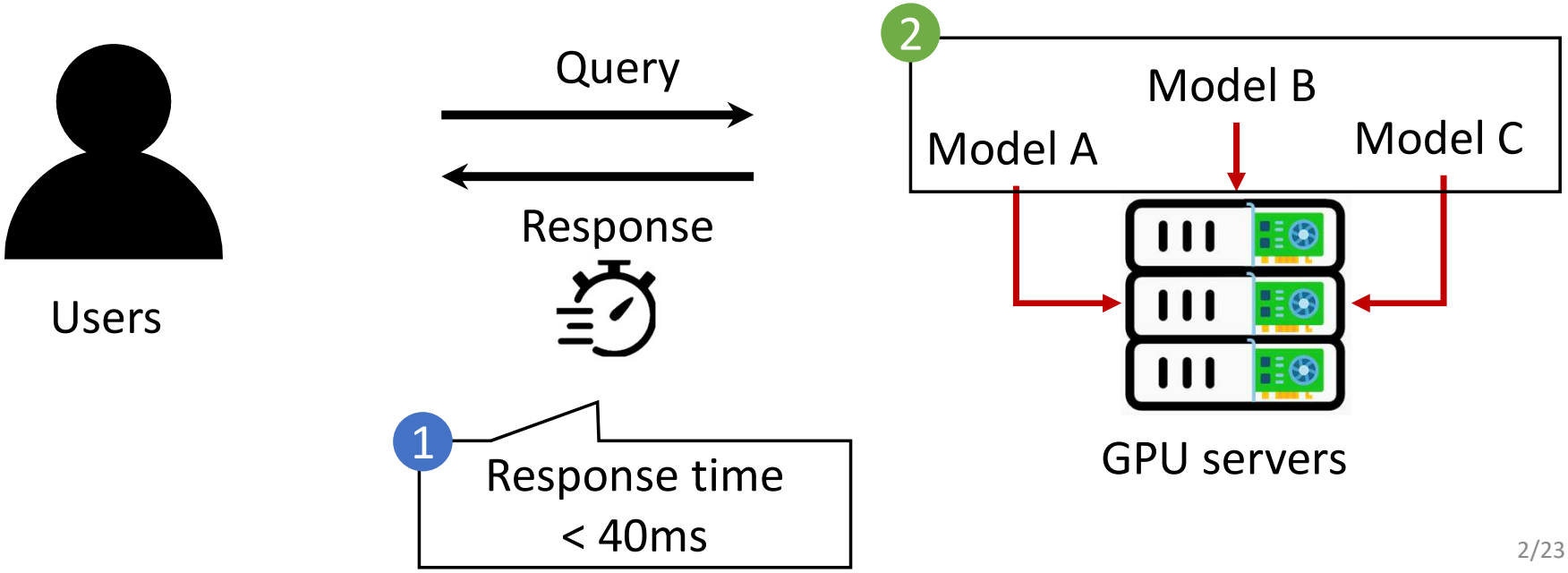
Serving Heterogeneous Machine Learning Models on Multi-GPU Servers with Spatio-Temporal Sharing

Seungbeom Choi, Sunho Lee, Yeonjae Kim,
Jongse Park, Youngjin Kwon, Jaehyuk Huh

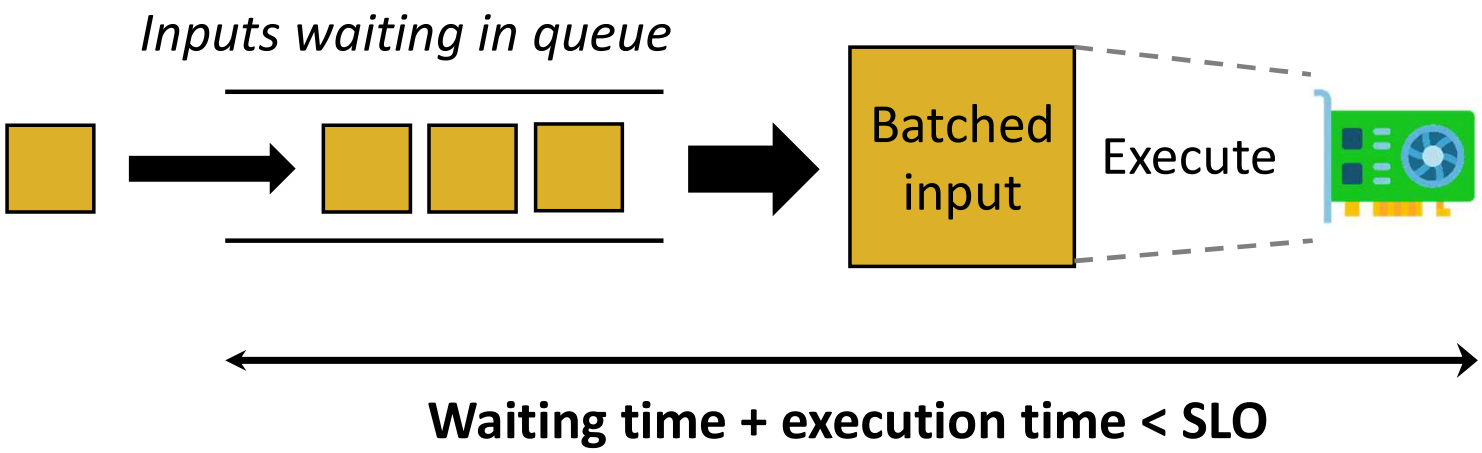
Machine Learning (ML) Inference in GPUs

- GPUs are widely adopted as inference accelerator
- Following **requirements** must be satisfied:
  1. Serve queries in a bounded time, *service-level objective* (SLO)
  2. Serve multiple-heterogeneous ML models



Prior Approach: Batching

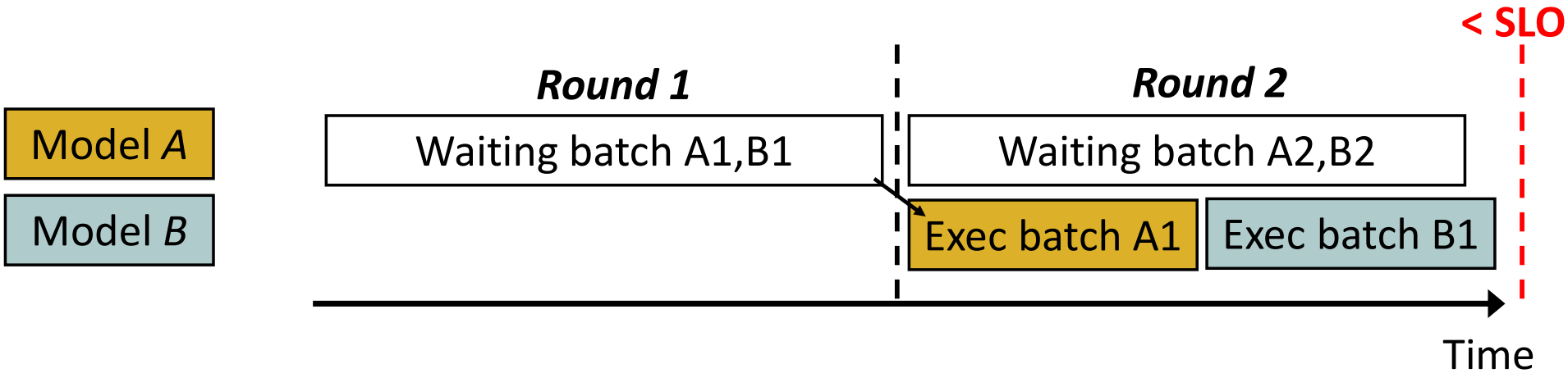
- **Batching:** Merge inputs to a single large input [1], [2], [3]
  - Improves throughput and utilization of GPU
  - **Batch size could not be huge due to SLO**



