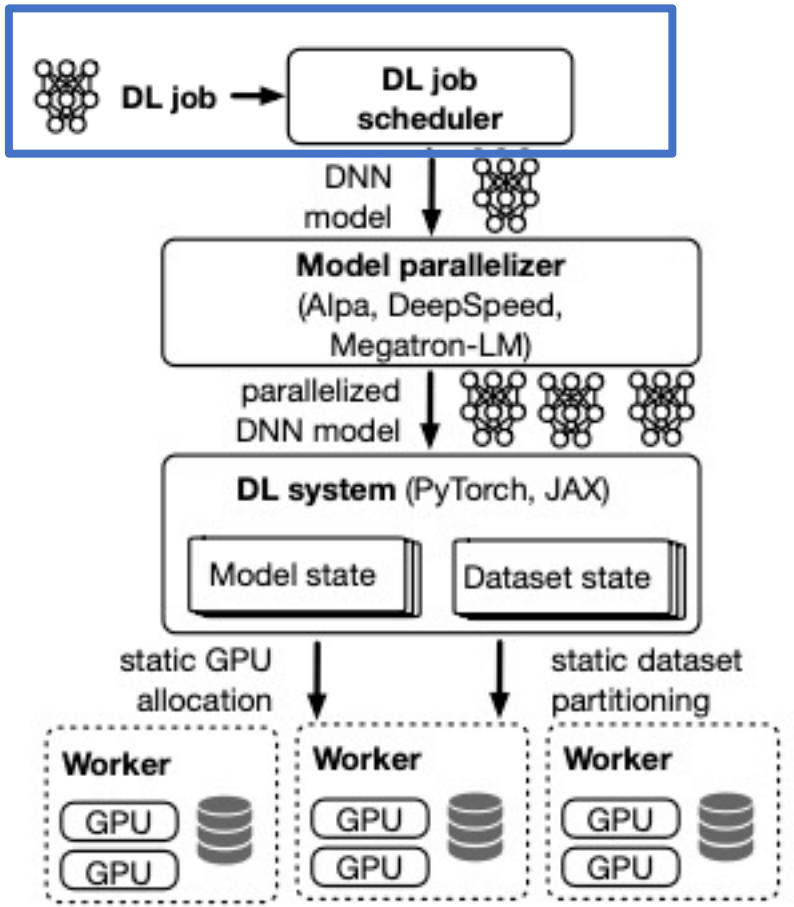
Dynamic Training Parallelization

Zhanghan Wang

2024/12/02

What can we do on top of them?



Dynamic Training Parallelization

Auto parallelization (Haitian/Hexu)

Torch Tensor Implementation (Haitian)

Collective Communication Lib (Tao)
DL Compilers (David)

Tenplex: Dynamic Parallelism for Deep Learning using Parallelizable Tensor Collections

Marcel Wagenländer
Imperial College London

Guo Li
Imperial College London

Bo Zhao
Aalto University

Luo Mai
University of Edinburgh

Peter Pietzuch
Imperial College London

SOSP 2024

Enabling Parallelism Hot Switching for Efficient Training of Large Language Models

Hao Ge*‡
gehao@stu.pku.edu.cn
Peking University

Fangcheng Fu*†‡
ccchengff@pku.edu.cn
Peking University

Haoyang Li‡
lihaoyang@stu.pku.edu.cn
Peking University

Xuanyu Wang‡
wxyz0001@pku.edu.cn
Peking University

Sheng Lin‡
linsh@stu.pku.edu.cn
Peking University

Yujie Wang‡
alfredwang@pku.edu.cn
Peking University

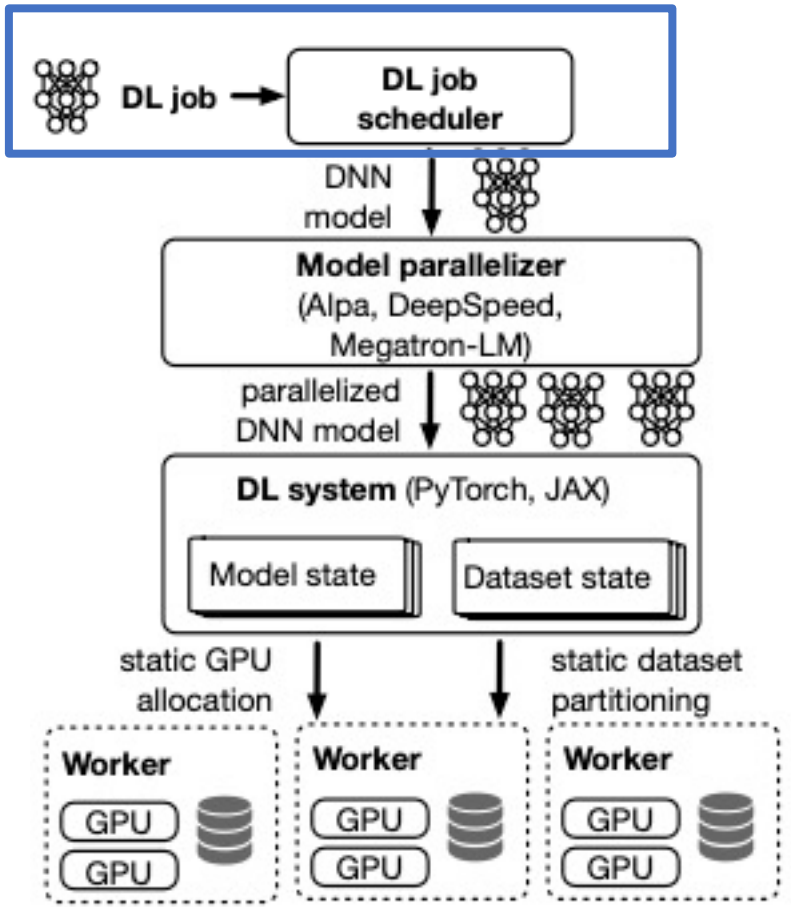
Xiaonan Nie‡
xiaonan.nie@pku.edu.cn
Peking University

