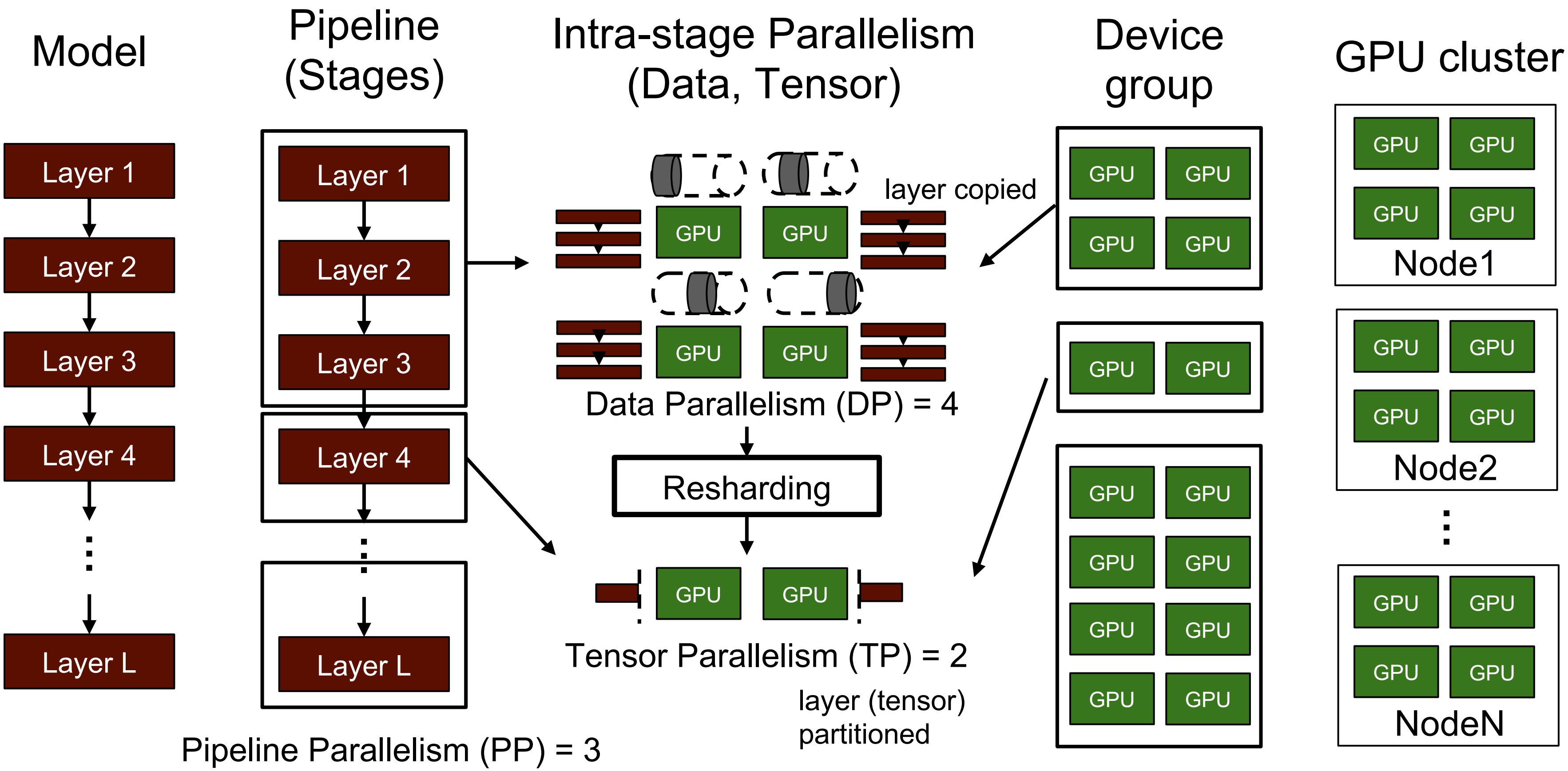
Metis: Fast Automatic Distributed Training on Heterogeneous GPUs

***Taegeon Um§** , *Byungsoo Oh§ , *Minyoung Kang§ , Woo-Yeon Lee§ , Goeun Kim§
Dongseob Kim§ , Youngtaek Kim§ , Mohd Muzzammil§ and Myeongjae Jeon¶

§ Samsung Research ¶ UNIST

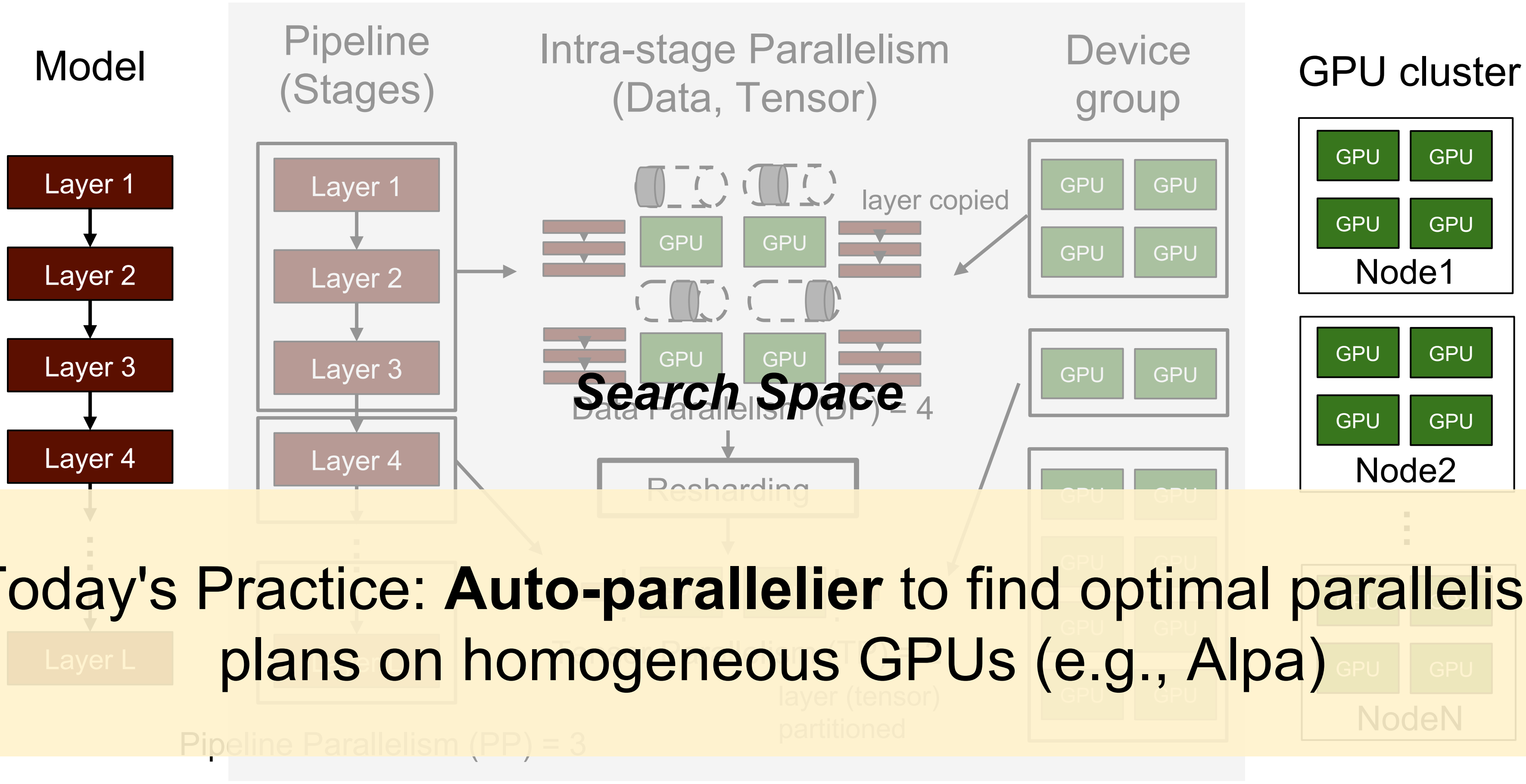
* Equal contribution

Automatic Distributed Training of Large Model



Automatic Distributed Training of Large Model

Samsung Research

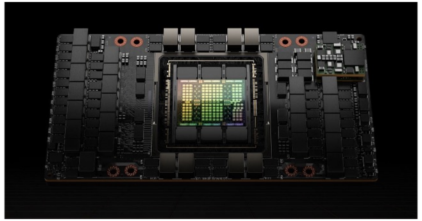


Today's Practice: **Auto-parallelier** to find optimal parallelism plans on homogeneous GPUs (e.g., Alpa)

