How Close Are We to GPU Data Processing?

Toward Disaggregated Data Systems in the AI Hardware Era

Jigao Luo

Joint Seminar

2026

Systems Group, TU Darmstadt



TECHNISCHE UNIVERSITÄT DARMSTADT

SYSTEMS

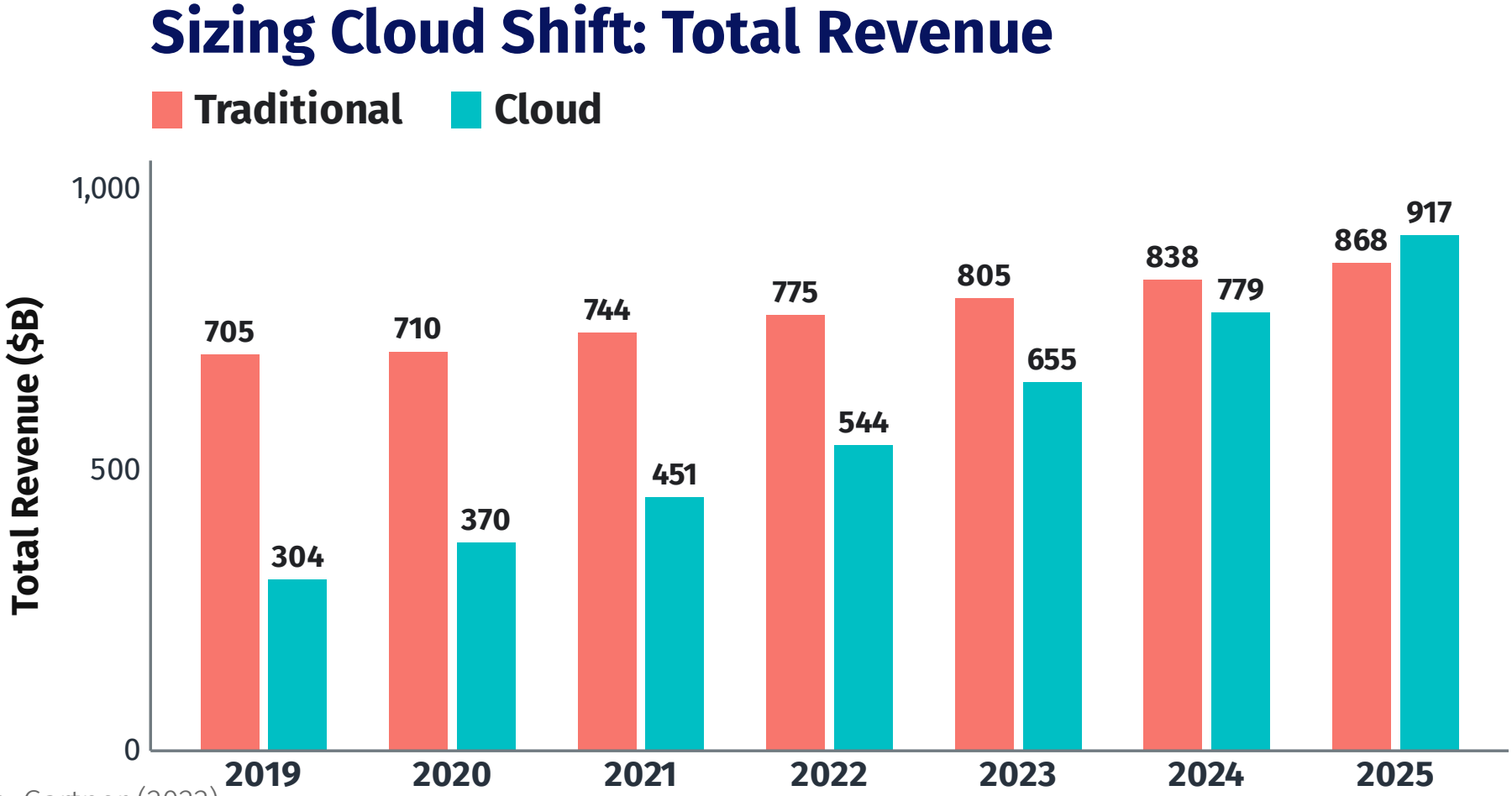
About Me

- Research: disaggregated GPU data systems
- Software Campus 27 with Snowflake on File Formats
- NVIDIA external contributor: Parquet reader
- Research Intern at Microsoft and Zilliz

Talk Structure

- Background: GPU Data Processing
- My Research
  - GPU File Format: DaMoN 26 🏆
  - Distributed GPU Engine PystachIO: VLDB 26
  - Storage Protocol: BTW 25
- The Road Ahead

Cloud Is a Success Story



Source: Gartner (2022)

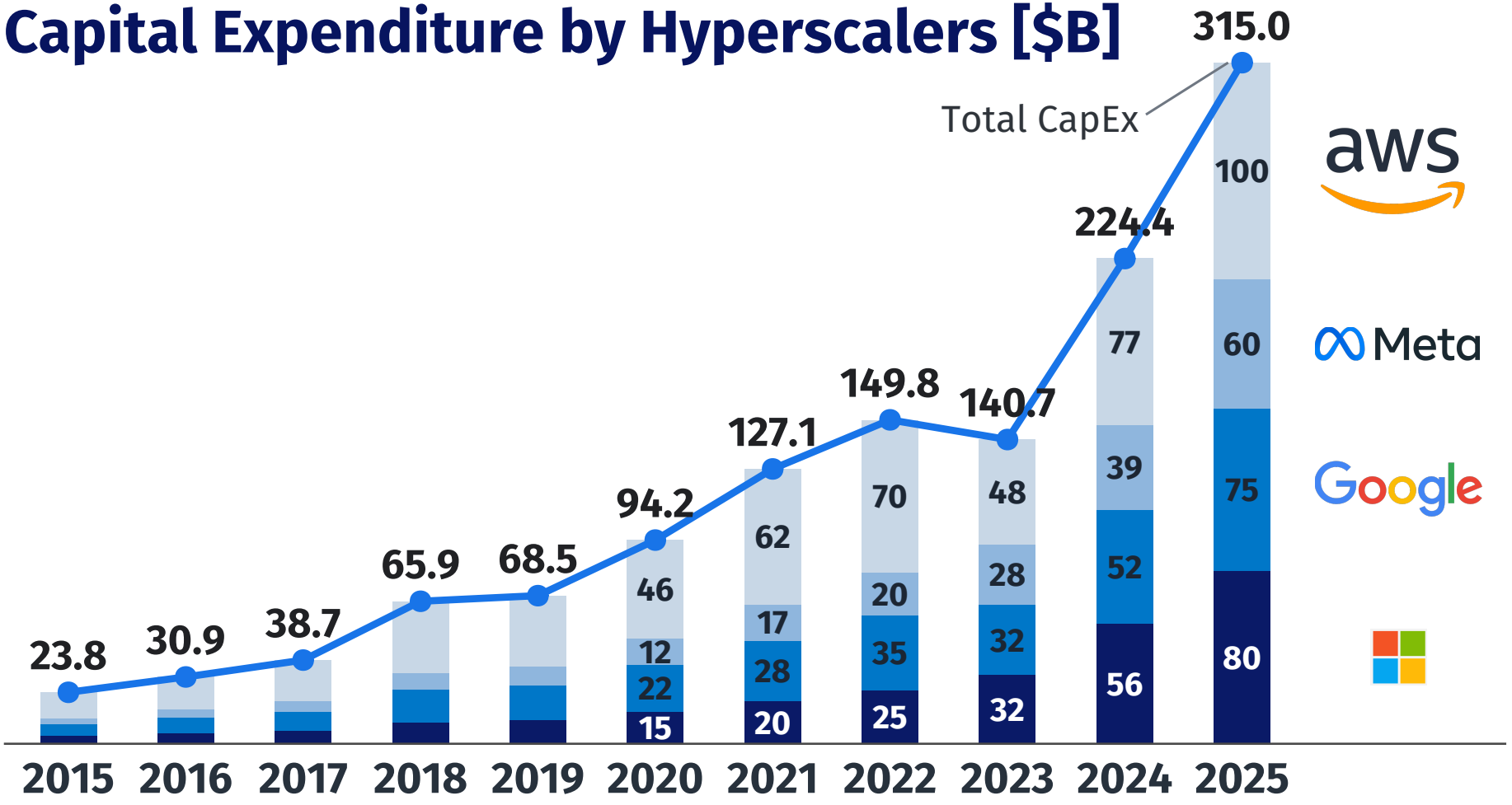
Cloud Database Is Also a Success Story

SaaS IPO Ranking

Rank	Company	IPO Year	IPO Size ($B)
1	**Snowflake**	2020	3.4
2	Qualtrics	2021	1.55
3	SailPoint	2025	1.38
4	UiPath	2021	1.34
5	Figma	2025	1.22

AI Investment Is Shaping the Future

Capital Expenditure by Hyperscalers [$B]



Source: DC Pulse

How to Ride the AI Investment Wave for DB?

How to Ride the AI Investment Wave for DB?

- Issue: Existing GPU DBs remain single-node only

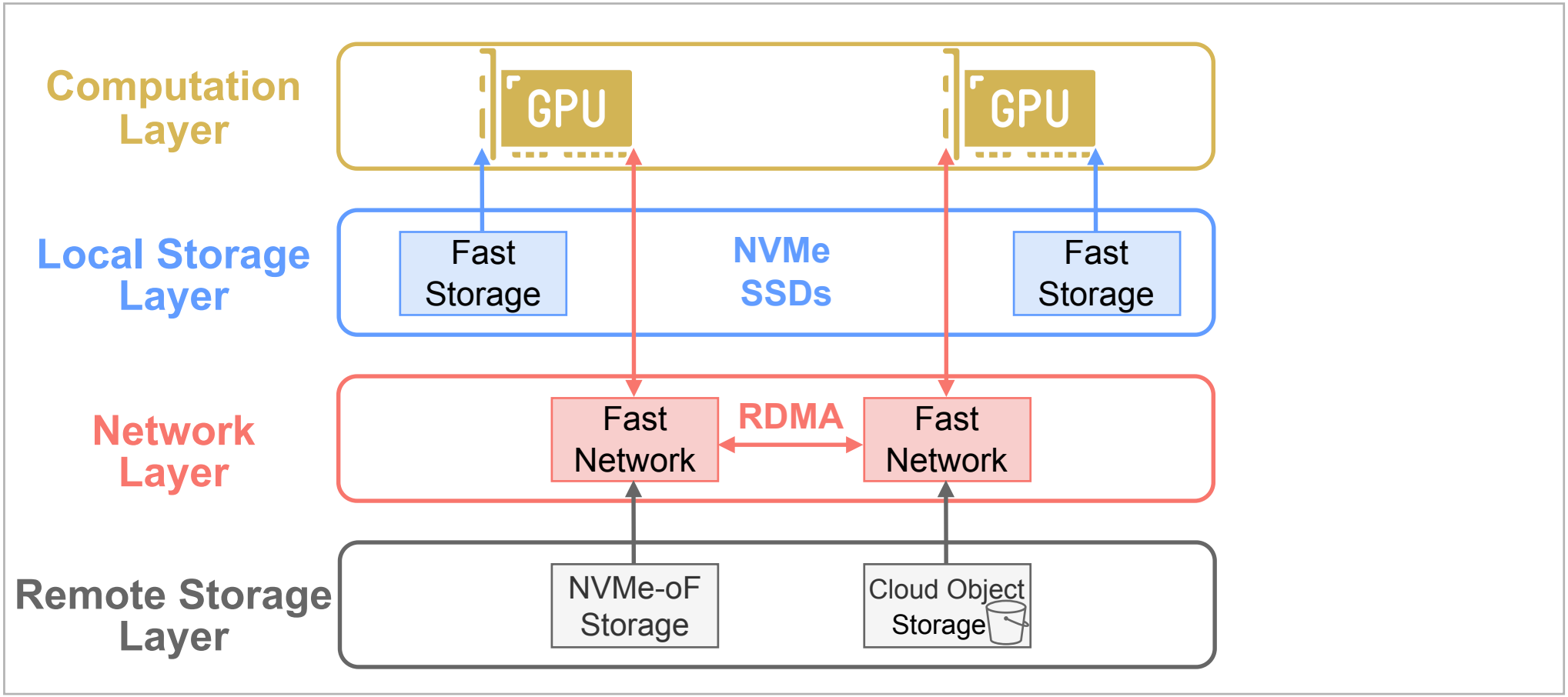
How to Ride the AI Investment Wave for DB?

- Issue: Existing GPU DBs remain single-node only
- **RQ: How do we build a GPU DB for disaggregation?**

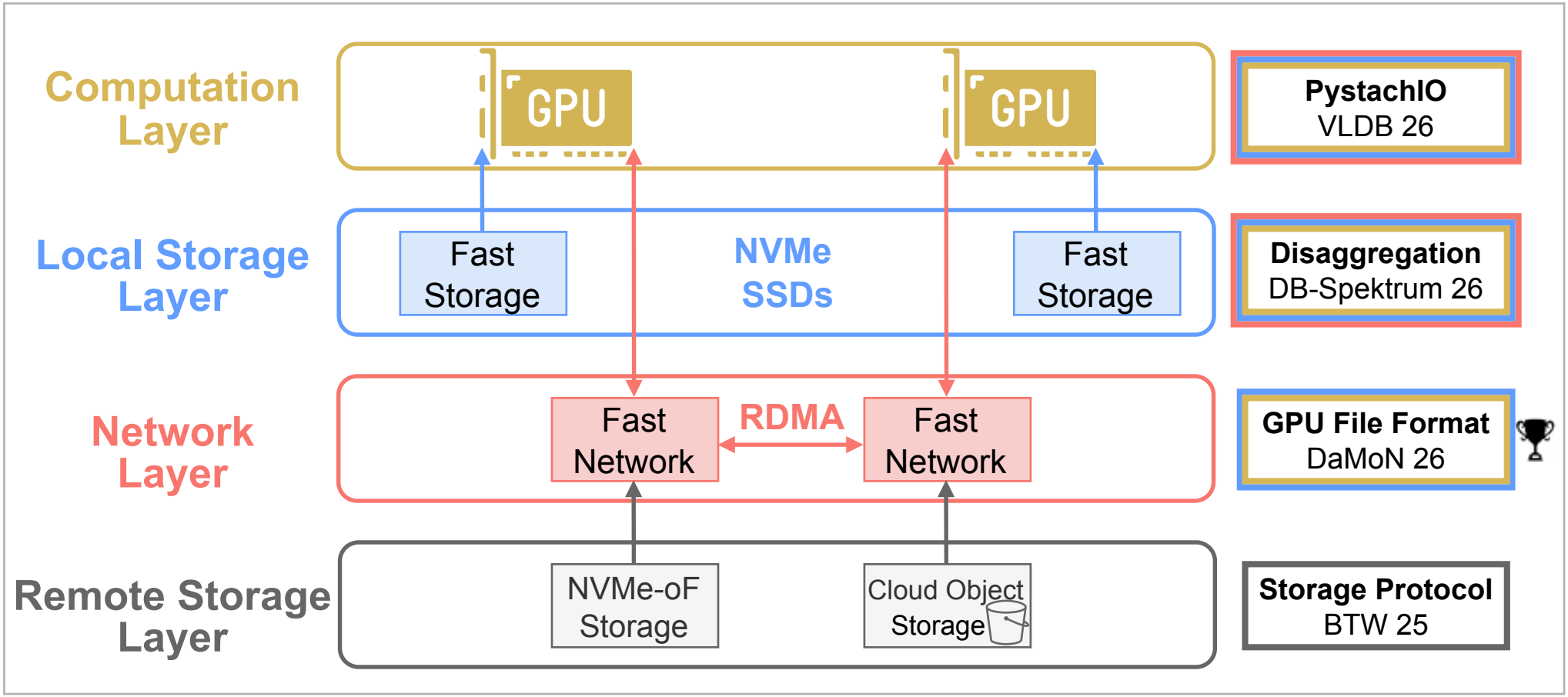
How to Ride the AI Investment Wave for DB?

- Issue: Existing GPU DBs remain single-node only
- **RQ: How do we build a GPU DB for disaggregation?**
- Challenges from fast networks and fast storage

My Research

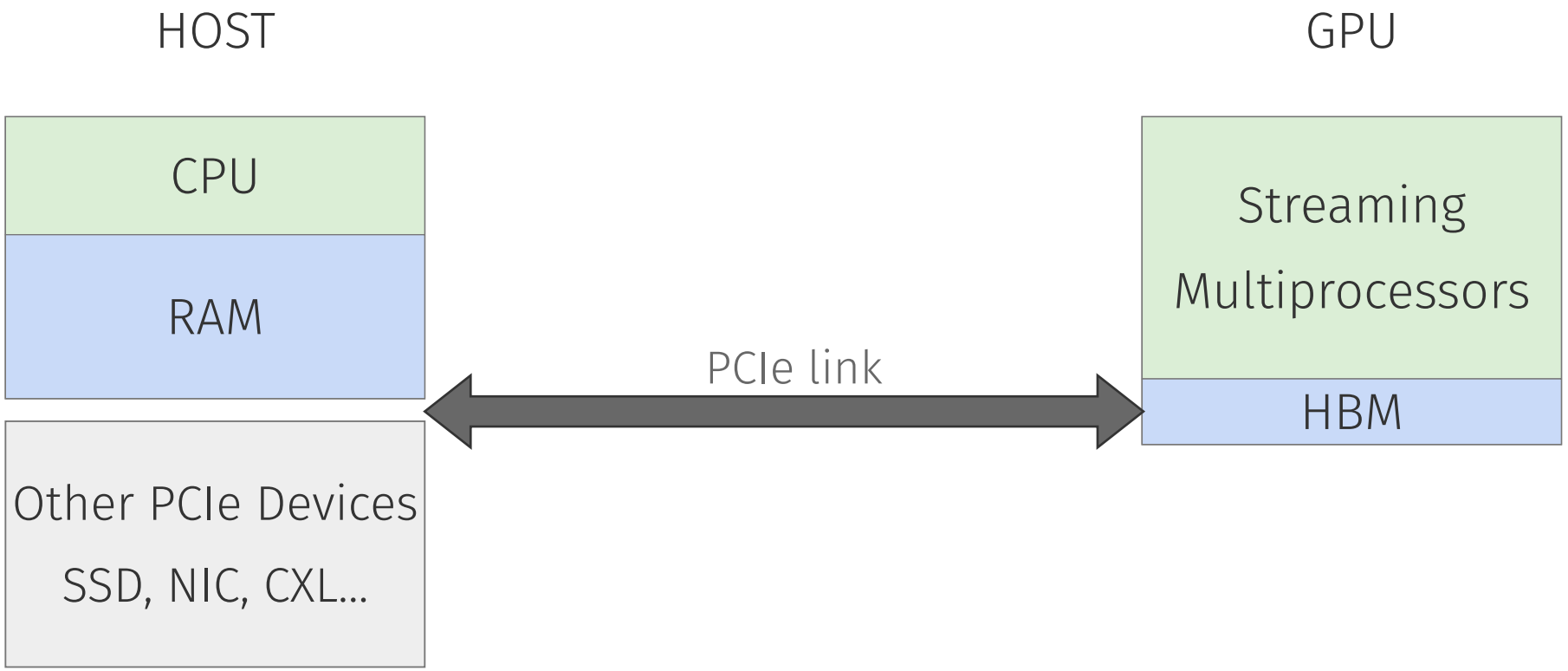

Computation Layer
GPU
GPU
Local Storage Layer
Fast Storage
NVMe SSDs
Fast Storage
Network Layer
Fast Network
RDMA
Fast Network
Remote Storage Layer
NVMe-oF Storage
Cloud Object Storage

My Research Status



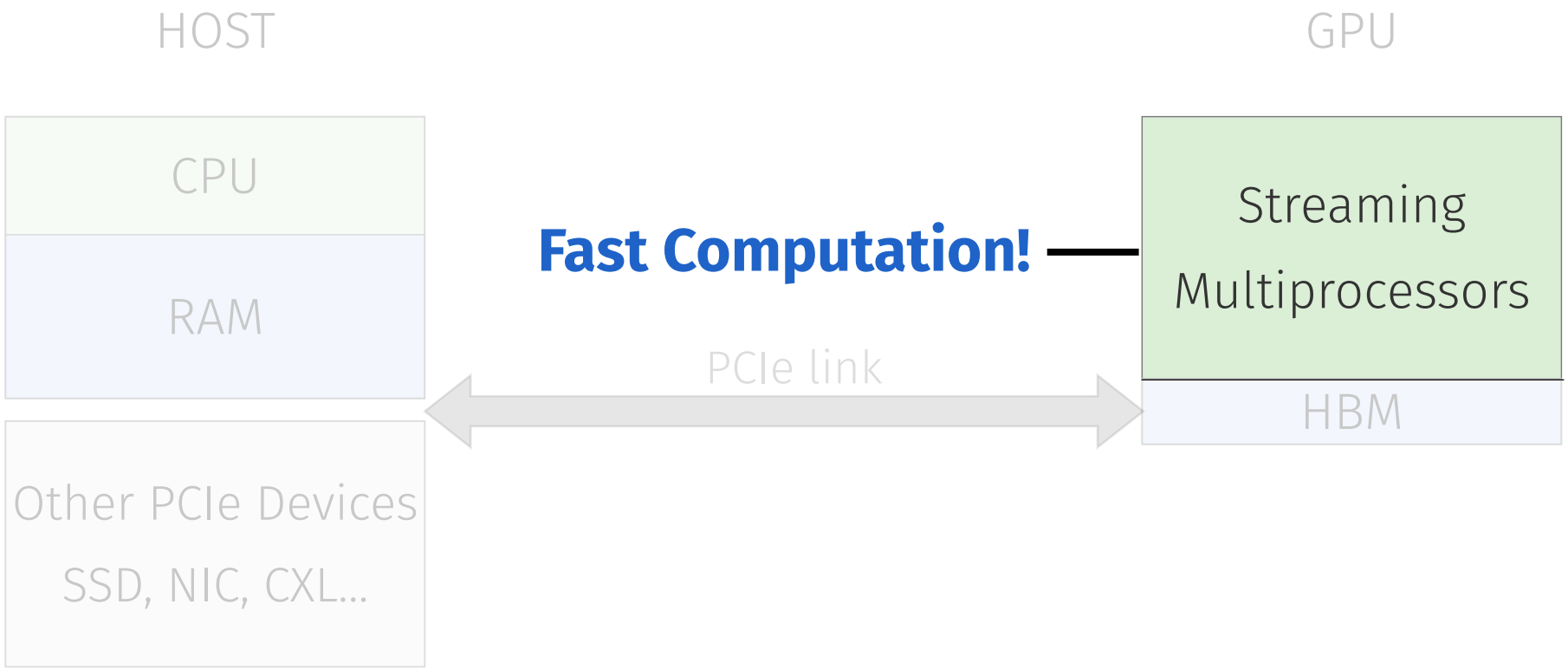
Background: GPU Data Processing

GPU System Primitives

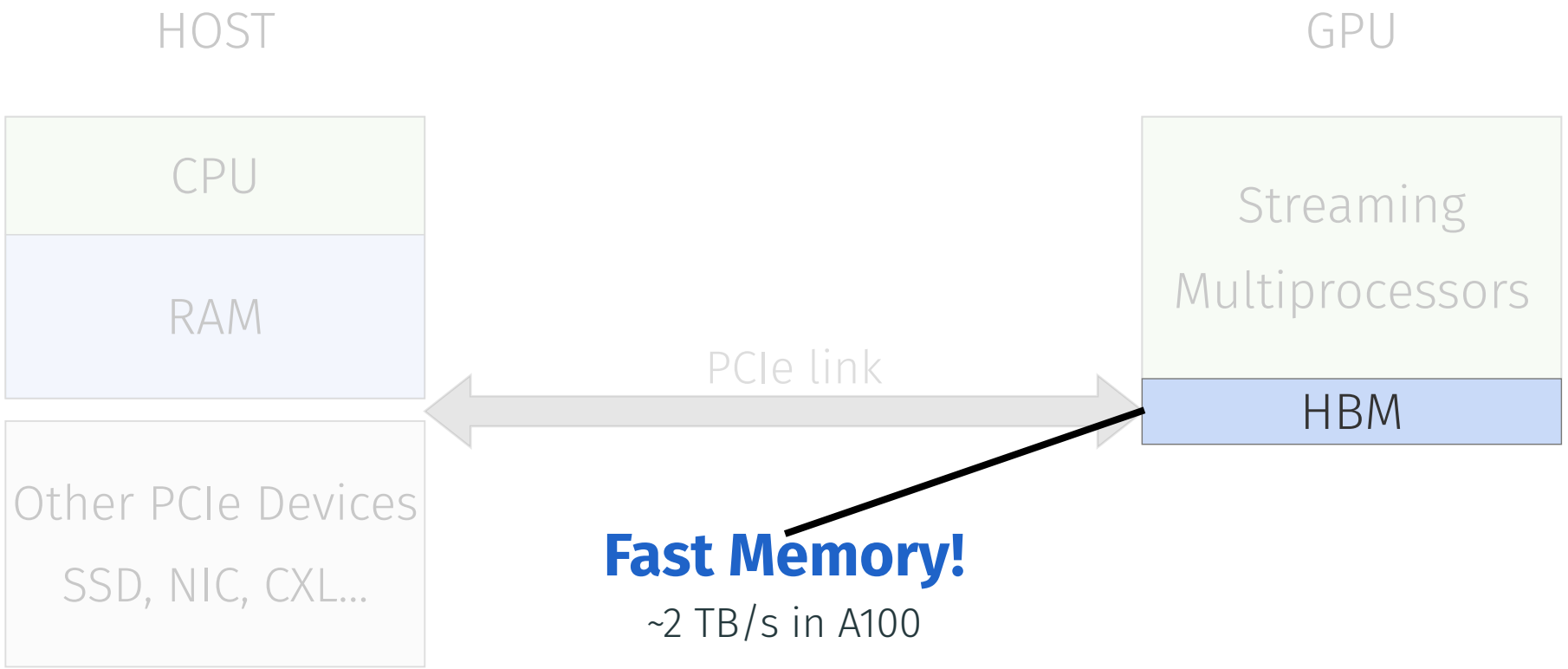


* NVLink C2C (chip-to-chip) excluded.

