ASPLOS 2023
Vancouver, Canada

Tsinghua University

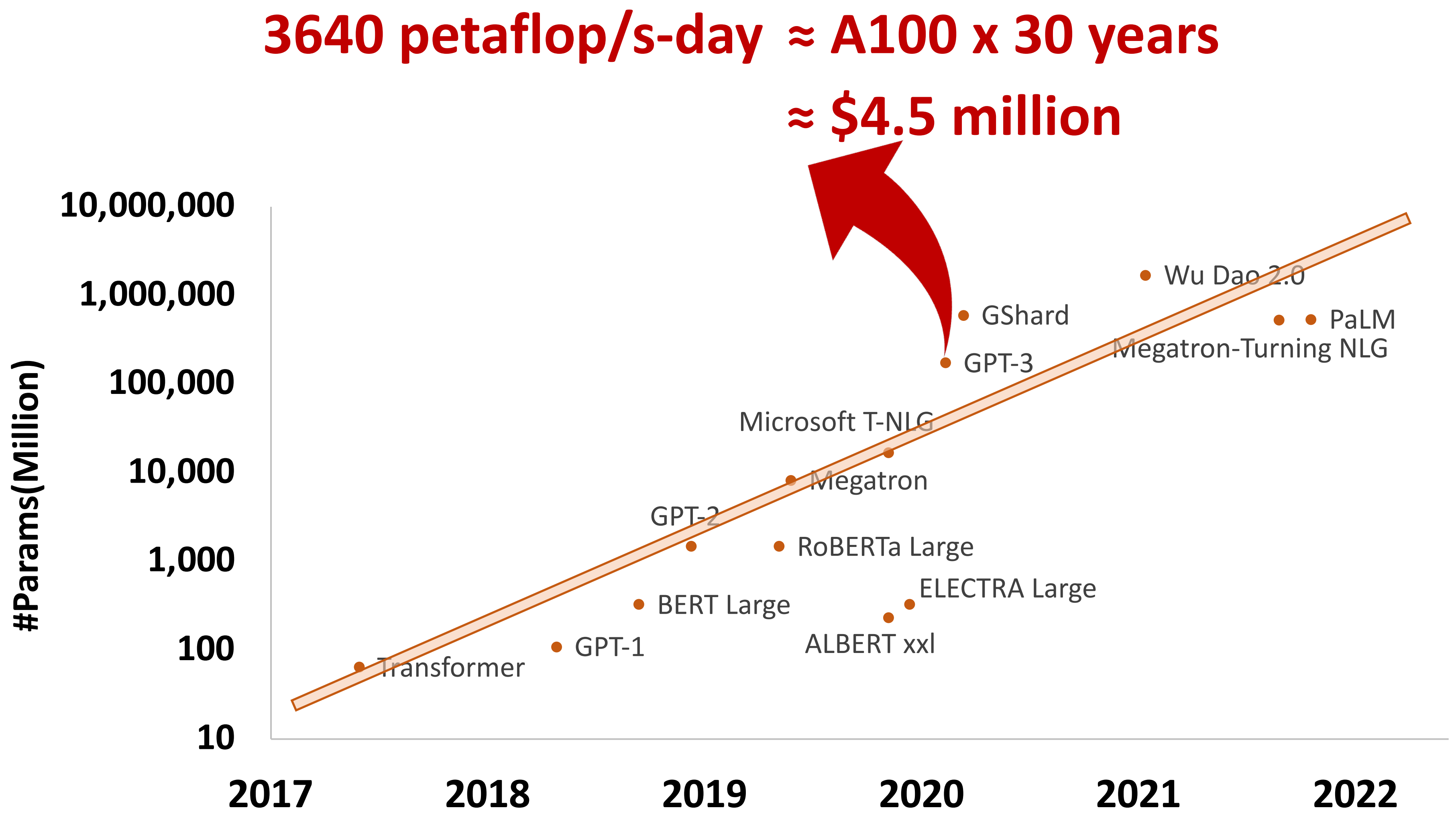
Mobius: Fine Tuning Large-Scale Models on Commodity GPU Servers

Yangyang Feng, Minhui Xie, Zijie Tian, Shuo Wang, Youyou Lu, and Jiwu Shu

Tsinghua University

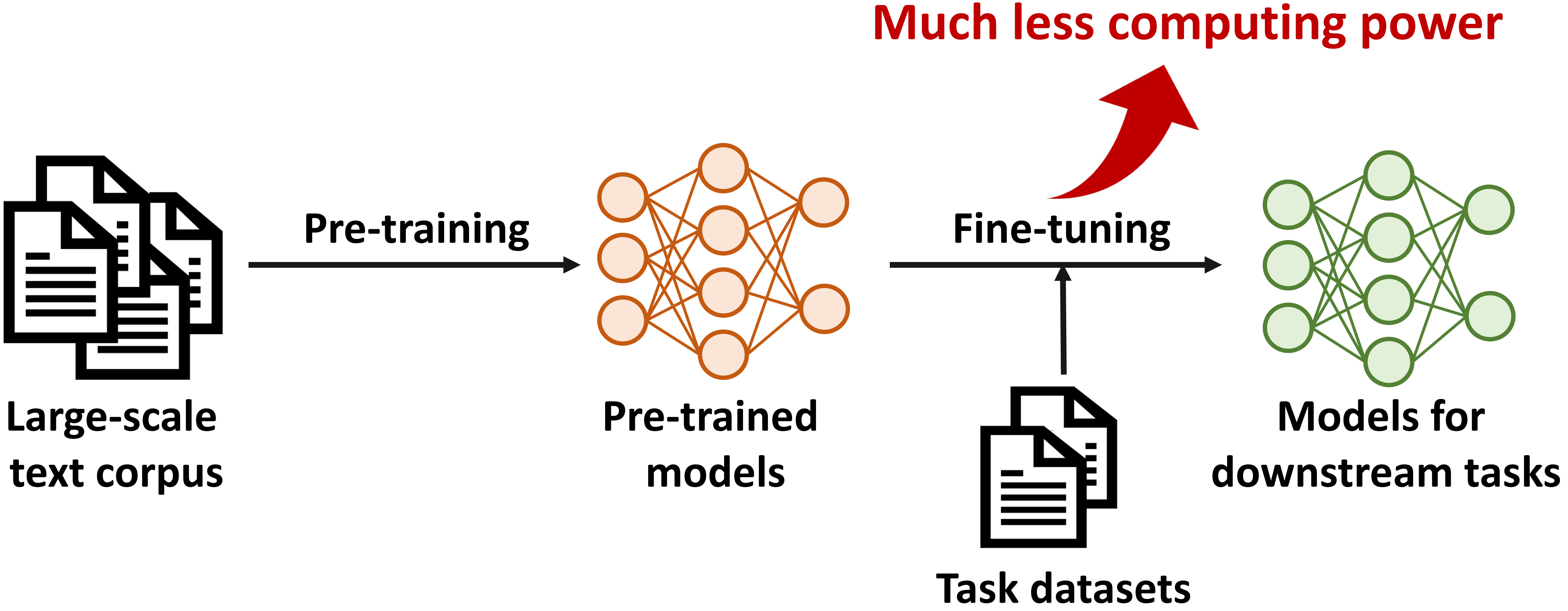
http://storage.cs.tsinghua.edu.cn

Explosive Growth of Model Size



[1] Brown, Tom, et al. "Language models are few-shot learners." Advances in neural information processing systems 33 (2020): 1877-1901.