Heterogeneous GPUs in GPU Clusters

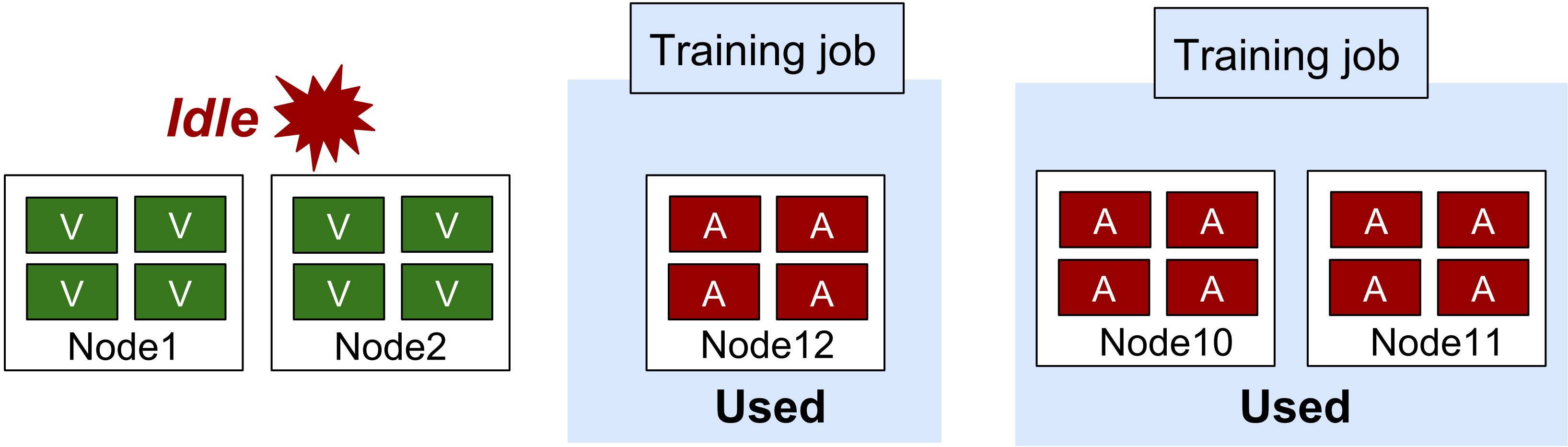
V100, Volta ('17) <

A100, Ampere ('20) <

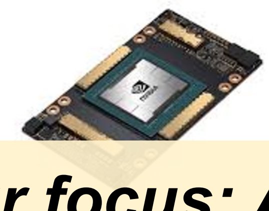
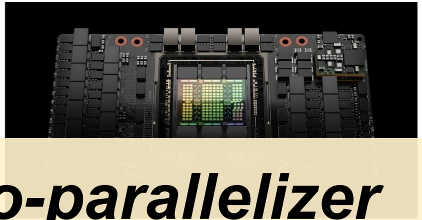


H100, Hopper ('23) <

B100, Blackwell ('24?)



Heterogeneous GPUs in GPU Clusters

Our focus: Auto-parallelizer on heterogeneous GPUs

V100, Volta ('17) < A100, Ampere ('20) < H100, Hopper ('23) < B100, Blackwell ('24?)

Training job

Distributed training on heterogeneous GPUs!

