Efficient Multi-GPU Shared Memory via Automatic Optimization of Fine-Grained Transfers

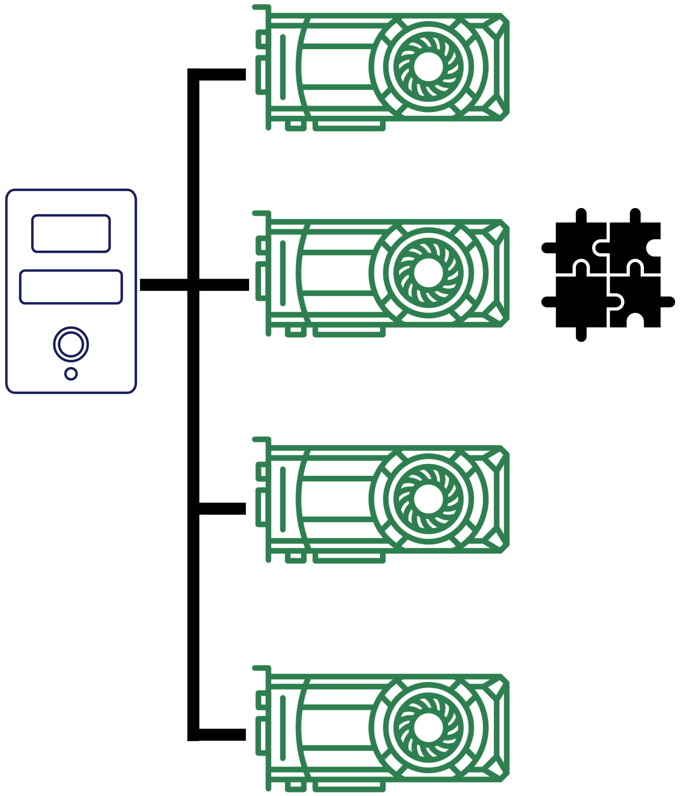
International Symposium on Computer Architecture (ISCA) 2021

June 14-19, 2021

Harini Muthukrishnan (University of Michigan), David Nellans (NVIDIA), Daniel Lustig (NVIDIA), Jeffrey Fessler (University of Michigan), Thomas Wenisch (University of Michigan)

Strong Scaling in Many GPU Systems

- Multi-GPU systems popular for highly parallel applications
- Compute phases scale reasonably well
- But communication often dominates application runtime



Inter-GPU communication forms the primary bottleneck for multi-GPU strong scaling

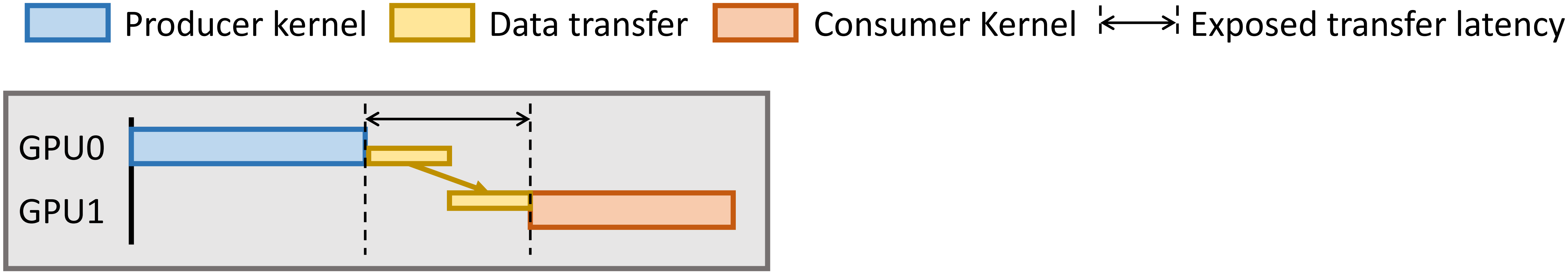
Contributions

- Explore a new paradigm for multi-GPU orchestration of shared data
  - Rely on proactive, decoupled fine-grained transfers
  - Better suited for strong scaling than traditional inter-GPU communication mechanisms
- Design PROACT: a joint compile and runtime system to auto-optimize fine-grained transfers
- Perform a comprehensive scalability study across GPU/interconnect architectures