Pre-training and Then Fine Tuning



Much less computing power

Pre-training

Fine-tuning

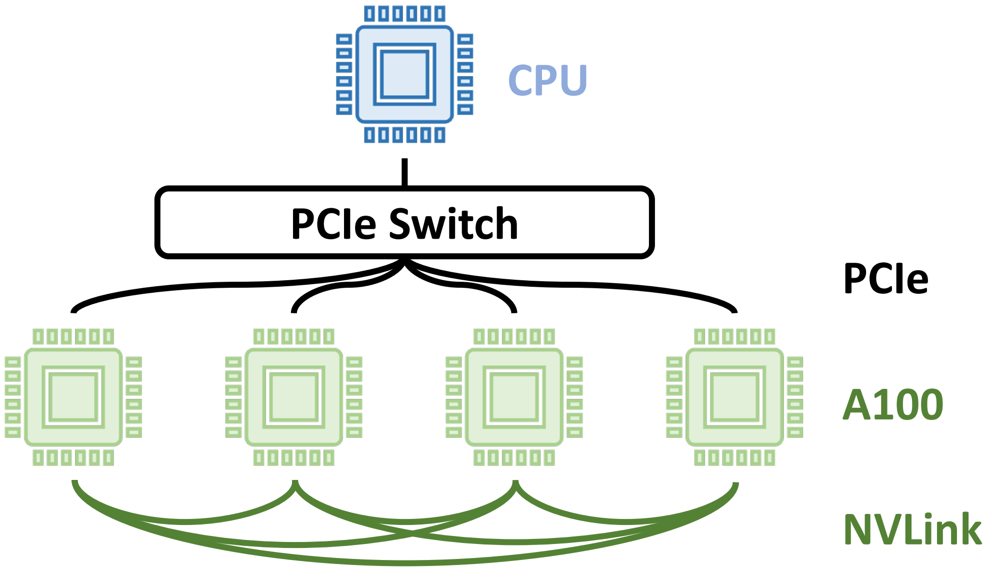
Large-scale text corpus

Pre-trained models

Task datasets

Models for downstream tasks

Commodity GPU Servers



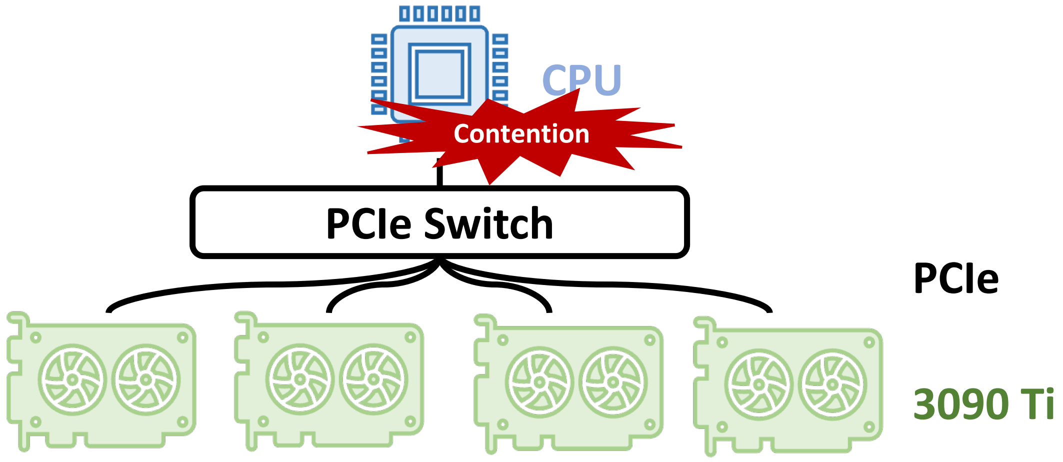
Data Center GPU Server

$100, 000

432 tensor cores

900 GB/s inter-GPUs comm.

GPU Direct P2P



Commodity GPU Server

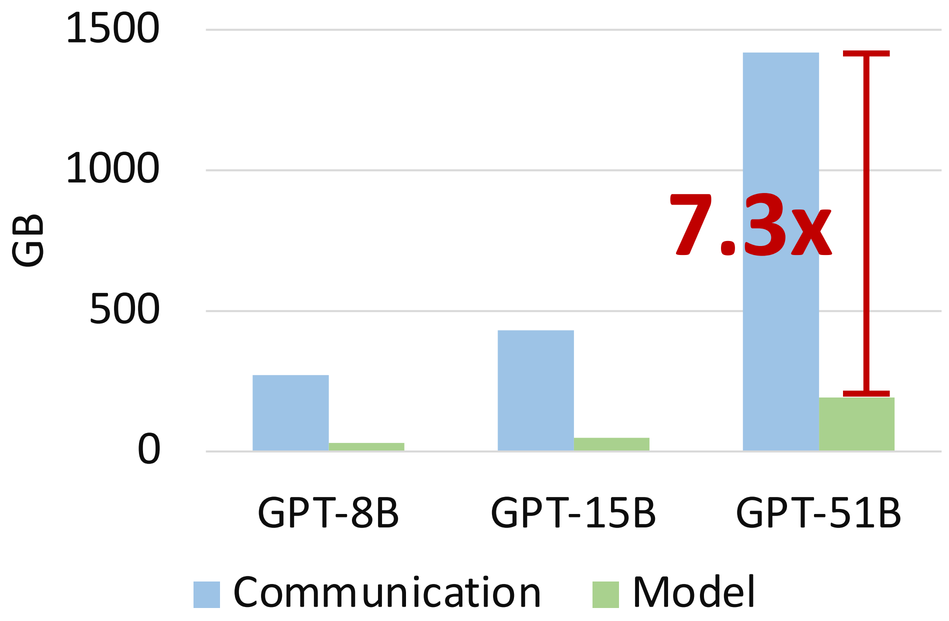
$10, 000

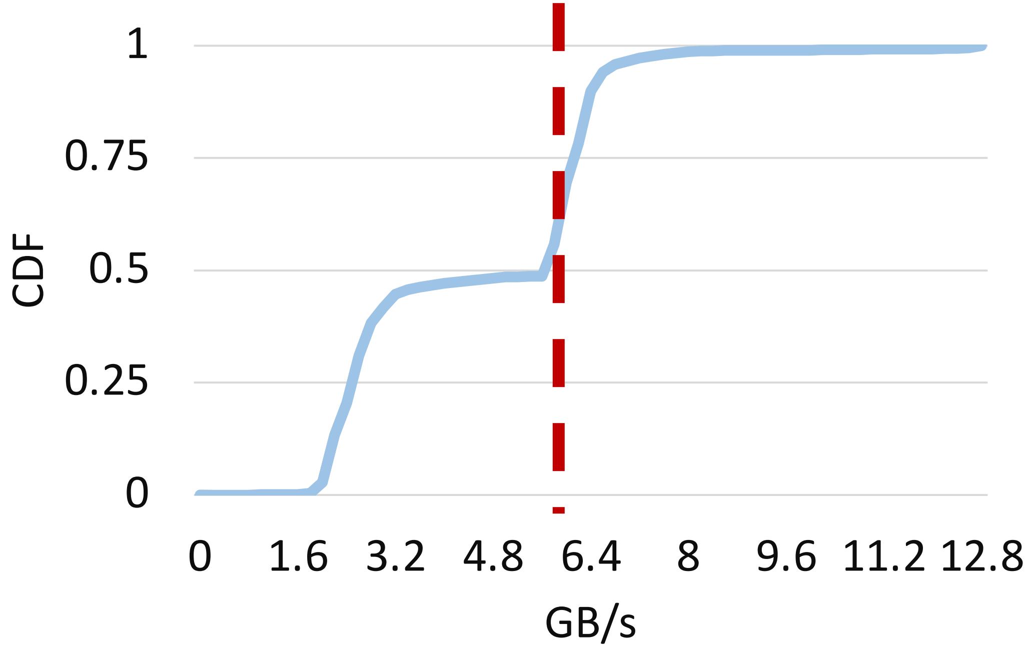