[1] Clipper [ATC'17]
[2] Clockwork [OSDI'20]
[3] Nexus [SOSP'19]

Prior Approach: Time-Sharing

- **Time-sharing:** Round-based interleaved execution of batches [1]
  - Increase utilization by reducing idle time on GPU
  - **Guarantee 2 rounds < SLO**



[1] Nexus [SOSP'19]

Prior Approach: Time-Sharing

- **Time-sharing:** Round-based interleaved execution of batches [1]
  - Increase utilization by reducing idleness on GPU
  - **Guarantee 2 rounds < SLO**

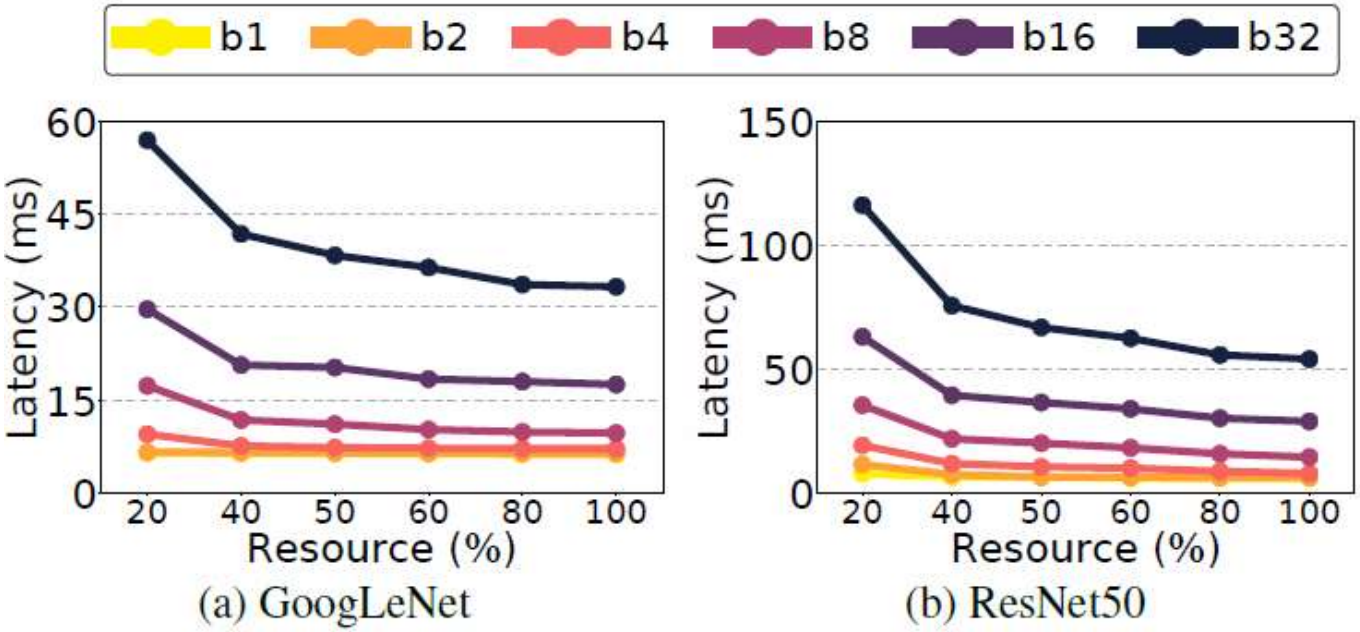
Problem with prior approaches

Batching and time-sharing inference, *underutilize GPUs*

[1] Nexus [SOSP'19]

