LithOS: An Operating System for Efficient Machine Learning on GPUs

Patrick H. Coppock, Brian Zhang, Eliot H. Solomon,
Vasilis Kypriotis, Leon Yang*, Bikash Sharma*, Dan Schatzberg*,
Todd C. Mowry, and Dimitrios Skarlatos

SOSP 2025

Carnegie Mellon University
Computer Science Department

* Meta

GPU Utilization Is Low Everywhere!

Meta

Device: **30%**
SM: **15%**
Train: **40%**

(more details in our paper)

Microsoft 52%

(Philly, ATC 2019)

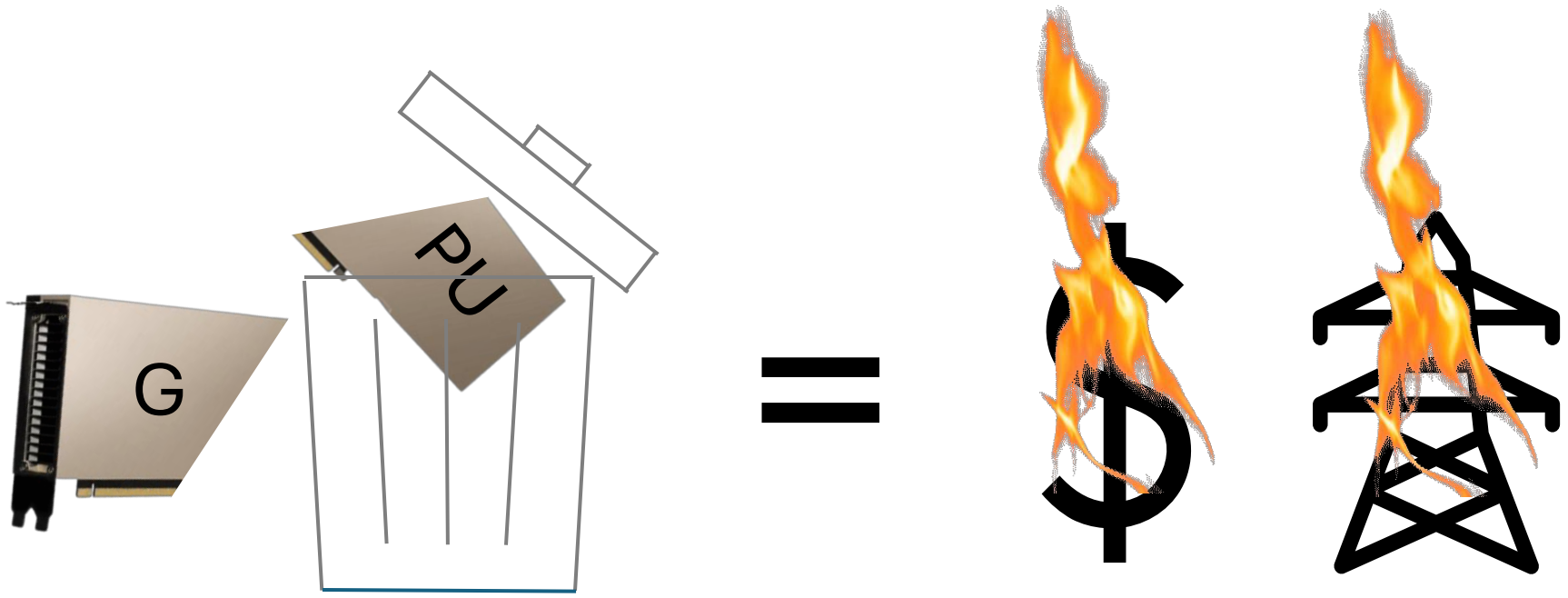
Alibaba 10%

(AntMan, OSDI 2021)

ByteDance 26%

(MuxFlow, arXiv)

Low GPU Utilization Is Costly



Throwing GPUs away wastes money and power!

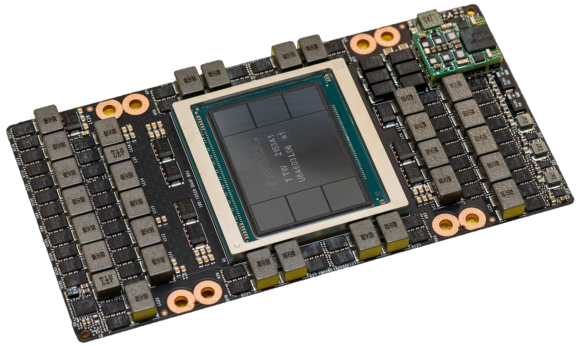
What Is the Role of the OS in the GPU Era?

Mainframes

1950s

- Expensive
- Scarce
- Single-Tenant

GPUs



2020s

Operating systems were invented to solve these problems!

LithOS: An Operating System for GPUs

Fully transparent to ML stack

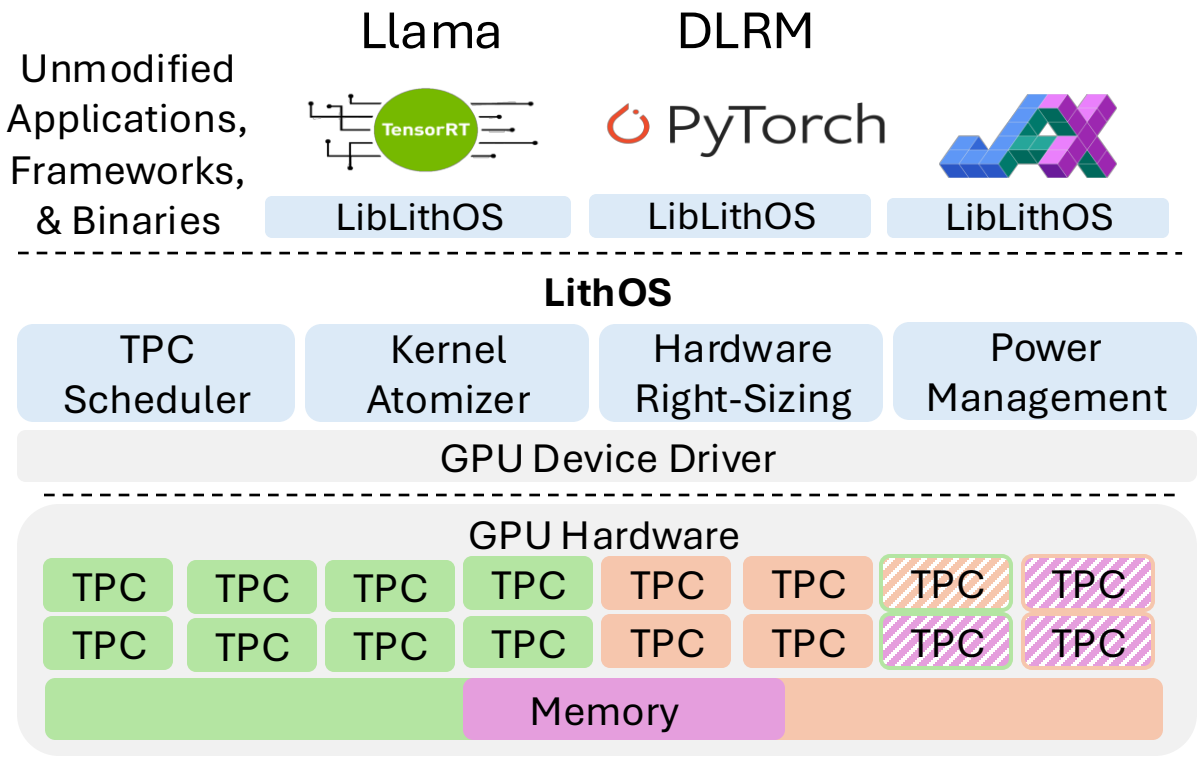
- No PTX, framework, compiler changes
- TPC Scheduling + Stealing
- Kernel Atomizer
- Right-size kernels to TPCs
- Transparent DVFS

Better performance

- **13x** lower tail latencies vs. NVIDIA
- **3x** lower tail latencies vs. SotA$_1$
- **1.6x** higher throughput vs. SotA$_2$

Easy resource saving

- Right-Sizing: **25%** capacity savings
- DVFS: **25%** energy savings

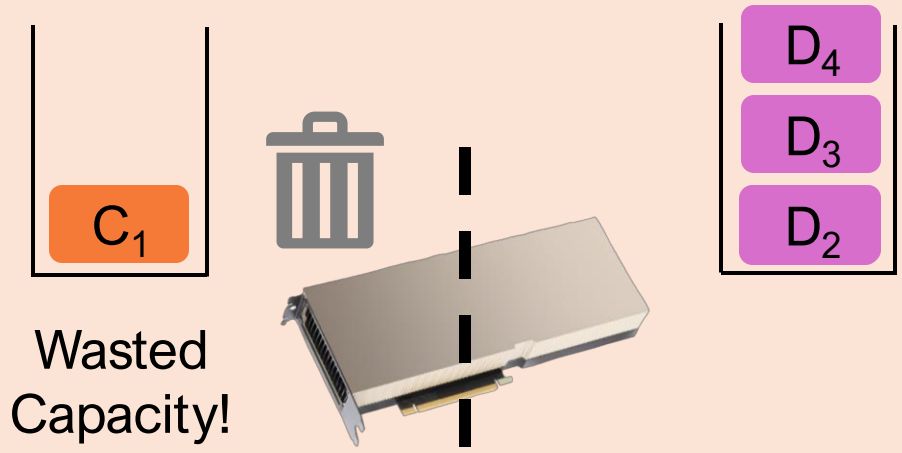


SotA GPU Sharing Methods Aren't Effective

Multi-Process Service (MPS)

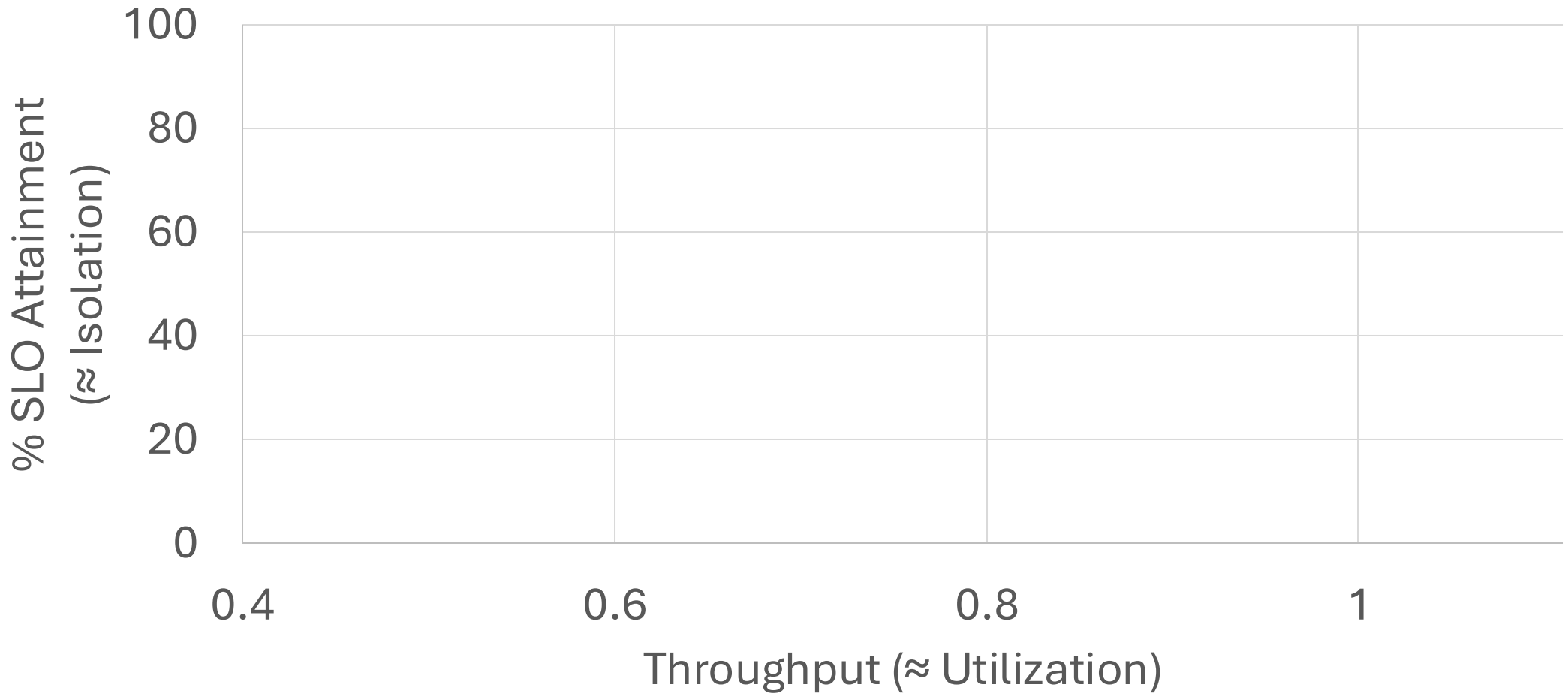
Time-sharing, no preemption → performance interference

