PystachIO: Efficient Distributed GPU Query Processing with PyTorch over Fast Networks & Fast Storage

VLDB 2026

Jigao Luo[*], Nils Boeschen[*], Muhammad El-Hindi, Carsten Binnig

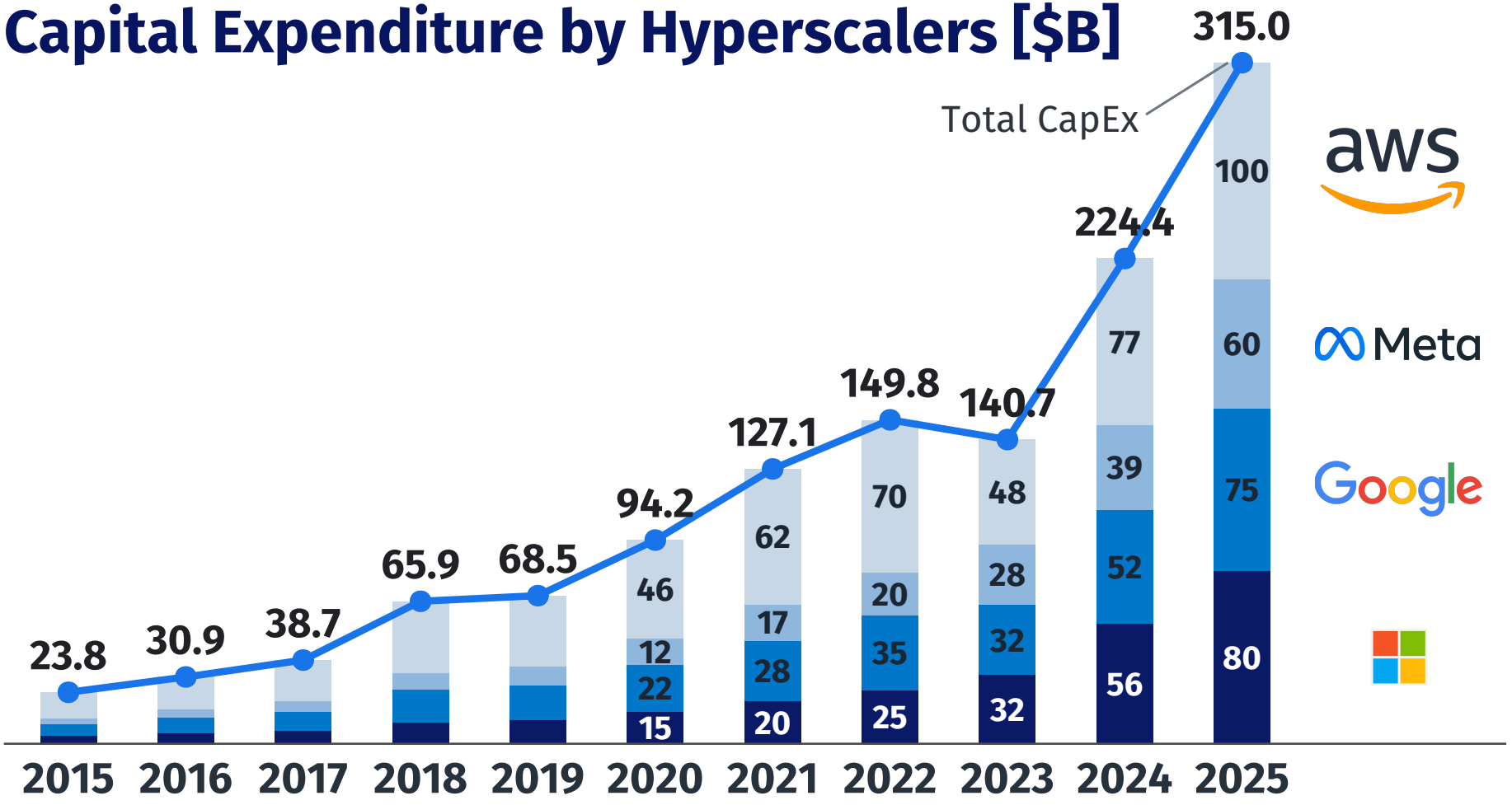
[*]: Equal contribution
Systems Group, Technische Universität Darmstadt

TECHNISCHE UNIVERSITÄT DARMSTADT

SYSTEMS

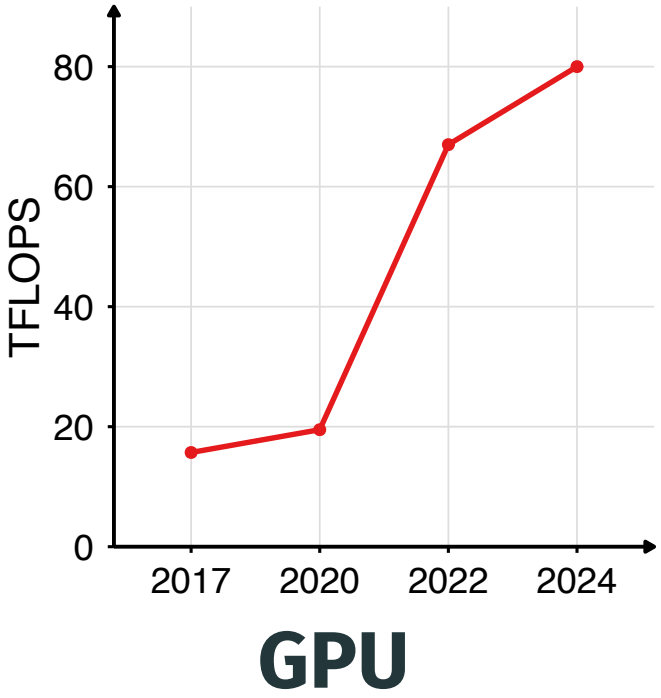
AI Investment Is Shaping the Future

Capital Expenditure by Hyperscalers [$B]

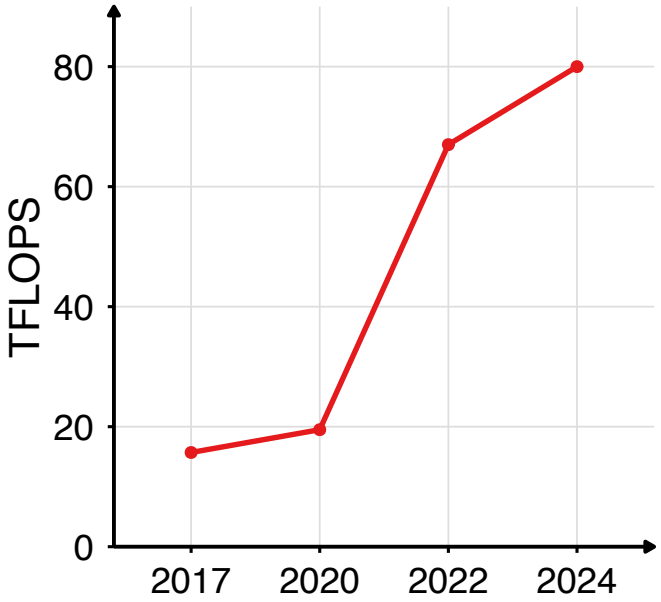


Source: DC Pulse

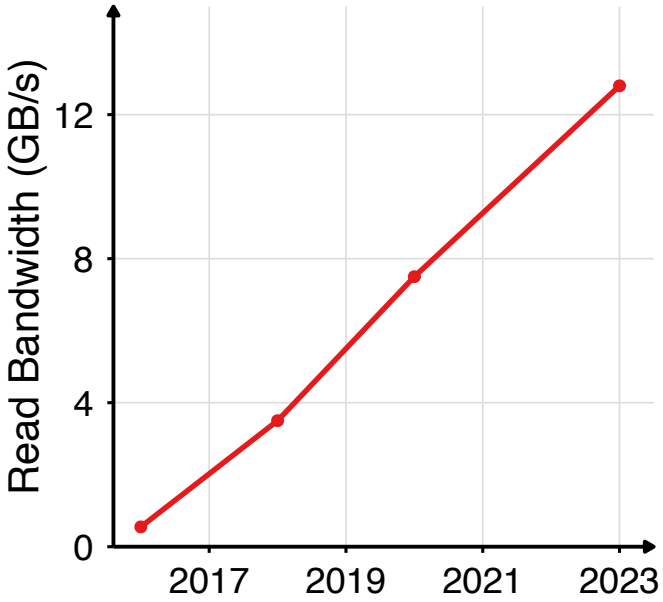
AI Investment Is Shaping the Future HW & DC



AI Investment Is Shaping the Future HW & DC

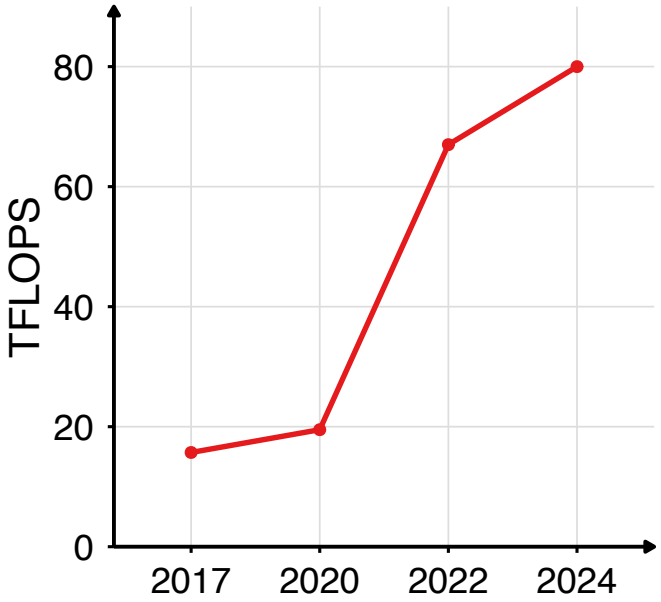


GPU

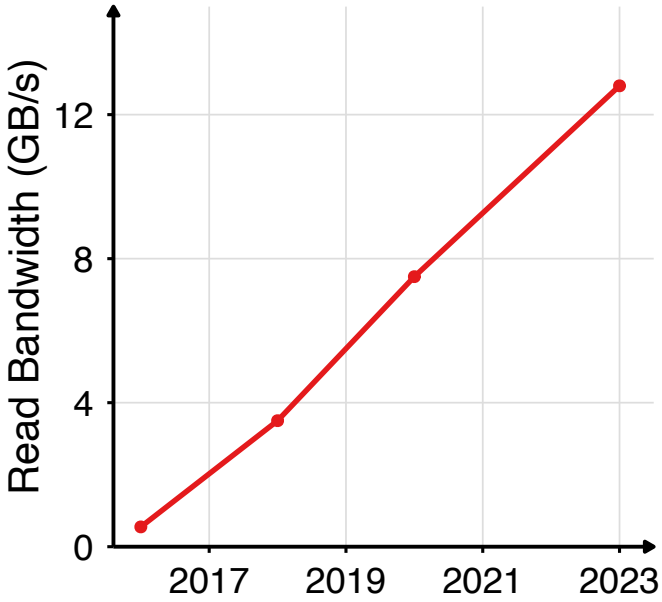


Storage: NVMe SSD

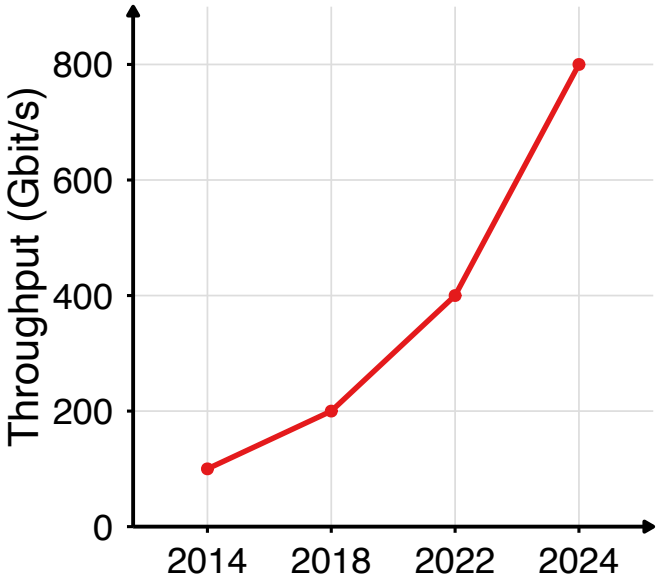
AI Investment Is Shaping the Future HW & DC



GPU



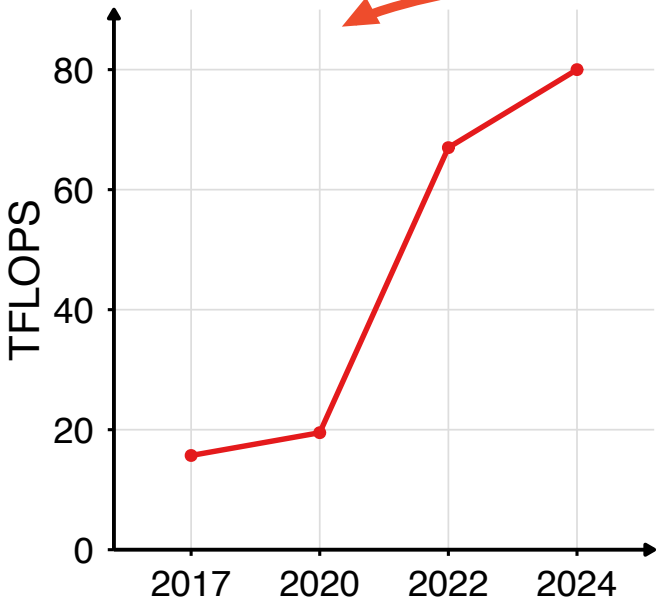
Storage: NVMe SSD



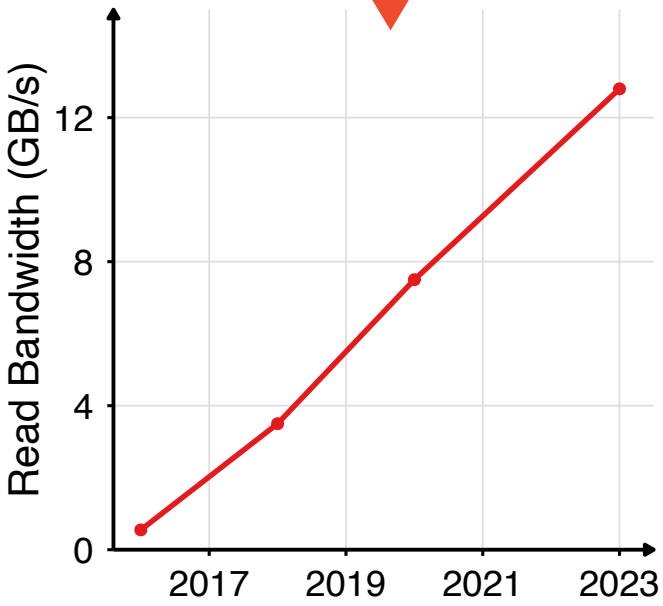
Network: RDMA NIC

AI Investment Is Shaping the Future HW & DC

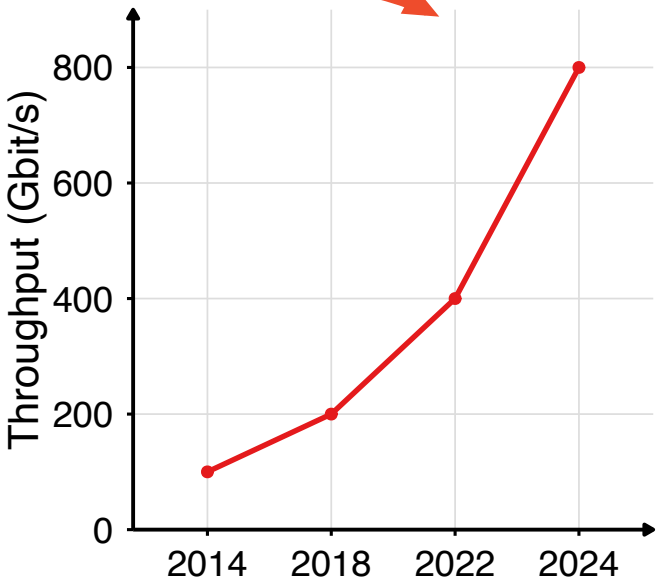
PyTorch



GPU



Storage:
NVMe SSD



Network:
RDMA NIC

Build a DB on PyTorch?

Query Processing on Tensor Computation Runtimes

Dong He[1], Supun Nakandala[2], Dalitso Banda[3], Rathijit Sen[3], Karla Saur[3], Kwanghyun Park[3],
Carlo Curino[3], Jesús Camacho-Rodríguez[3], Konstantinos Karanasos[4], Matteo Interlandi[3]
[1]University of Washington, [2]University of California, San Diego, [3]Microsoft, [4]Meta
[1]donghe@cs.washington.edu, [2]snakanda@eng.ucsd.edu, [3]firstname.lastname@microsoft.com, [4]kkaranasos@fb.com

VLDB 22

What is missing?

	PyTorch
GPU	✓
Storage	✓
Network	✓

What is missing?

	PyTorch	TQP
GPU	✓	✓
Storage	✓	✗
Network	✓	✗

What is missing?