336 tensor cores

16 GB/s inter-GPUs comm.

NO GPU Direct P2P

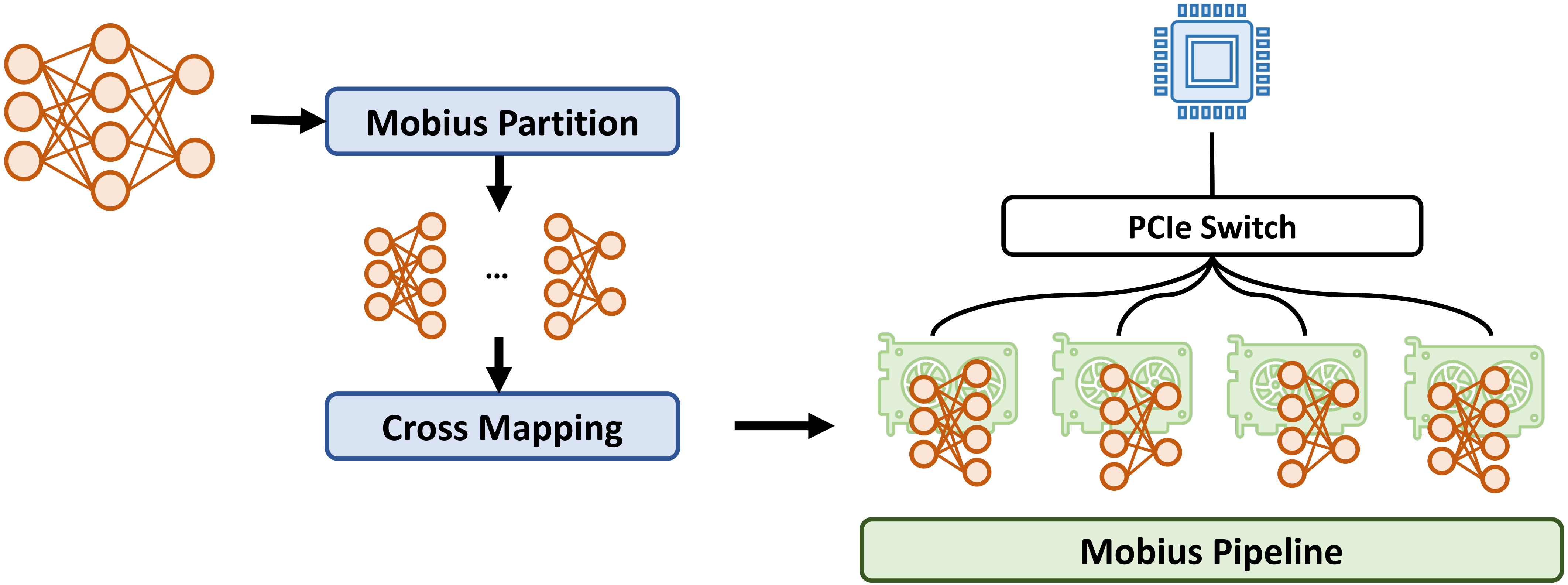
Training on Commodity GPU Servers



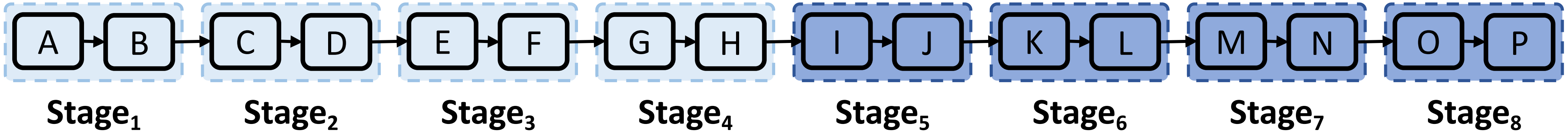


70% of training time is spent on communication

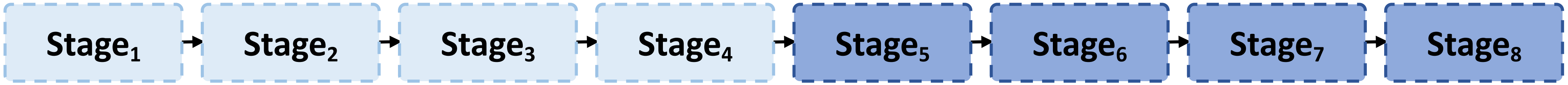
Mobius Overall



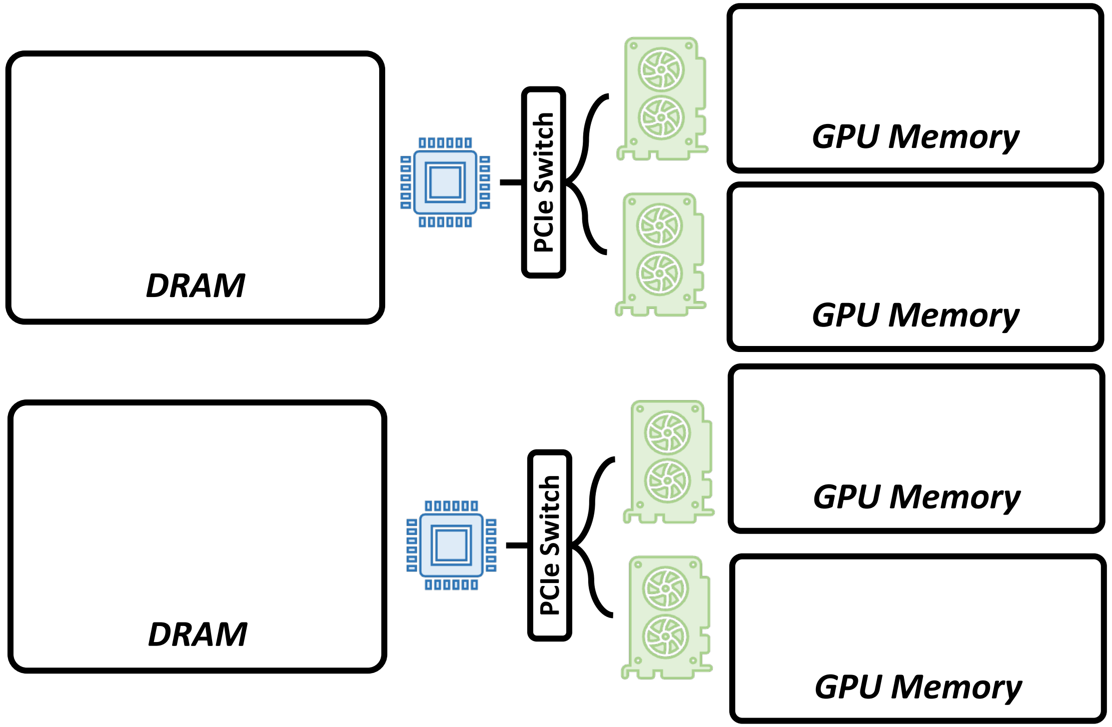
Mobius Pipeline



Mobius Pipeline



Stage$_1$ → Stage$_2$ → Stage$_3$ → Stage$_4$ → Stage$_5$ → Stage$_6$ → Stage$_7$ → Stage$_8$



