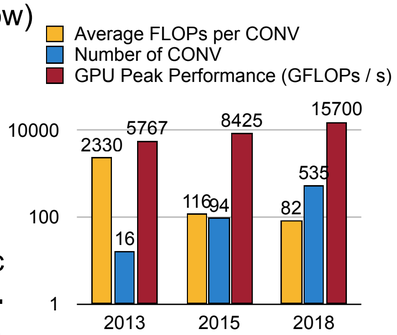
IOS: Inter-Operator Scheduler for CNN Acceleration

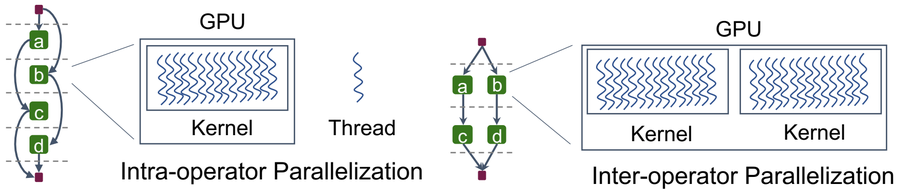
Yaoyao Ding[1] Ligeng Zhu[2] Zhihao Jia[3] Gennady Pekhimenko[1] Song Han[2]
University of Toronto, Massachusetts Institute of Technology, Carnegie Mellon University

We provide scripts to reproduce results in every figure and table!

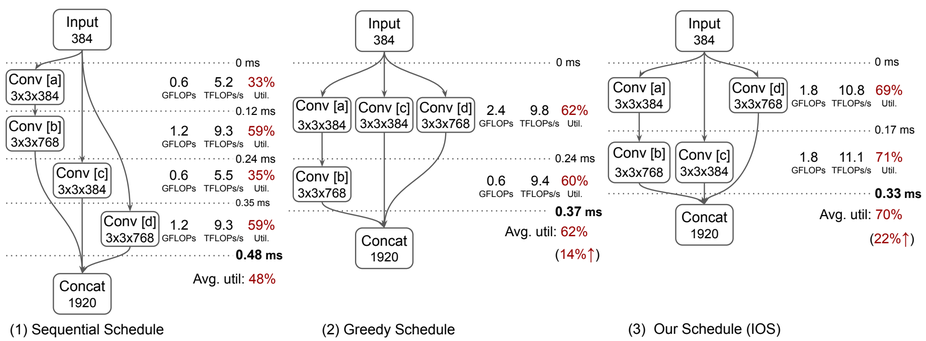
Overview

- Existing frameworks (PyTorch, TensorFlow) focus on **intra-operator parallelization**.
- Only utilizing intra-operator parallelism suffers from device under-utilization problem, especially for small op & power GPU.
- Therefore, we propose IOS — a dynamic programming algorithm scheduling **inter-operator parallelization** of CNN models.





Explore More Schedules is Important



(1) Sequential Schedule (2) Greedy Schedule (3) Our Schedule (IOS)

(1) **Sequential Schedule**: the default choice for most frameworks, but leads to **insufficient utilization** as only one operator at a stage.

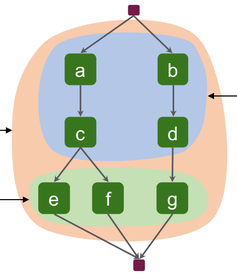
(2) **Wavefront Schedule**: a greedy method that execute all available operators stage by stage. It is sub-optimal due to **unbalanced** schedule.

(3) **IOS Schedule (ours)**: explores schedule space **exhaustively**, balances the computation in each stage, and best utilizes the hardware.