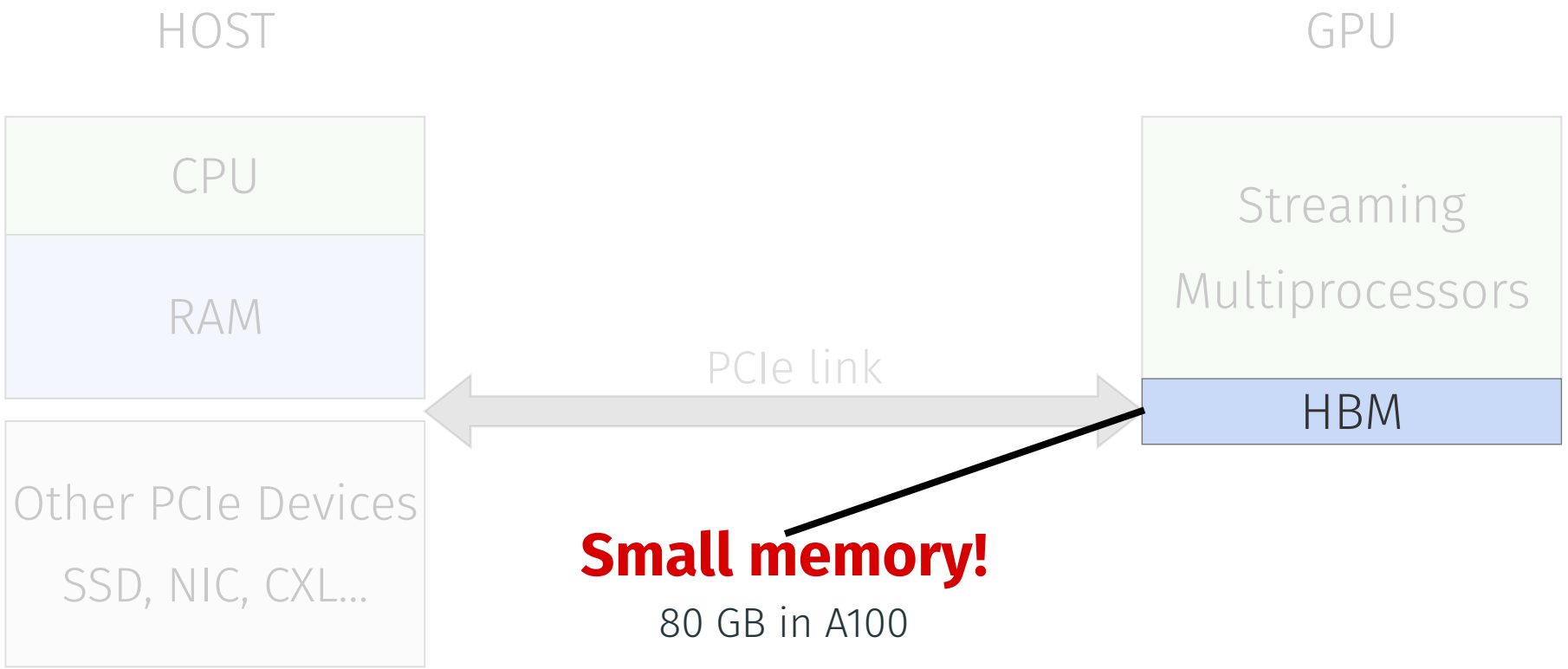
GPU System Primitives



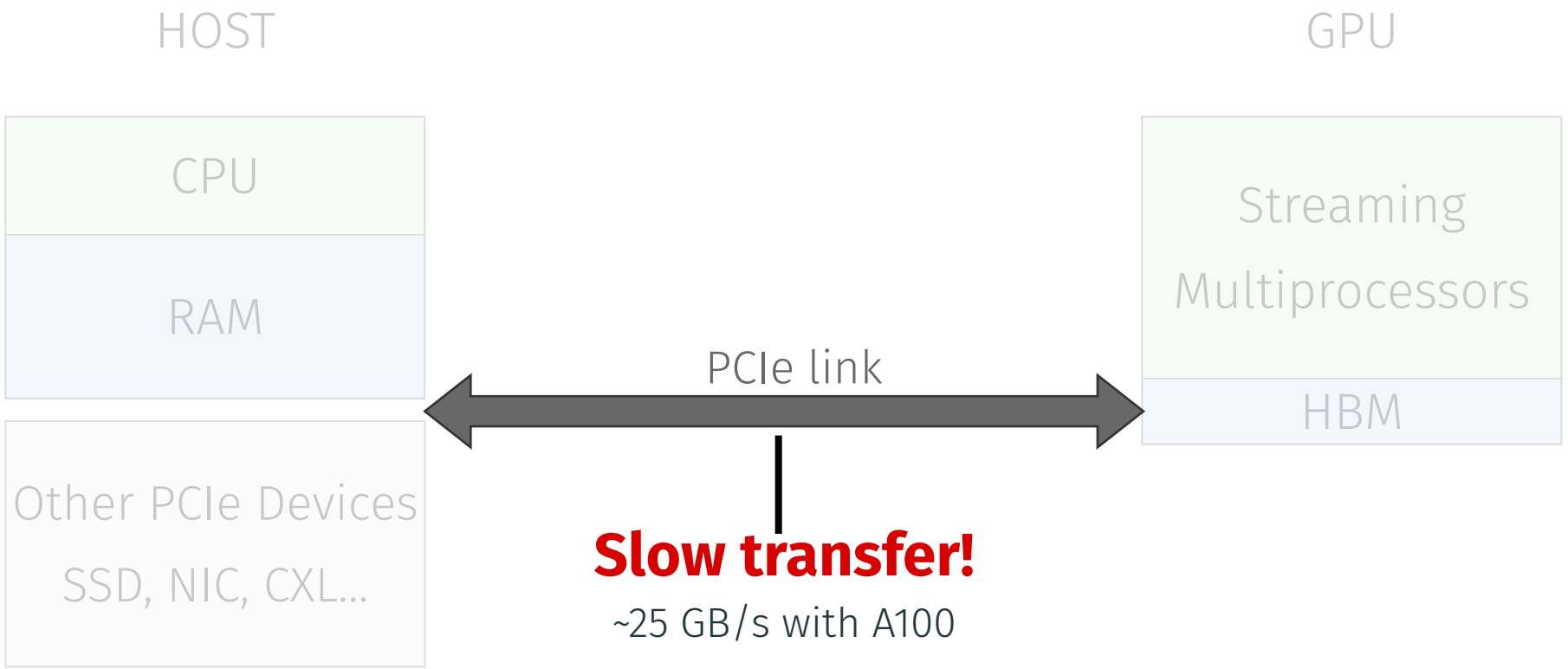
GPU System Primitives



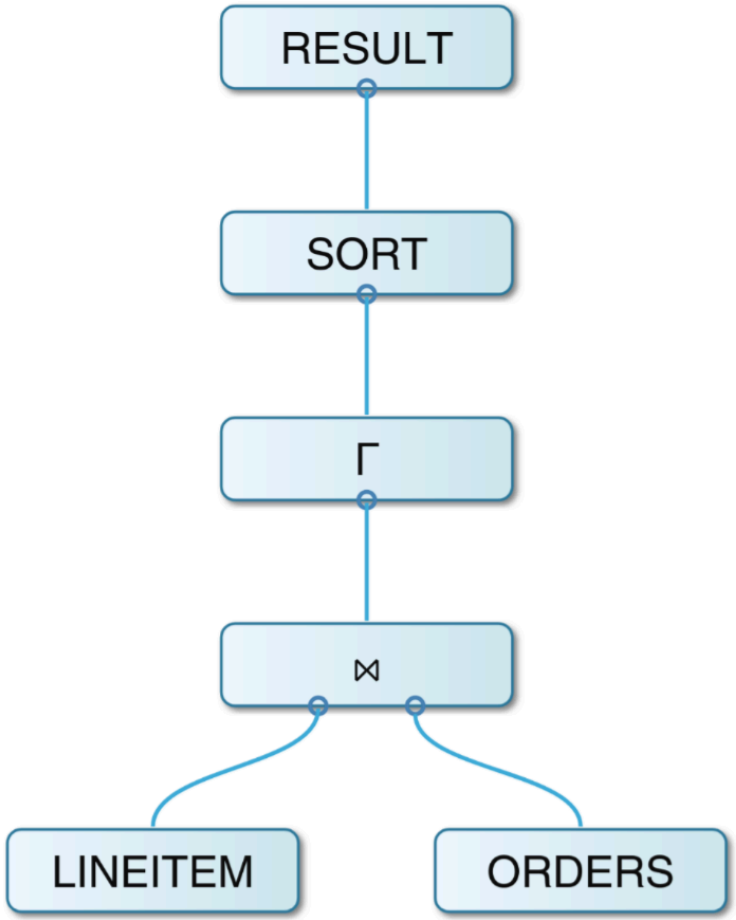
GPU System Primitives: Problems & Challenges



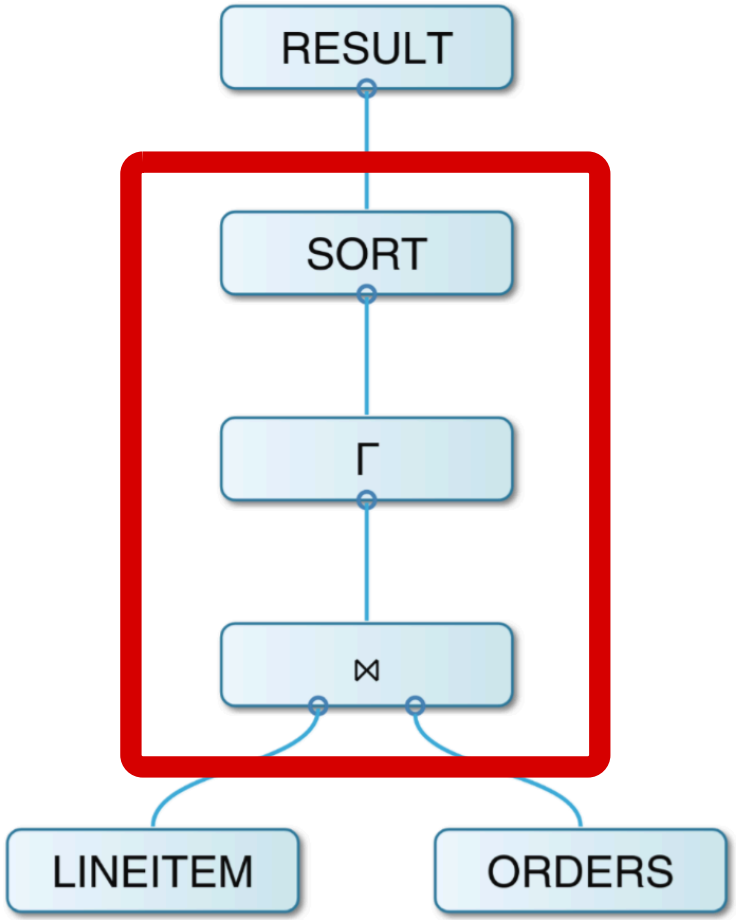
GPU System Primitives: Problems & Challenges



GPU Data Systems

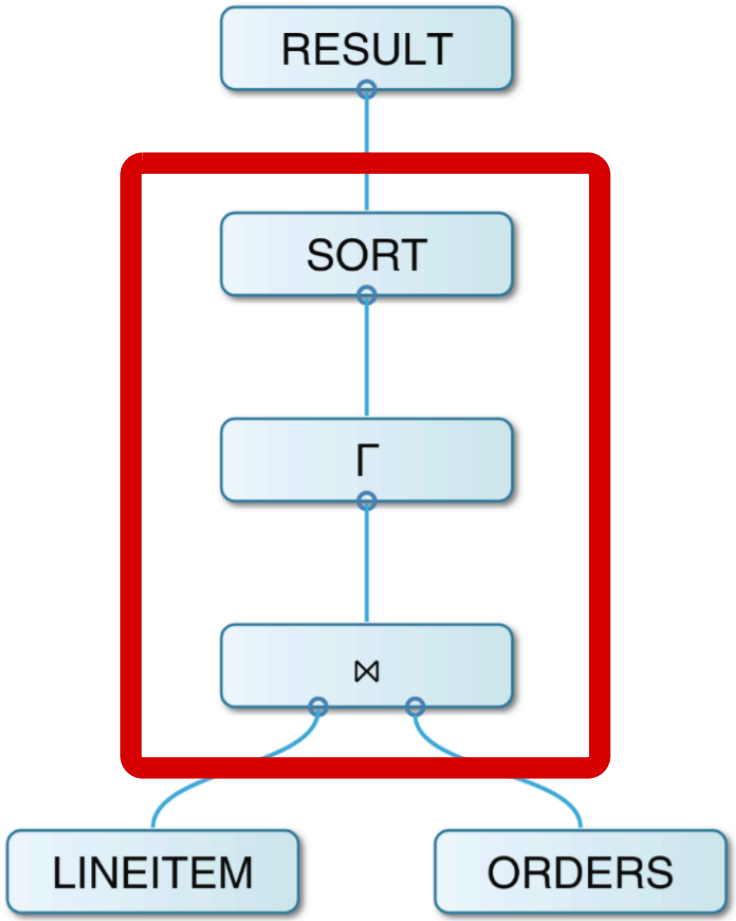


GPU Data Systems



1. join kernel
2. groupby kernel
3. sort kernel

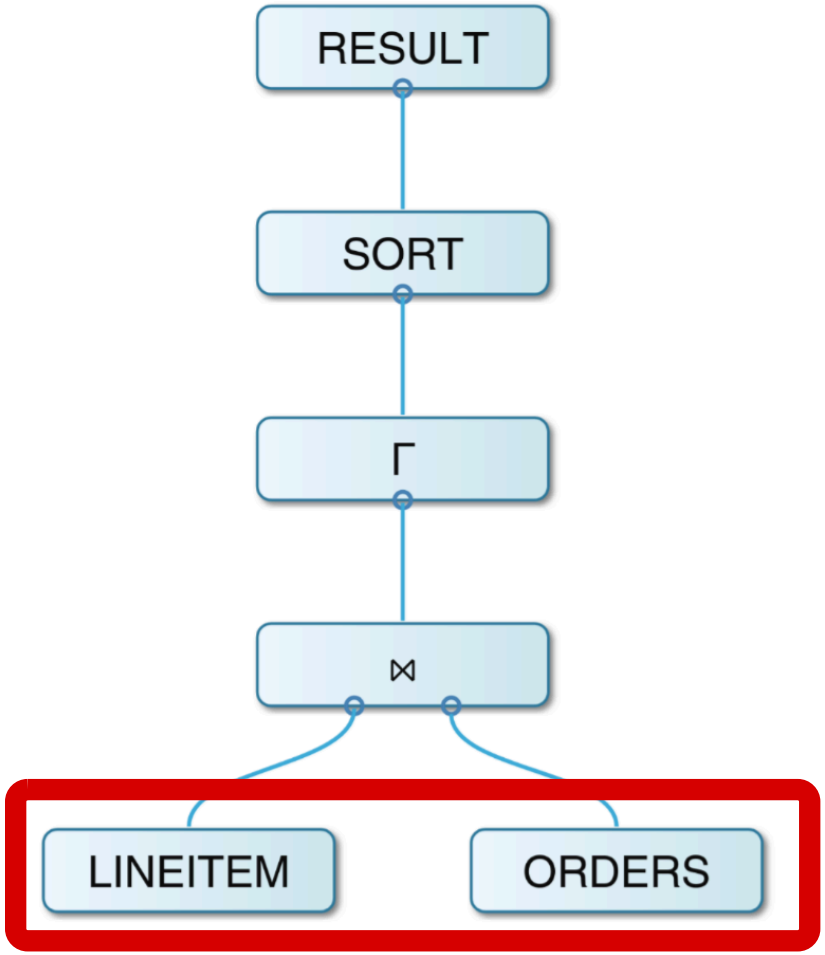
GPU Data Systems



1. join kernel
2. groupby kernel
3. sort kernel

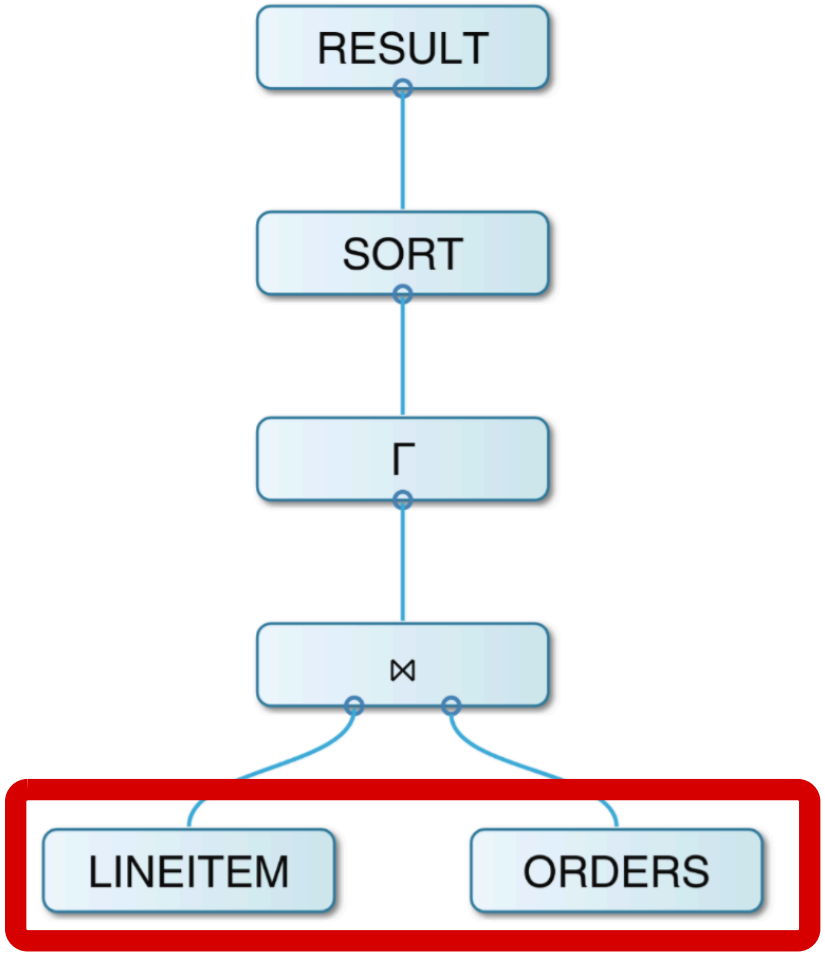
Not in this talk

GPU Data Systems



Where to put base tables?

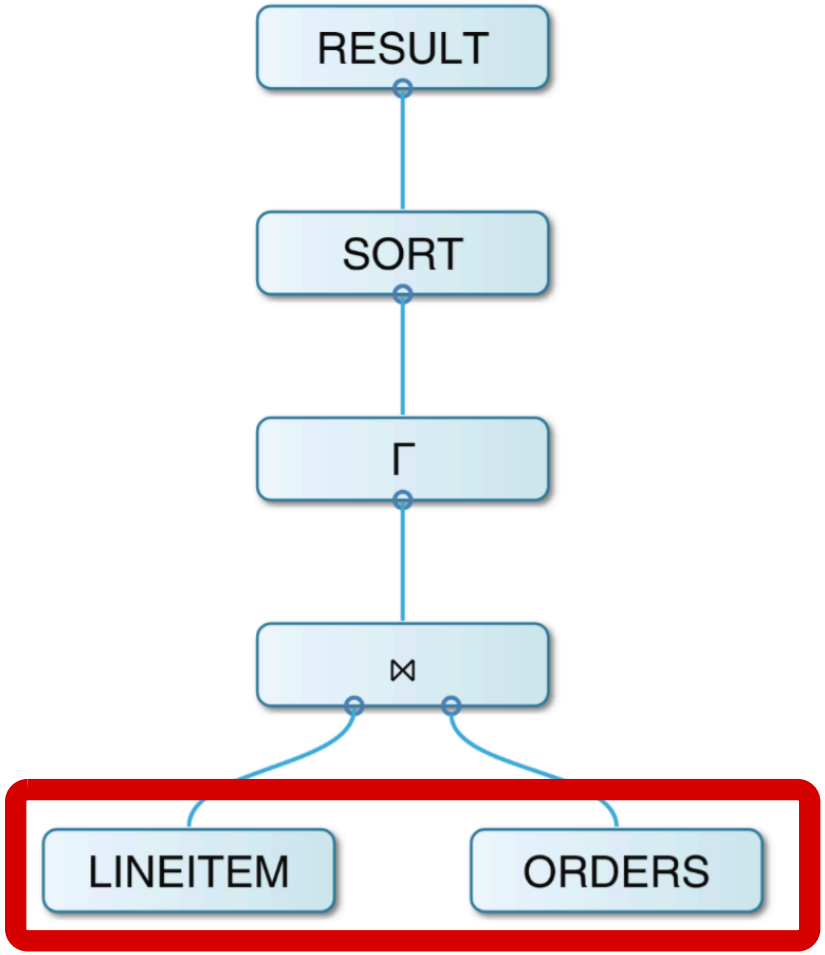
GPU Data Systems



Where to put base tables?

- GPU Memory: HBM

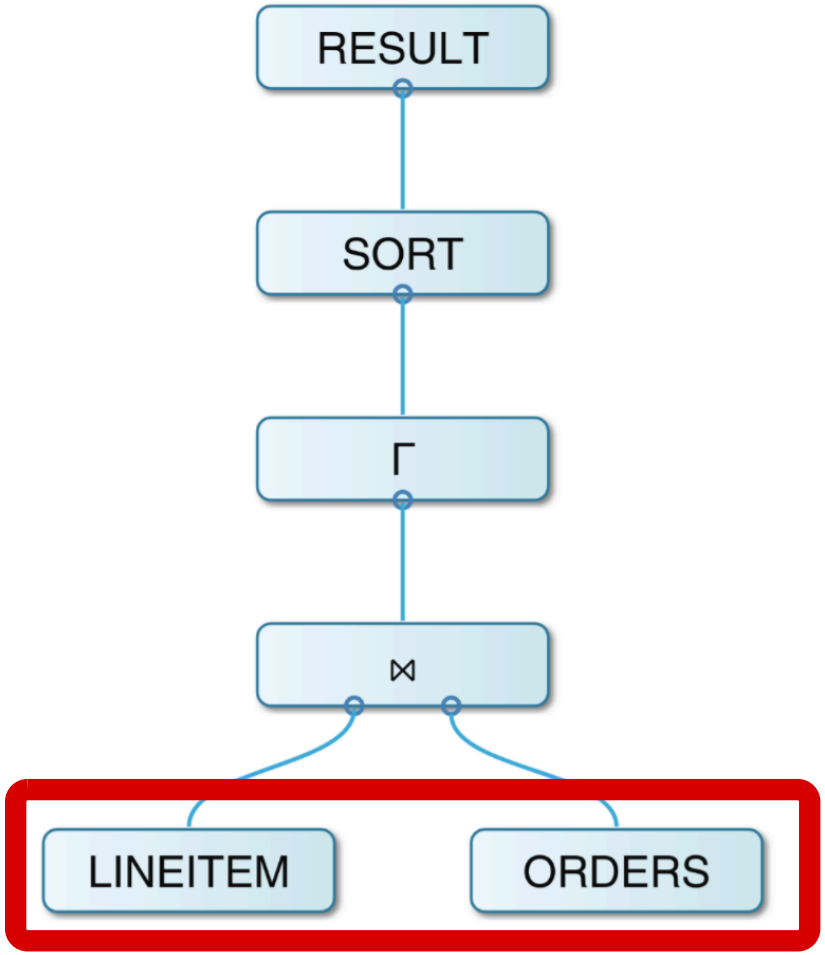
GPU Data Systems



Where to put base tables?

- GPU Memory: HBM
- CPU Memory

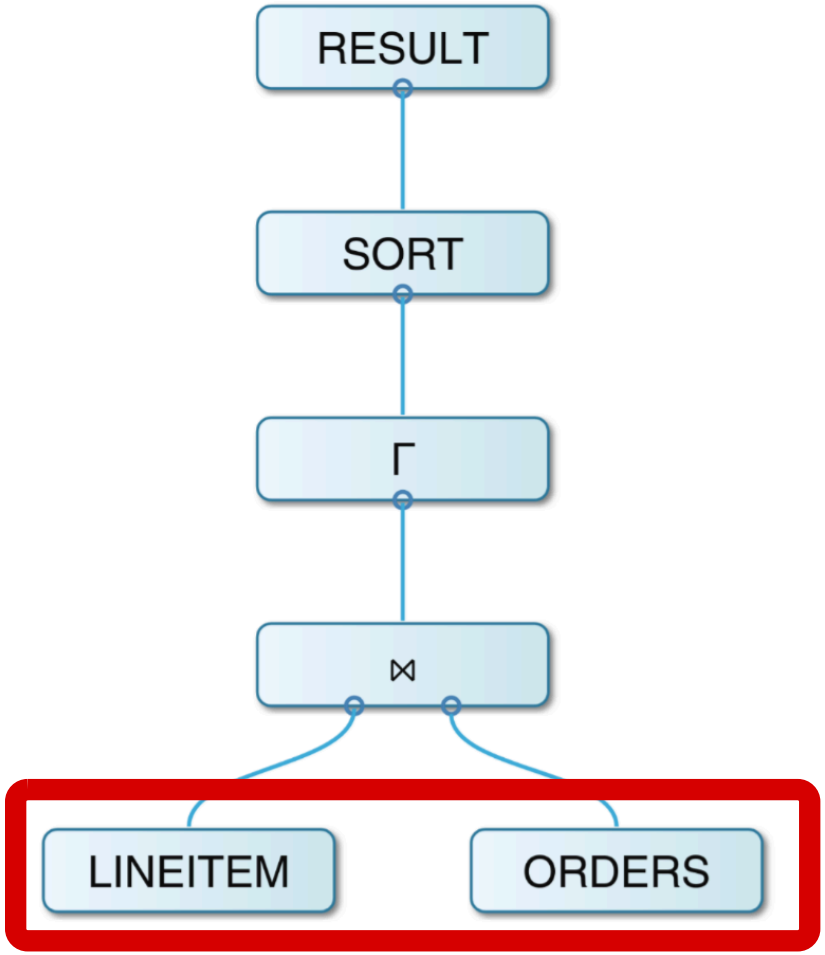
GPU Data Systems



Where to put base tables?

- GPU Memory: HBM
- CPU Memory
- SSDs

GPU Data Systems



Where to put base tables?

- GPU Memory: HBM
- CPU Memory
- SSDs
- NICs: remote storage

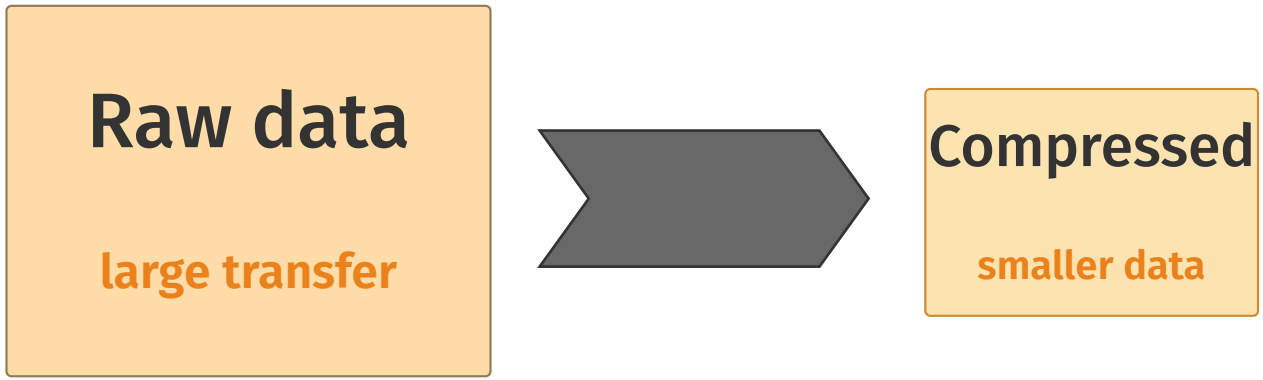
Figure Source: HyPer DB

Common Technique 1: Compression

Raw data

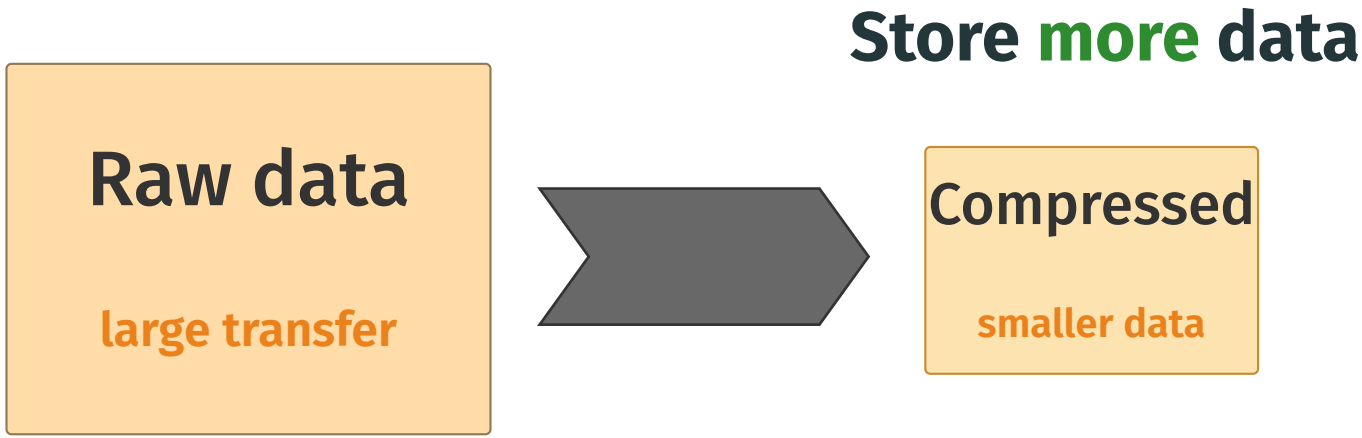
large transfer

Common Technique 1: Compression



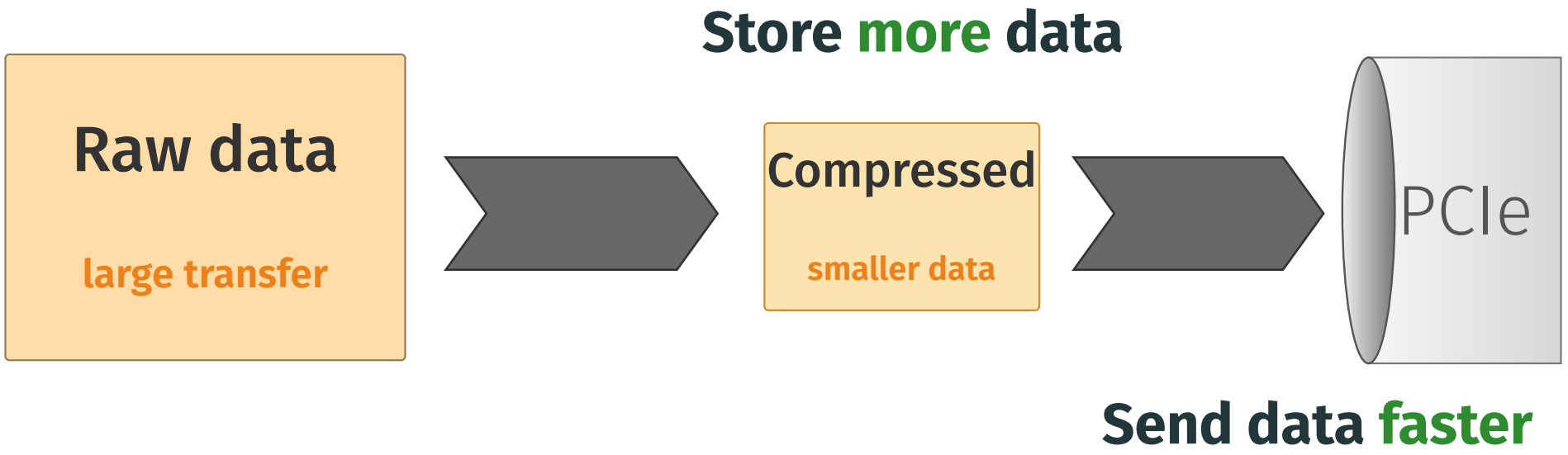
* Including heavyweight general compression and also lightweight encodings.