DRAM

PCIe Switch

GPU Memory

GPU Memory

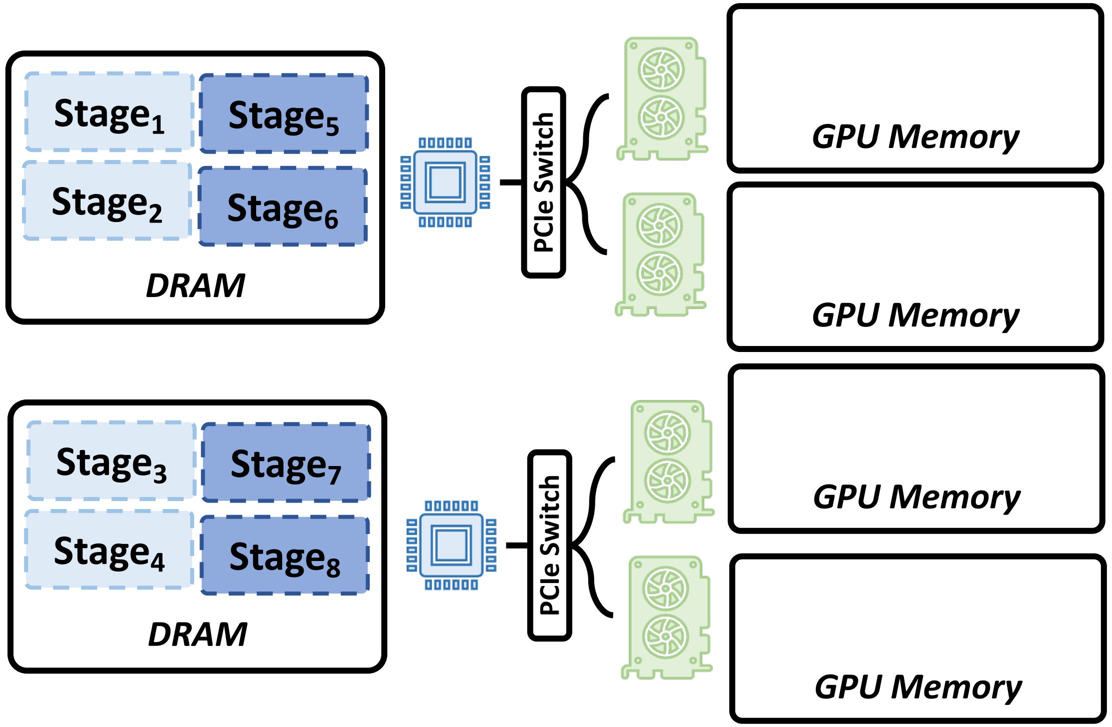
DRAM

PCIe Switch

GPU Memory

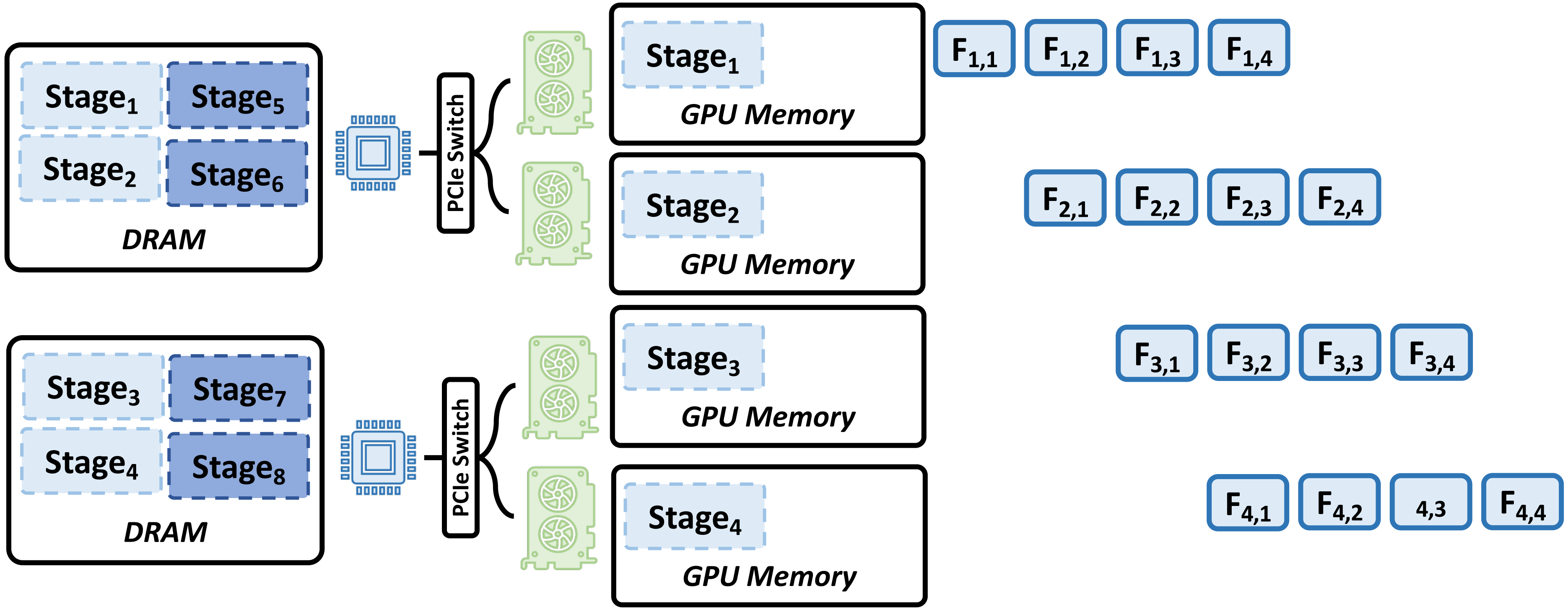
GPU Memory

Mobius Pipeline



Mobius Pipeline

$F_{i,\,j}$ Stage$_i$'s execution on jth microbatch

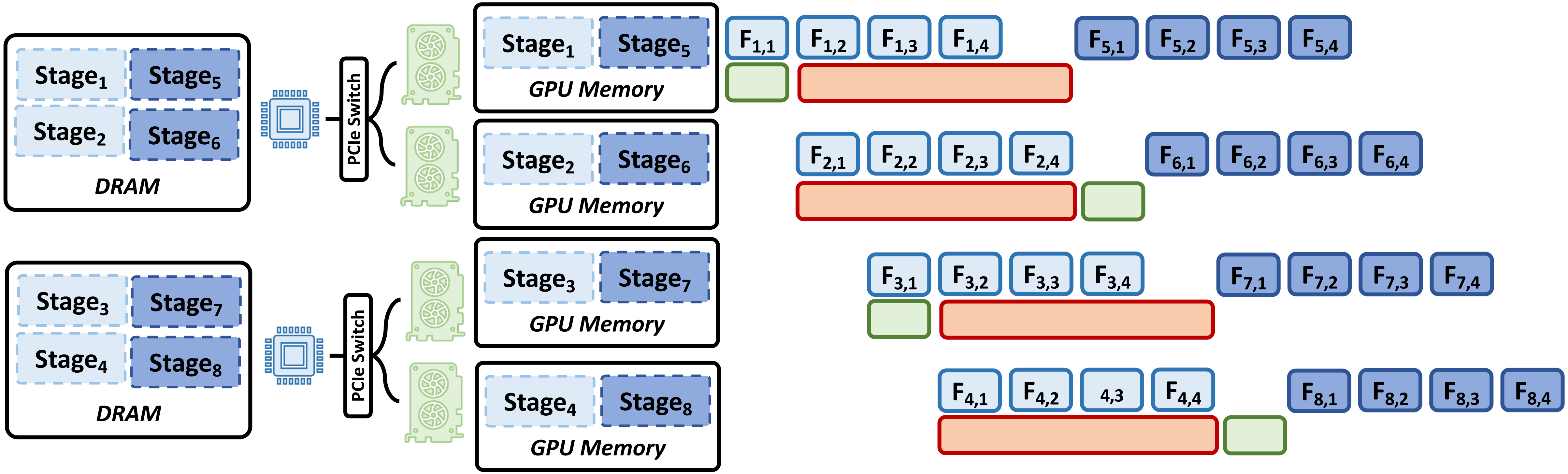


Mobius Pipeline

$F_{i,j}$ Stage$_i$'s execution on jth microbatch

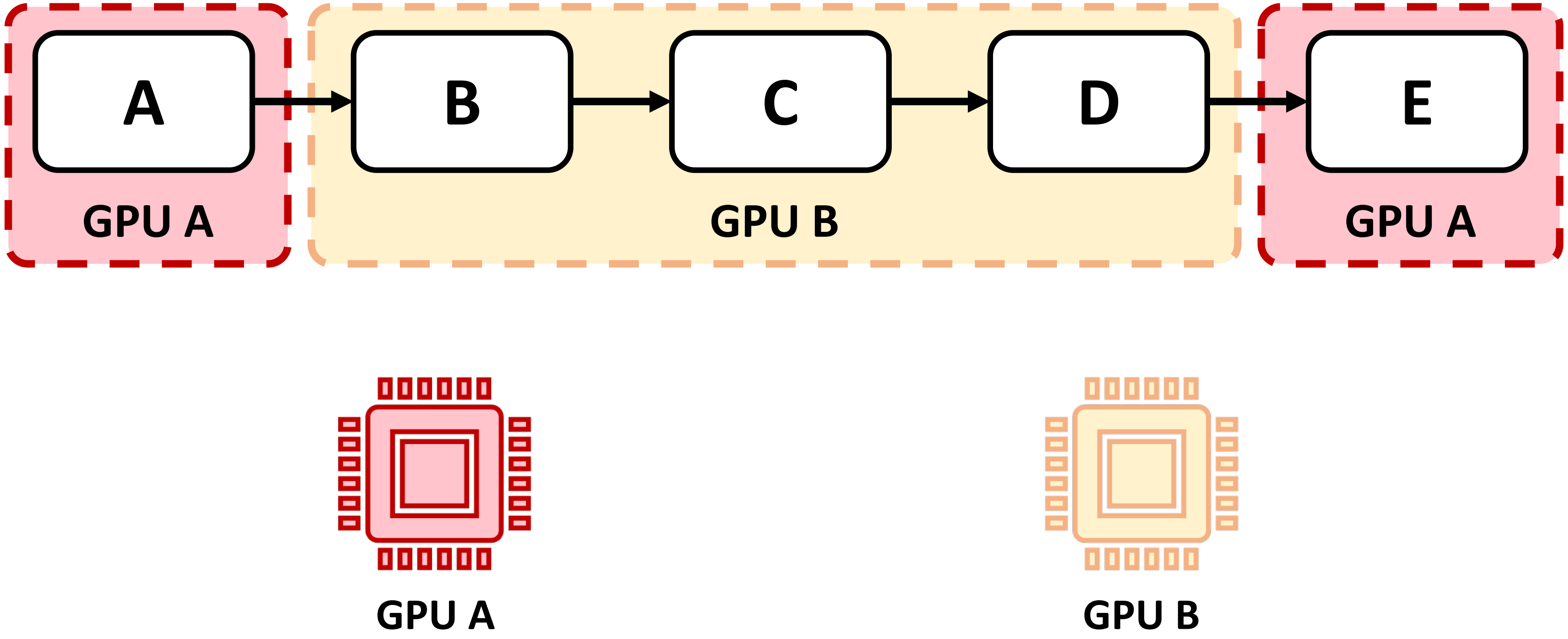
🟩 Communication without contention

🟥 Communication with contention


Stage$_1$
Stage$_5$
Stage$_2$
Stage$_6$
DRAM
Stage$_3$
Stage$_7$
Stage$_4$
Stage$_8$
DRAM
PCIe Switch
PCIe Switch
Stage$_1$
Stage$_5$
GPU Memory
Stage$_2$
Stage$_6$
GPU Memory
Stage$_3$
Stage$_7$
GPU Memory
Stage$_4$
Stage$_8$
GPU Memory
$F_{1,1}$ $F_{1,2}$ $F_{1,3}$ $F_{1,4}$ $F_{5,1}$ $F_{5,2}$ $F_{5,3}$ $F_{5,4}$
$F_{2,1}$ $F_{2,2}$ $F_{2,3}$ $F_{2,4}$ $F_{6,1}$ $F_{6,2}$ $F_{6,3}$ $F_{6,4}$
$F_{3,1}$ $F_{3,2}$ $F_{3,3}$ $F_{3,4}$ $F_{7,1}$ $F_{7,2}$ $F_{7,3}$ $F_{7,4}$
$F_{4,1}$ $F_{4,2}$ 4,3 $F_{4,4}$ $F_{8,1}$ $F_{8,2}$ $F_{8,3}$ $F_{8,4}$