	PyTorch	TQP	Cloud OLAP
GPU	✓	✓	✗
Storage	✓	✗	✓
Network	✓	✗	✓

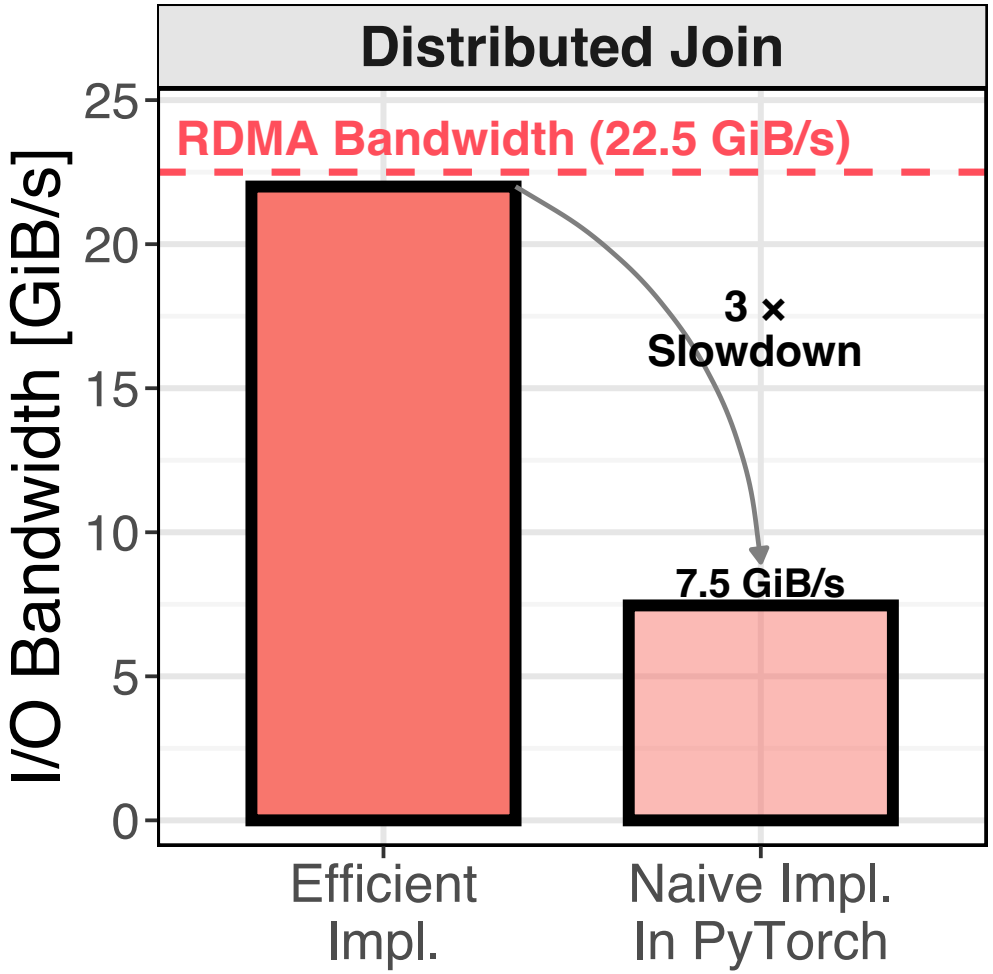
What is missing?

	PyTorch	TQP	Cloud OLAP
GP			
Stor			
Network	✓	✗	✓

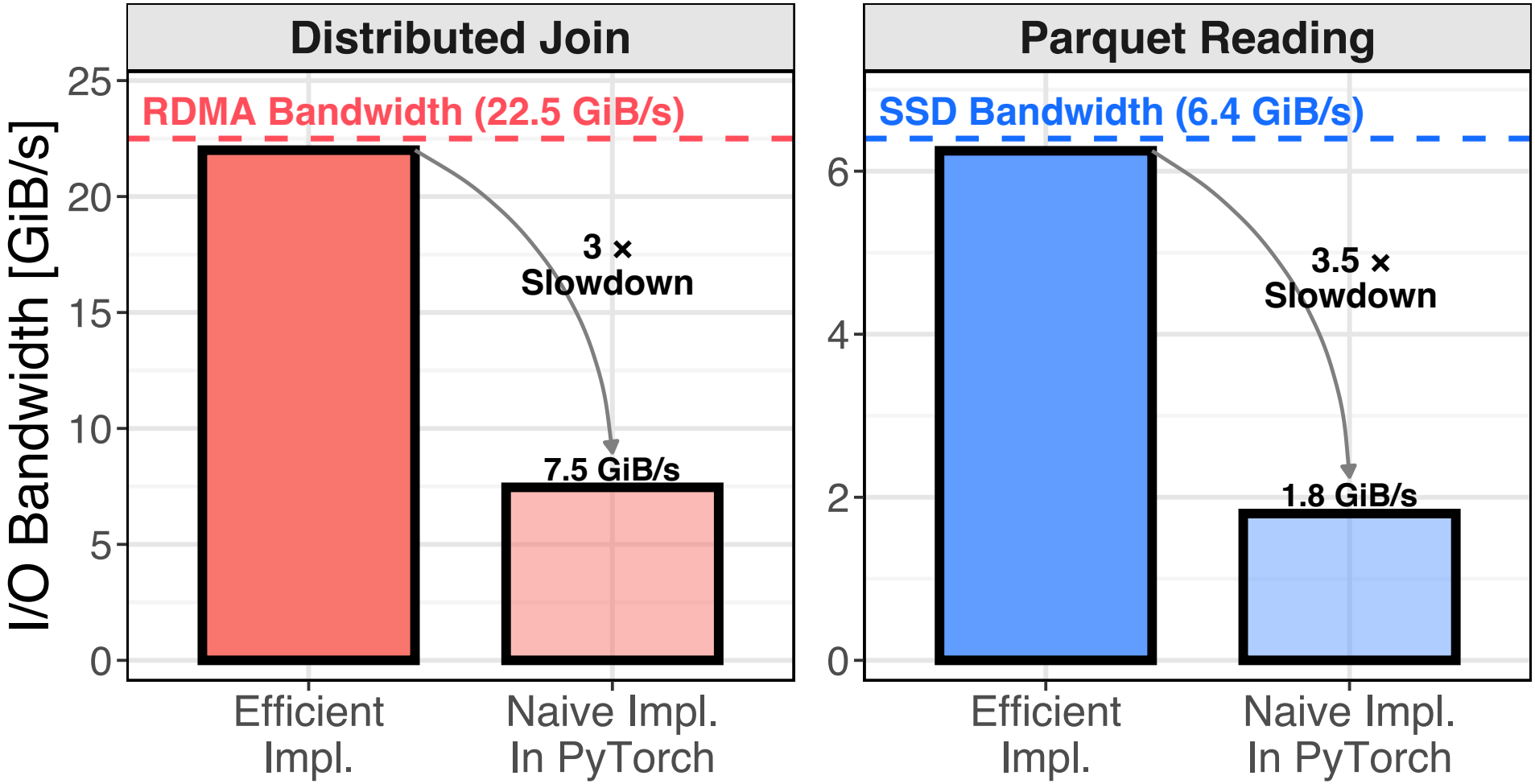
How to build a distributed GPU DB on PyTorch?

Challenge: Network and Storage I/O

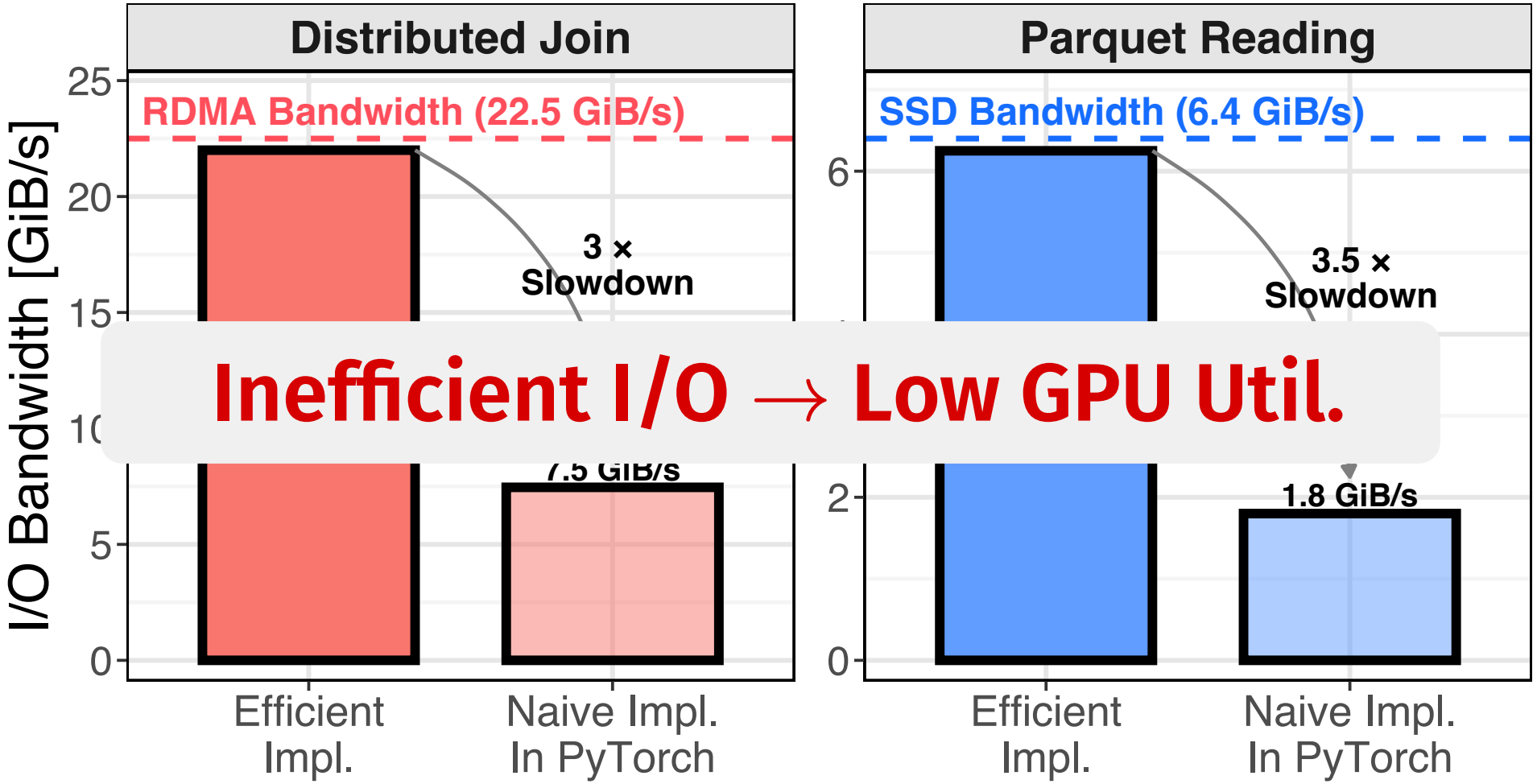
Challenge: Network and Storage I/O



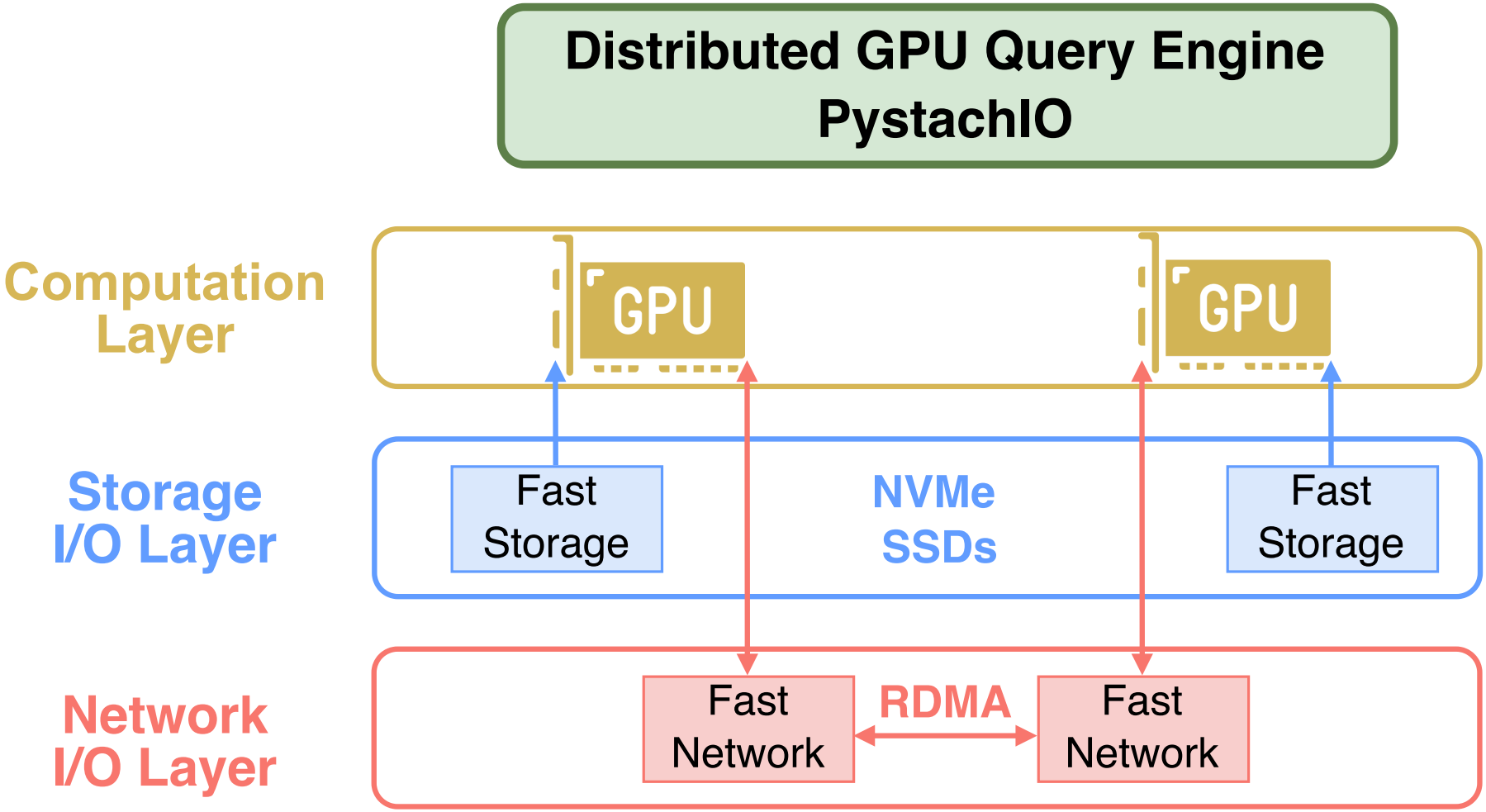
Challenge: Network and Storage I/O



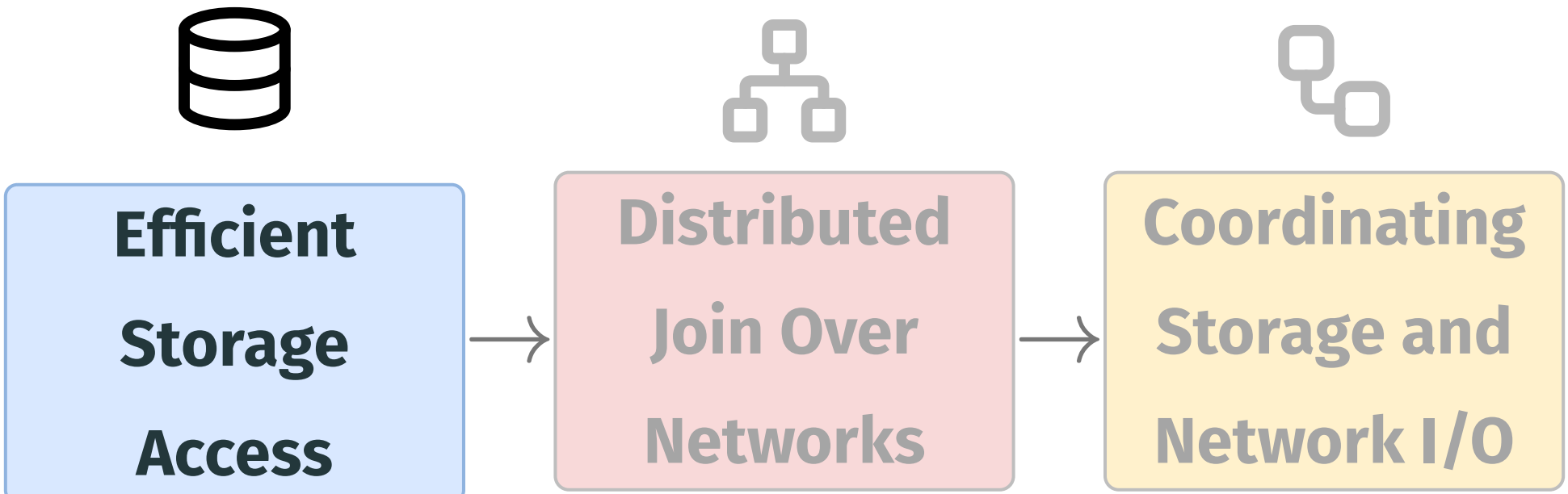
Challenge: Network and Storage I/O



PystachIO

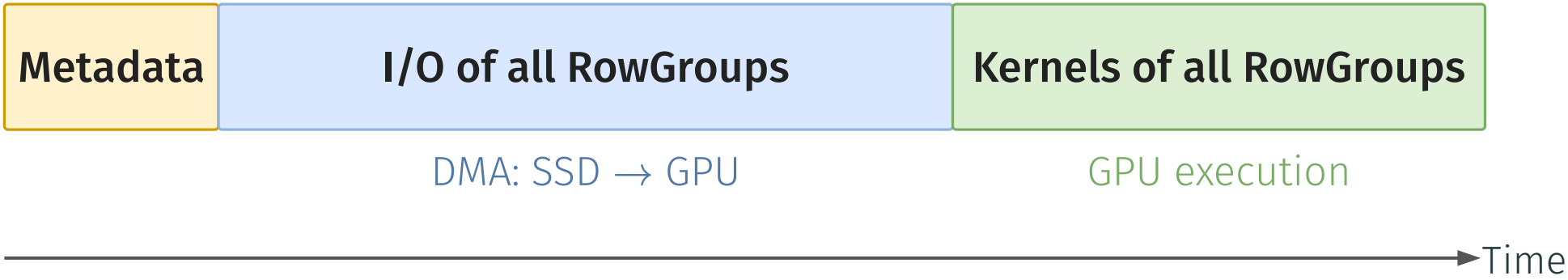

Distributed GPU Query Engine
PystachIO
Computation Layer
GPU
GPU
Storage I/O Layer
Fast Storage
NVMe SSDs
Fast Storage
Network I/O Layer
Fast Network
RDMA
Fast Network