Node1	Node2	Node12
V V V V	V V V V	A A A A

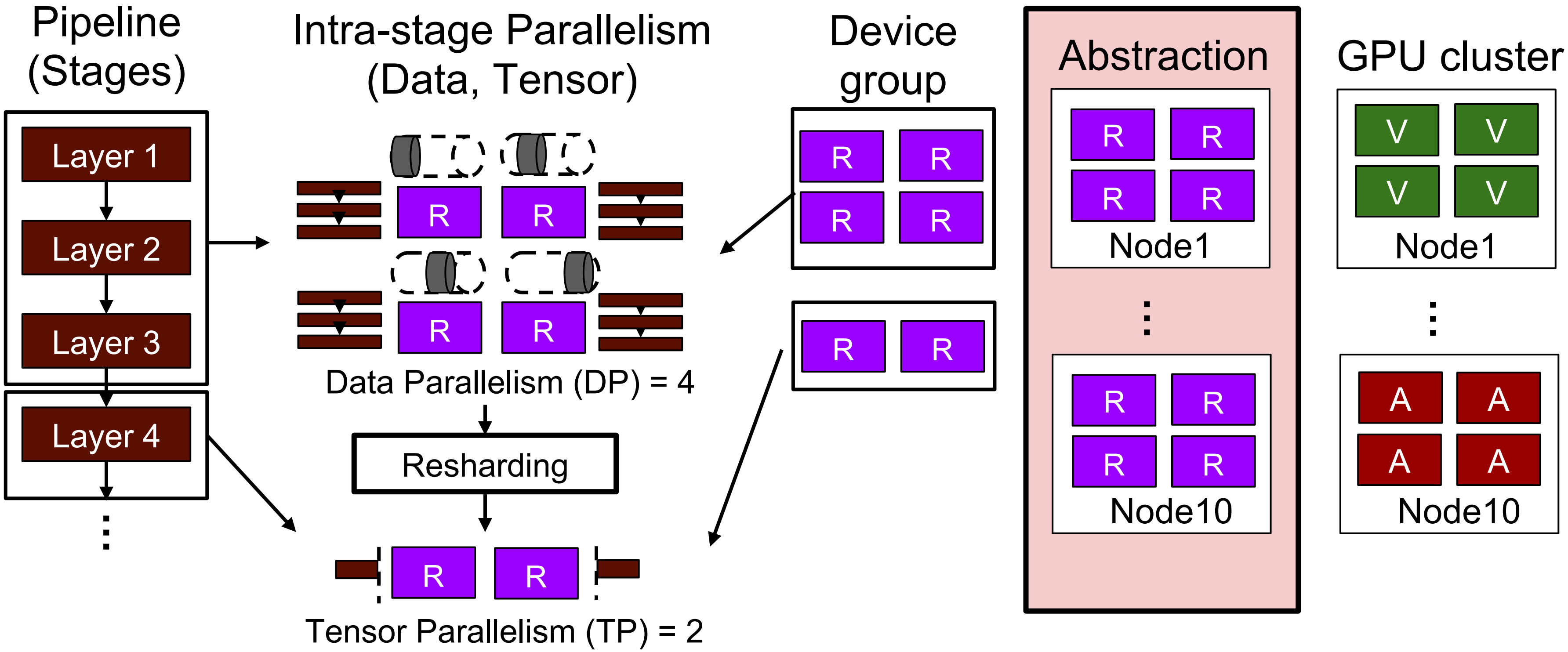
Good performance, Good utilization 😊

Training job

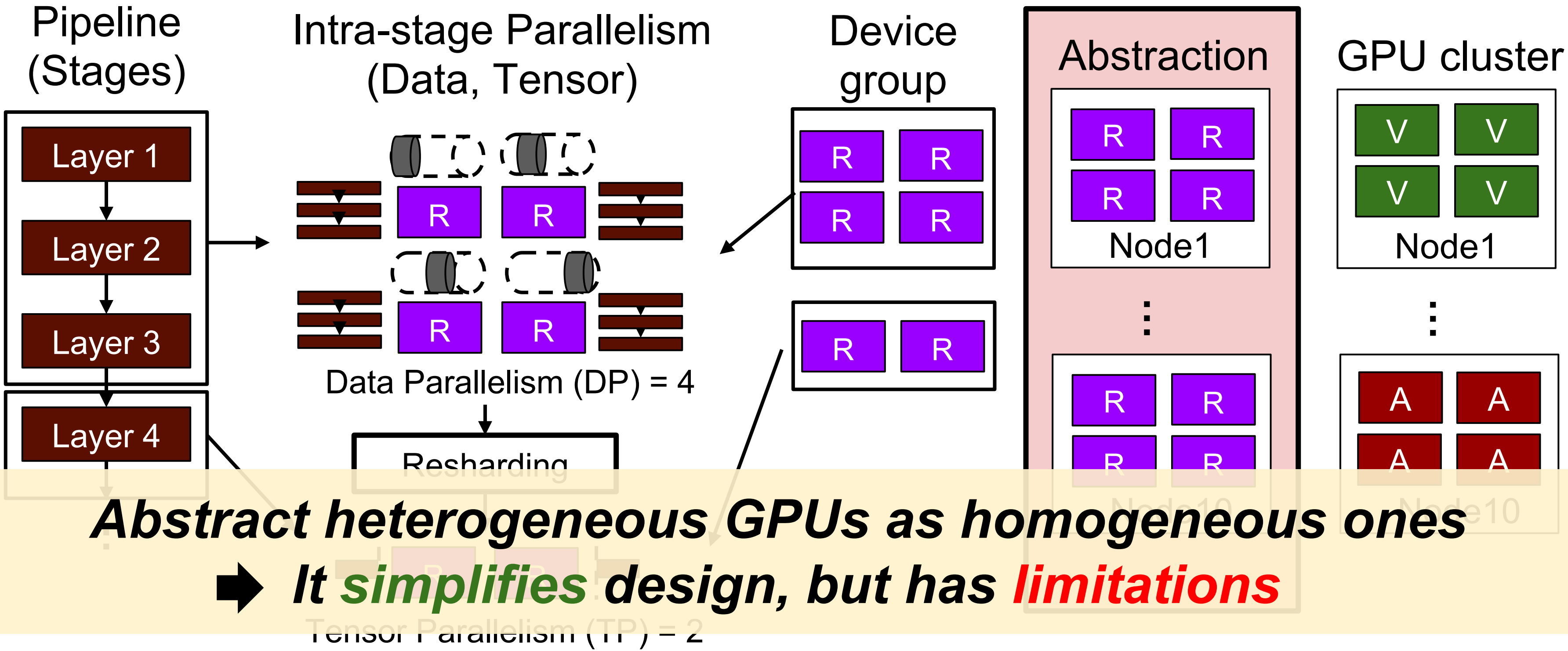
Node10	Node11
A A A A	A A A A

Used

Existing Auto-Parallelizer on Heterogeneous GPUs



Existing Auto-Parallelizer on Heterogeneous GPUs

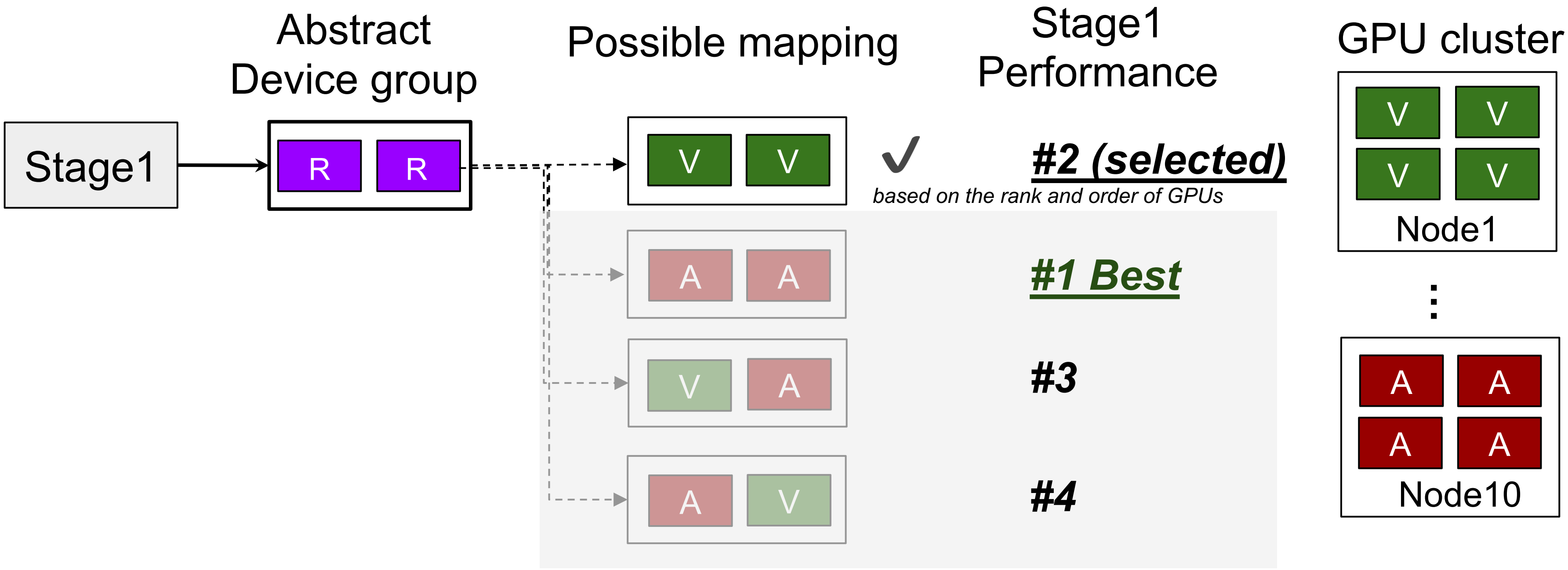


Abstract heterogeneous GPUs as homogeneous ones

➡ It simplifies design, but has limitations

Limitations of Existing Work

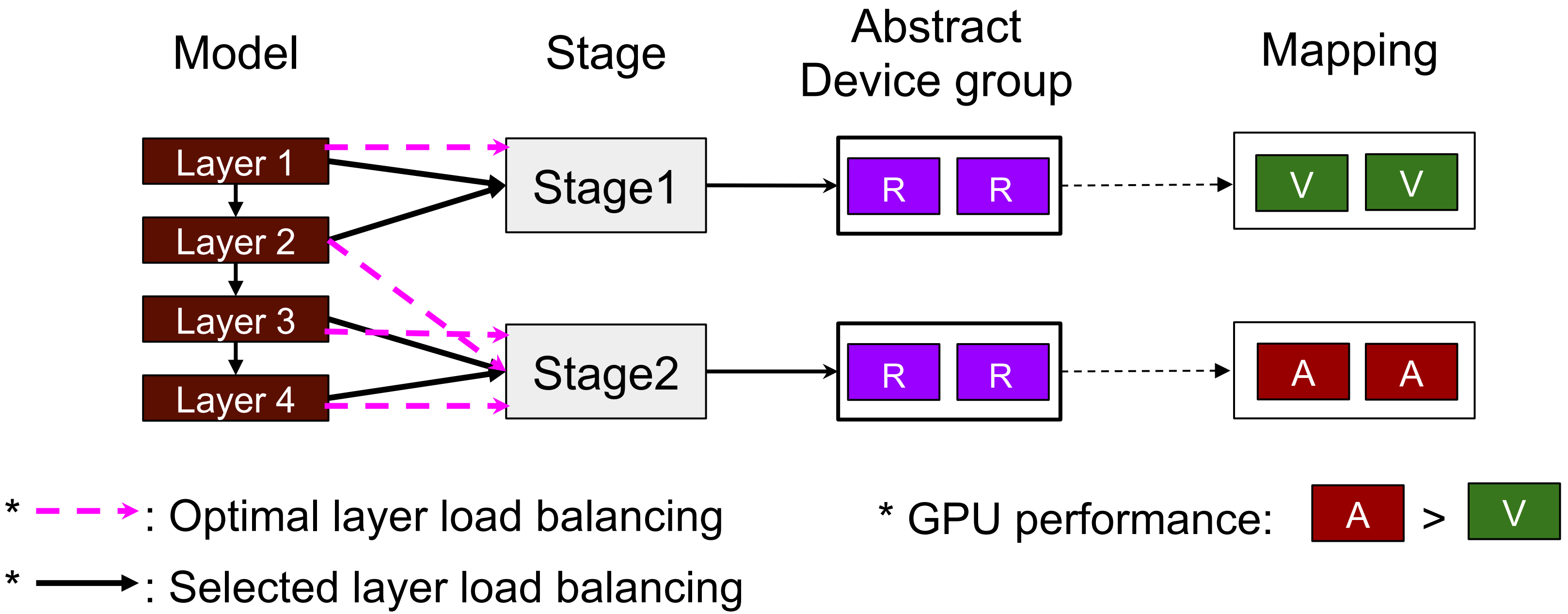
1) *Unexplored device groups*



Not explored

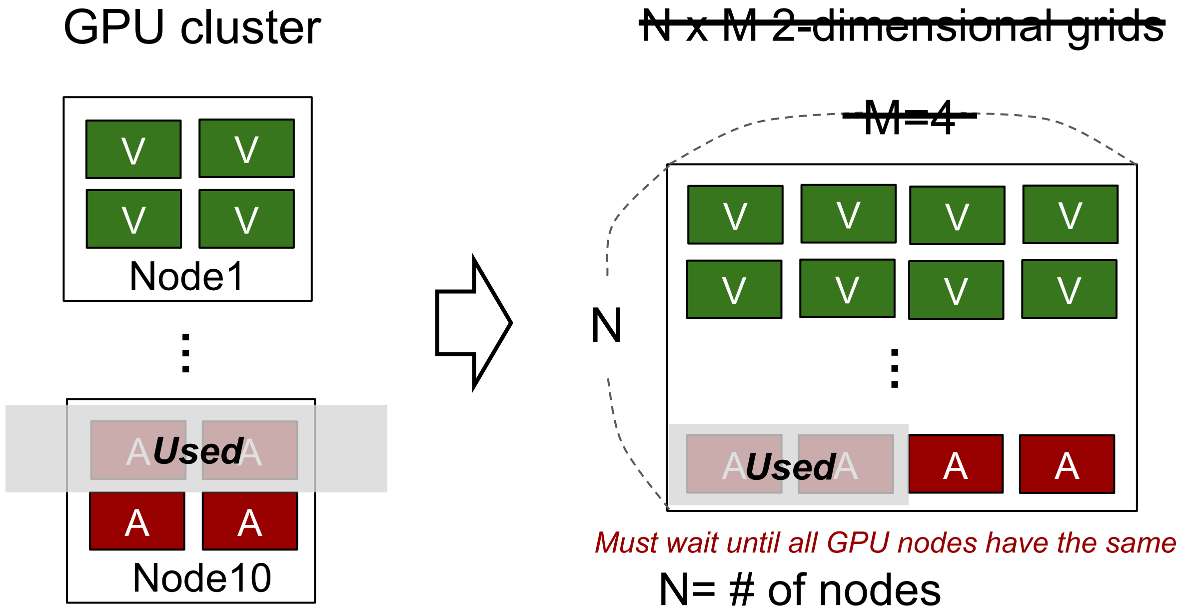
Limitations of Existing Work

2) *Load balancing based on the number of GPUs*



Limitations of Existing Work

3) *Cluster shape constrained by 2-D grids*



Must wait until all GPU nodes have the same number of available GPUs