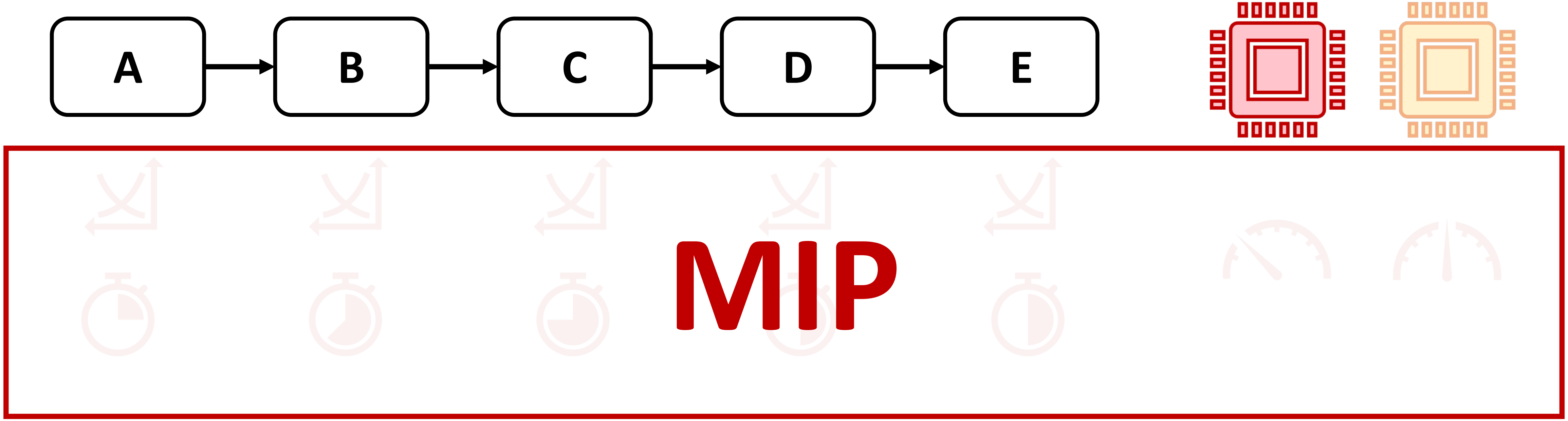
Two Partition Questions

- How many stages are in each GPU?
- How many layers are in each stage?



Mobius Partition

- Profile each layer’s **memory footprint and computation overhead**
- Profile **hardware performance**, i.e. bandwidth
- Use mixed integer program (MIP) to fine the **optimal partition scheme**



Mobius Partition

minimize *Training time of one step*

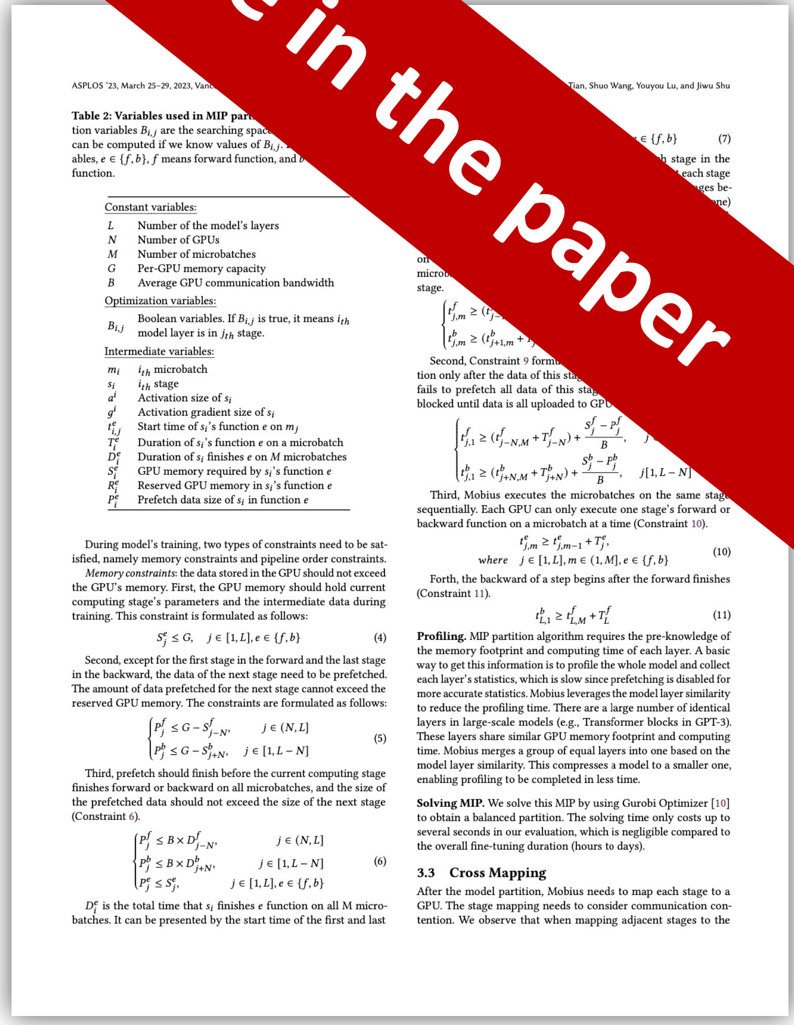
subject to *Memory constraints*

- *Memory required by computation*
- *Memory required by prefetching*

Pipeline order constraints

- *Stage execution order*
- *Microbatch execution order*

More in the paper



ASPLOS '23, March 25–29, 2023, Vanc

Tian, Shuo Wang, Youyou Lu, and Jiwu Shu

Table 2: Variables used in MIP part
tion variables $B_{i,j}$ are the searching spac
can be computed if we know values of $B_{i,j}$.
ables, $e \in \{f, b\}$, f means forward function, and
function.

Constant variables:	
L	Number of the model's layers
N	Number of GPUs
M	Number of microbatches
G	Per-GPU memory capacity
B	Average GPU communication bandwidth
Optimization variables:	
$B_{i,j}$	Boolean variables. If $B_{i,j}$ is true, it means i_{th} model layer is in j_{th} stage.
Intermediate variables:	
m_i	i_{th} microbatch
s_i	i_{th} stage
a^i	Activation size of s_i
g^i	Activation gradient size of s_i
$t^e_{i,j}$	Start time of s_i's function e on m_j
T^e_i	Duration of s_i's function e on a microbatch
D^e_i	Duration of s_i finishes e on M microbatches
S^e_i	GPU memory required by s_i's function e
R^e_i	Reserved GPU memory in s_i's function e
P^e_i	Prefetch data size of s_i in function e

During model's training, two types of constraints need to be satisfied, namely memory constraints and pipeline order constraints.

Memory constraints: the data stored in the GPU should not exceed the GPU's memory. First, the GPU memory should hold current computing stage's parameters and the intermediate data during training. This constraint is formulated as follows:

$$S^e_j \le G, \quad j \in [1, L], e \in \{f, b\} \tag{4}$$

Second, except for the first stage in the forward and the last stage in the backward, the data of the next stage need to be prefetched. The amount of data prefetched for the next stage cannot exceed the reserved GPU memory. The constraints are formulated as follows:

$$\begin{cases} P^f_j \le G - S^f_{j-N}, & j \in (N, L] \\ P^b_j \le G - S^b_{j+N}, & j \in [1, L-N] \end{cases} \tag{5}$$

Third, prefetch should finish before the current computing stage finishes forward or backward on all microbatches, and the size of the prefetched data should not exceed the size of the next stage (Constraint 6).

$$\begin{cases} P^f_j \le B \times D^f_{j-N}, & j \in (N, L] \\ P^b_j \le B \times D^b_{j+N}, & j \in [1, L-N] \\ P^e_j \le S^e_j, & j \in [1, L], e \in \{f, b\} \end{cases} \tag{6}$$

D^e_i is the total time that s_i finishes e function on all M microbatches. It can be presented by the start time of the first and last

$\in \{f, b\}$ (7)

h stage in the
each stage
ges be-
ne)

on
micro
stage.

$$\begin{cases} t^f_{j,m} \ge (t^f_{j- \\ t^b_{j,m} \ge (t^b_{j+1,m} + \end{cases}$$