Achieves 11x mean performance improvement on a 16-GPU system

5.3x better than the next best programming paradigm

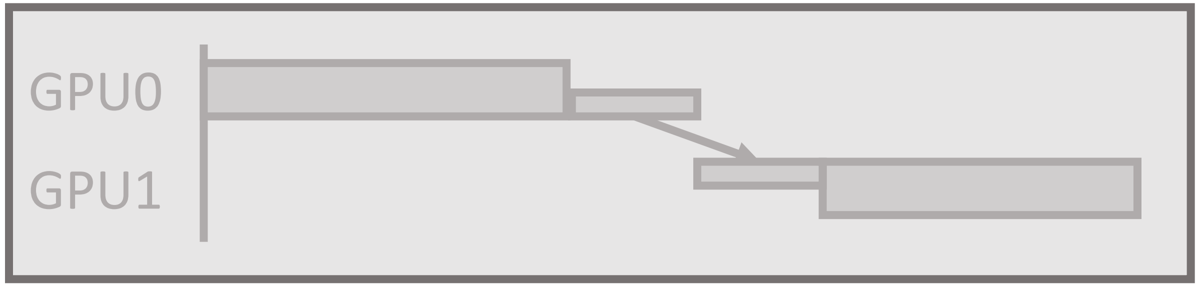
Multi-GPU Programming Challenges



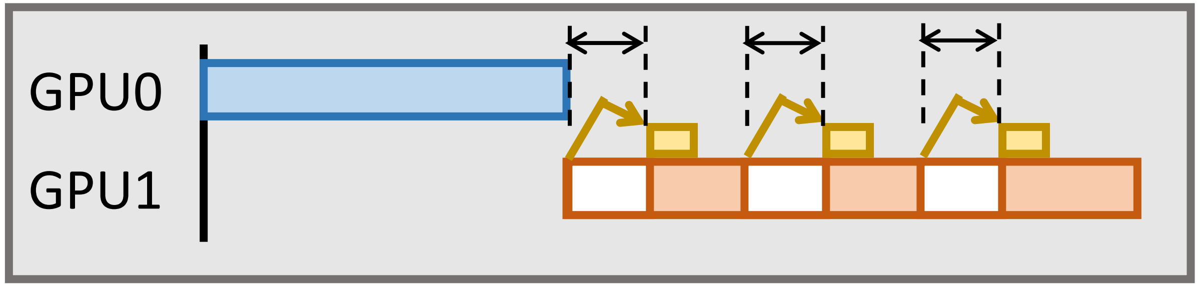
Bulk DMA (cudaMemcpy) exposes transfer latency

Multi-GPU Programming Challenges

Producer kernel   Data transfer   Consumer Kernel   Exposed transfer latency



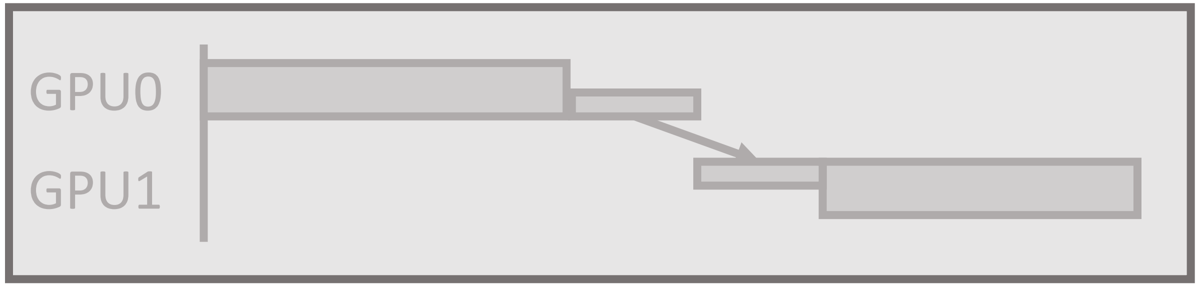
Bulk DMA (cudaMemcpy) exposes transfer latency



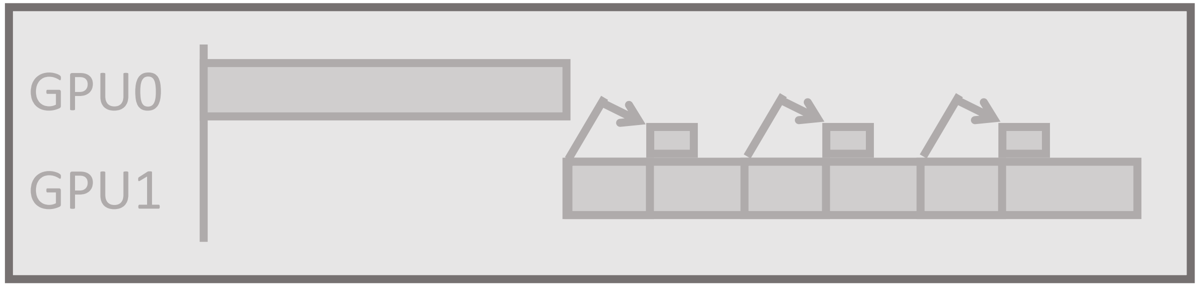
Peer-to-peer loads expose remote load latency