Multi-Instance GPU (MIG)

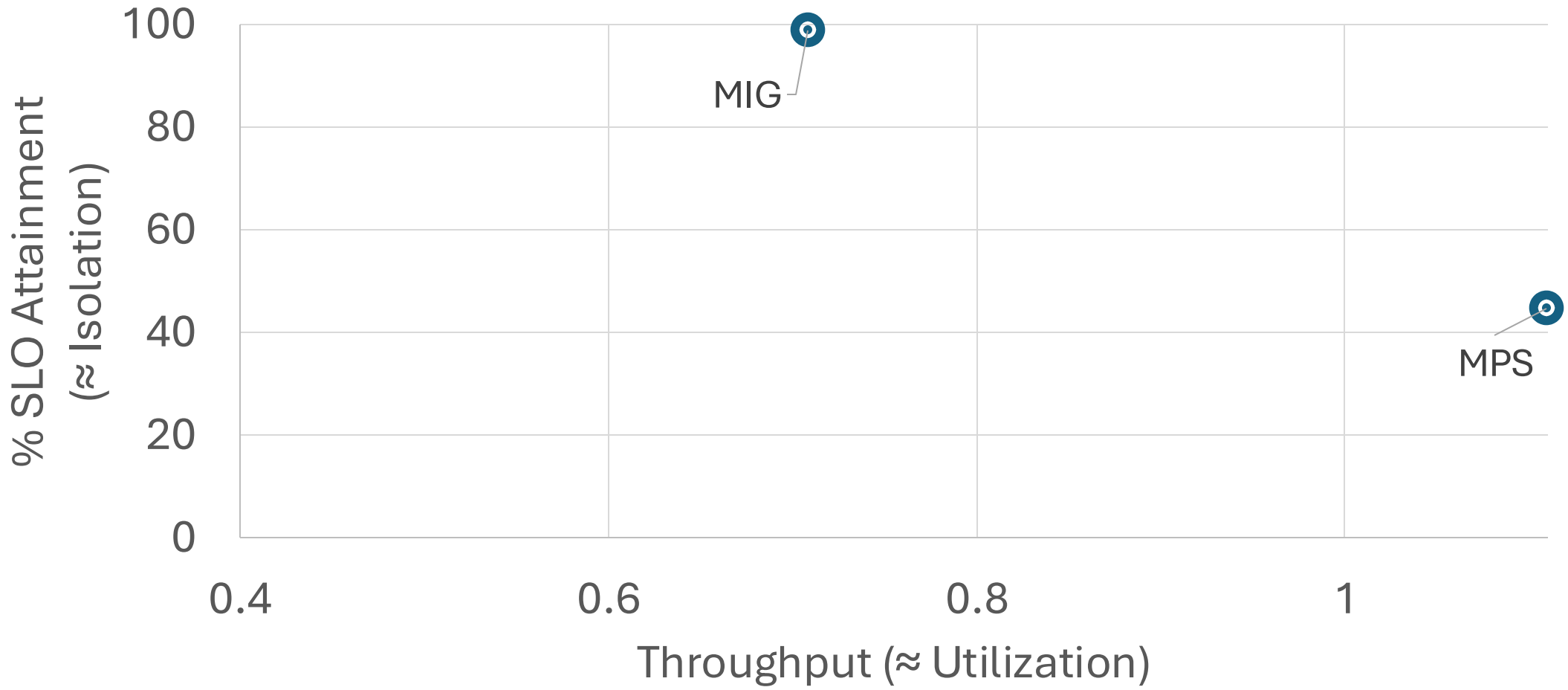


Coarse-grained, fixed partitions → poor utilization

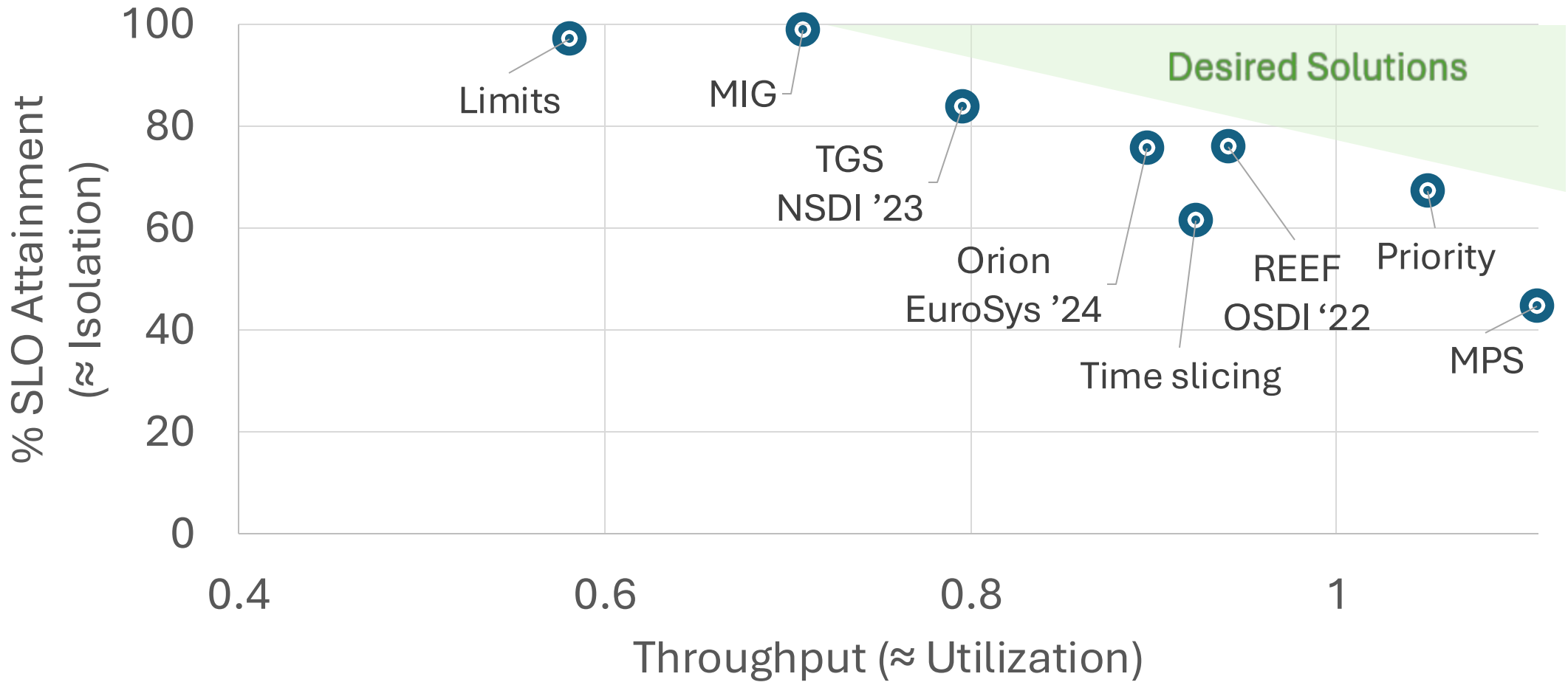
The Latency SLO vs. Throughput Trade-off



The Latency SLO vs. Throughput Trade-off



The Latency SLO vs. Throughput Trade-off



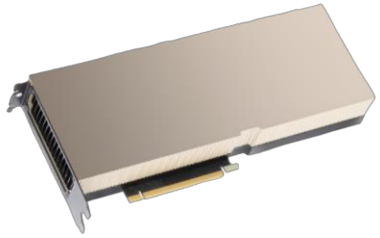
~~GPUs Can't Be Shared Efficiently~~

CPU

Sharing Happens Transparently

Threads Scheduled to Specific Cores

Preemption



GPU

Invasive Modifications Required

No Control of Where Kernels Execute

No Preemption

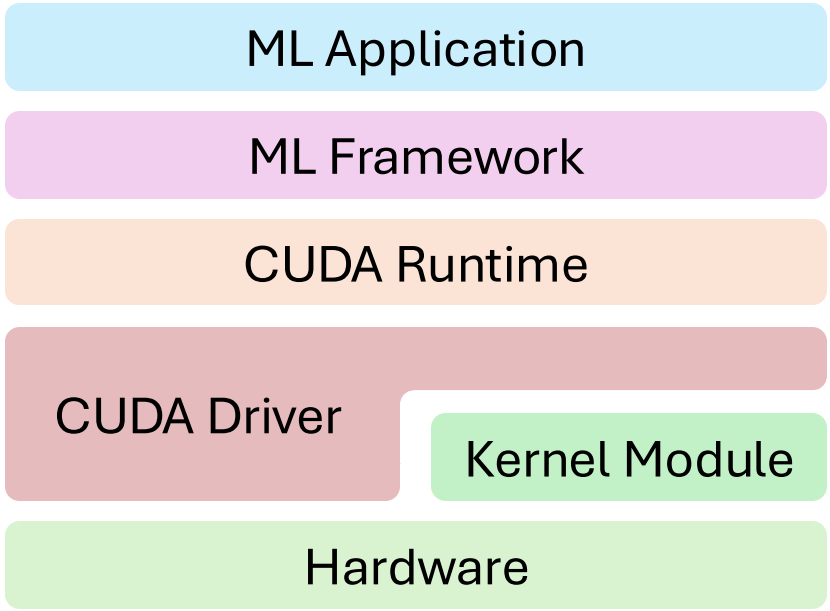
GPU + LithOS

Interposition + Reverse Engineering

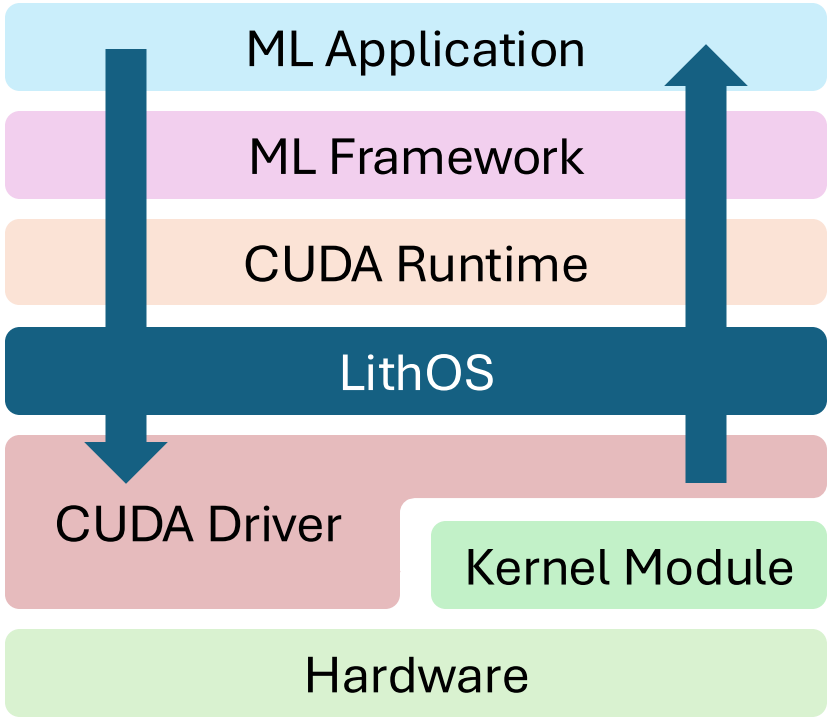
TPC Scheduling

Kernel Atomization

LithOS Interposition + Reverse Engineering



LithOS Interposition + Reverse Engineering



TPC Masking

Prelude Kernels

Advantages

- No need to modify application software (CUDA C++, PTX, etc.)
- Pulls kernel scheduling logic out of closed-source software and hardware
- Easy to port across hardware and driver versions