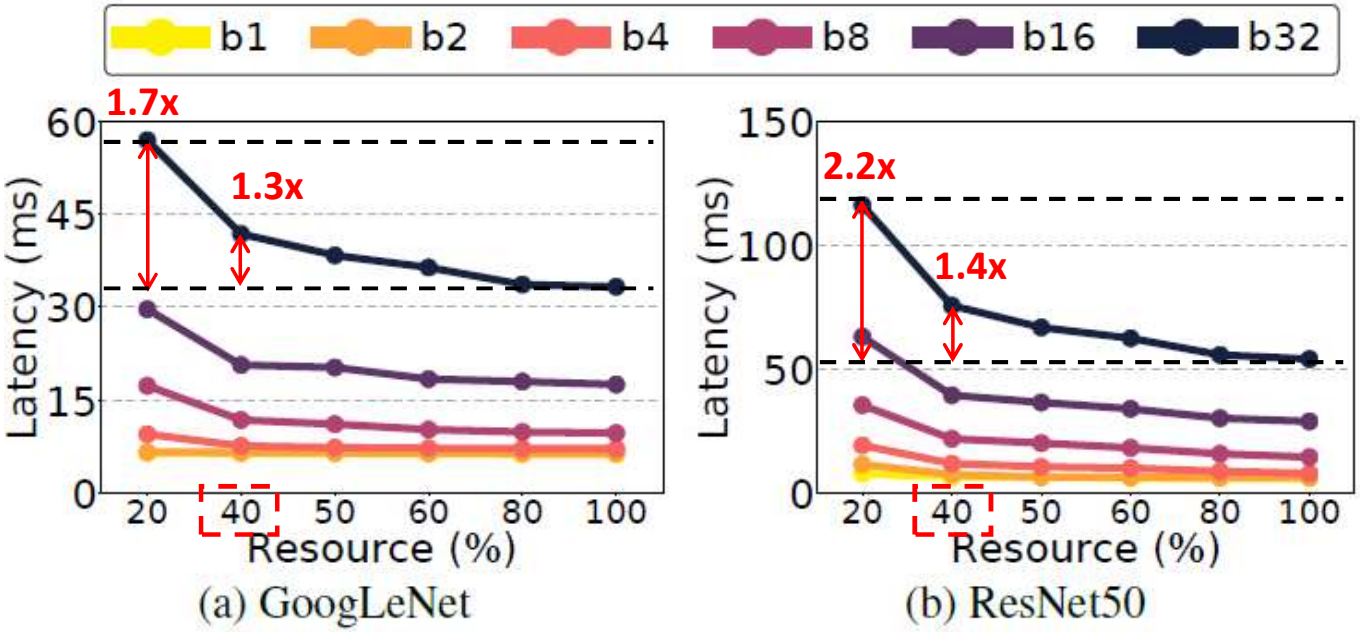
Underutilized Resources

- Measured latency vs. computing resources w/ varying batch size



Underutilized Resources

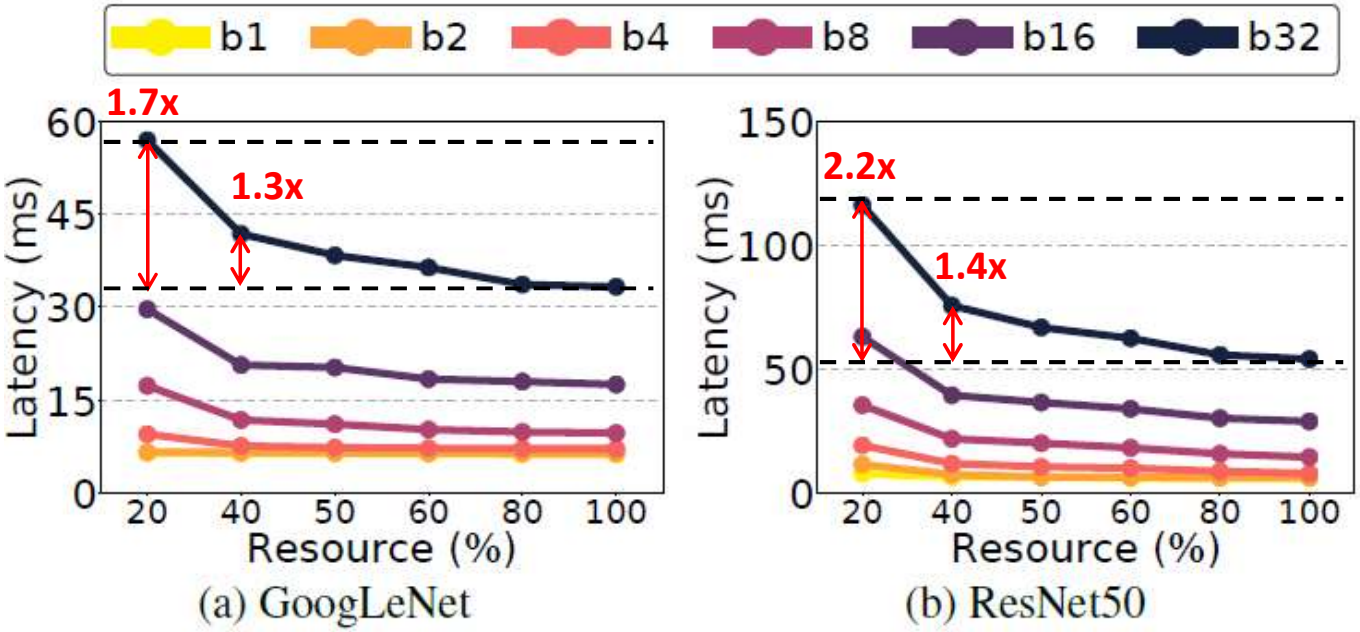
- Measured latency vs. computing resources w/ varying batch size