Hailin Zhang‡
z.hl@pku.edu.cn
Peking University

Xupeng Miao
xupeng@purdue.edu
Purdue University

Bin Cui†‡§
bin.cui@pku.edu.cn
Peking University

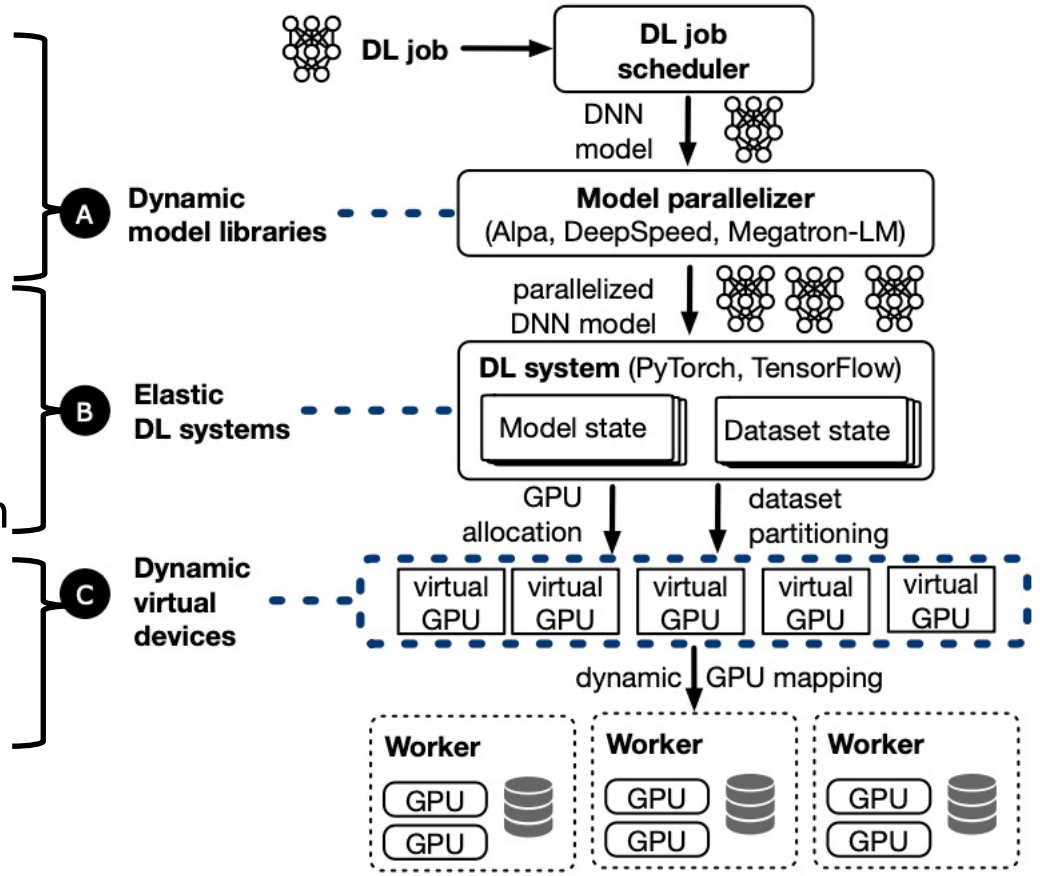
Dynamic Training Parallelization



- Why?
  - Tenplex (resource changes)
    - Elasticity (cloud)
    - Redeployment (hardware maintenance)
    - Failure recovery (GPU failure)
  - HotSPa
    - The optimal parallelism strategies varies for different sequence lengths.
- Thus, we may require changing parallelism strategy dynamically.

Dynamic Training Parallelization

- How is dynamic parallelization supported now?
  - Model parallelizers
    - Alpa, MegatronLM, DeepSpeed…
    - Can re-plan upon resource changing;
    - But require manual re-partitioning.
  - Elastic DL Systems
    - Elastic Horovod, Torch Distributed…
    - Require manual re-partitioning
    - No support for full multi-dimensional parallelism
  - Virtual Devices (Virtualize GPUs)
    - VirtualFlow, EasyScale, Singularity…
    - Only supports dynamic DP

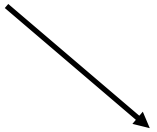


Dynamic Training Parallelization

- How is dynamic parallelization supported now?
  - Model parallelizers
    - Alpa, MegatronLM, DeepSpeed…
    - Can re-plan upon resource changing;
    - But require manual re-partitioning.
  - Elastic DL Systems
    - Elastic Horovod, Torch Distributed…
    - Require manual re-partitioning
    - No support for full multi-dimensional parallelism
  - Virtual Devices (Virtualize GPUs)
    - VirtualFlow, EasyScale, Singularity…
    - Only supports dynamic DP

Challenges or Missing parts

- Auto re-partition
- Full multi-dim parallelism
- Low reconfiguration overhead



Means changing from a parallelism strategy to another

Dynamic Training Parallelization

Challenges or Missing parts

- Auto re-partition
- Full multi-dim parallelism
- Low reconfiguration overhead

} **States Abstraction** of model/data tensors and their transformation

} **Reconfig Planning**: find low-cost reconfig plan to transfer or re-load tensors.

TENPLEX: Dynamic Parallelism for Deep Learning using Parallelizable Tensor Collections

Marcel Wagenländer
Imperial College London

Guo Li
Imperial College London

Bo Zhao
Aalto University

Luo Mai
University of Edinburgh

Peter Pietzuch
Imperial College London

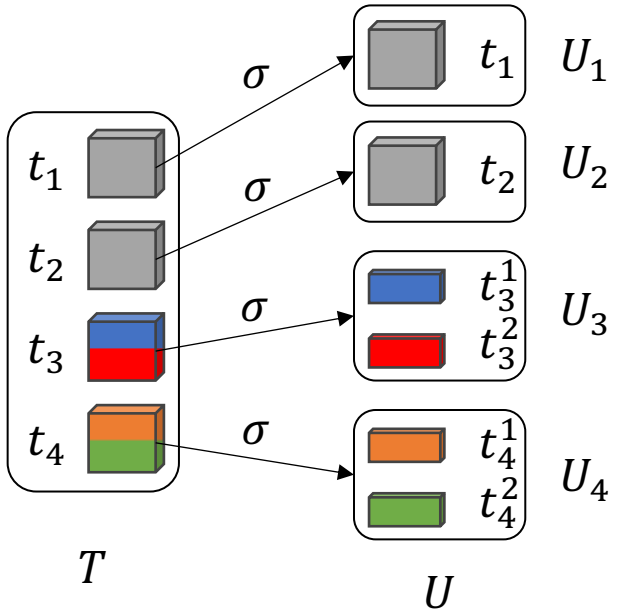
SOSP 2024

Tenplex – State Abstraction

- Parallelizable Tensor Collection (PTC)