Common Technique 1: Compression



* Including heavyweight general compression and also lightweight encodings.

Common Technique 1: Compression



* Including heavyweight general compression and also lightweight encodings.

Common Technique 2: Overlapping

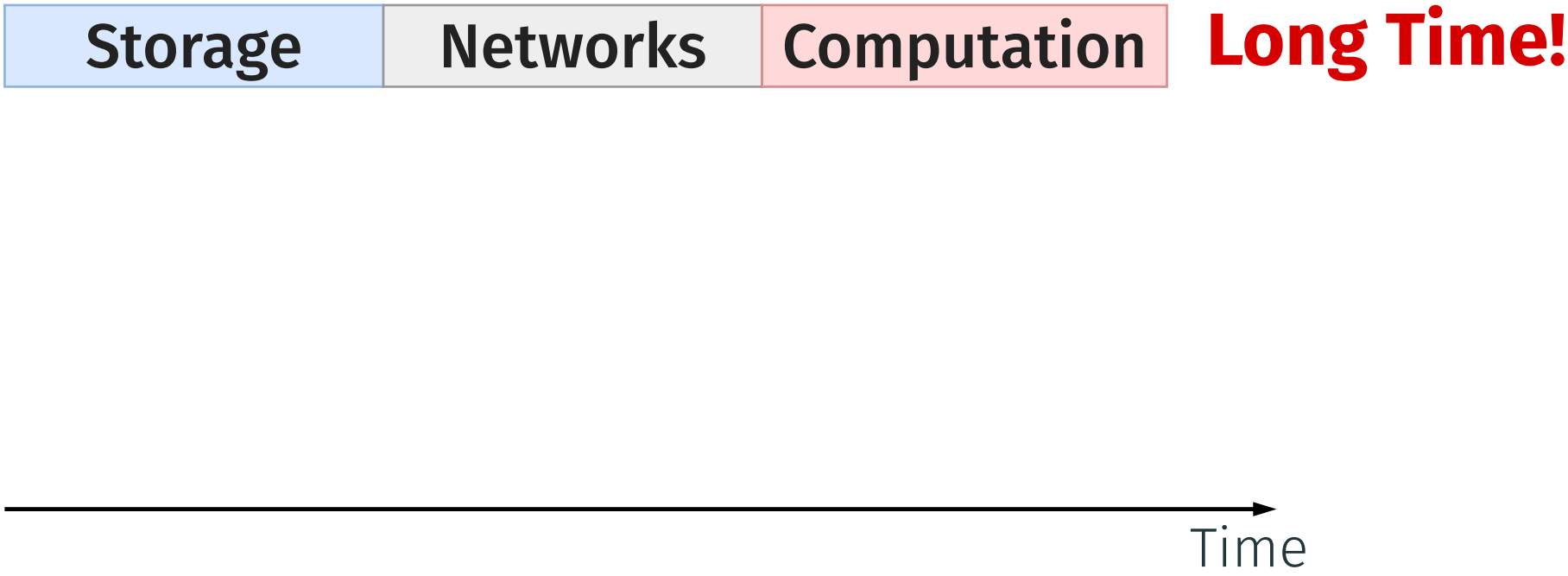
Blocking



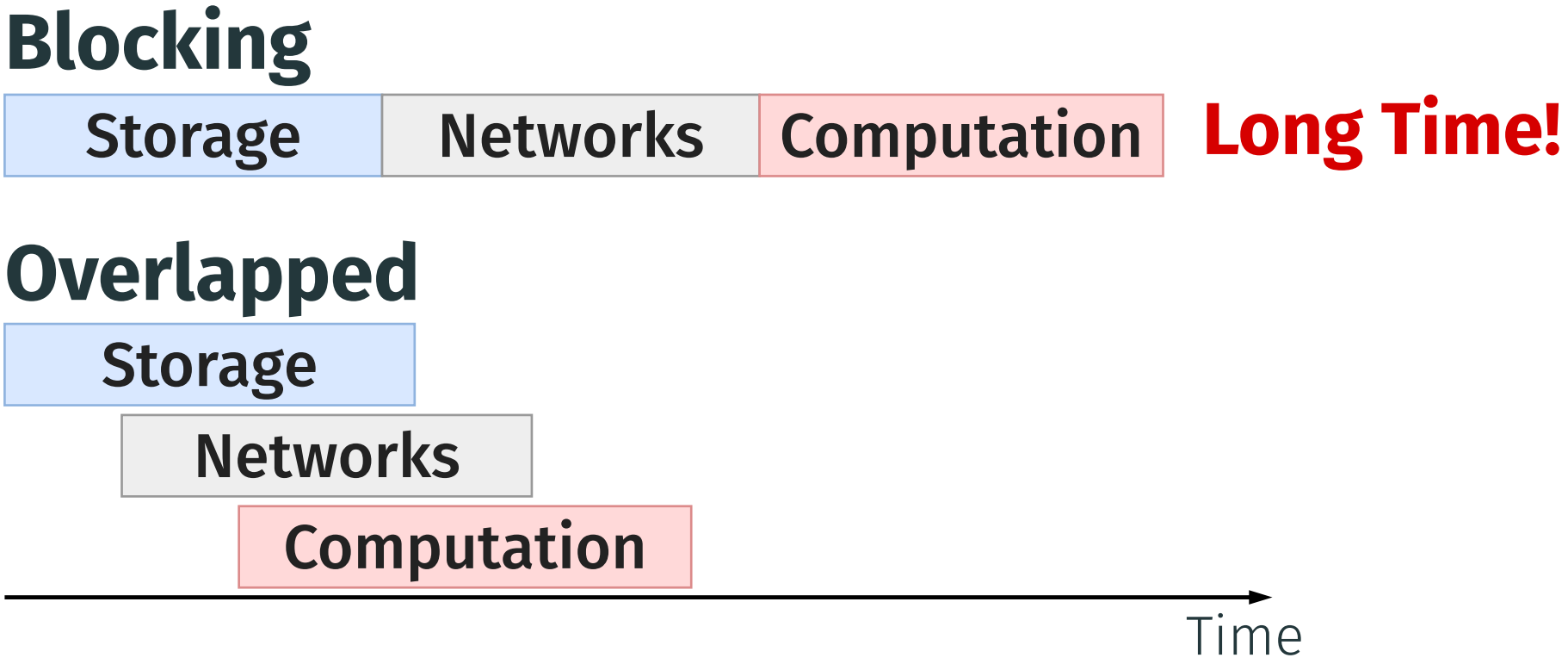


Common Technique 2: Overlapping

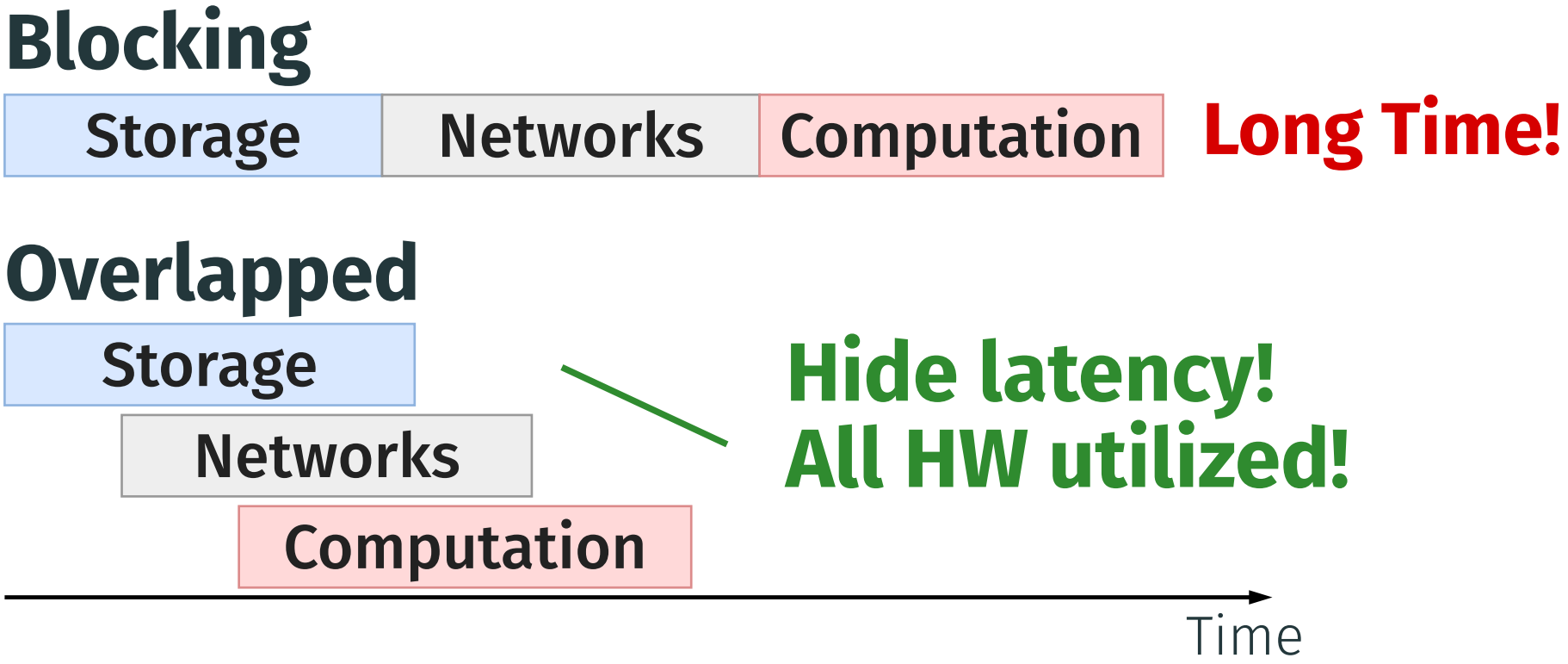
Blocking



Common Technique 2: Overlapping



Common Technique 2: Overlapping



1. GPU File Format

DaMoN 2026 🏆

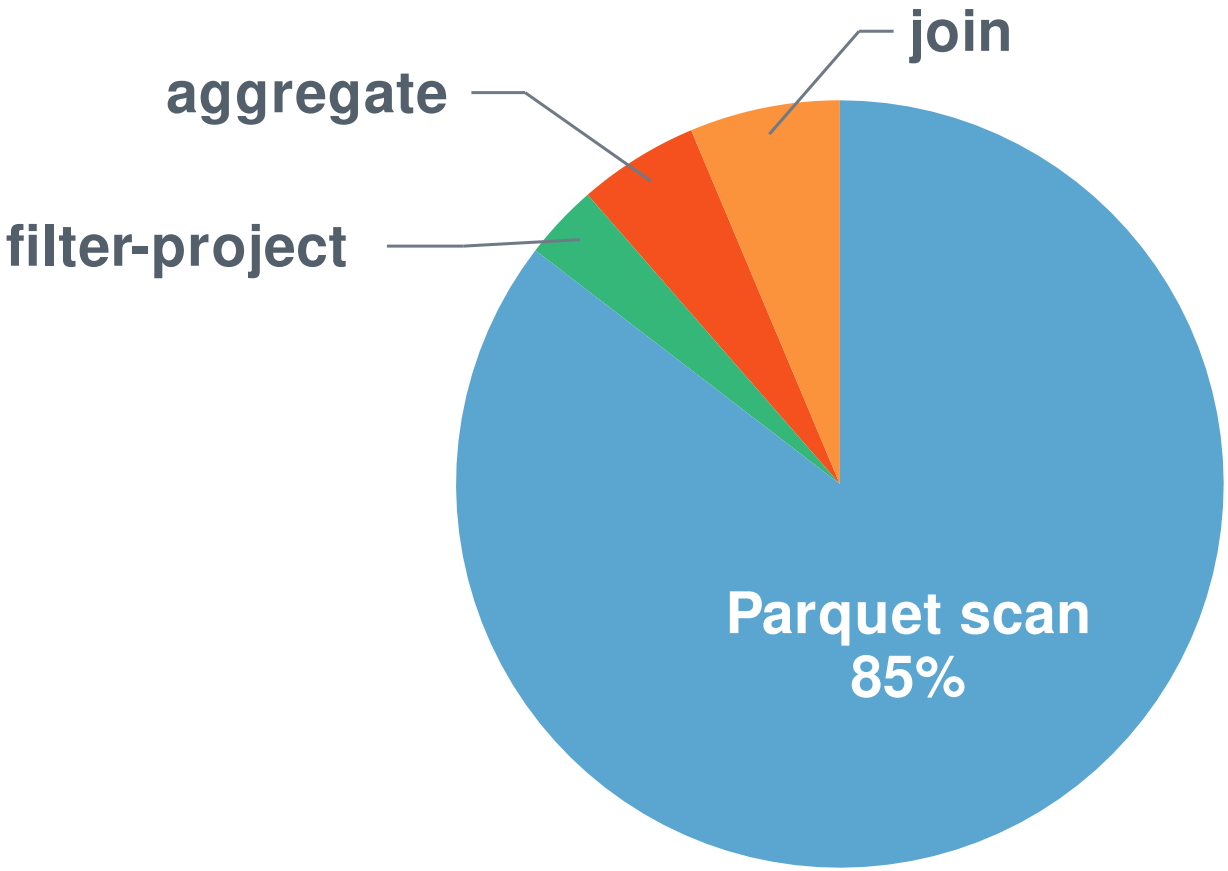
Problem: GPU Bottleneck With Parquet

Problem: GPU Bottleneck With Parquet

Is Parquet Bad for GPUs?

Problem: GPU Bottleneck With Parquet

Is Parquet Bad for GPUs?



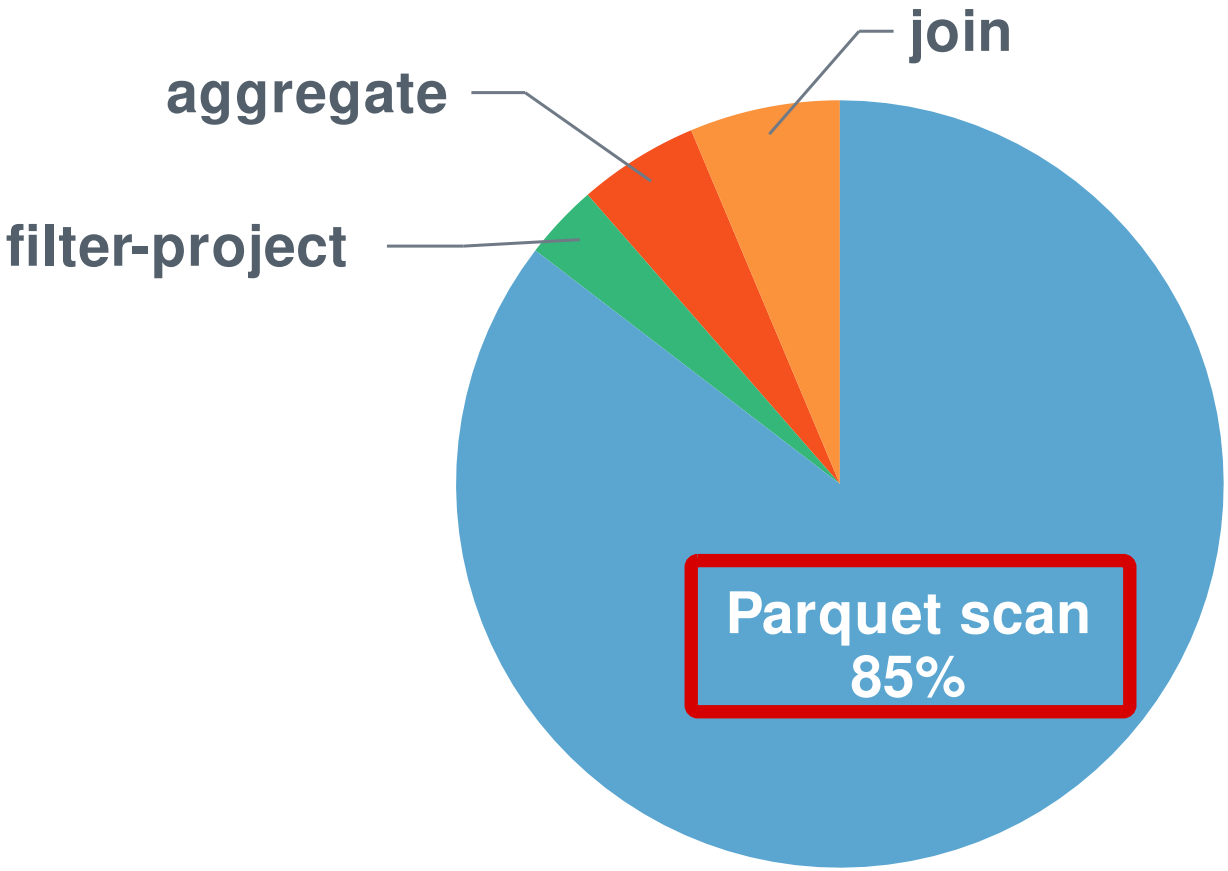
GPU TPC-H SF100 runtime breakdown.

Source: Accelerating Velox with RAPIDS cuDF, Velox-cuDF Team, 2025

Problem: GPU Bottleneck With Parquet

Is Parquet Bad for GPUs?

- **85%** in Parquet Scan



GPU TPC-H SF100 runtime breakdown.

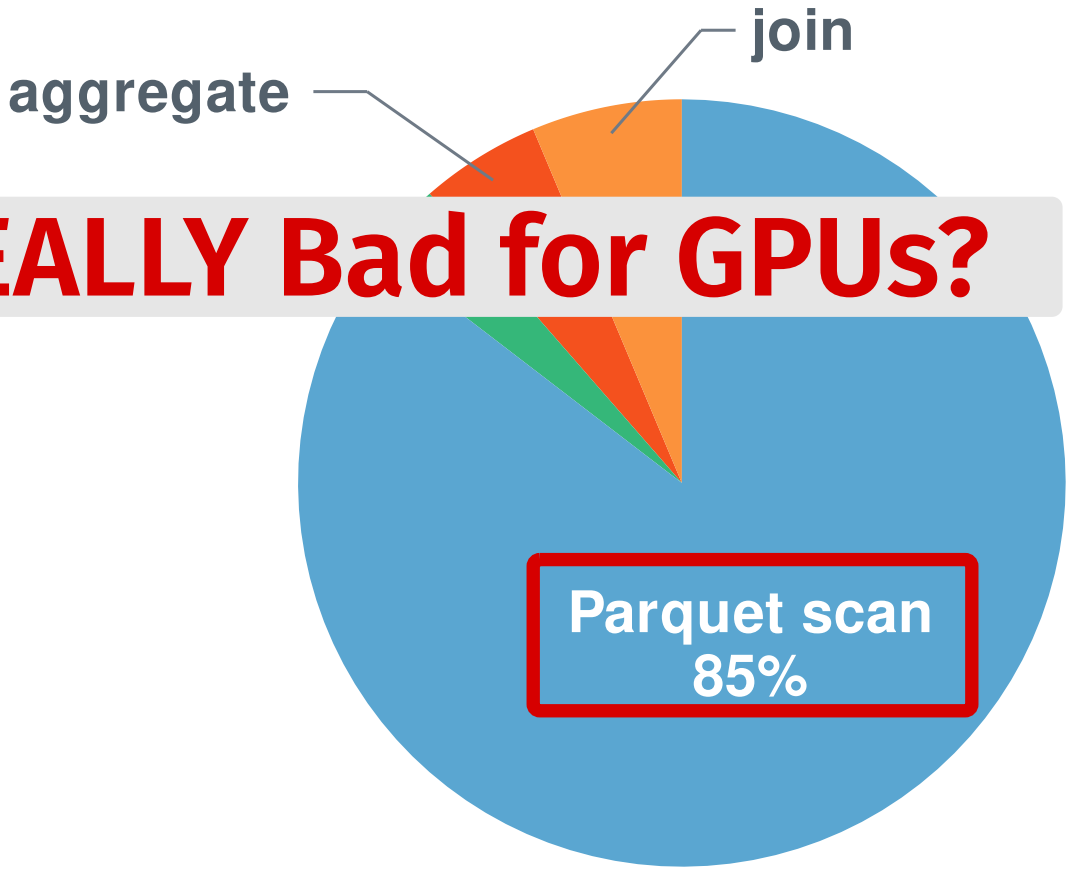
Source: Accelerating Velox with RAPIDS cuDF, Velox-cuDF Team, 2025

Problem: GPU Bottleneck With Parquet

Is Parquet REALLY Bad for GPUs?

Is Parquet Bad for GPUs?

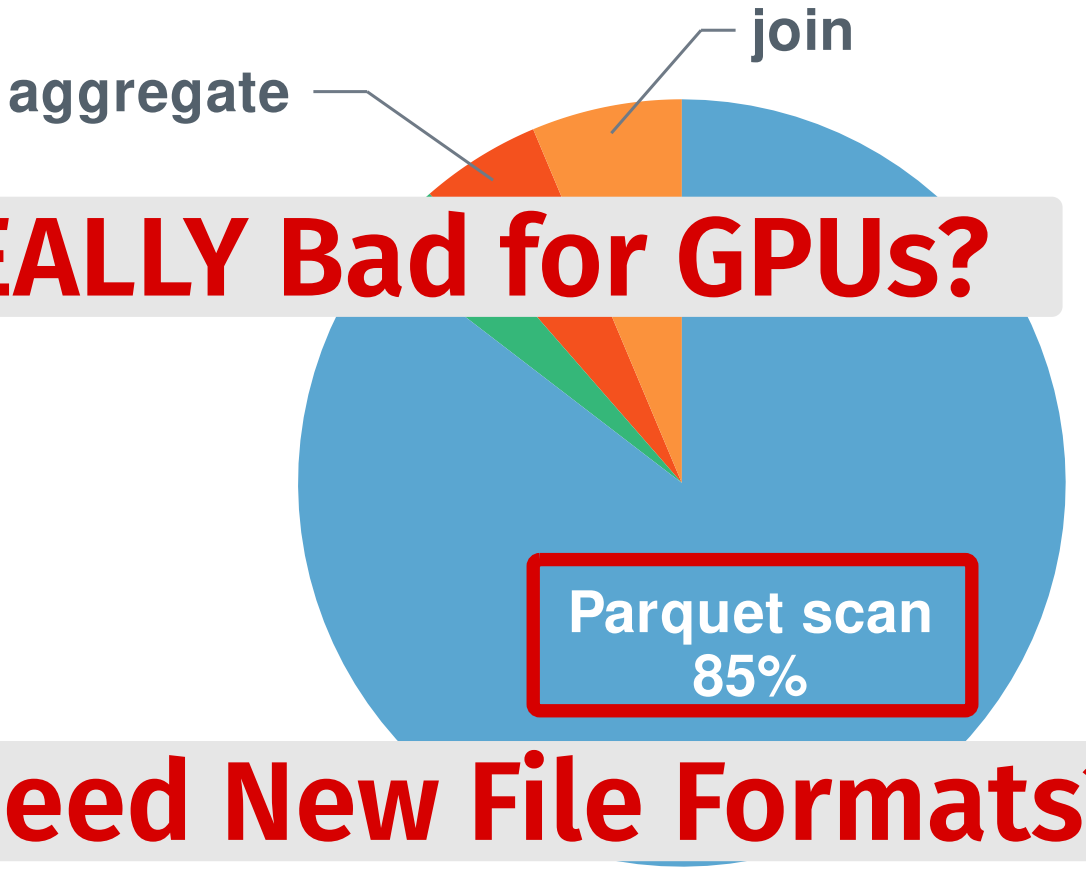
- **85%** in Parquet Scan



GPU TPC-H SF100 runtime breakdown.

Source: Accelerating Velox with RAPIDS cuDF, Velox-cuDF Team, 2025

Problem: GPU Bottleneck With Parquet



Is Parquet REALLY Bad for GPUs?

Is Parquet Bad for GPUs?

- **85%** in Parquet Scan

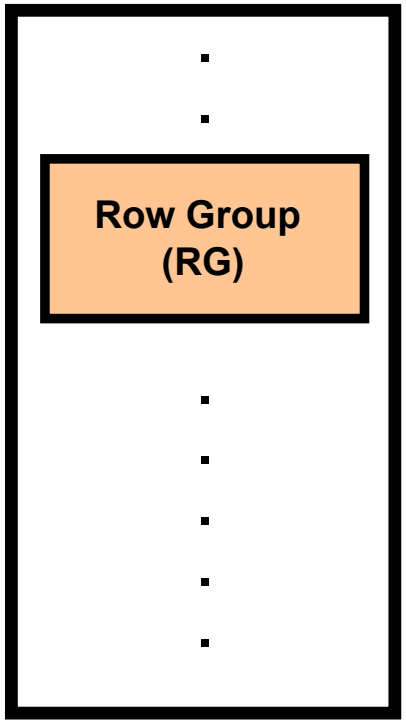
Do GPUs REALLY Need New File Formats?

GPU TPC-H SF100 runtime breakdown.

