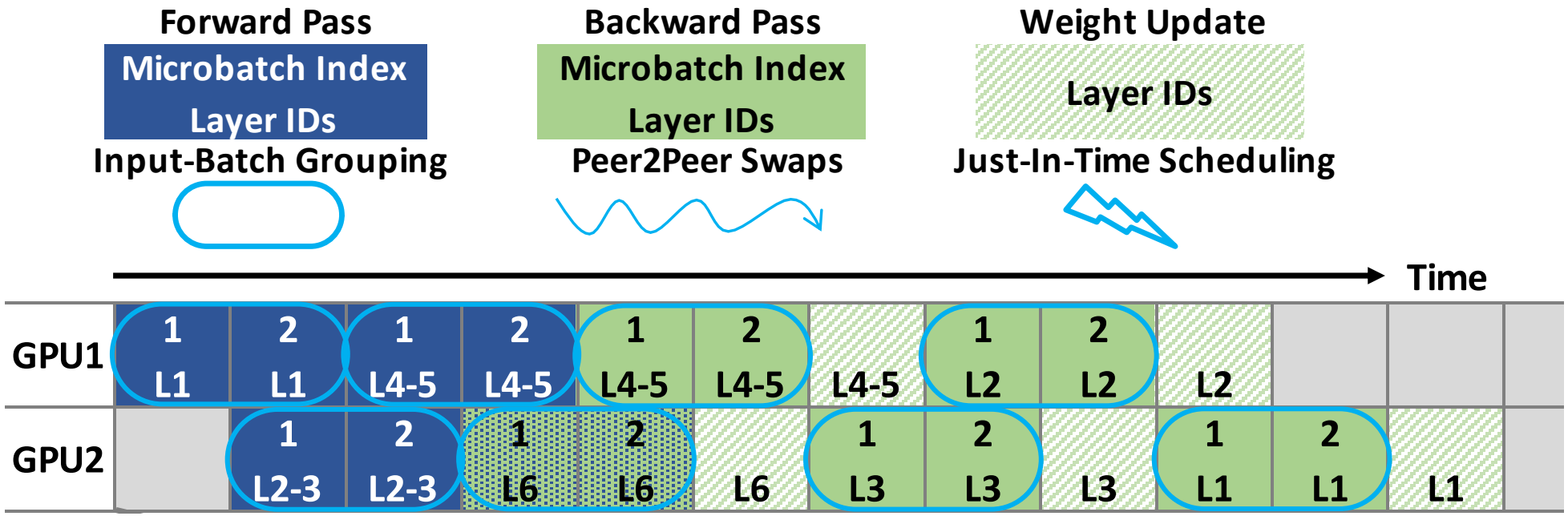
Example: Training 6-layer DNN on 2 GPUs with Harmony



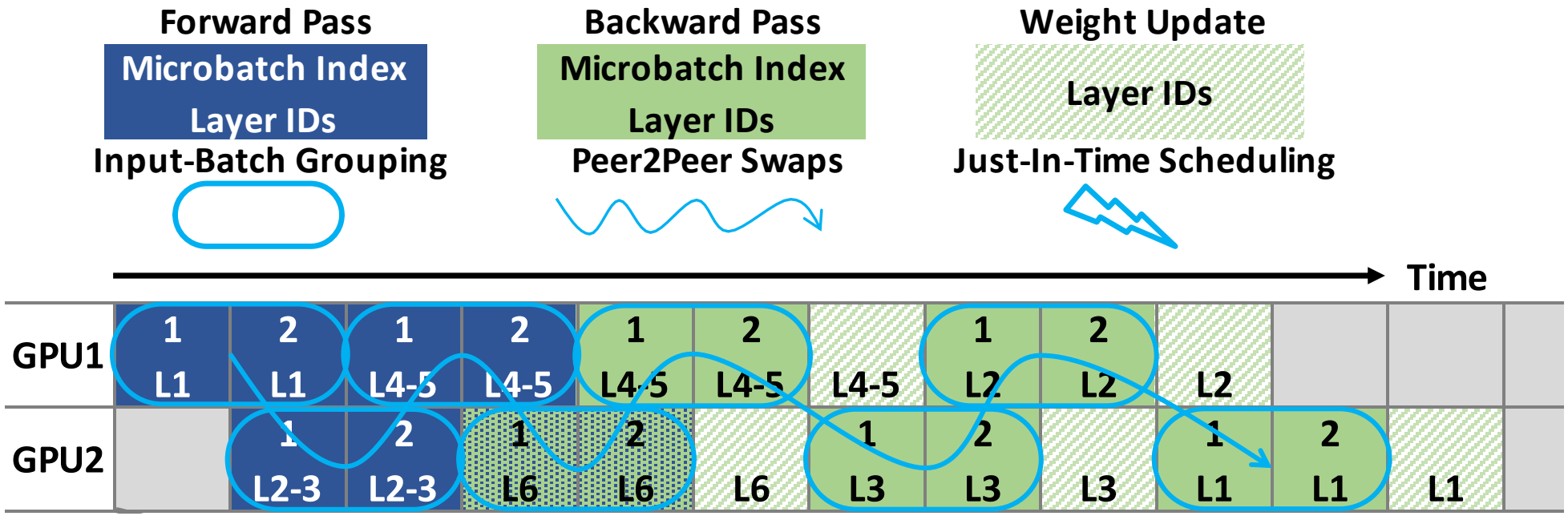
Example: Training 6-layer DNN on 2 GPUs with Harmony



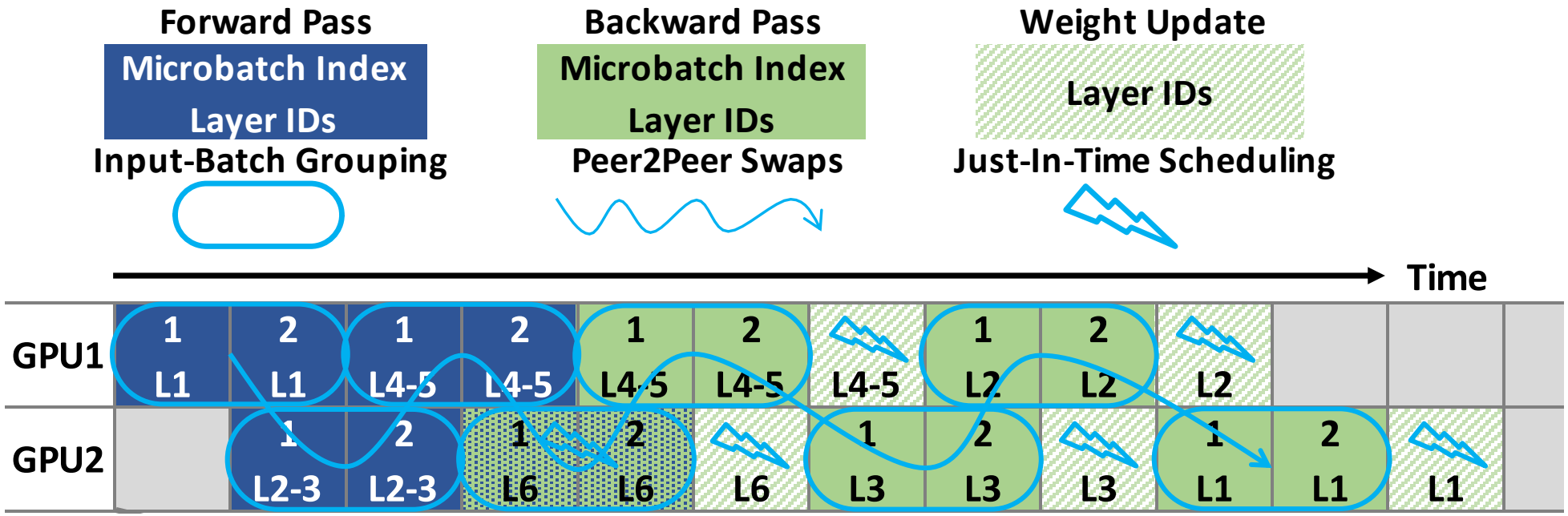
Example: Training 6-layer DNN on 2 GPUs with Harmony



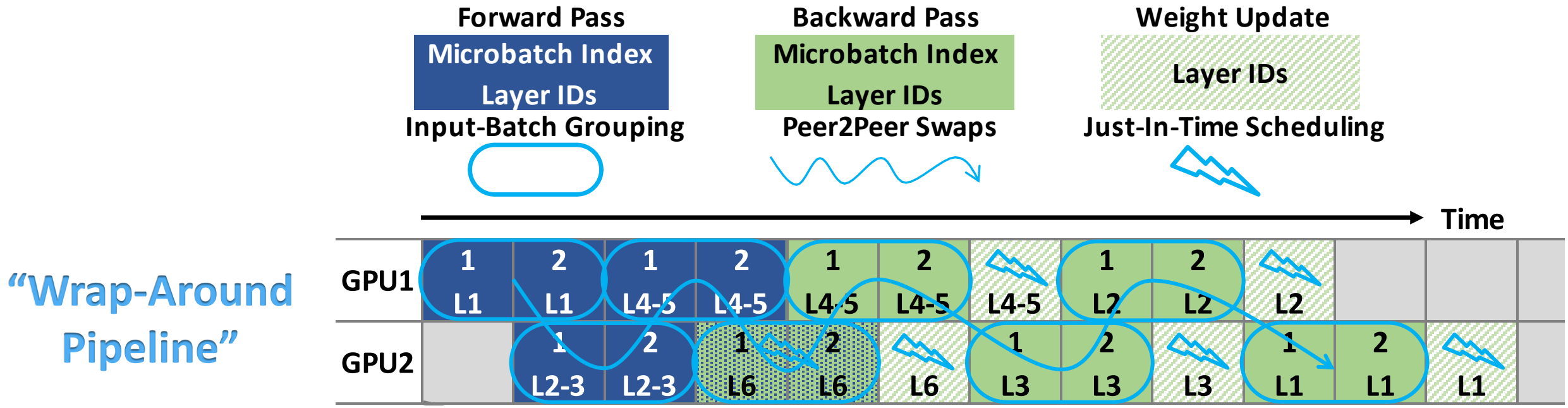
Example: Training 6-layer DNN on 2 GPUs with Harmony