$$\text{PTC} = (T, \sigma, \phi, \alpha)$$

  - $T = D \cup M = \{t_1, \ldots, t_n\}$. Here, D for dataset tensors and M for model tensors.
  - σ slices a tensor $t_i \in T$ into sub-tensors $U_1 = \sigma(t_i) = \{t_i^1, \ldots, t_i^m\}$.

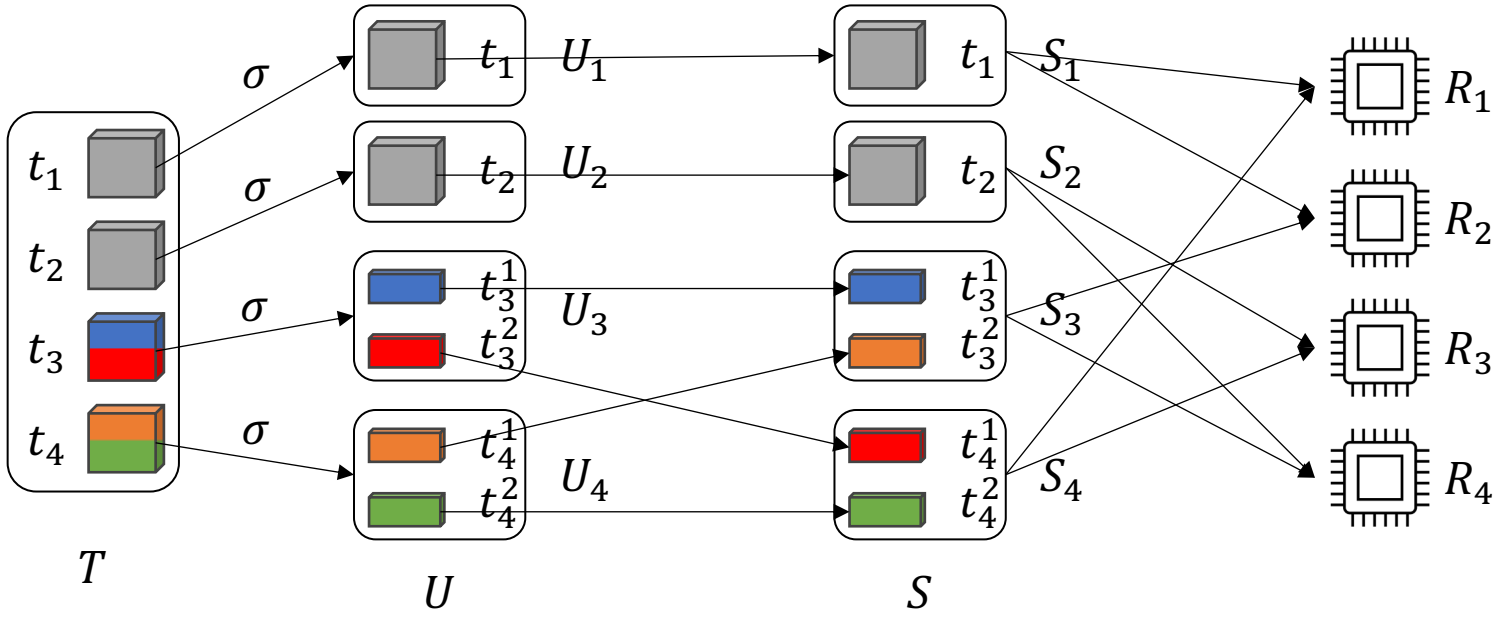


Tenplex – State Abstraction

- Parallelizable Tensor Collection (PTC)

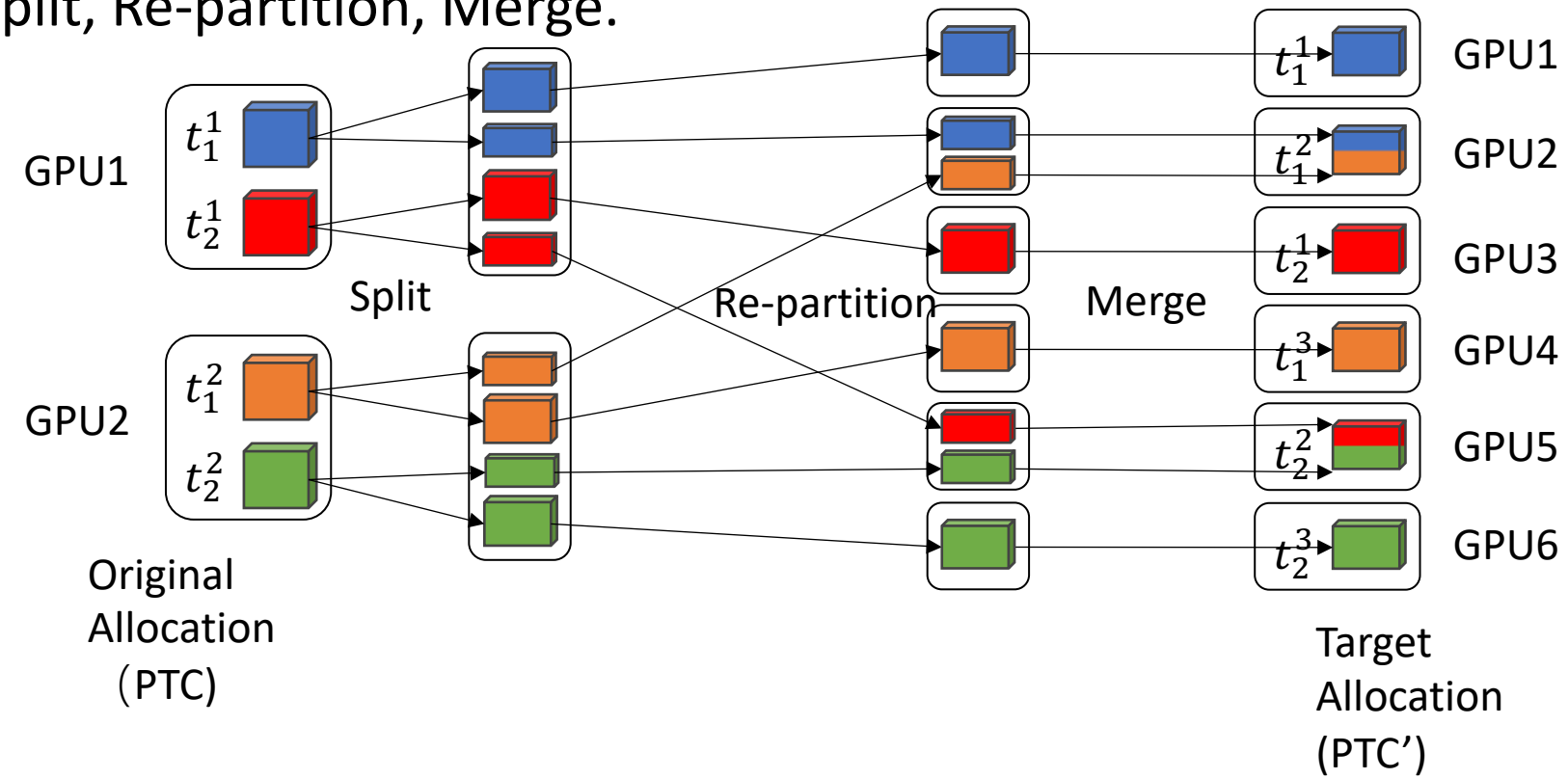
$$\text{PTC} = (T, \sigma, \phi, \alpha)$$