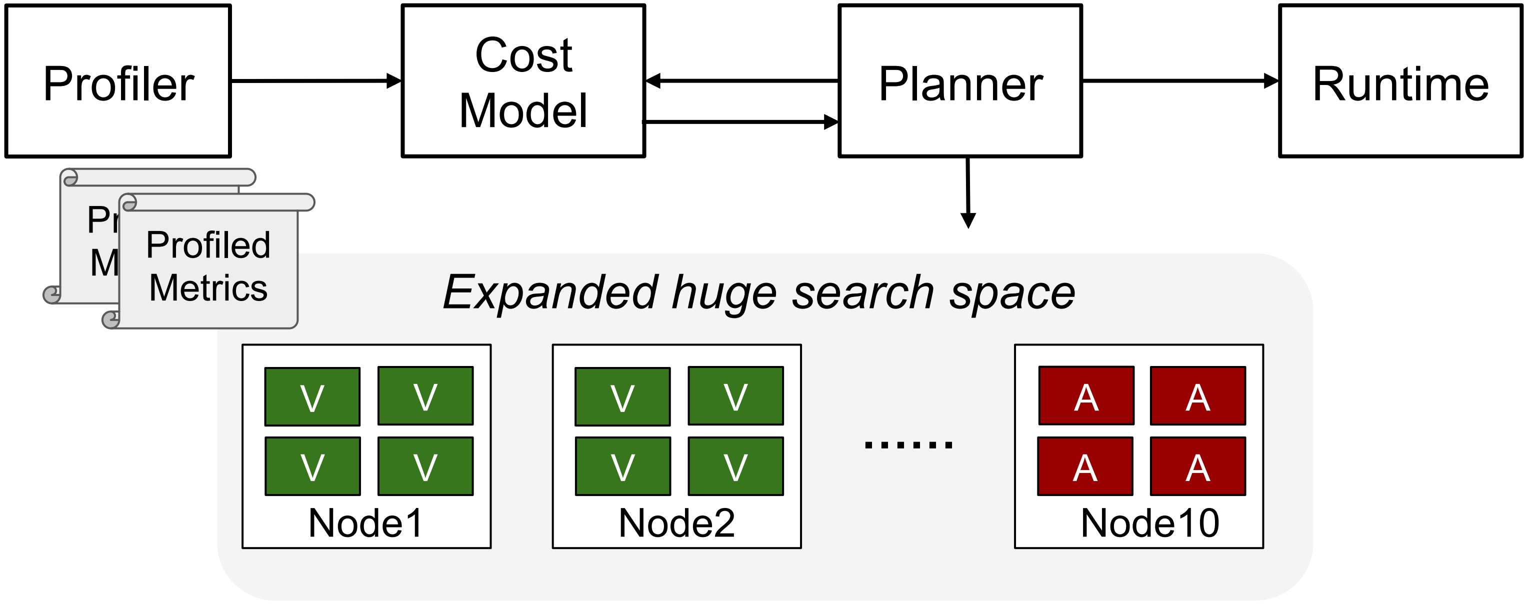
N= # of nodes

M=# of GPUs per node

Breaks 2-D abstraction ⇒ Not supported

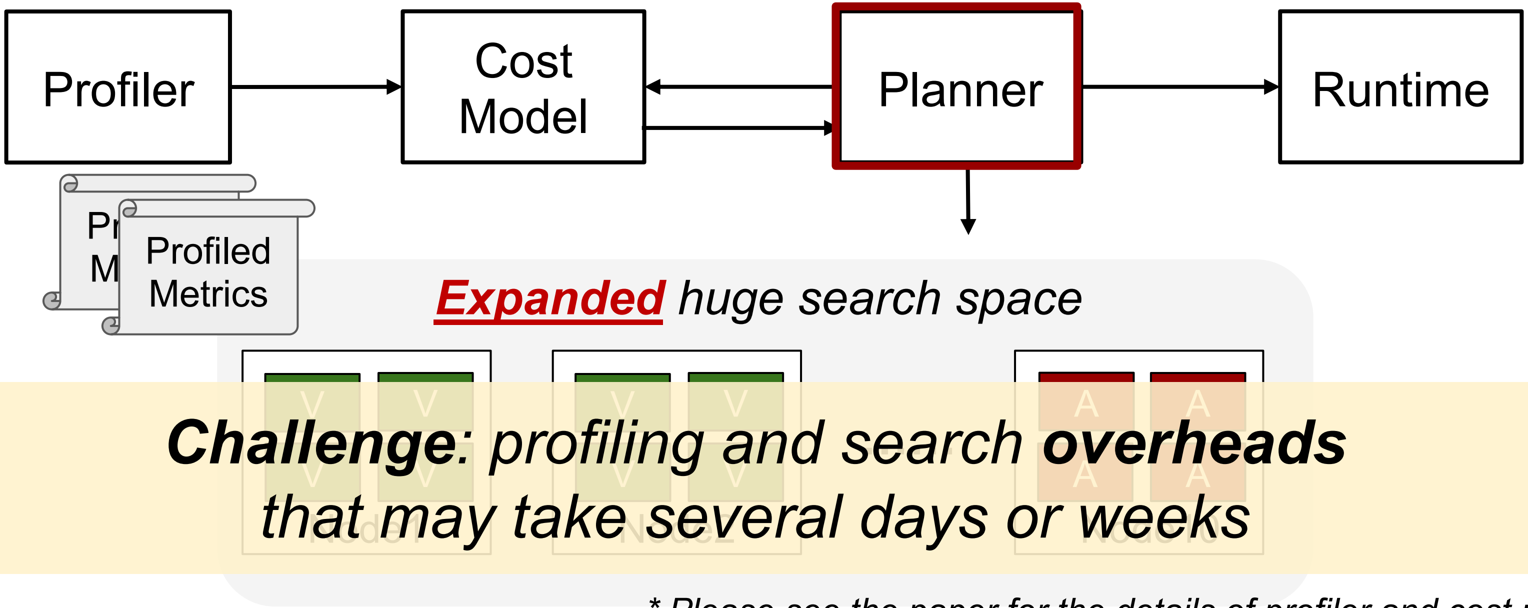
Metis: Overview

<u>Expands</u> the search space of plans *by being aware of heterogeneous computing, memory, and number of GPUs*



Metis: Overview

***Expands** the search space of plans* *by being aware of heterogeneous computing, memory, and number of GPUs*

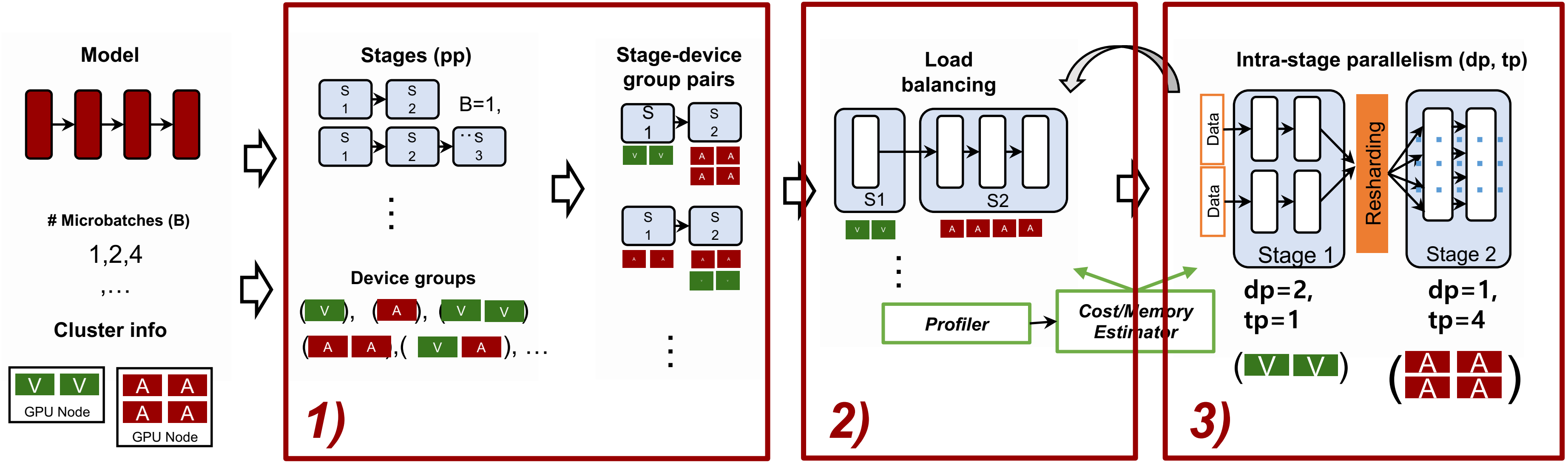


***Challenge**: profiling and search **overheads** that may take several days or weeks*

** Please see the paper for the details of profiler and cost model*

Metis: Hetero-Aware Search Algorithm

1) *Pruning inefficient/similar combinations of stage-device group pairs*
2) *Balancing layers across stages with capacity-aware allocation*
3) *Navigating efficient intra-stage plans based on DP/TP characteristics*