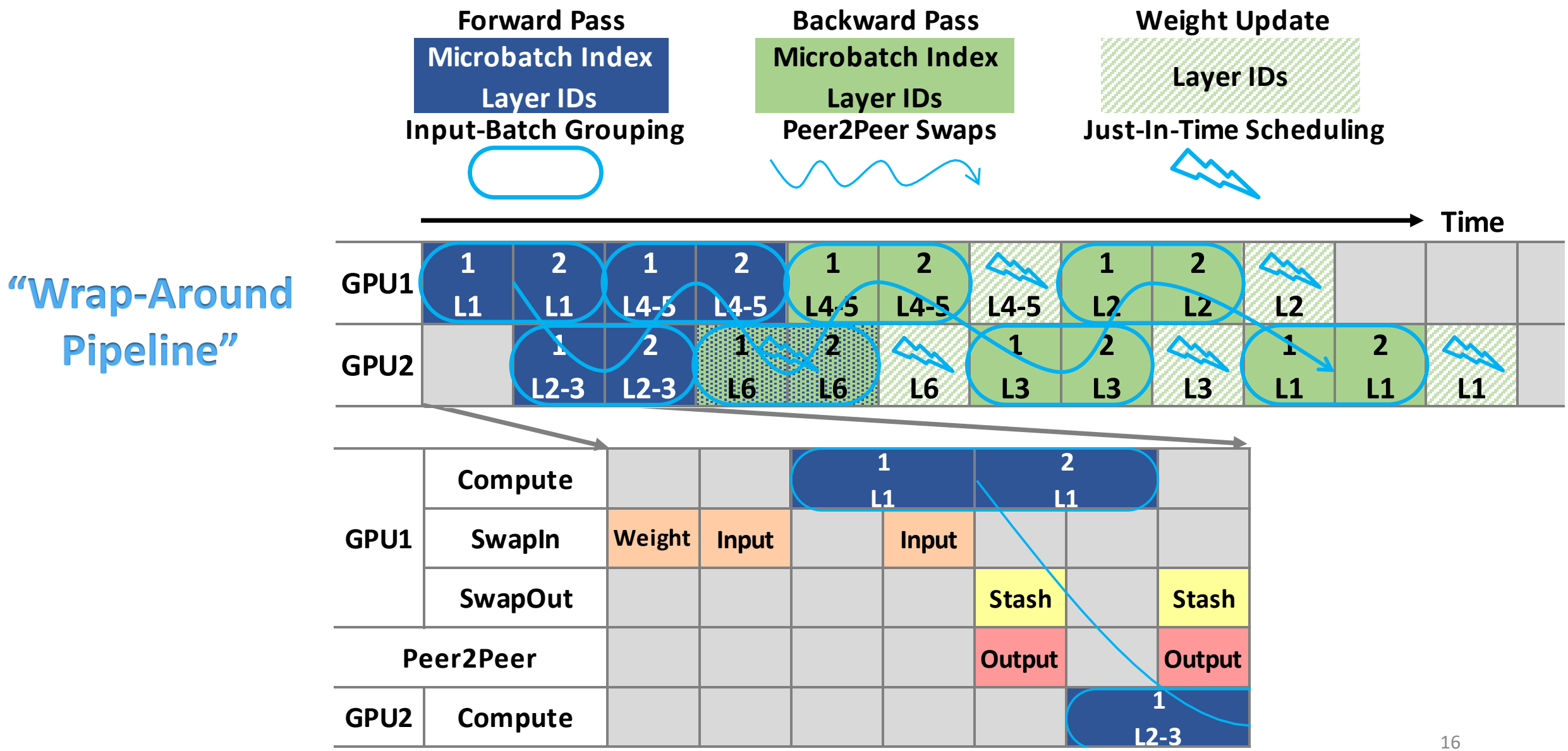
Example: Training 6-layer DNN on 2 GPUs with Harmony