  - ϕ partitions U into sub-collections $S = \phi(U) = \{S_1, \ldots, S_p\}$
  - α allocates the sub-collections to GPUs. (A sub-collection can be allocated to multiple GPUs)



Tenplex – Reconfig Planning

- Reconfigure PTC $= (T, \sigma, \phi, \alpha)$ into PTC' $= (T, \sigma', \phi', \alpha')$
- Steps:
  - Split, Re-partition, Merge.



Tenplex workflow

- Tensor Store
  - Numpy to support most DL Systems
- State Transformer
  - 3. Reconfig Plan Generation
  - 4. Execute the plan
- Fault tolerance
  - If at least one DP replica survives, Tenplex retrieves updated states from GPU (instead of loading checkpoint and starting from previous step).

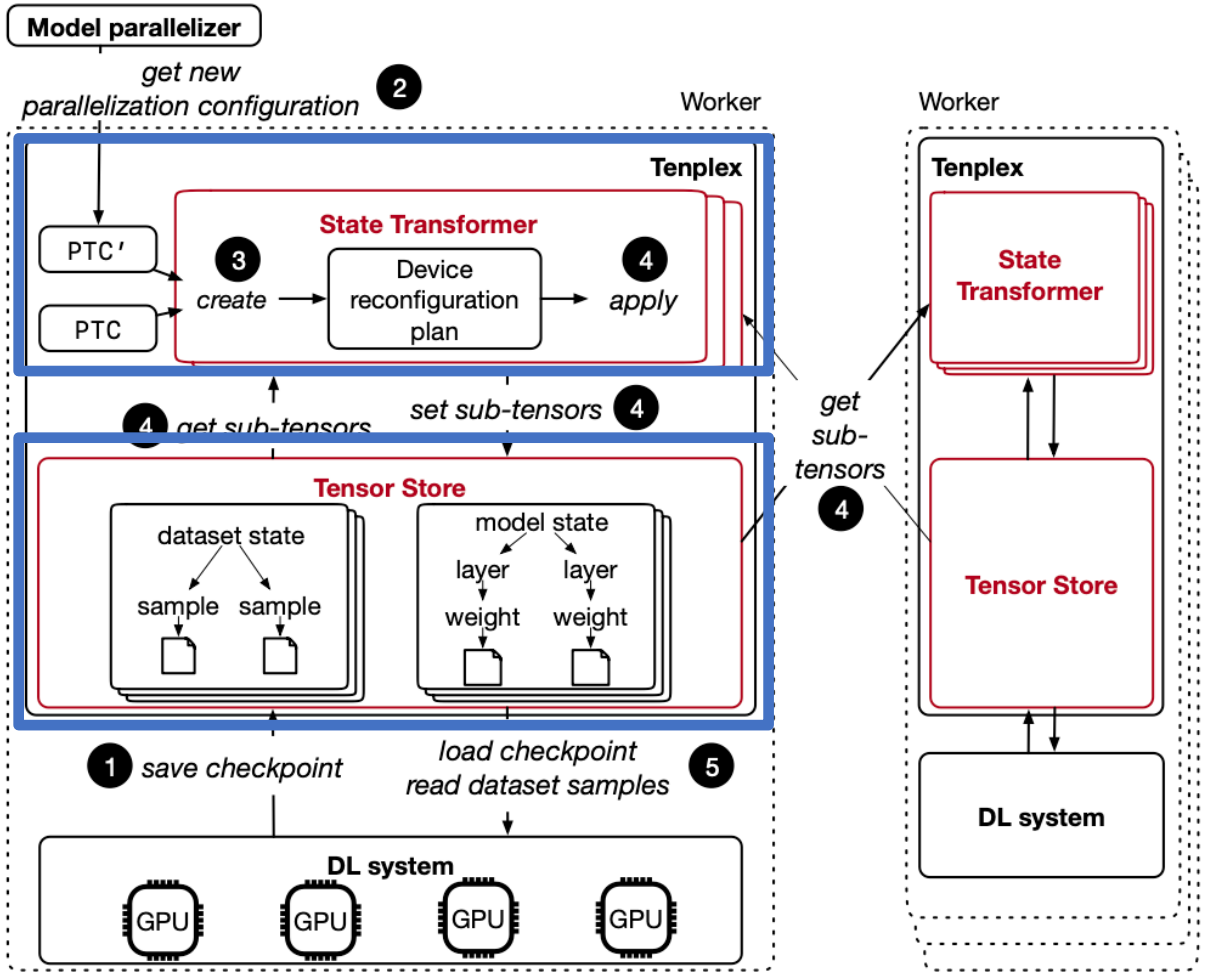
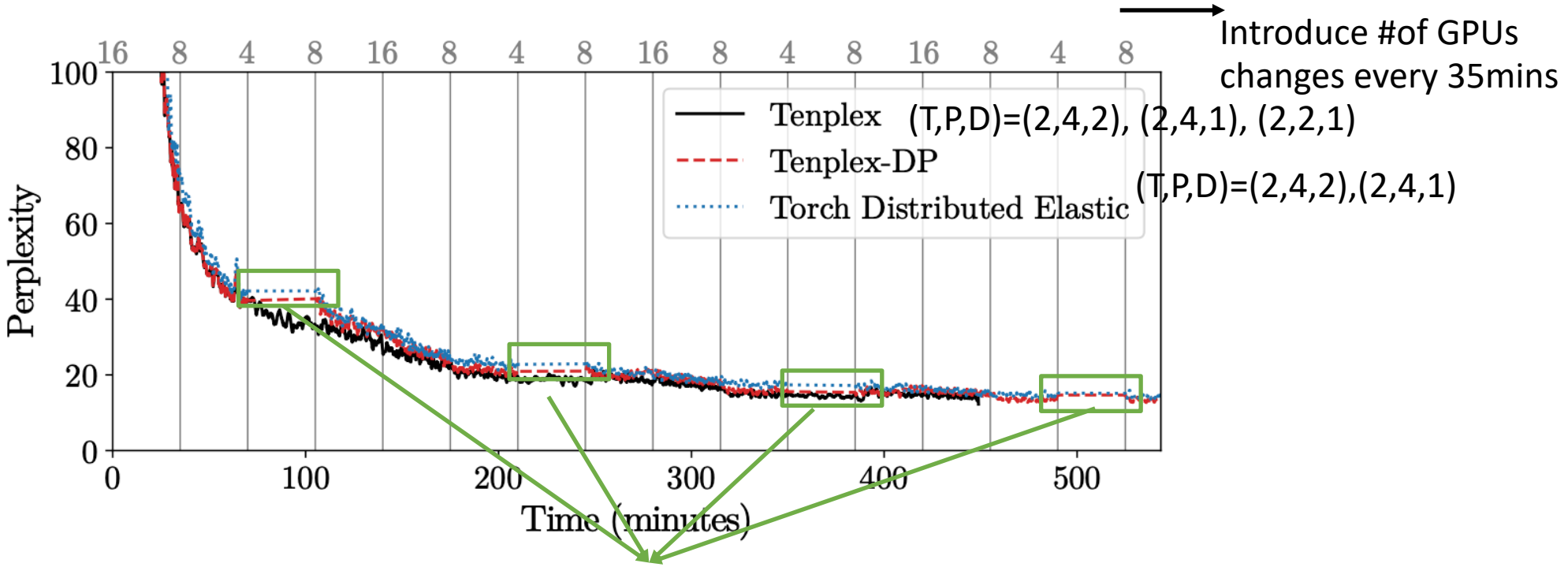