PystachIO: Storage



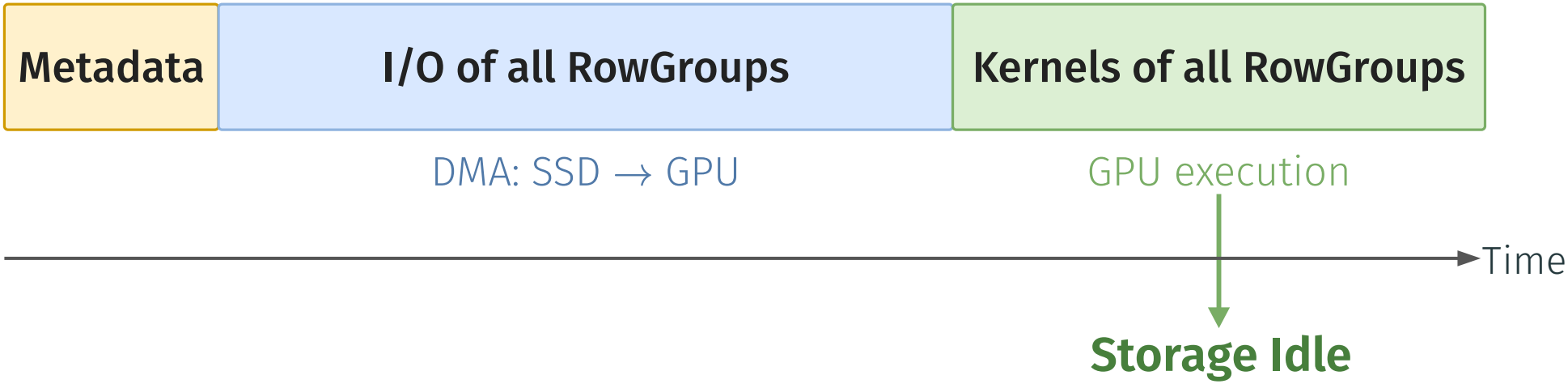
Inefficient Storage Access: Blocking Parquet Reader

`read_parquet()` in one CPU thread:



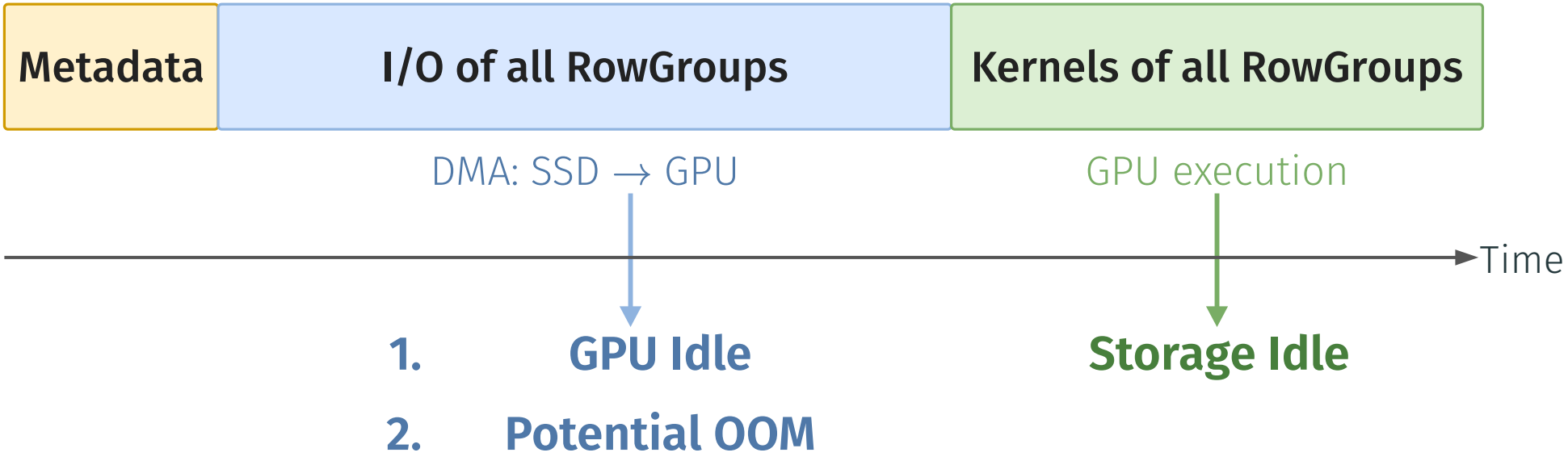
Inefficient Storage Access: Blocking Parquet Reader

`read_parquet()` in one CPU thread:

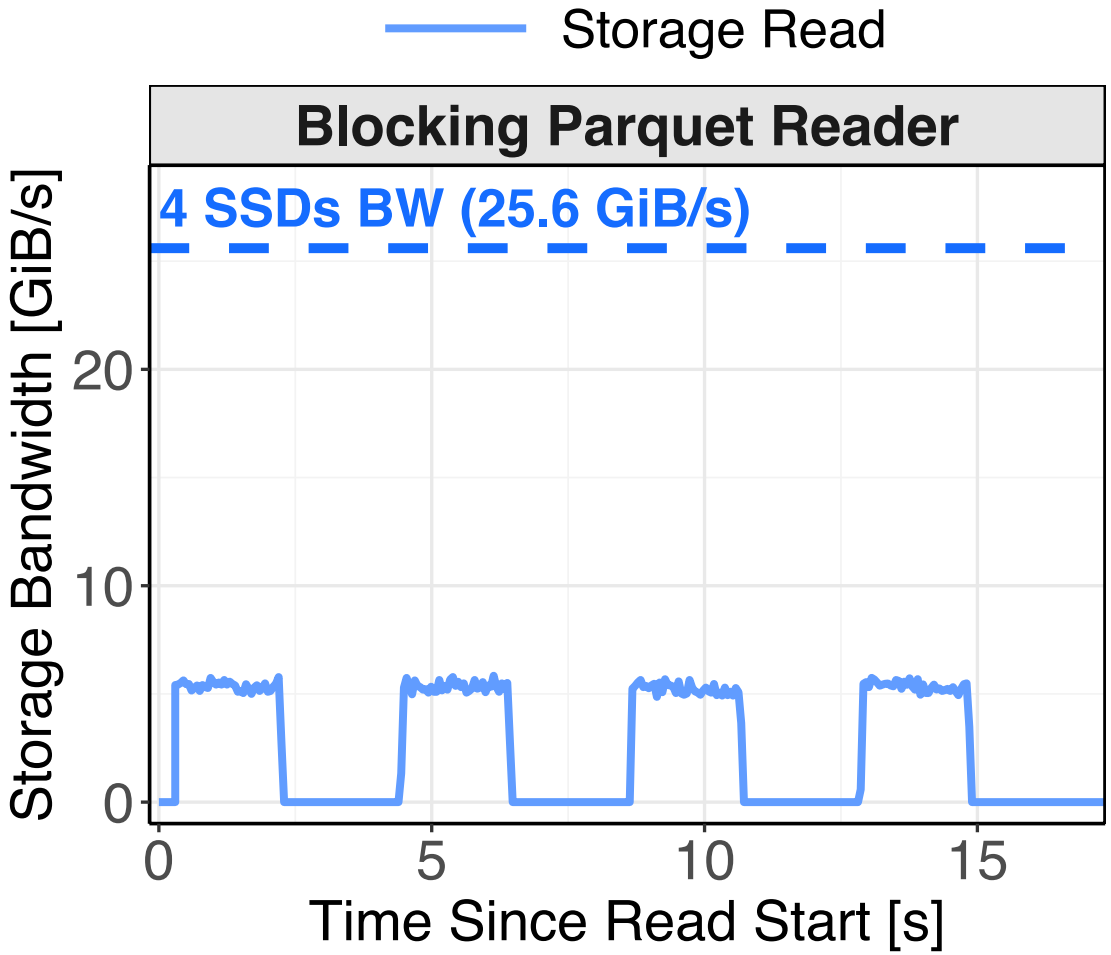


Inefficient Storage Access: Blocking Parquet Reader

`read_parquet()` in one CPU thread:

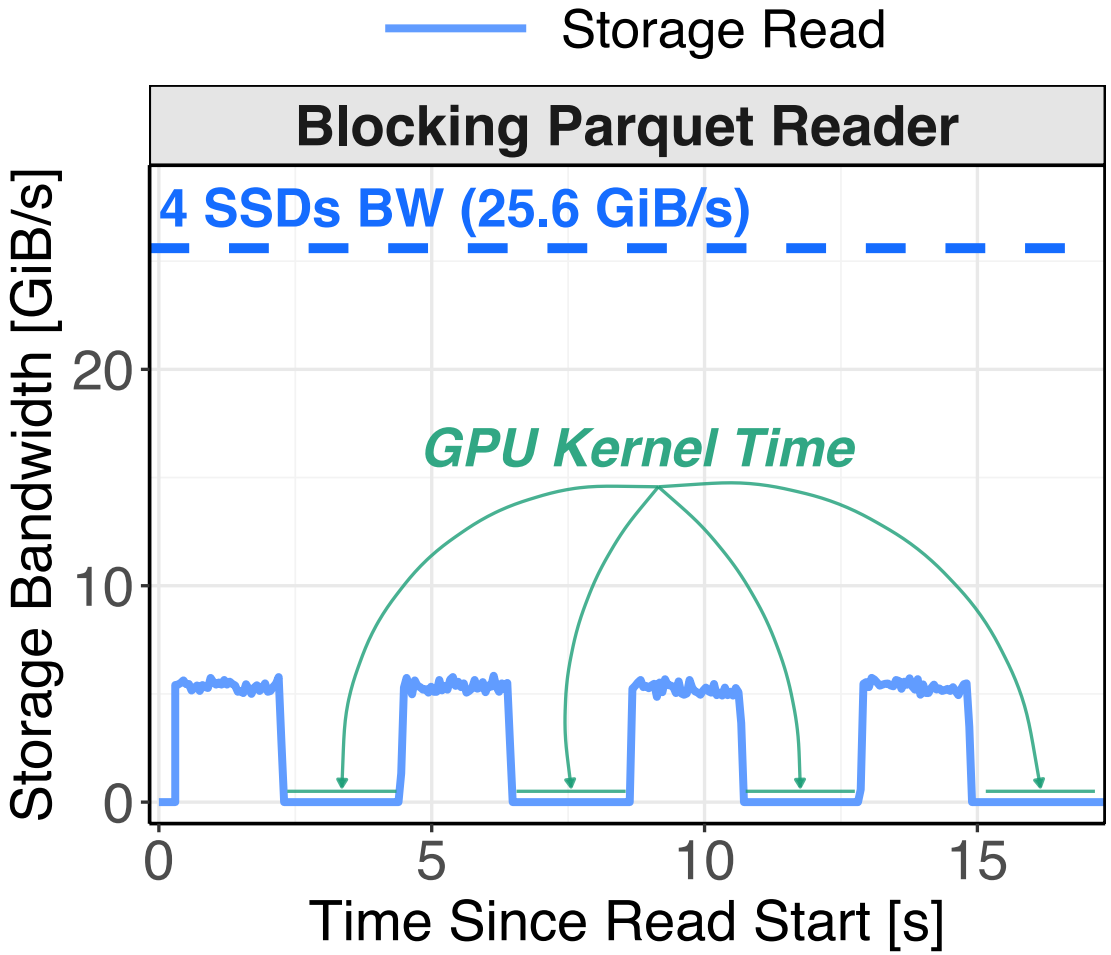


Inefficient Storage Access: Blocking Parquet Reader



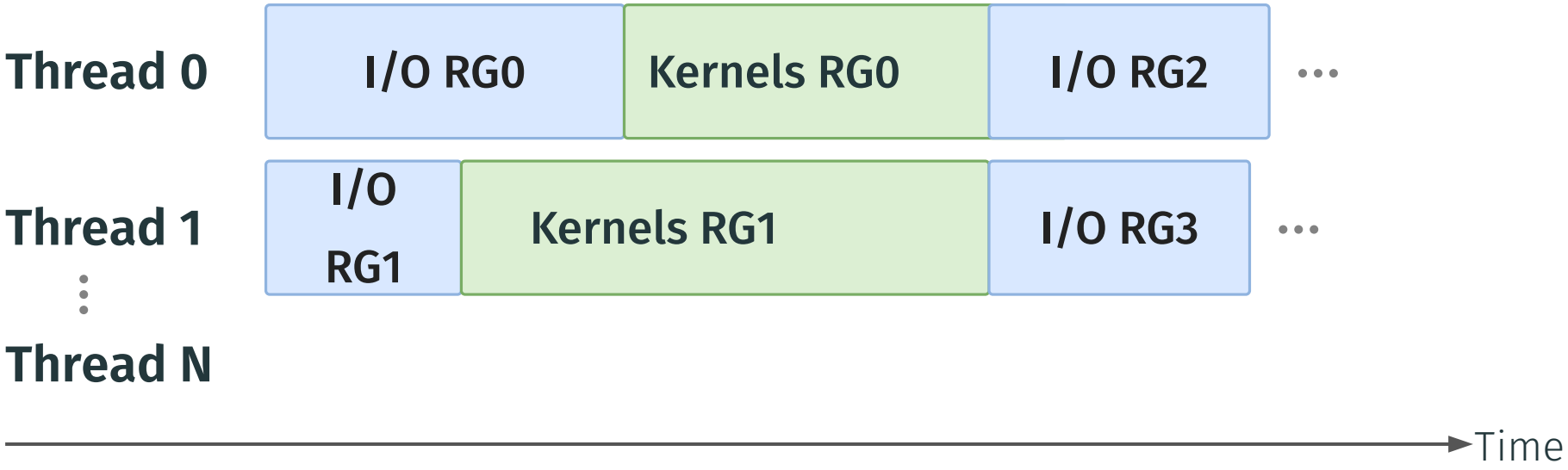
Storage microbenchmark with one GPU (A100, 40GB) and 4 NVMe SSD in PCIe 4.0.

Inefficient Storage Access: Blocking Parquet Reader

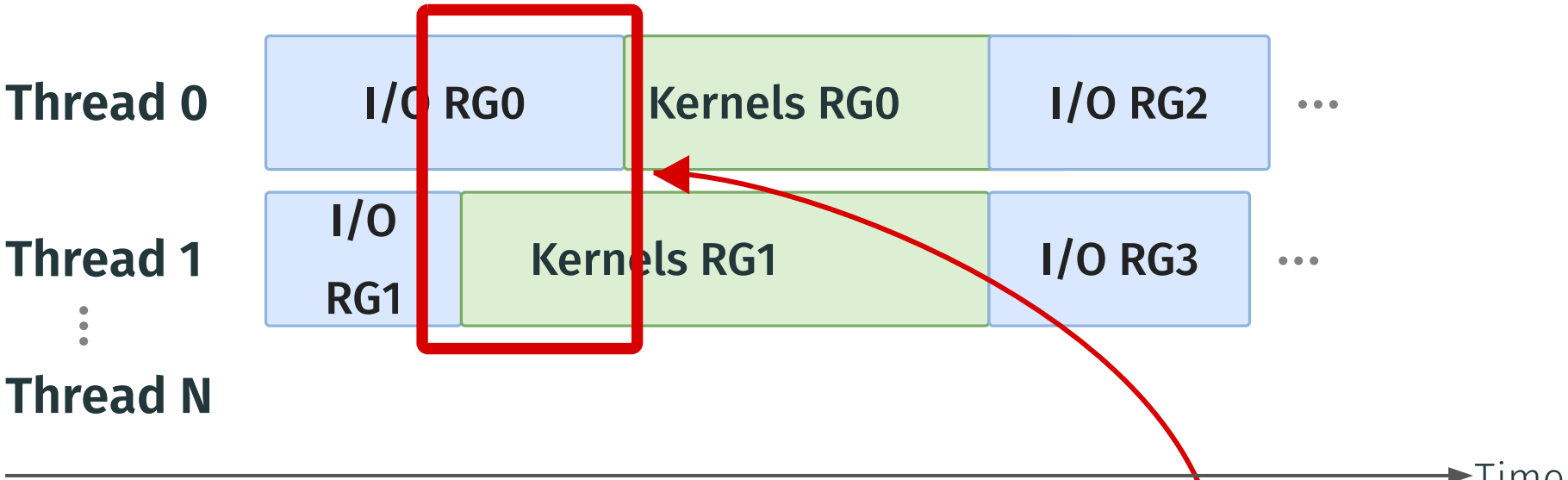


Storage microbenchmark with one GPU (A100, 40GB) and 4 NVMe SSD in PCIe 4.0.