Multi-GPU Programming Challenges

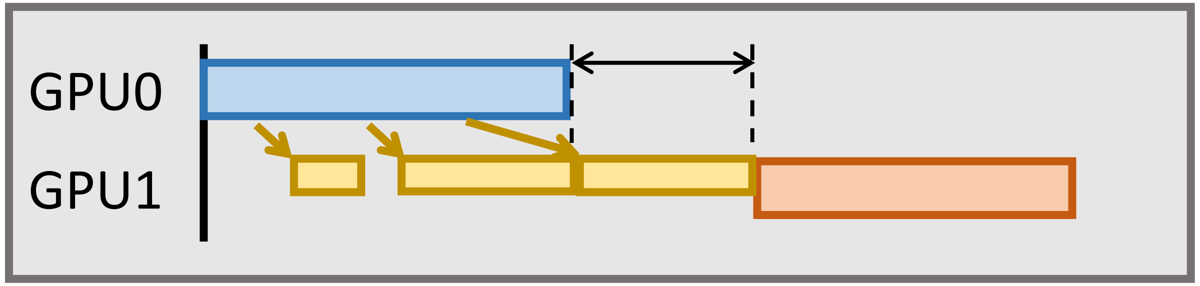
Producer kernel   Data transfer   Consumer Kernel   Exposed transfer latency