Second, Constraint 9 form
tion only after the data of this sta
fails to prefetch all data of this sta
blocked until data is all uploaded to GPU

$$\begin{cases} t^f_{j,1} \ge (t^f_{j-N,M} + T^f_{j-N}) + \frac{S^f_j - P^f_j}{B}, & j \\ t^b_{j,1} \ge (t^b_{j+N,M} + T^b_{j+N}) + \frac{S^b_j - P^b_j}{B}, & j[1, L-N] \end{cases}$$

Third, Mobius executes the microbatches on the same stage sequentially. Each GPU can only execute one stage's forward or backward function on a microbatch at a time (Constraint 10).

$$\begin{aligned} & t^e_{j,m} \ge t^e_{j,m-1} + T^e_j, \\ where \quad & j \in [1, L], m \in (1, M], e \in \{f, b\} \end{aligned} \tag{10}$$

Forth, the backward of a step begins after the forward finishes (Constraint 11).

$$t^b_{L,1} \ge t^f_{L,M} + T^f_L \tag{11}$$

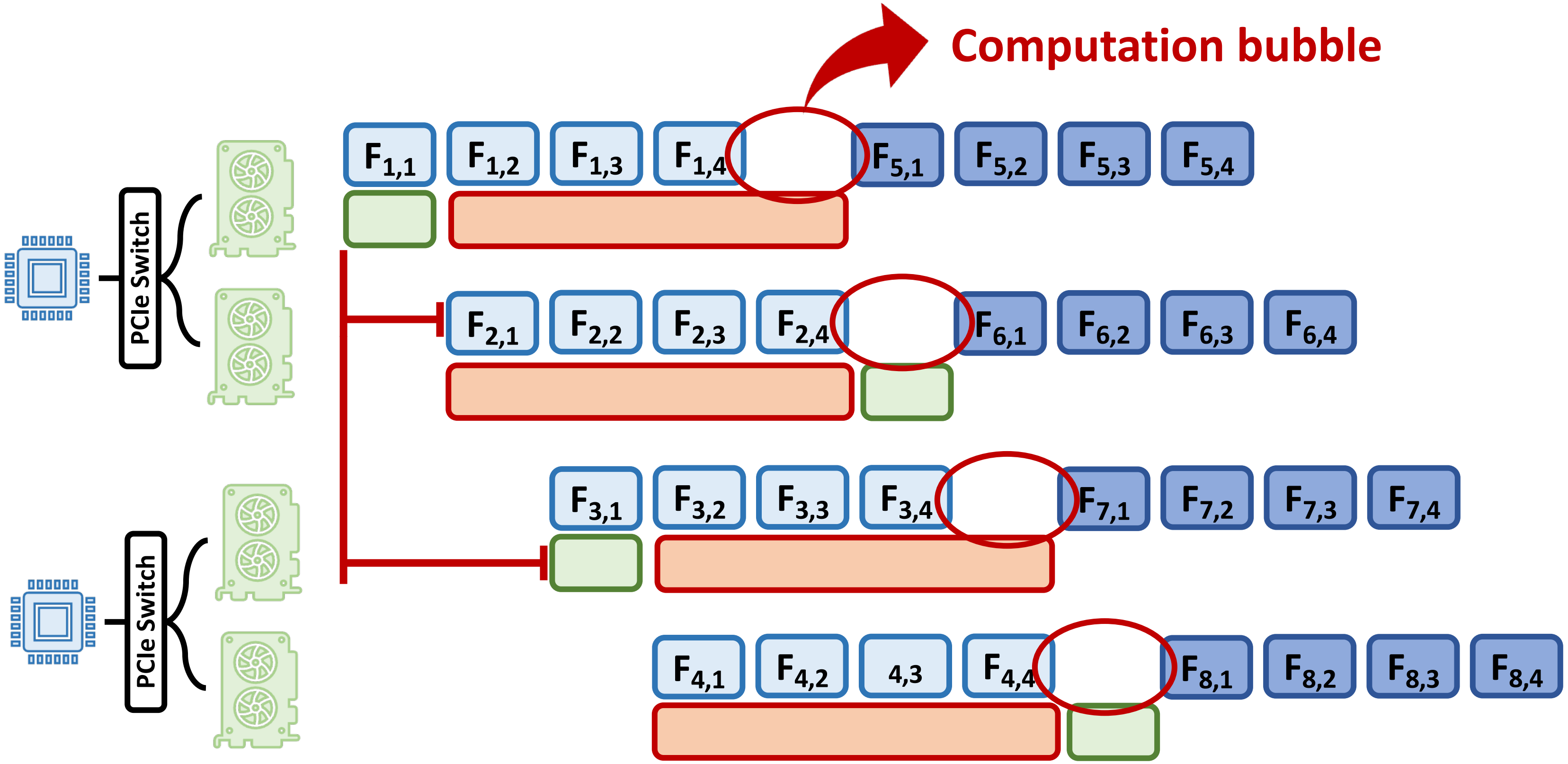
Profiling. MIP partition algorithm requires the pre-knowledge of the memory footprint and computing time of each layer. A basic way to get this information is to profile the whole model and collect each layer's statistics, which is slow since prefetching is disabled for more accurate statistics. Mobius leverages the model layer similarity to reduce the profiling time. There are a large number of identical layers in large-scale models (e.g., Transformer blocks in GPT-3). These layers share similar GPU memory footprint and computing time. Mobius merges a group of equal layers into one based on the model layer similarity. This compresses a model to a smaller one, enabling profiling to be completed in less time.

Solving MIP. We solve this MIP by using Gurobi Optimizer [10] to obtain a balanced partition. The solving time only costs up to several seconds in our evaluation, which is negligible compared to the overall fine-tuning duration (hours to days).

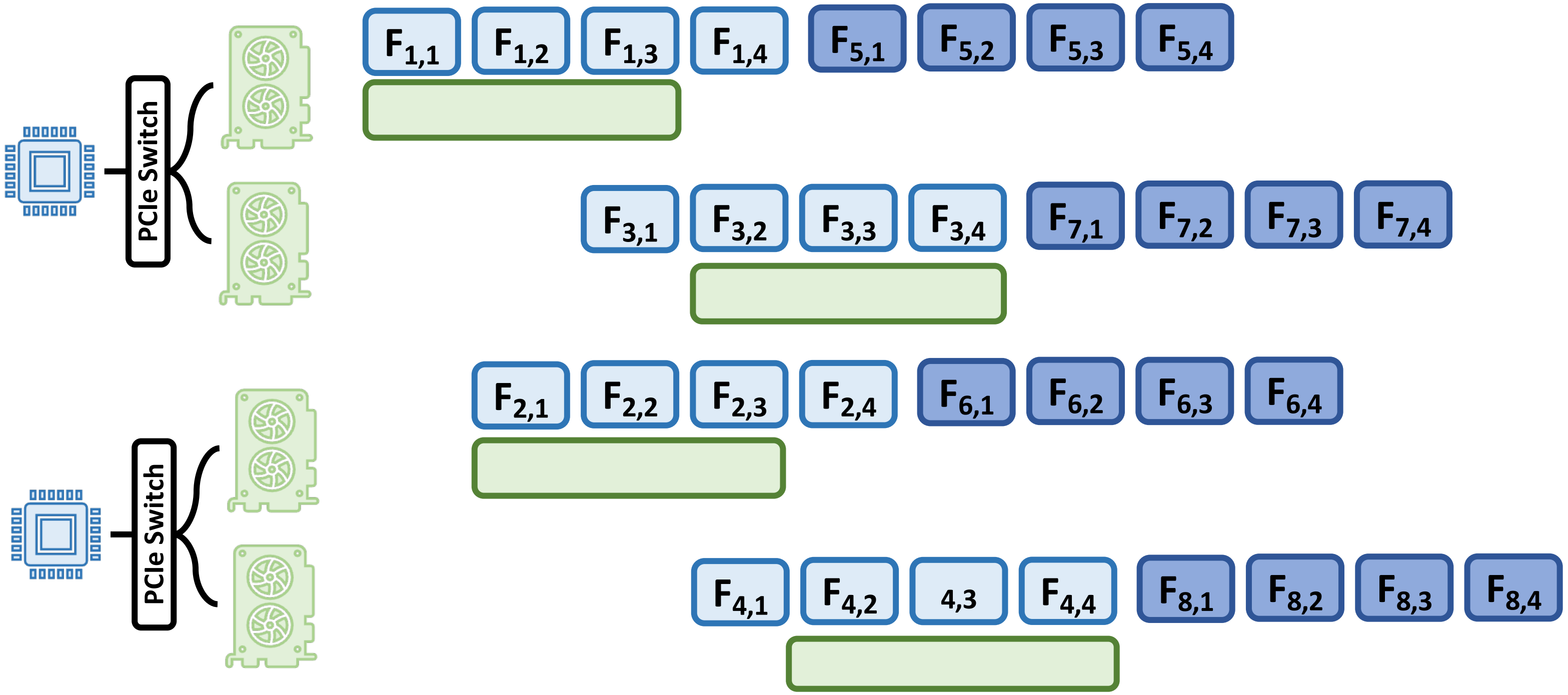
3.3 Cross Mapping

After the model partition, Mobius needs to map each stage to a GPU. The stage mapping needs to consider communication contention. We observe that when mapping adjacent stages to the