Source: Accelerating Velox with RAPIDS cuDF, Velox-cuDF Team, 2025

Background: Apache Parquet

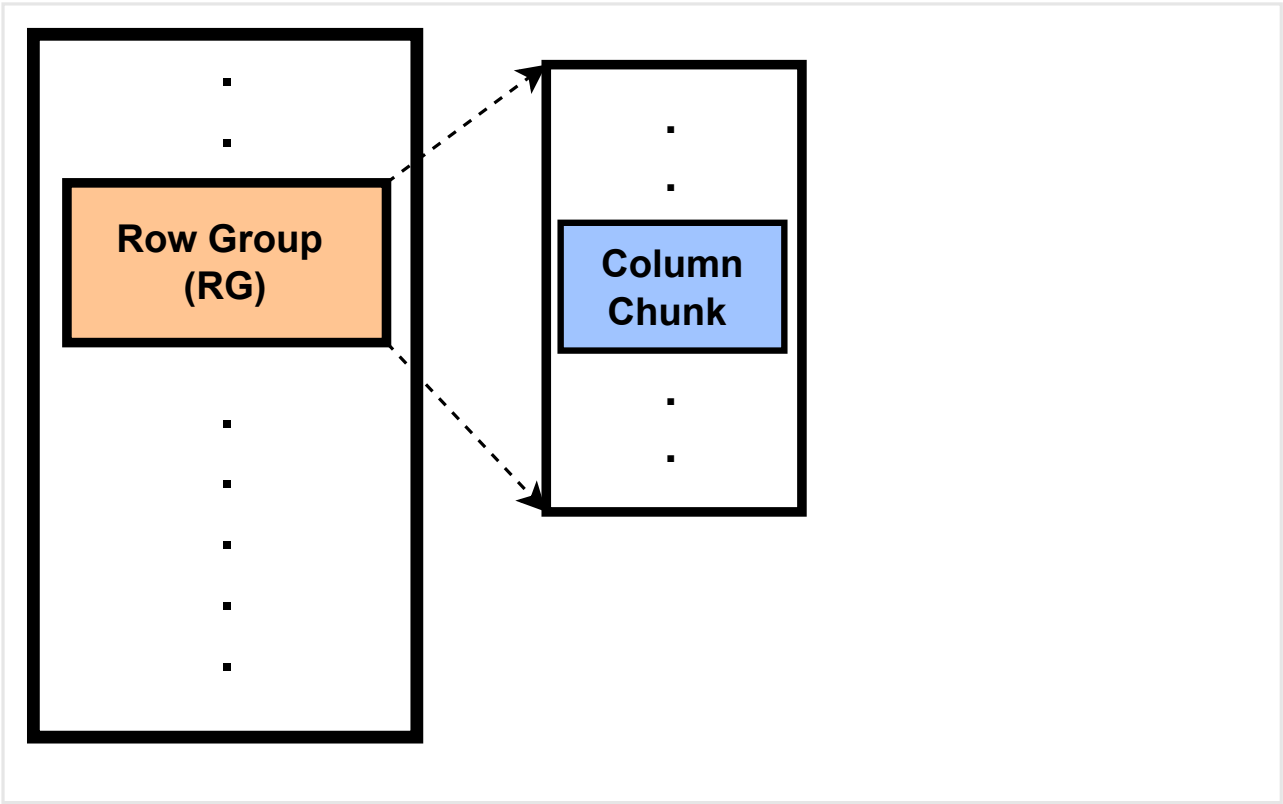
- Row group



Parquet

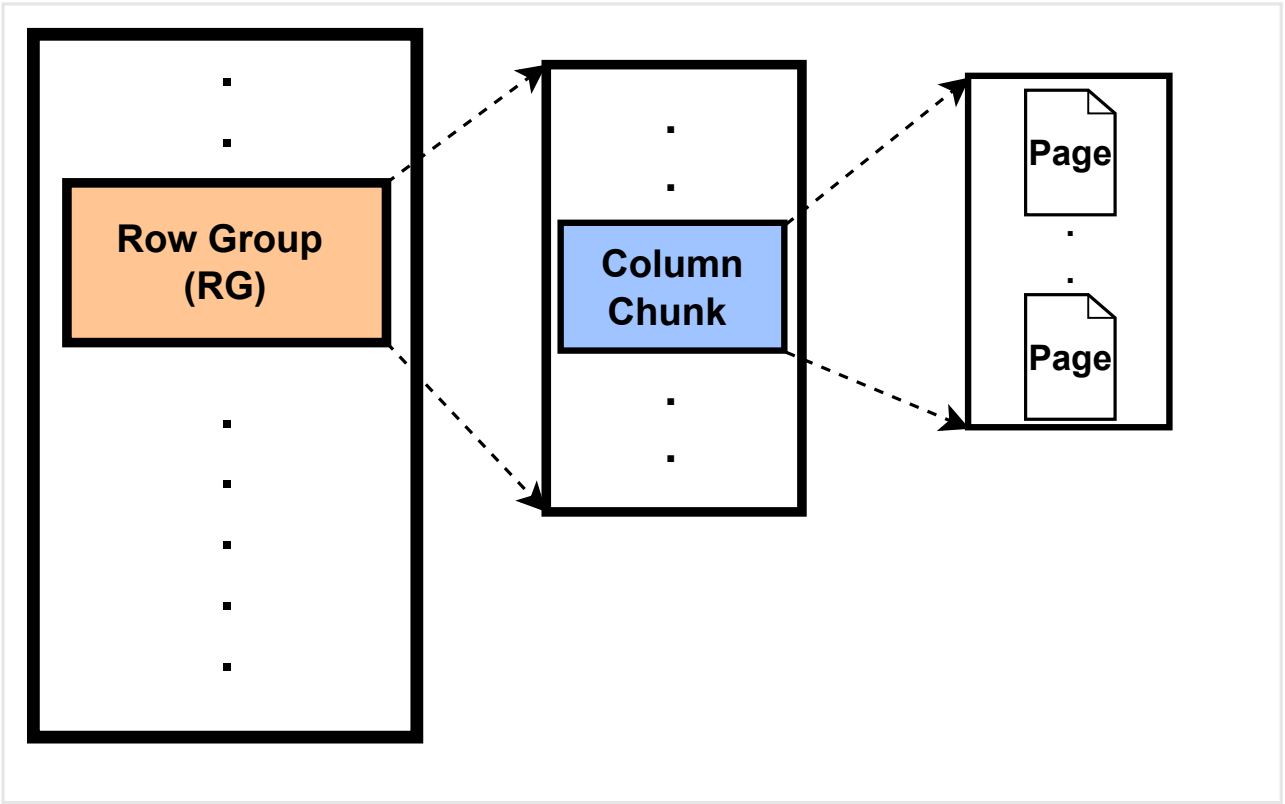
Background: Apache Parquet

- Row group
- Column chunk



Background: Apache Parquet

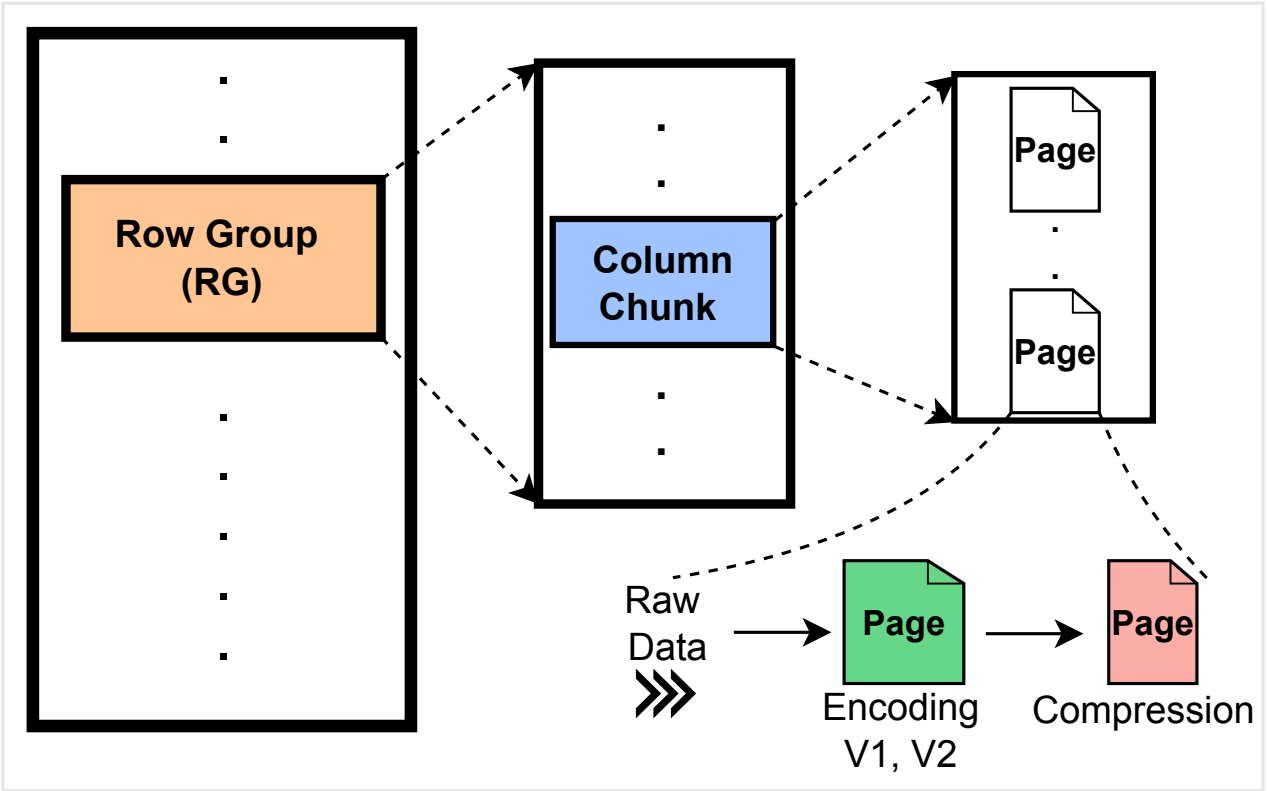
- Row group
- Column chunk
- Pages:



Parquet

Background: Apache Parquet

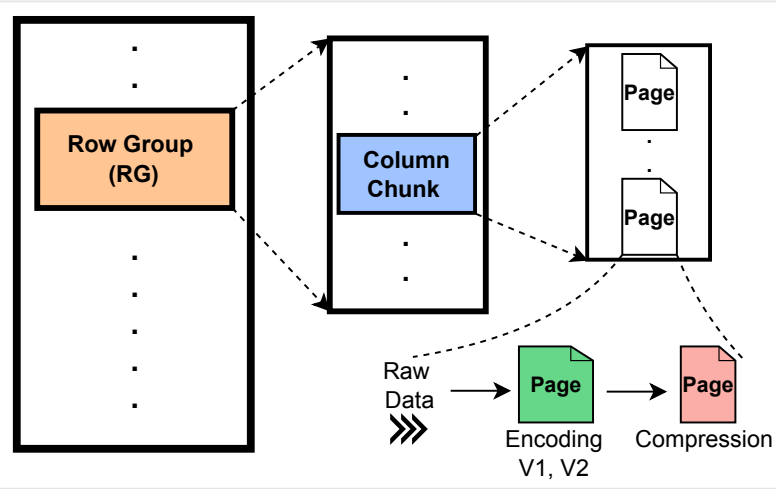
- Row group
- Column chunk
- Pages:
  - Encoded
  - Compressed



Parquet

Parquet Configuration Defaults for CPUs

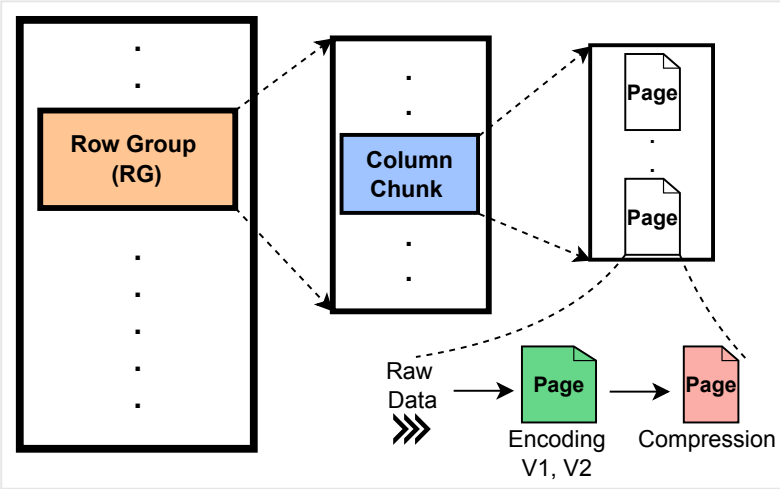
CPU Parquet Config. Defaults in DuckDB:



Parquet Configuration Defaults for CPUs

CPU Parquet Config. Defaults in DuckDB:

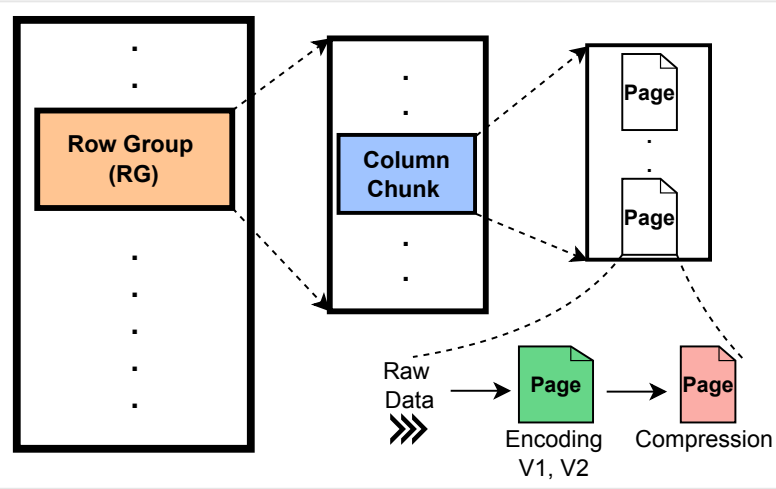
- Row group: 122880 rows



Parquet Configuration Defaults for CPUs

CPU Parquet Config. Defaults in DuckDB:

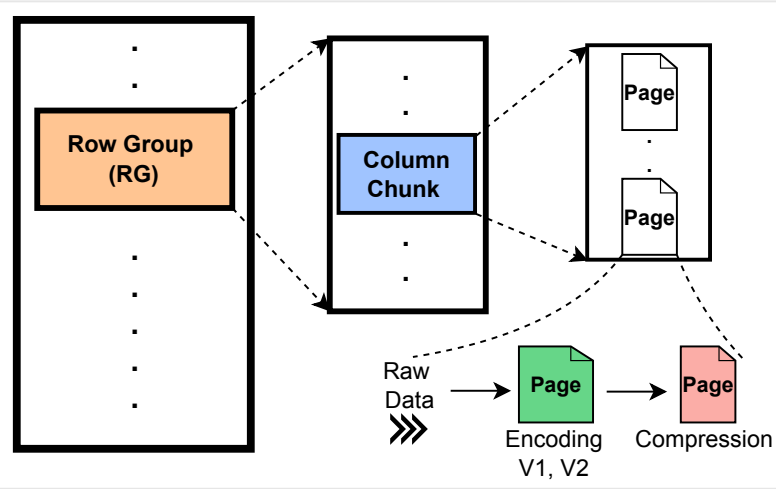
- Row group: 122880 rows
- Column chunk: 1 page only



Parquet Configuration Defaults for CPUs

CPU Parquet Config. Defaults in DuckDB:

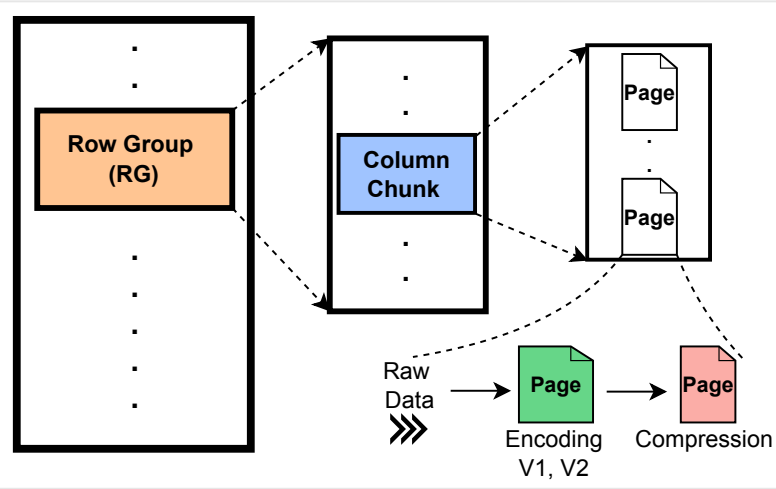
- Row group: 122880 rows
- Column chunk: 1 page only
- Encoding: V1 only



Parquet Configuration Defaults for CPUs

CPU Parquet Config. Defaults in **DuckDB**:

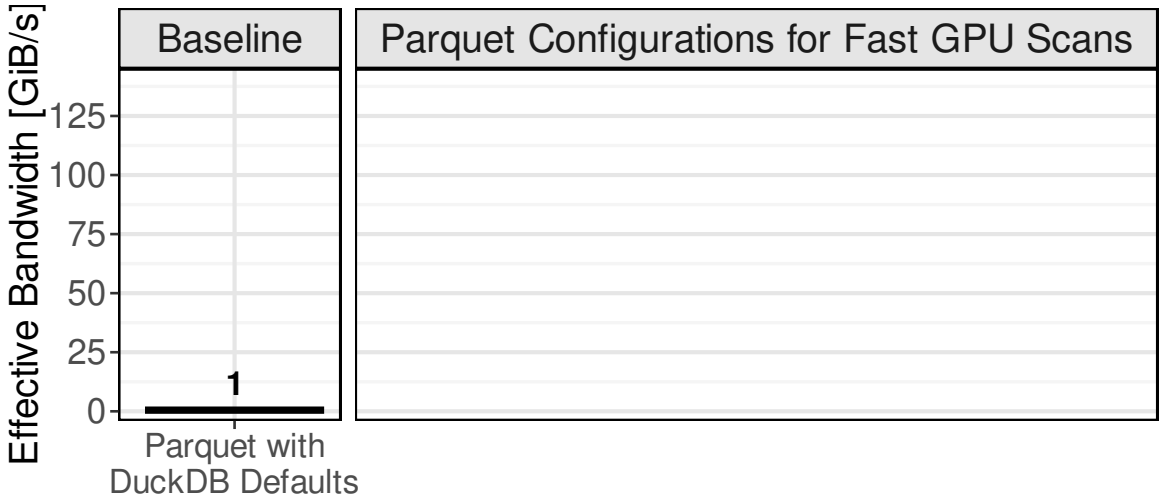
- Row group: 122880 rows
- Column chunk: 1 page only
- Encoding: V1 only
- Compression: used even without size reduction



Issue: CPU Defaults Config. ≠ GPU Optimal Config.

CPU Parquet Config. Defaults in DuckDB:

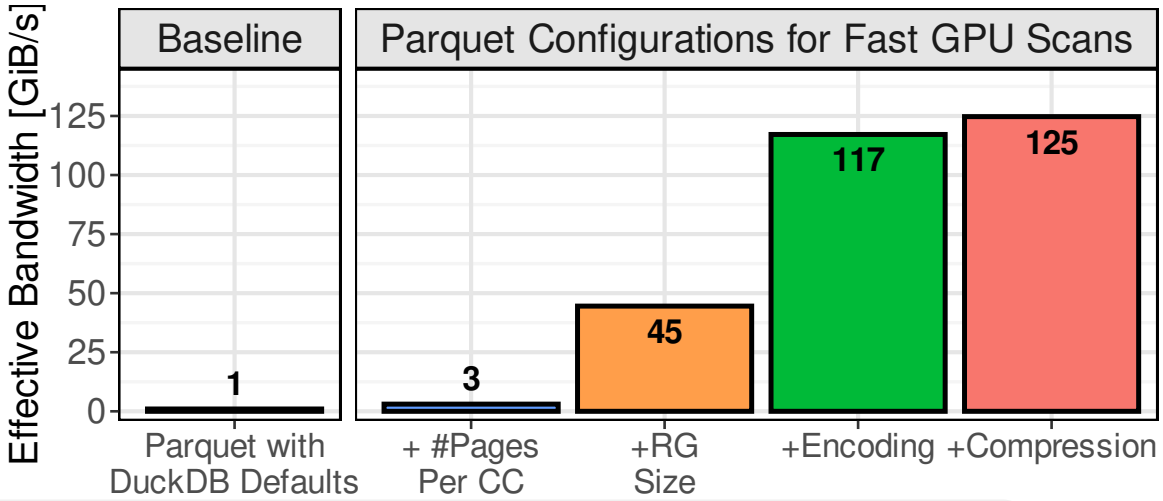
- Row group: 122880 rows
- Column chunk: 1 page only
- Encoding: V1 only
- Compression: used even without size reduction



Issue: CPU Defaults Config. ≠ GPU Optimal Config.

CPU Parquet Config. Defaults in DuckDB:

- Row group: 122880 rows
- Column chunk: 1 page only
- Encoding: V1 only
- Compression: used even without size reduction

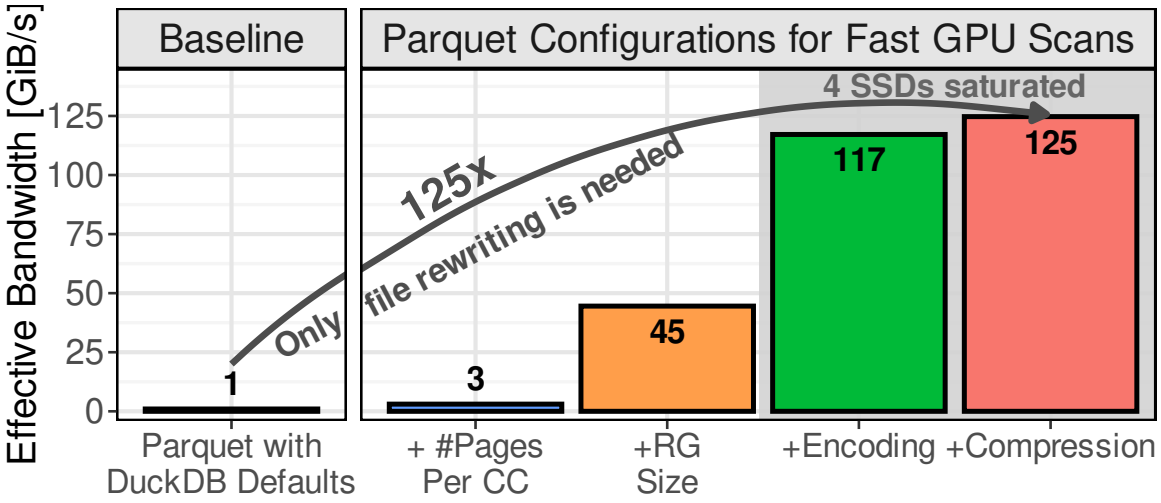


CPU Defaults Config. ≠ GPU Optimal Config.

Issue: CPU Defaults Config. ≠ GPU Optimal Config.

CPU Parquet Config. Defaults in DuckDB:

- Row group: 122880 rows
- Column chunk: 1 page only
- Encoding: V1 only
- Compression: used even without size reduction



2. Distributed GPU Engine *PystachIO*

VLDB 2026

What is missing in current GPU DBs?

	CUDA-X Lib
GPU Compute	✓
Storage	✓
Network	✓

What is missing in current GPU DBs?

	CUDA-X Lib	TQP
GPU Compute	✓	✓
Storage	✓	✗
Network	✓	✗

What is missing in current GPU DBs?

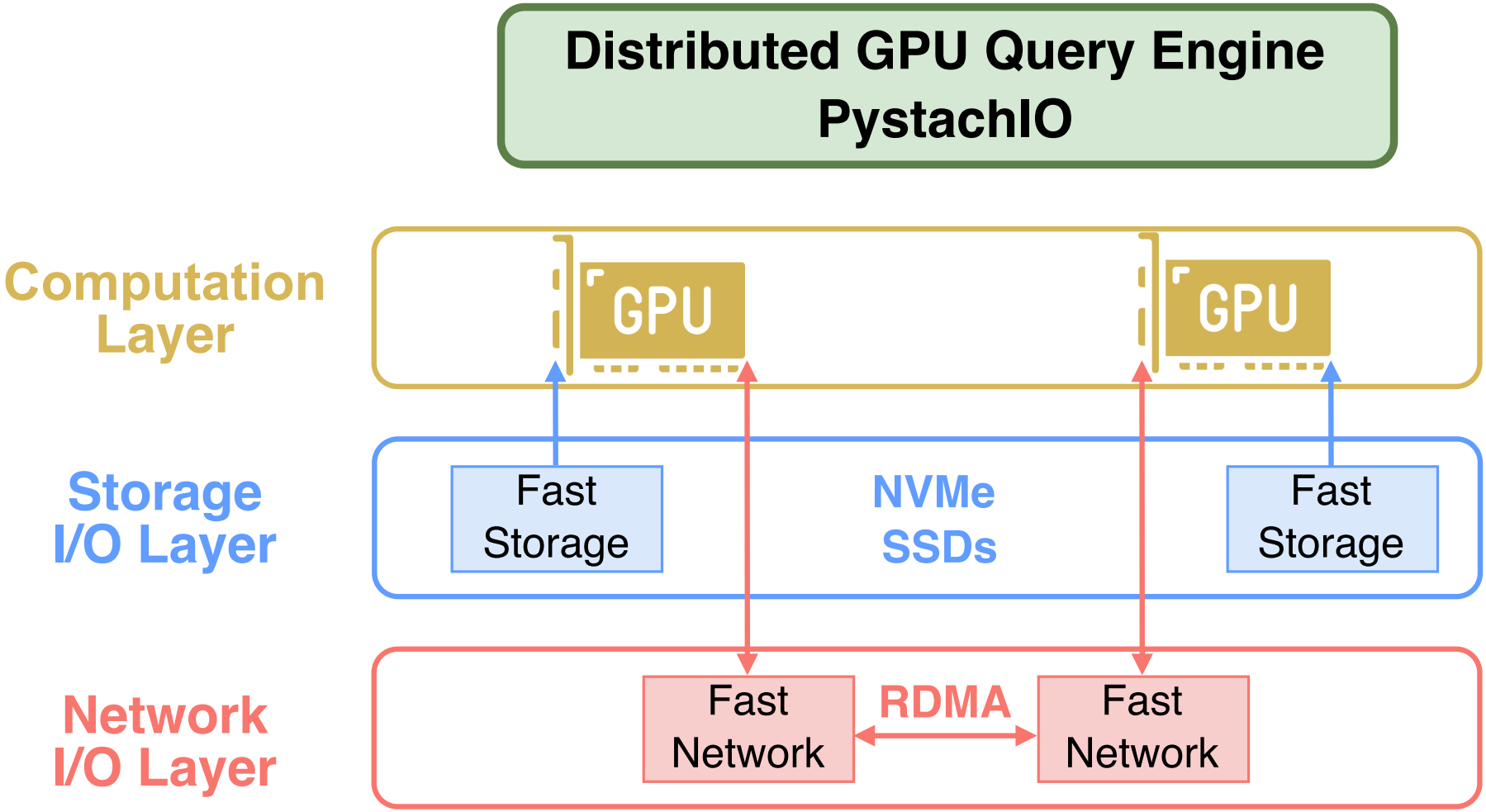
	CUDA-X Lib	TQP	Cloud OLAP
GPU Compute	✓	✓	✗
Storage	✓	✗	✓
Network	✓	✗	✓

What is missing in current GPU DBs?

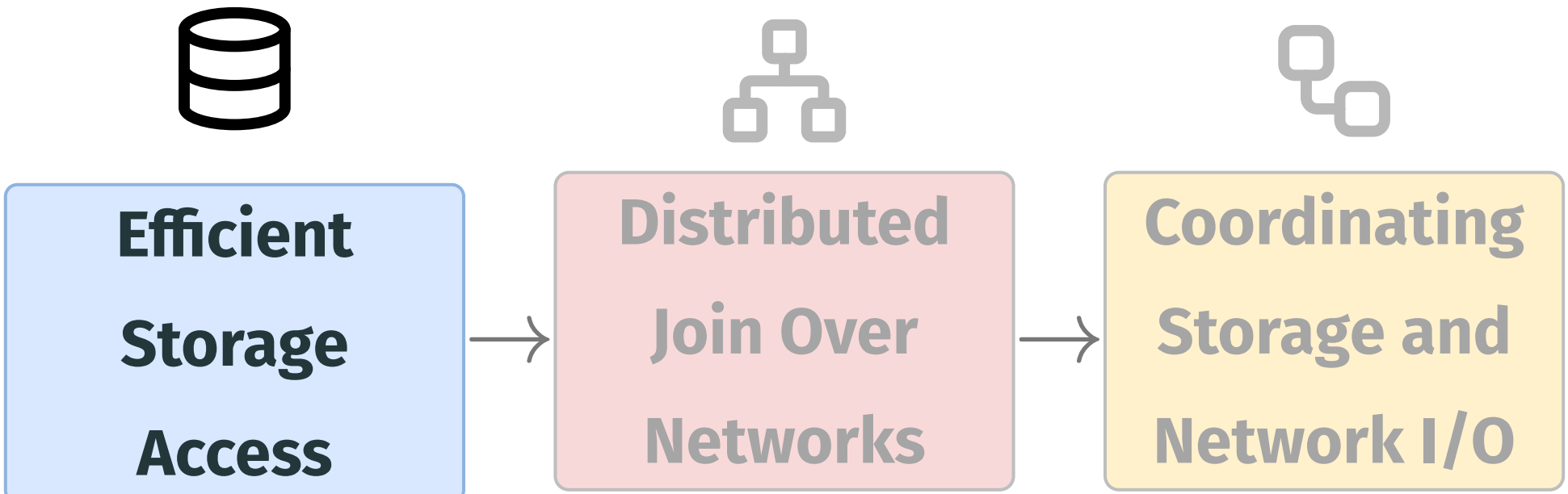
	CUDA-X Lib	TQP	Cloud OLAP
GI			
Network	✓	✗	✓

How to build a distributed GPU DB with storage & network?