Example: Training 6-layer DNN on 2 GPUs with Harmony



Example: Training 6-layer DNN on 2 GPUs with Harmony



Harmony's Key Performance Optimizations

Input-batch Grouping

Just-In-Time Scheduling

Generalized Peer2Peer Swap

Layer Packing

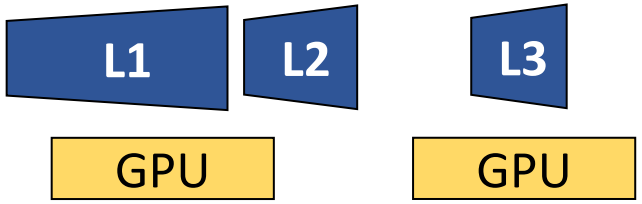
Harmony's Key Performance Optimizations

Input-batch Grouping

Just-In-Time Scheduling

Generalized Peer2Peer Swap

Layer Packing



Unbalanced Load

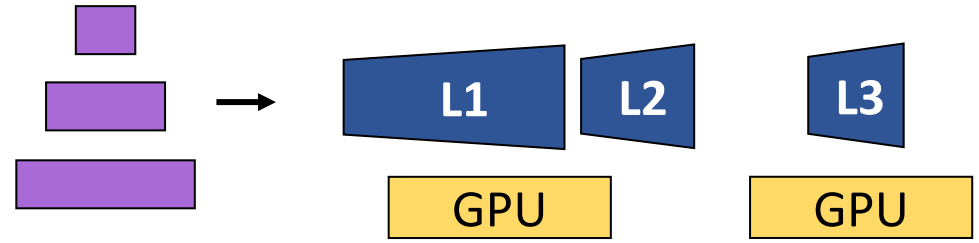
Harmony's Key Performance Optimizations

Input-batch Grouping

Just-In-Time Scheduling

Generalized Peer2Peer Swap

Layer Packing



Different Microbatch Size

Unbalanced Load

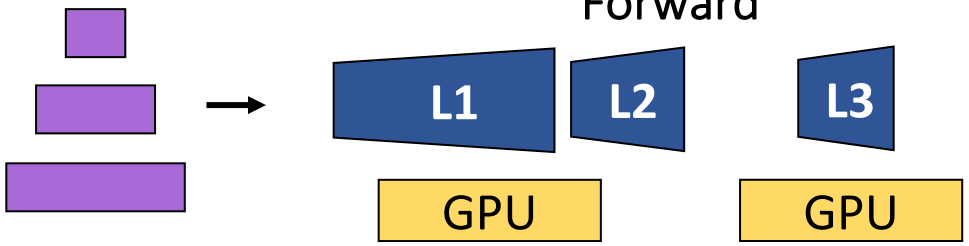
Harmony's Key Performance Optimizations

Input-batch Grouping

Just-In-Time Scheduling

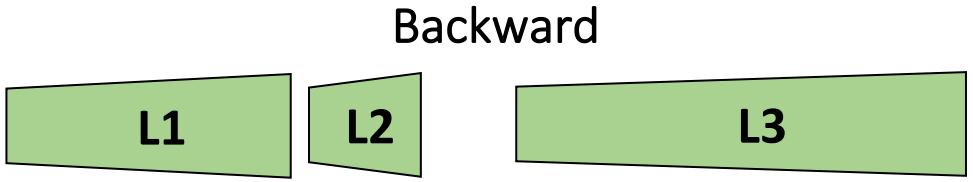
Generalized Peer2Peer Swap

Layer Packing



Different Microbatch Size

Unbalanced Load



Distinct Forward and Backward

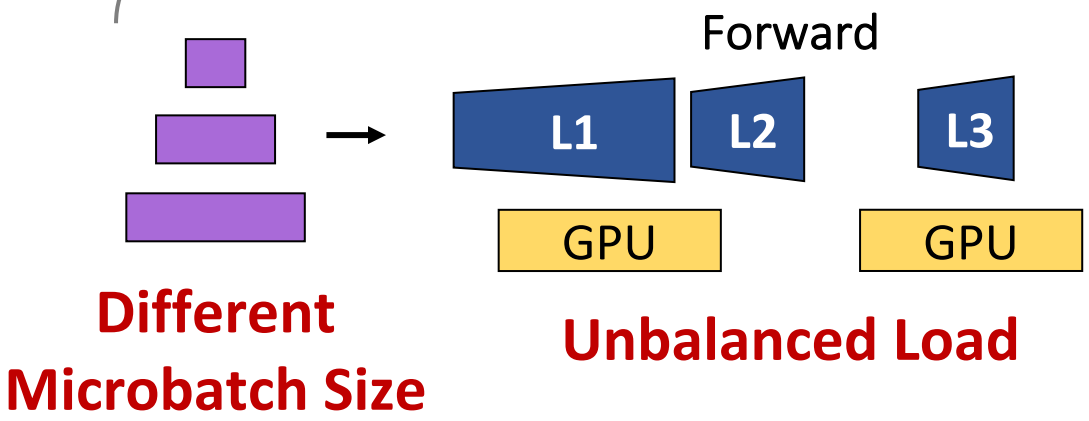
Harmony's Key Performance Optimizations

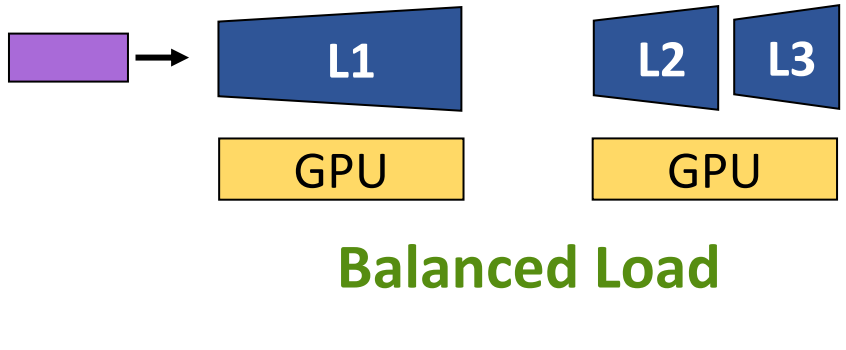
Input-batch Grouping

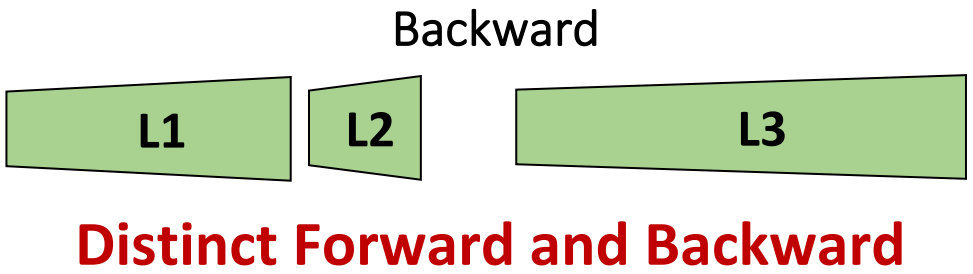
Just-In-Time Scheduling

Generalized Peer2Peer Swap

Layer Packing







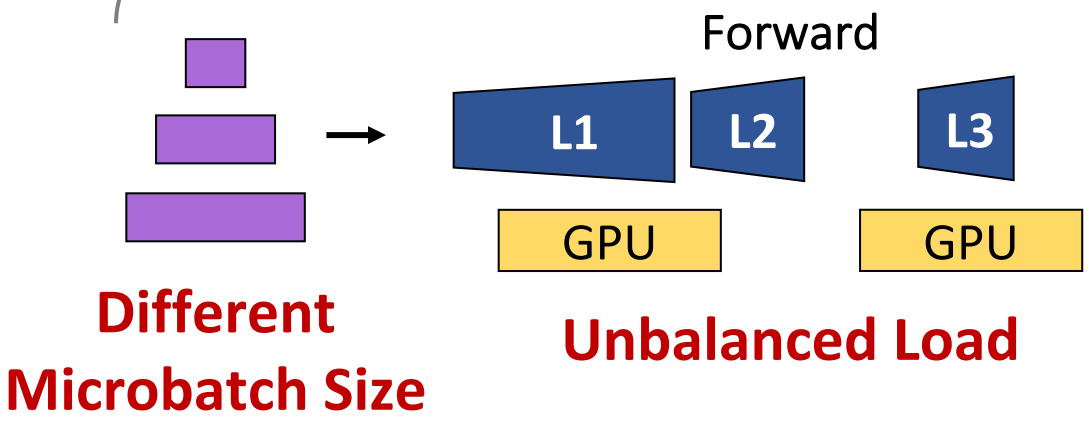
Harmony's Key Performance Optimizations

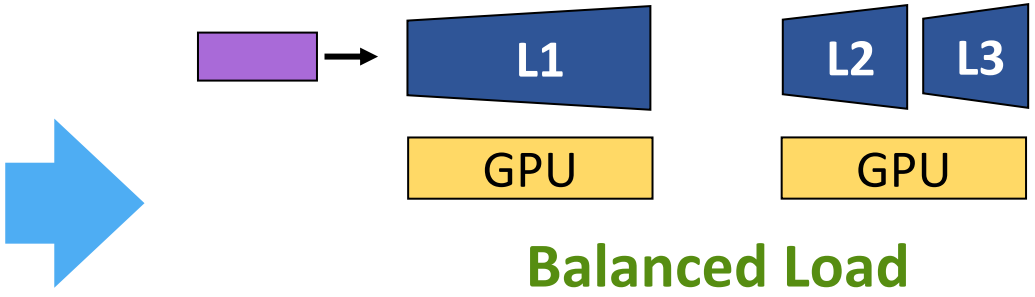
Input-batch Grouping

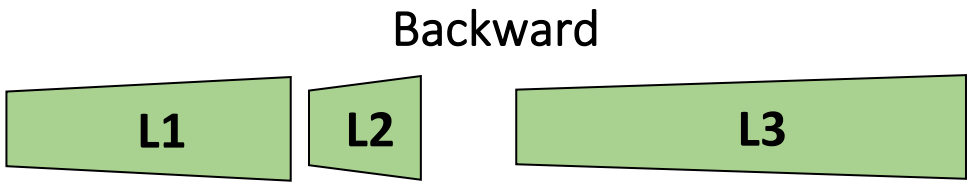
Just-In-Time Scheduling

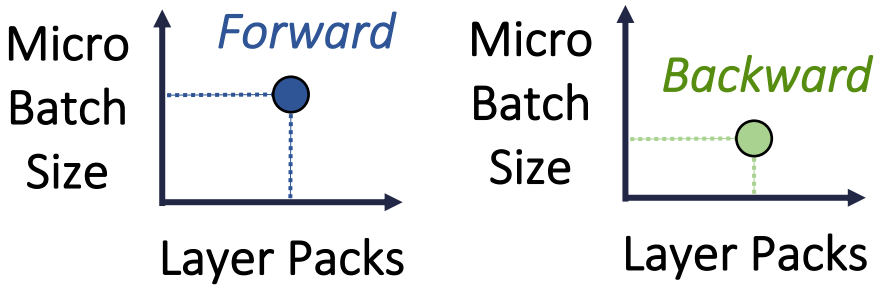
Generalized Peer2Peer Swap

Layer Packing









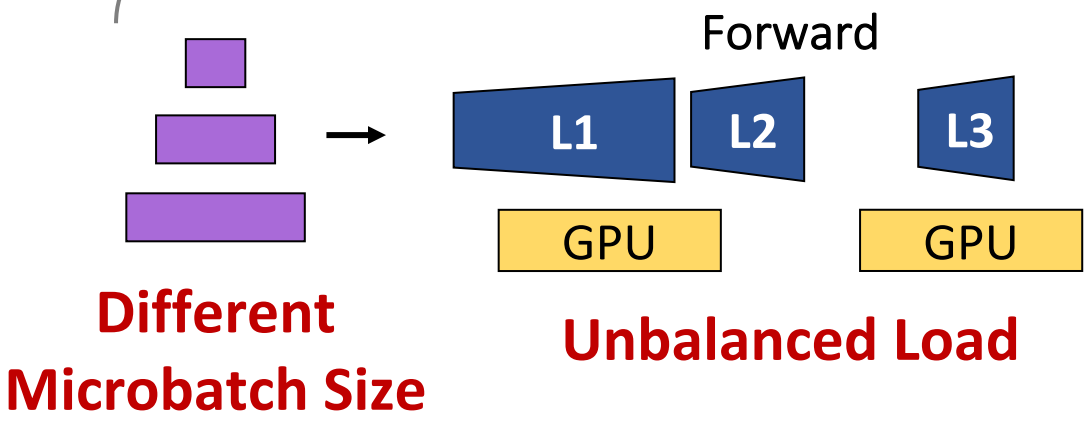
Harmony's Key Performance Optimizations

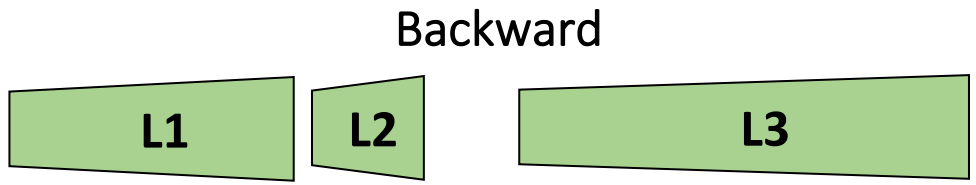
Input-batch Grouping

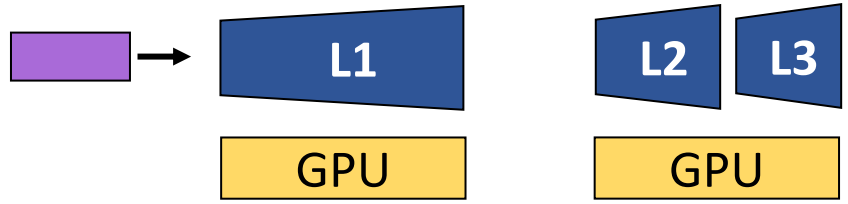
Just-In-Time Scheduling

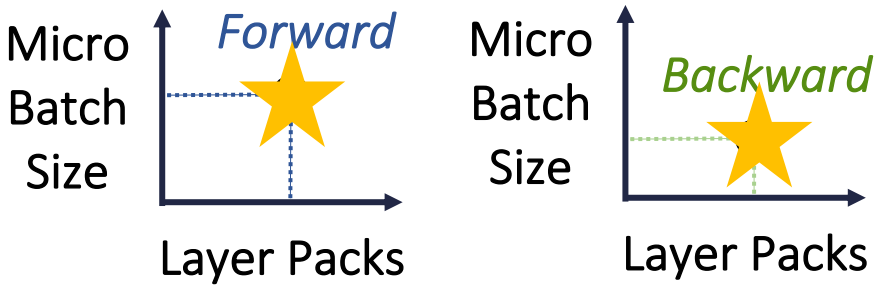
Generalized Peer2Peer Swap

Layer Packing









Best 4-D Configuration

Harmony Implementation and Evaluation

54K LOC

PyTorch 1.5

CUDA 10.1

FP32 Precision

Harmony Implementation and Evaluation

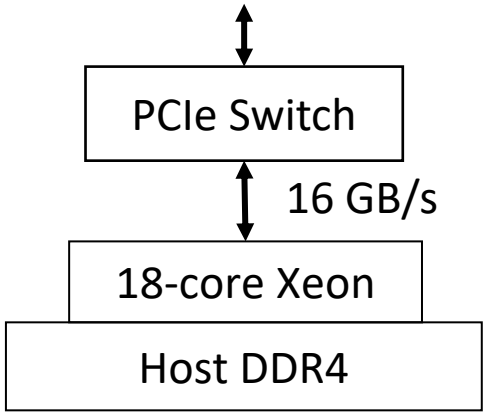
54K LOC

PyTorch 1.5

CUDA 10.1

FP32 Precision

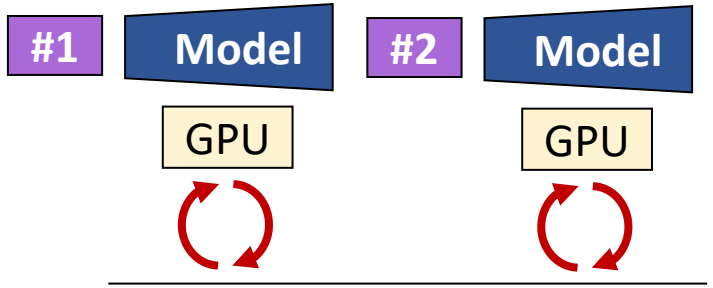
4x GTX-1080Ti (11GB)



Highly Optimized Baselines[+]

[+] *with Various Memory Optimizations*

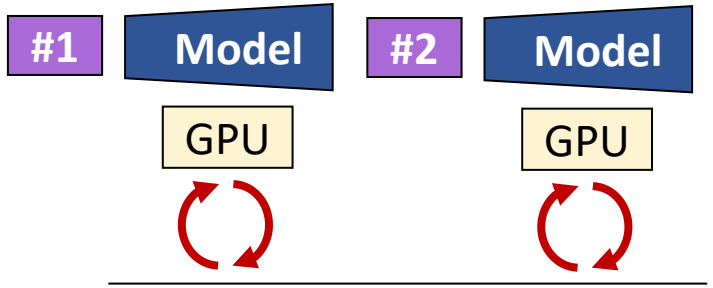
Data-**P**arallelism + Per-GPU **Swap***



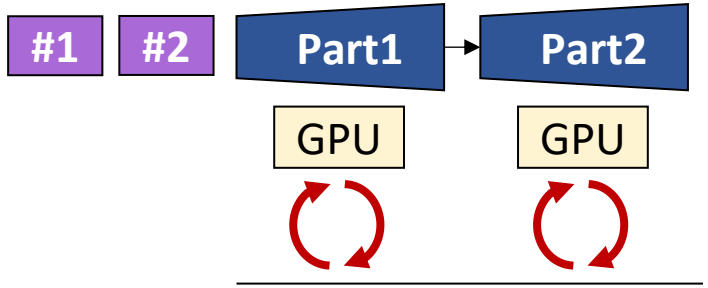
Highly Optimized Baselines+

+ with Various Memory Optimizations

Data-**P**arallelism + Per-GPU **Swap***



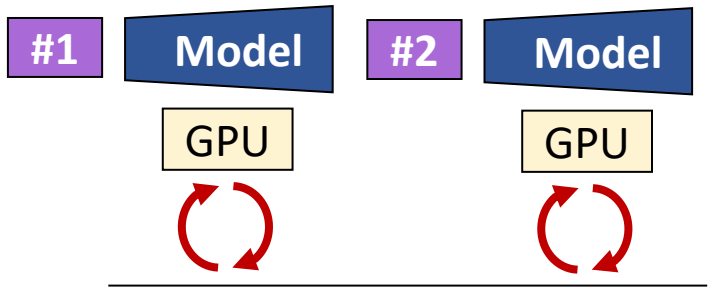
GPipe + Per-GPU **Swap***



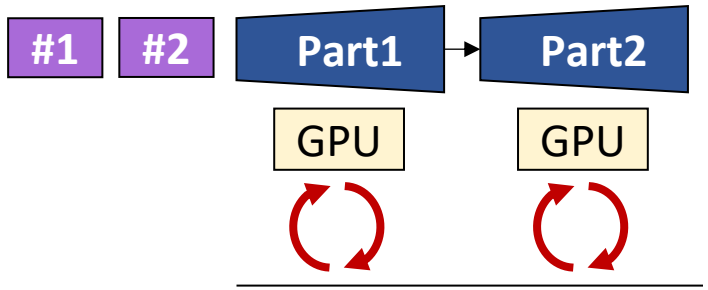
Highly Optimized Baselines[+]

[+] *with Various Memory Optimizations*

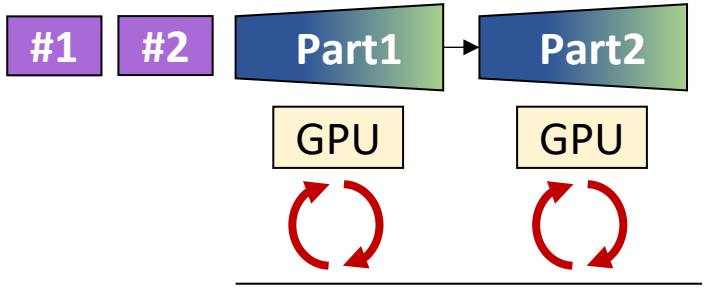
Data-**P**arallelism + Per-GPU **Swap***