Communication Contention

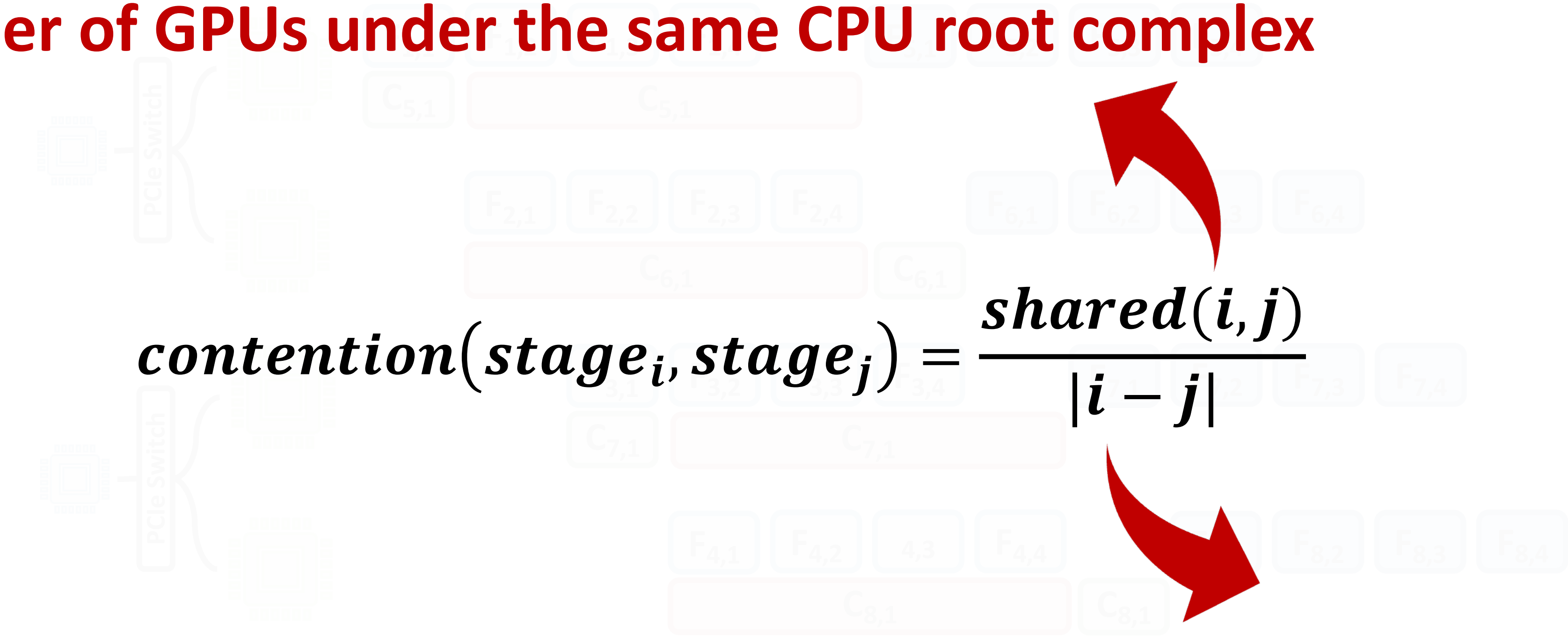


Cross Mapping


PCIe Switch
PCIe Switch
$F_{1,1}$ $F_{1,2}$ $F_{1,3}$ $F_{1,4}$ $F_{5,1}$ $F_{5,2}$ $F_{5,3}$ $F_{5,4}$
$F_{3,1}$ $F_{3,2}$ $F_{3,3}$ $F_{3,4}$ $F_{7,1}$ $F_{7,2}$ $F_{7,3}$ $F_{7,4}$
$F_{2,1}$ $F_{2,2}$ $F_{2,3}$ $F_{2,4}$ $F_{6,1}$ $F_{6,2}$ $F_{6,3}$ $F_{6,4}$
$F_{4,1}$ $F_{4,2}$ 4,3 $F_{4,4}$ $F_{8,1}$ $F_{8,2}$ $F_{8,3}$ $F_{8,4}$