Efficient Storage Access: Chunking

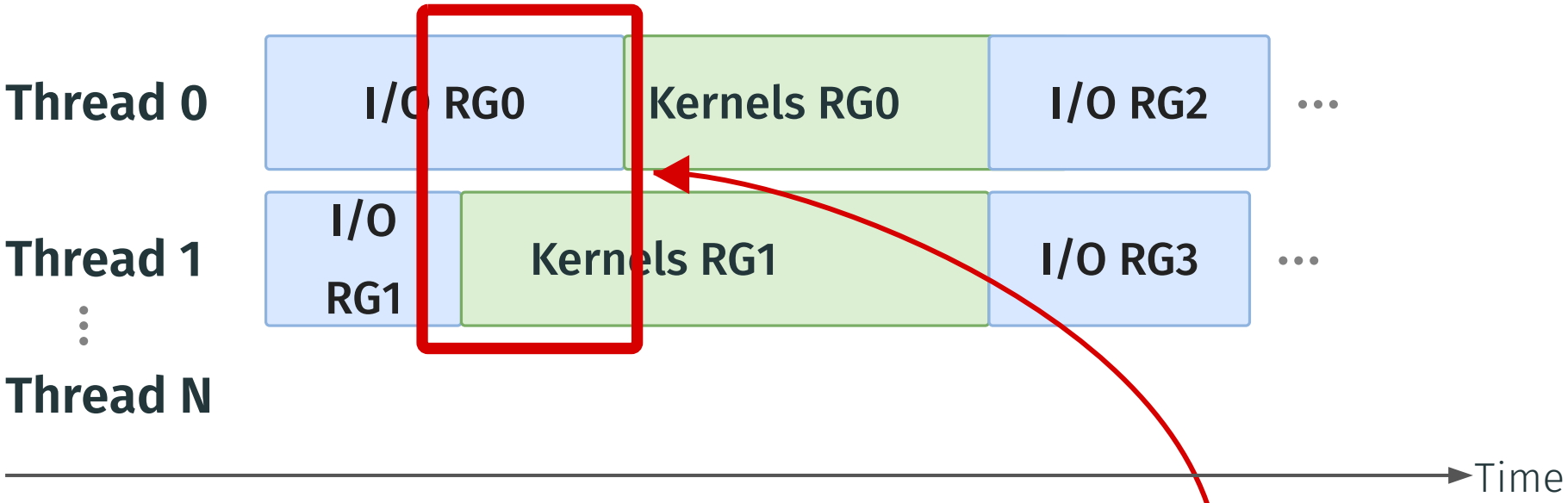


Efficient Storage Access: Chunking



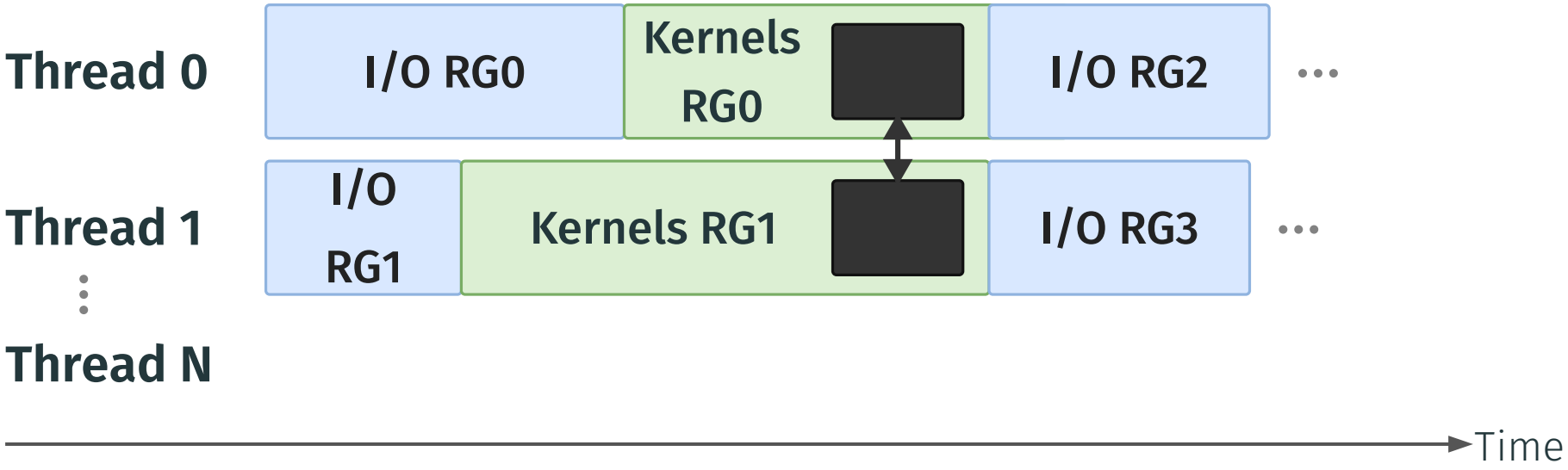
1. Overlapping I/O and Computation

Efficient Storage Access: Chunking

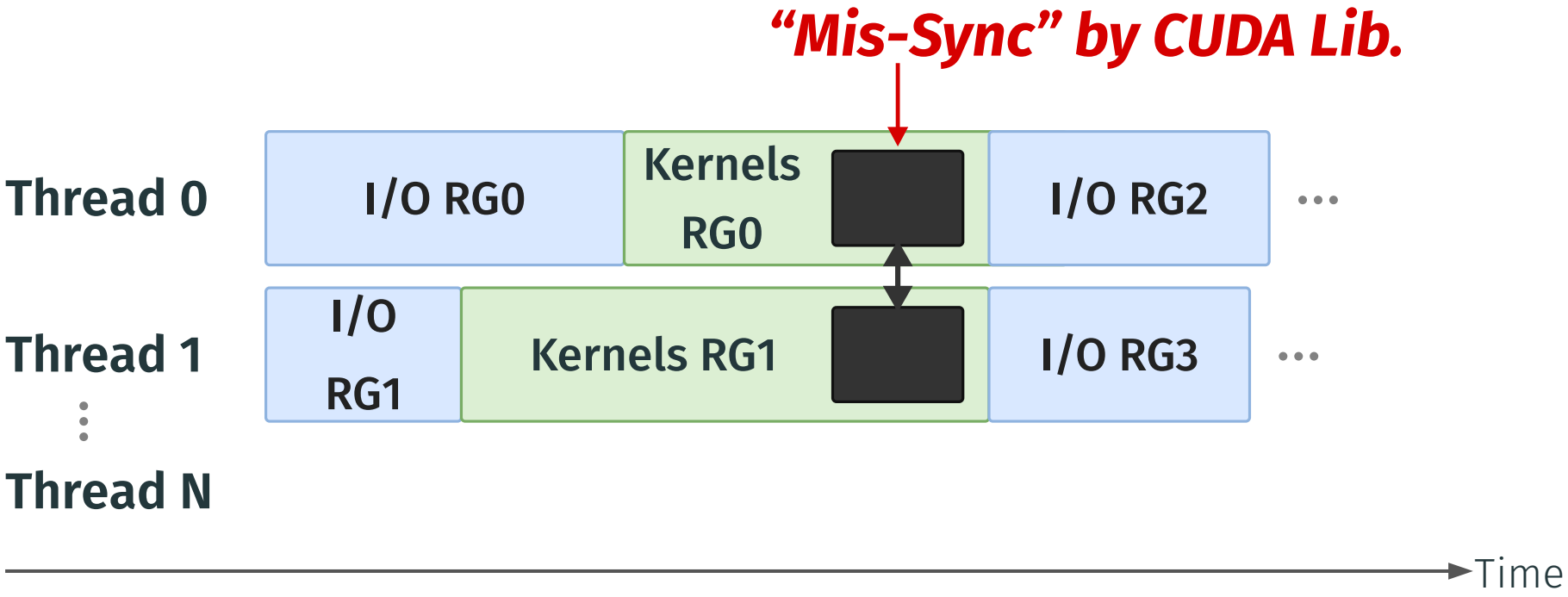


1. Overlapping I/O and Computation
2. Less Memory Pressure

Efficient Storage Access: Mis-Sync Elimination

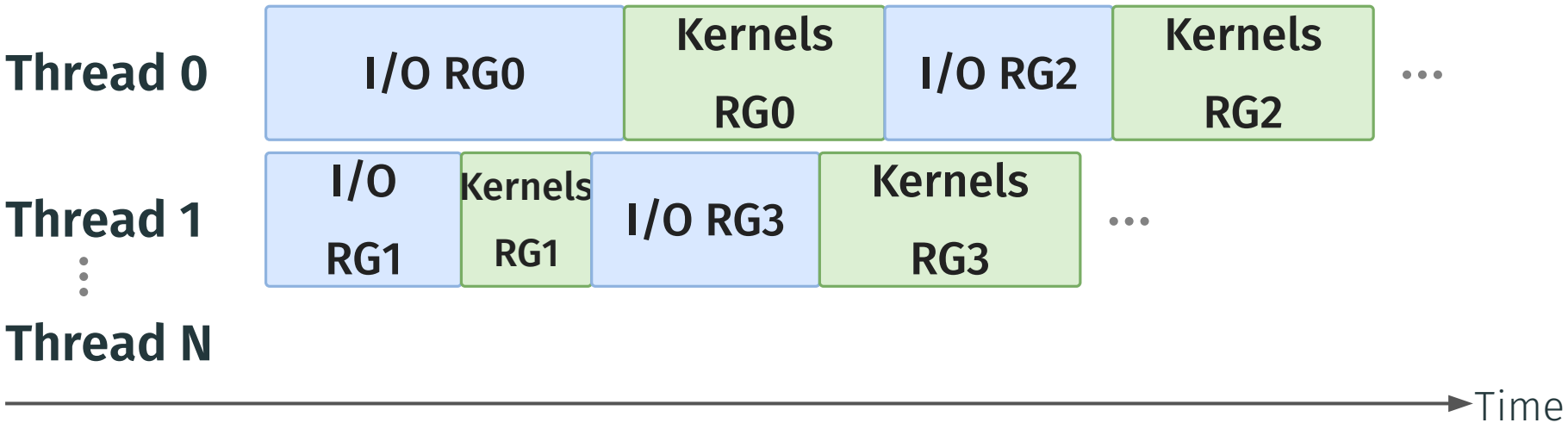


Efficient Storage Access: Mis-Sync Elimination

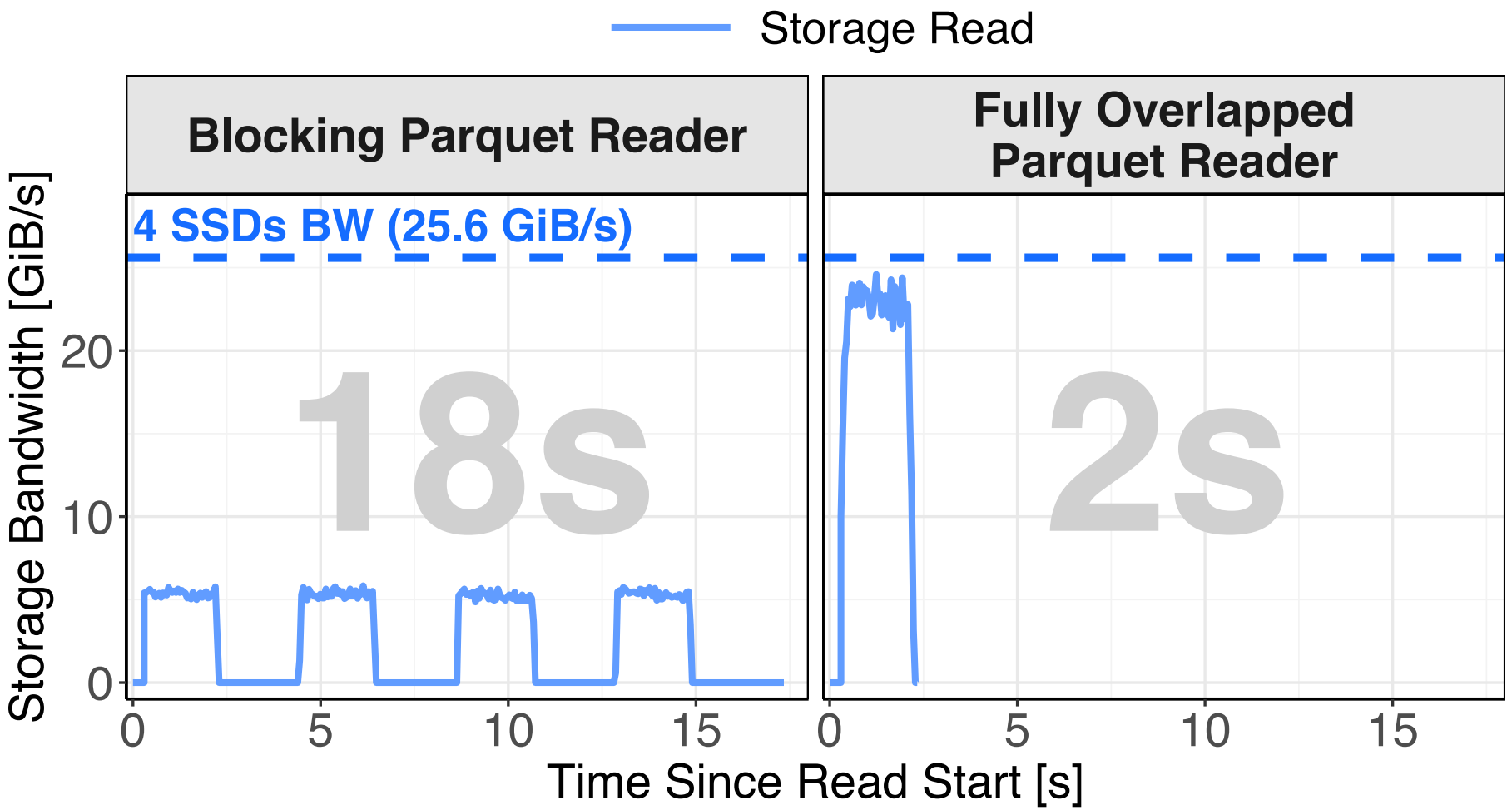


Efficient Storage Access: Mis-Sync Elimination

Fully Overlapped Parquet Reader

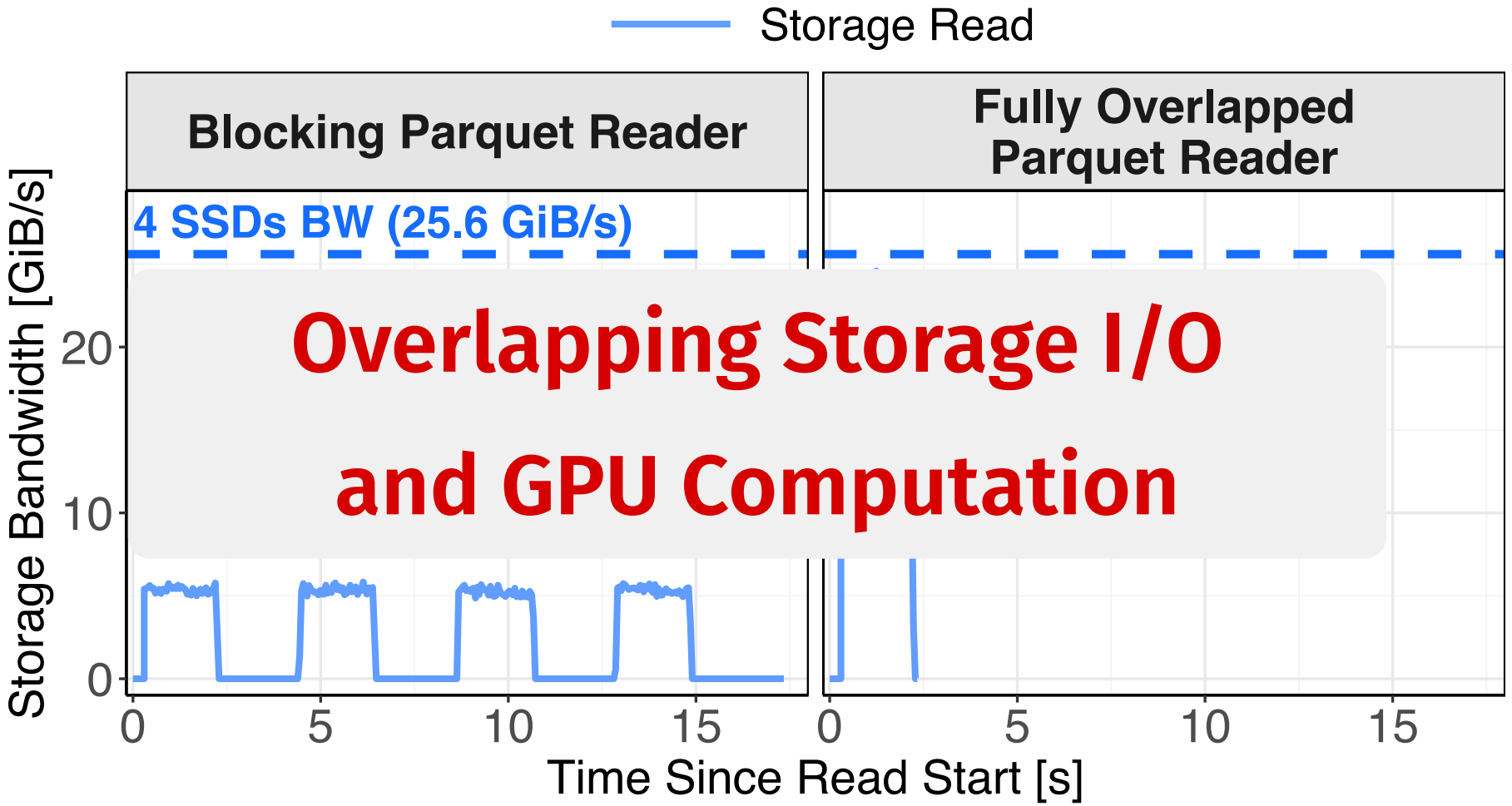


Efficient Storage Access: Mis-Sync Elimination



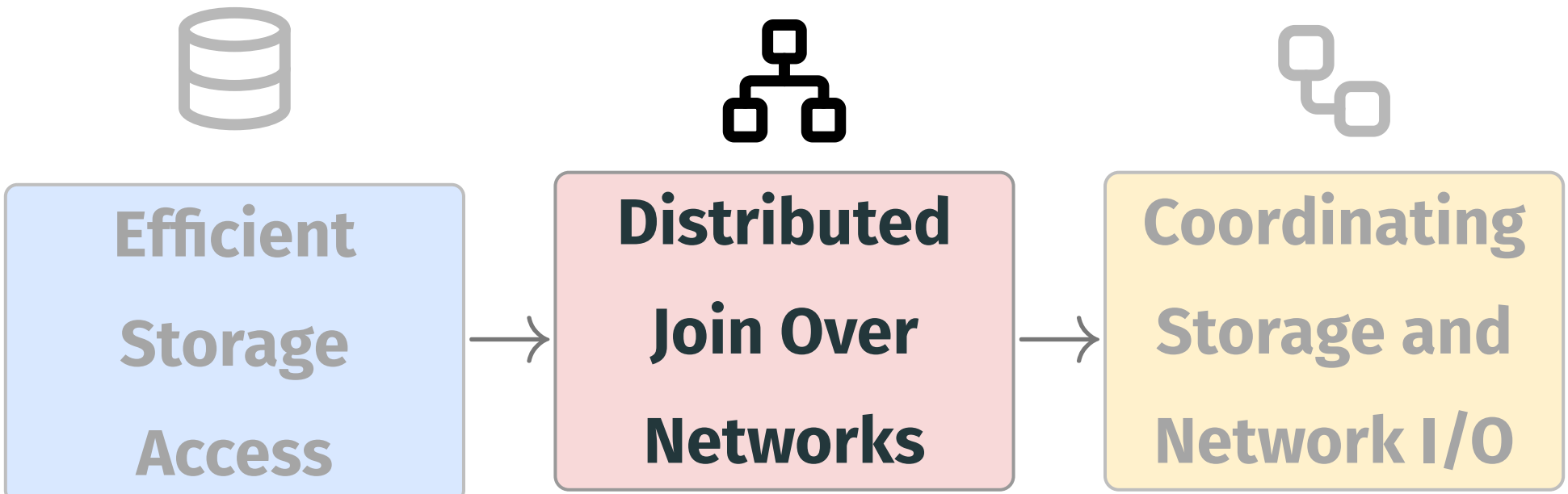
Storage microbenchmark with one GPU (A100, 40GB) and 4 NVMe SSD in PCIe 4.0.