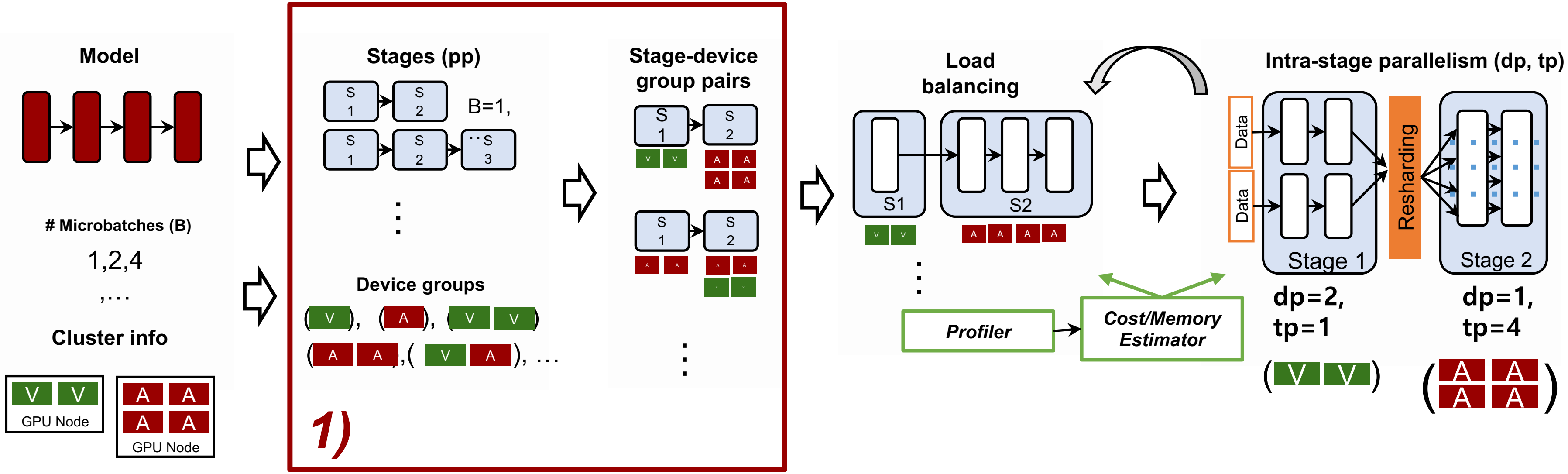
 [ATC '24] Metis: Fast Automatic Distributed Training on Heterogeneous GPUs

Metis: Hetero-Aware Search Algorithm

1) ***Pruning inefficient/similar combinations of stage-device group pairs***

2) *Balancing layers across stages with capacity-aware allocation*

3) *Navigating efficient intra-stage plans based on DP/TP characteristics*



[ATC '24] Metis: Fast Automatic Distributed Training on Heterogeneous GPUs

Samsung Research

Metis: Hetero-Aware Search Algorithm

1) *Pruning inefficient combinations of stage-device group pairs*

* DG=Device Group

		Stage1	Stage2	Stage3	Stage4
Pair#1		DG1	DG2	DG3	DG4
Pair#2	4!	DG1	DG2	DG4	DG3
Pair#3		DG1	DG4	DG2	DG3
⋮			⋮		
		DG5	DG6	DG7	DG8
⋮	4!	DG5	DG6	DG8	DG7
		DG5	DG8	DG6	DG7

Metis: Hetero-Aware Search Algorithm

1) *Pruning inefficient combinations of stage-device group pairs*

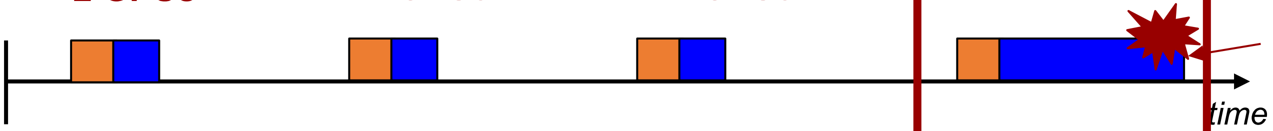
* DG=Device Group

	Stage1	Stage2	Stage3	Stage4
Computational load:	2	2	2	32
	DG1	DG2	DG3	DG4
	2 GPUs	**2 GPUs**	**2 GPUs**	**32 GPUs**

: comp.

: comm.

Imbalanced intra-stage comm.

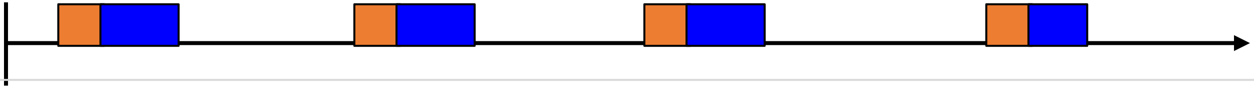


High intra-stage comm.
Bottleneck in the pipeline

Pipeline execution time ∝ the longest stage execution time

Computational load:	10	10	10	8
	DG1	DG2	DG3	DG4
	10 GPUs	**10 GPUs**	**10 GPUs**	**8 GPUs**

Balanced intra-stage comm.
⇒ Better performance



Metis: Hetero-Aware Search Algorithm

1) Pruning inefficient combinations of stage-device group pairs

* DG=Device Group

	Stage1	Stage2	Stage3	Stage4
Computational load:	2	2	2	32
	DG1	DG2	DG3	DG4
	2 GPUs	2 GPUs	2 GPUs	32 GPUs

: comp.

: comm.

Imbalanced intra-stage comm.

High intra-stage comm.
Bottleneck in the pipeline

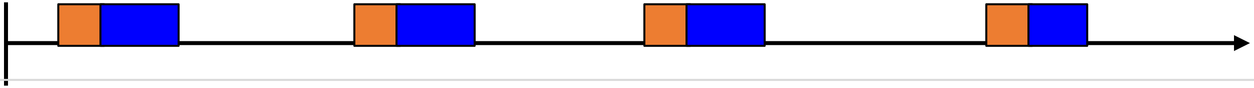
time

Pruning highly skewed size of device groups

Pipeline execution time ∝ the longest stage execution time

	DG1	DG2	DG3	DG4
Computational load:	10	10	10	8
	10 GPUs	10 GPUs	10 GPUs	8 GPUs

Balanced intra-stage comm.
⇒ Better performance



Metis: Hetero-Aware Search Algorithm

1) *Pruning similar combinations of stage-device group pairs*

Repeated layers

	Embed	*Transformer*	*Transformer*	*Transformer*	*Transformer*	*LM Head*
	Stage1	Stage2	Stage3	Stage4	Stage5	Stage6
Pair#1	DG1 V	DG2 V	DG3 V	DG4 V	DG5 V	DG6 A
Pair#2	DG1 V	DG2 V	DG3 V	DG4 V	DG6 A	DG5 V
Pair#3	DG1 V	DG2 V	DG3 V	DG6 A	DG4 V	DG5 V
Pair#4	DG1 V	DG2 V	DG6 A	DG3 V	DG4 V	DG5 V
Pair#5	DG1 V	DG6 A	DG2 V	DG3 V	DG4 V	DG5 V
Pair#6	DG6 A	DG1 V	DG2 V	DG3 V	DG4 V	DG5 V

Metis: Hetero-Aware Search Algorithm

1) *Pruning similar combinations of stage-device group pairs*

Repeated layers

	Embed	*Transformer*	*Transformer*	*Transformer*	*Transformer*	*LM Head*
	Stage1	Stage2	Stage3	Stage4	Stage5	Stage6
Pair#1	DG1 V	DG2 V	DG3 V	DG4 V	DG5 V	DG6 A
Pair#2	DG1 V	DG2 V	DG3 V	DG4 V	DG6 A	DG5 V
Pair#3	DG1 V	DG2 V	DG3 V	DG6 A	DG4 V	DG5 V
Pair#4	DG1 V	DG2 V	DG6 A	DG3 V	DG4 V	DG5 V
Pair#5	DG1 V	DG6 A	DG2 V	DG3 V	DG4 V	DG5 V
Pair#6	DG6 A	DG1 V	DG2 V	DG3 V	DG4 V	DG5 V

Pruning pairs that have the same input/output patterns (shapes), same computations / loads

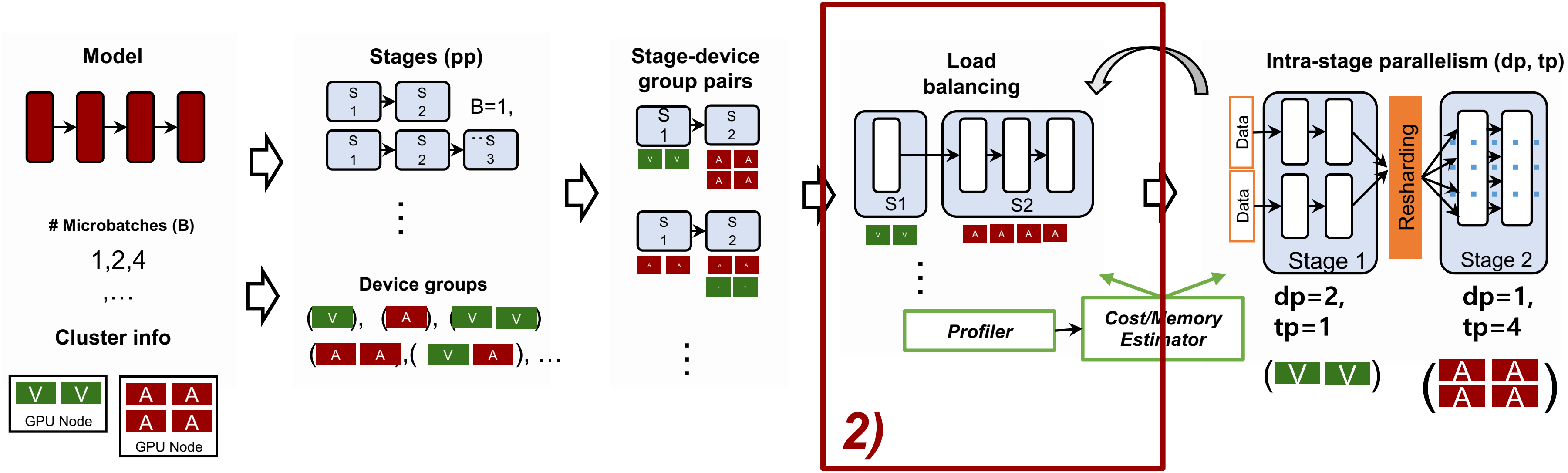
[ATC '24] Metis: Fast Automatic Distributed Training on Heterogeneous GPUs