Fig. 8. TENPLEX architecture

Tenplex Evaluation - Speedup

- To run to the same step, Tenplex takes 298 mins, Tenplex-DP takes 576mins, Torch takes 548mins.
  - Torch only supports dynamic DP; Tenplex-DP only reconfigures DP.



Tenplex-DP and Torch paused. This is 35*4=140mins

Tenplex Evaluation - Others

- Redeployment time. E.g., migrating from one set of 8 GPUs to another set of 8 GPUs.
- Failure recovery time.
- Reconfiguration time.
- …

TENPLEX: Dynamic Parallelism for Deep Learning using Parallelizable Tensor Collections

Marcel Wagenländer
Imperial College London

Guo Li
Imperial College London

Bo Zhao
Aalto University

Luo Mai
University of Edinburgh

Peter Pietzuch
Imperial College London

SOSP 2024

Enabling Parallelism Hot Switching for Efficient Training of Large Language Models

Hao Ge*‡
gehao@stu.pku.edu.cn
Peking University

Fangcheng Fu*†‡
ccchengff@pku.edu.cn
Peking University

Haoyang Li‡
lihaoyang@stu.pku.edu.cn
Peking University

Xuanyu Wang‡
wxyz0001@pku.edu.cn
Peking University

Sheng Lin‡
linsh@stu.pku.edu.cn
Peking University

Yujie Wang‡
alfredwang@pku.edu.cn
Peking University

Xiaonan Nie‡
xiaonan.nie@pku.edu.cn
Peking University

Hailin Zhang‡
z.hl@pku.edu.cn
Peking University

Xupeng Miao
xupeng@purdue.edu
Purdue University

Bin Cui†‡§
bin.cui@pku.edu.cn
Peking University

HotSPa

- Tenplex is for handling resources changing.
- HotSPa actively find best parallelization strategy given different sequence lengths.
  - Their observation:

> In this paper, we first reveal the under-explored fact that the optimal parallelism strategy varies even for the sequences within a single mini-batch. Motivated by this, we present

Why Sequence Length Matters?

The maximum supported seq len of the model

- Longer sequence requires higher context length and more memory
- More memory requires higher TP degree (or other parallelism)
- Higher TP means lower DP degree, increasing latency due to communication.

Table 1. Running time (in seconds) of different tensor parallelism degrees (LLaMA2-7B, 8GPUs, *DP=8/TP*) when processing the same amount of tokens.

Seq Len	# Seqs	*TP=1*	*TP=2*	*TP=4*	*TP=8*
1K	512	**13.9**	14.7	16.2	19.5
2K	256	**14.2**	15.1	16.6	19.8
4K	128	**14.9**	15.8	17.4	20.6
8K	64	OOM	**17.4**	19.1	22.1
16K	32	OOM	OOM	**21.8**	25.0
32K	16	OOM	OOM	OOM	**30.8**

Why Sequence Length Matters?