Efficient Storage Access: Mis-Sync Elimination

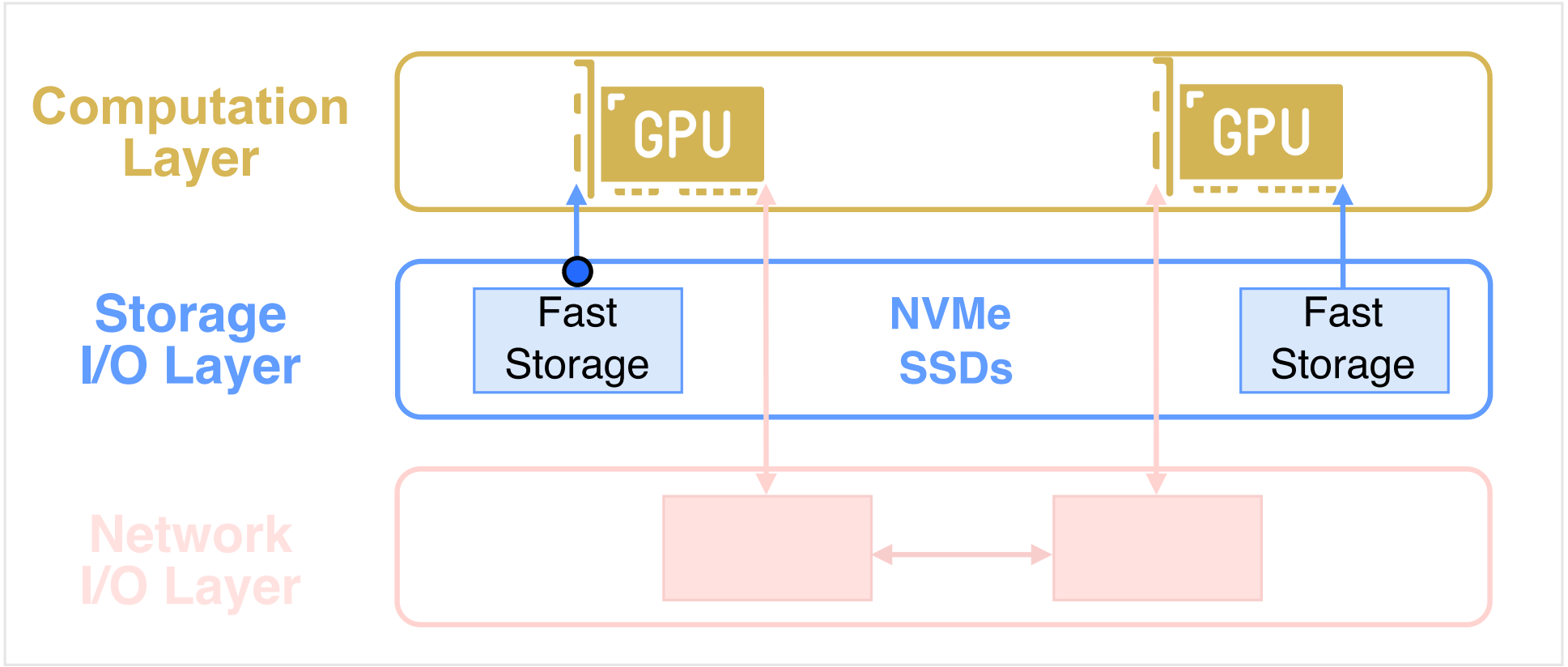


Storage microbenchmark with one GPU (A100, 40GB) and 4 NVMe SSD in PCIe 4.0.