Metis: Hetero-Aware Search Algorithm

1) Pruning inefficient/similar combinations of stage-device group pairs

2) Balancing layers across stages with capacity-aware allocation

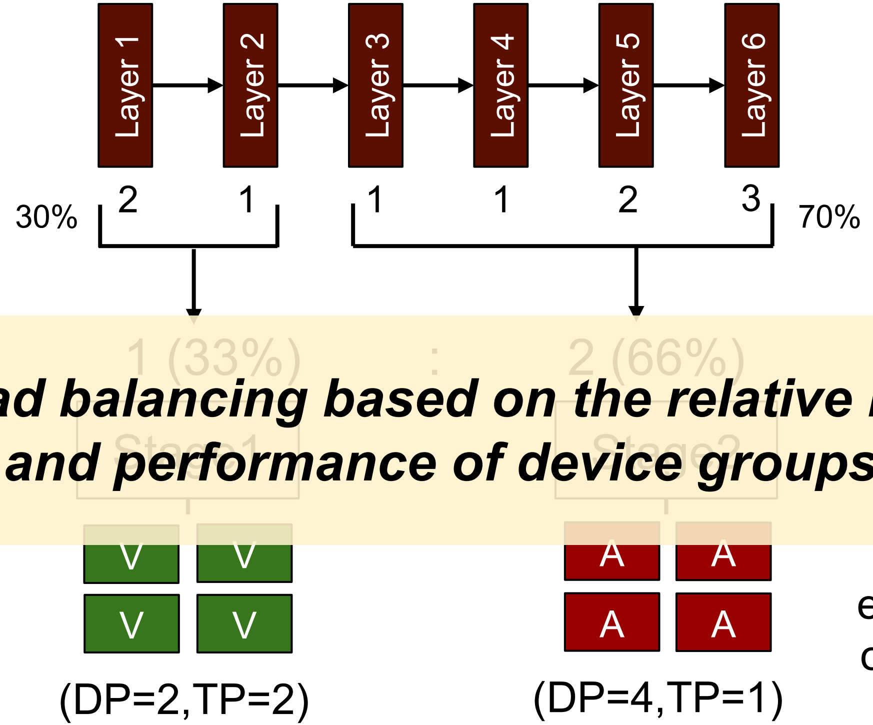
3) Navigating efficient intra-stage plans based on DP/TP characteristics



Metis: Hetero-Aware Search Algorithm

2) *Balancing layers across stages with capacity-aware allocation*

Layer load



Relative performance

O(L) layer load balancing based on the relative load of layers and performance of device groups

Estimated execution time of a model: 10s

Estimated execution time of a model: 5s

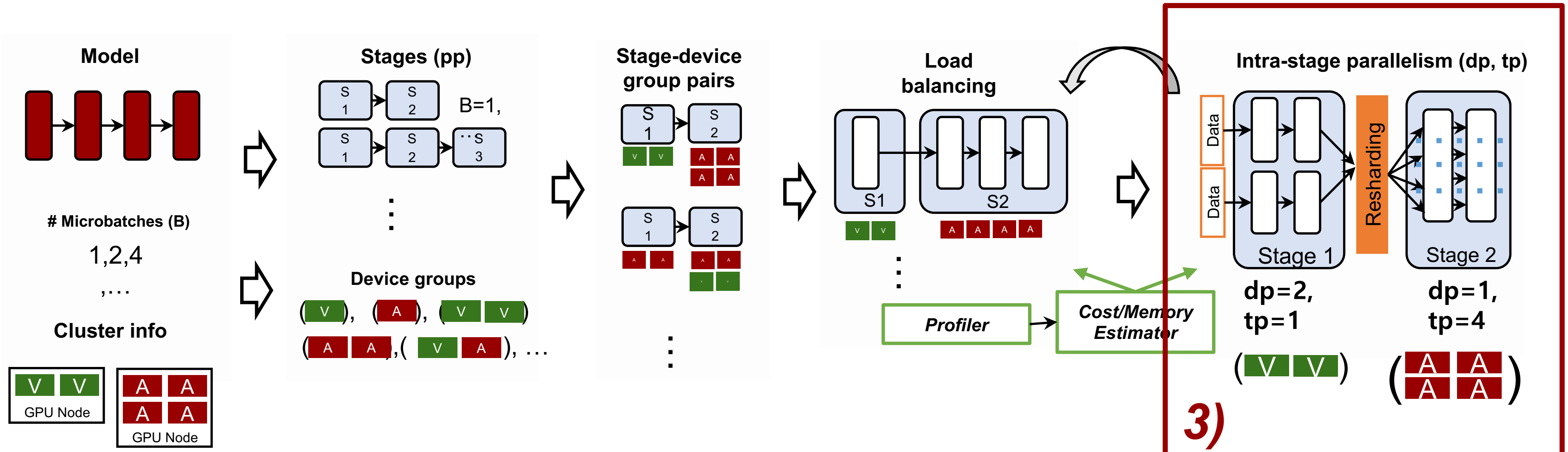
Assigned device groups

Metis: Hetero-Aware Search Algorithm

1) *Pruning inefficient/similar combinations of stage-device group pairs*

2) *Balancing layers across stages with capacity-aware allocation*

3) ***Navigating efficient intra-stage plans based on DP/TP characteristics***



[ATC '24] Metis: Fast Automatic Distributed Training on Heterogeneous GPUs

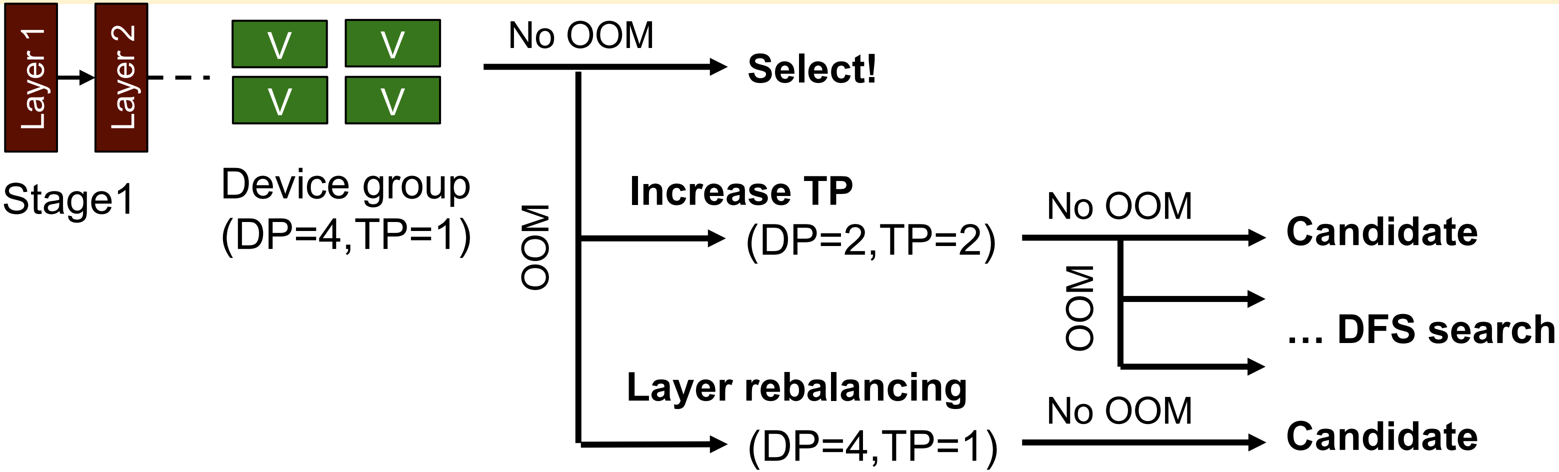
Metis: Hetero-Aware Search Algorithm

3) Navigating efficient intra-stage plans based on DP/TP characteristics

- *DP => All-reduce for gradient after mini-batch*
- *TP => All-reduce for each forward/backward*