PystachIO

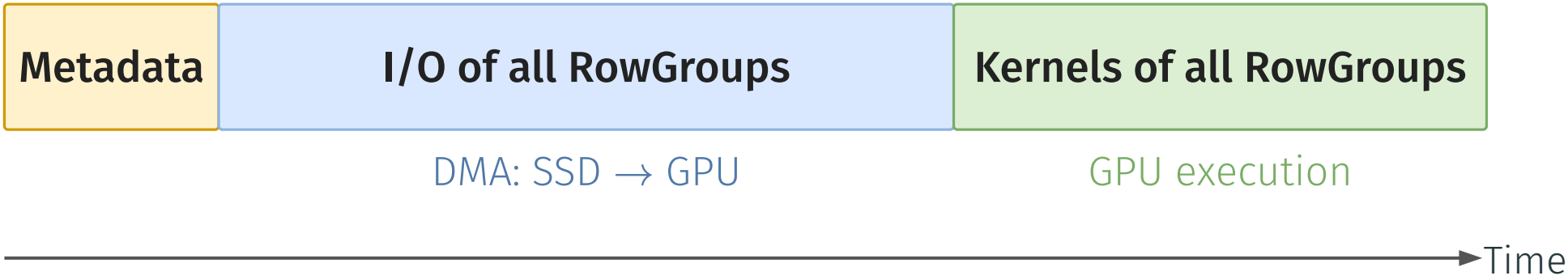

Distributed GPU Query Engine
PystachIO
Computation Layer
GPU
GPU
Storage I/O Layer
Fast Storage
NVMe SSDs
Fast Storage
Network I/O Layer
Fast Network
RDMA
Fast Network

PystachIO: Storage



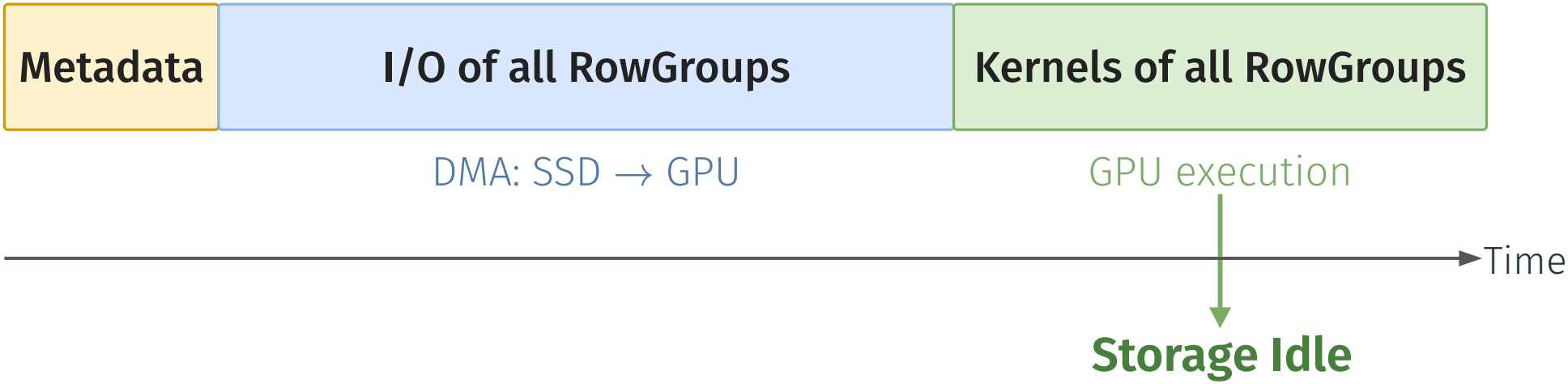
Inefficient Storage Access: Blocking Parquet Reader

`read_parquet()` in one CPU thread:



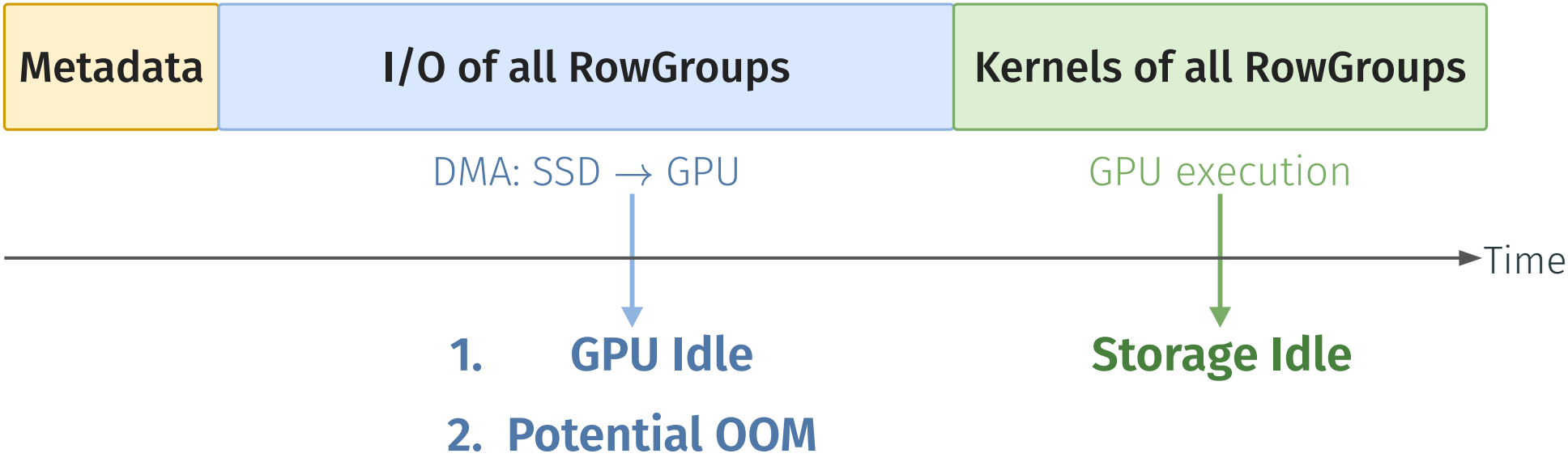
Inefficient Storage Access: Blocking Parquet Reader

`read_parquet()` in one CPU thread:

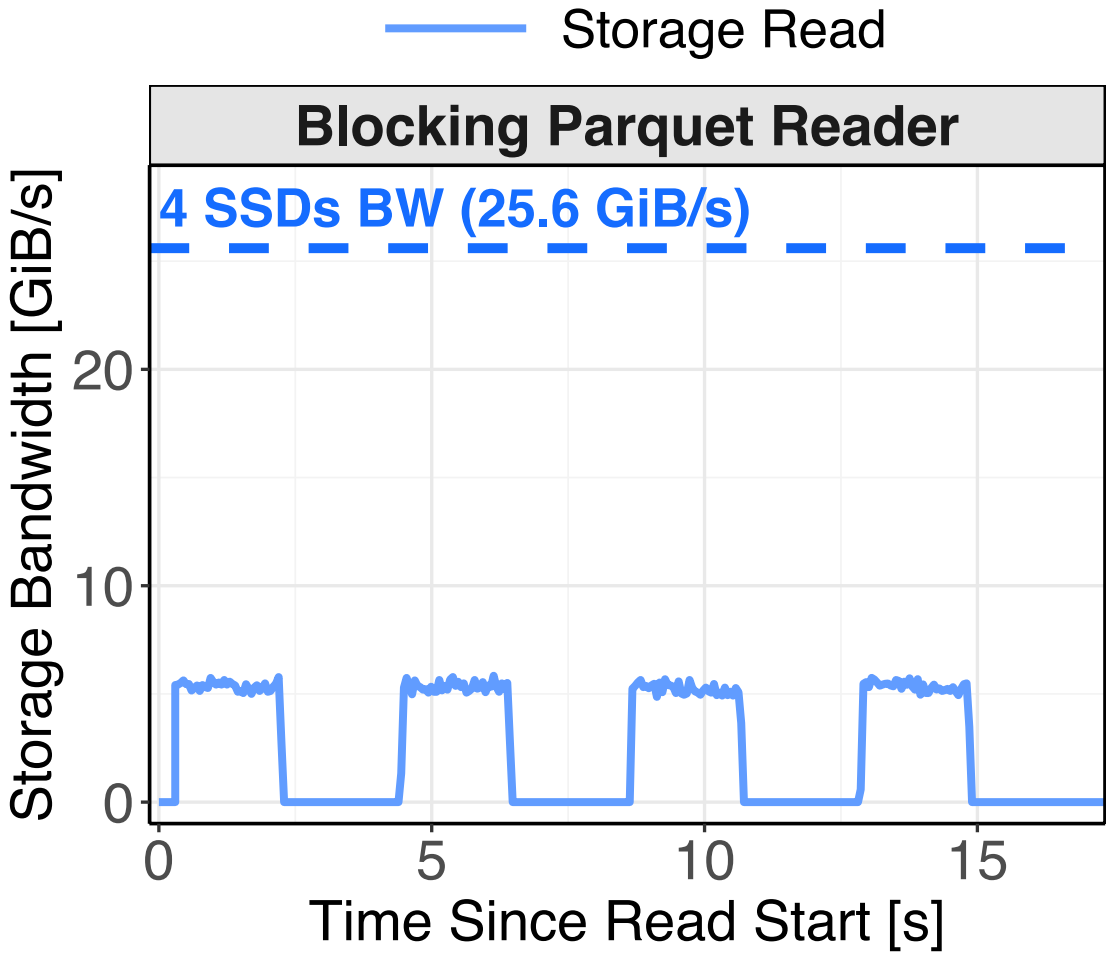


Inefficient Storage Access: Blocking Parquet Reader

`read_parquet()` in one CPU thread:

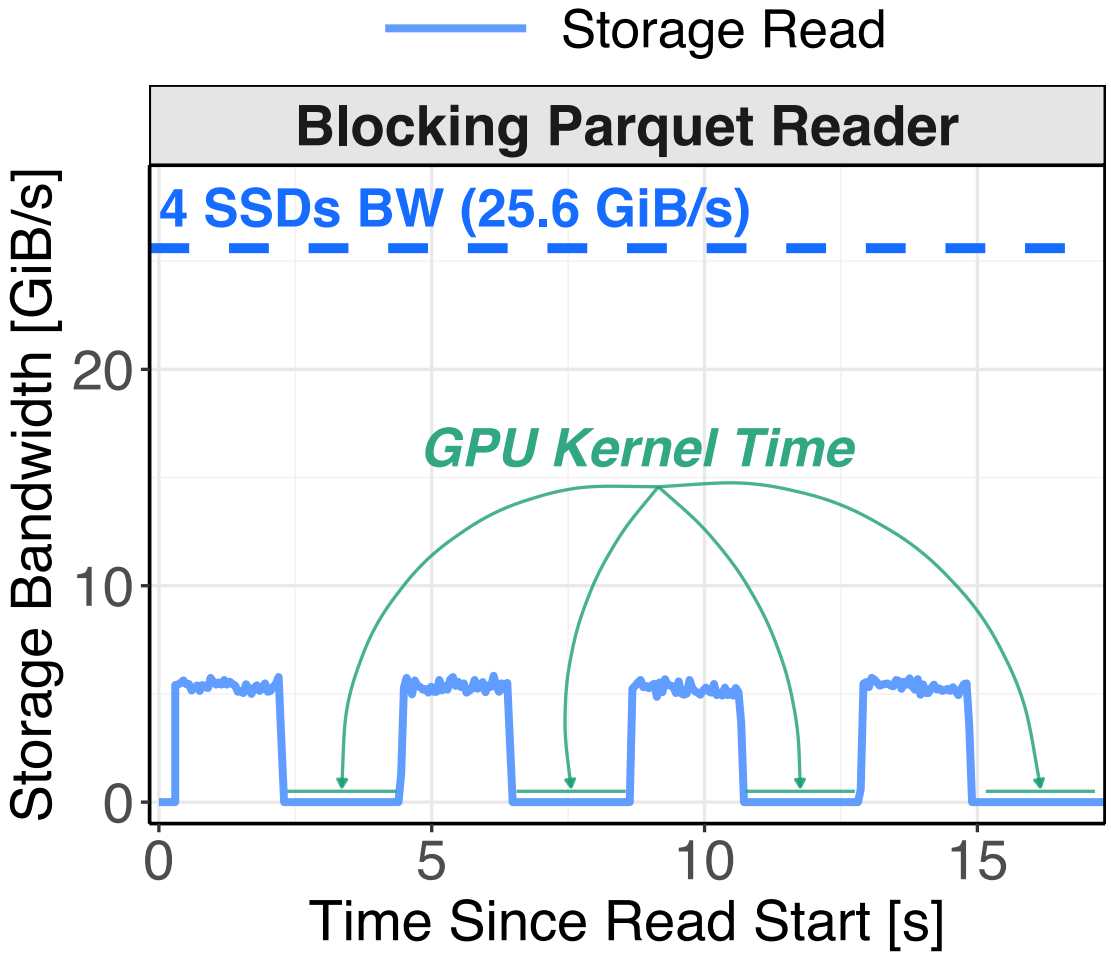


Inefficient Storage Access: Blocking Parquet Reader



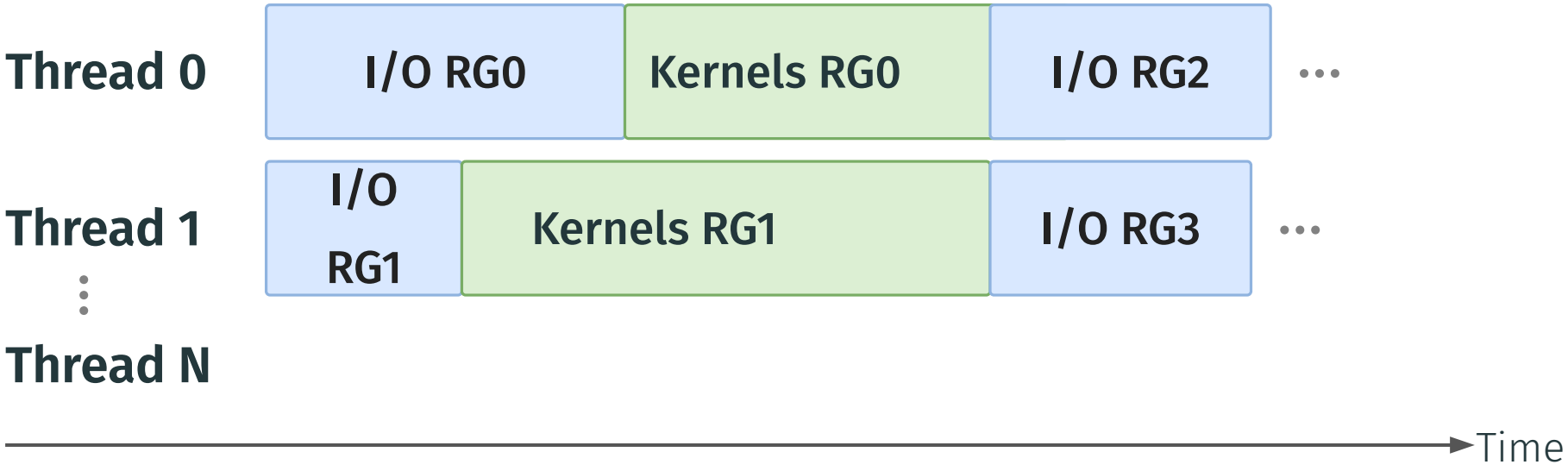
Storage microbenchmark with one GPU (A100, 40GB) and 4 NVMe SSD in PCIe 4.0.

Inefficient Storage Access: Blocking Parquet Reader

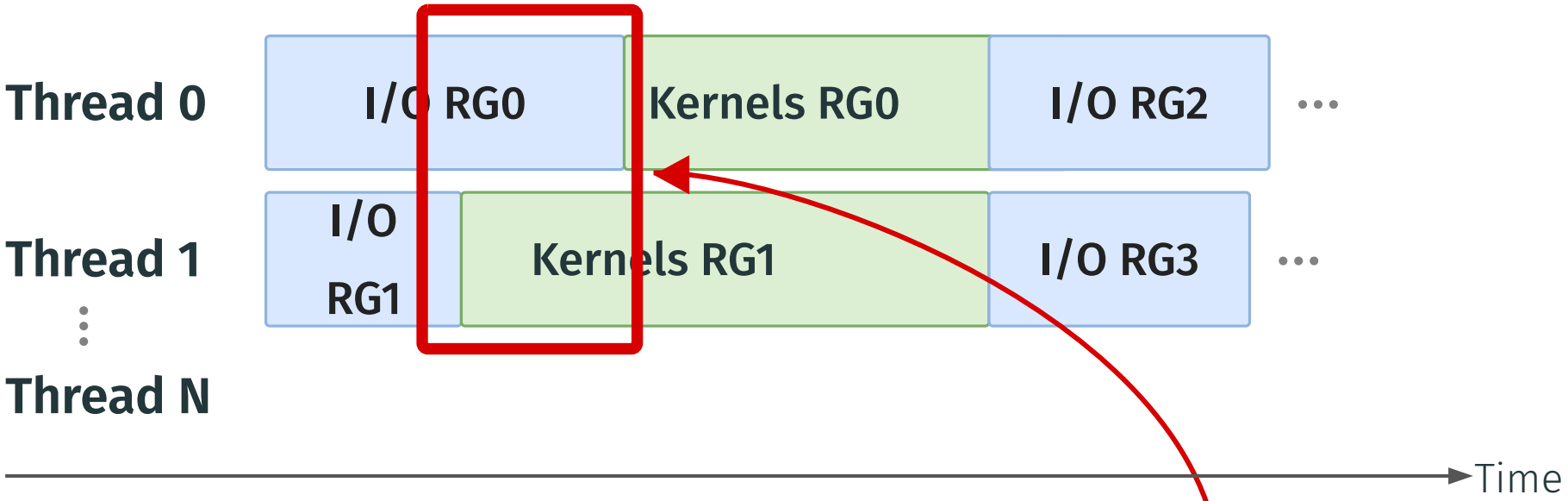


Storage microbenchmark with one GPU (A100, 40GB) and 4 NVMe SSD in PCIe 4.0.

Efficient Storage Access: Chunking

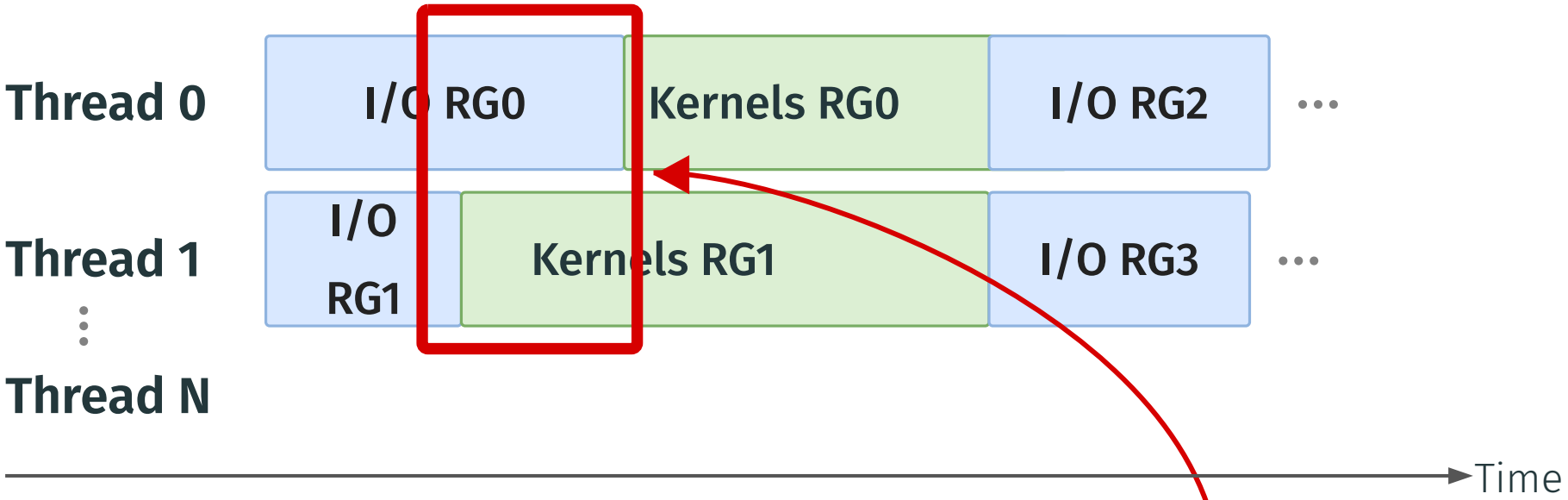


Efficient Storage Access: Chunking



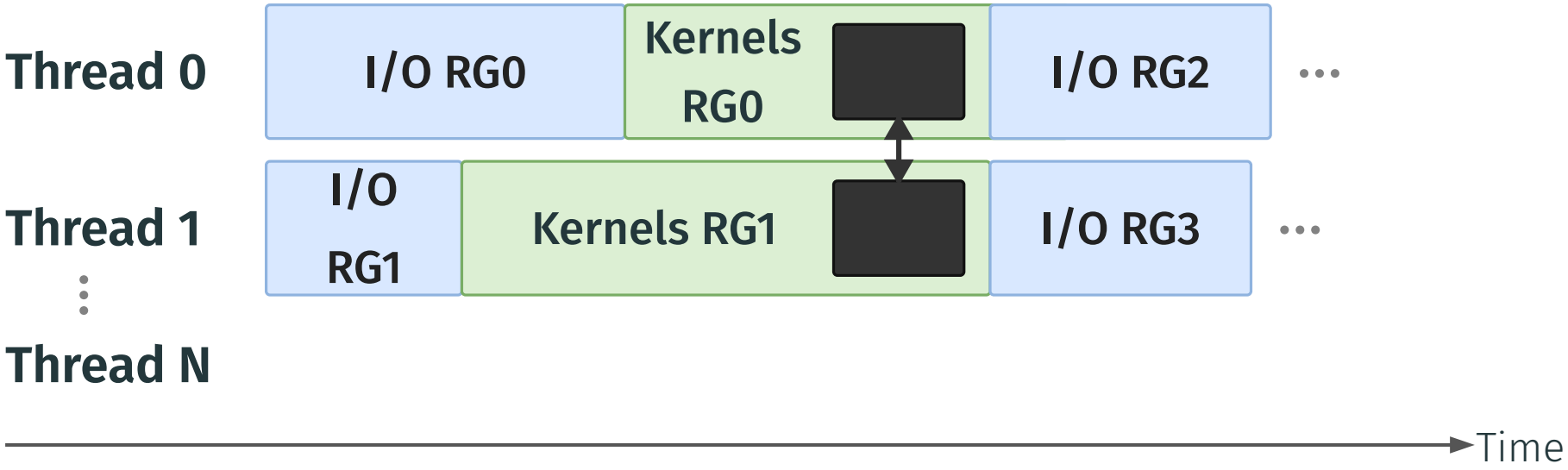
1. Overlapping I/O and Computation

Efficient Storage Access: Chunking

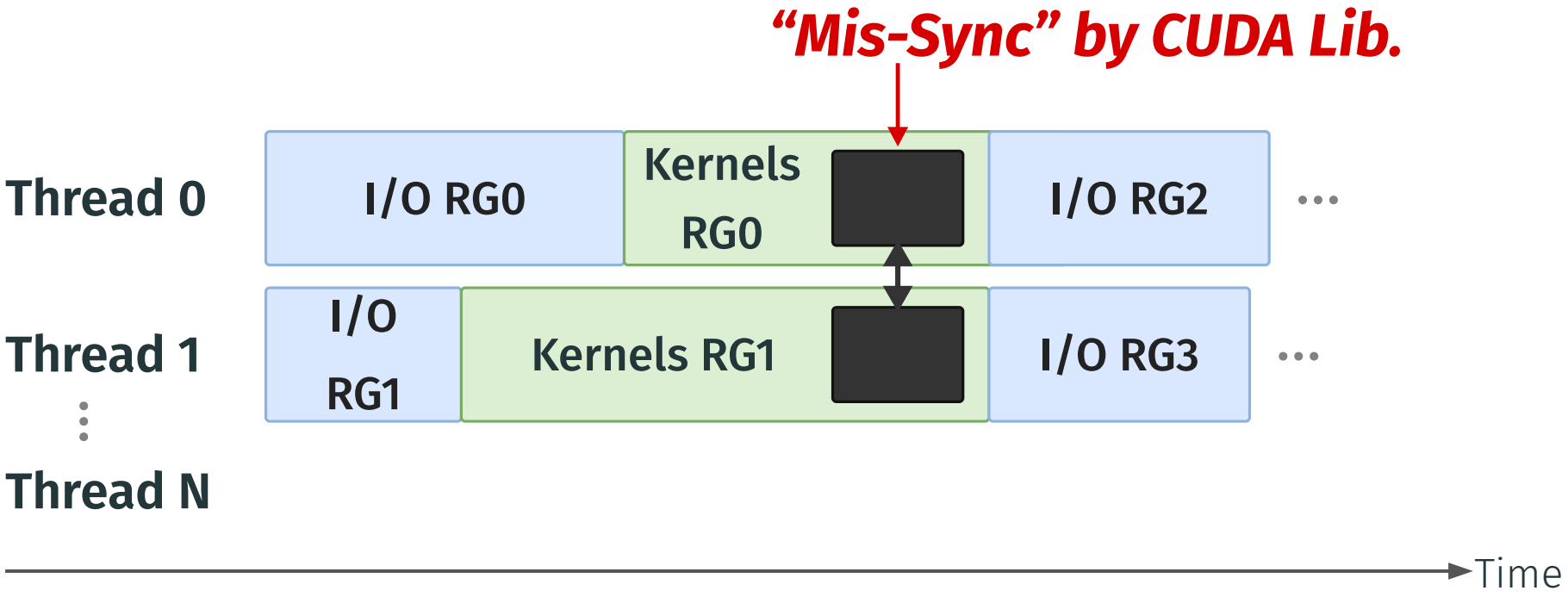


1. Overlapping I/O and Computation
2. Less Memory Pressure

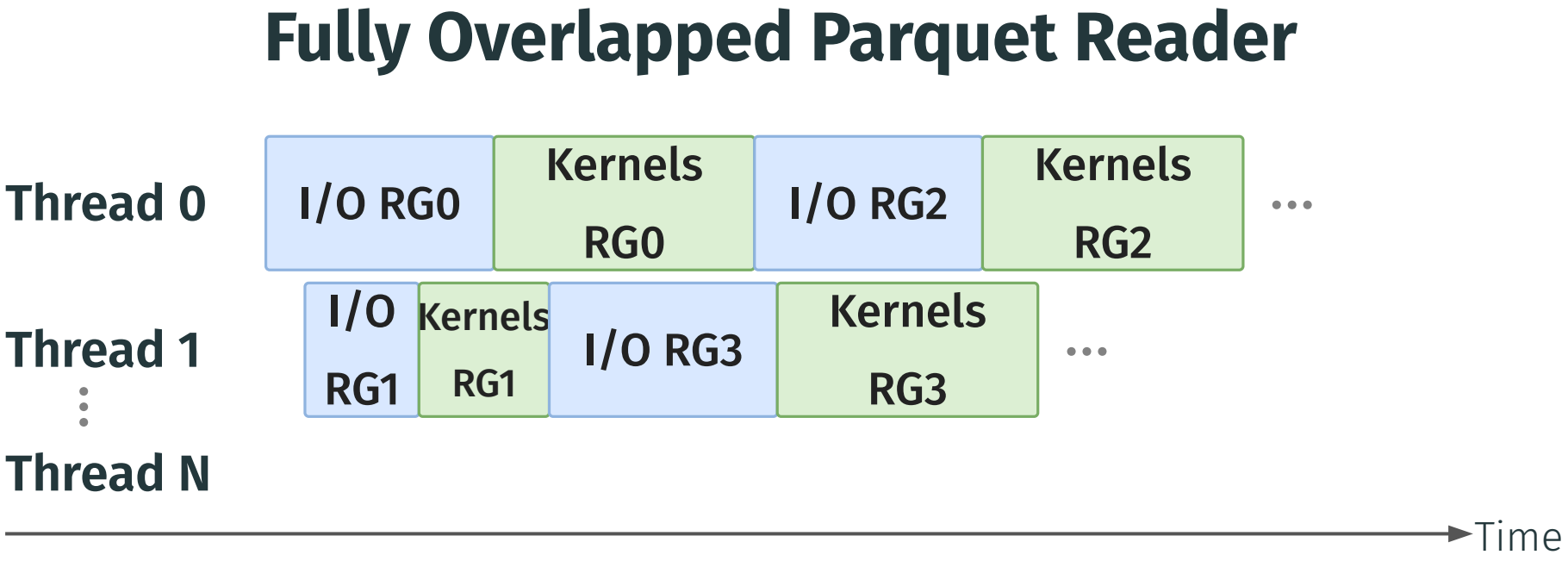
Efficient Storage Access: Mis-Sync Elimination



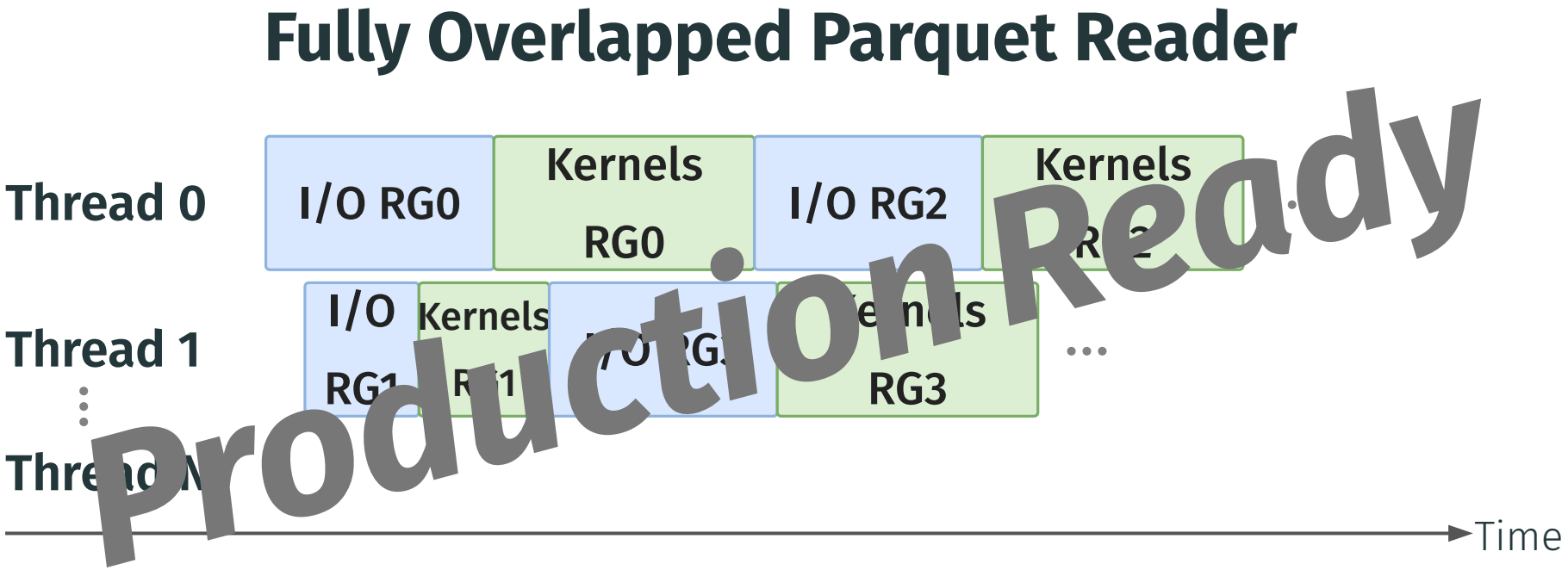
Efficient Storage Access: Mis-Sync Elimination