Bulk DMA (cudaMemcpy) exposes transfer latency



Peer-to-peer loads expose remote load latency

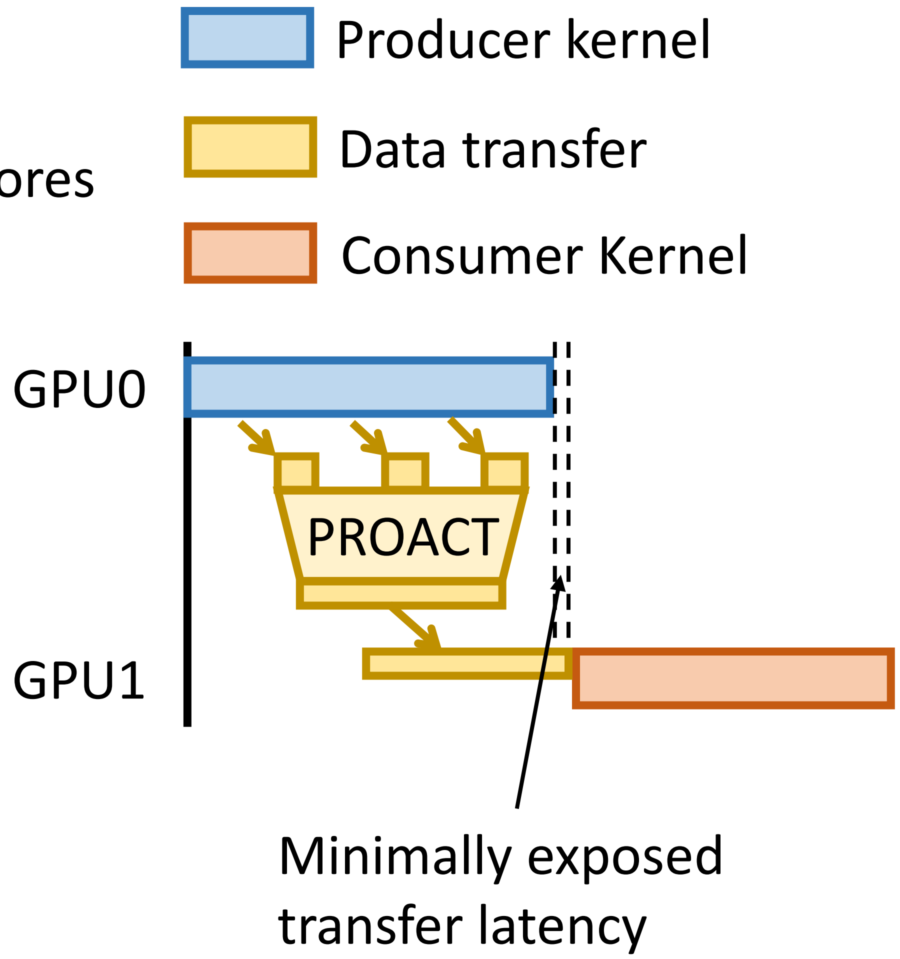


Peer-to-peer stores result in inefficient interconnect utilization

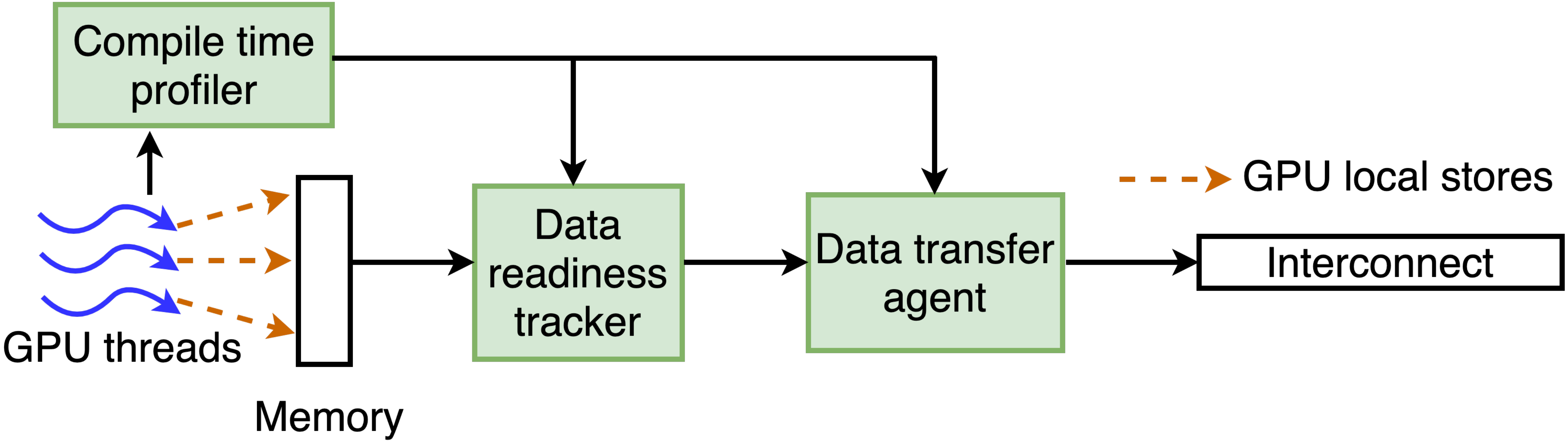
PROACT for Multi-GPU Systems

- Proactive and optimized decoupled transfers
  - Producers push data to consumers with fine-grained stores

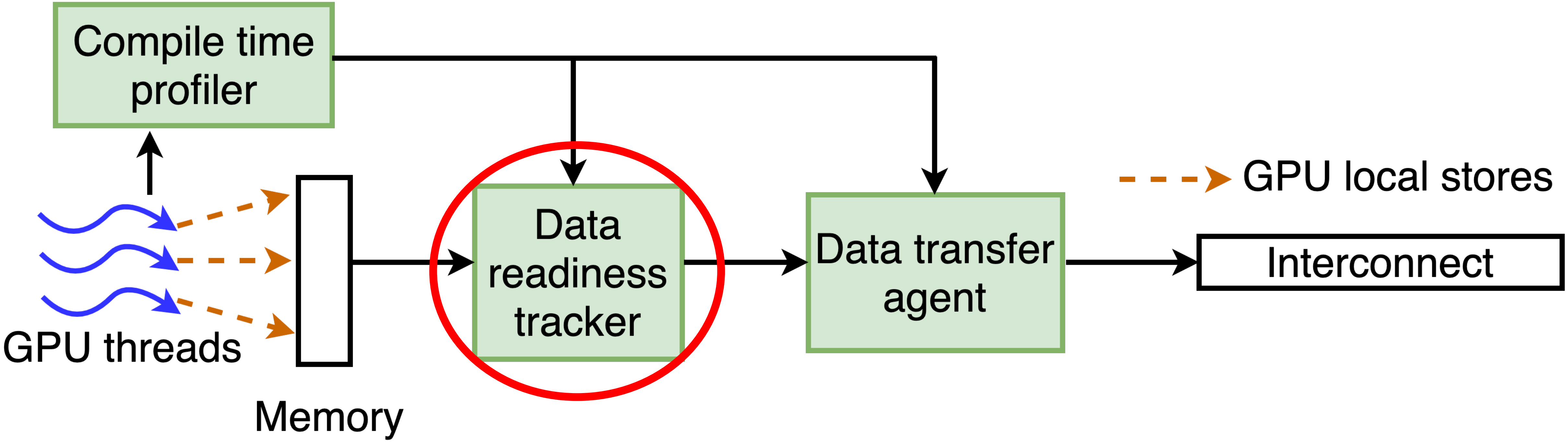
- Profile guided selection of transfer mechanism, granularity, and resource allocation
  - Balances overlap of data transfers with computation
  - Maximizes opportunity for write coalescing
  - Implementation aggregates programmatic fine-grained transfers into interconnect-efficient large transfers