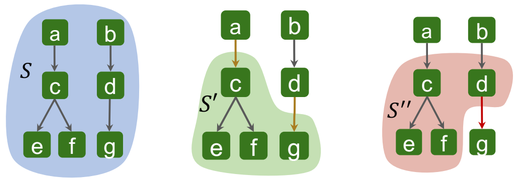
Inter-Operator Scheduler

$$Latency[S] = \min_{S' \text{ can be a last stage of } S} \left(Latency[S - S'] + StageLatency(S') \right)$$

- $Latency[S]$: Best schedule's latency of S
- S is the ops to be scheduled
- S' is a candidate for last stage of S
- $S - S'$ are ops remaining to be scheduled in a sub-problem
- $StageLatency(S')$: Latency of stage S'



The time complexity of the dynamic programming is: $O((\frac{n}{d}+1)^{2d})$

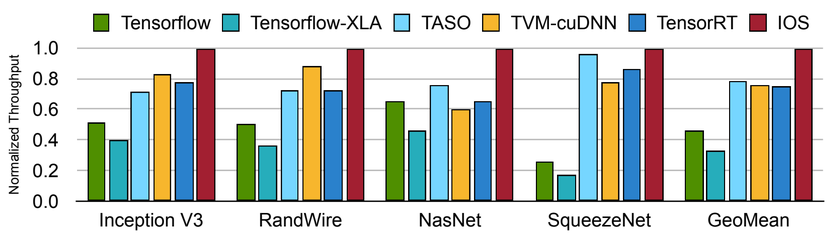


n : number of operators
d : maximum number of concurrent operators

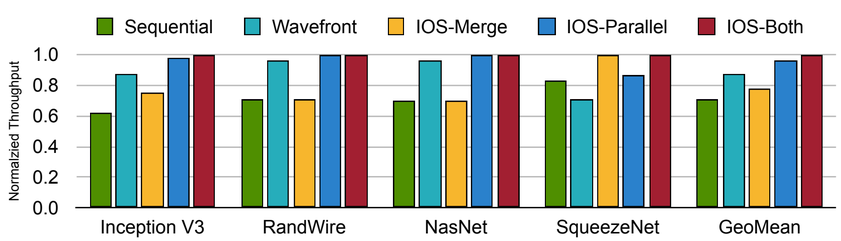
S: ops to be scheduled S': Last stage candidate S'': **NOT** Last stage candidate

S' can be a last stage of S $\Leftrightarrow$ There is no edge from S' to $S - S'$

IOS Accelerates Inference

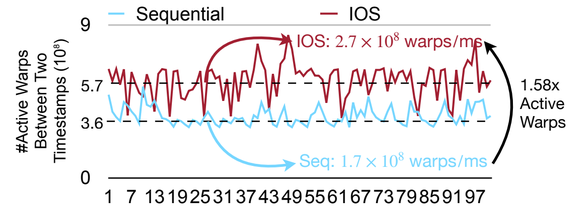


Comparison of cuDNN-based Frameworks



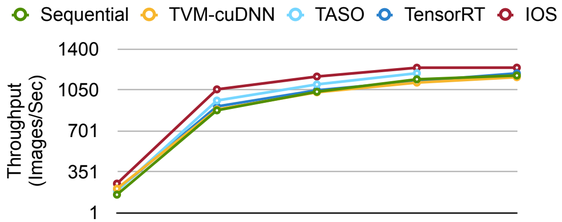
Comparison of Different Schedules

More Active Warps



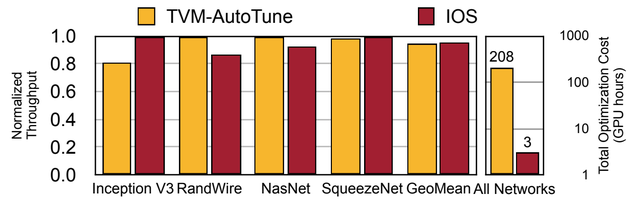
IOS Schedule has More Active Warps per ms

Large Batch Size



Consistent Improvement for Larger Batch Sizes (Inception V3 is used as benchmark)

IOS v.s. AutoTVM



AutoTVM and IOS are ***orthogonal*** and can be combined to further boost the performance

Schedule Specialization

Specialization for Different Batch Sizes		Optimized for 1	Optimized for 32	Optimized for 128
Execute on	1	**4.03**	4.50	4.63
	32	29.21	**27.44**	27.93
	128	105.98	103.74	**103.29**

Specialization for Batch Sizes

Specialization for Different Devices		Optimized for K80	Optimized for V100
Execute on	K80	**13.87**	14.65
	V100	4.49	**4.03**

Specialization for Devices

Specialized Schedules achieves the best performance