- Distribution of sequence lengths is skewed.
- Thus, a randomly sampled mini-batch usually contains few long sequences with a lot of short sequences.

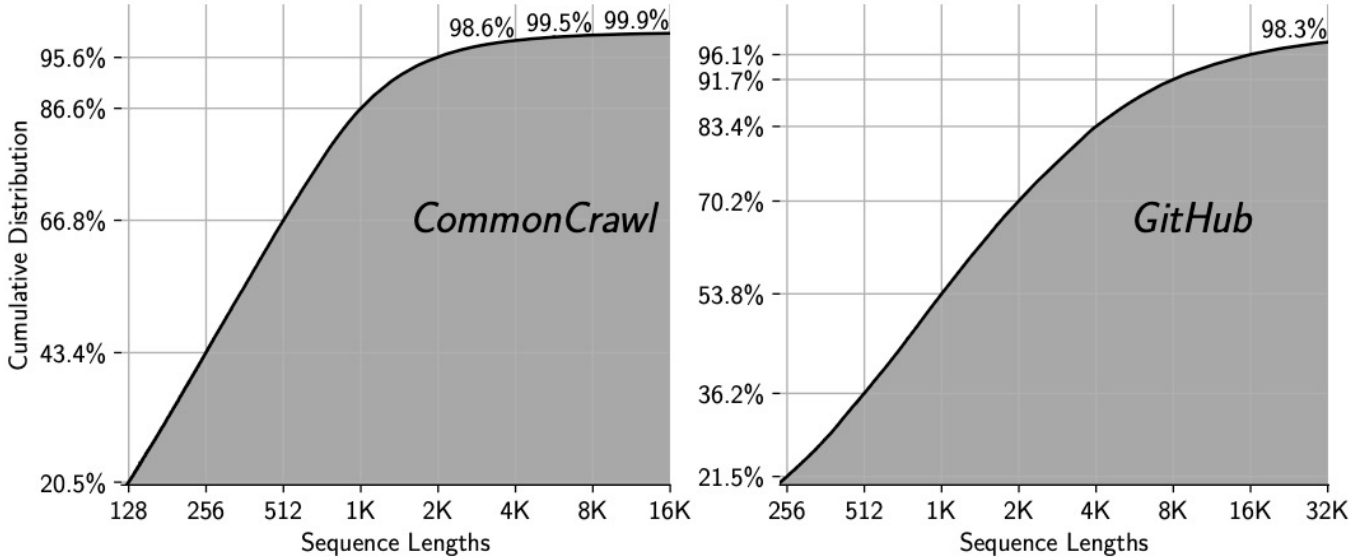
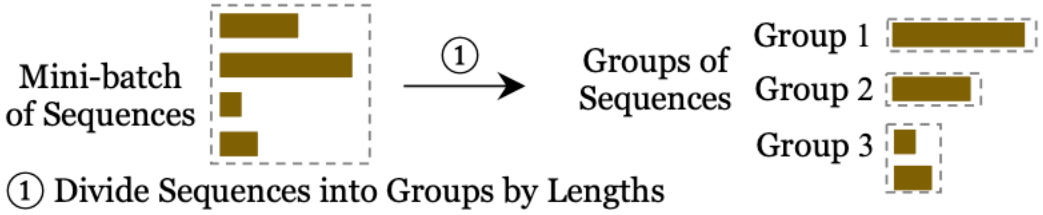


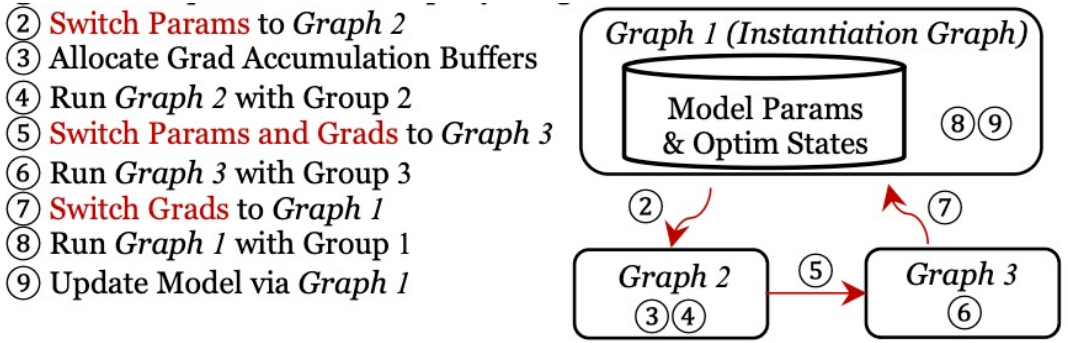
Figure 3. Cumulative distributions of sequence lengths.

How to solve this?

- Partition sequences into multiple groups,



- For each group, use different parallelism strategies.



Dynamic Training Parallelization (Revisit)

- Challenges or Missing parts
  - Auto re-partition
  - Full multi-dim parallelism
  - Low reconfiguration overhead