Diminishing return
beyond 40%

Underutilized Resources

- Measured latency vs. computing resources w/ varying batch size

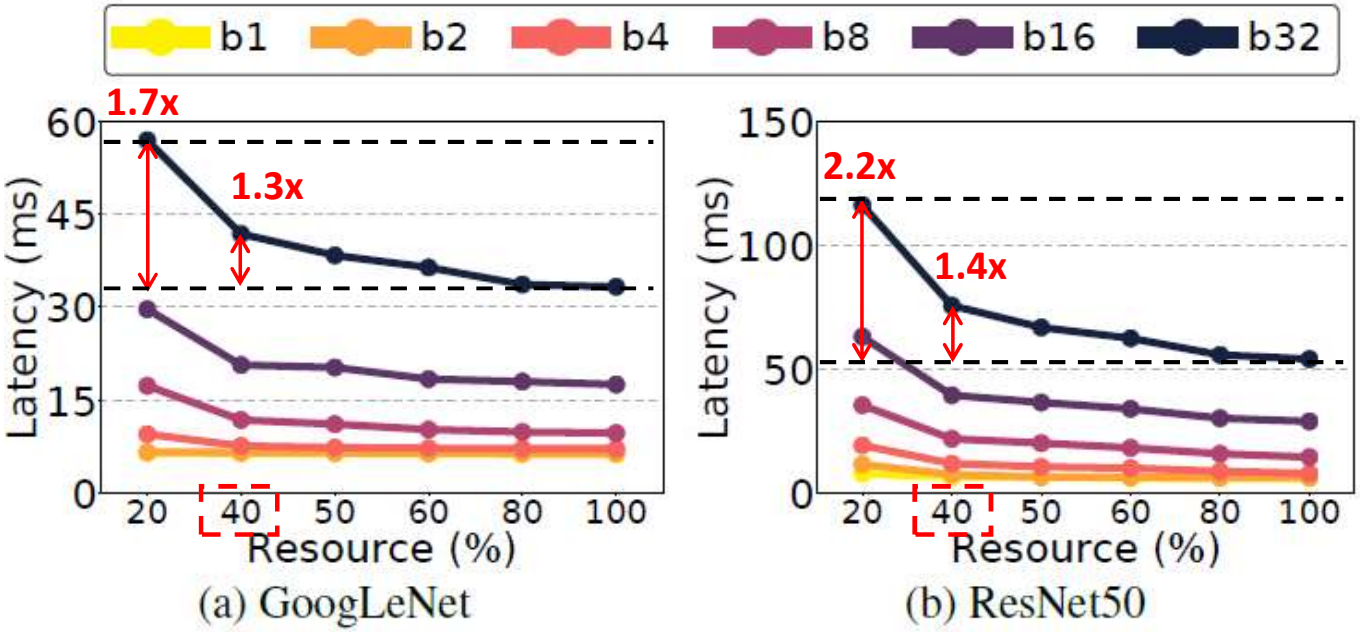


Diminishing return **beyond 40%**

Little improvement in **smaller batch** sizes

Underutilized Resources

- Measured latency vs. computing resources w/ varying batch size



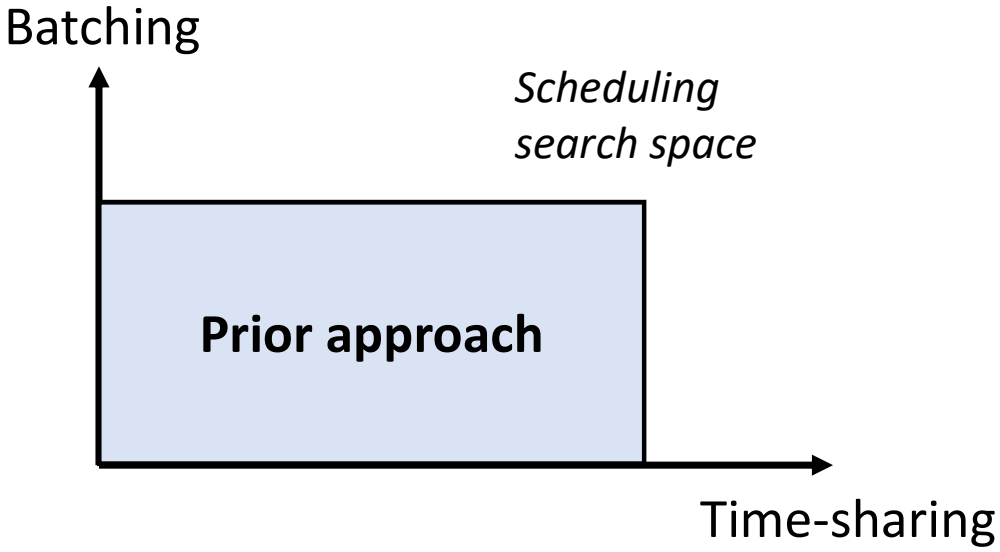
Diminishing return **beyond 40%**

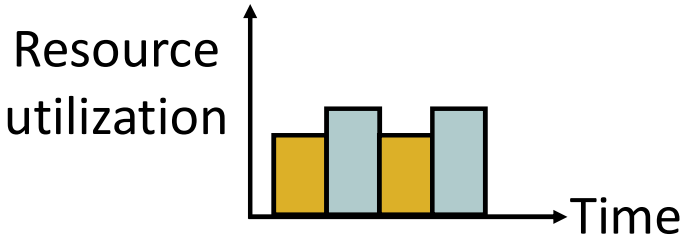
Little improvement in **smaller batch** sizes

Opportunities for improving performance with better resource utilization

New Opportunity: Spatio-temporal Scheduling

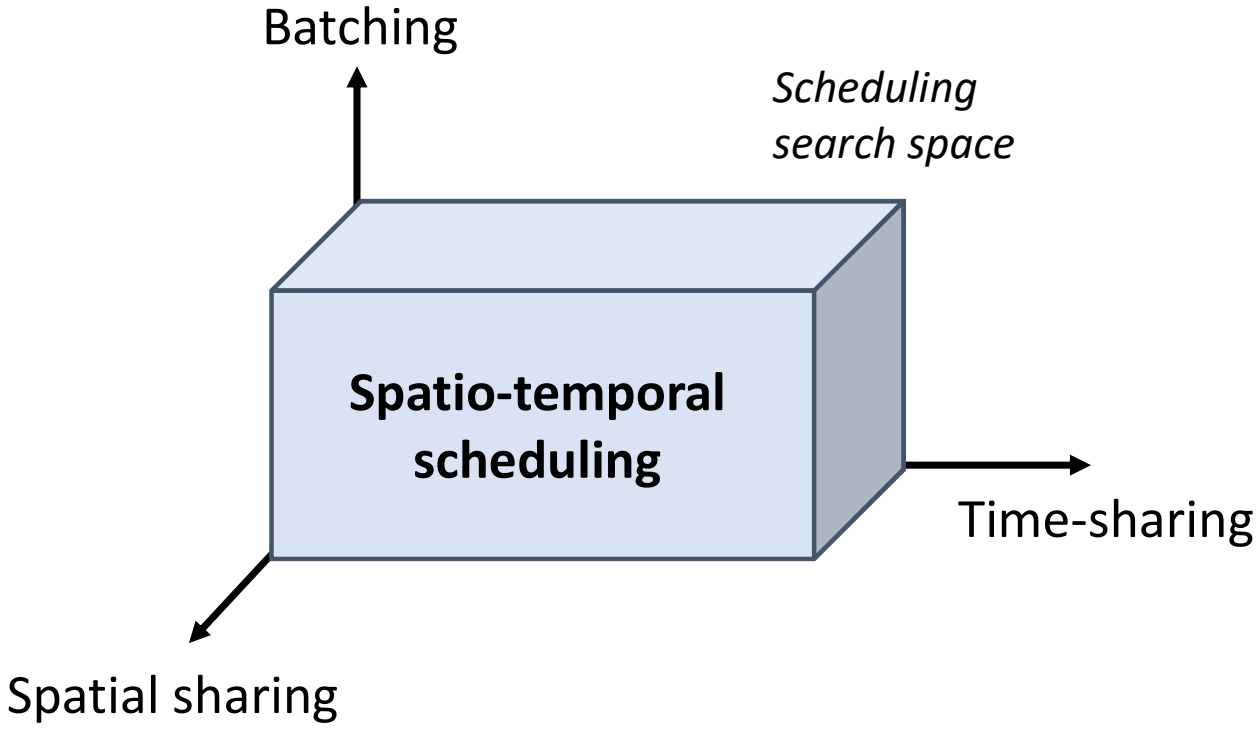
- **Spatio-temporal scheduling**:
  - Schedule tasks with batching, time-sharing, **and spatial sharing**

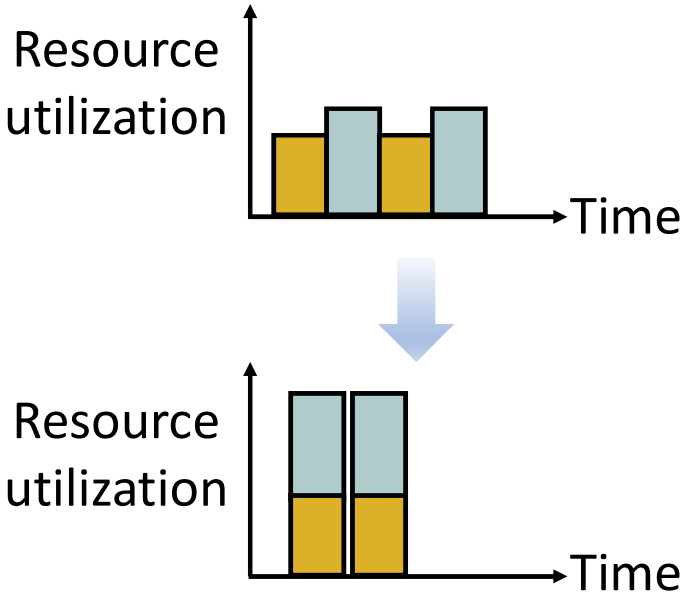




New Opportunity: Spatio-temporal Scheduling

- **Spatio-temporal scheduling**:
  - Schedule tasks with batching, time-sharing, **and spatial sharing**