GPipe + Per-GPU **Swap***



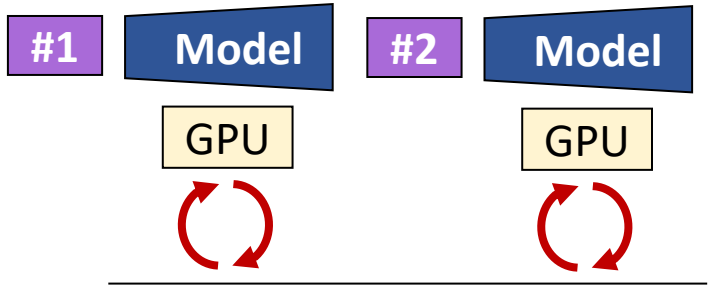
PipeDream**2BW** + Per-GPU **Swap***



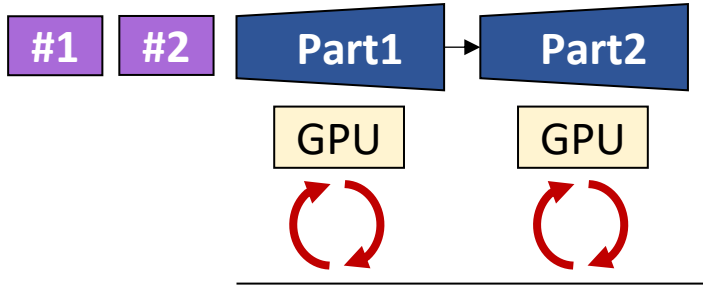
Highly Optimized Baselines+

+ with Various Memory Optimizations

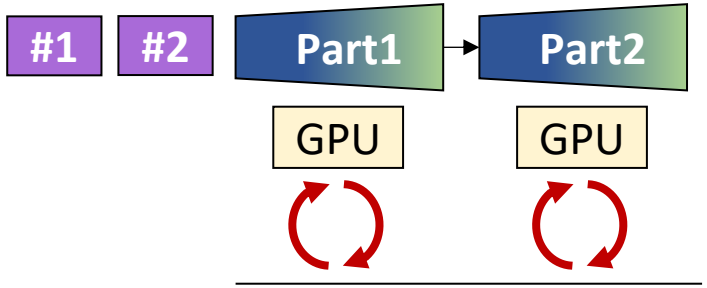
Data-**P**arallelism + Per-GPU **Swap***