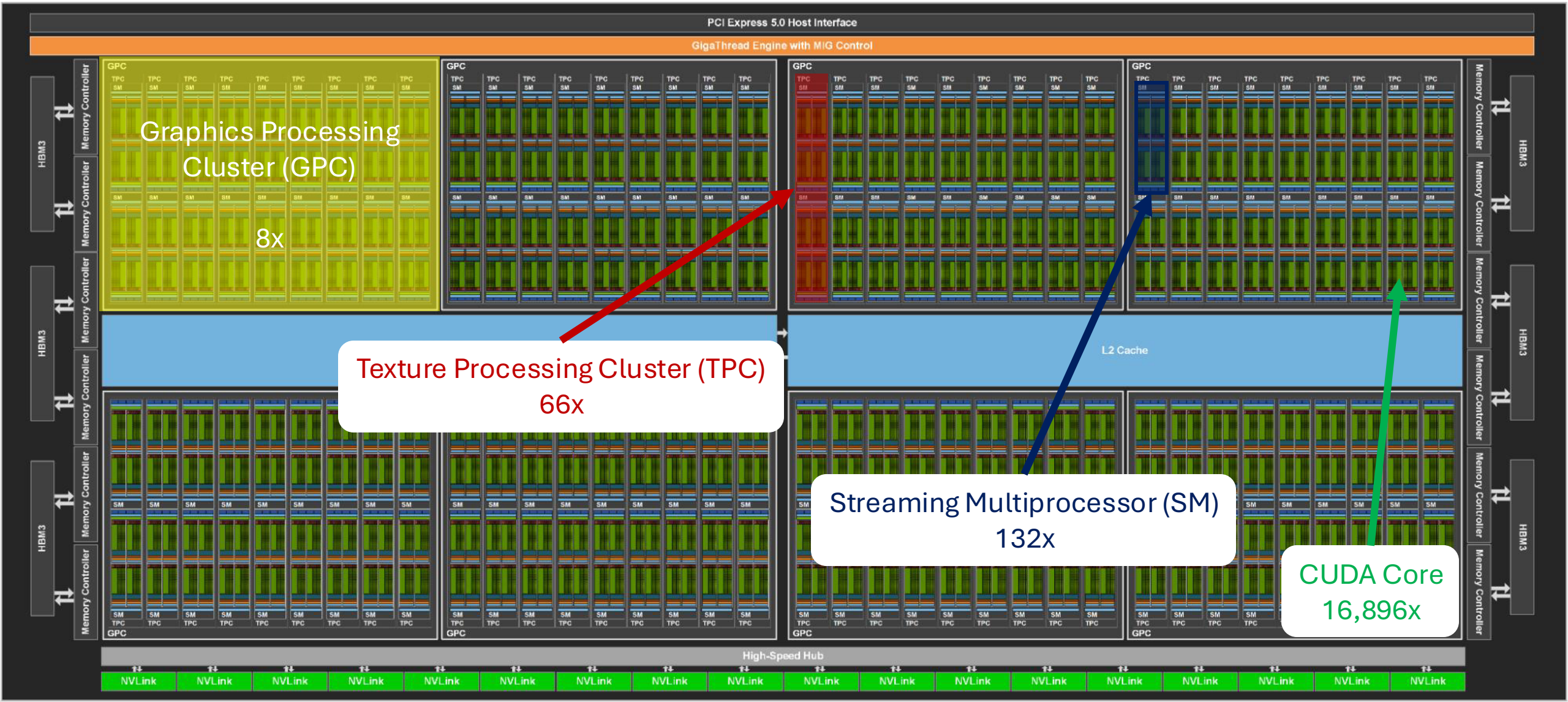
Vision

- Software from LithOS down is part of an integrated, open OS

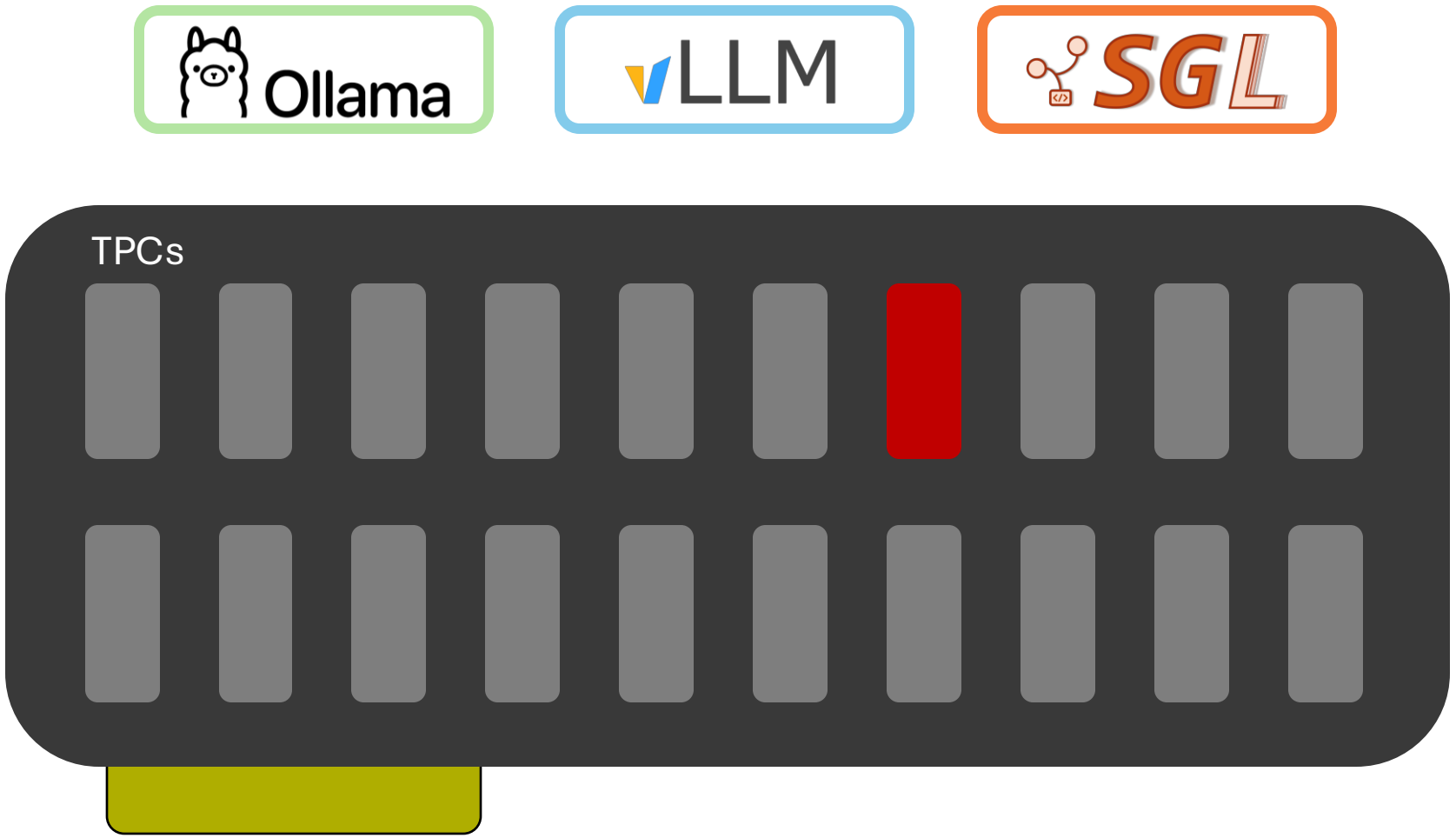
GPU Organization 101

MIG partitions on GPC boundaries

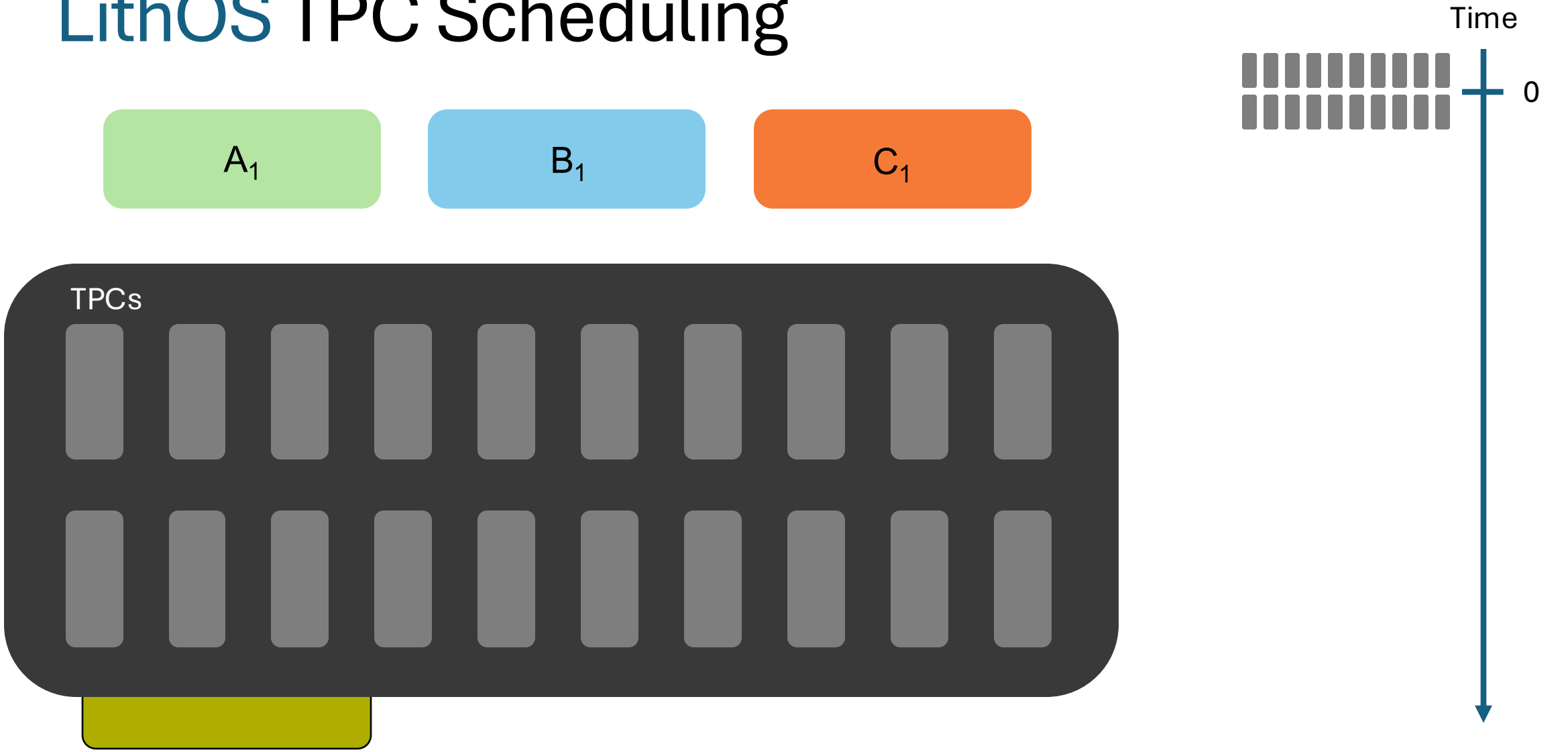
LithOS partitions on TPC boundaries