Efficient Storage Access: Mis-Sync Elimination



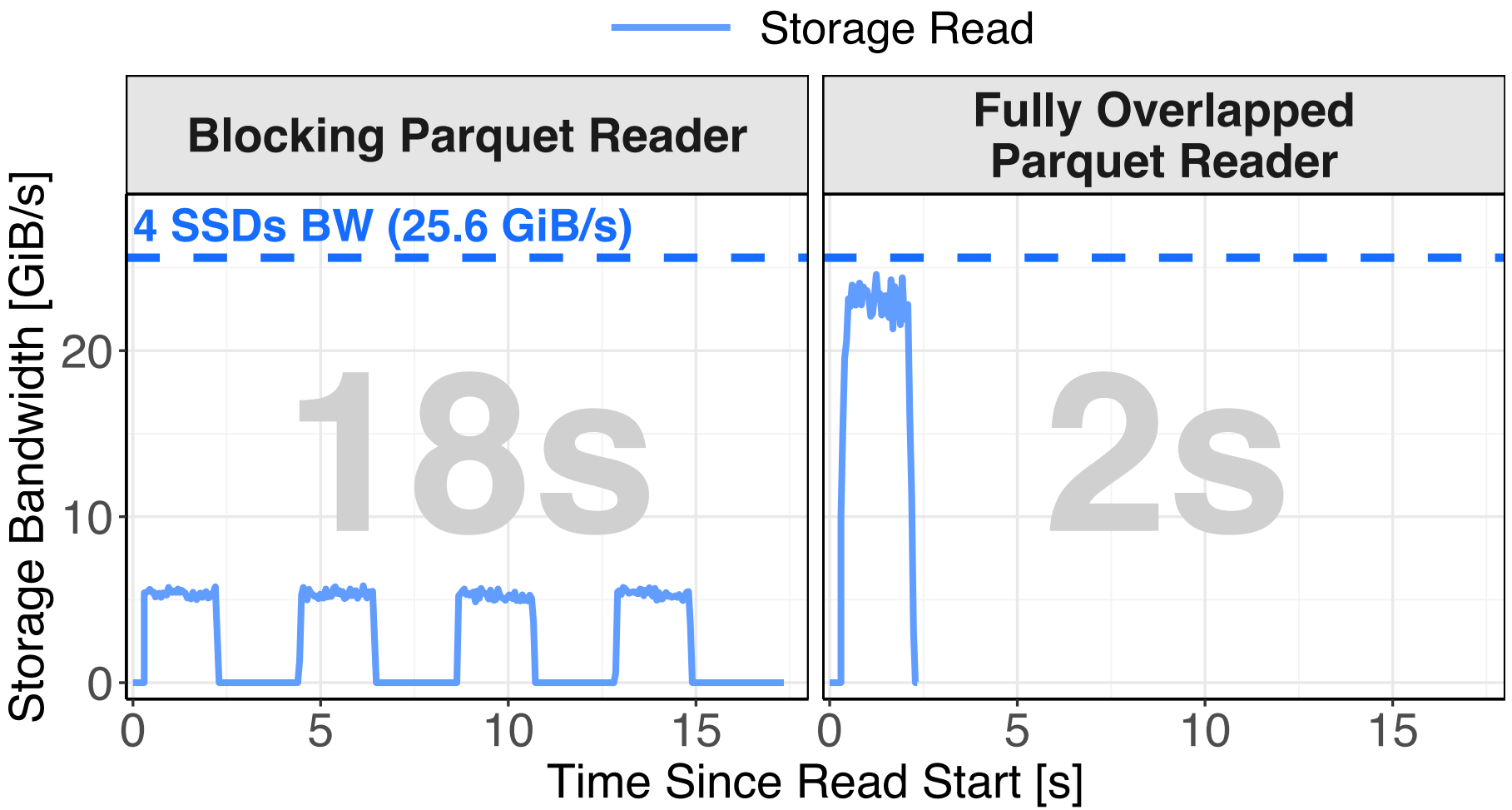
Efficient Storage Access: Mis-Sync Elimination



Production Ready

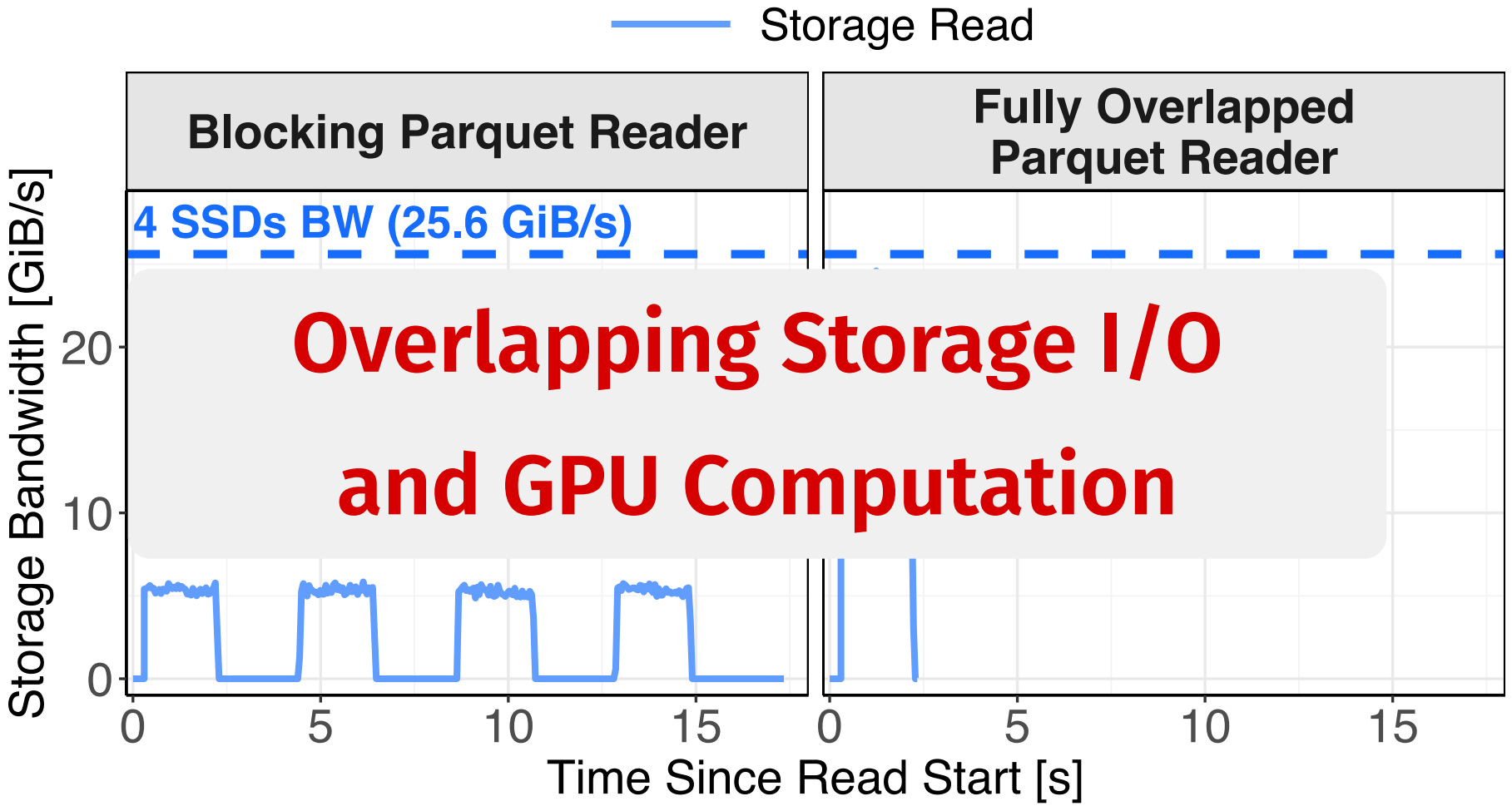
cuDF issue #18892: [Story] Towards a faster Parquet reader with pipelining and multistream optimization, Jigao Luo.
cuDF issue #21272: Migrate libcudf from Thrust to CUB-based algorithms, Bradley Dice.

Efficient Storage Access: Mis-Sync Elimination



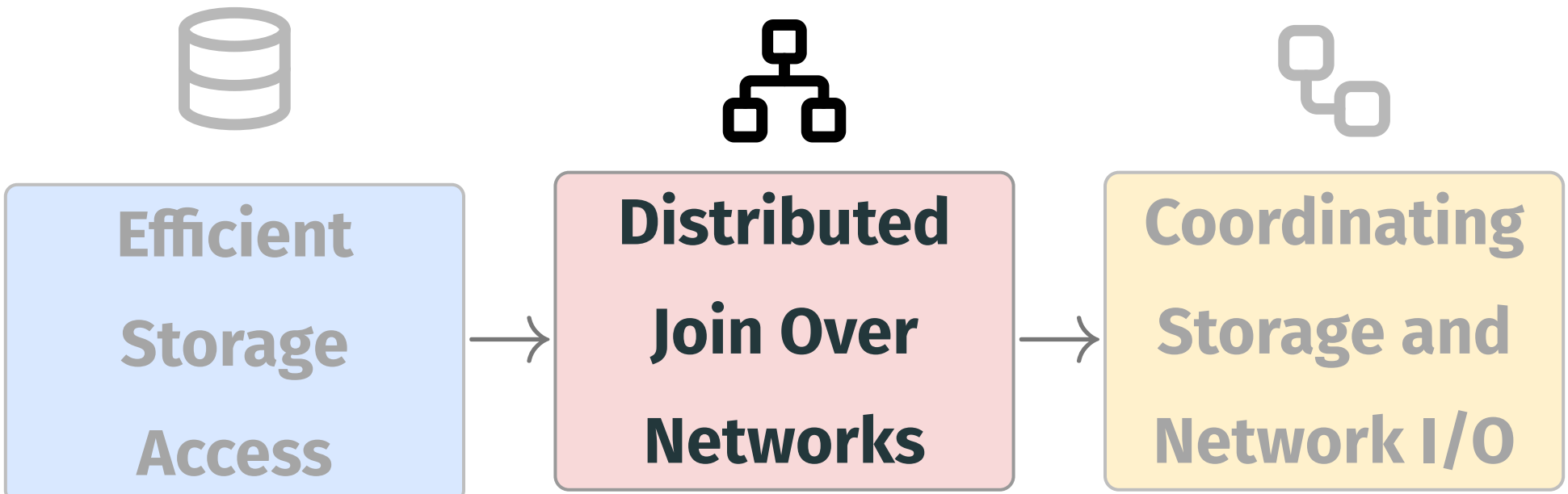
Storage microbenchmark with one GPU (A100, 40GB) and 4 NVMe SSD in PCIe 4.0.

Efficient Storage Access: Mis-Sync Elimination

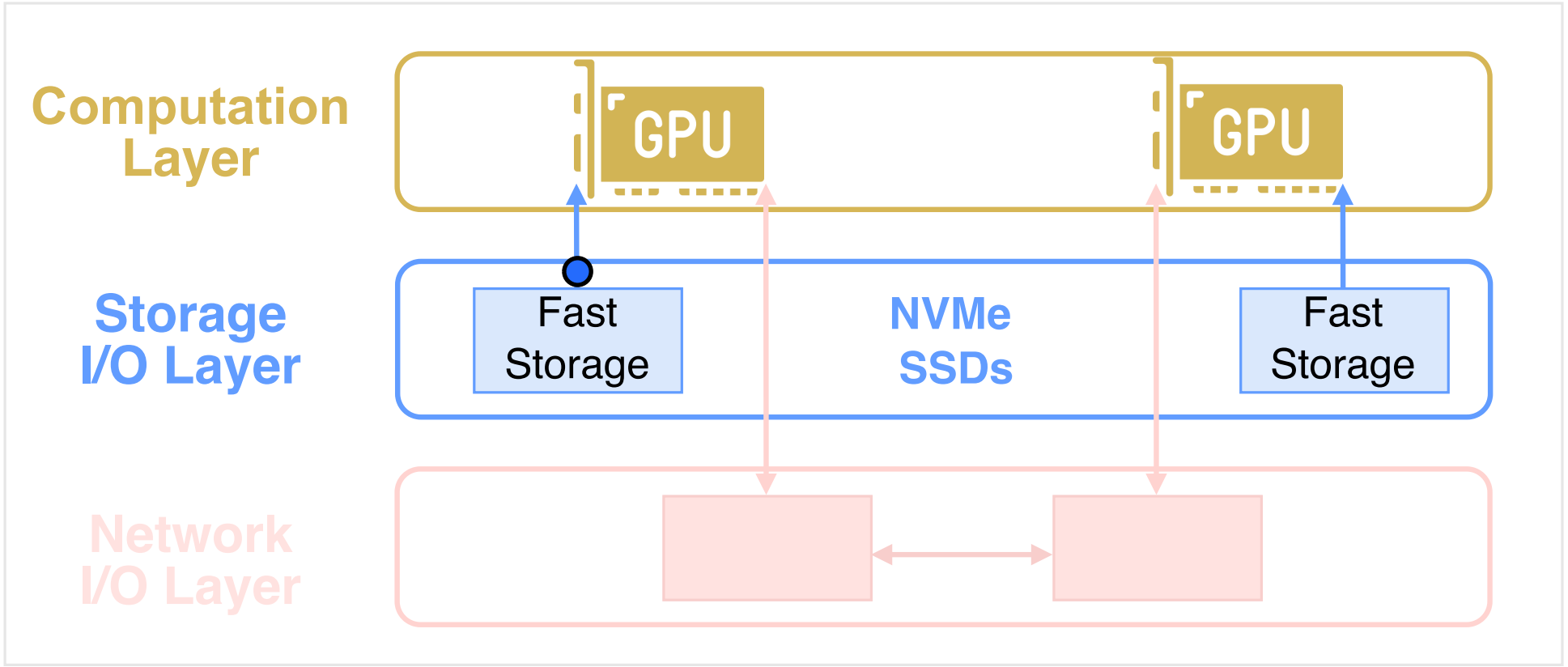


Storage microbenchmark with one GPU (A100, 40GB) and 4 NVMe SSD in PCIe 4.0.