GPipe + Per-GPU **Swap***



PipeDream**2BW** + Per-GPU **Swap***

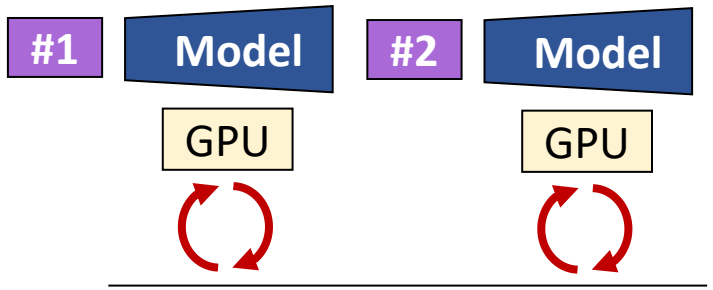


Highly Optimized Baselines+

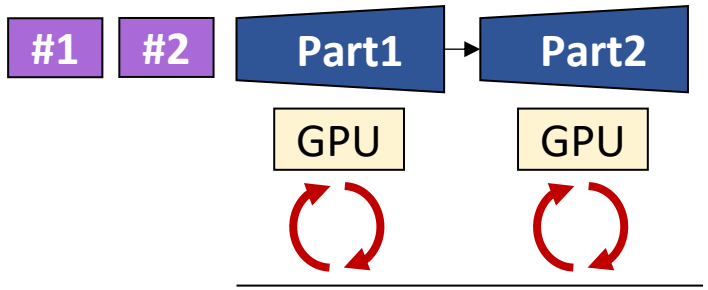
+ with Various Memory Optimizations

**IBM's Large-Model-Support Library*

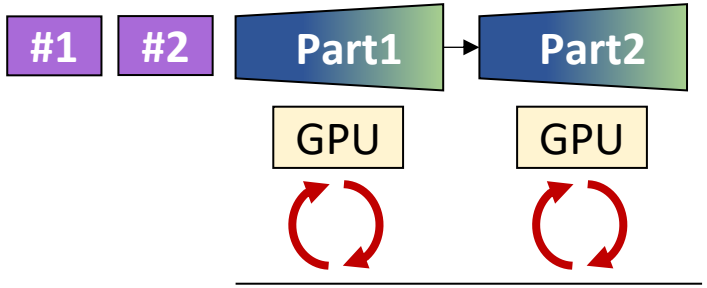
Data-Parallelism + Per-GPU Swap*



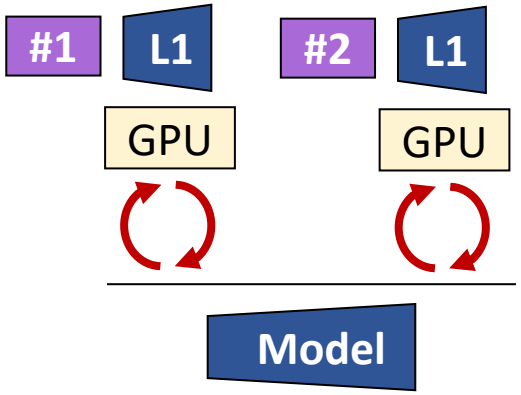
GPipe + Per-GPU Swap*



PipeDream**2BW** + Per-GPU **Swap***



ZeRO Infinity



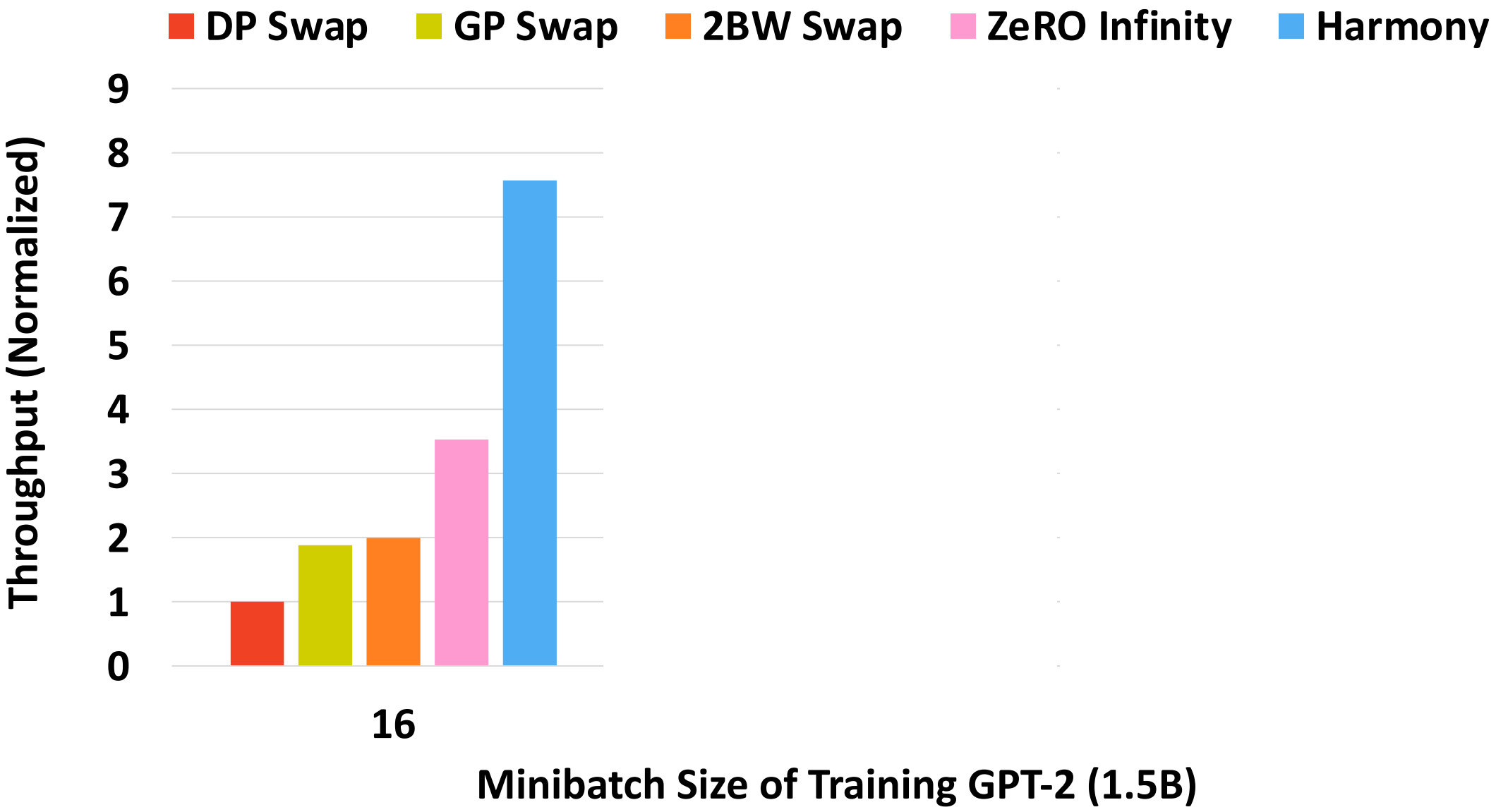
Highly Optimized Baselines+

+ with Various Memory Optimizations

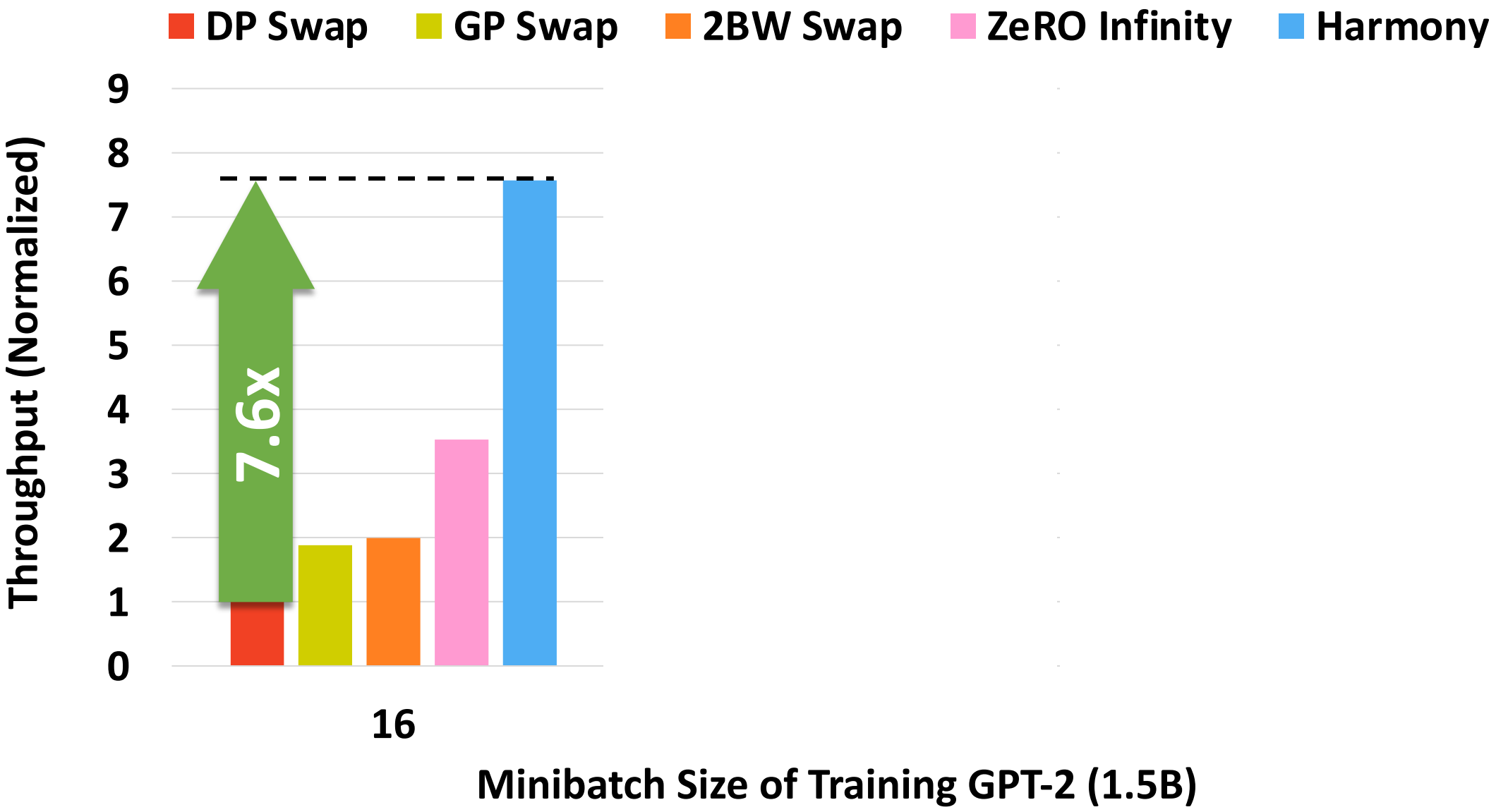
**IBM's Large-Model-Support Library*

Boosting Training Throughput with Harmony

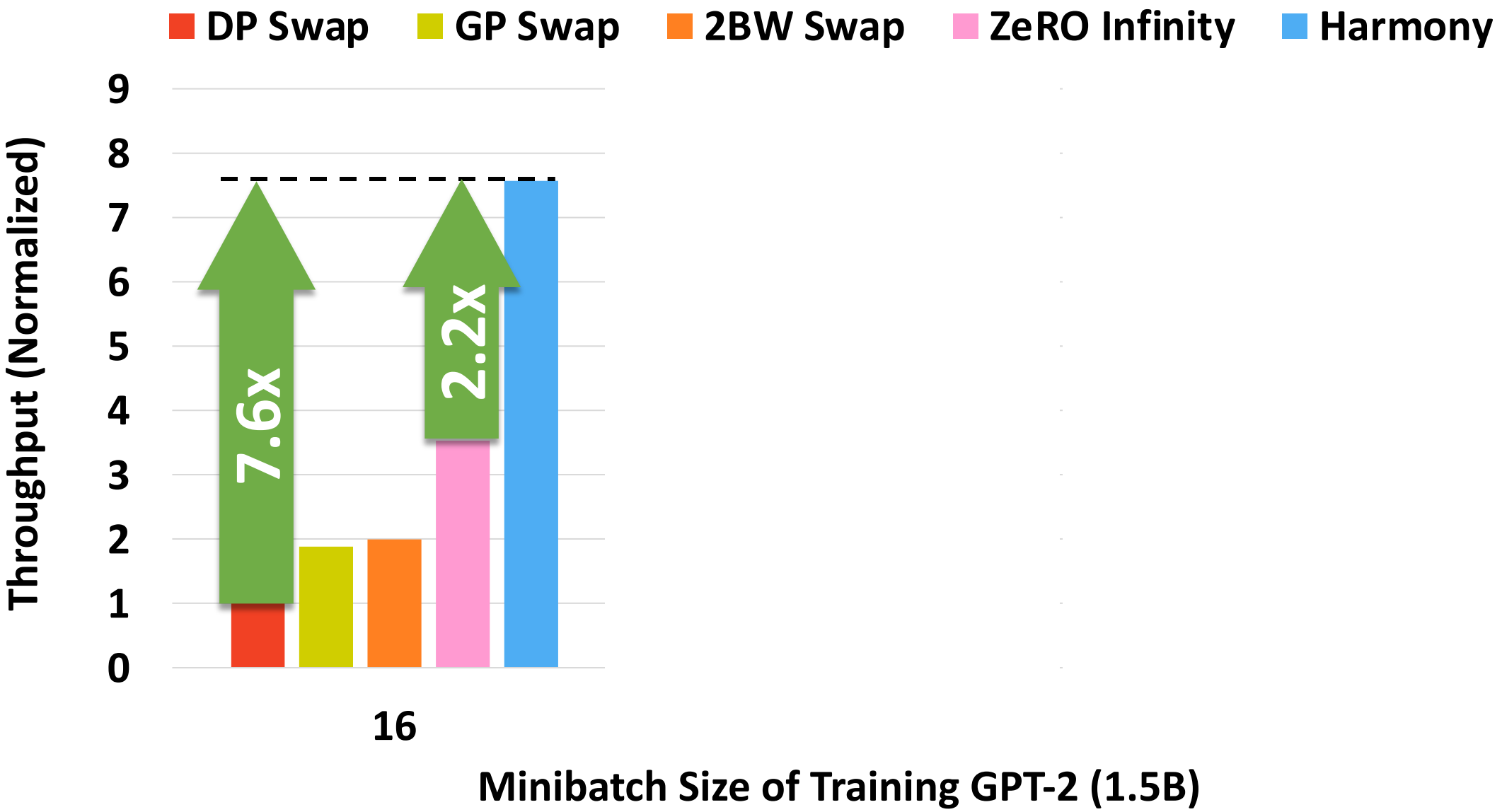
Boosting Training Throughput with Harmony



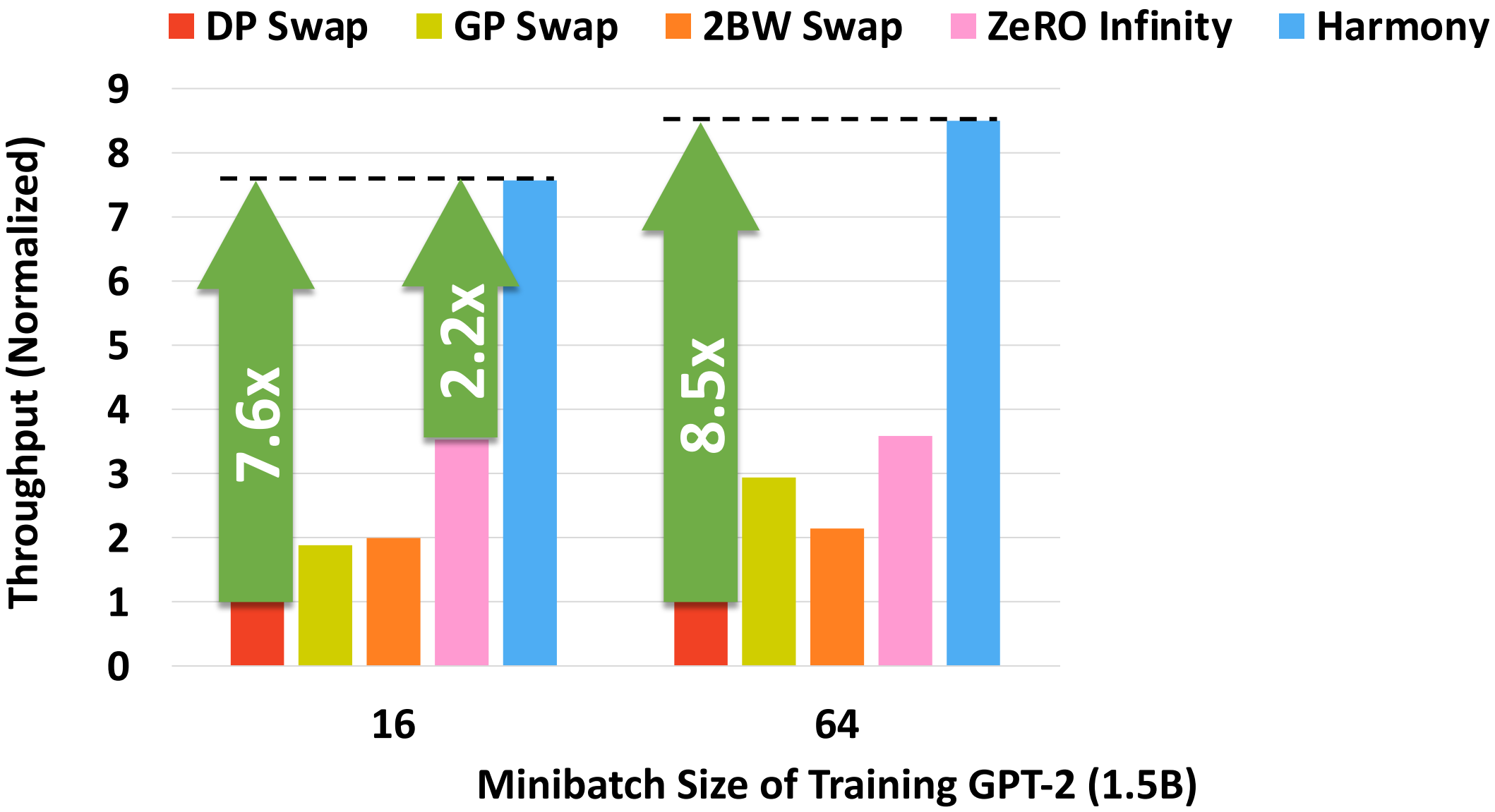
Boosting Training Throughput with Harmony



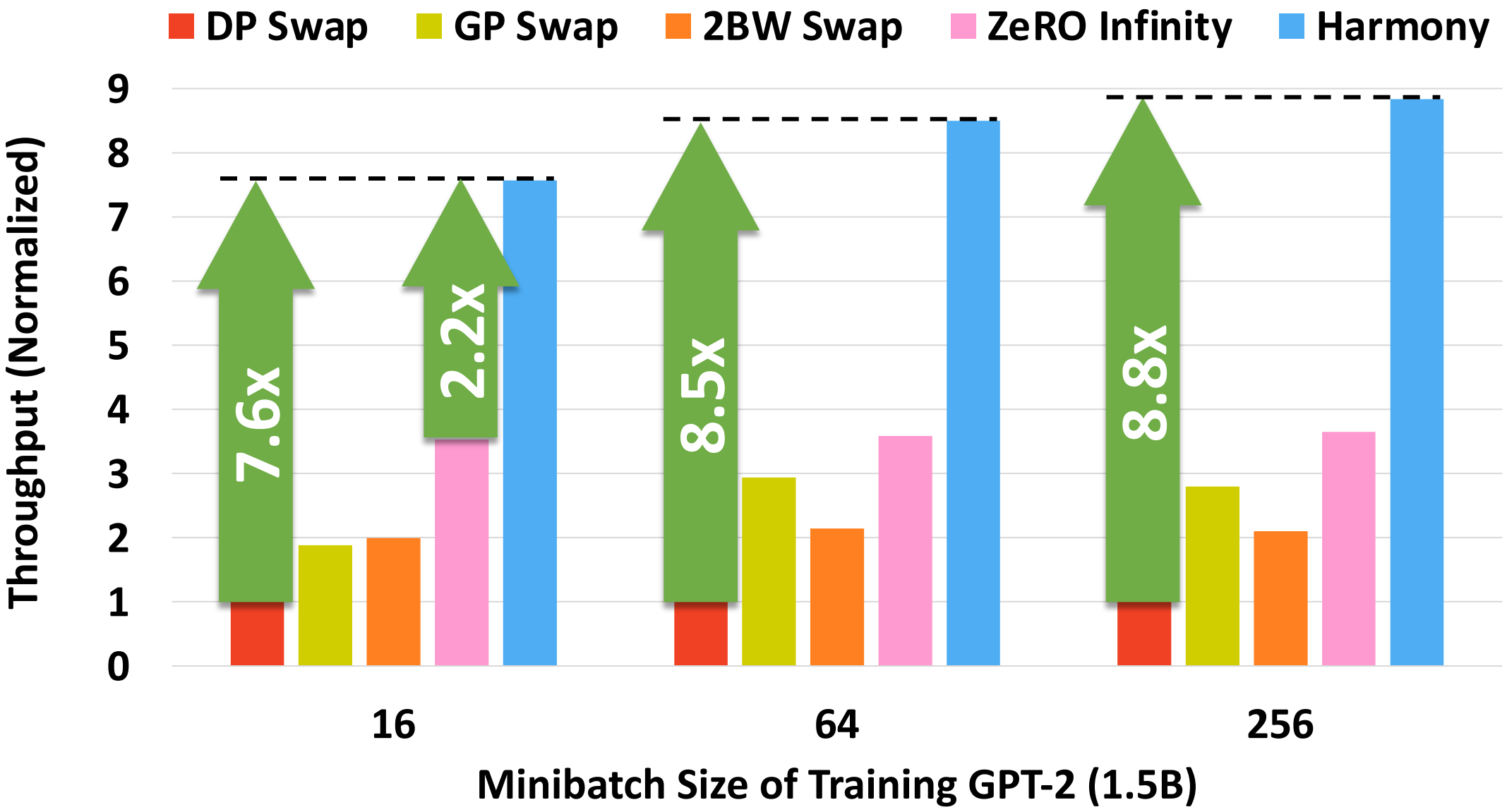
Boosting Training Throughput with Harmony



