fiJIT Fast GPU Primitives for Multiplexed Deep Learning Inference

riselab UC Berkeley

Paras Jain, Simon Mo, Ajay Jain, Alexey Tumanov, Joseph Gonzalez, Ion Stoica

Background/Context:

- ML inference increasingly important for soft real-time latency-sensitive applications

Problem:

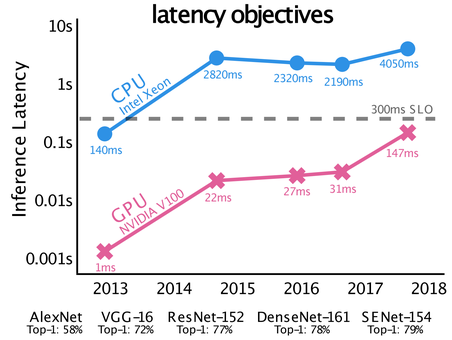
- Model complexity increases —> latency increases; too slow for soft-real time applications
- GPU —> more attractive for inference, suffers low utilization (under latency constraints)

Solution:

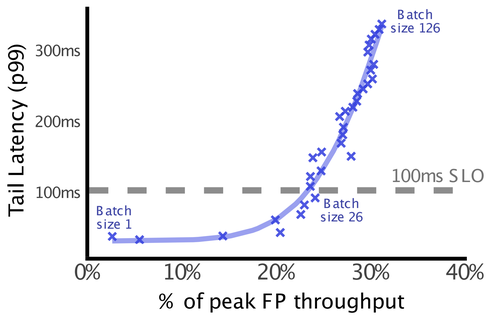
- *State of the art*: temporal multiplexing
- *Proposal*: multiplex GPUs across multiple models and time
- *Approaches*: kernel coalescing, kernel constraining
- *Goal*: VLIW inspired JIT compiler that enables serverless abstractions for GPU

Motivation: Online inference leads to low GPUs utilization

As models continue to grow, GPUs are the only way to meet online inference latency objectives

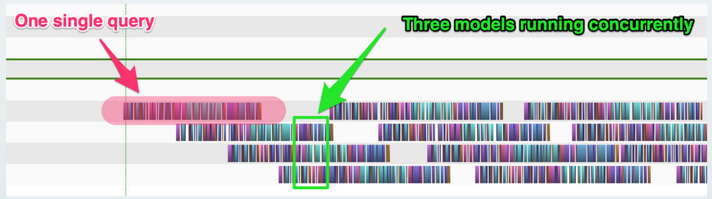


GPUs are under-utilized in online inference due to small batch-sizes

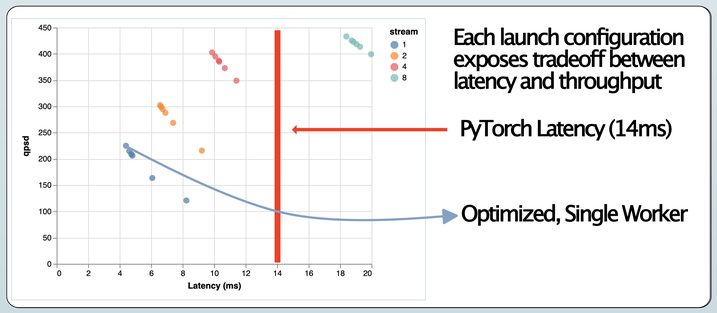


Key requirements for GPU kernel multiplexing

- Runtime performance must be **predictable**
- Multiplexing should increase **resource-efficiency**



- The runtime should be **fair** to all its tenants:
  - Equal treatment: same SLO guarantees
  - Equal outcome: strict inter-tenant isolation
- Enable throughput vs latency tradeoff

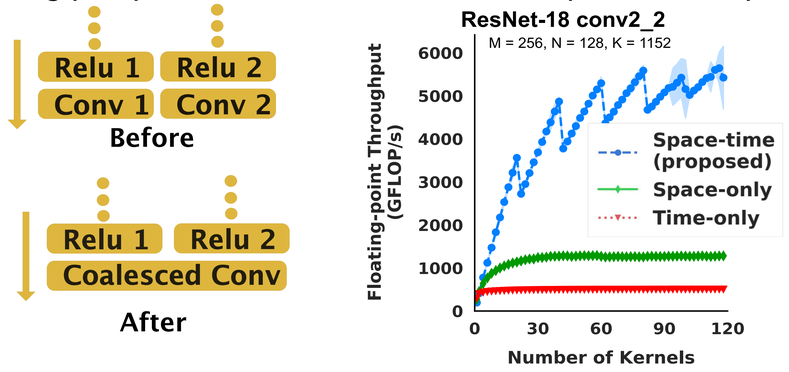


Time-only multiplexing: poor resource-efficiency

- **Time-multiplexing**: on device scheduler enables interleaved execution of multiple CUDA contexts (no parallel execution)
- **Pro**: Guaranteed isolation between tenants and predictability
- **Con**: Sharply degraded throughput and increased latencies

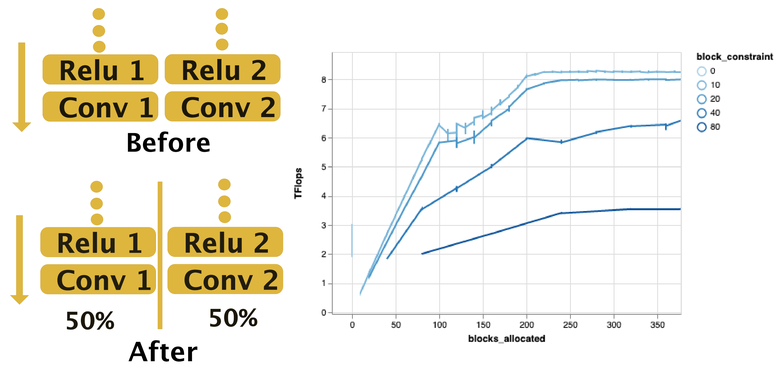
Space-Time multiplexing: coalesce similar kernels

7x throughput improvement when coalesce ResNet kernels compared to time multiplex

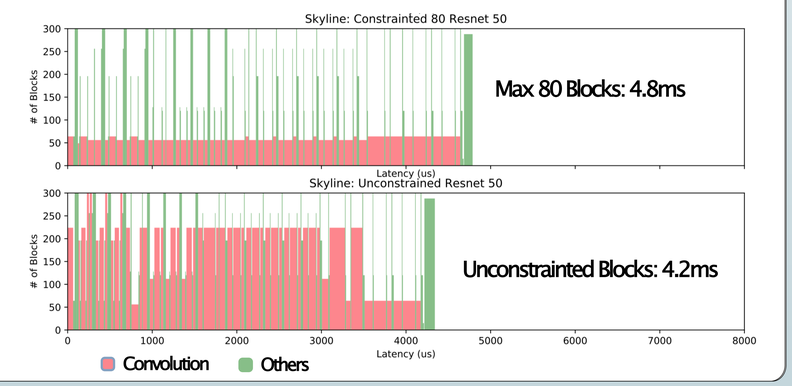


GPU Partitioning: constraint each kernel to # of blocks

Constrained kernels achieves 2.6x throughput when dispatched together



Using less number of GPU blocks incurs little latency overhead



Vision: VLIW JIT Compiler -> Serverless GPU