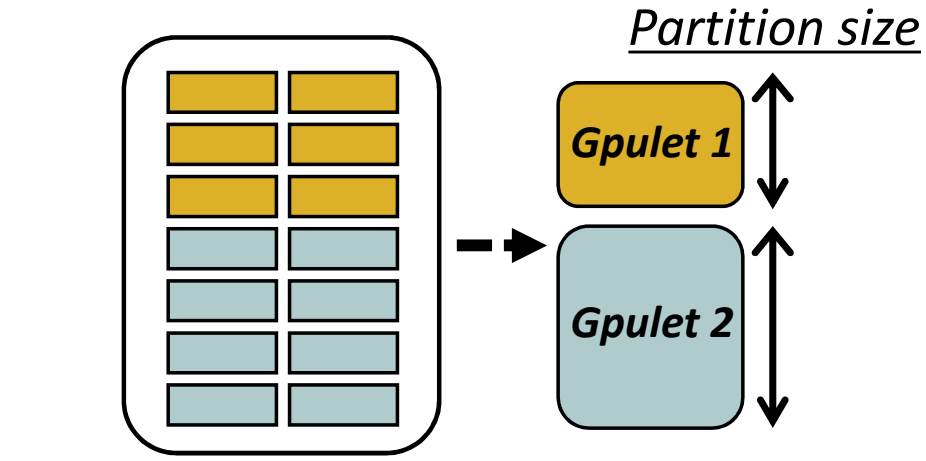


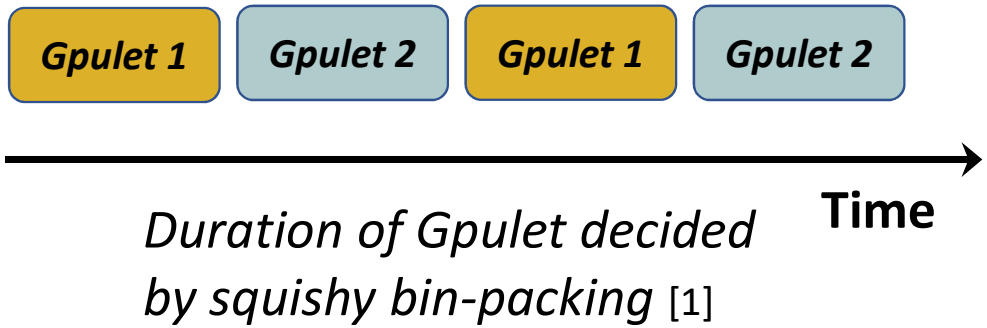


Better utilization → Improved throughput

New Abstraction: Gpulet

- Need an abstraction of spatial/temporal resource
- **Gpulet**: A share of spatial/temporal partition of GPU resource



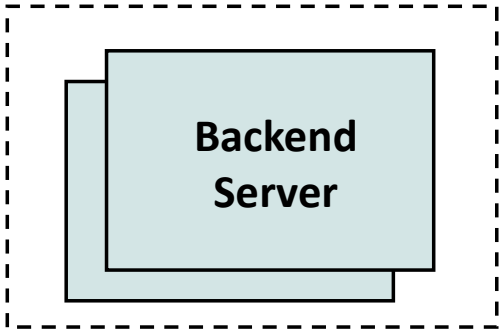


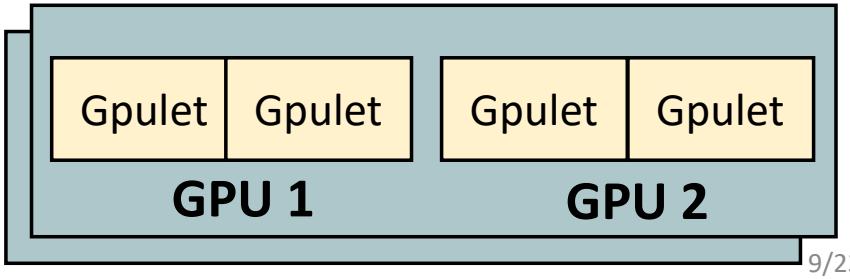
[1] Nexus [SOSP'19]