LogicGraph + ***DistConfig*** + ***Graph Compiler***

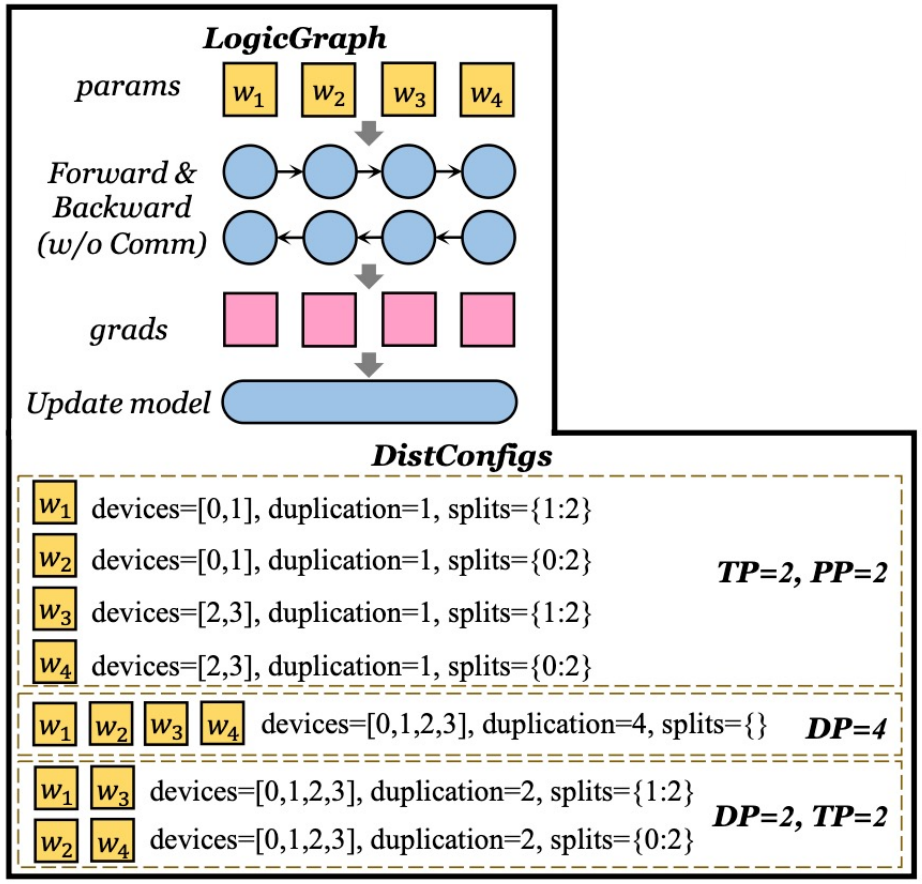
States Abstraction of model/data tensors and their transformation

Reconfig Planning: find low-cost reconfig plan to transfer or re-load tensors.

Hot Switch Planner

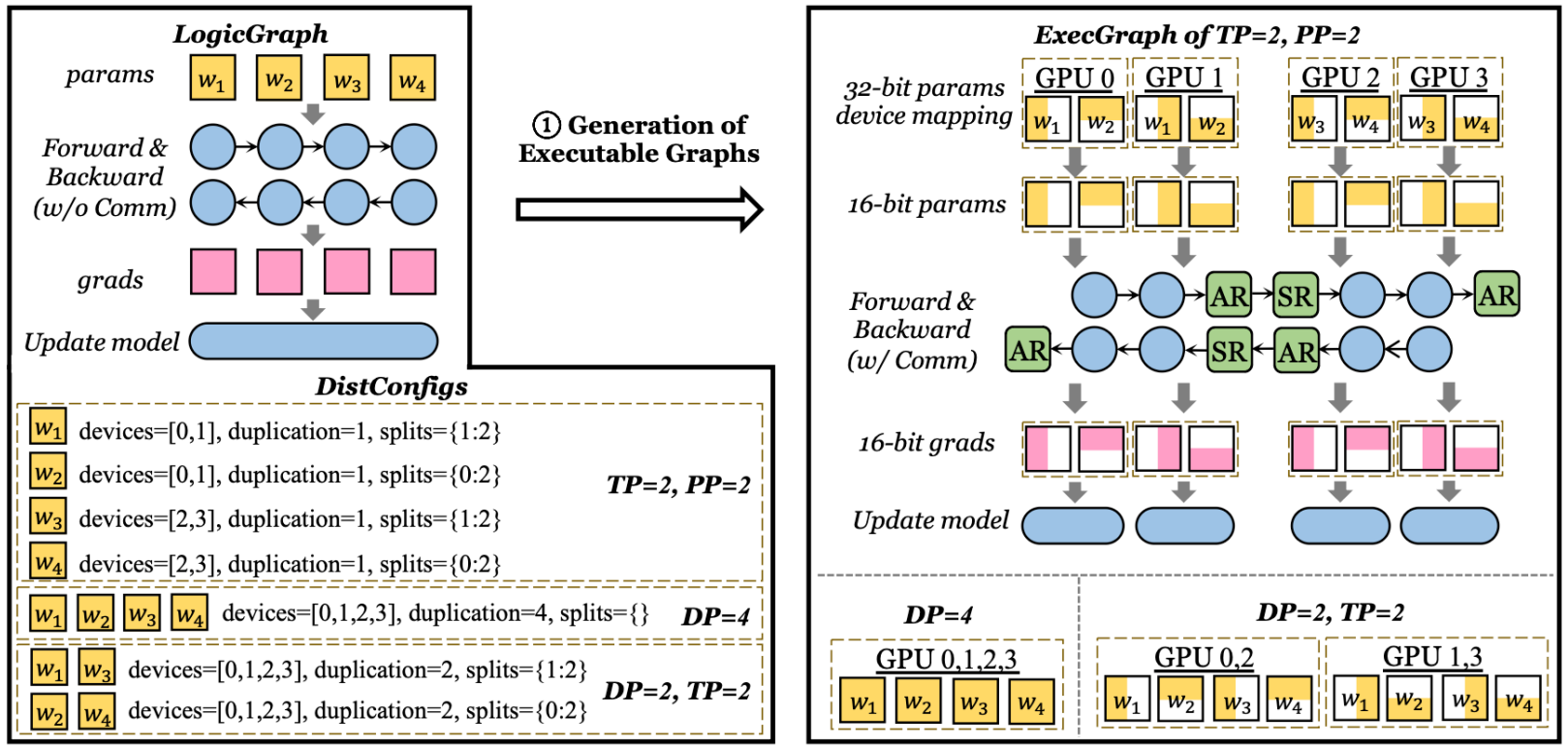
HotSPa – State Abstraction

- LogicGraph
  - No communication
- DistConfig
  - Annotate each tensor (manually)
    - Assigned devices
    - DP degree
    - How to Split ({dim: #of splits})
      - For TP…