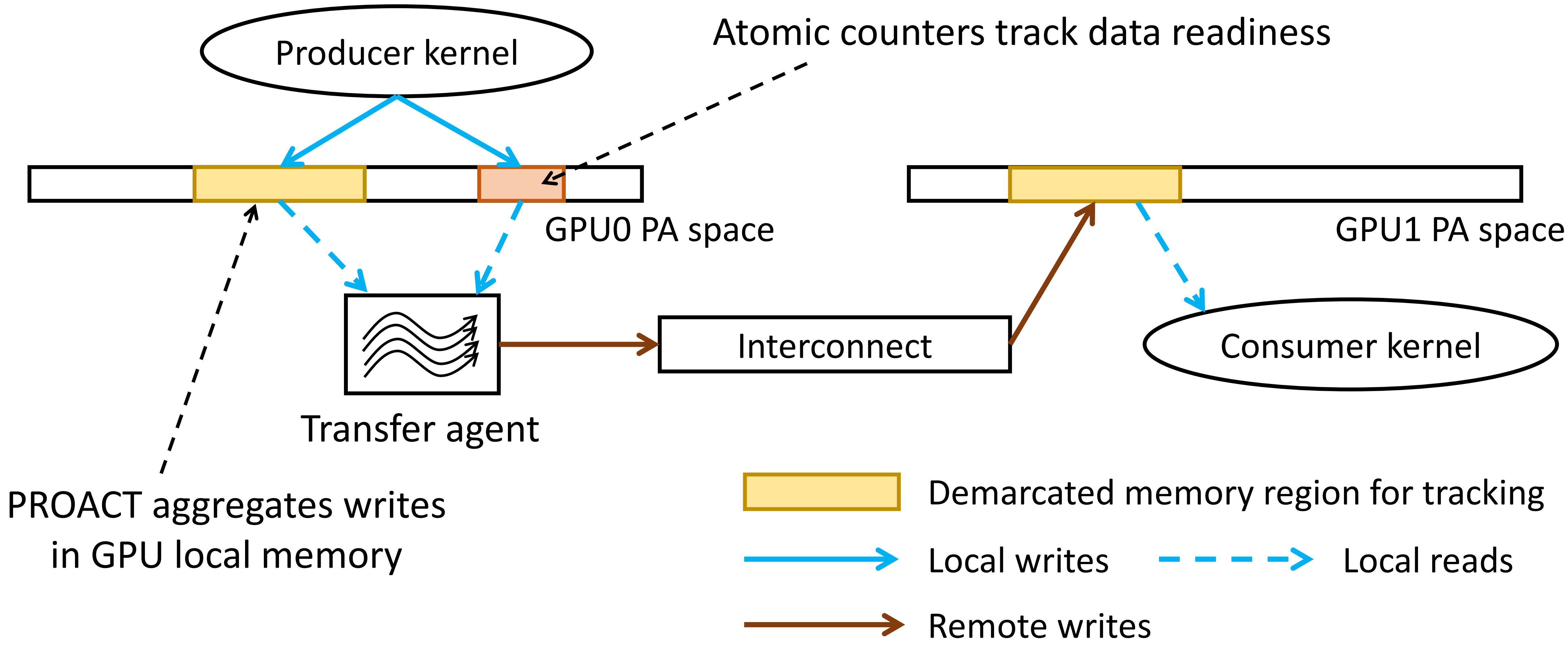
PROACT Design Overview



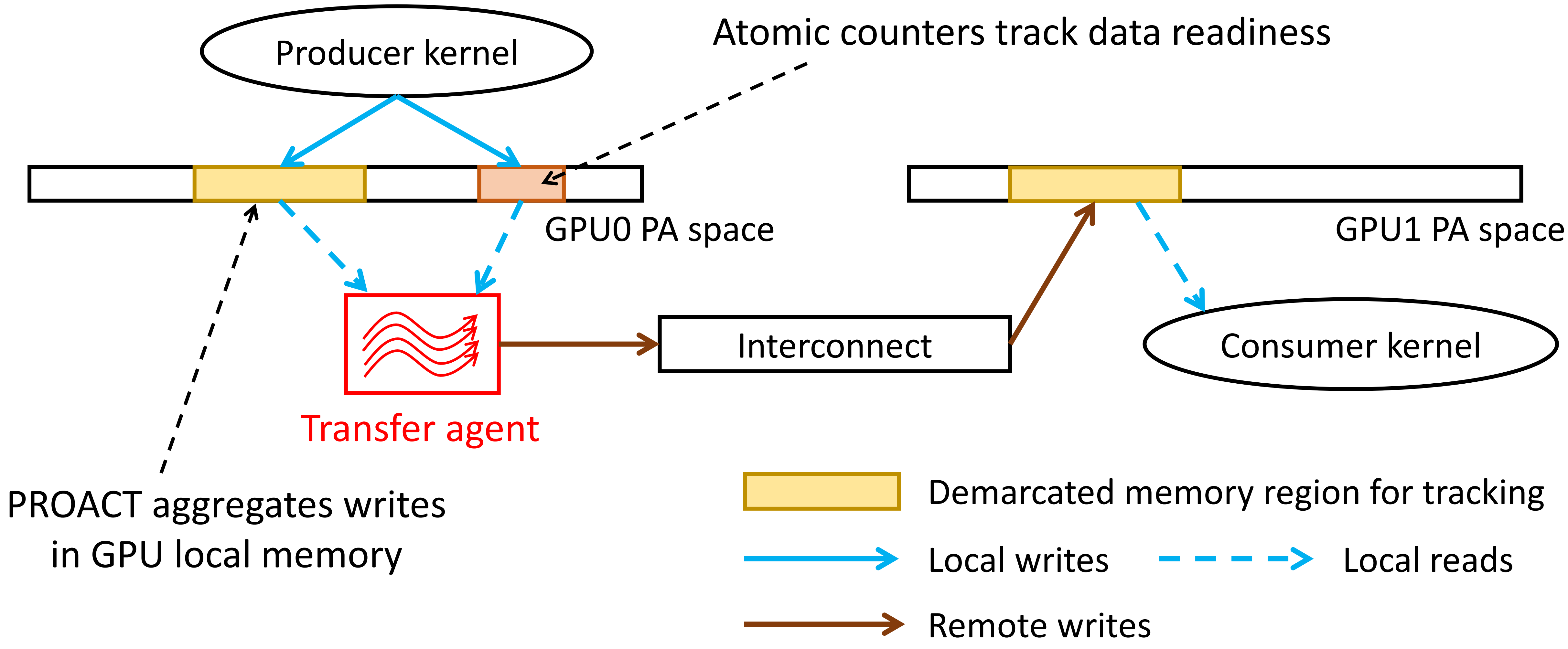
Tracking Data Transfer Readiness



Tracking Data Transfer Readiness



Tracking Data Transfer Readiness

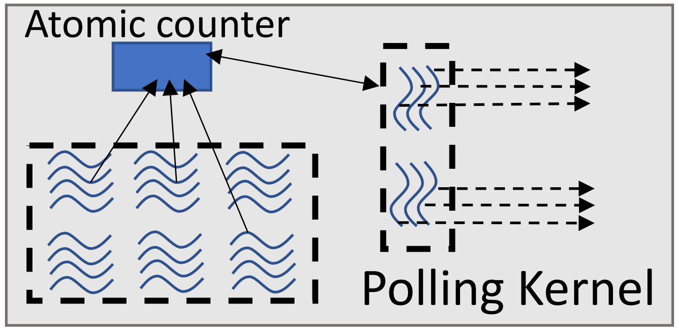


Transfer Agent: Mechanisms

<u>Decoupled transfers</u>

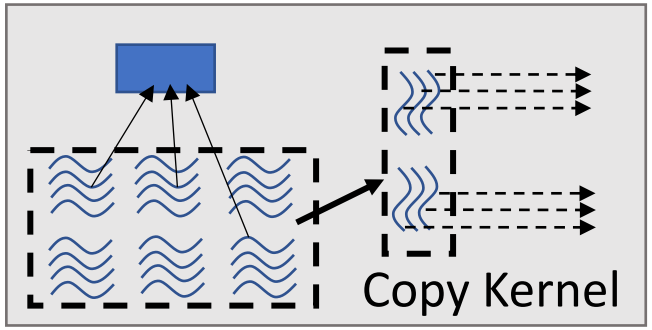
<u>Inline transfers</u>

Polling

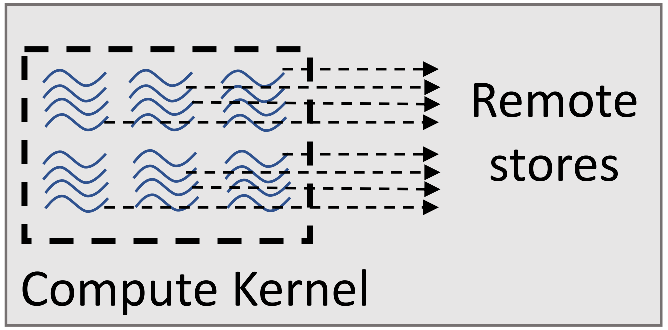


+ Decoupled transfers
- Compete with execution

CUDA Dynamic Parallelism



+ No polling overhead
- High initiation latency



+ No tracking overhead
- Poor interconnect efficiency

PROACT compiler profiler identifies the best transfer mechanism for each application and system architecture