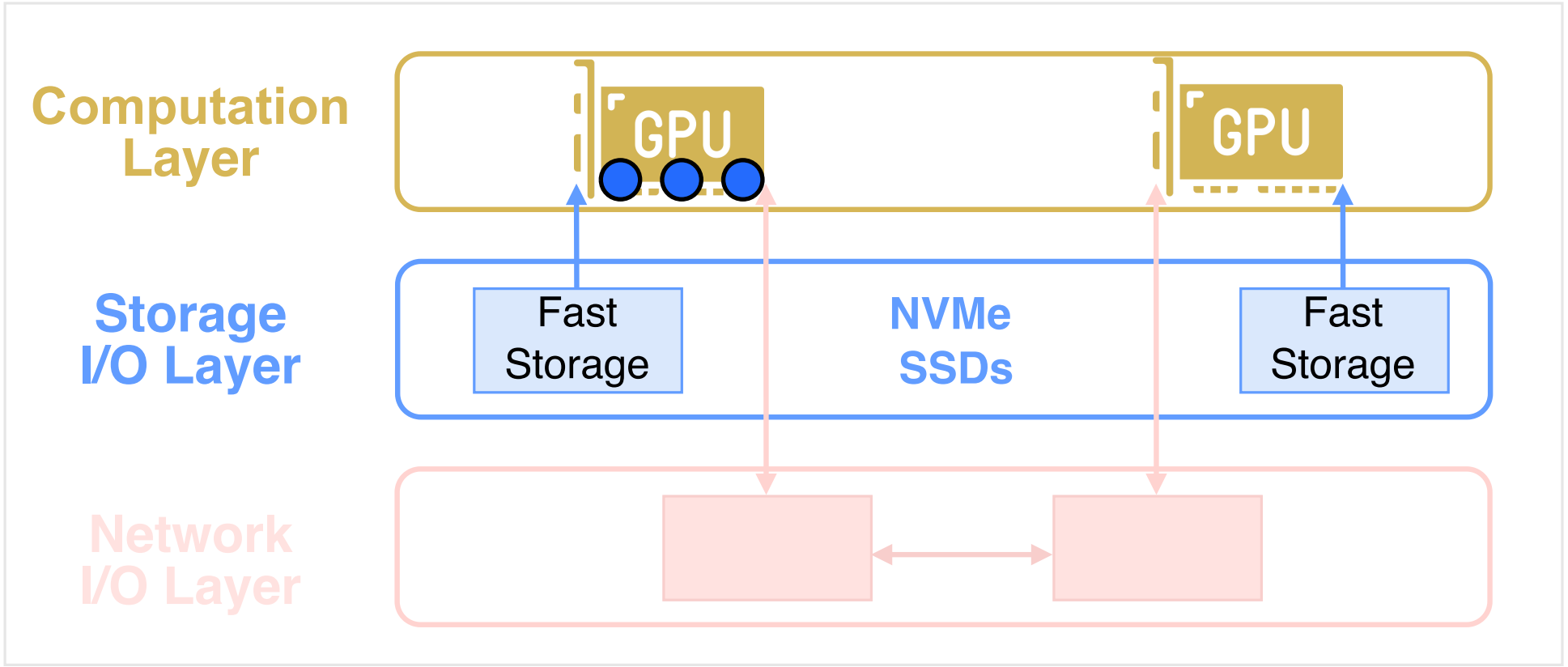
PystachIO: Network



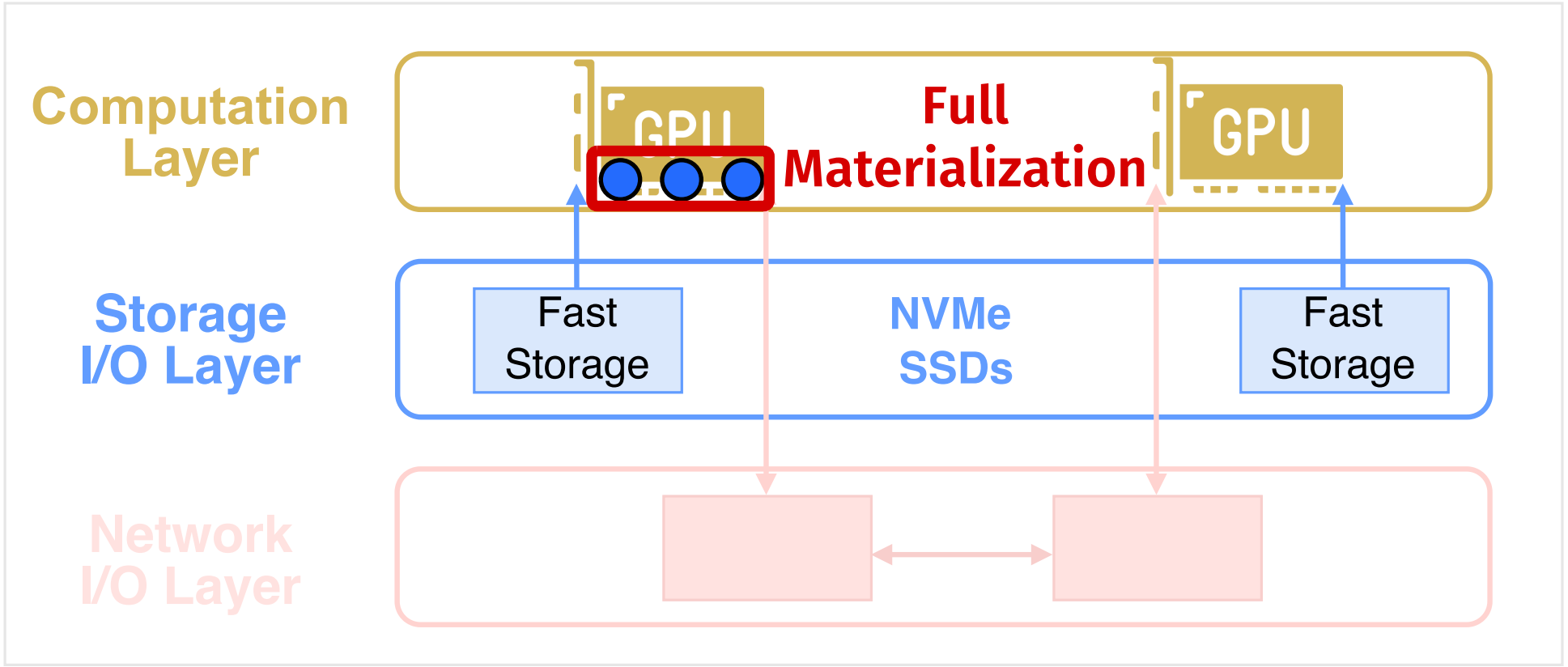
Run a Distributed Query



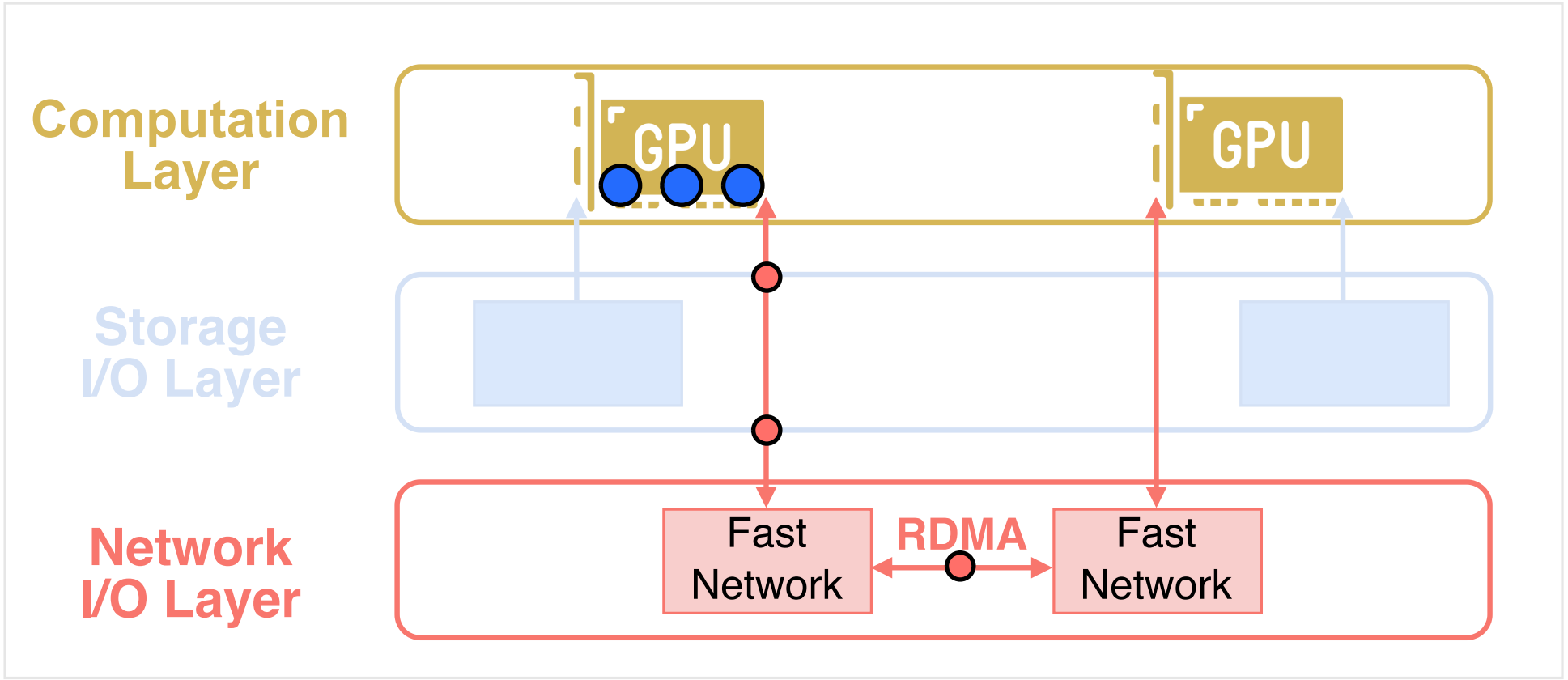
Run a Distributed Query



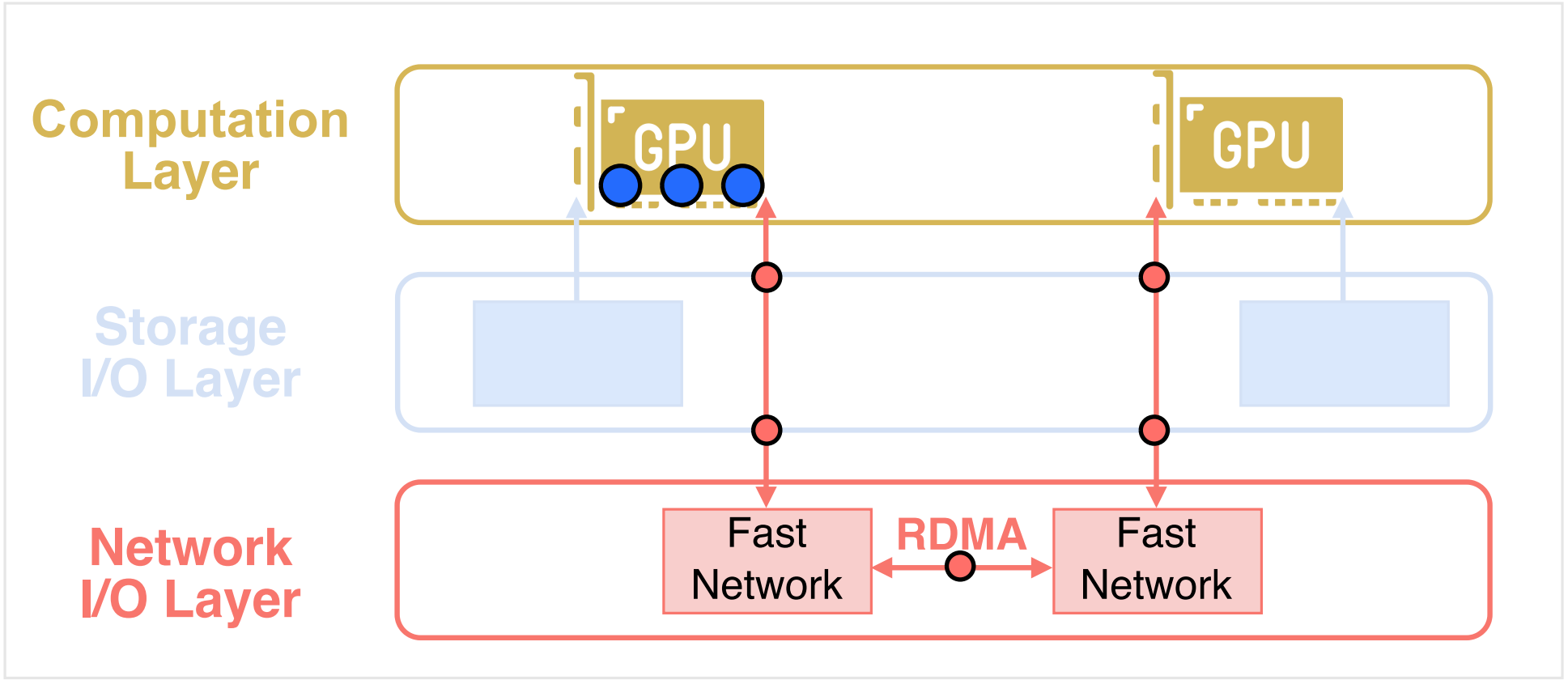
Run a Distributed Query



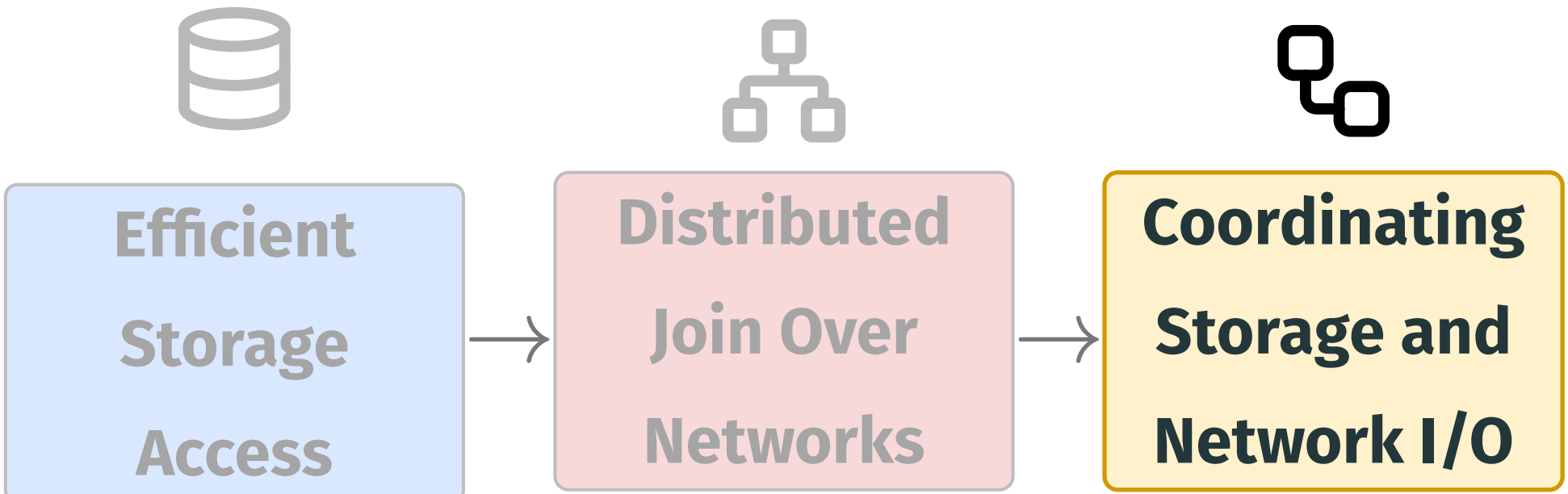
Run a Distributed Query



Run a Distributed Query

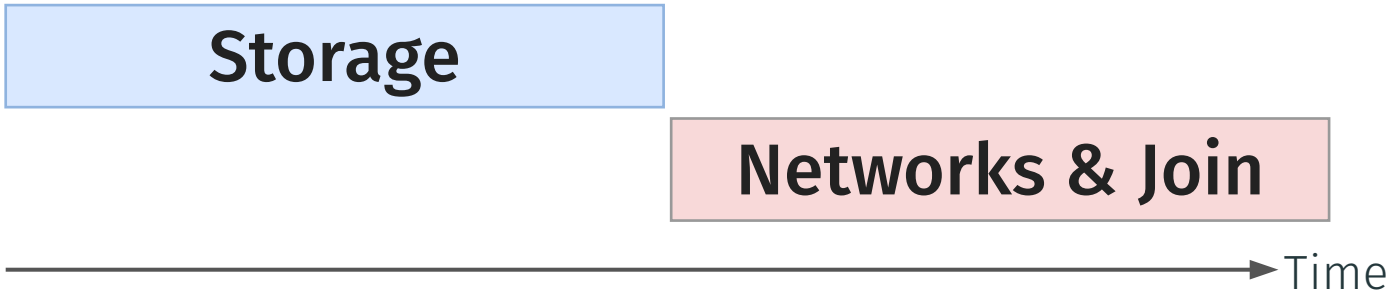


PystachIO: Coordinating I/O



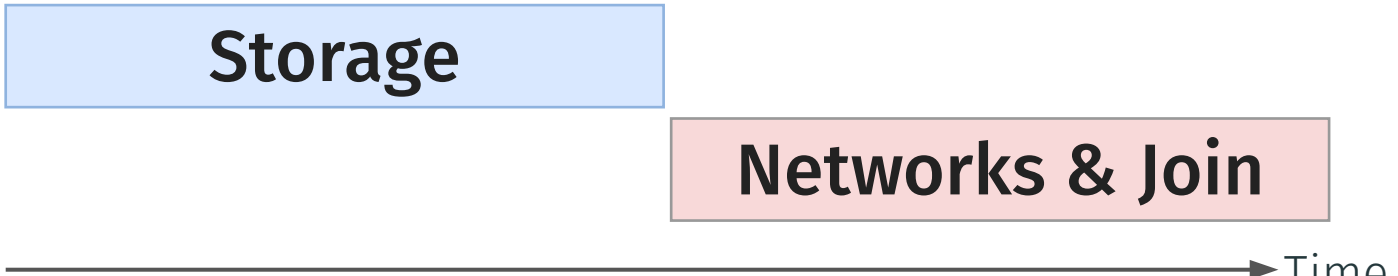
Fast Storage, then Fast Network

- Storage optimized
- Network optimized



Fast Storage, then Fast Network

- Storage optimized
- Network optimized



– Slow Runtime

– High Risk of OOM

– Only One I/O Device Active

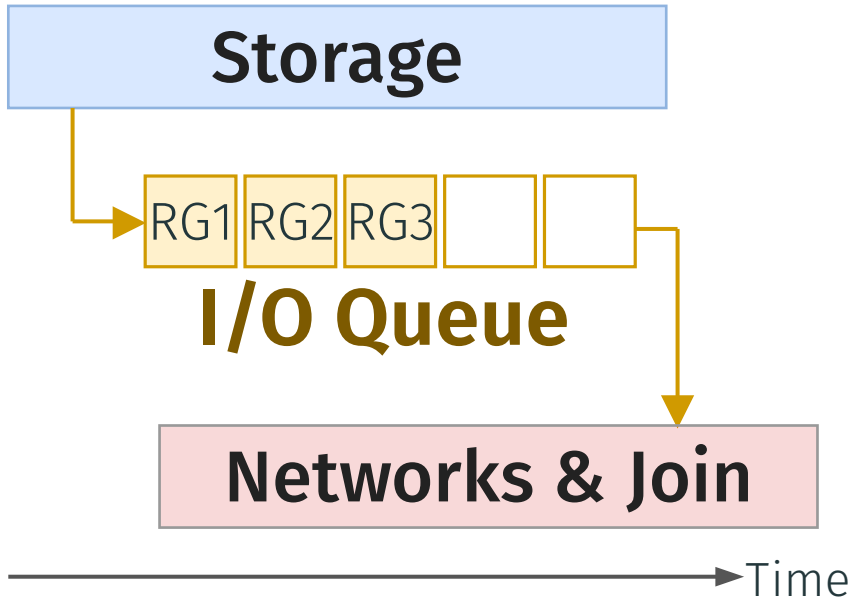
Fast Storage, then Fast Network

- Storage
- Networ

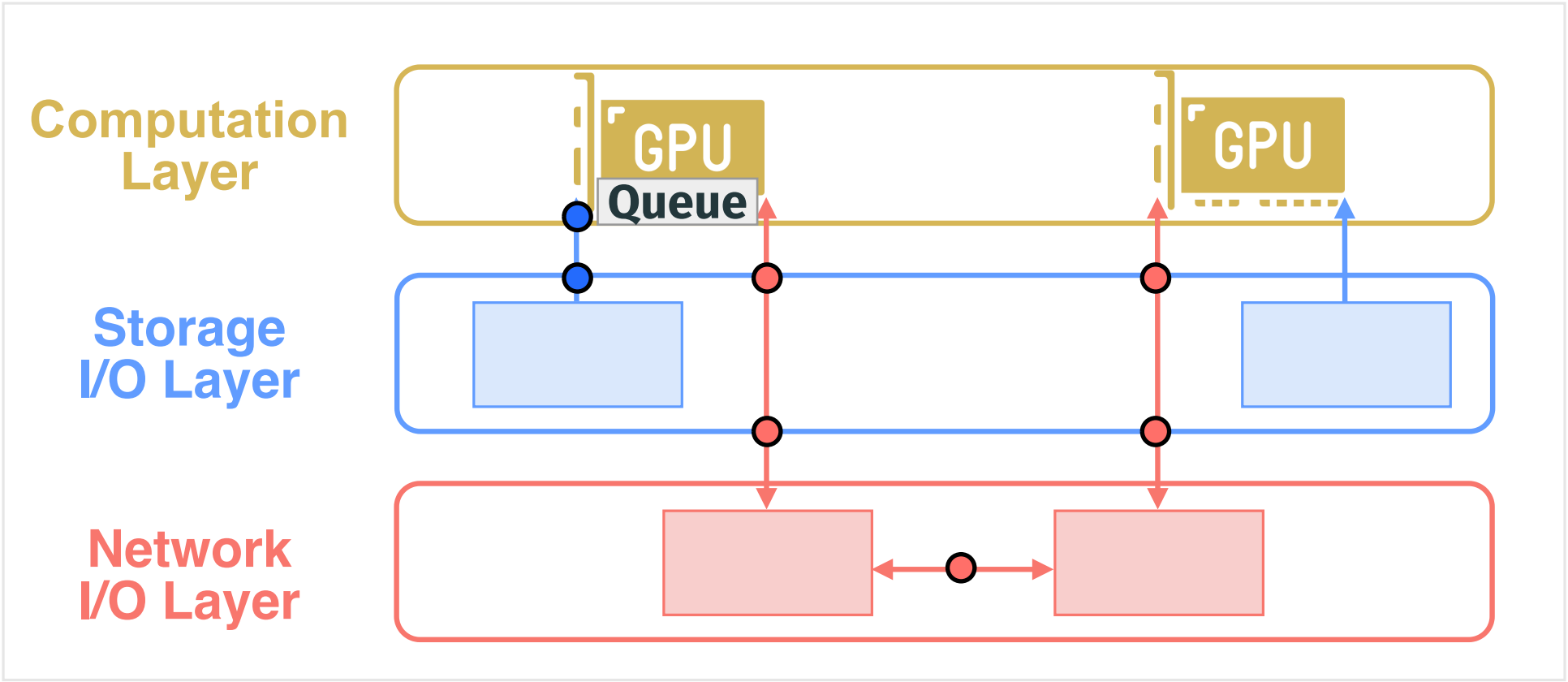
Is It Possible to Overlap Storage and Network?

I/O Queue For Coordination

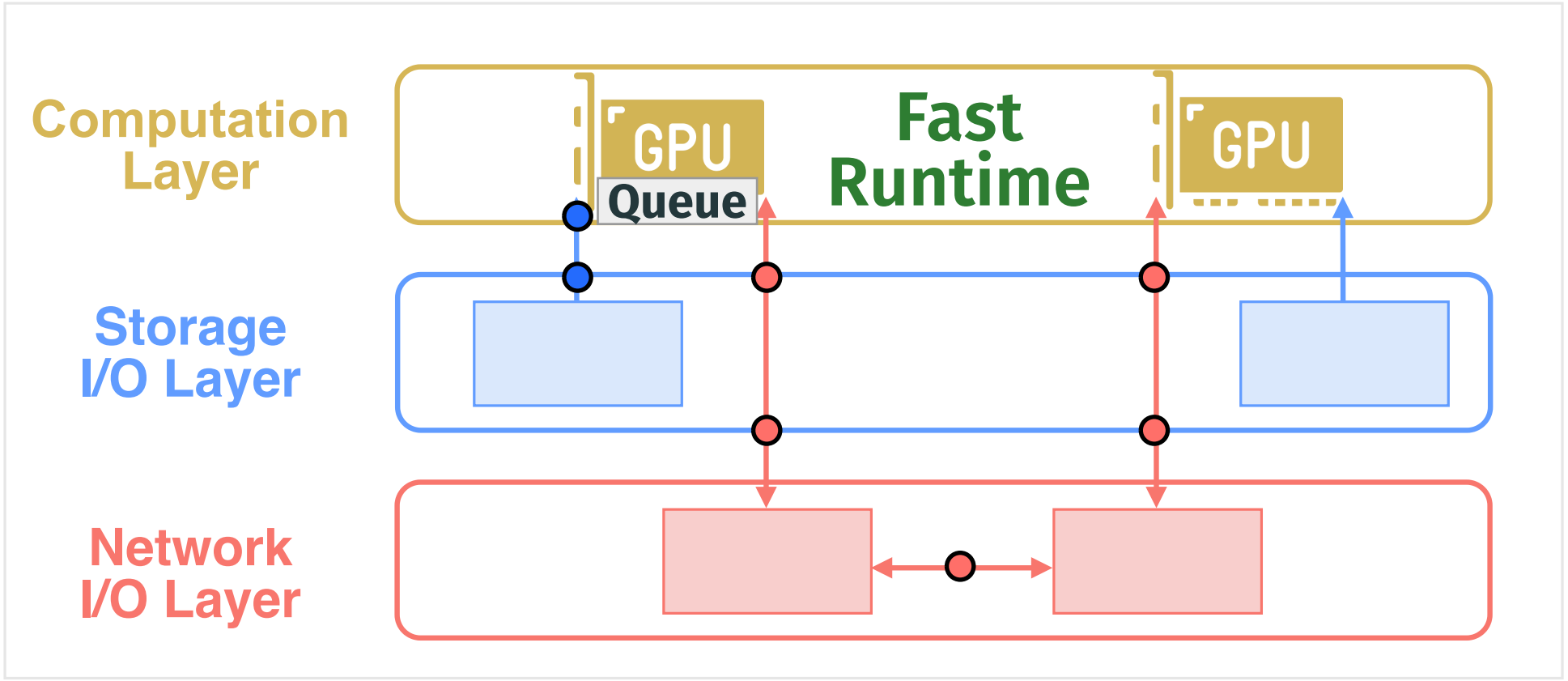
- Storage: enqueuing
- Network & Join: dequeuing
- Then network shuffling



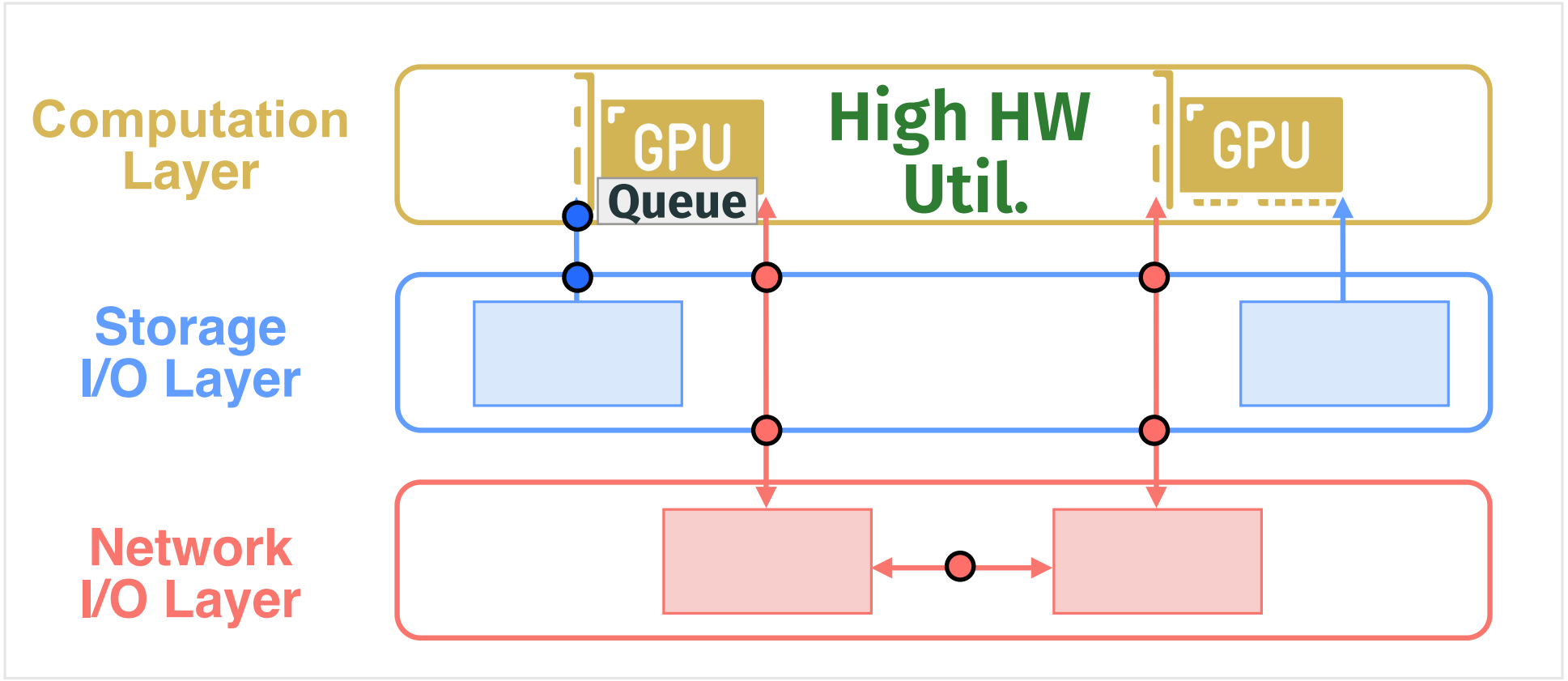
Overlapping Network, Storage, and Computation



