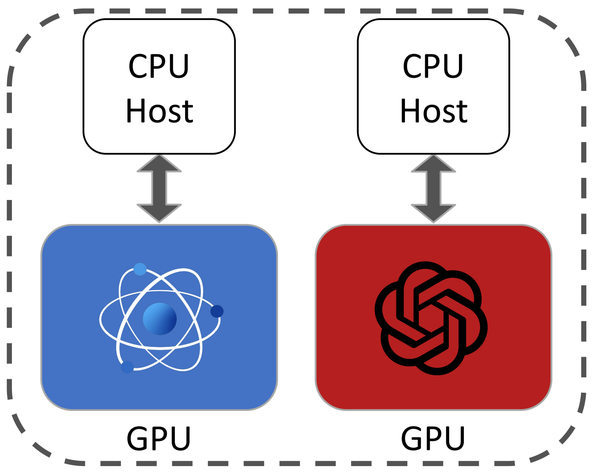
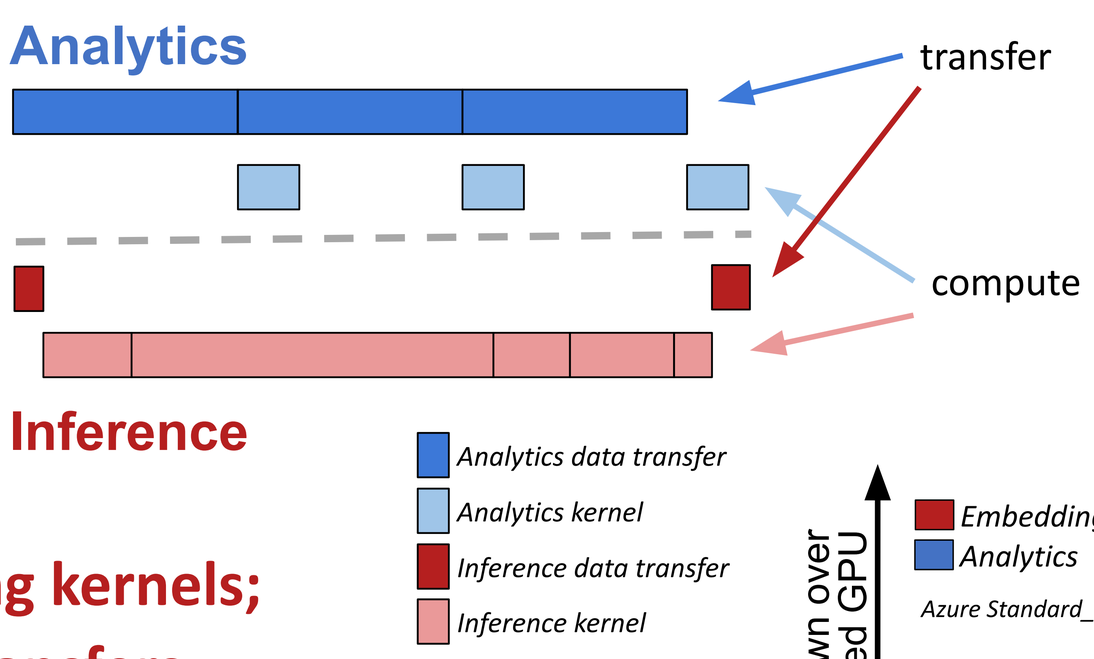
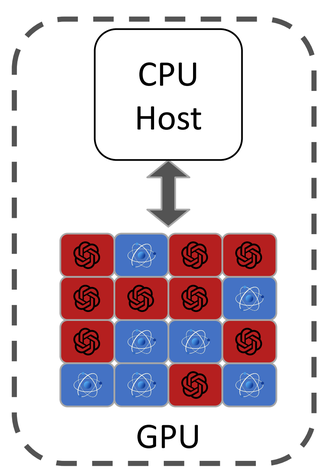
# Dava Movement-Aware GPU Sharing for Data-Intensive Systems