PROACT Implementation Options

- PROACT components can be prototyped in HW or SW
  - HW: automatic update of counters to regions and triggering of transfers
  - SW: CUDA library support to proxy HW functionalities with minimal programmer overhead
- Overheads of SW implementation (see paper for details)
  - Combined effect of 15% increase in compute kernel time on average
  - Benefits of PROACT exceed SW overheads

We implement a SW prototype to evaluate PROACT across four different GPU and interconnect generations

Systems Under Study

GPU Architectures

Kepler

Pascal

Volta

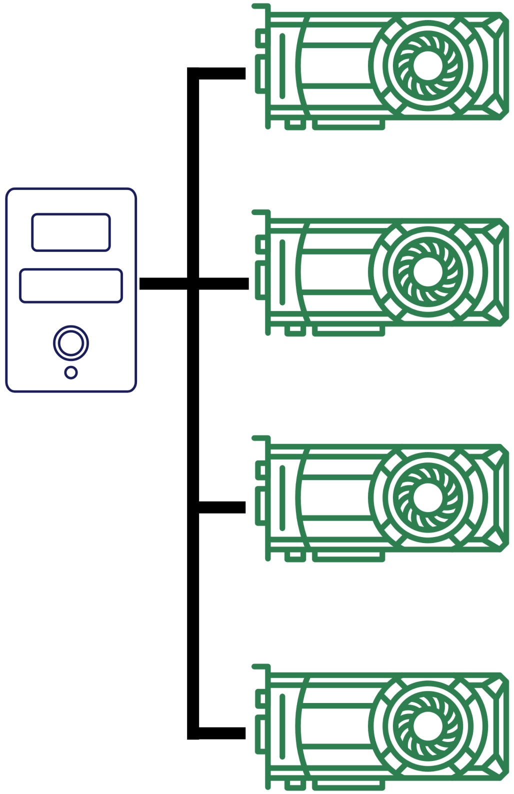
Interconnects

PCIe3.0

NVLink

NVLink2

NVSwitch



Workloads From

Scientific computing

Medical Imaging

Graph Processing

Recommender systems

Number of GPUs

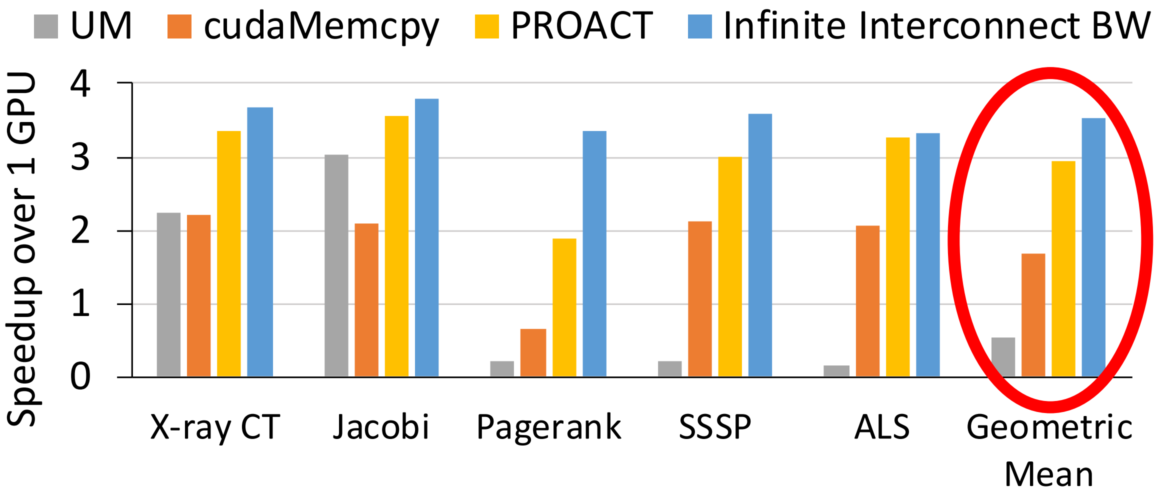
1-16

Programming Techniques Compared

Technique	Description
Unified Memory	UM with hand-coded cudaMemAdvise hints readMostly, preferredLocation, AccessedBy
cudaMemcpy	cudaMemcpy only at kernel boundaries
PROACT	**Proactive decoupled fine-grained transfers**
Infinite Interconnect BW	Advantage of fine-grained copies excluding all data transfer overheads (calculated)