Map adjacent stages to GPUs not under the same CPU root complex

Cross Mapping

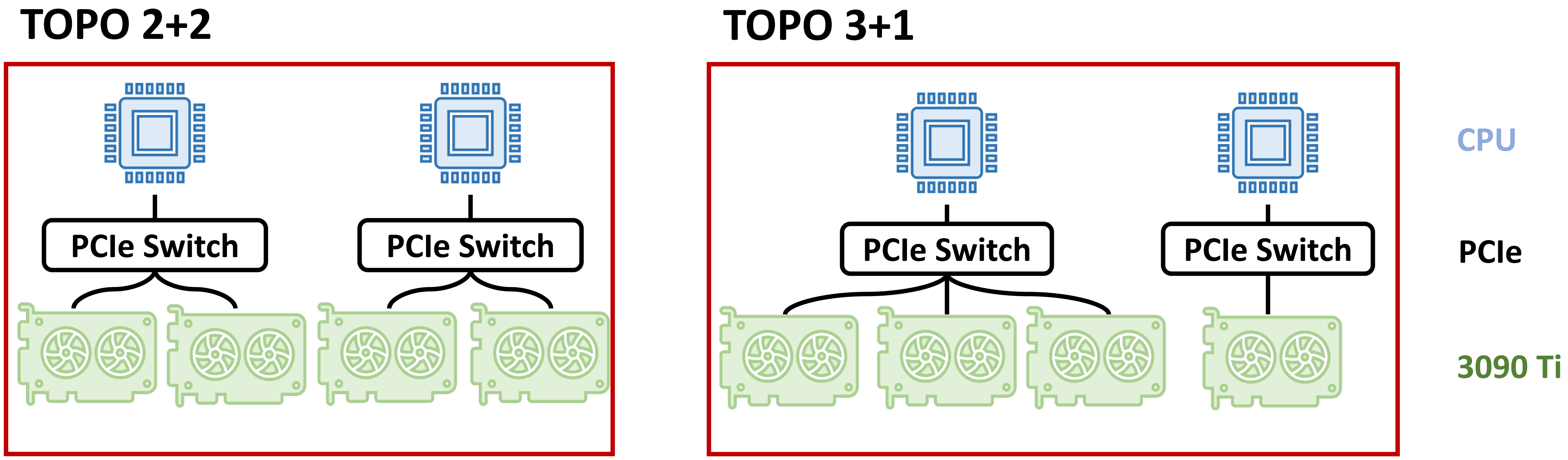


Number of GPUs under the same CPU root complex

$$\boldsymbol{contention(stage_i, stage_j) = \frac{shared(i, j)}{|i - j|}}$$

Time difference to upload the two stages' data

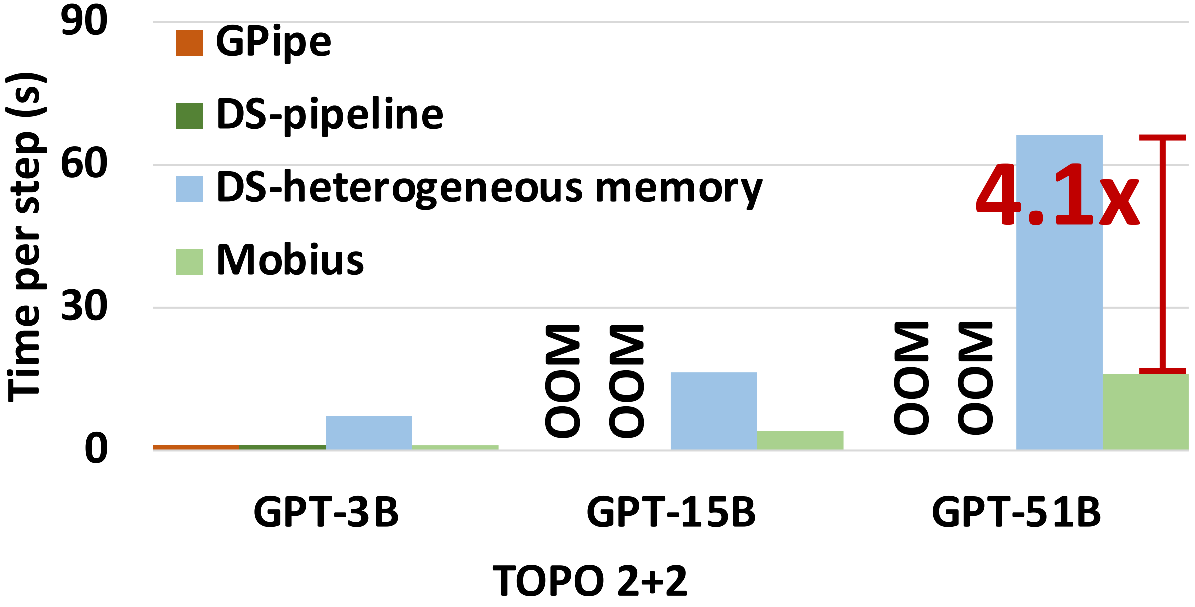
Experimental Setup

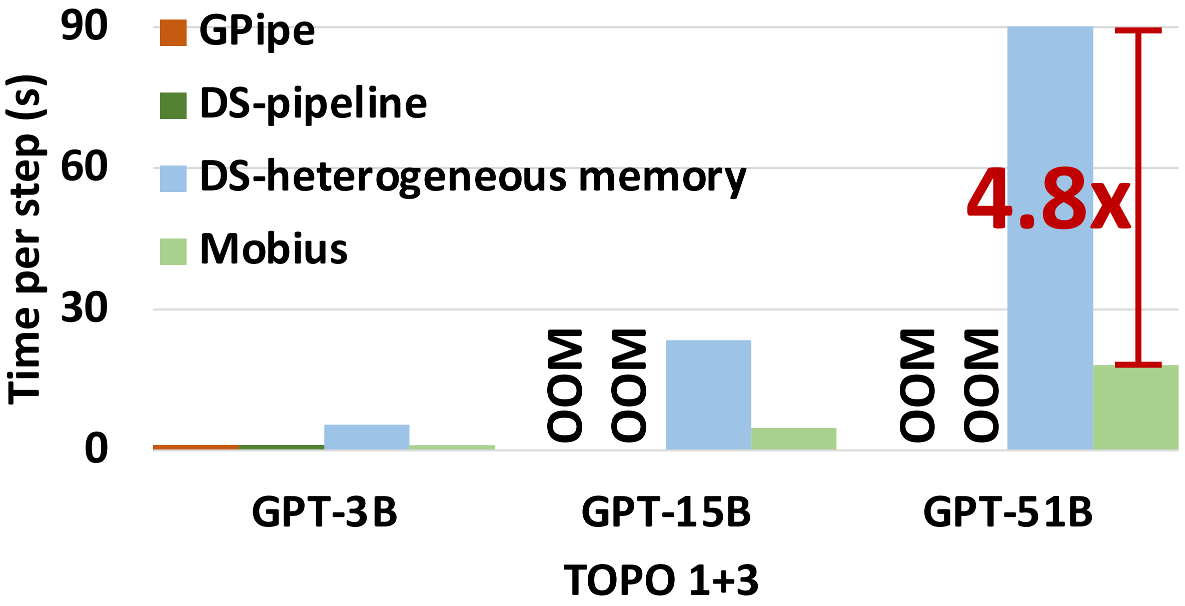


GPU Topologies

Overall Results

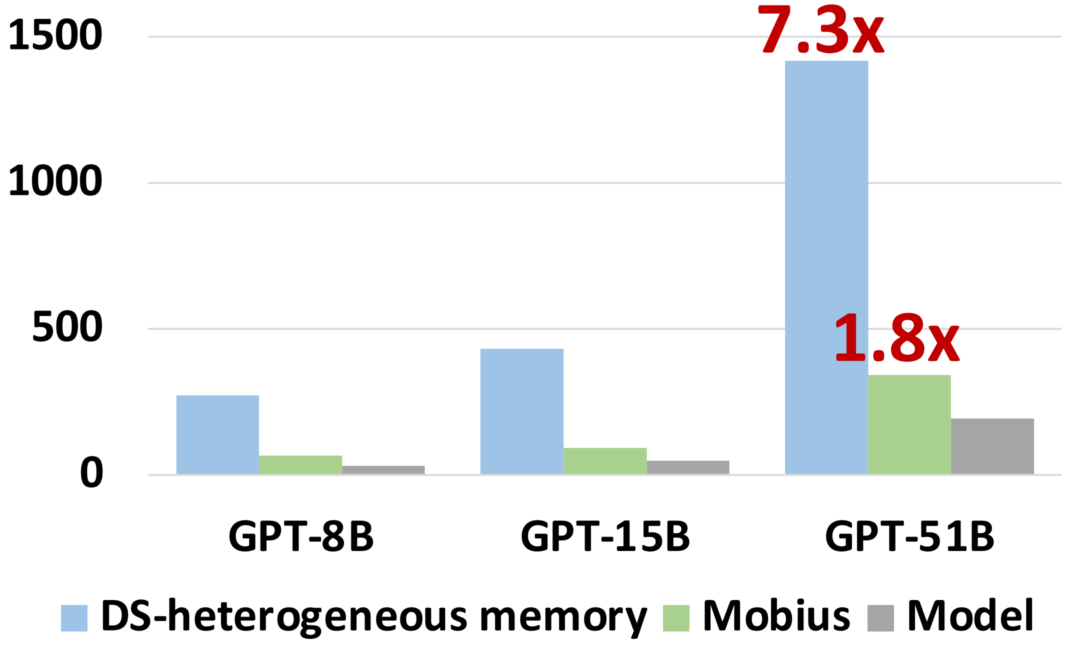
- Mobius and DeepSpeed with heterogeneous memory mode are able to **train larger models**
- Mobius **decreases per-step training time**
- Mobius brings more **significant performance improvement** when the GPU topology has **more severe communication contention**

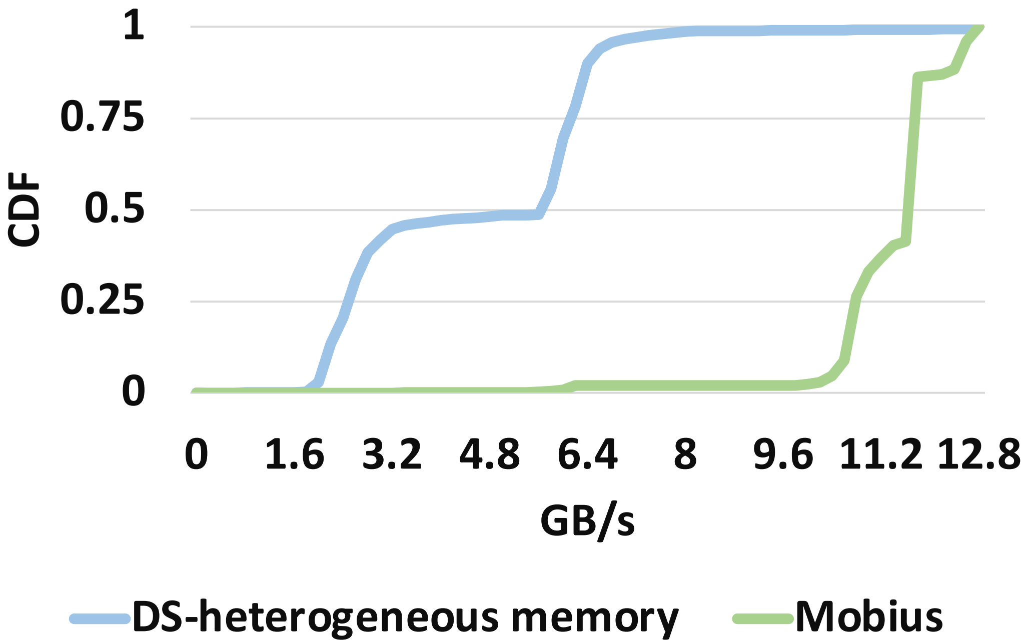




Communication Analysis

- DeepSpeed with heterogeneous memory mode requires frequent GPU all-to-all collective communications, while Mobius pipeline only transfers **small activations and activation gradients**
- **More than half of the data** is transferred at a bandwidth of more than 12 GB/s in Mobius





Conclusion

- Commodity GPU server is an **affordable** option for fine-tuning large-scale models However, communication resources on commodity GPU servers are scarce
- We propose **Mobius** to reduce communication traffic and mitigate communication contention problem
  - **Mobius pipeline**: heterogeneous memory-based pipeline training scheme
  - **Mobius partition**: find the optimal partition scheme
  - **Cross mapping**: mitigate communication contention
- Mobius significantly reduces the training time by **3.8-5.1 times** compared with the prior art

Thanks

Mobius: Fine Tuning Large-Scale Models on Commodity GPU Servers

Yangyang Feng, Minhui Xie, Zijie Tian, Shuo Wang, Youyou Lu, and Jiwu Shu

Tsinghua University

http://storage.cs.tsinghua.edu.cn
Email: fyy21@mails.tsinghua.edu.cn