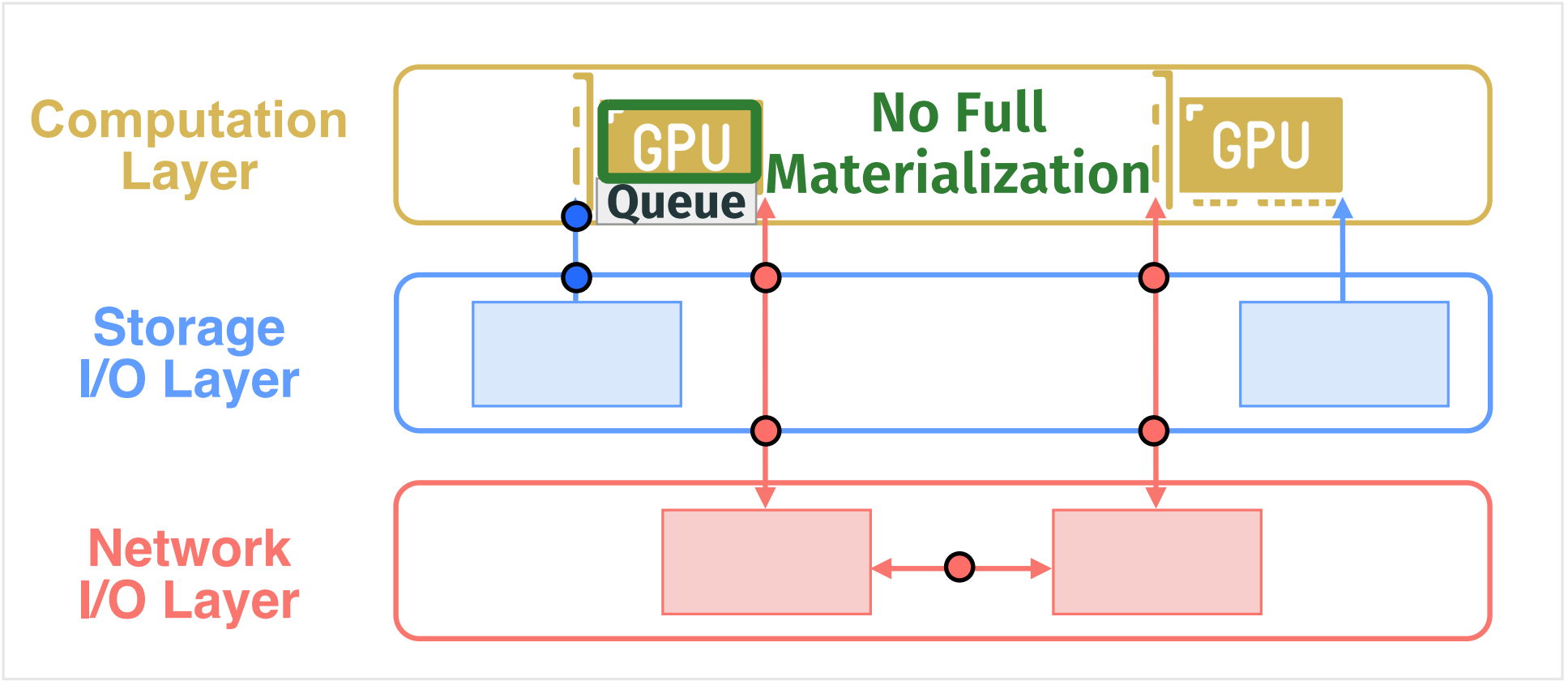
Overlapping Network, Storage, and Computation



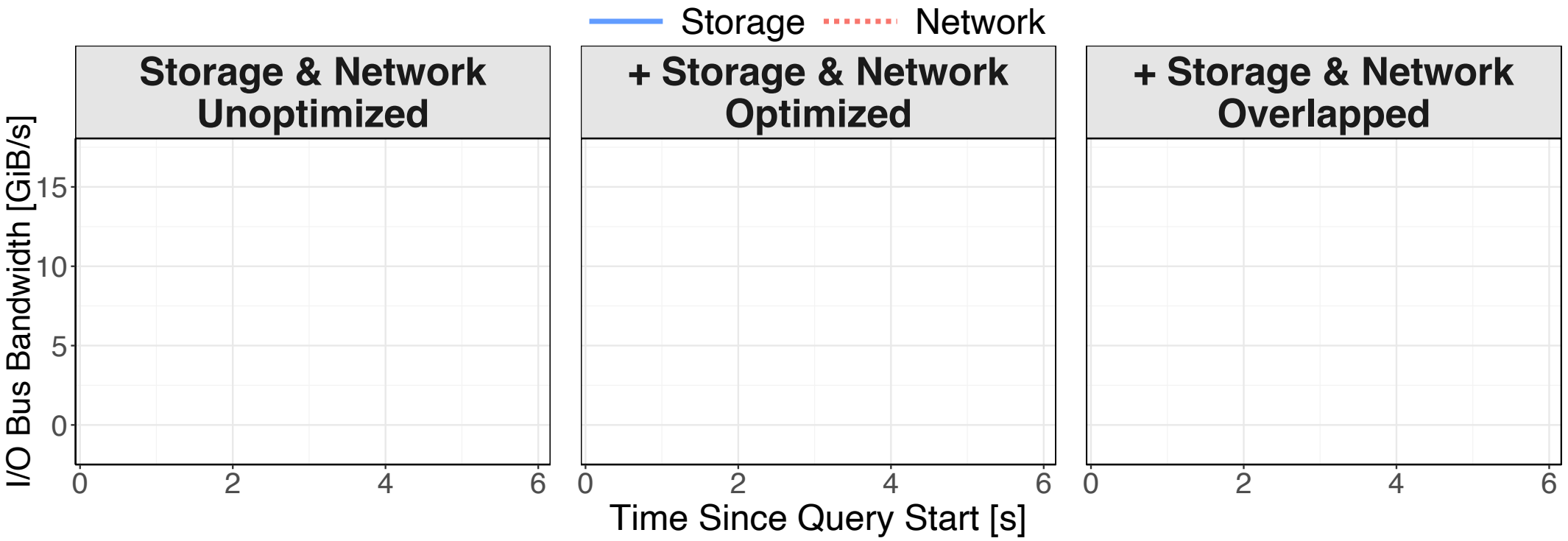
Overlapping Network, Storage, and Computation



Overlapping Network, Storage, and Computation

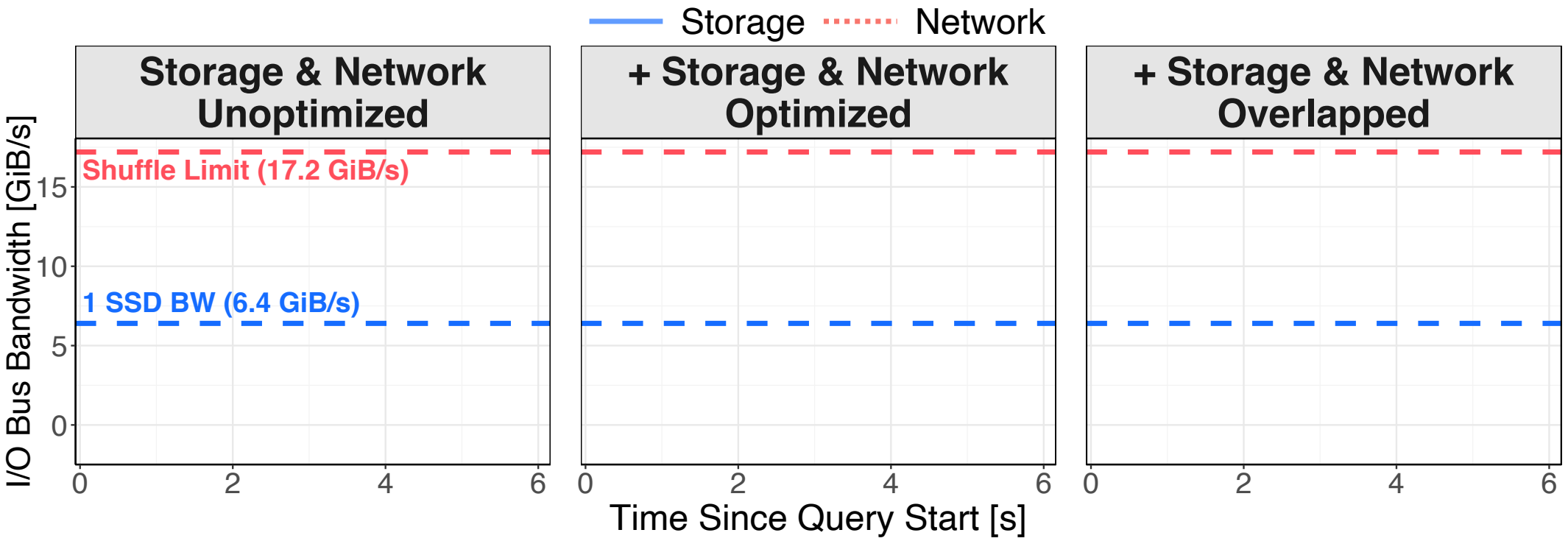


Overlapping Network, Storage, and Computation



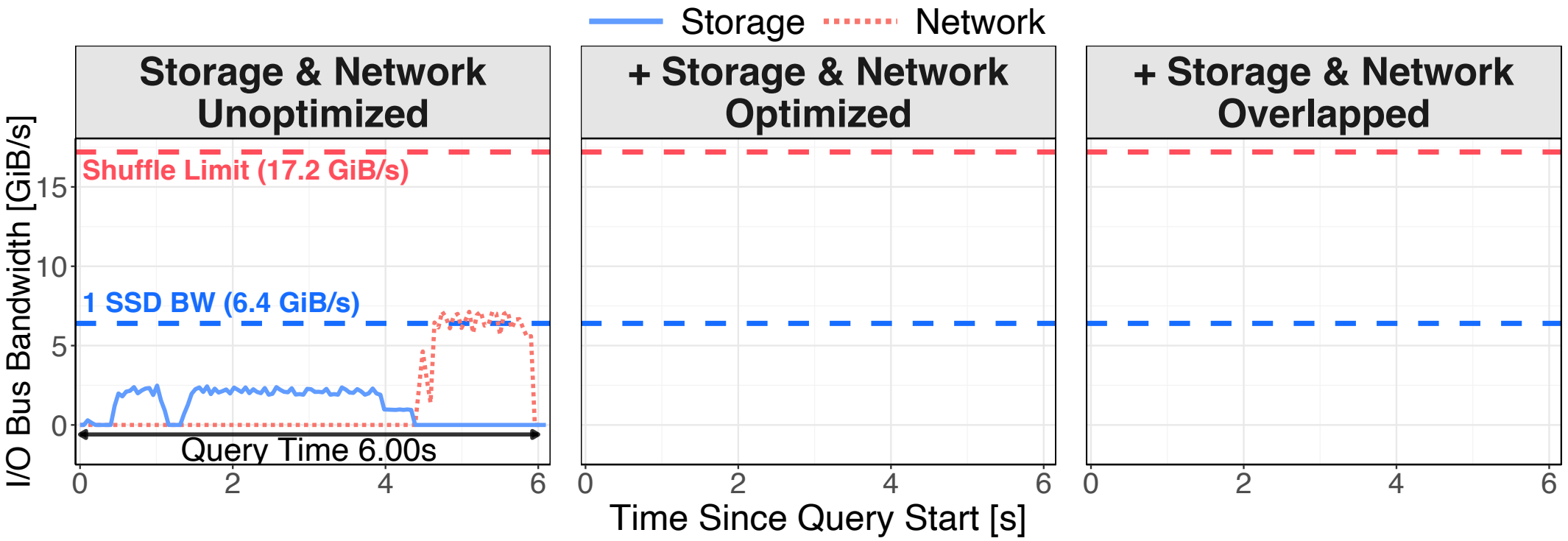
TPC-H Query 3 (SF 500) runtime executed on 2 GPUs (Each GPU has one SSD and one NIC)

Overlapping Network, Storage, and Computation



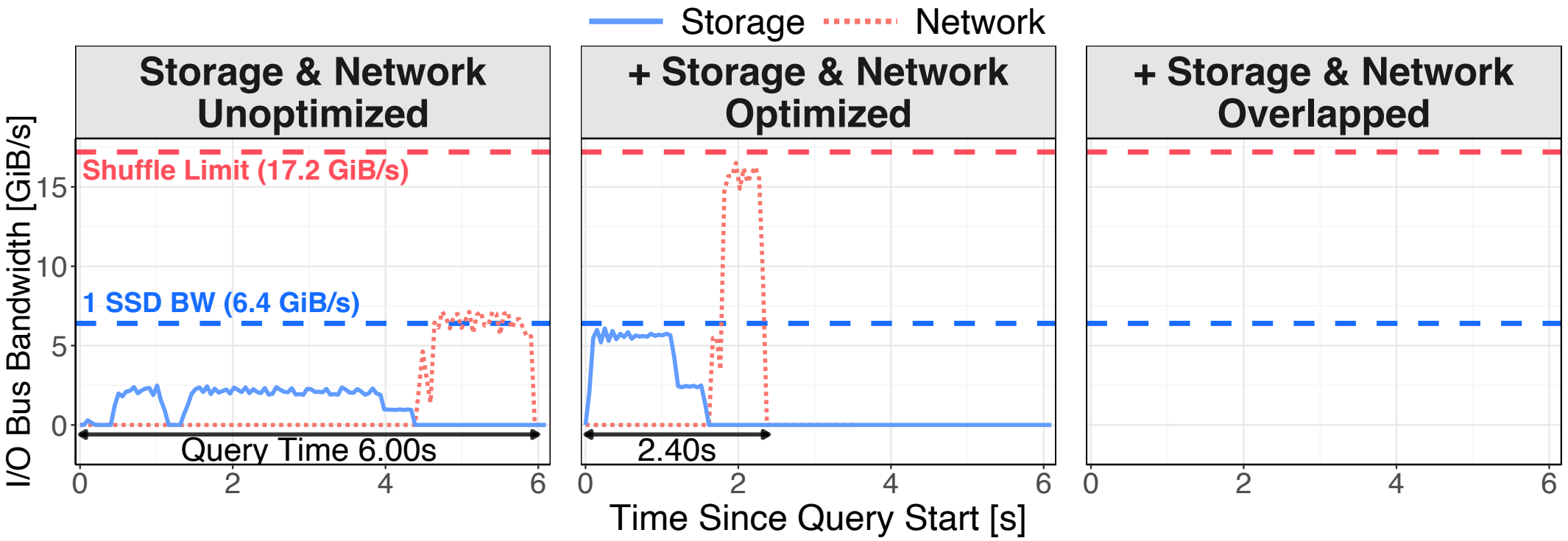
TPC-H Query 3 (SF 500) runtime executed on 2 GPUs (Each GPU has one SSD and one NIC)

Overlapping Network, Storage, and Computation



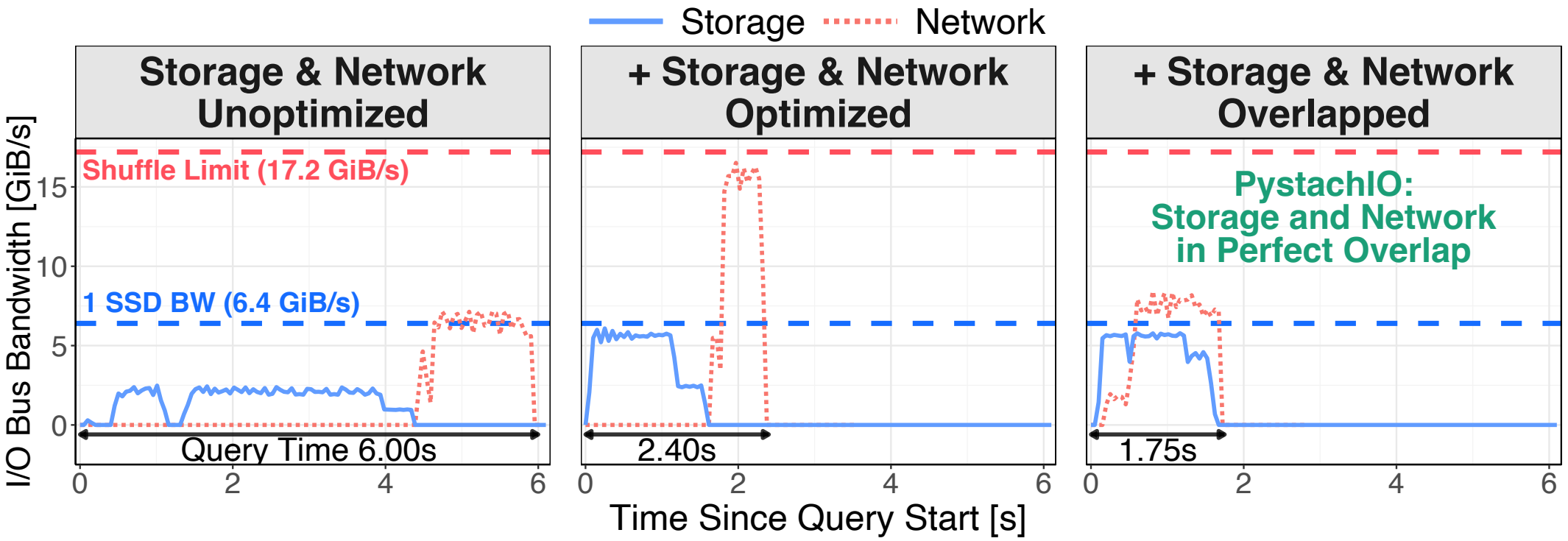
TPC-H Query 3 (SF 500) runtime executed on 2 GPUs (Each GPU has one SSD and one NIC)

Overlapping Network, Storage, and Computation



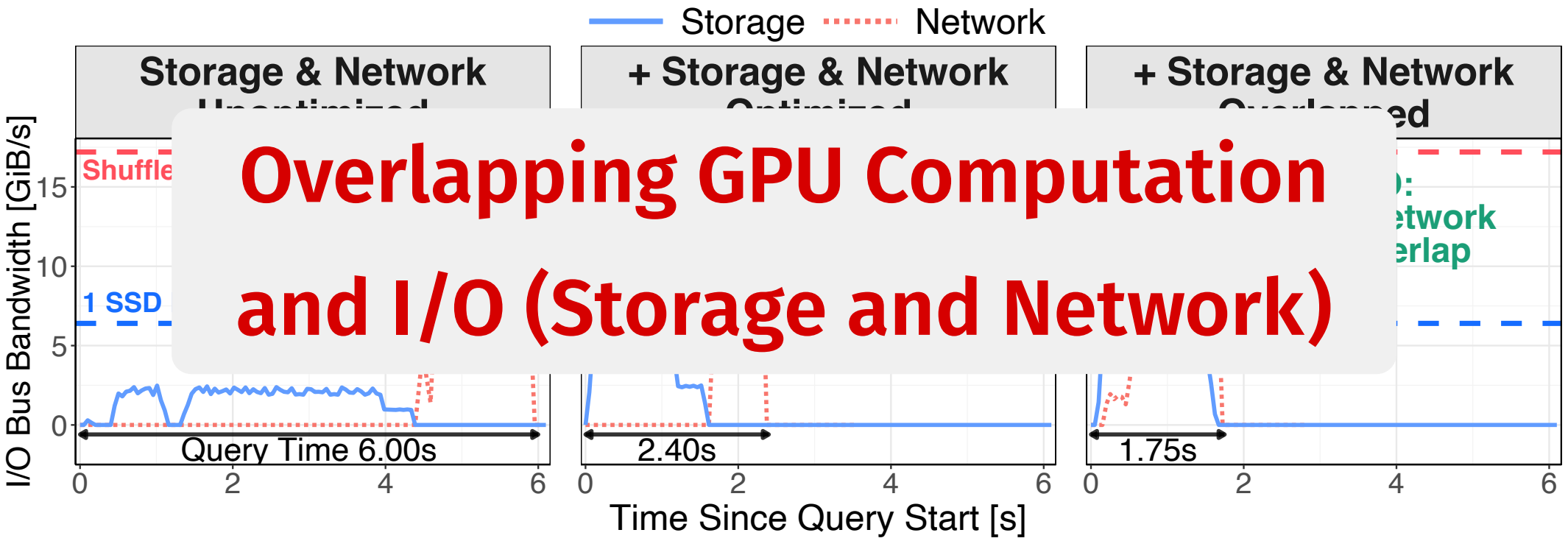
TPC-H Query 3 (SF 500) runtime executed on 2 GPUs (Each GPU has one SSD and one NIC)