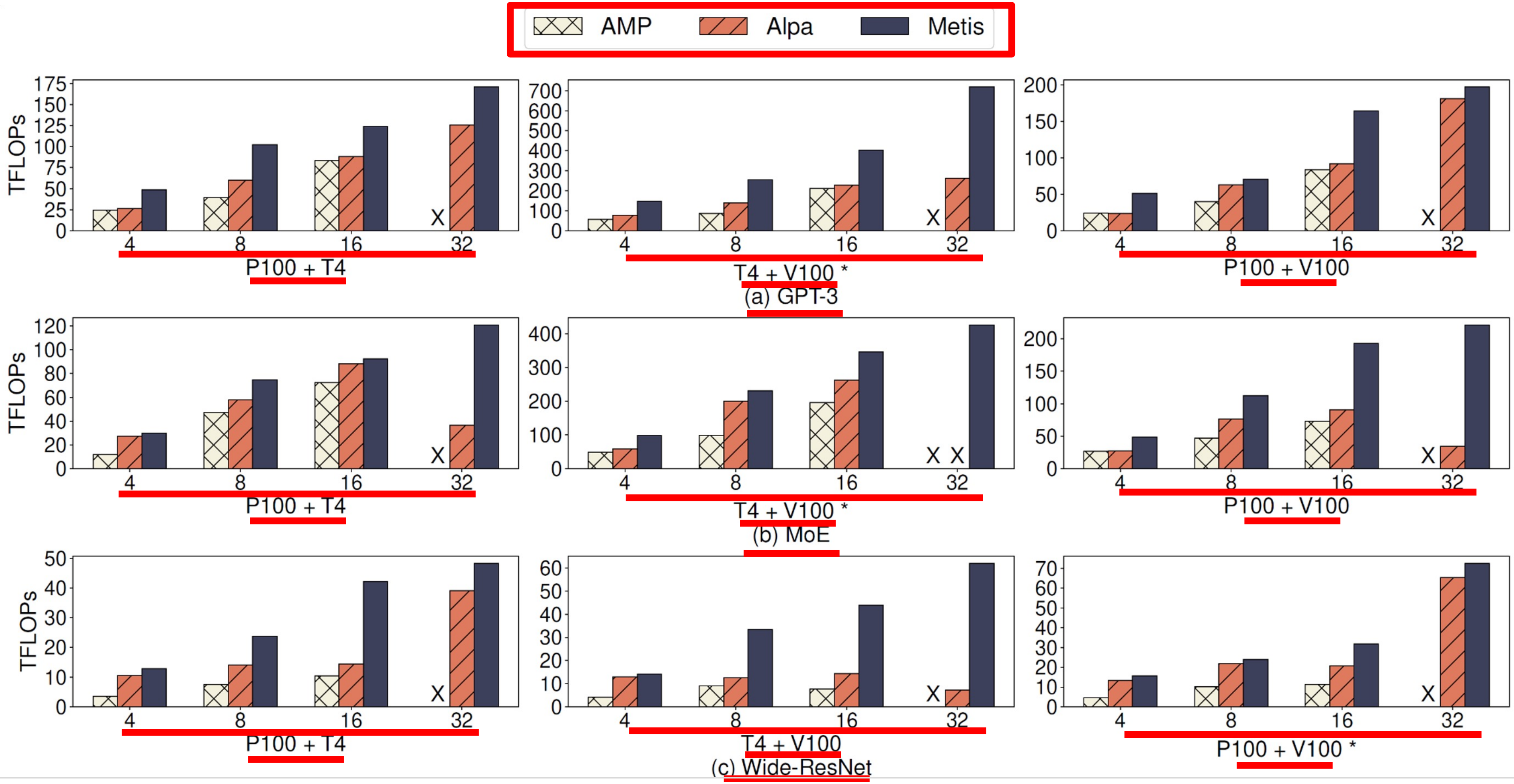
Efficient DFS search that prioritizes DP over TP with OOM detection