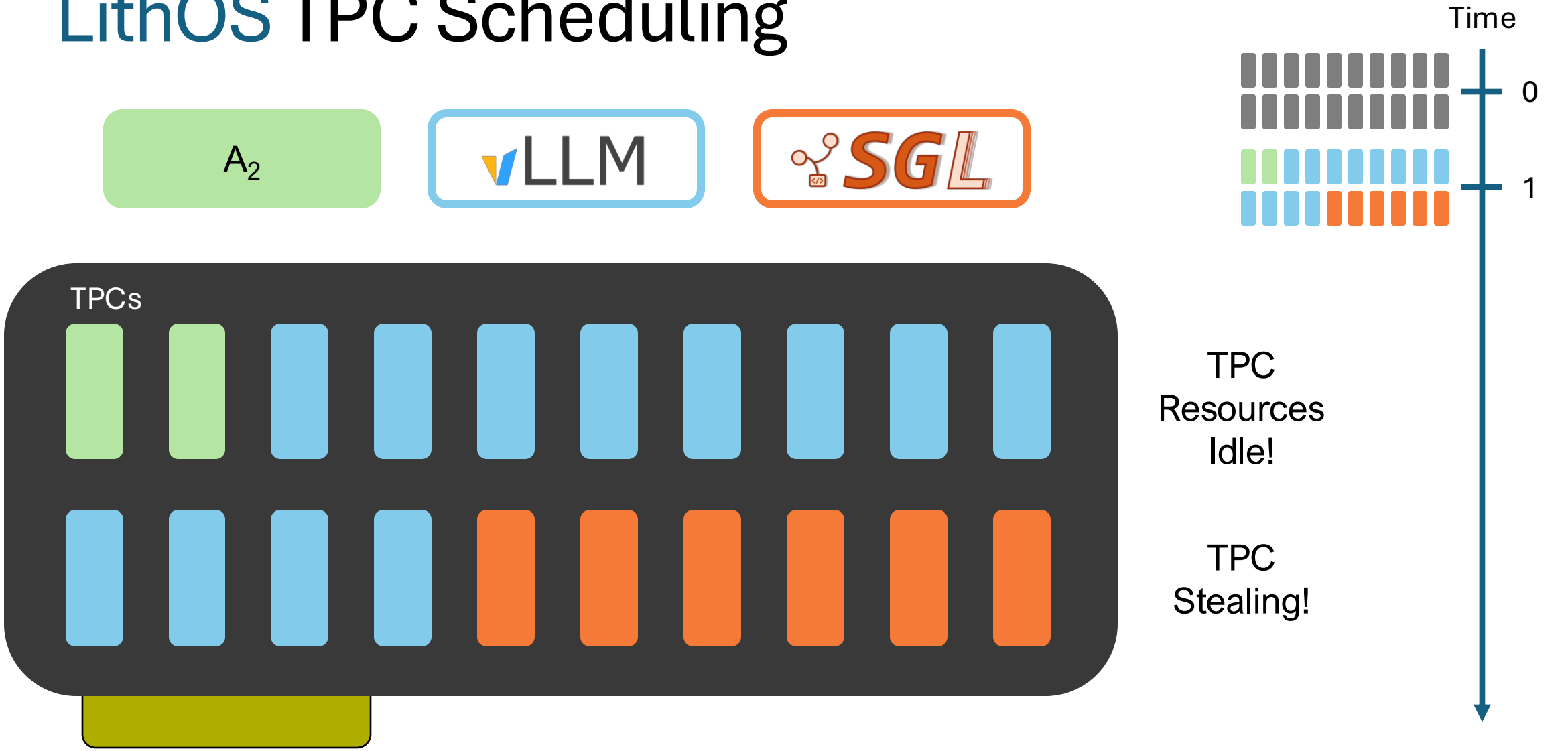
LithOS TPC Scheduling



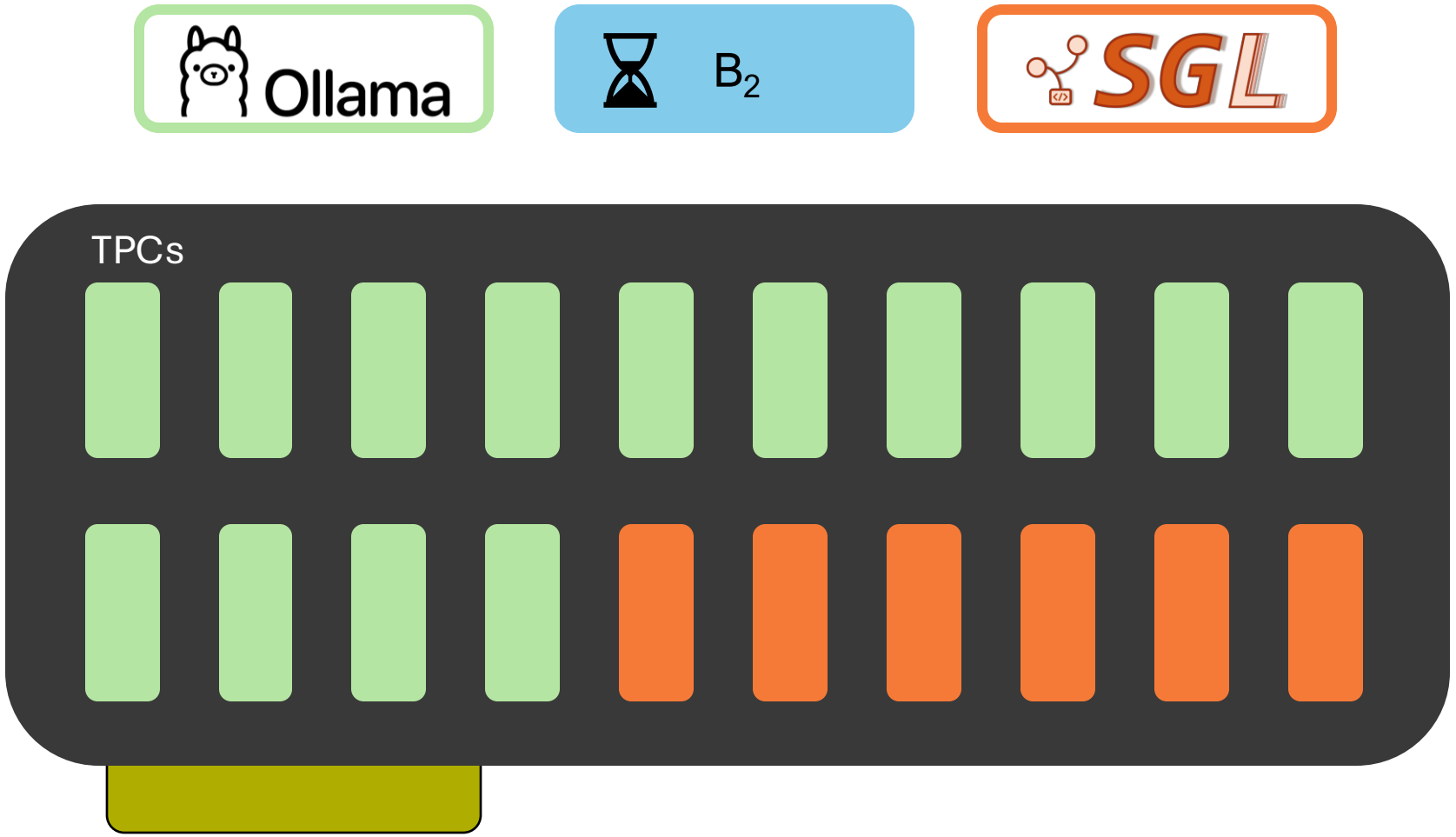
LithOS TPC Scheduling

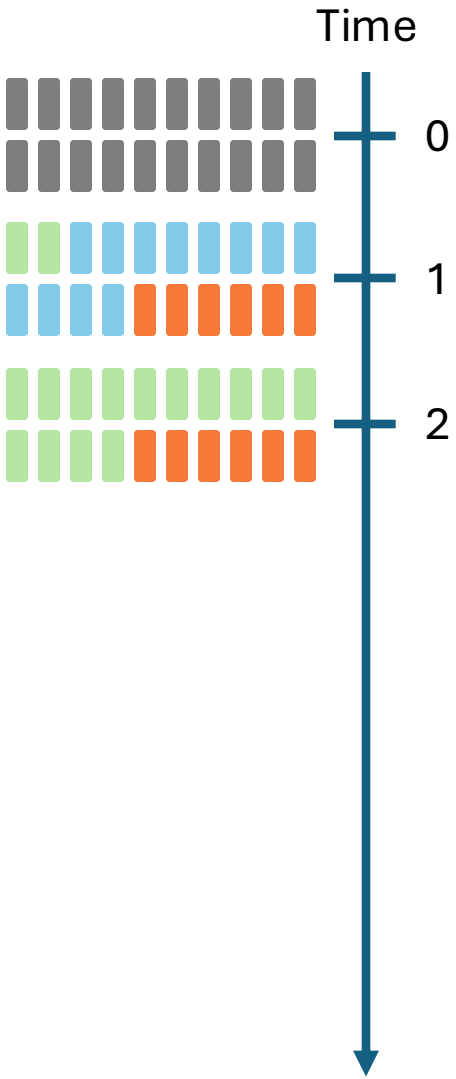


LithOS TPC Scheduling

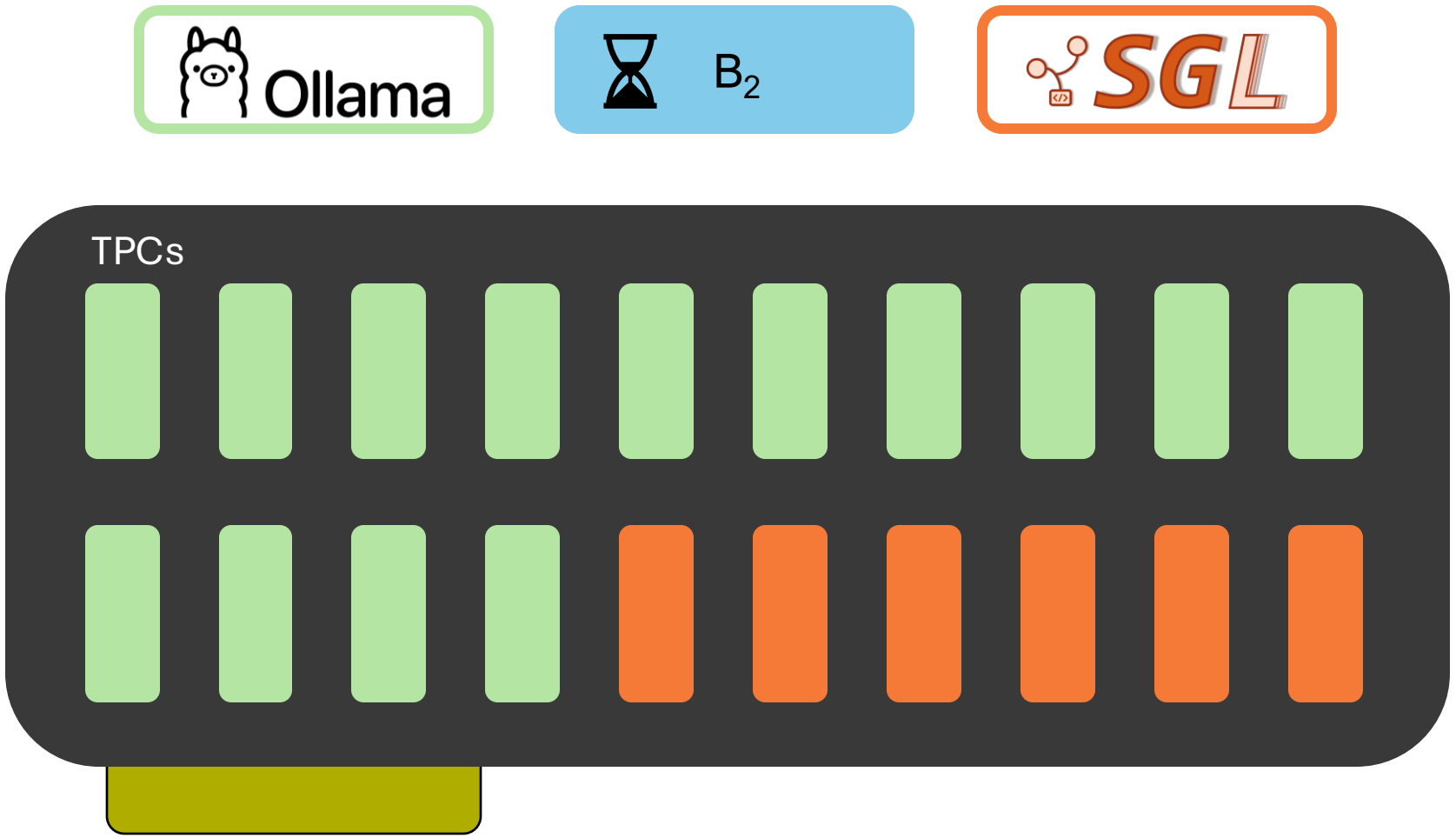


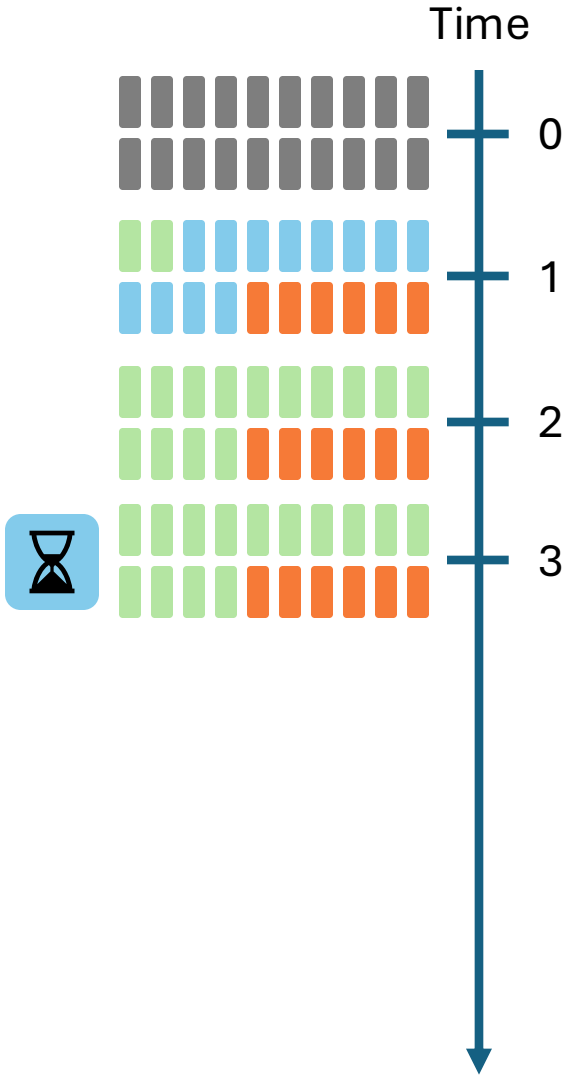