Overview of Gpulet Scheduling Framework

Frontend Server

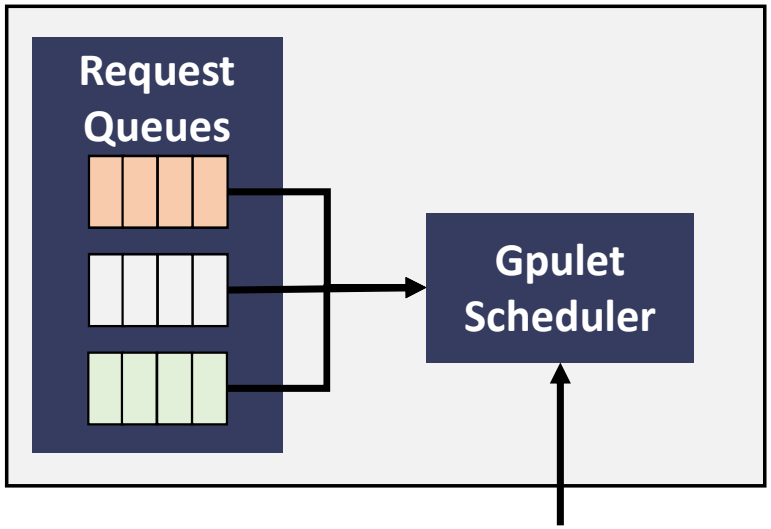
Backend Servers



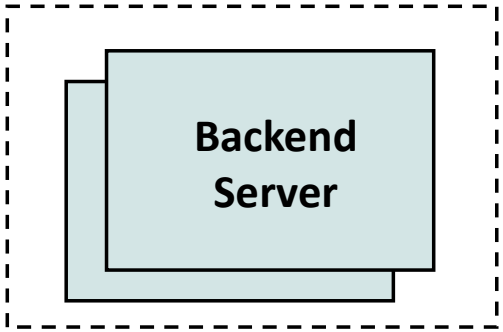


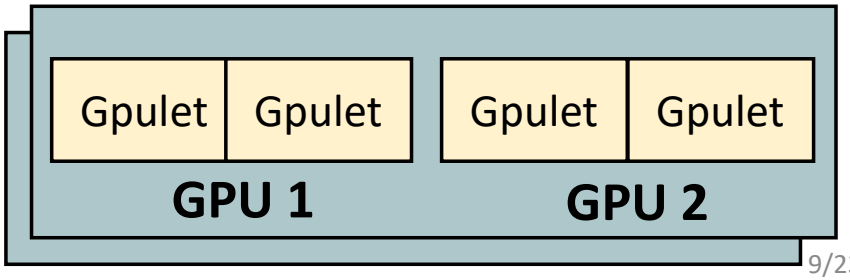
Overview of Gpulet Scheduling Framework

Frontend Server

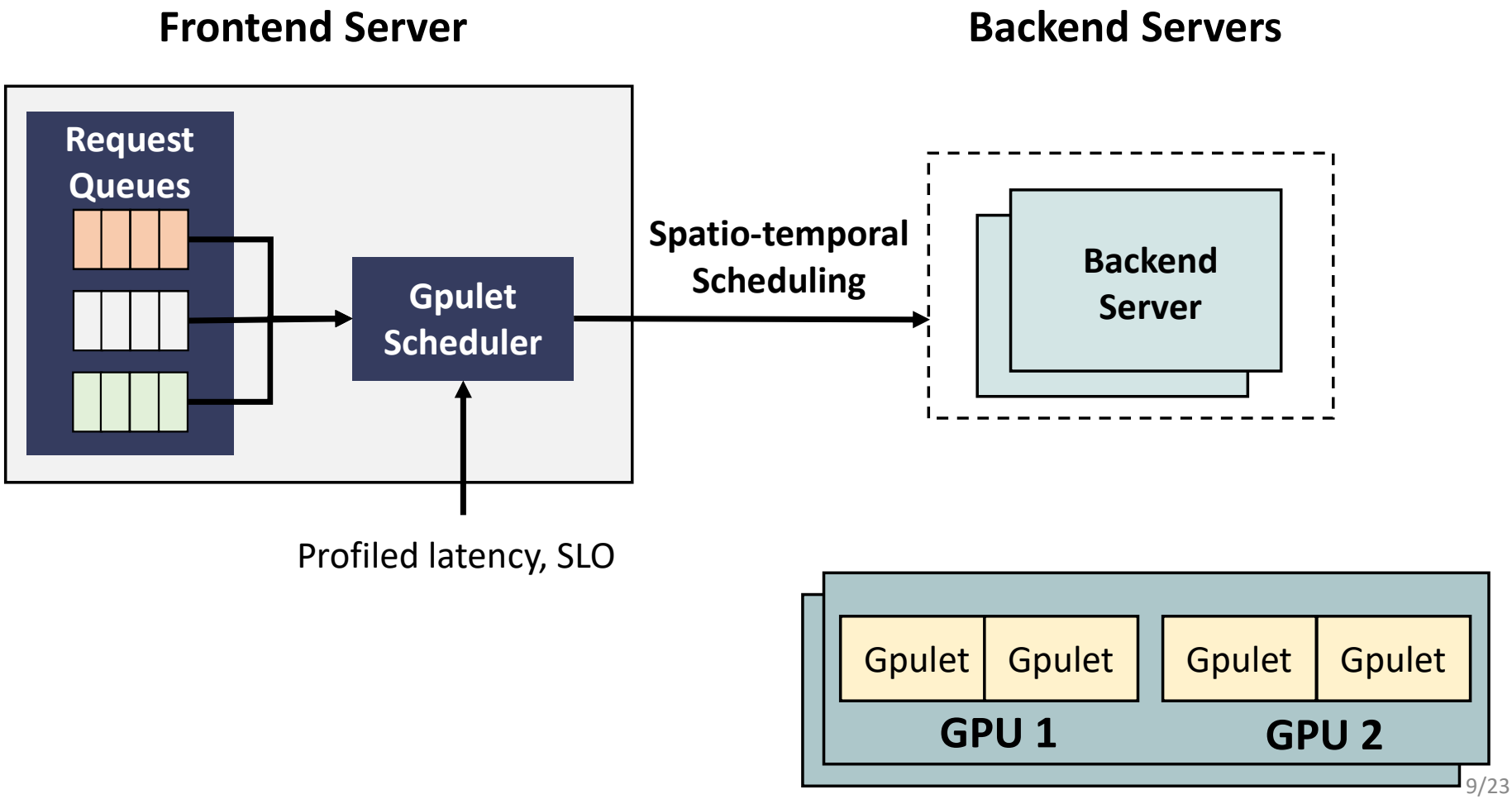


Backend Servers

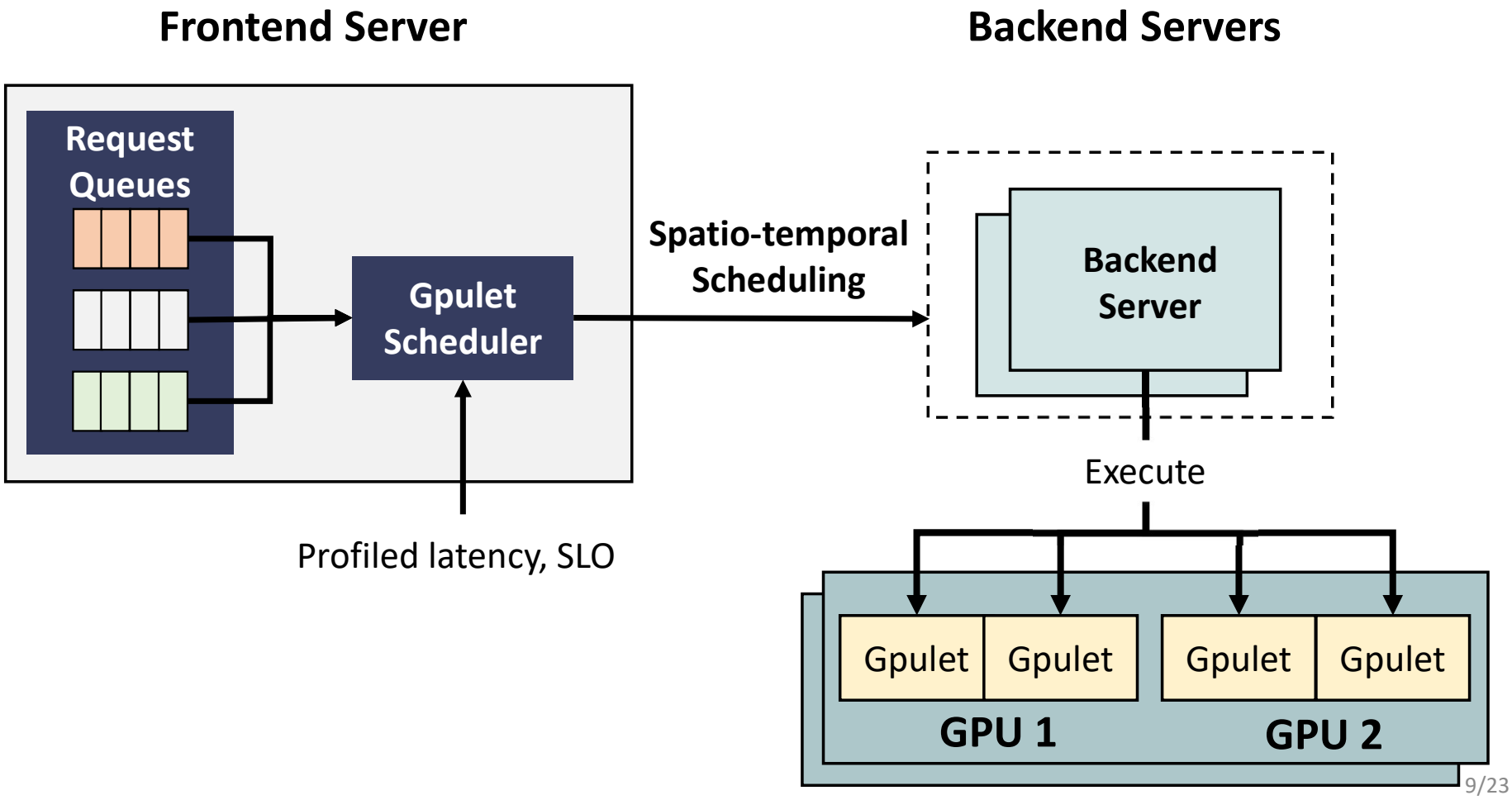




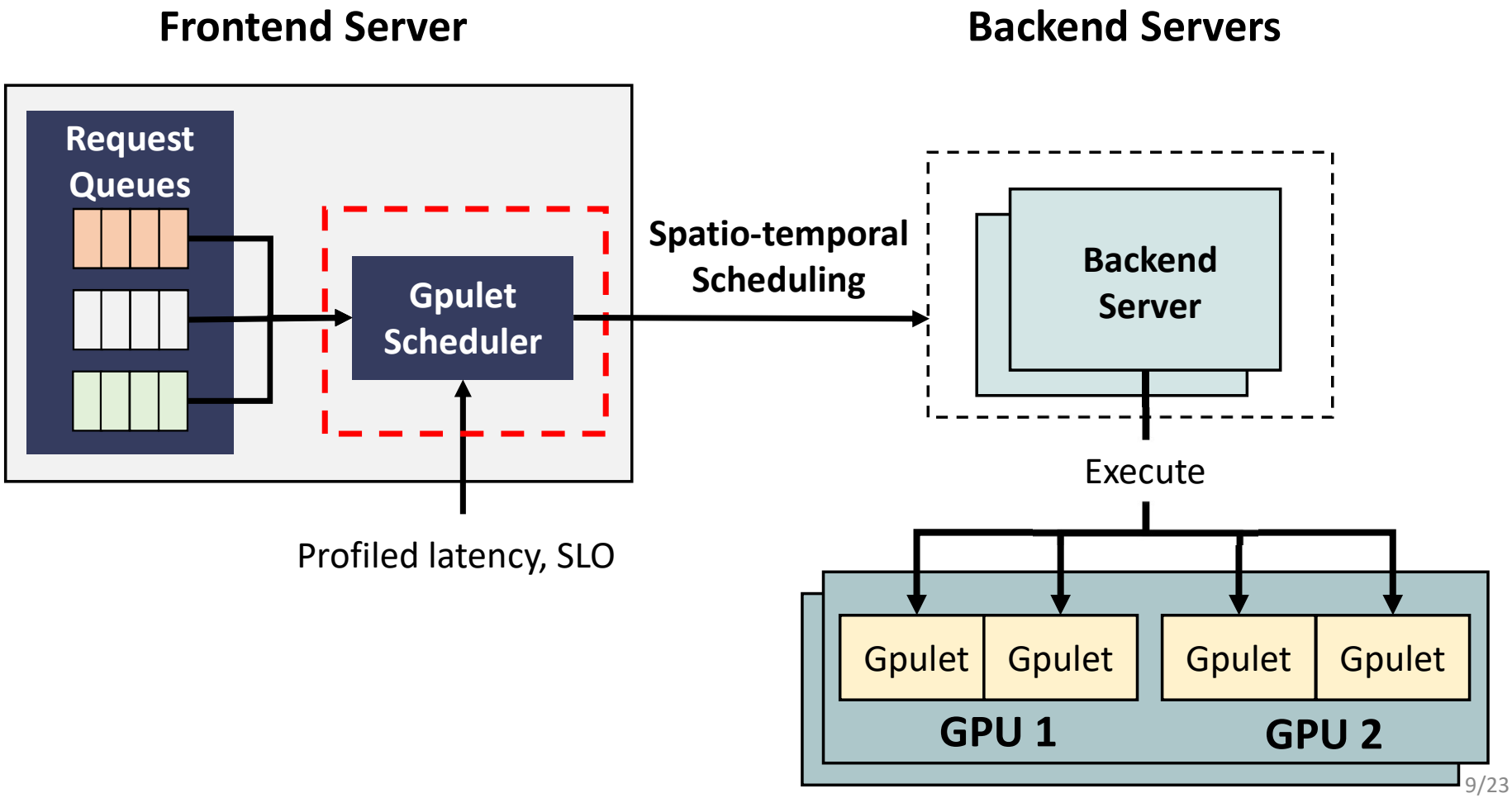
Overview of Gpulet Scheduling Framework



Overview of Gpulet Scheduling Framework

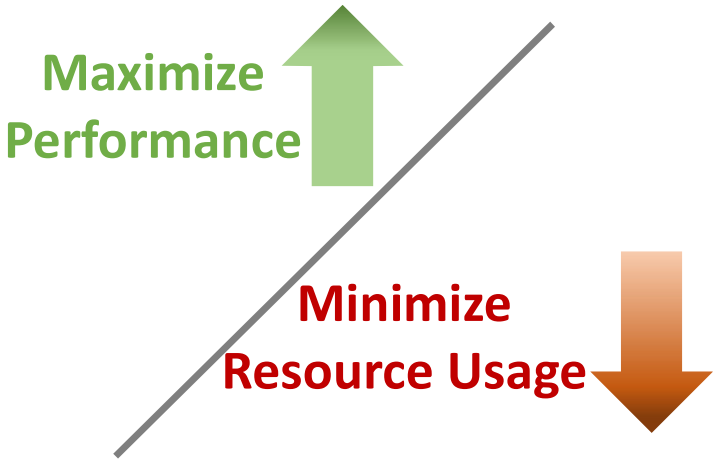


Overview of Gpulet Scheduling Framework



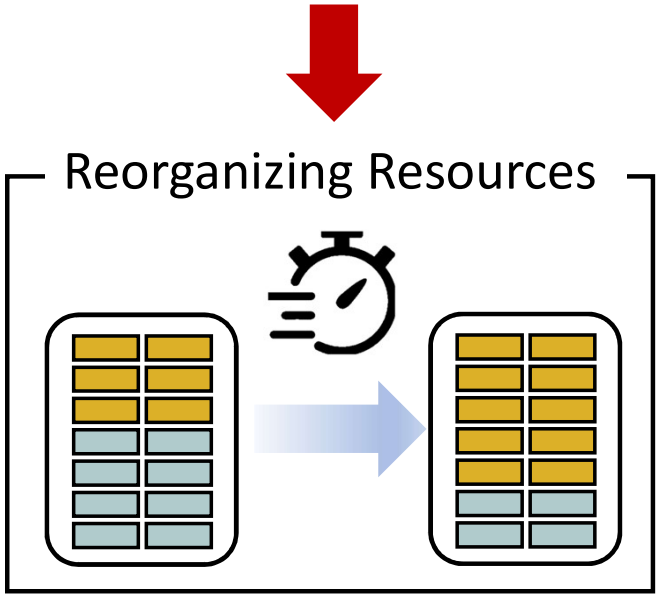
Design Overview of Gpulet Scheduler