Yi Jiang, Hamish Nicholson, Viktor Sanca, Anastasia Ailamaki
firstname.lastname@epfl.ch

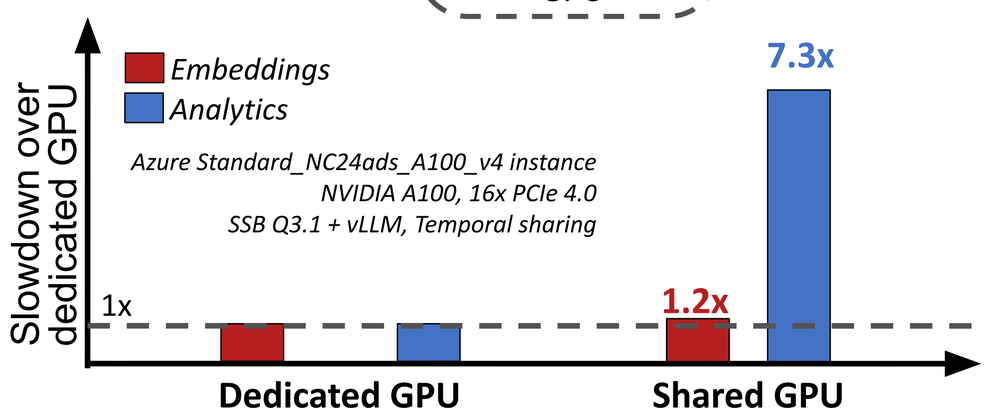
## 1. Dedicated GPUs waste compute and IC BW; sharing hurts performance



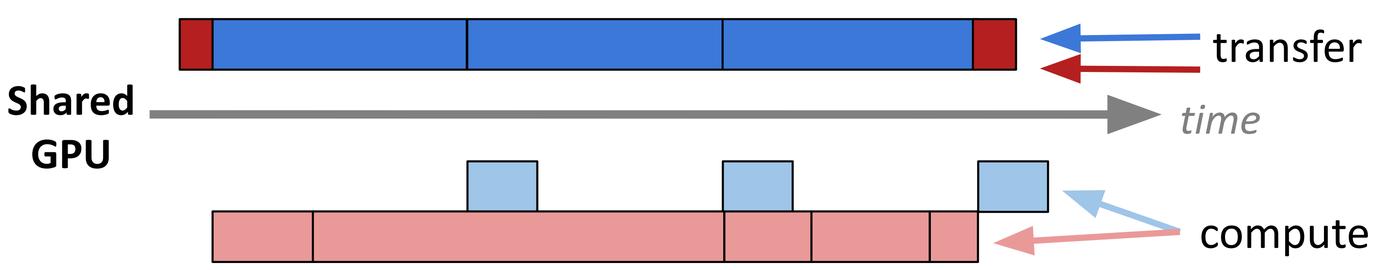




**Idle interconnect during kernels; idle compute during transfers**




## 2. Hybrid workloads have complementary resource patterns

Colocating transfer- w/ device-intensive workloads




**One workload's idle resource is another's opportunity**

## 3. The IC as a first-class resource

- Online workload characterization

$$\text{Interconnect intensity} = \frac{T_{\text{data\_transfer}}}{T_{\text{kernel}}}$$