LithOS TPC Scheduling



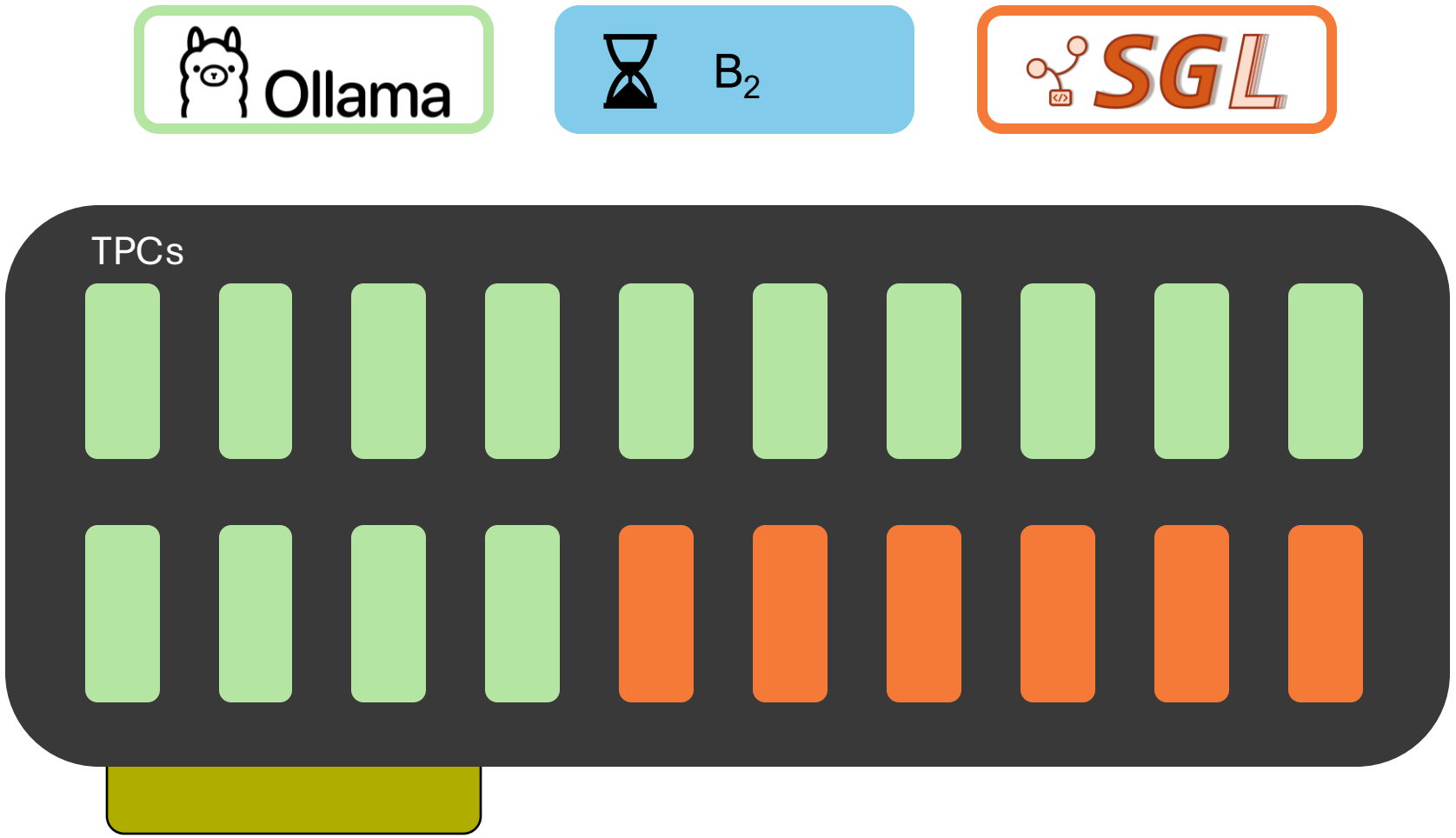


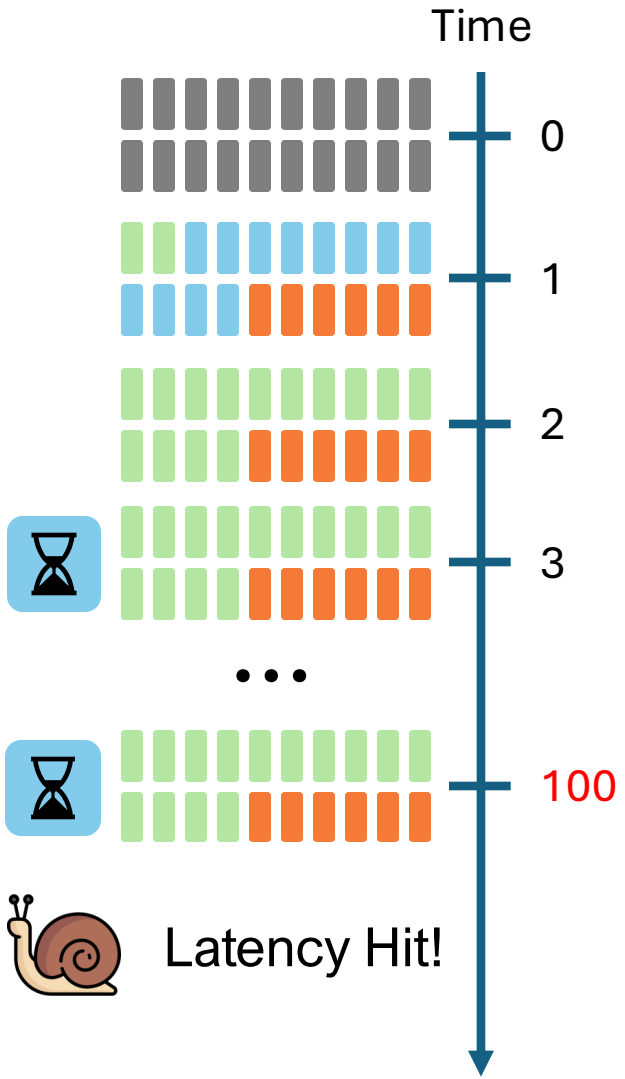
LithOS TPC Scheduling





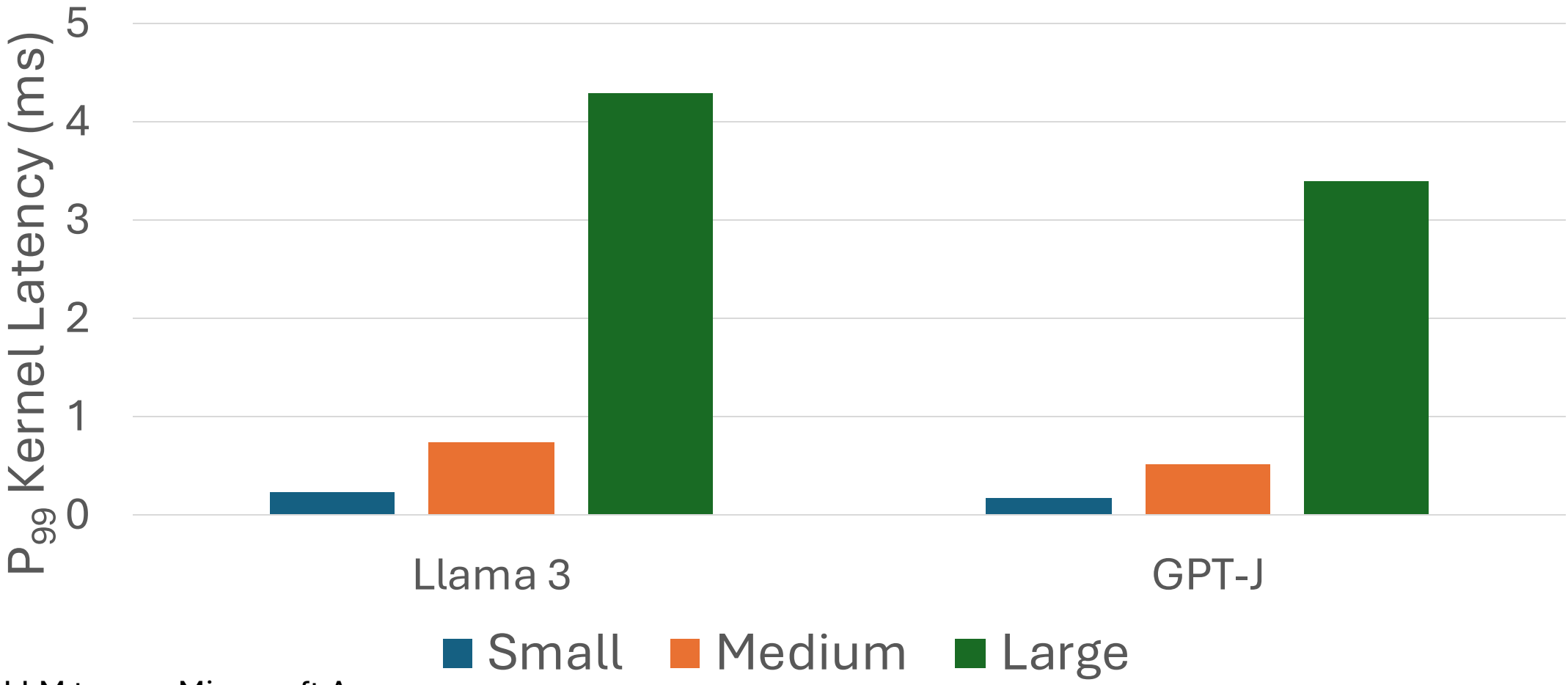
LithOS TPC Scheduling





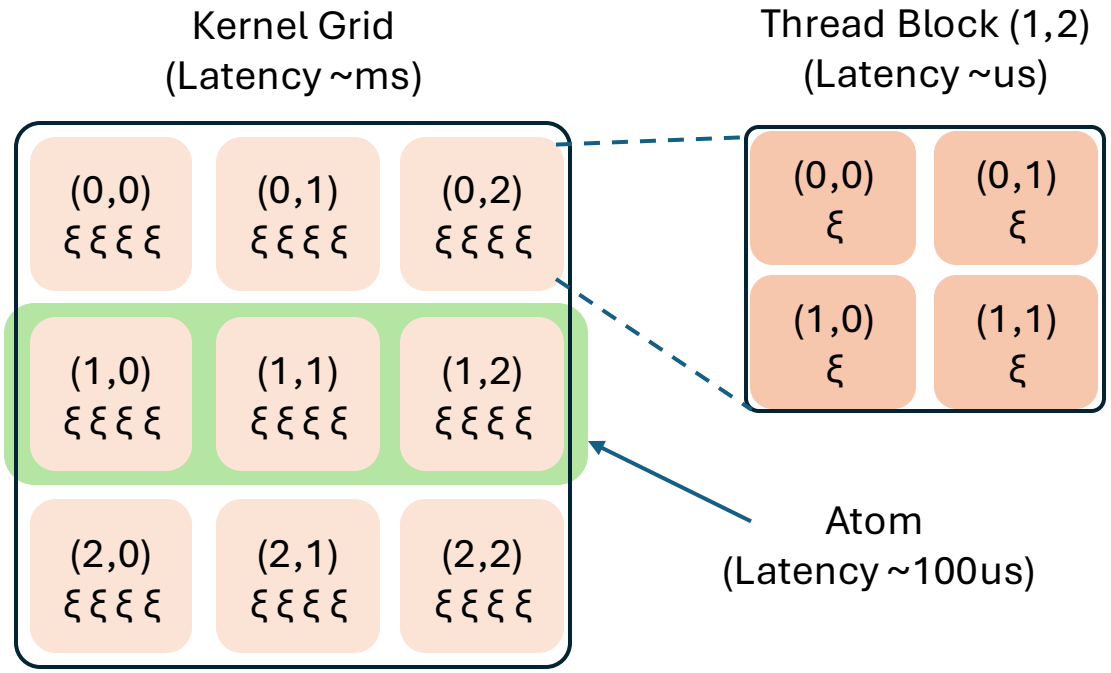
Kernel Runtimes Can Be Long