Overlapping Network, Storage, and Computation



TPC-H Query 3 (SF 500) runtime executed on 2 GPUs (Each GPU has one SSD and one NIC)

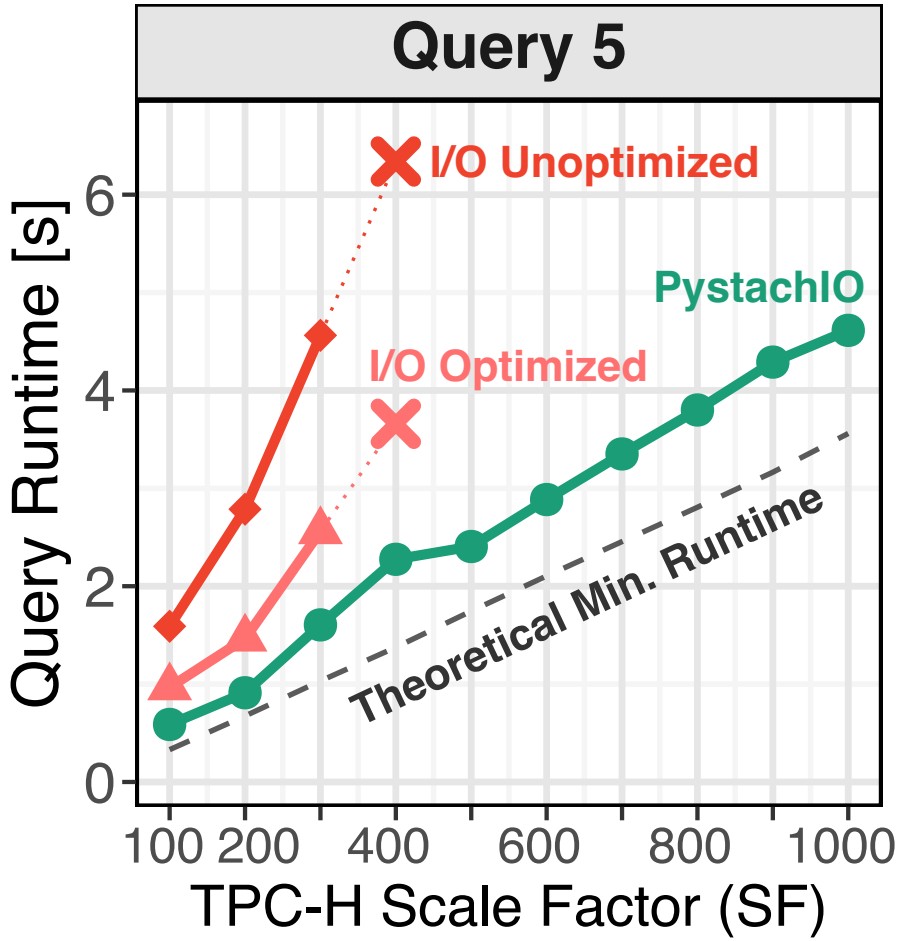
Overlapping Network, Storage, and Computation



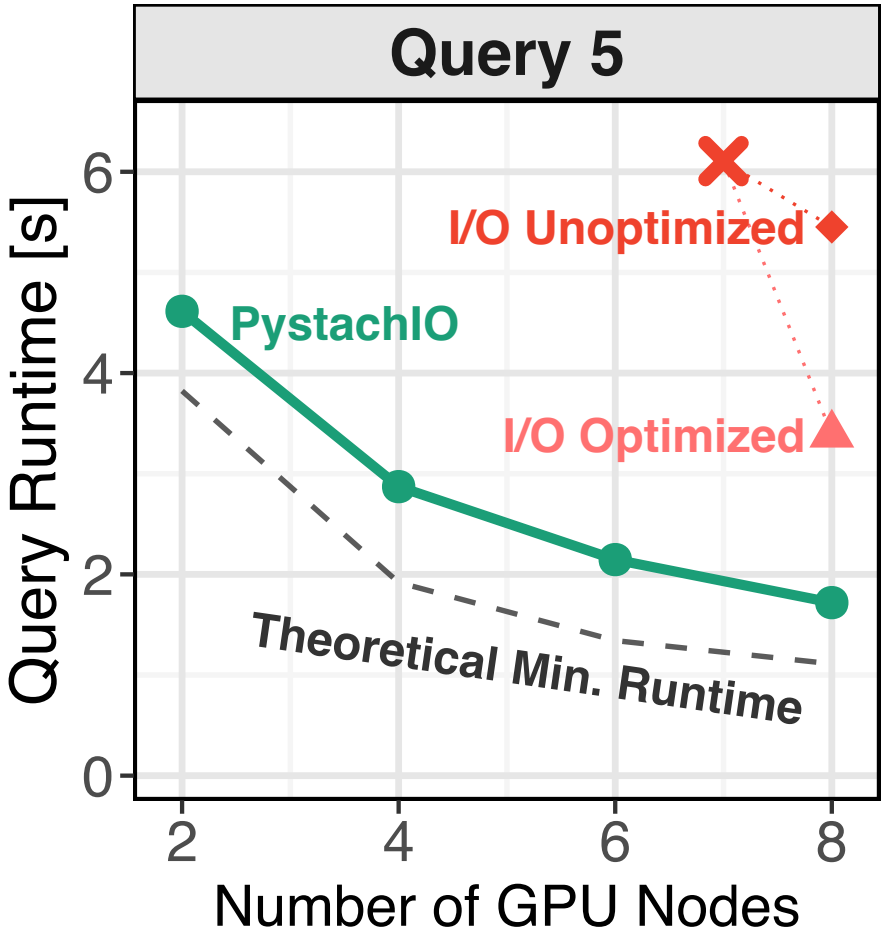
Overlapping GPU Computation and I/O (Storage and Network)

TPC-H Query 3 (SF 500) runtime executed on 2 GPUs (Each GPU has one SSD and one NIC)

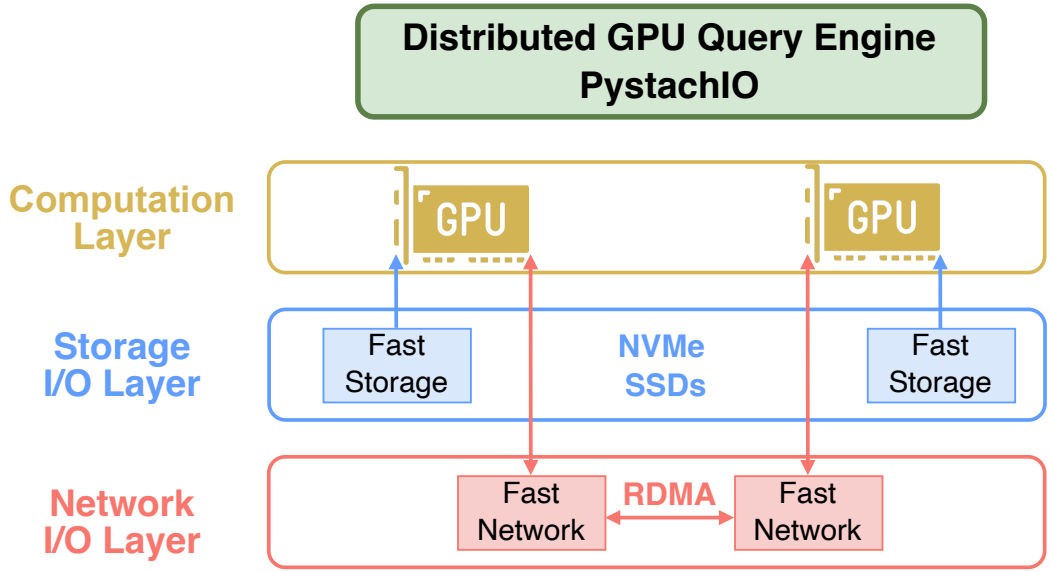
Scalability: Dataset Size with 2 GPUs



Scalability: Number of GPUs with SF1000



Conclusion



- Efficient I/O
- Overlapping GPU & I/O
- I/O Coordination: Storage & Network

Paper

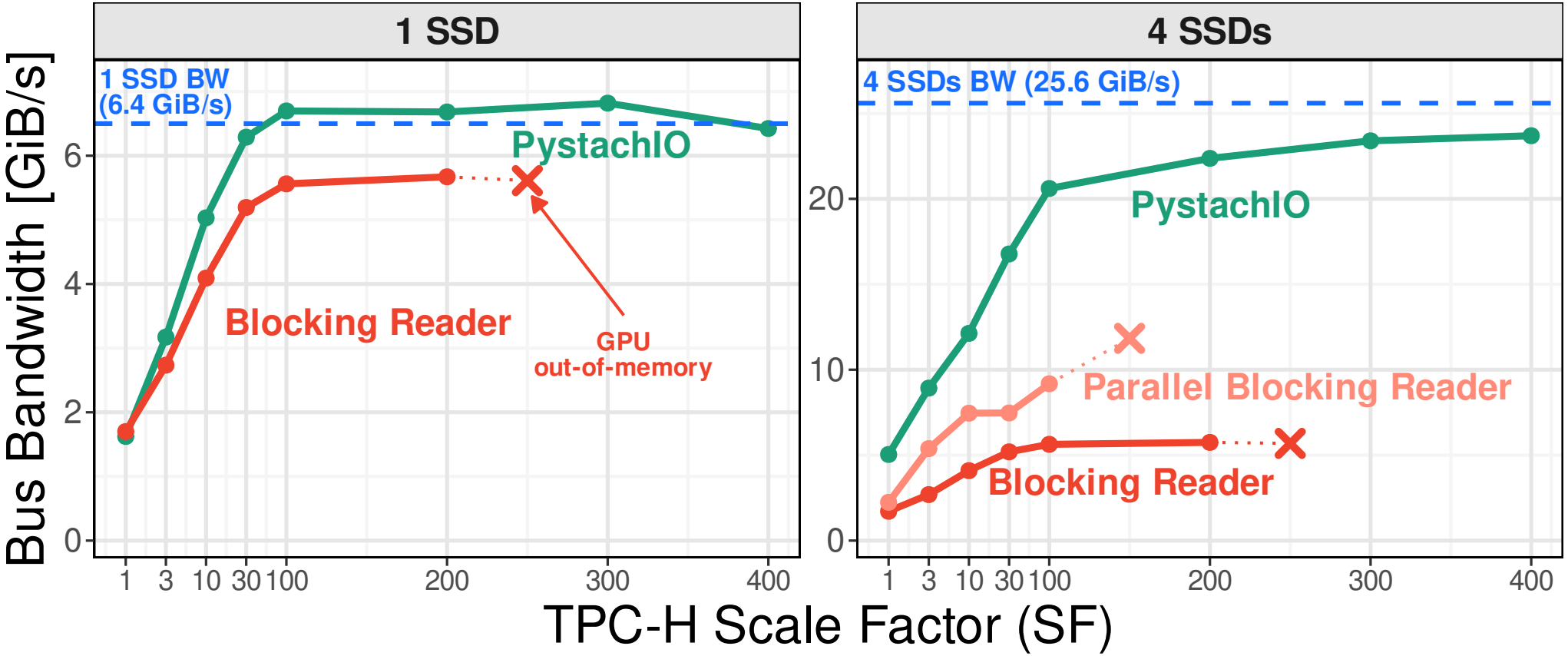
Code

TECHNISCHE UNIVERSITÄT DARMSTADT

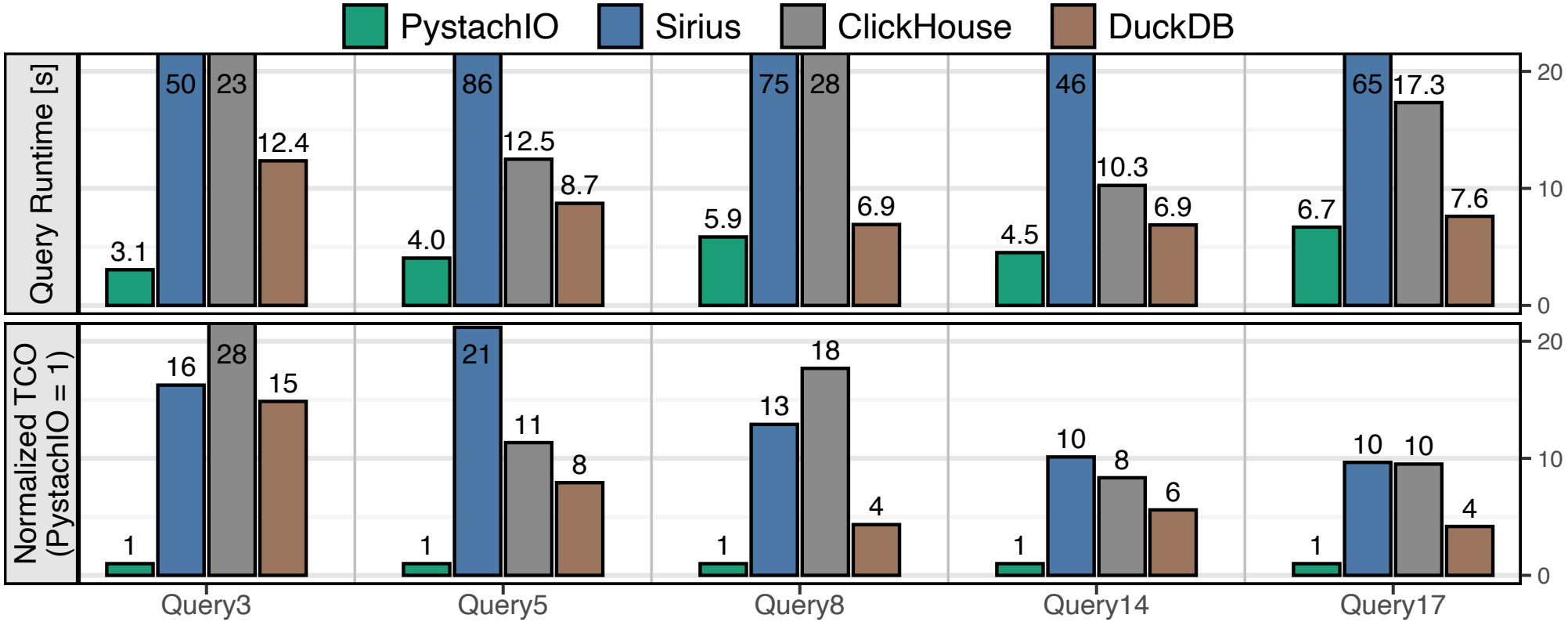
SYSTEMS

Thanks For Your Attention!

[BACKUP] Storage Scalability: Dataset Size

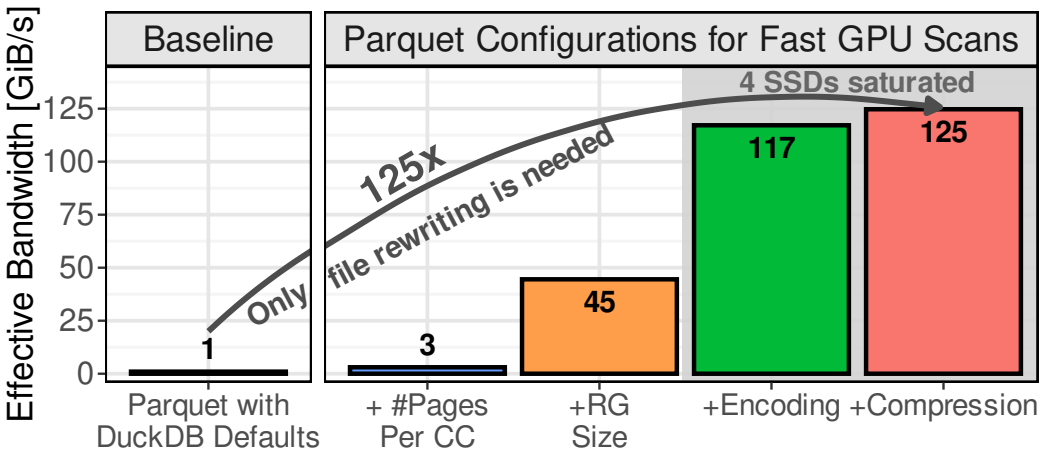


[BACKUP] System Comparison



Instance Rental Cost: NVIDIA A100 40GB GPU: $1.99/h (Lambda Labs)
AMD EPYC CPU Node: $7.30/h (AWS r7a.24xlarge, On-Demand)

[BACKUP] DaMoN'26 Conclusion



- GPU-aware Parquet configuration matters: 4 Insights
- Parquet performs well once optimized for GPUs
- GPU-optimized config. ≠ CPU-optimized config.
- Understand & optimize Parquet **before** replacing it!

Paper

Code

TECHNISCHE UNIVERSITÄT DARMSTADT

SYSTEMS