Boosting Training Throughput with Harmony

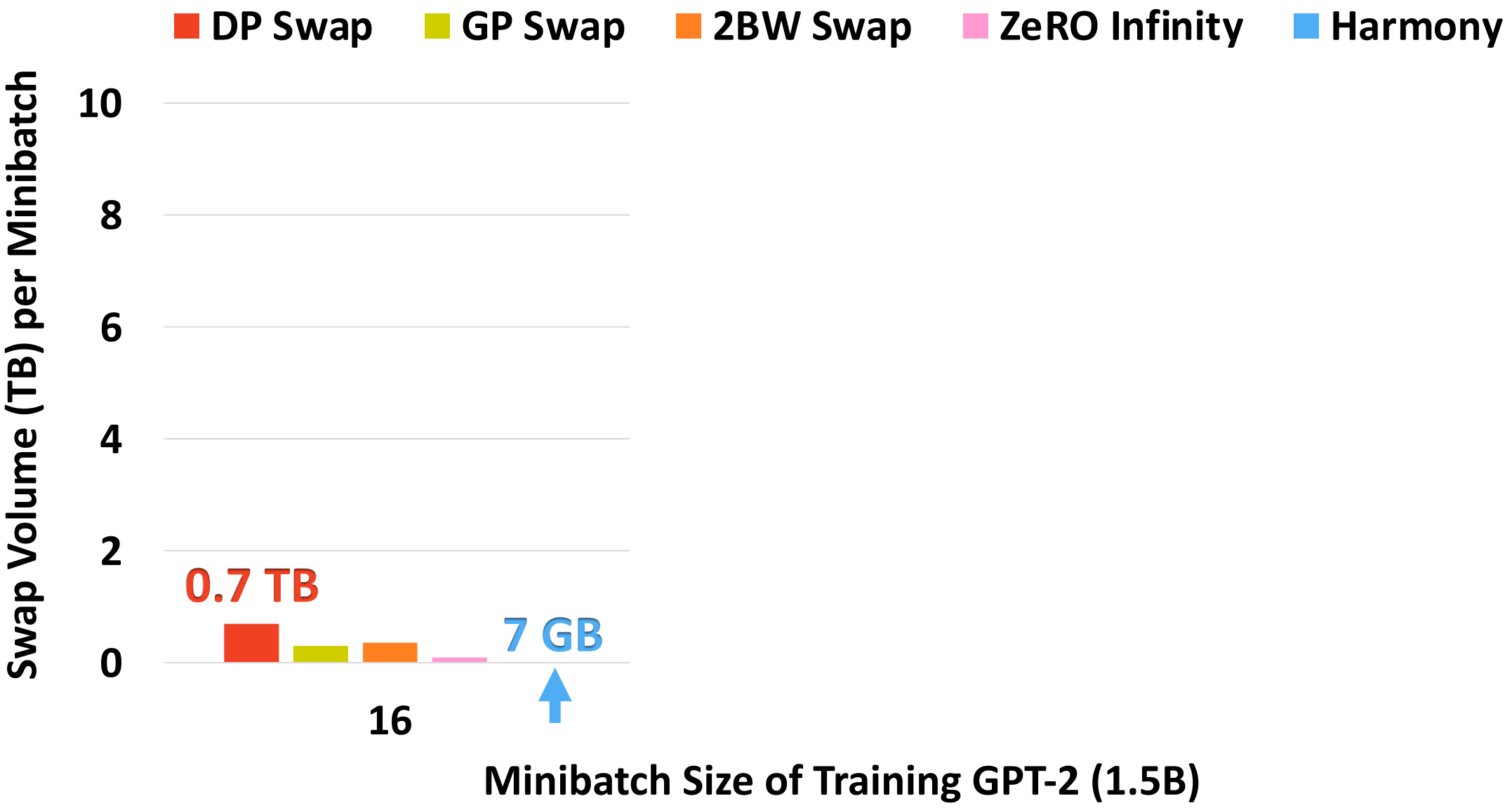


Boosting Training Throughput with Harmony

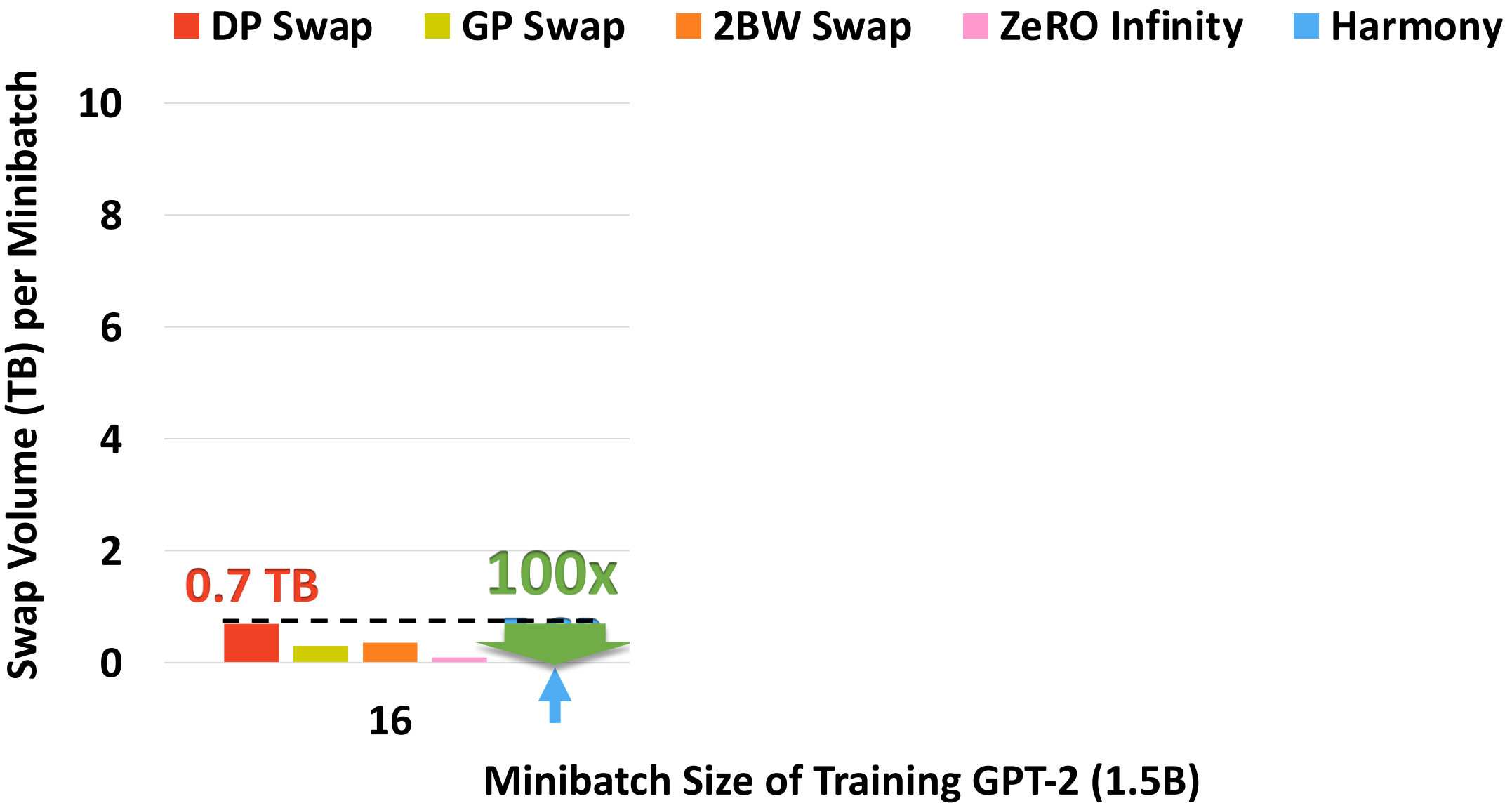


Reducing Swap Loads with Harmony

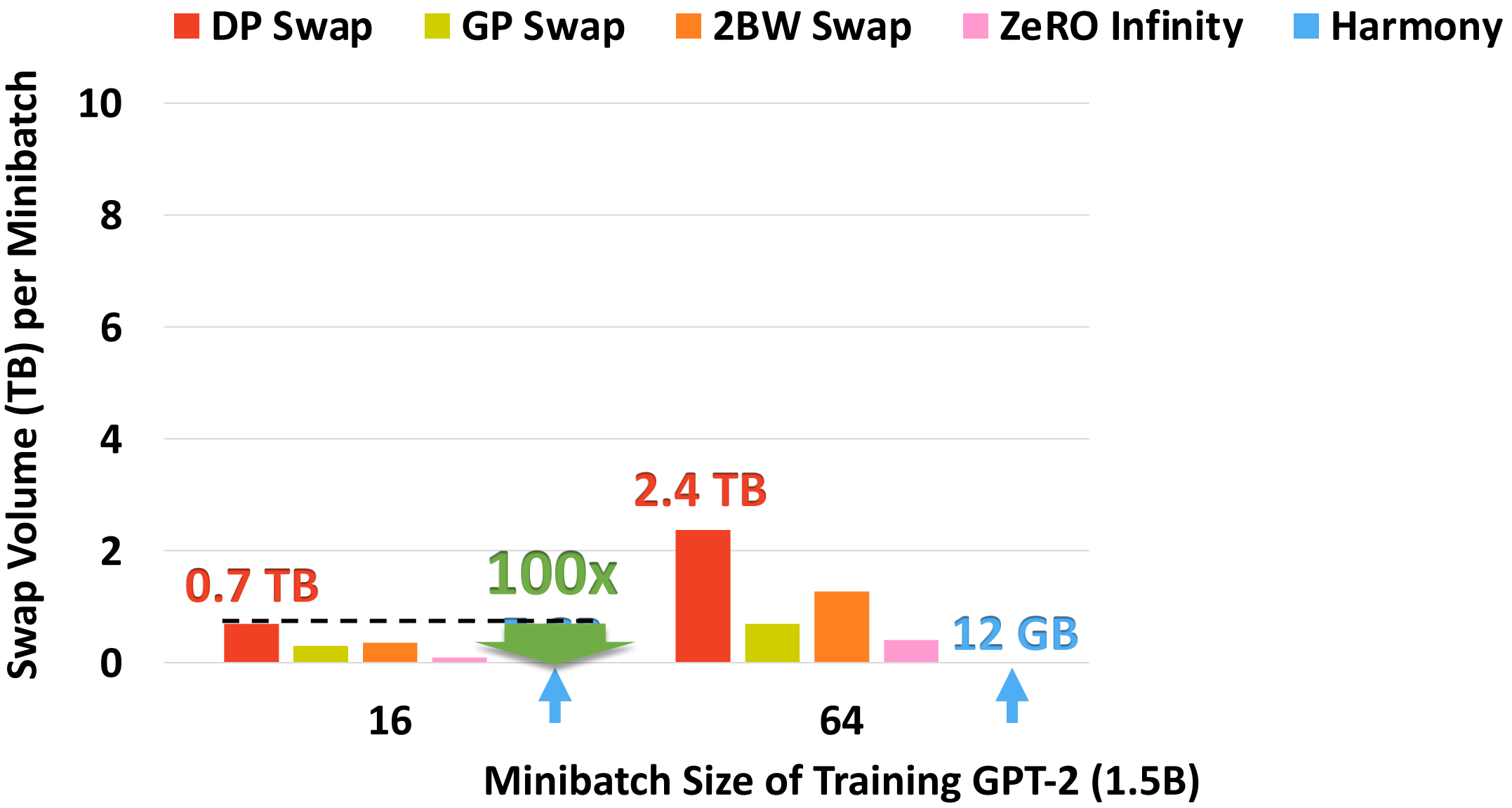
Reducing Swap Loads with Harmony



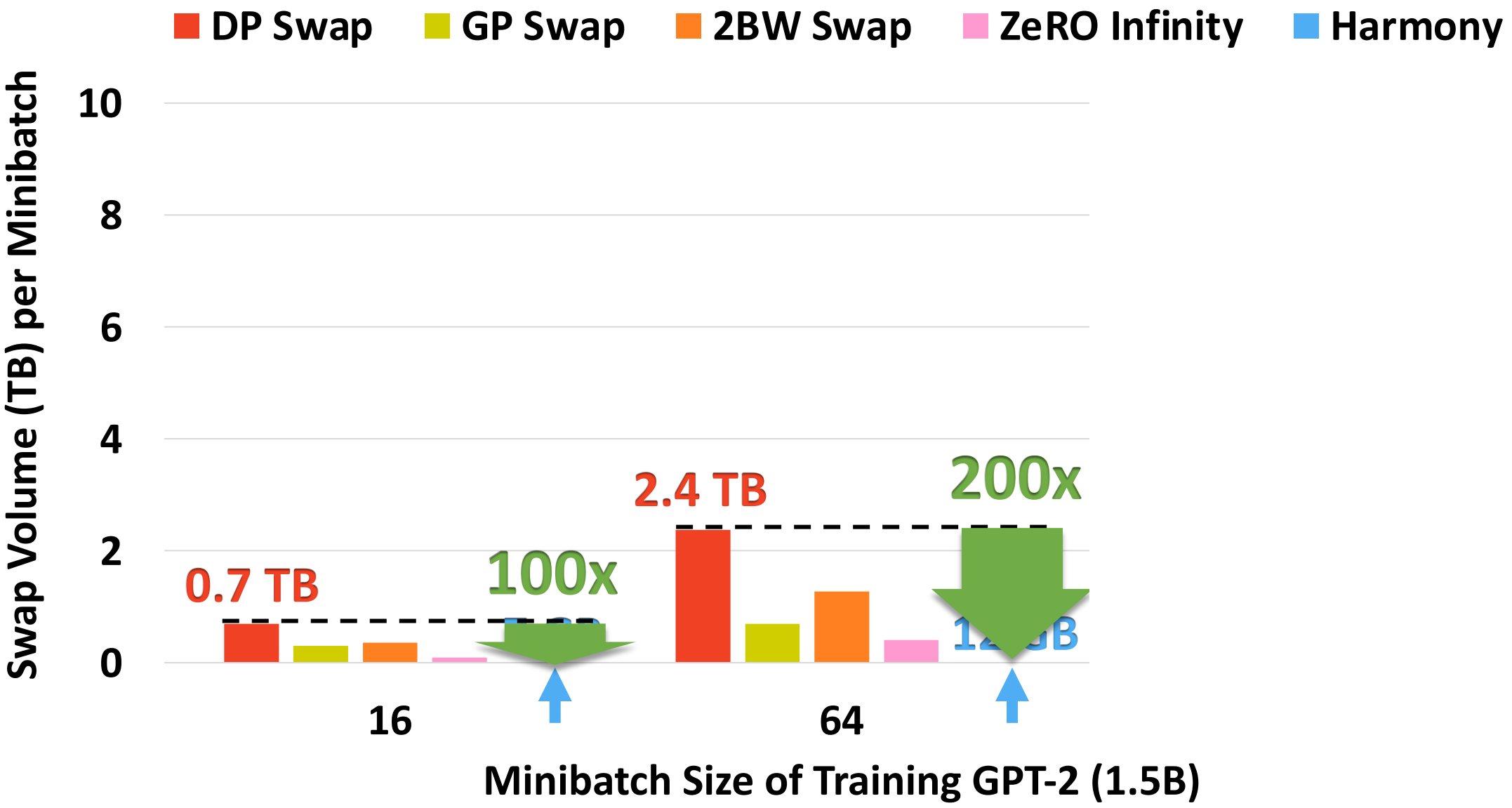
Reducing Swap Loads with Harmony



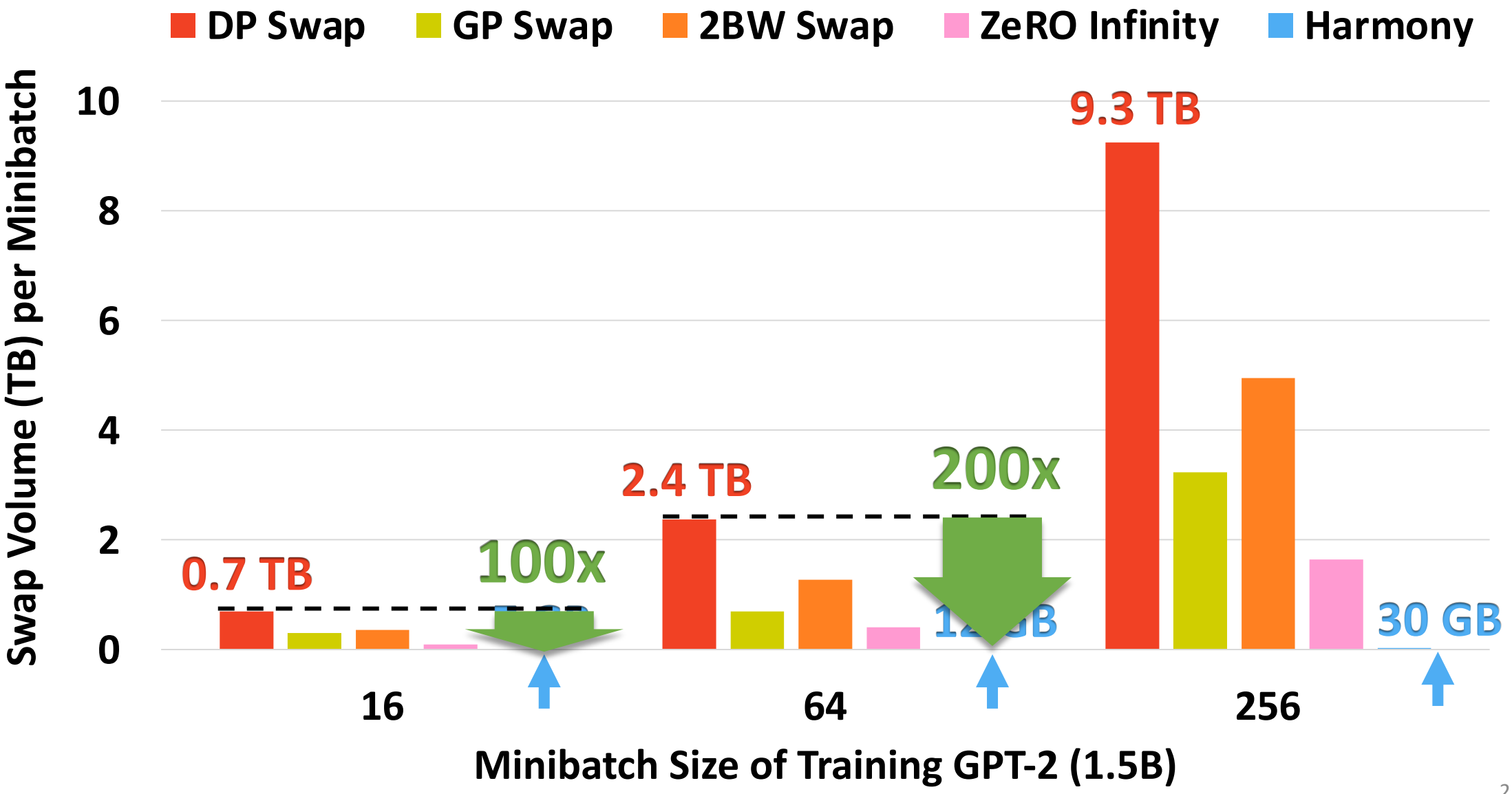
Reducing Swap Loads with Harmony



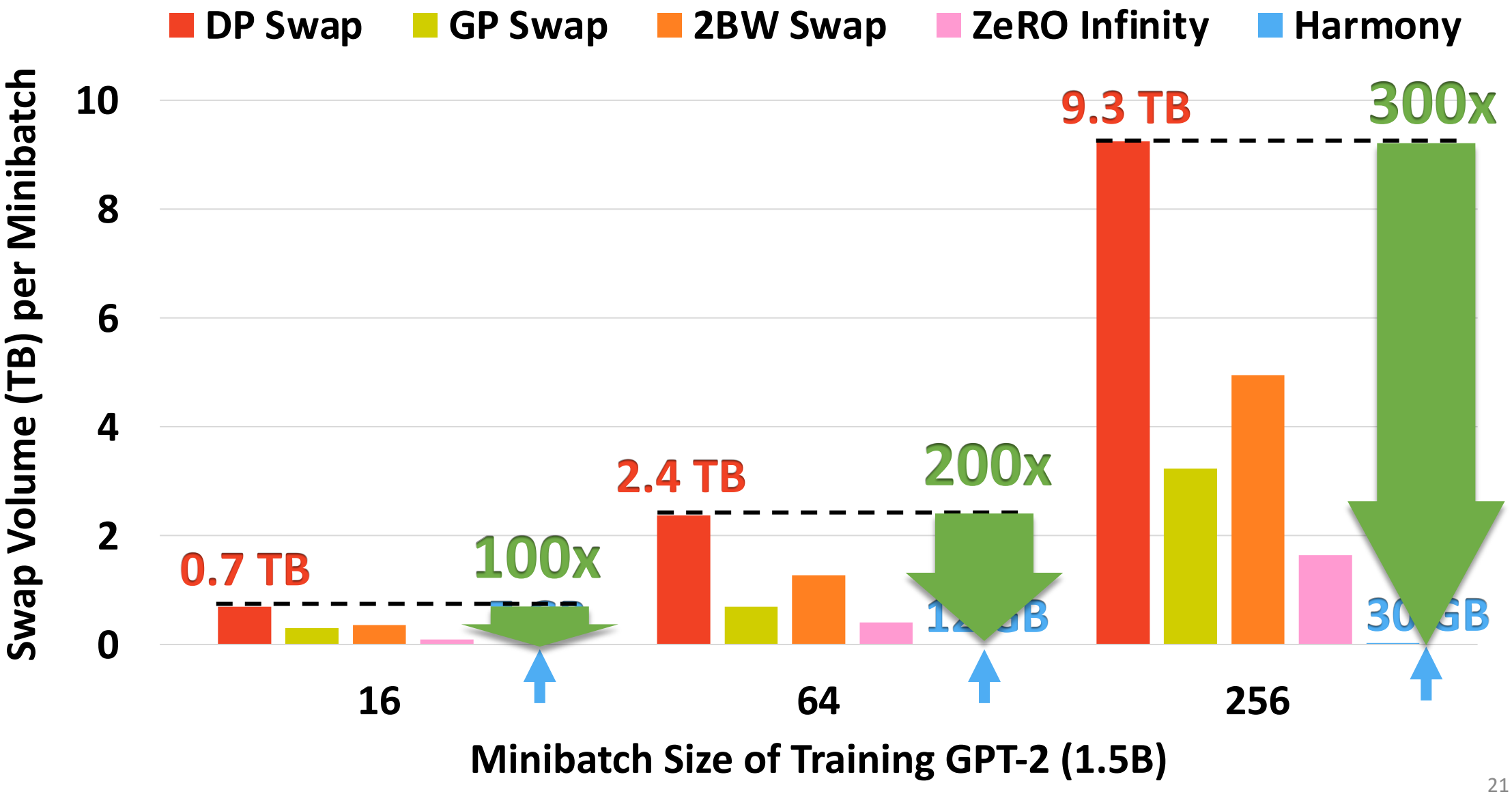
Reducing Swap Loads with Harmony



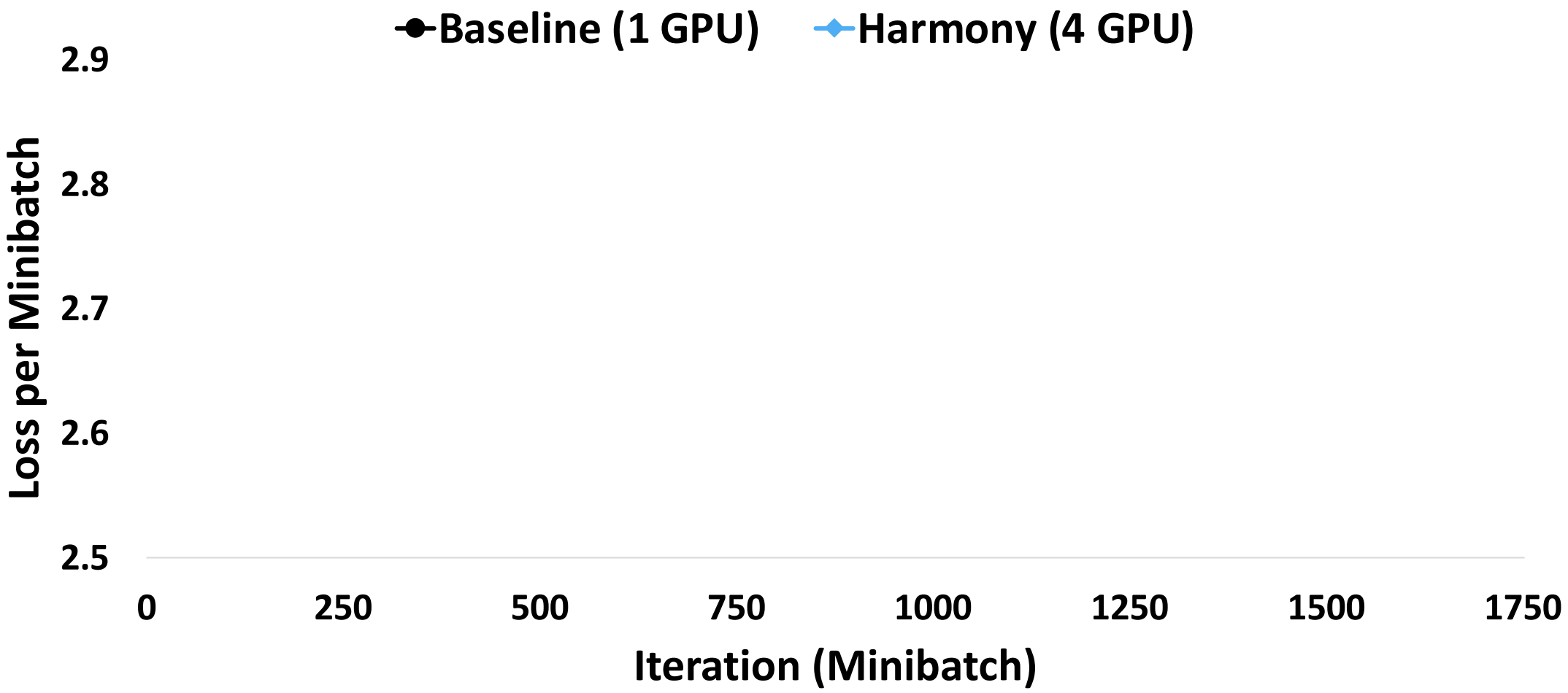
Reducing Swap Loads with Harmony



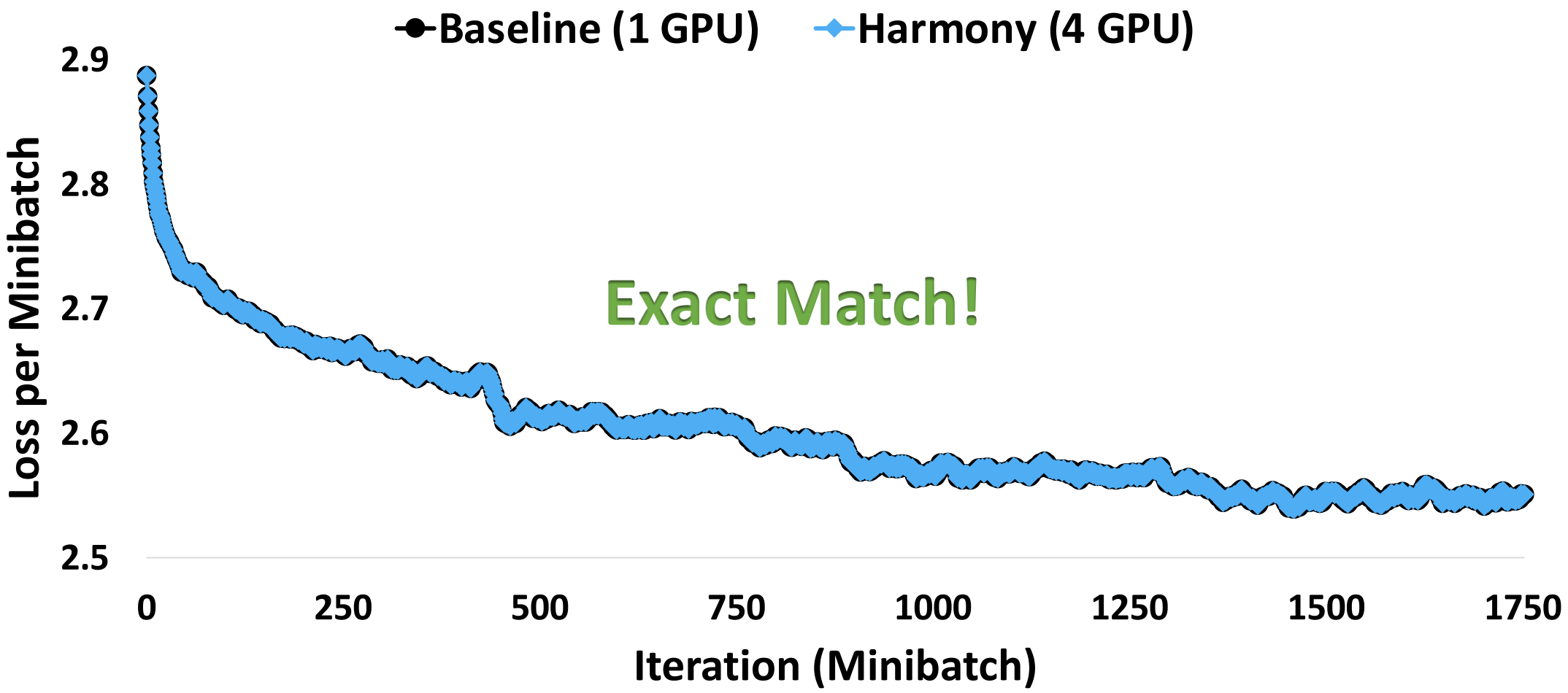
Reducing Swap Loads with Harmony



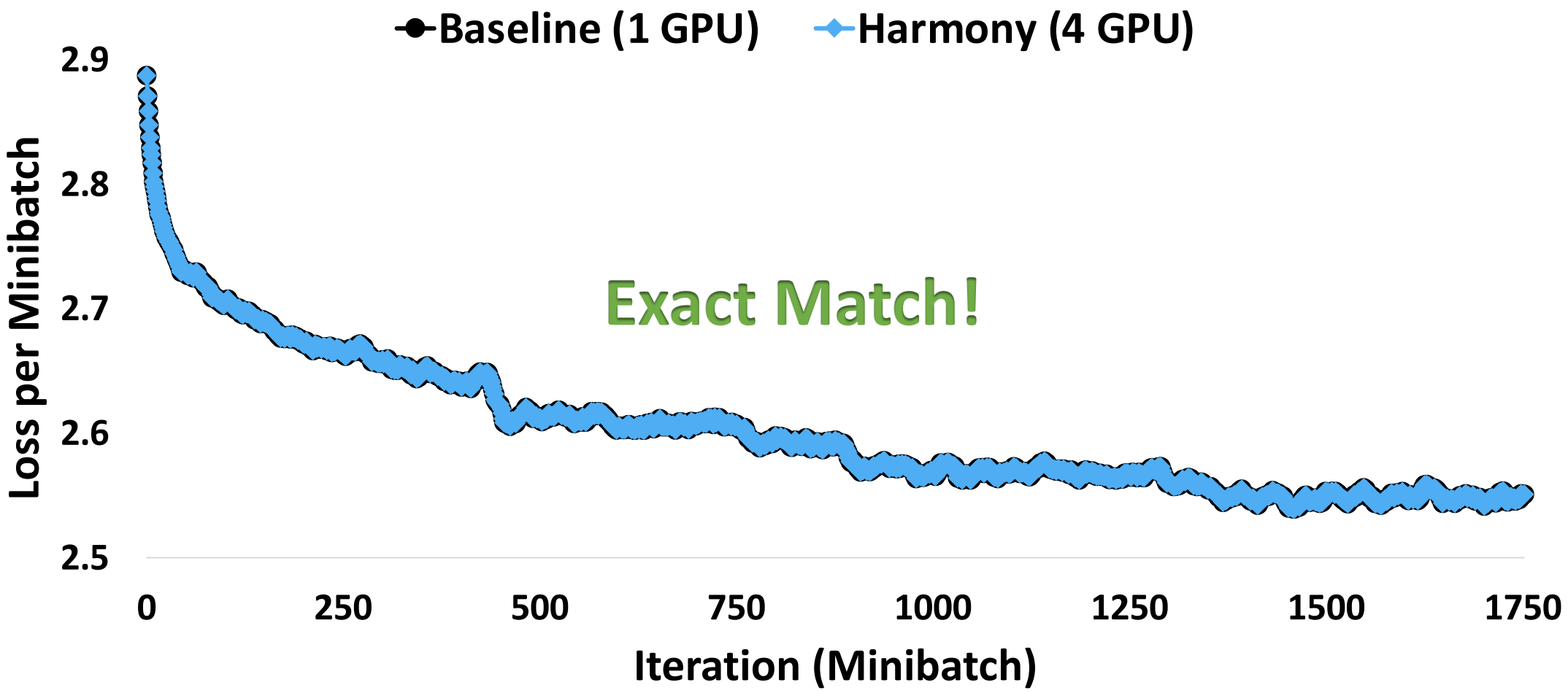
Correctness of Training in Harmony



Correctness of Training in Harmony



Correctness of Training in Harmony



Harmony provides both "*Single GPU Abstraction*" and "*Synchronous SGD*"

Summary

Summary