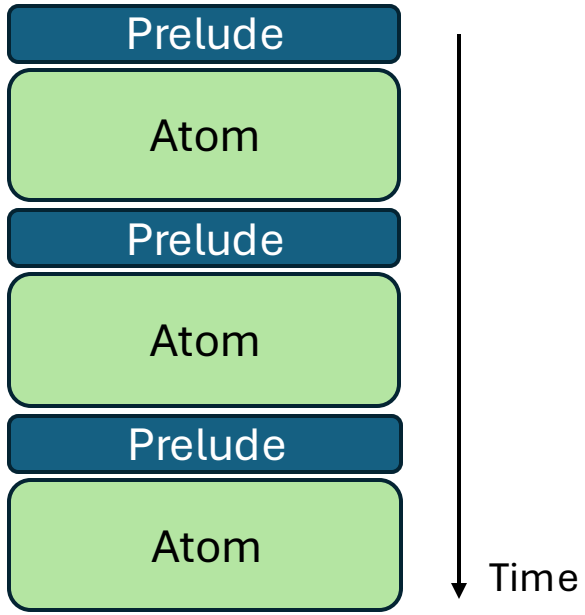
More data in the paper!



* DynamoLLM trace - Microsoft Azure

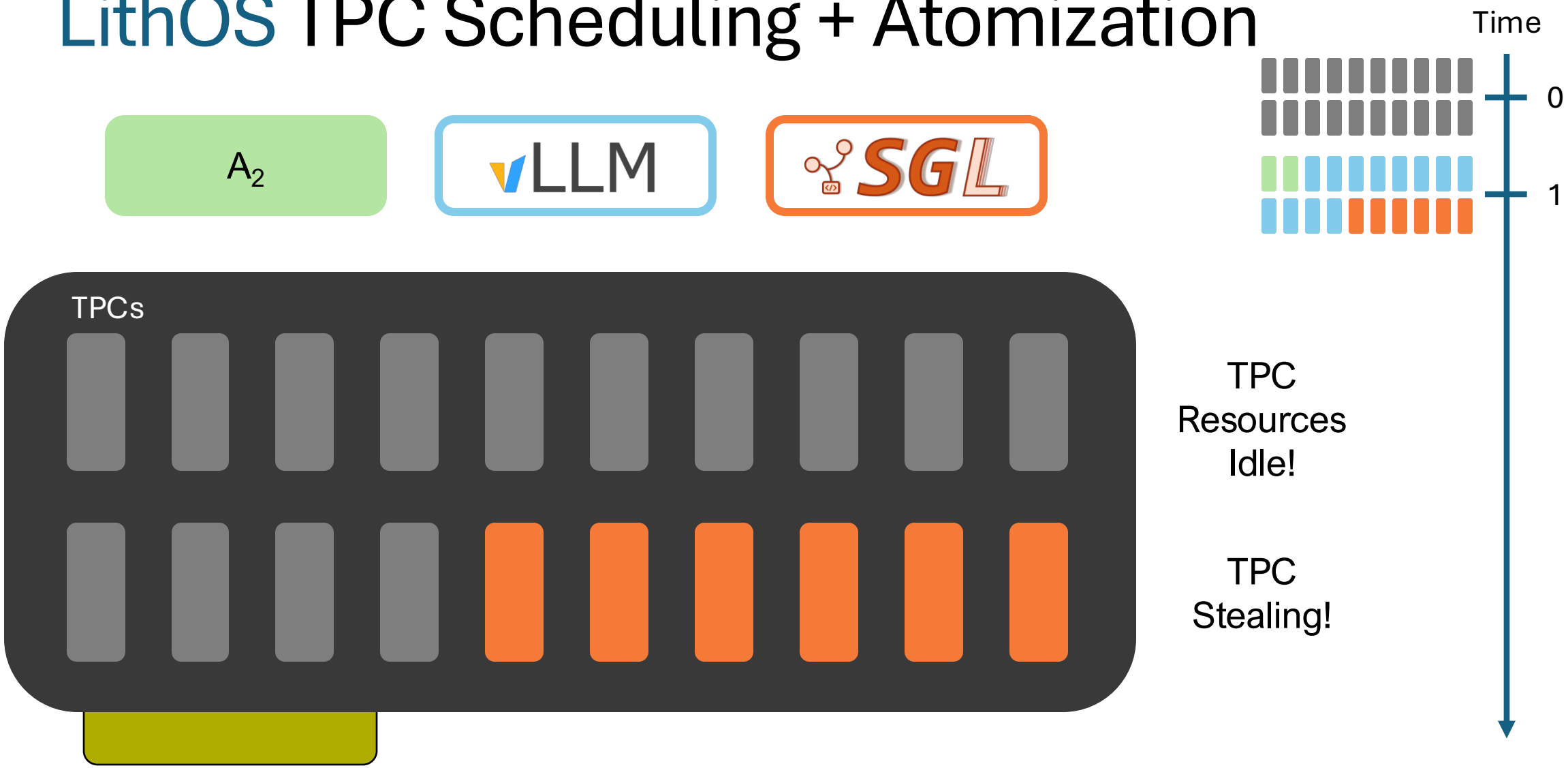
Transparent Kernel Atomization For Preemption



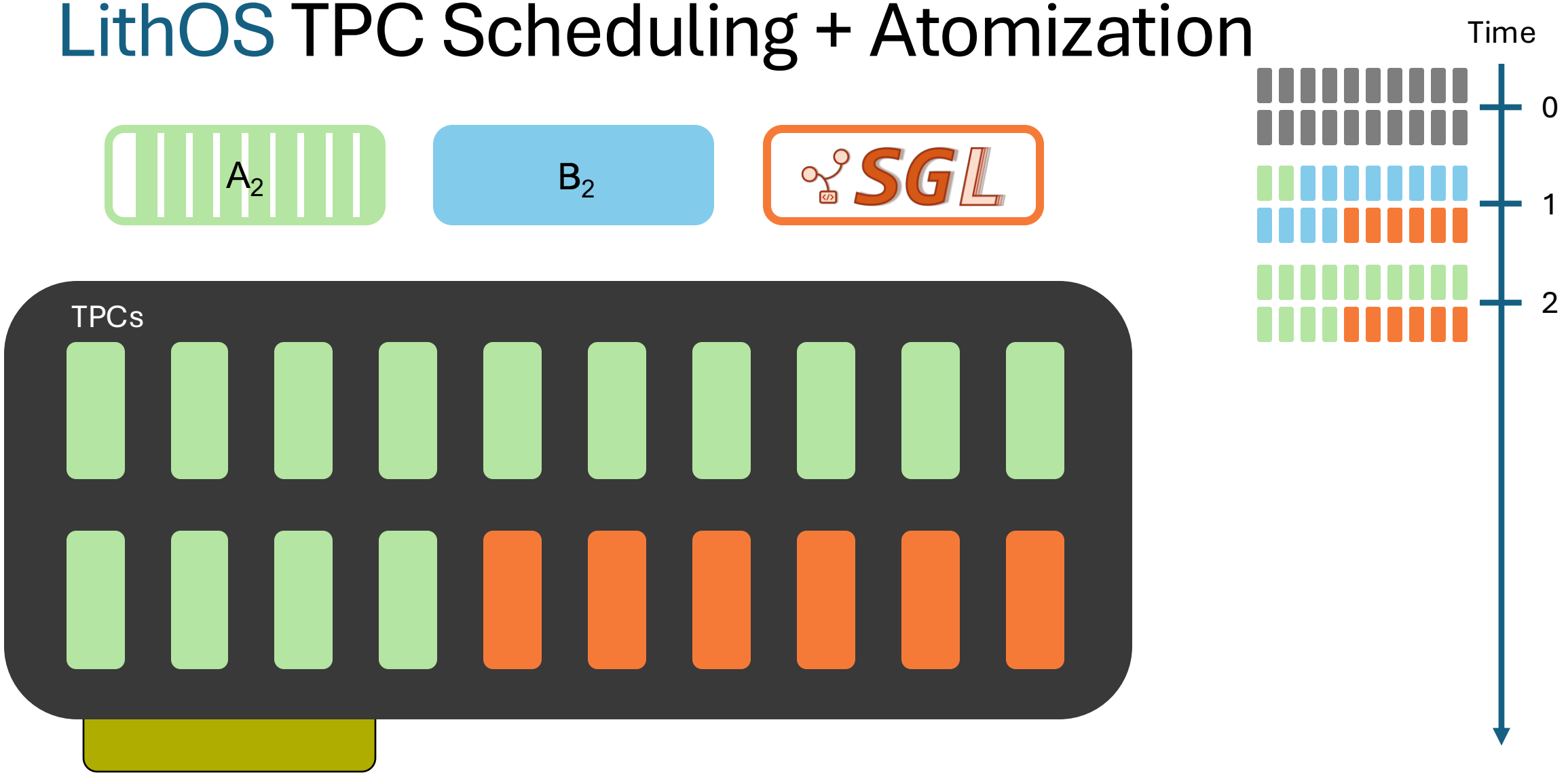


Thread-block-level scheduling in software!