Cost-effective scheduling

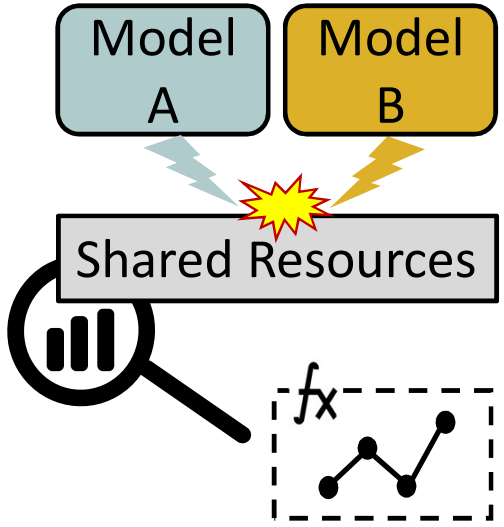


Dynamic reorganization

Scheduling event



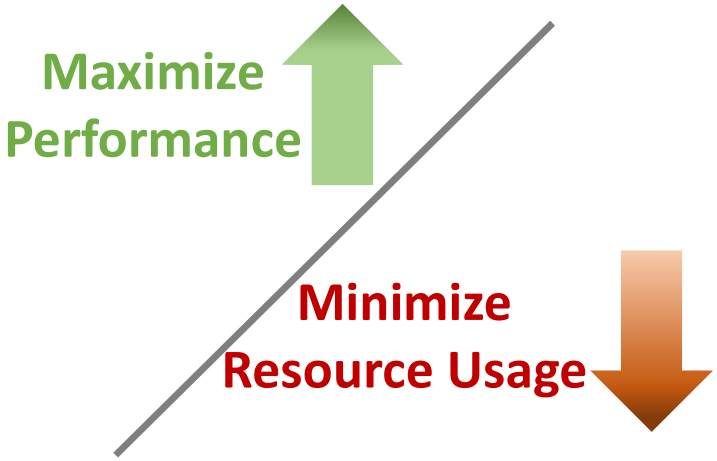
Interference prediction



Please refer to the paper!

Design Overview of Gpulet Scheduler

Cost-effective scheduling



Dynamic reorganization

Interference prediction

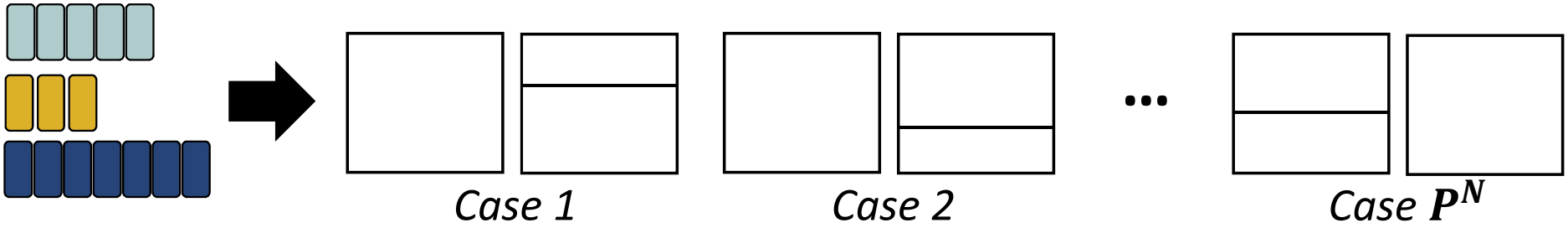
Please refer to the paper!

Scheduling Gpulets

- **Challenge:** Large search space for spatial scheduling
  - *P* spatial partitioning choices for *N* GPUs: $\boldsymbol{P^N}$ **cases to search exhaustively**

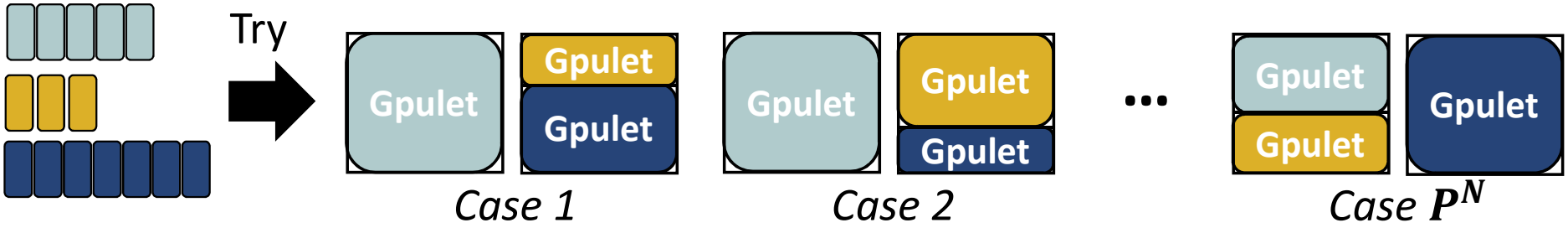
Scheduling Gpulets

- **Challenge:** Large search space for spatial scheduling