HotSPa – State Abstraction

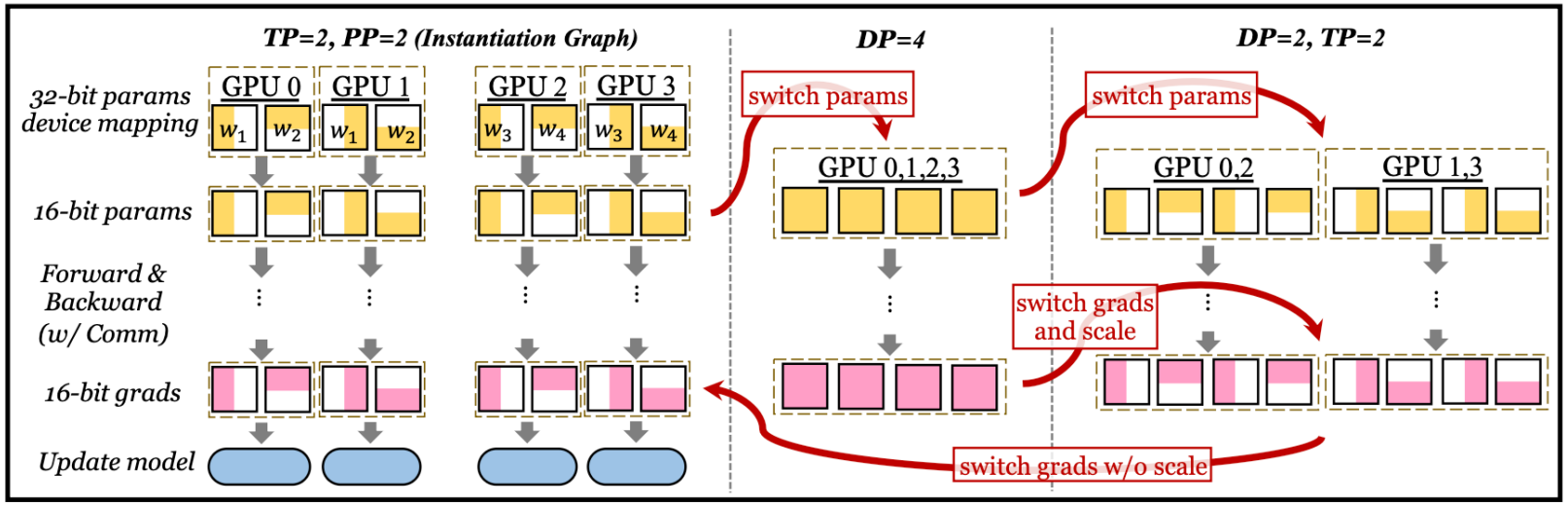
- Graph Compiler (LogicGraph + DistConfig → ExecGraph)
  - Insert type casting, communication, accumulation.



- Order?
- Transfer?

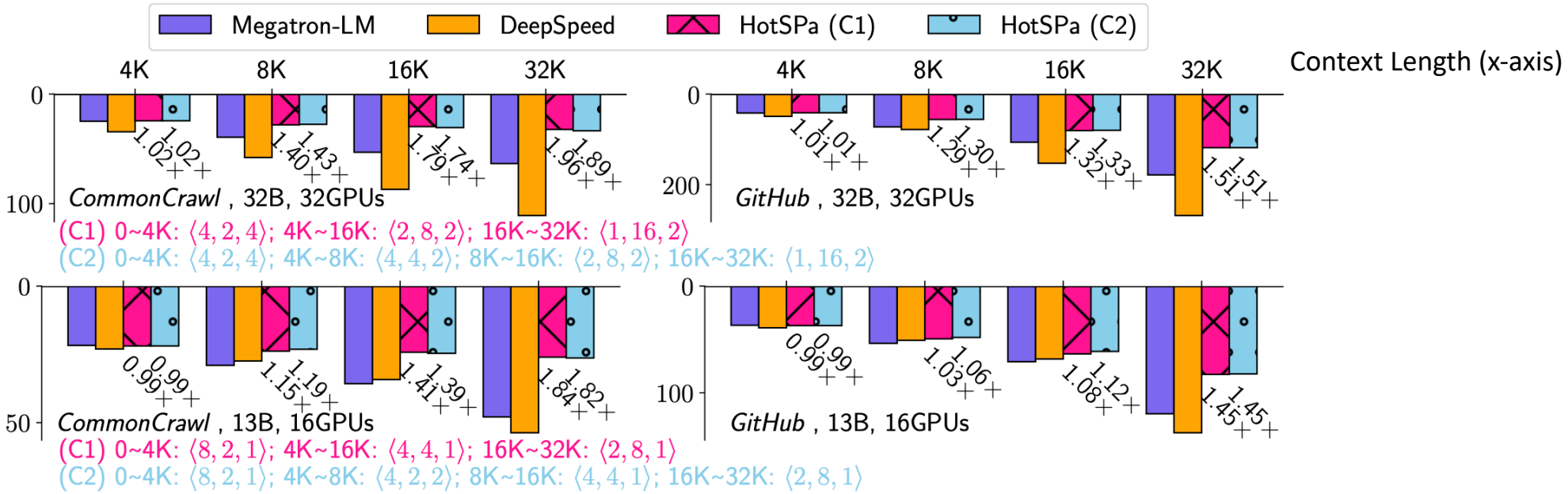
HotSPa – Reconfig Planning

- Instantiation Base Graph (for mixed-precision training)
  - Minimize memory occupation
- Order of ExecGraph
  - Minimize total switching cost
- Switching plan and its cost (Heuristic Greedy Algorithm)
  - Metrics: Total volume of inter/intra-node transferring.
  - Prefer intra-node communication and minimize the maximum sending volume of all devices.



HotSPa Evaluation

- Running time of Megatron-LM and DeepSpeed increases as context length increases;
- Running time of HotSPa increases much slower.



Part of evaluation results.

Conclusion

- Different motivation
  - Tenplex: resource changes
  - HotSPa: sequences can be re-grouped to use better parallelization
- Similar state abstraction
  - Tenplex also records dataset read position
  - HotSPa maintains LogicGraph for GraphCompiler to compile into ExecGraph
- Reconfig planning
  - Tenplex is more passive
    - Just moves tensors from source to destination.
  - HotSPa is more active
    - Finds a set of DistConfigs in advance and orchestrates them in a good order.
    - Also uses some heuristics.
- Results
  - Tenplex allows resource changes
  - HotSPa optimizes specifically for the skewed sequence lengths.