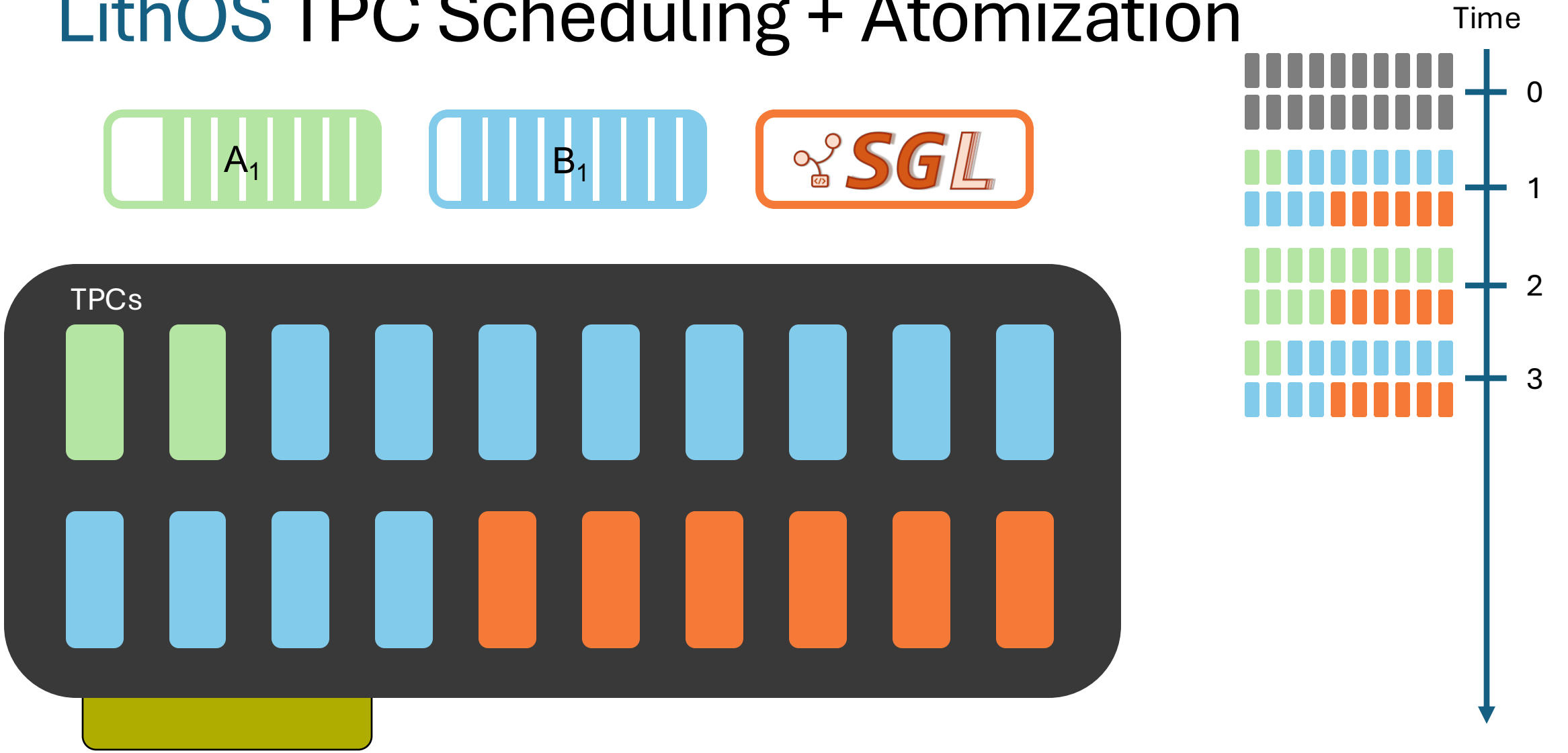
LithOS TPC Scheduling + Atomization



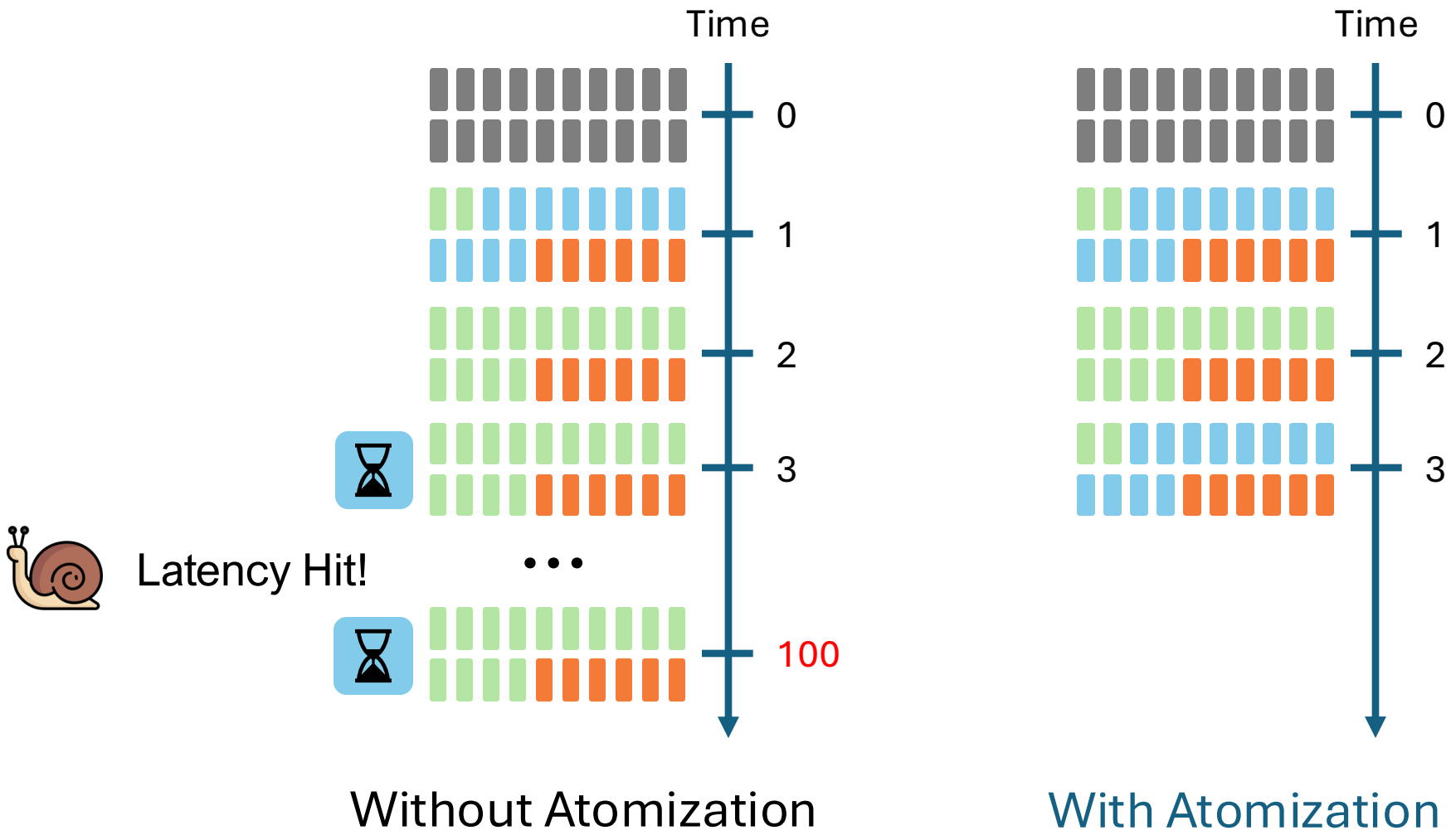
LithOS TPC Scheduling + Atomization



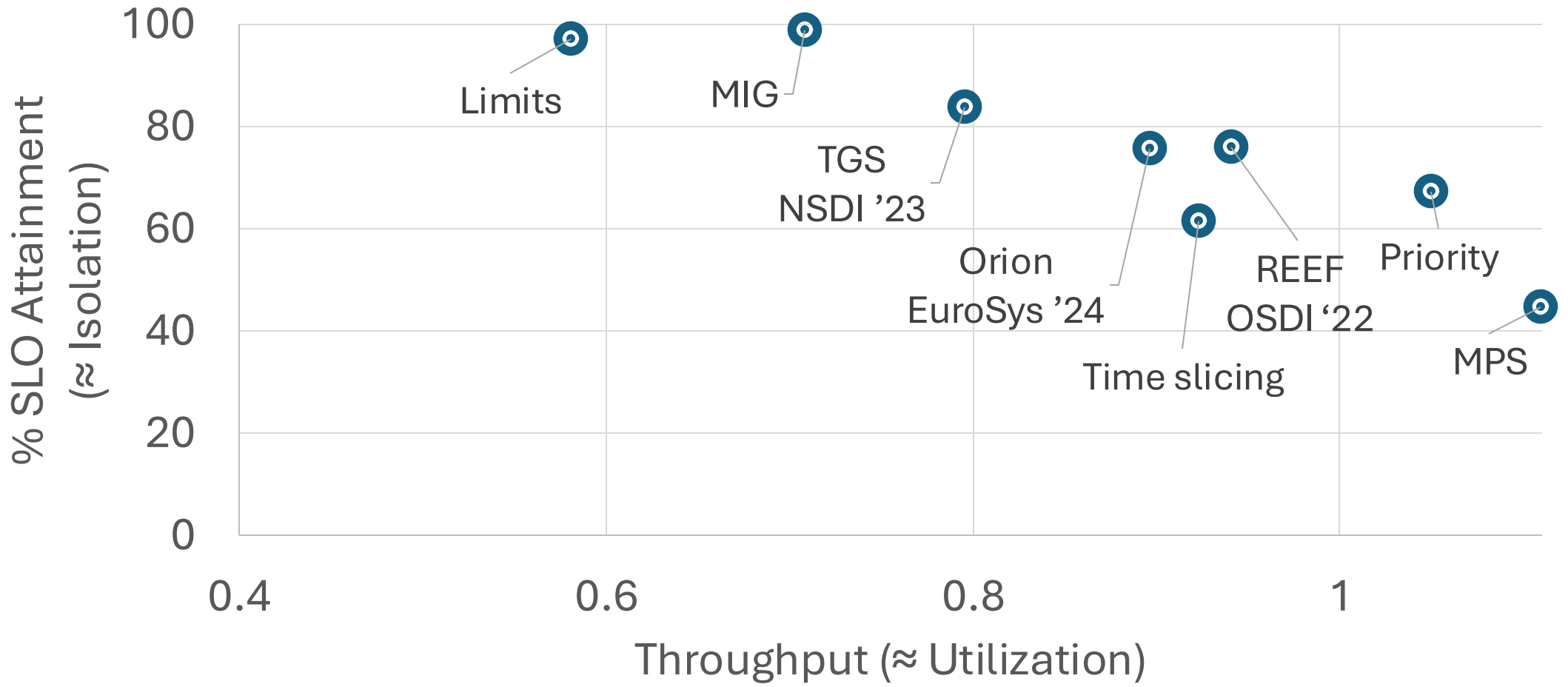
LithOS TPC Scheduling + Atomization



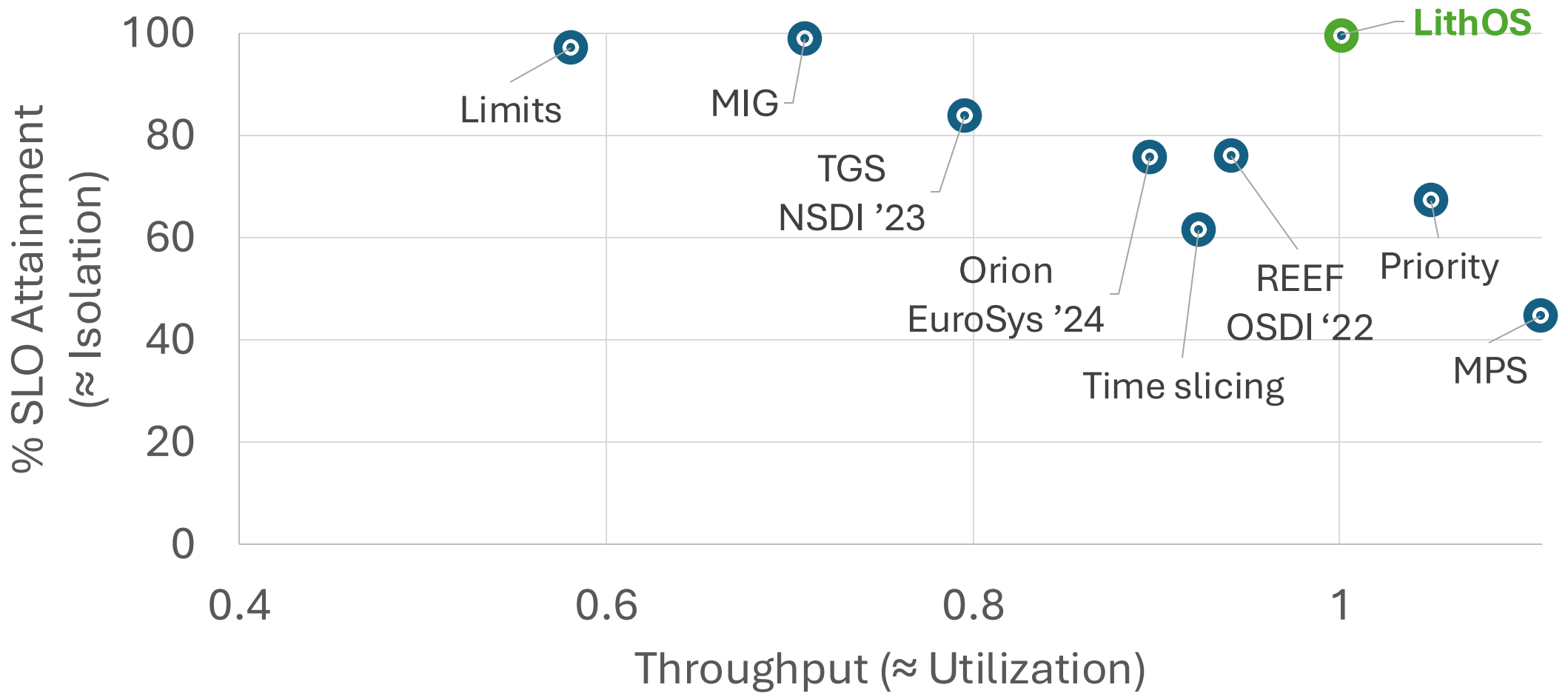
LithOS TPC Scheduling + Atomization



The Latency SLO vs. Throughput Trade-off



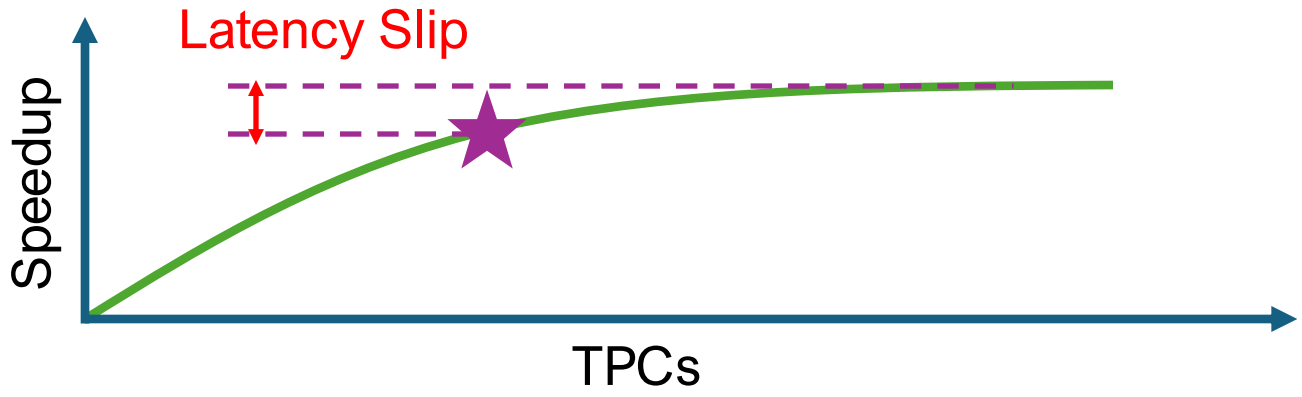
The Latency SLO vs. Throughput Trade-off



TPC Right-Sizing and Fine-Grained DVFS

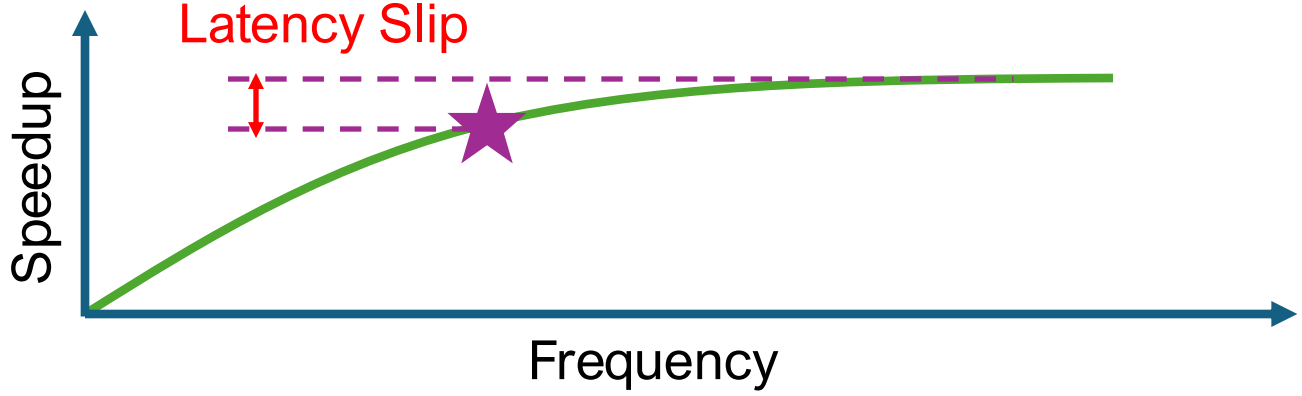
Right-Sizing

25% capacity savings



Dynamic Voltage Frequency Scaling

25% energy savings



Check Out the Paper for More Details!

Discussion

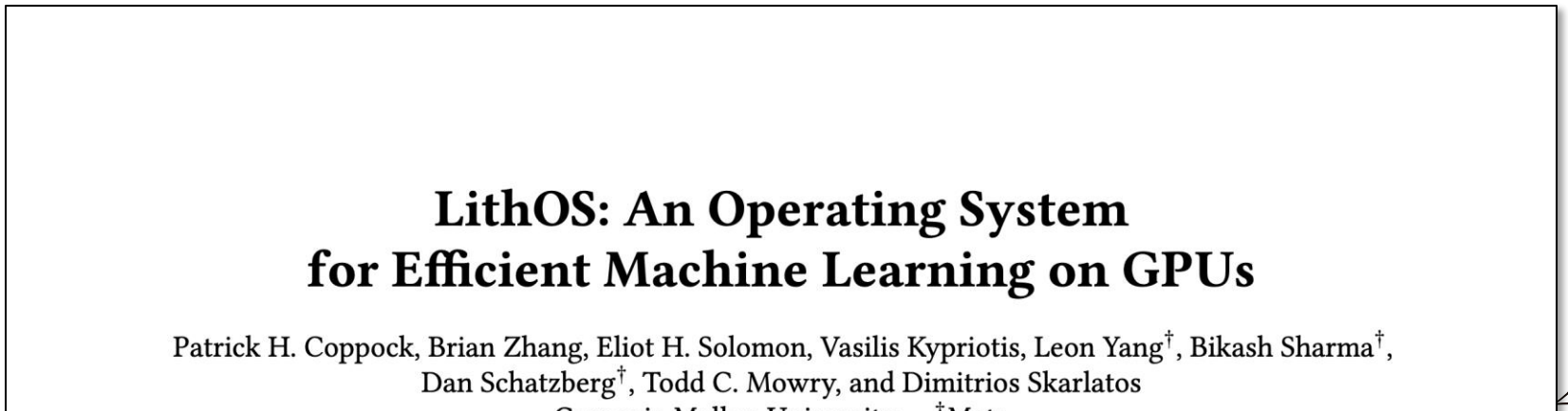
- Abstractions for GPU OSes

Mechanisms

- Kernel/TPC Right-Sizing
- DVFS
- Online latency predictor
- Reverse engineering techniques

Results