- **Training massive models requires huge amount of GPU memory**

Summary

- Training massive models requires huge amount of GPU memory
- Virtualizing GPU memory incurs excessive swap overhead

Summary

- Training massive models requires huge amount of GPU memory
- Virtualizing GPU memory incurs excessive swap overhead
- Harmony enables efficient training of massive models on commodity servers

Summary

- Training massive models requires huge amount of GPU memory
- Virtualizing GPU memory incurs excessive swap overhead
- Harmony enables efficient training of massive models on commodity servers

7.6x Training Speedup

Summary

- Training massive models requires huge amount of GPU memory
- Virtualizing GPU memory incurs excessive swap overhead
- Harmony enables efficient training of massive models on commodity servers

7.6x Training Speedup

300x Swap Load Reduction

Summary

- Training massive models requires huge amount of GPU memory
- Virtualizing GPU memory incurs excessive swap overhead
- Harmony enables efficient training of massive models on commodity servers

7.6x Training Speedup

300x Swap Load Reduction

Synchronous SGD

Summary

- Training massive models requires huge amount of GPU memory
- Virtualizing GPU memory incurs excessive swap overhead
- Harmony enables efficient training of massive models on commodity servers

7.6x Training Speedup

300x Swap Load Reduction

Synchronous SGD

Linear Scalability (see paper)

Summary

- Training massive models requires huge amount of GPU memory
- Virtualizing GPU memory incurs excessive swap overhead
- Harmony enables efficient training of massive models on commodity servers

7.6x Training Speedup

300x Swap Load Reduction

Synchronous SGD

Linear Scalability (see paper)

github.com/msr-fiddle/harmony

Training Massive Models in “Harmony”

Youjie Li
li238@illinois.edu

Amar Phanishayee Derek Murray Jakub Tarnawski Nam Sung Kim

VLDB 2022