Results: 4-GPU Volta System

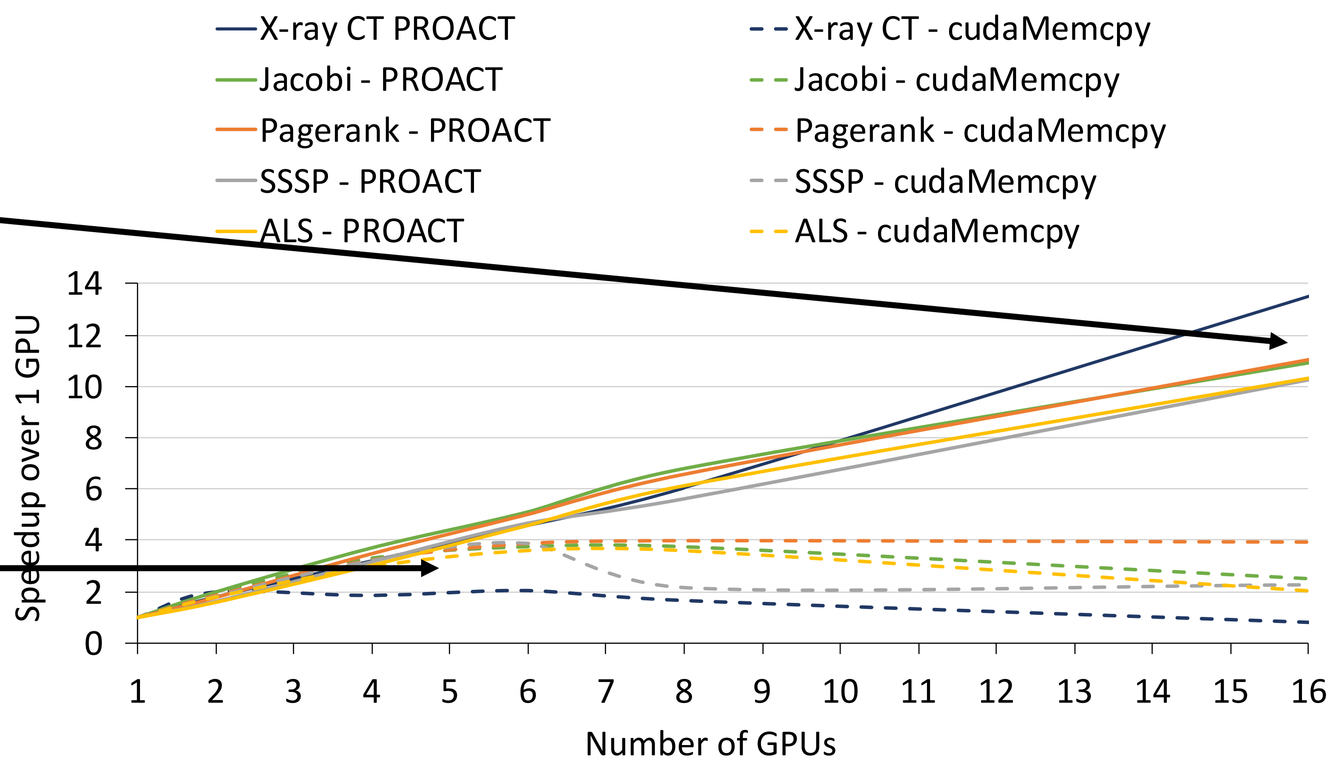
- UM performs poorly: expensive page faults
- cudaMemcpy: exposed transfer latency hurts
- PROACT outperforms UM and cudaMemcpy
  - Performs better than coarse-grained page level data movement
  - Achieves fine-grained compute-communication overlap and minimally exposes transfer latency



PROACT enables 2.9x speedup capturing 83% of the available opportunity

Scalability on 16-GPU Volta System

- PROACT achieves:
  - 11x mean speedup over 1 GPU
  - 77% of available opportunity
  - 5.3x better scaling than cudaMemcpy
- cudaMemcpy only scales to 5 GPUs
  - Exposed communication then dominates application runtime



*UM omitted due to poor scaling

PROACT achieves scalable multi-GPU performance by enabling efficient decoupled fine-grained transfers between GPUs

Summary

- Existing multi-GPU communication paradigms scale poorly:
  - Exposed communication time
  - Poor interconnect utilization over program life
- Proactive stores between GPUs is the most effective way to overlap computation and communication, but suffers from poor interconnect utilization
- PROACT provides intelligent data orchestration and improves interconnect utilization
  - SW library prototype evaluated over 4 different multi-GPU systems
  - Achieves 2.9x and 11x over a single GPU on a 4 and 16 GPU system respectively
  - Provides a new pathway to multi-GPU performance scalability in the future