- Meta GPU fleet profiling
- Kernel scaling vs. compute/freq.
- Inference-Inference stacking
- Inference-Training stacking
- Ablation studies



LithOS: An Operating System for Efficient Machine Learning on GPUs

Patrick H. Coppock, Brian Zhang, Eliot H. Solomon, Vasilis Kypriotis, Leon Yang†, Bikash Sharma†, Dan Schatzberg†, Todd C. Mowry, and Dimitrios Skarlatos

LithOS: An Operating System for GPUs

Fully transparent to ML stack

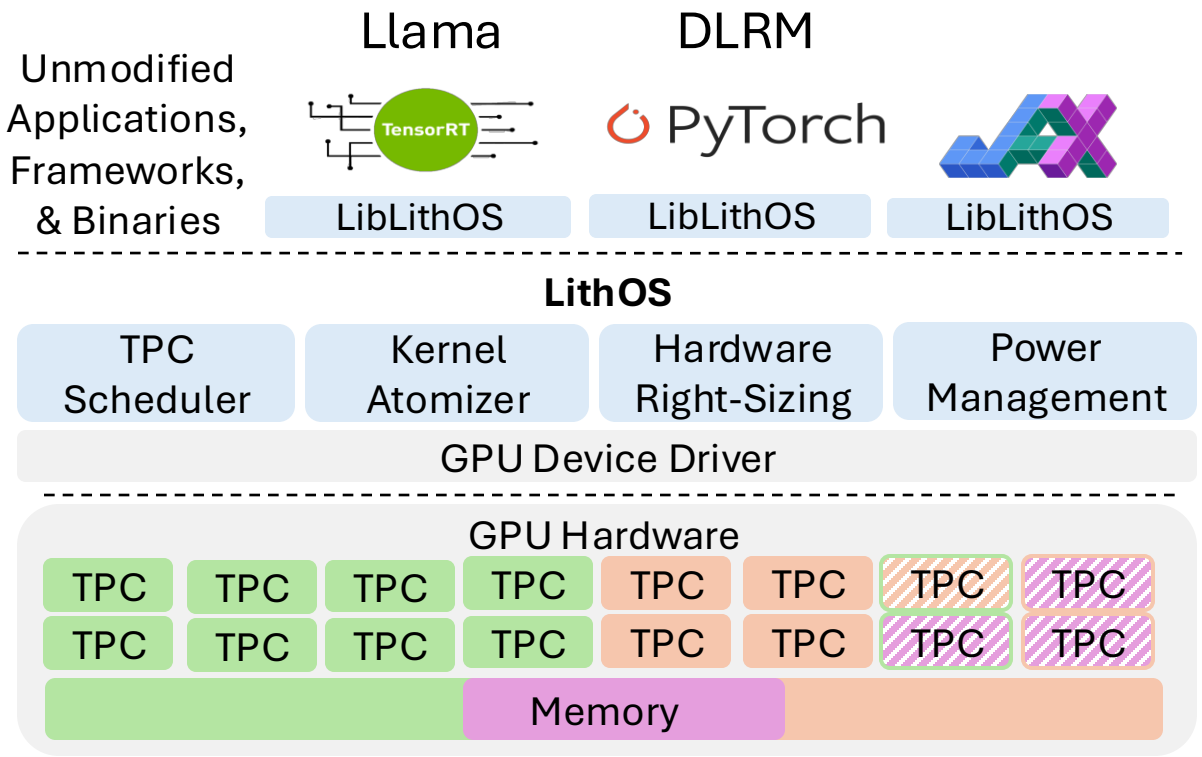
- No PTX, framework, compiler changes
- TPC Scheduling + Stealing
- Kernel Atomizer
- Right-size kernels to TPCs
- Transparent DVFS

Better performance

- **13x** lower tail latencies vs. NVIDIA
- **3x** lower tail latencies vs. SotA$_1$
- **1.6x** higher throughput vs. SotA$_2$

Easy resource saving

- Right-Sizing: **25%** capacity savings
- DVFS: **25%** energy savings



Thanks! Any questions?