DOPPLER: Dual-Policy Learning For Device Assignment In Asynchronous Dataflow Graphs

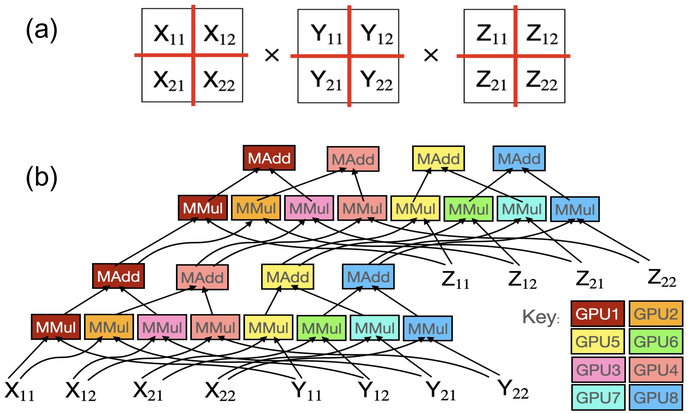
Xinyu Yao, Daniel Bourgeois, Abhinav Jain, Yuxin Tang, Jiawen Yao, Zhimin Ding, Arlei Silva, Christopher Jermaine

Motivation

- Deep learning systems like PyTorch execute computations in a level-wise, lock-step synchronous manner, whereas an alternative is fully asynchronous (work-conserving) scheduling:

Model	Work-conserving	Synchronous
ChainMM	139 (ms)	185.3 (ms)
FFNN	50.2 (ms)	76.9 (ms)

- Consider the matrix multiplication chain X x Y x Z decomposed to run on eight GPUs by (a) sharding each matrix four ways, along with (b) a fine-grained data graph with potential assignment:

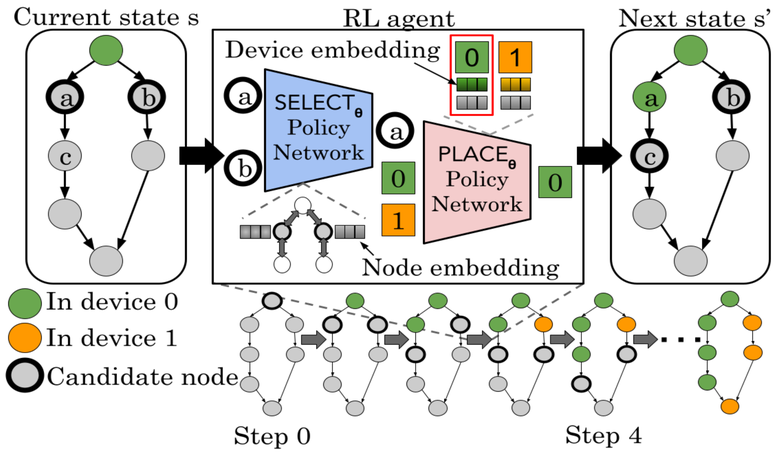


Contributions

- Investigate the device assignment problem in a multi-GPU work-conserving system.
- Introduce dual-policy learning to **first learn the approximated traversal order of nodes** before assigning to device.
- Propose ***DOPPLER***, a three-stage training framework that improves scheduling on the fly via continual learning during deployment, along with two pretraining stages.
- Our experiments show that DOPPLER achieves up to 52.7% lower execution times compared to the best baseline.

DOPPLER Framework

- **OUR GOAL**: Given a ***computation graph*** G = ⟨V, E⟩ with nodes V={v_1, v_2,..., v_n}, edges E = {e_1, e_2,..., e_m}, and devices D, we aim to generate a ***device assignment*** A = {A_{v1} , A_{v2} , ..., A_{vn} } where each $A_v \in A$ takes some value in D, such that ***execution time*** *within a work-conserving system is minimized.*

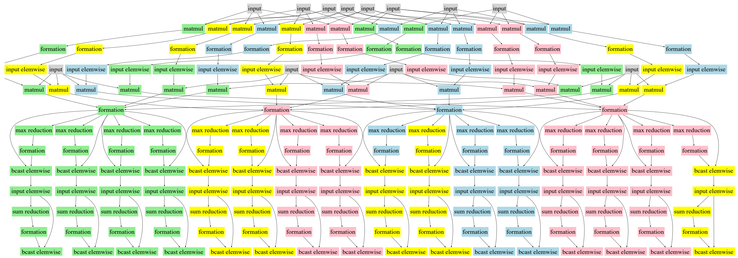


Experiments & Results

- We study the following questions, focusing on the quality of assignments produced by DOPPLER (Q1–Q5) and the scalability of DOPPLER's dual-policy networks (Q6). See Q2,Q3, Q5,Q6 in paper.
- (Q1) Comparison between our solutions and existing alternatives in milliseconds:

	4 GPUs						Runtime Reduction	
Model	Crit. Path	Placeto	GDP	EnumOpt.	Doppler-sim	Doppler-sys	Baseline	EnumOpt.
ChainMM	230.4 ± 4.3	137.1 ± 2.2	198.0 ± 3.3	139.0 ± 10.0	**122.5** ± 4.0	123.4 ± 2.5	10.7%	11.9%
FFNN	217.8 ± 11.3	126.3 ± 5.8	100.3 ± 3.2	50.2 ± 2.5	49.9 ± 1.1	**47.4** ± 0.7	52.7%	5.6%
Llama-block	230.9 ± 8.7	411.5 ± 19.7	336.5 ± 8.4	172.7 ± 5.0	191.5 ± 6.0	**160.3** ± 4.3	30.6%	7.2%
Llama-layer	292.6 ± 5.8	295.1 ± 7.0	231.5 ± 5.1	174.8 ± 4.7	167.0 ± 3.4	**150.6** ± 4.2	48.5%	13.8%

- (Q4) ***DOPPLER***'s Assignment on Feed-Forward Neural Network:



Key Takeaway

- ***DOPPLER*** explicitly tries to learn a latent node ordering, to make the assignment problem easier.

Corresponding Author: Xinyu Yao | Email: xy38@rice.edu | Github Link: https://github.com/xinyuyao/Doppler