**> 1**: *Transfer-intensive*

**≤ 1**: *Device-intensive*

A runtime measure of workload bottleneck

## 4. Dynamic GPU sharing at runtime

Monitor data transfers and GPU kernel execution

- Dynamic adaptation

*Transfer-intensive:*

*- prioritize to saturate interconnect bandwidth*

*Device-intensive:*

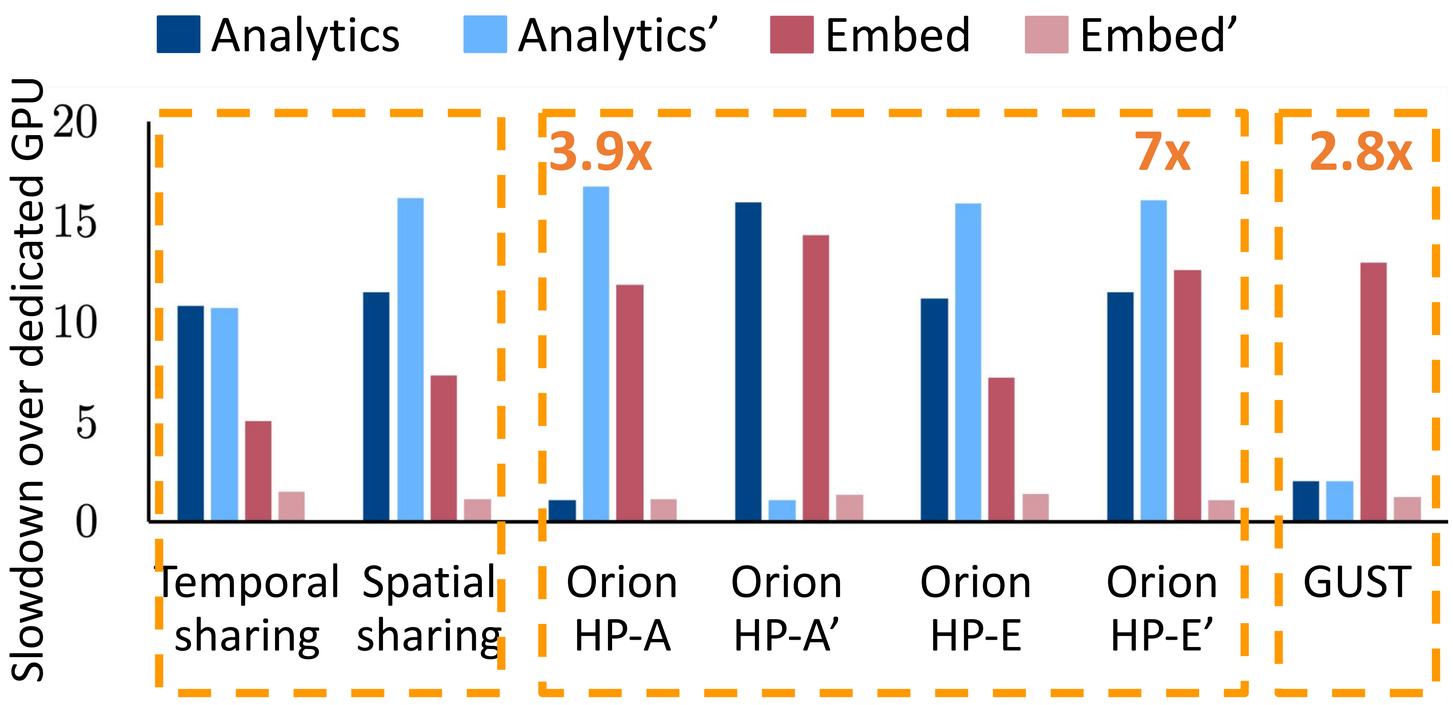
*- fill in available GPU cycles*

CUDA API call interception
- CUDA events and CUPTI:
  - memory management operations
  - kernel launch operations
- Prototype scheduler: LD_PRELOAD

**No need for workload level changes**

## 5. Evaluate GUST under high contention

*Azure Standard_NC24ads_A100_v4 instance*
*NVIDIA A100, 16x PCIe 4.0*
***SSB Q1.1 + Q3.1 + Embed Small + Embed Large***




## 6. Data movement-aware GPU sharing reduces resource stalls

- GPU sharing doesn't solve underutilization
- Interconnect intensity enables dynamic adaptation
- GUST improves slowdown relative to dedicated GPUs from 3.9-7x to 2.8x

DATA-INTENSIVE APPLICATIONS AND SYSTEMS

EPFL