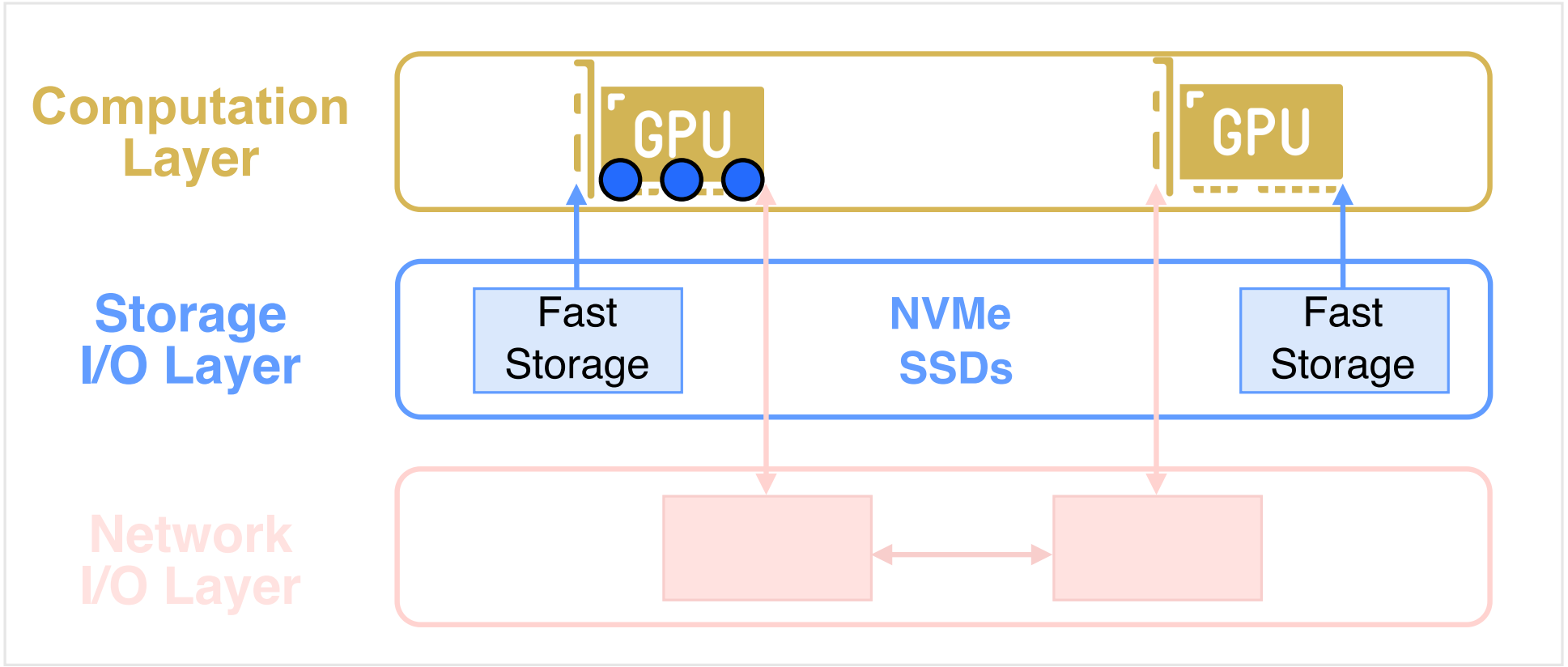
PystachIO: Network



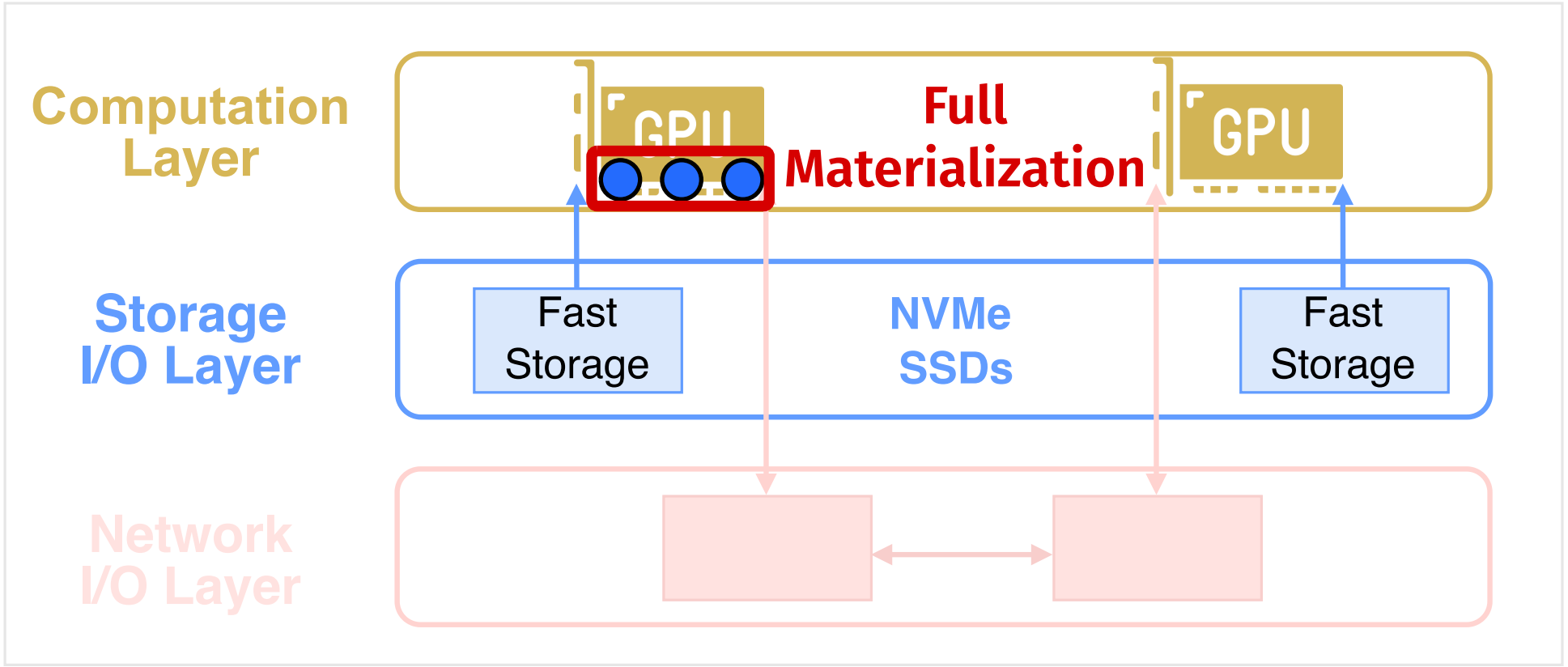
Run a Distributed Query



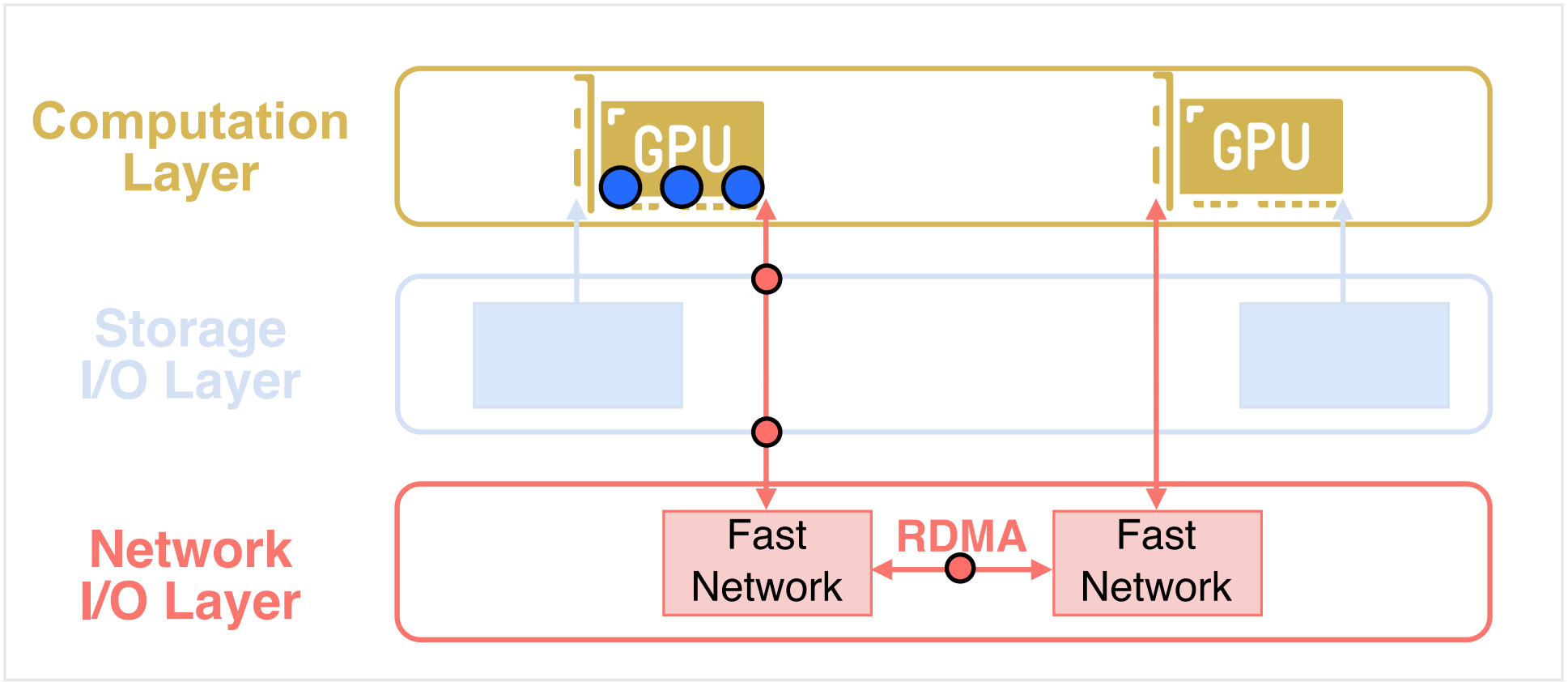
Run a Distributed Query



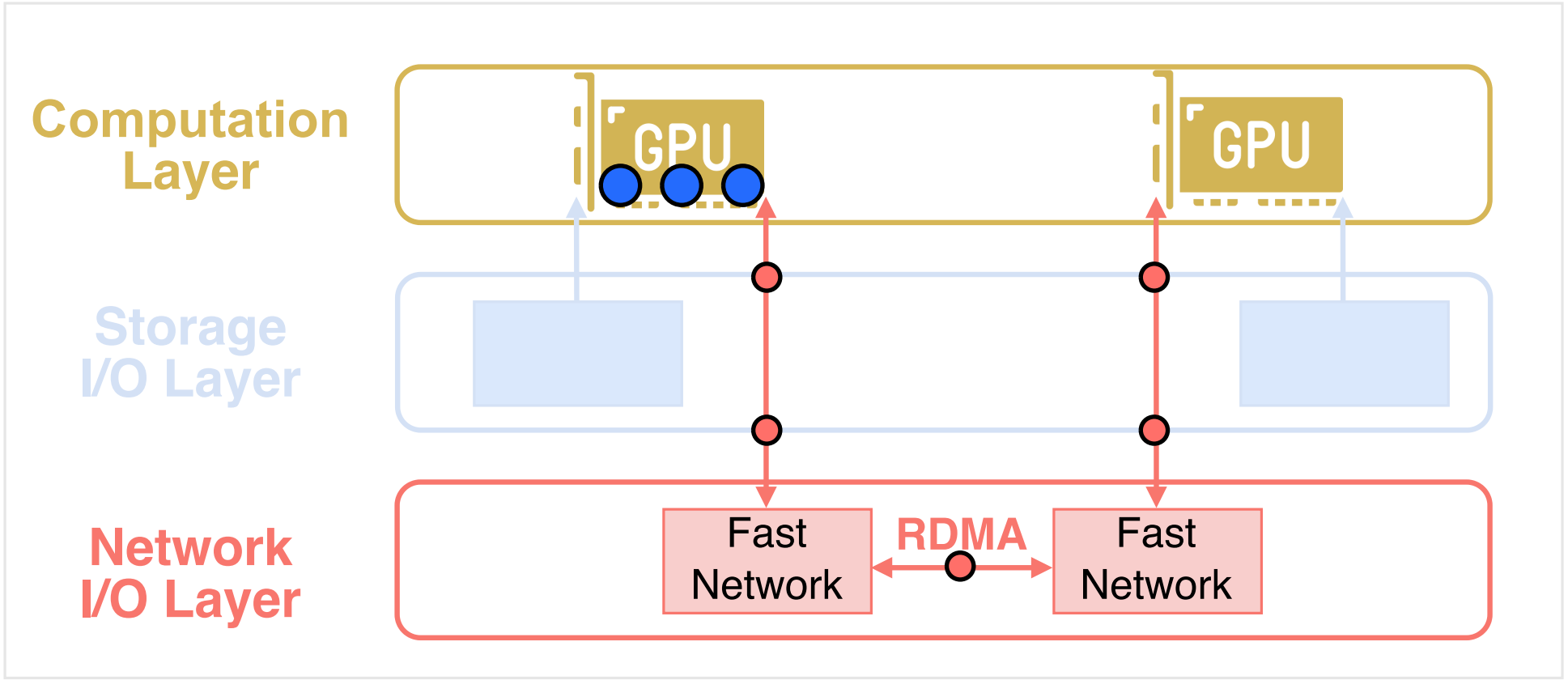
Run a Distributed Query



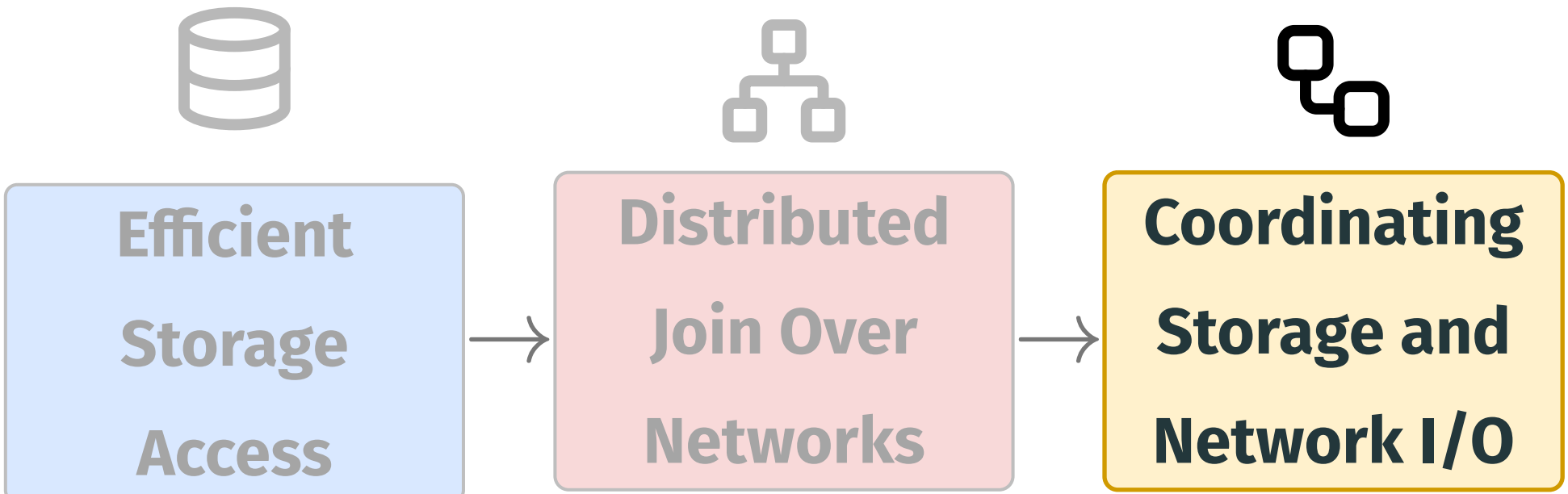
Run a Distributed Query



Run a Distributed Query

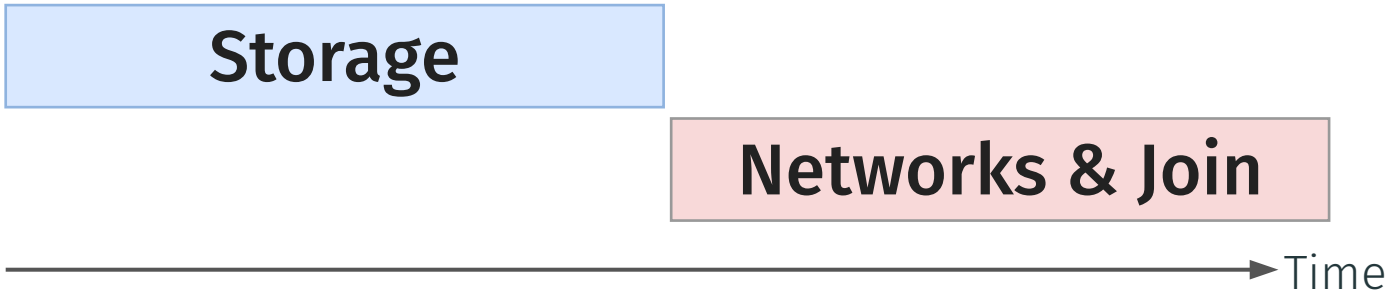


PystachIO: Coordinating I/O



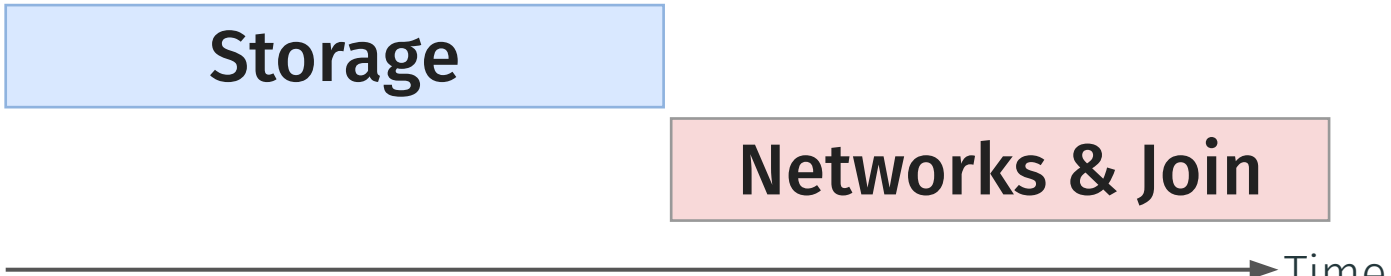
Fast Storage, then Fast Network

- 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

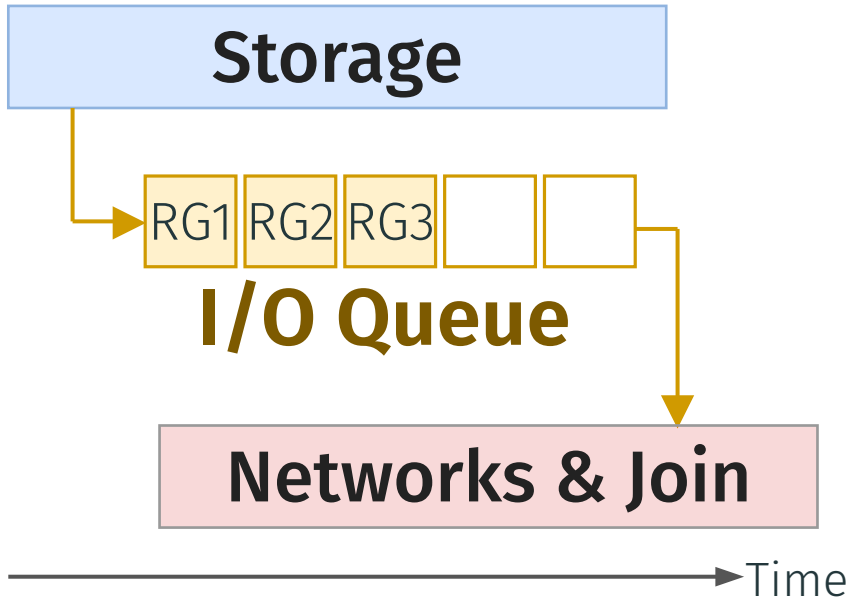
- Storage
- Networ

Is It Possible to Overlap Storage and Network?

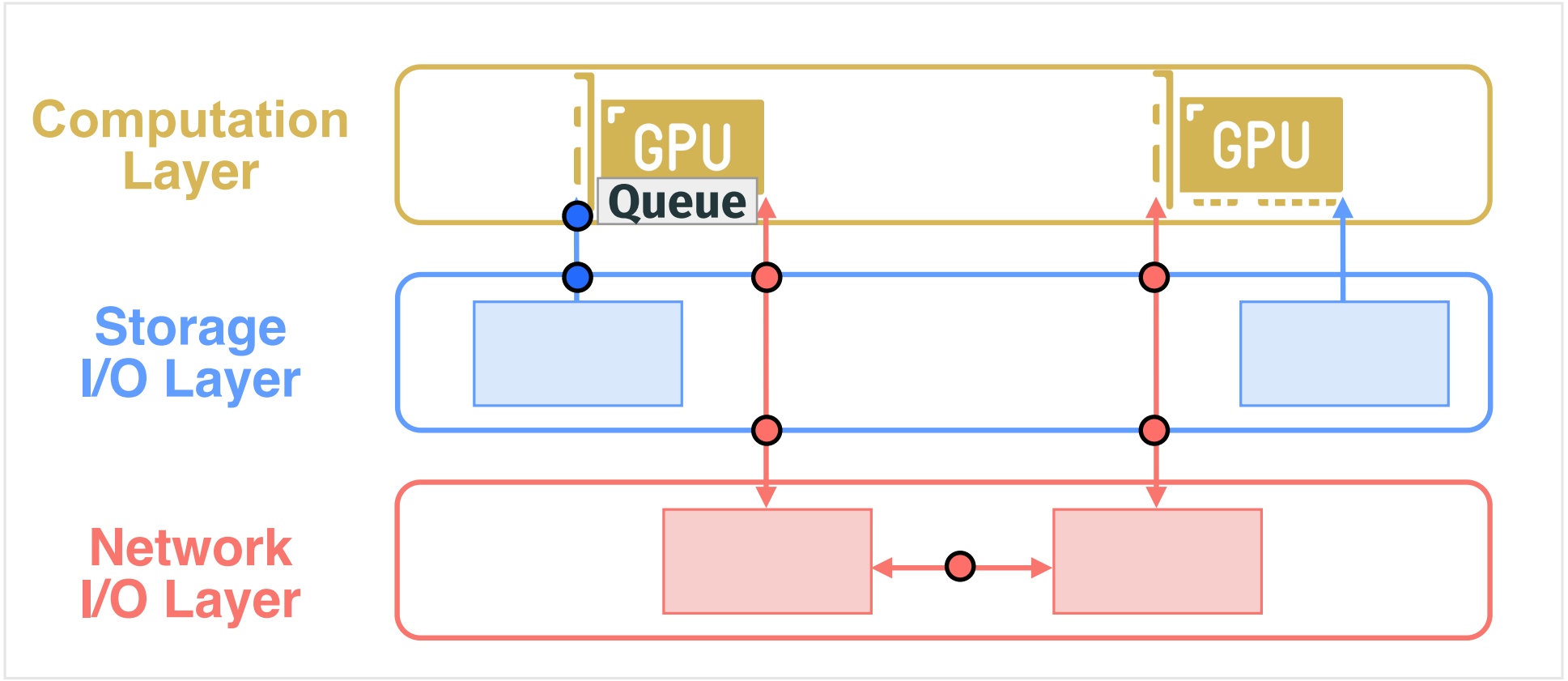


I/O Queue For Coordination

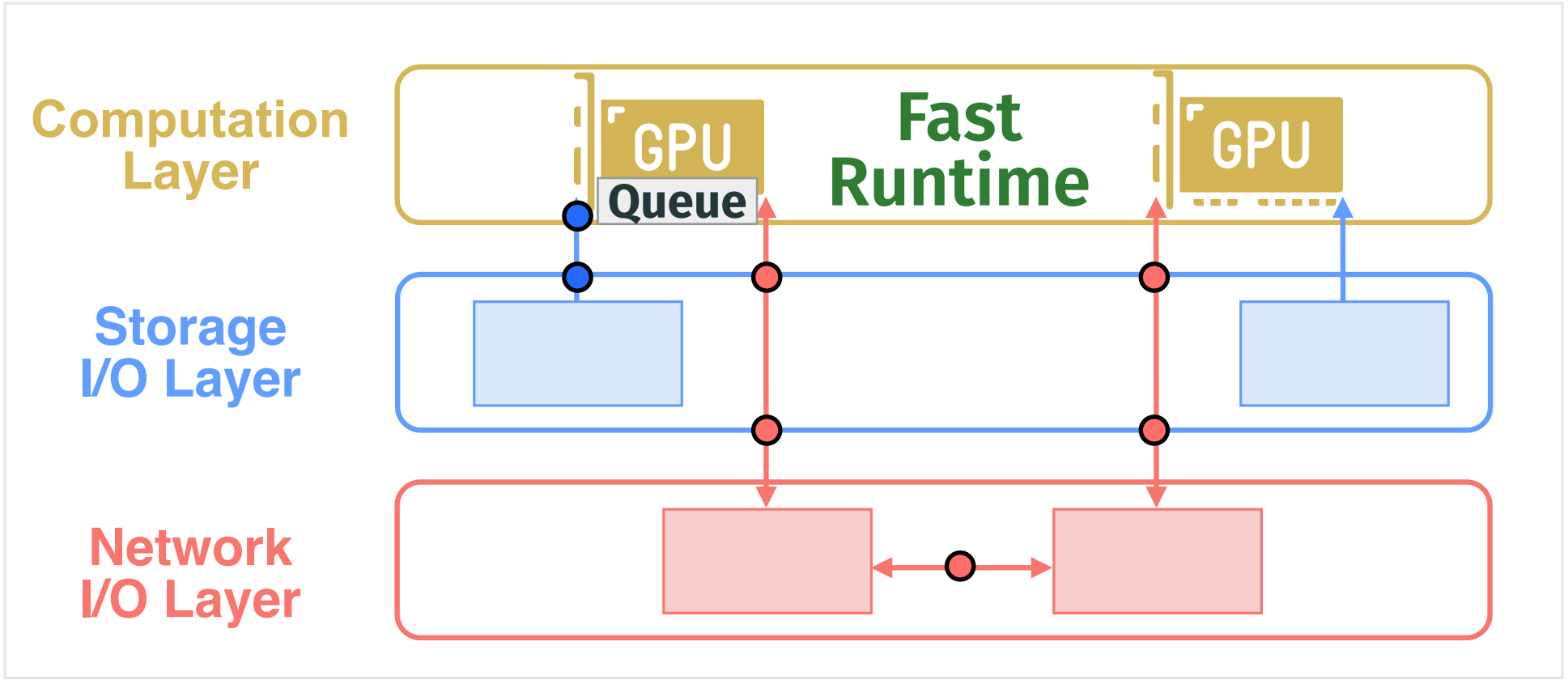
- Storage: enqueuing
- Network & Join: dequeuing
- Then network shuffling



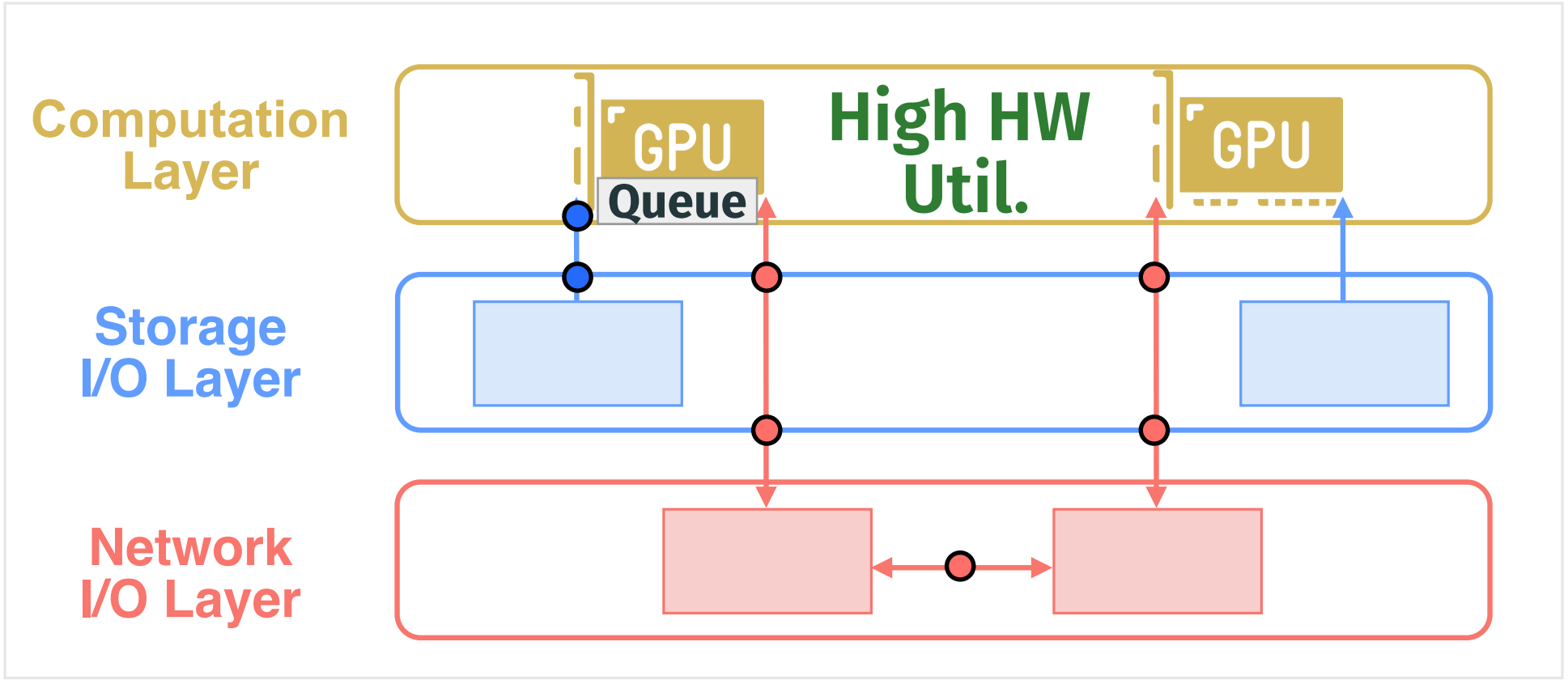
Overlapping Network, Storage, and Computation



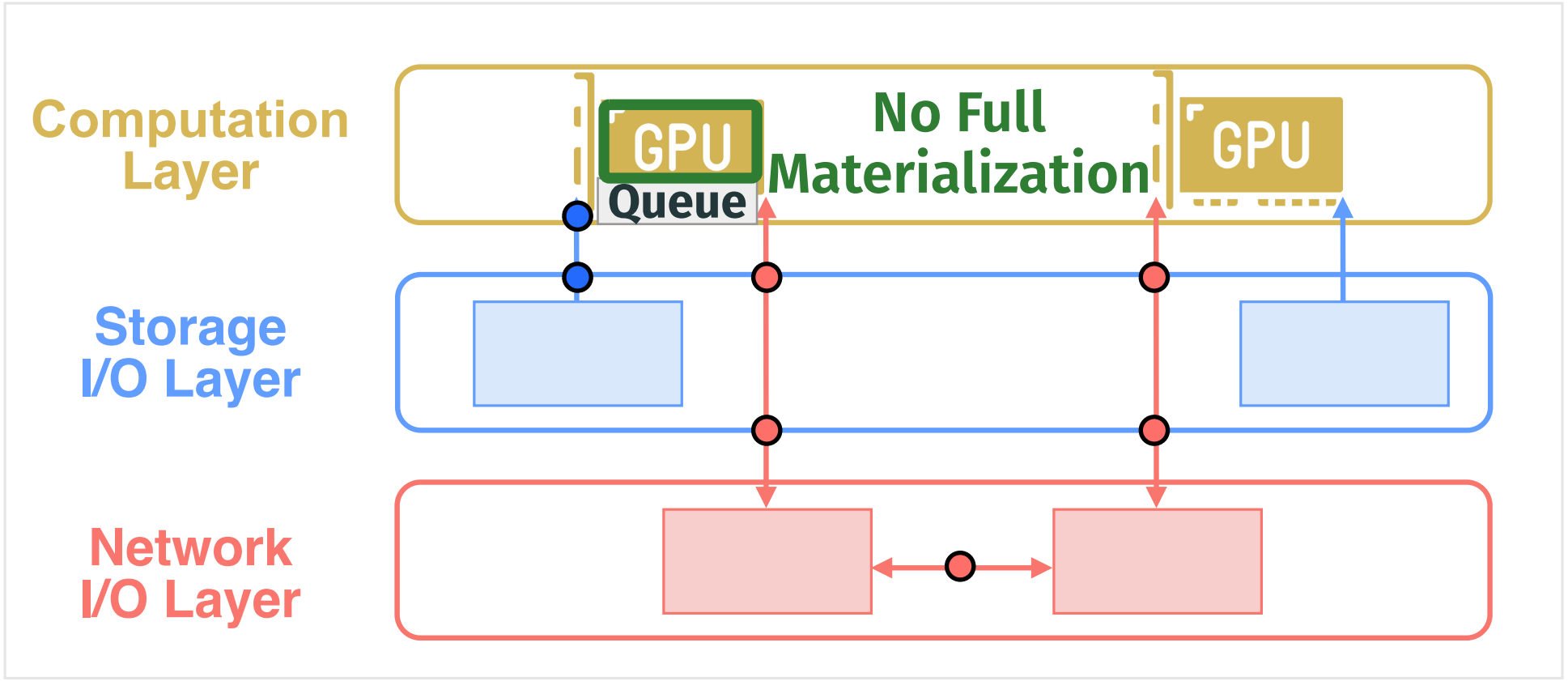
Overlapping Network, Storage, and Computation



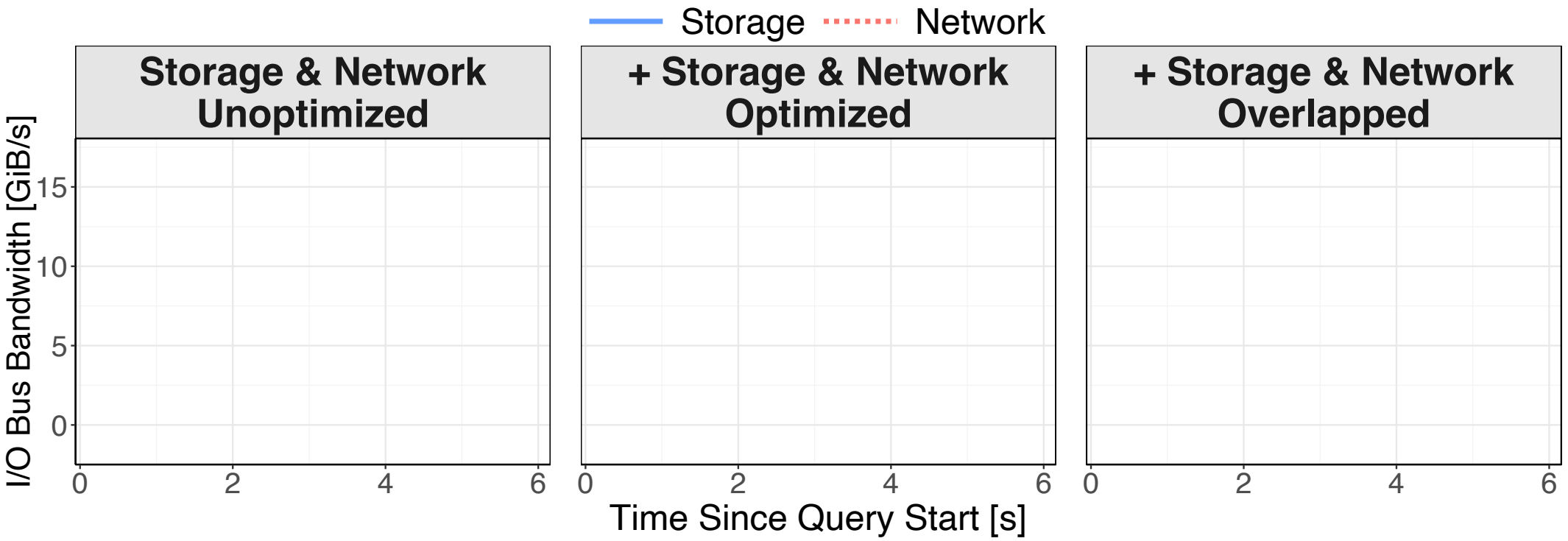
Overlapping Network, Storage, and Computation



Overlapping Network, Storage, and Computation

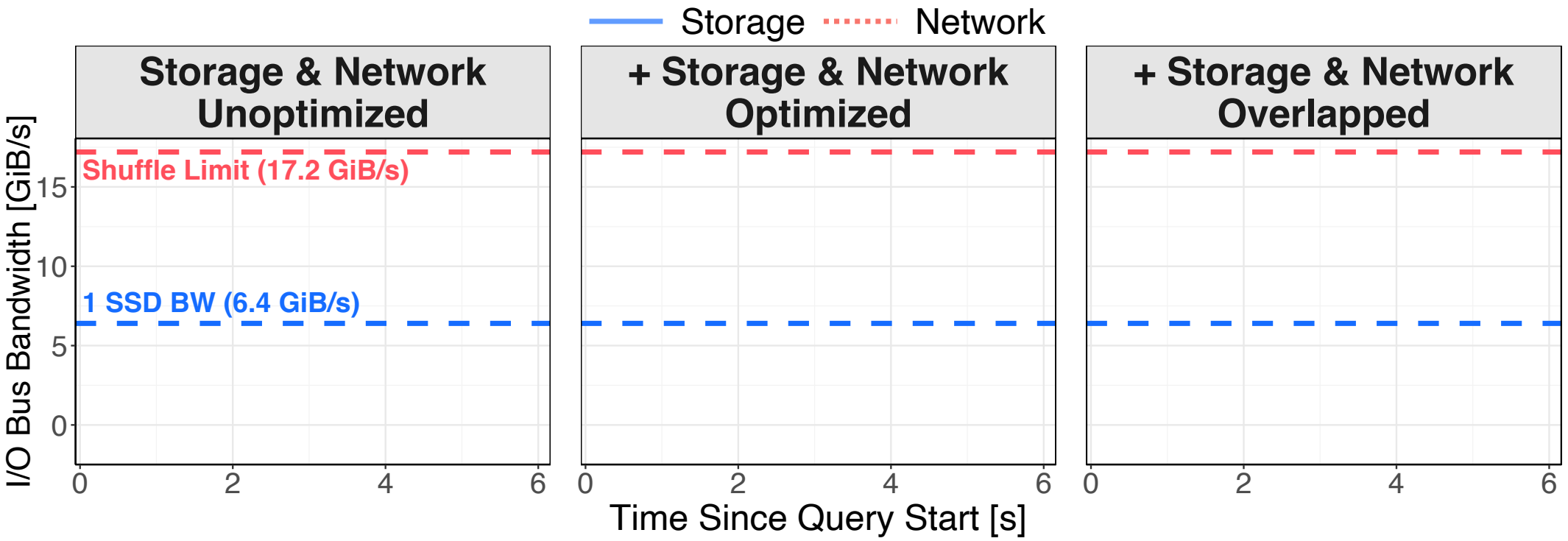


Overlapping Network, Storage, and Computation



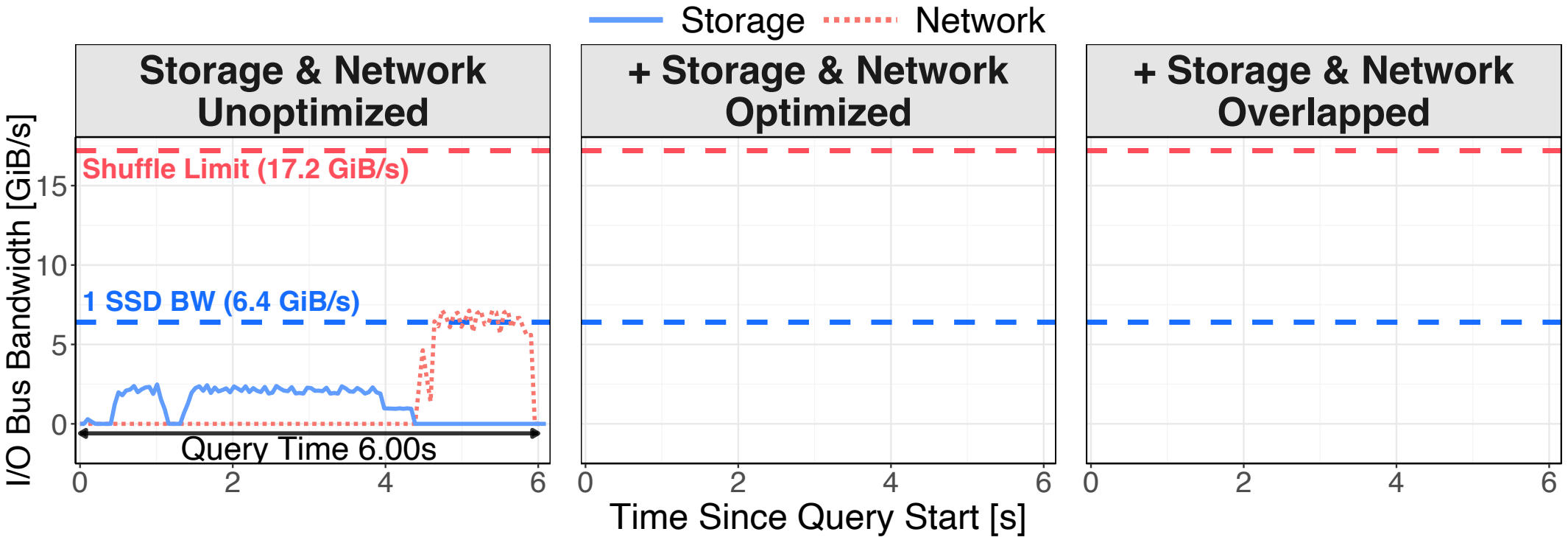
TPC-H Query 3 (SF 500) runtime executed on 2 GPUs (Each GPU has one SSD and one NIC)

Overlapping Network, Storage, and Computation



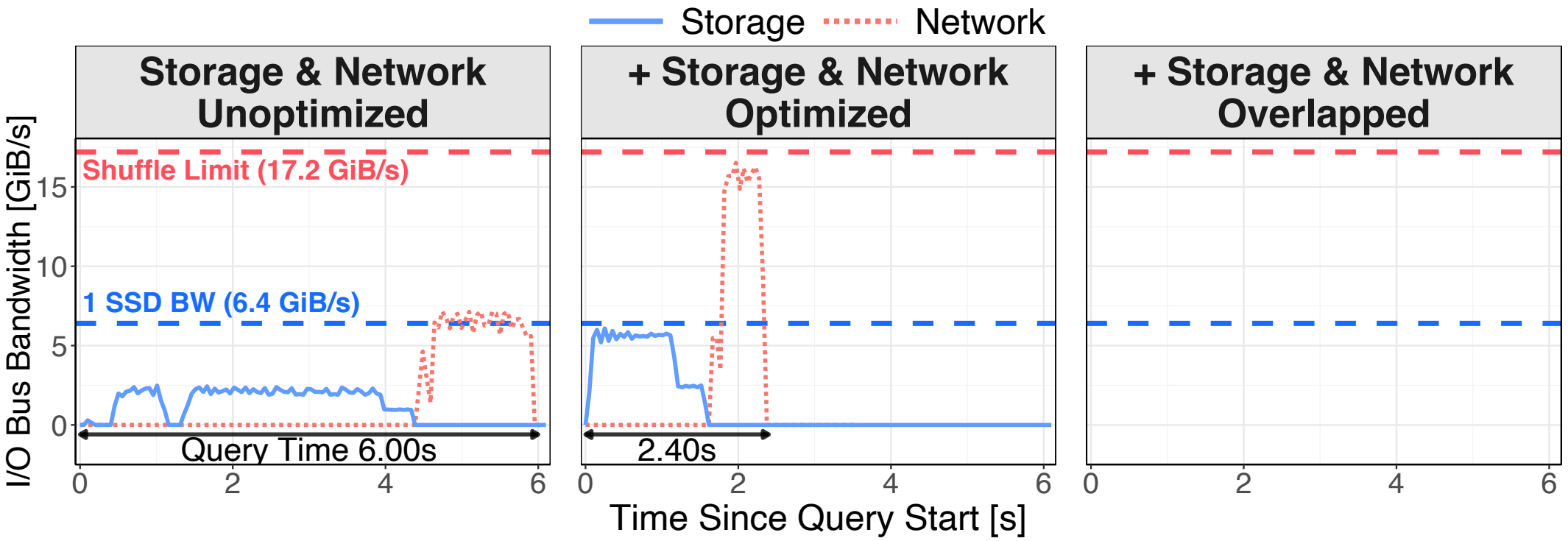
TPC-H Query 3 (SF 500) runtime executed on 2 GPUs (Each GPU has one SSD and one NIC)