  - *P* spatial partitioning choices for *N* GPUs: $\boldsymbol{P^N}$ **cases to search exhaustively**



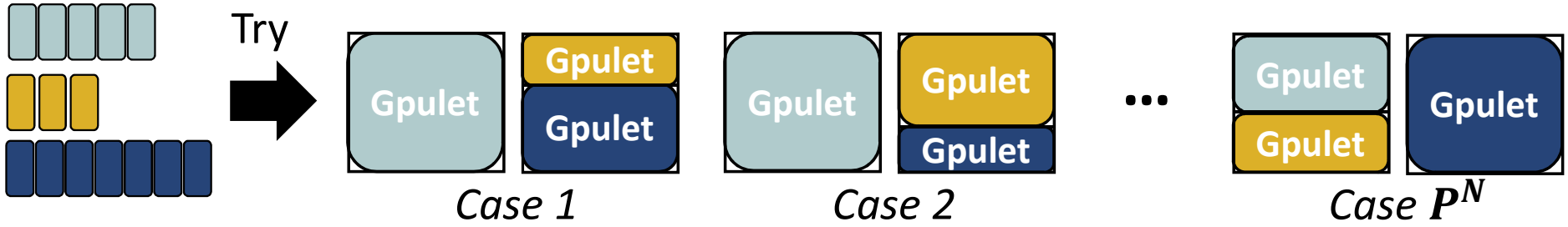
Scheduling Gpulets

- **Challenge:** Large search space for spatial scheduling
  - *P* spatial partitioning choices for *N* GPUs: $\boldsymbol{P^N}$ **cases to search exhaustively**

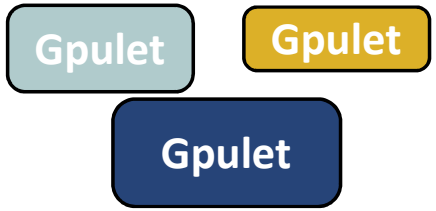


Scheduling Gpulets

- **Challenge:** Large search space for spatial scheduling
  - *P* spatial partitioning choices for *N* GPUs: $\boldsymbol{P^N}$ **cases to search exhaustively**



- **Main idea**: Allocate partitions to GPUs **incrementally**



cost-effective partition