- *TP comm. overhead > DP comm. overhead*

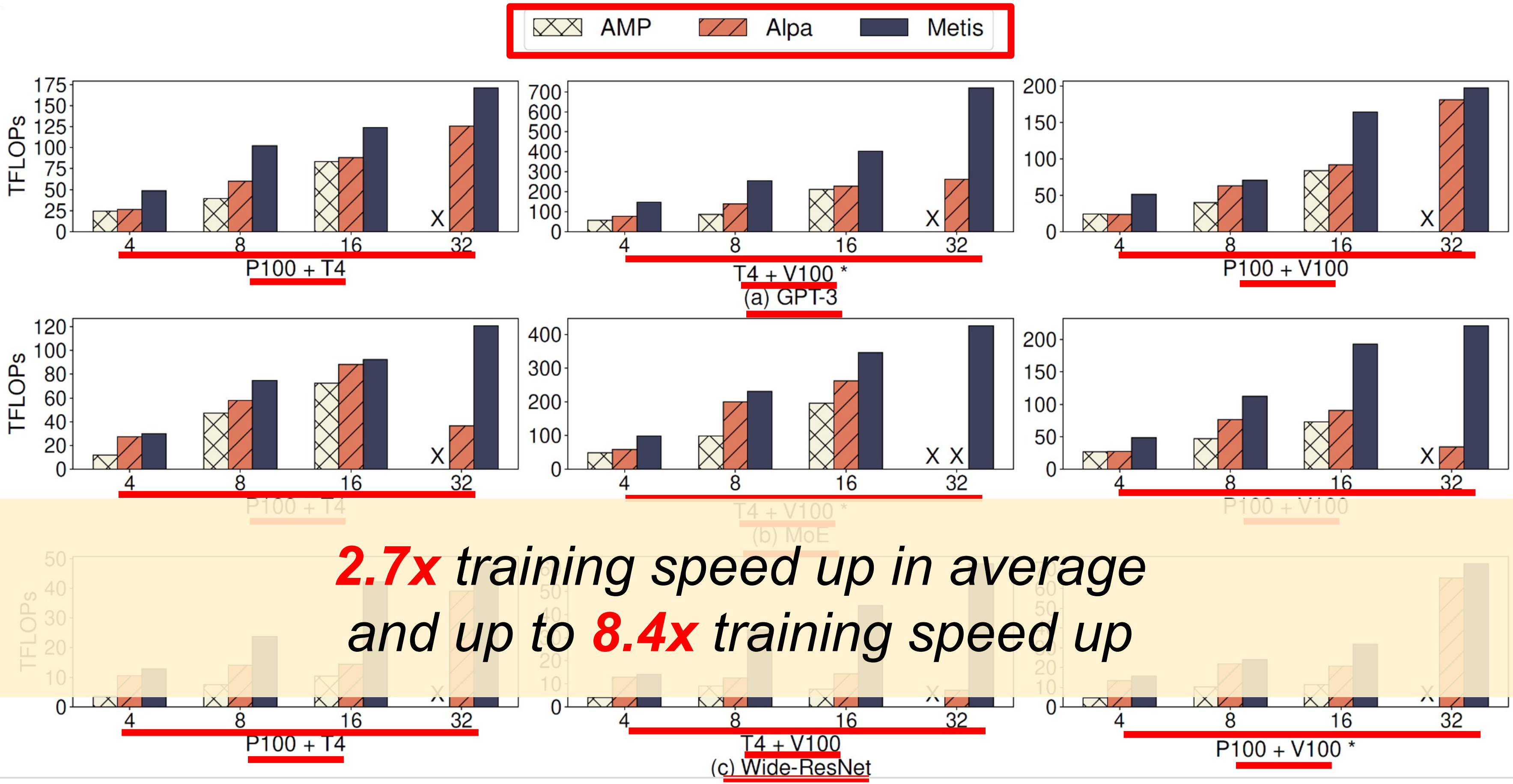


Evaluation: Training Speed

Samsung Research



Evaluation: Training Speed

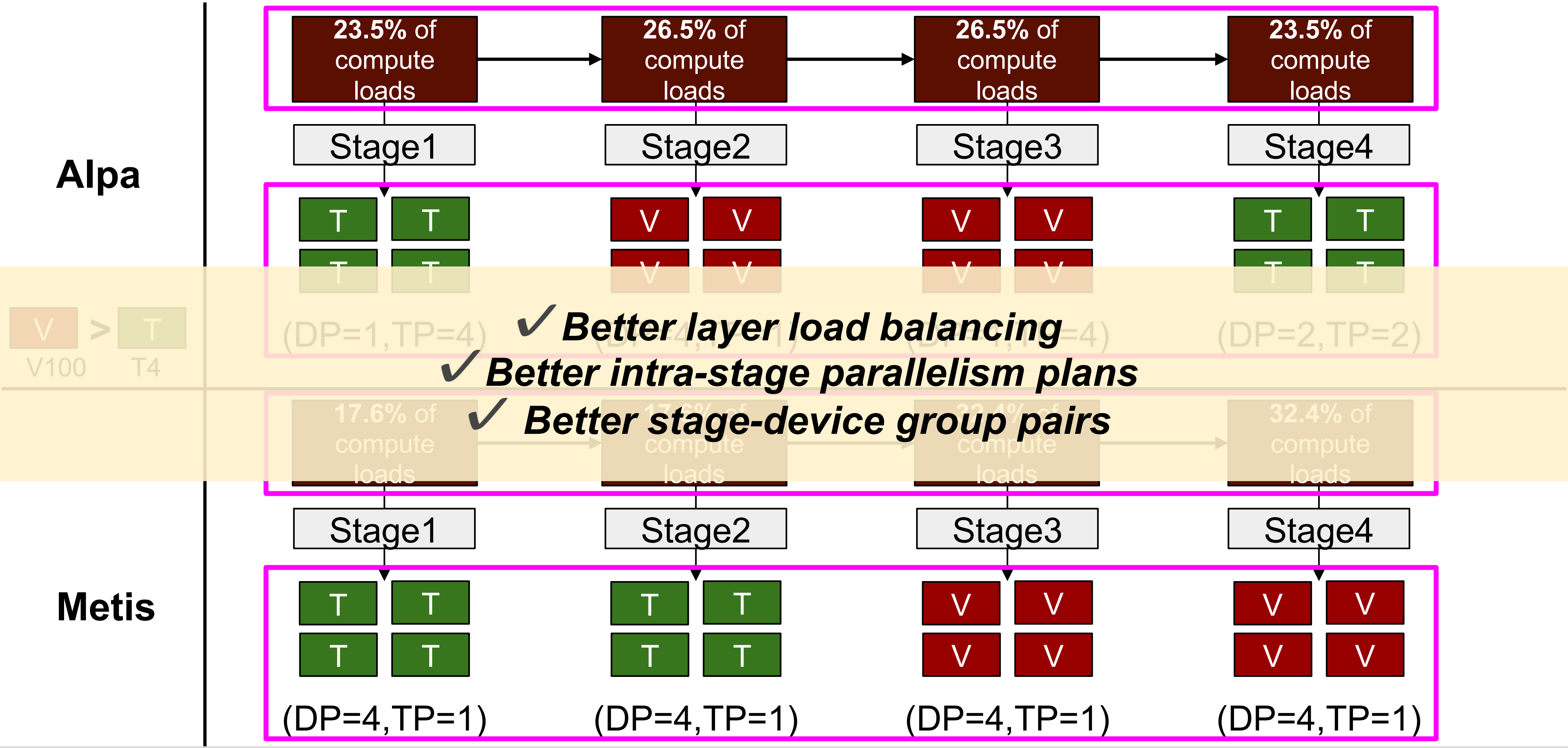


***2.7x** training speed up in average*
*and up to **8.4x** training speed up*

[ATC '24] Metis: Fast Automatic Distributed Training on Heterogeneous GPUs

Evaluation: Plan Comparison

** GPT-3, 16 GPUs, T4+V100*



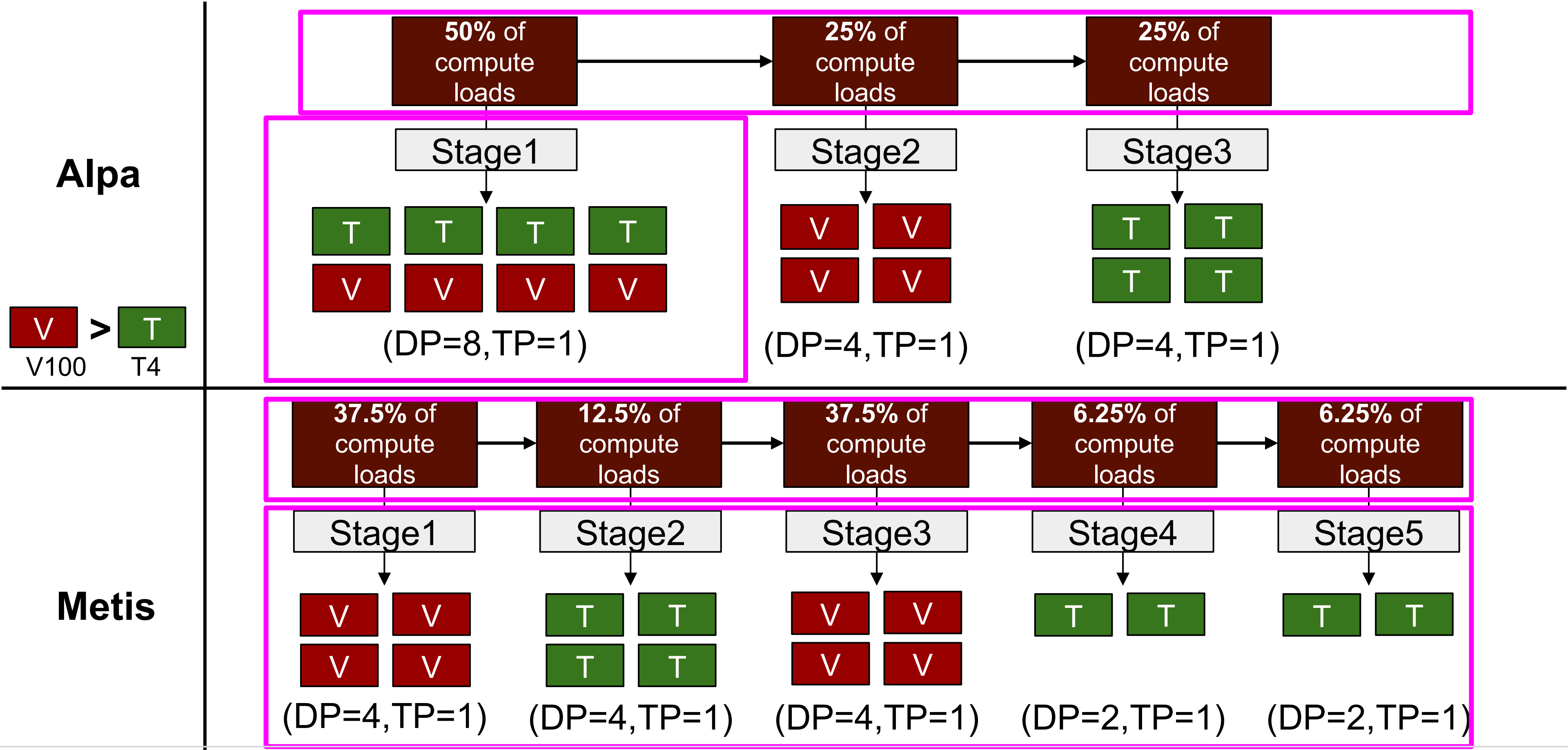
✓ ***Better layer load balancing***

✓ ***Better intra-stage parallelism plans***

✓ ***Better stage-device group pairs***

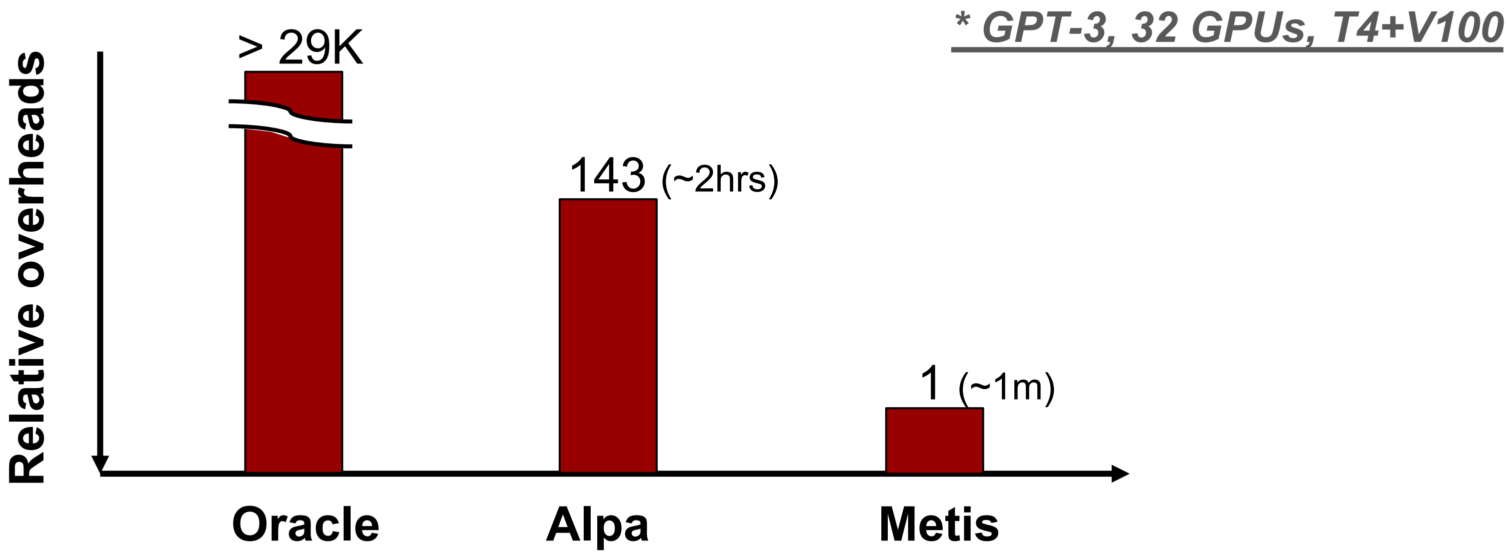
[ATC '24] Metis: Fast Automatic Distributed Training on Heterogeneous GPUs

Evaluation: Plan Comparison *<u>* MoE, 16 GPUs, T4+V100</u>*



[ATC '24] Metis: Fast Automatic Distributed Training on Heterogeneous GPUs

Evaluation: Searching Overhead



***143x** (vs Alpa) and **29000x** (vs Oracle) search speed-up*
while finding near-optimal plans (5% diff compared to Oracle)

Conclusion

- *Show* that the **existing auto-parallelizer is not optimized for** distributed training on heterogeneous GPUs
- *Design* **Metis**, a system that automatically finds good parallelism plans on heterogeneous GPUs **by expanding the search space**
- *Develop* a **new hetero-aware search algorithm** that prunes inefficient plans, balances layers based on capacity-awareness, and finds efficient intra-stage parallelism while prioritizing DP over TP
- *Evaluate* that Metis **improves training speed by up to 8x with less search overheads compared to SoTA**
- *Believe* that Metis can be adopted to **improve training speed and GPU utilization** and used for **cost-efficient distributed training** on heterogeneous GPU clusters

Samsung Research

Thank you

taegeon.um@samsung.com
https://taegeonum.github.io