Overlapping Network, Storage, and Computation



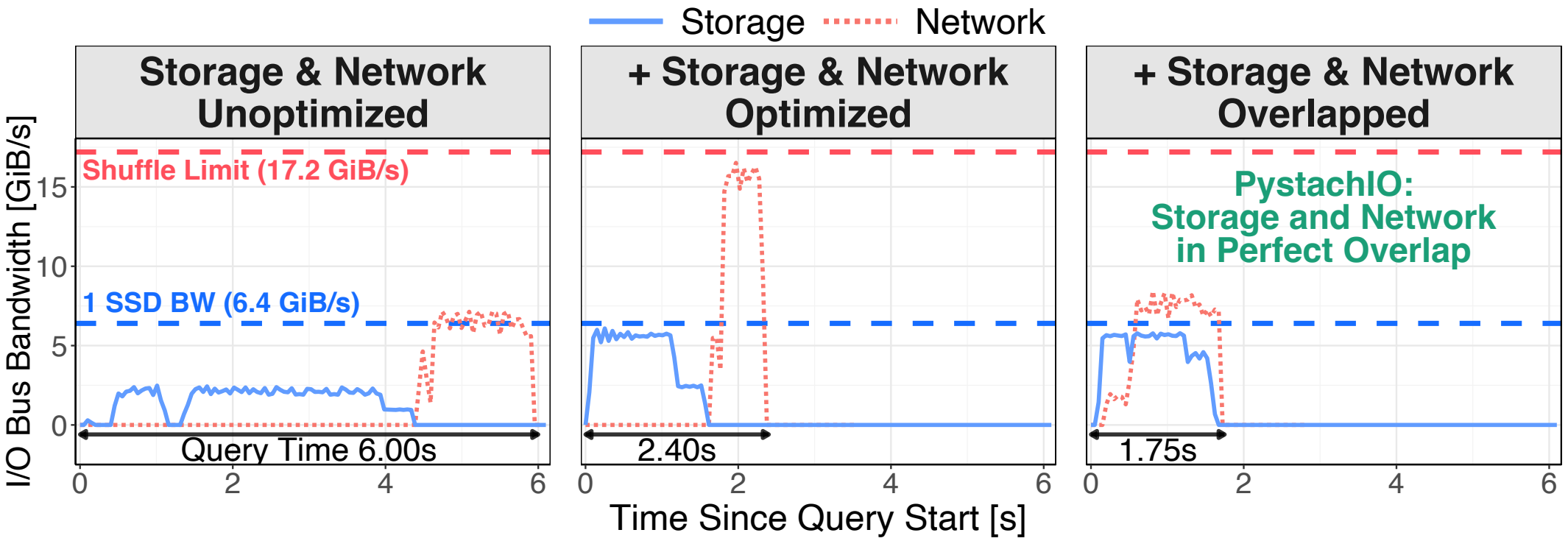
TPC-H Query 3 (SF 500) runtime executed on 2 GPUs (Each GPU has one SSD and one NIC)

Overlapping Network, Storage, and Computation



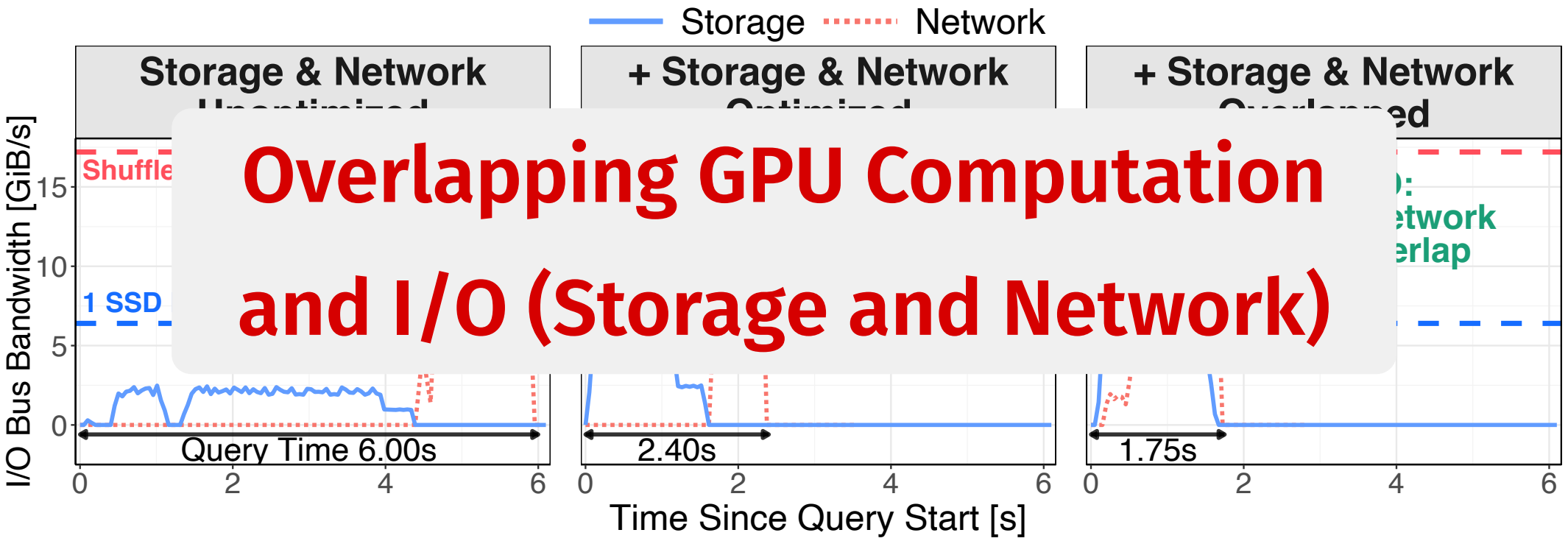
TPC-H Query 3 (SF 500) runtime executed on 2 GPUs (Each GPU has one SSD and one NIC)

Overlapping Network, Storage, and Computation



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

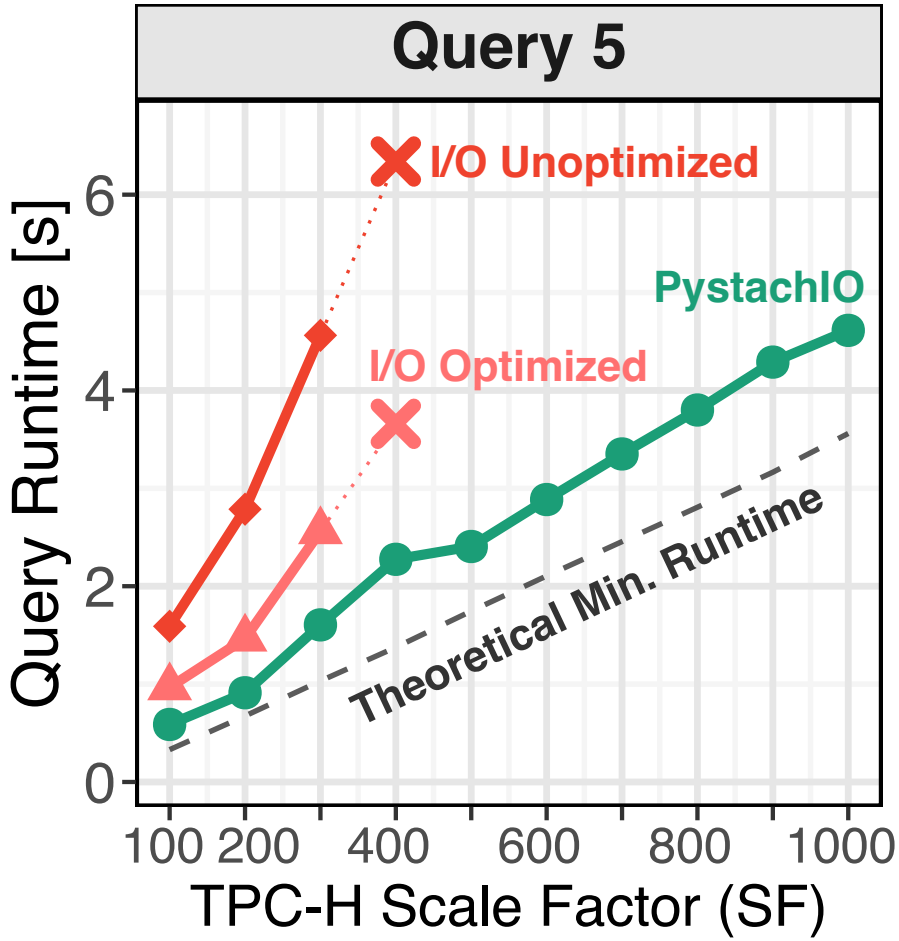
Overlapping Network, Storage, and Computation



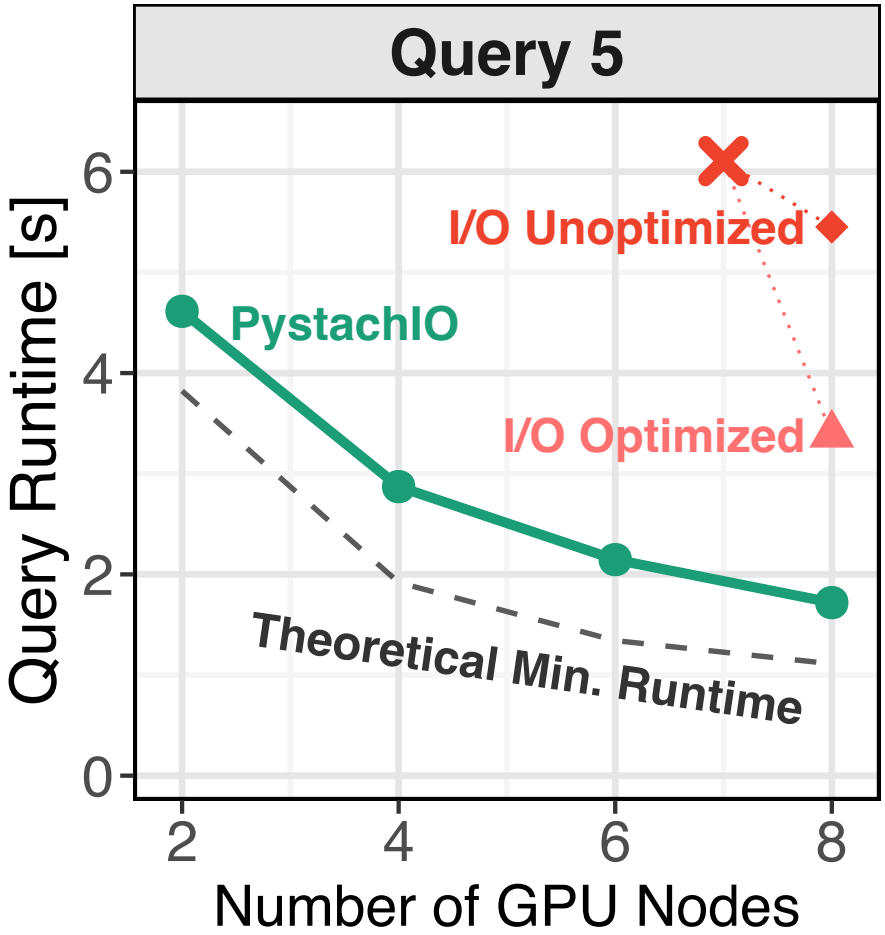
Overlapping GPU Computation and I/O (Storage and Network)

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



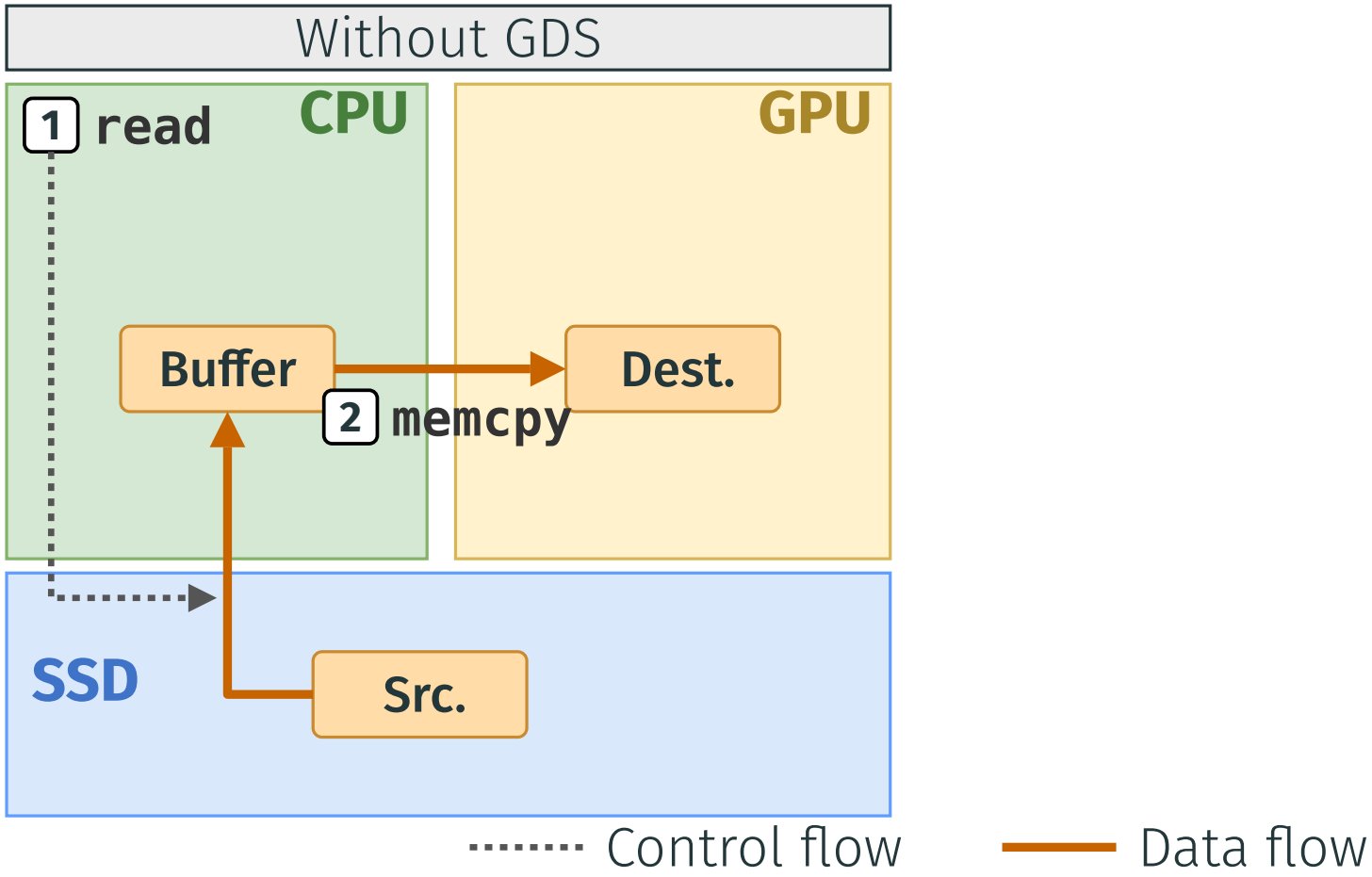
Scalability: Number of GPUs with SF1000



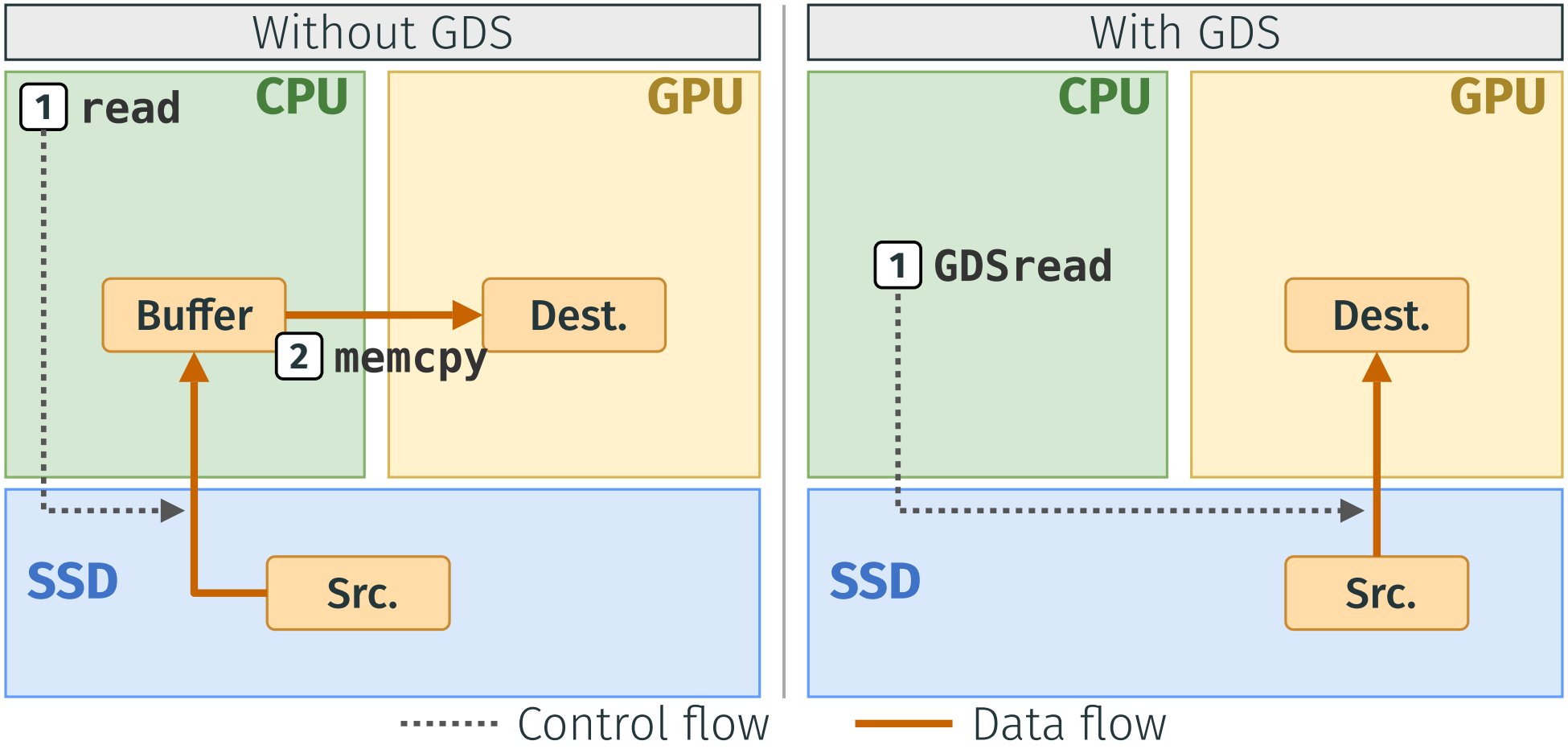
3. Storage Protocol

BTW 2025

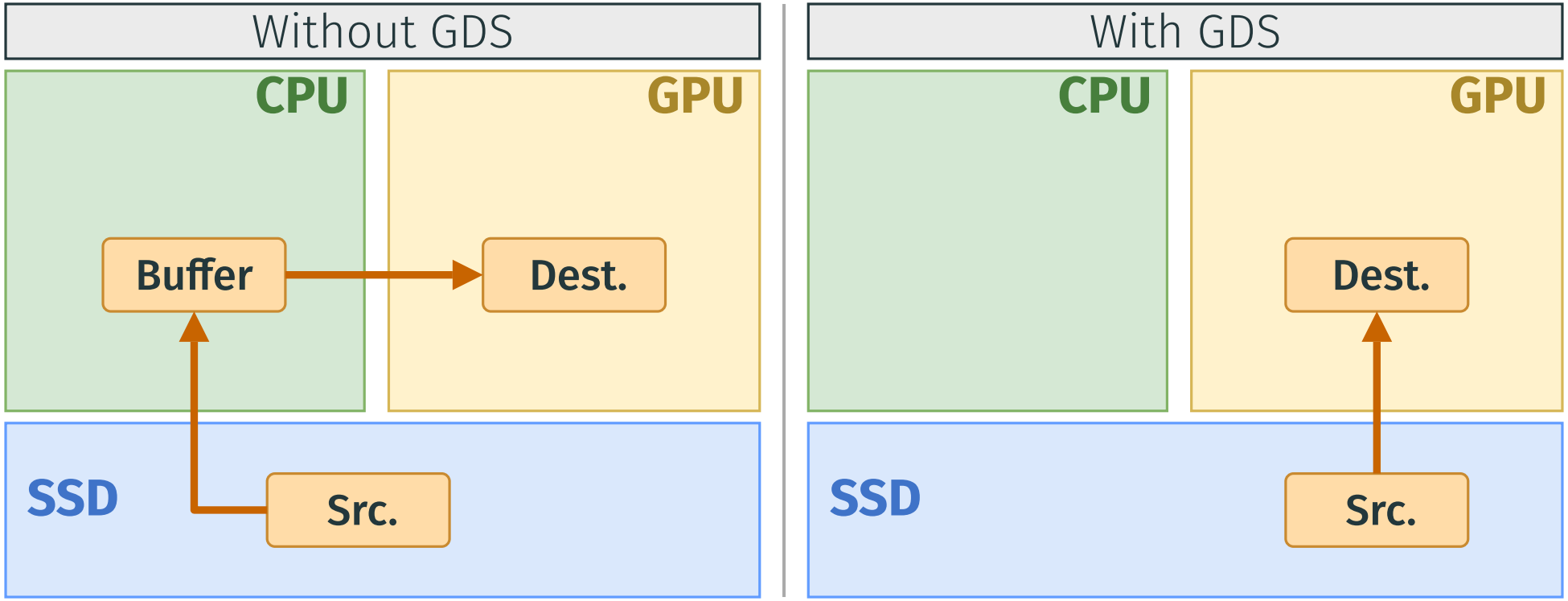
GPU Direct Storage (GDS)



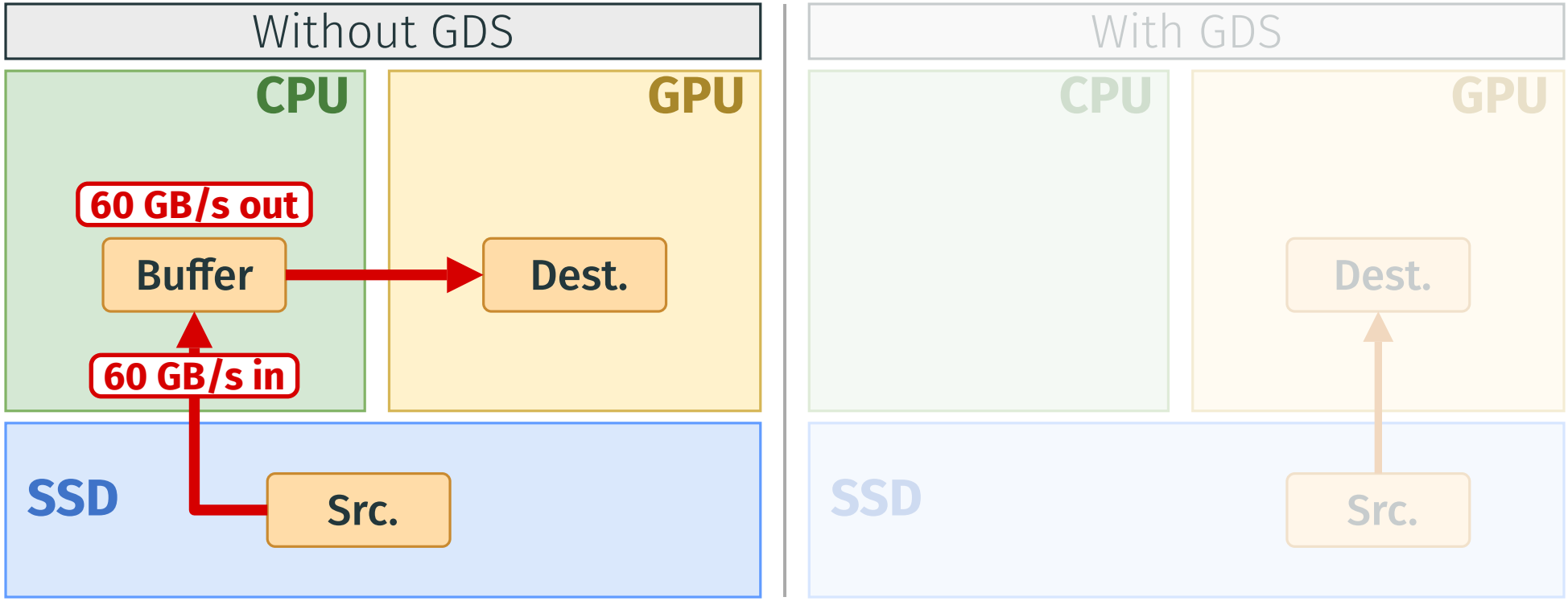
GPU Direct Storage (GDS)



GDS Benefits: Host Memory BW

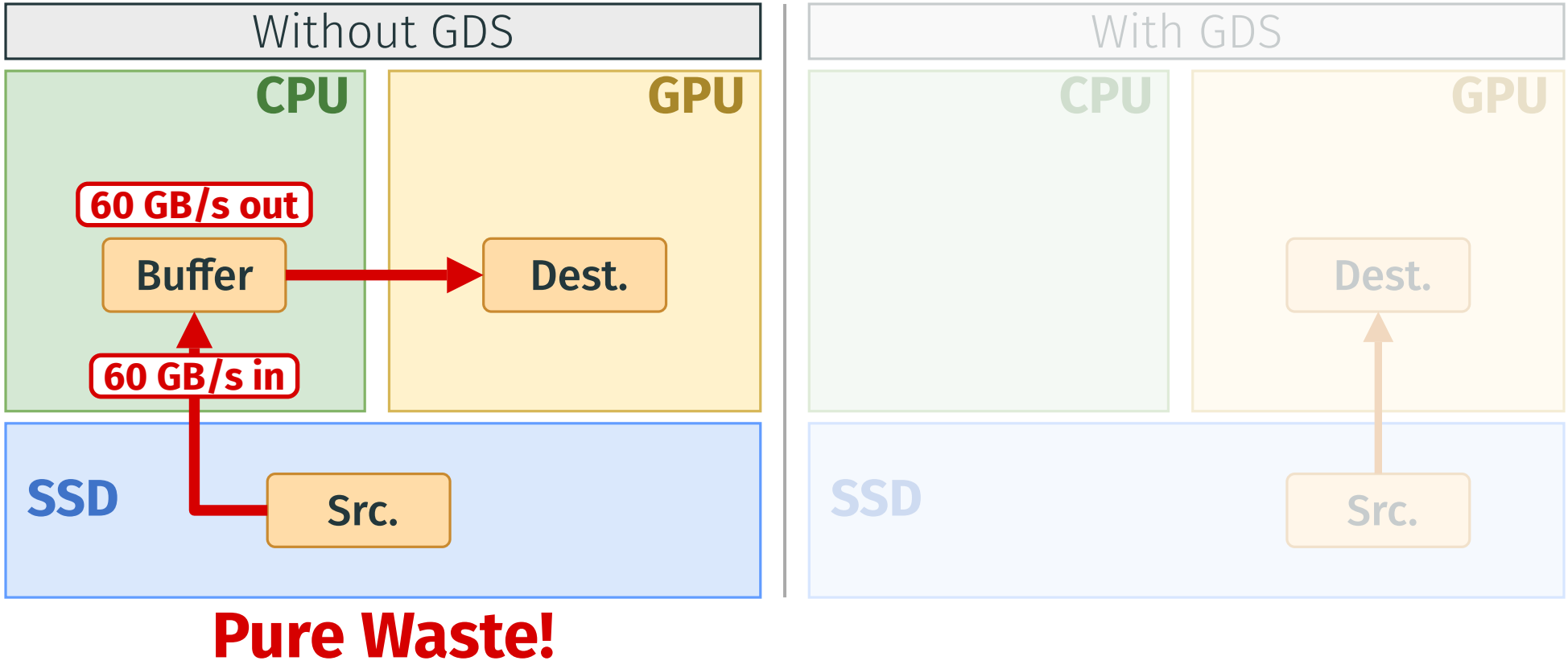


GDS Benefits: Host Memory BW



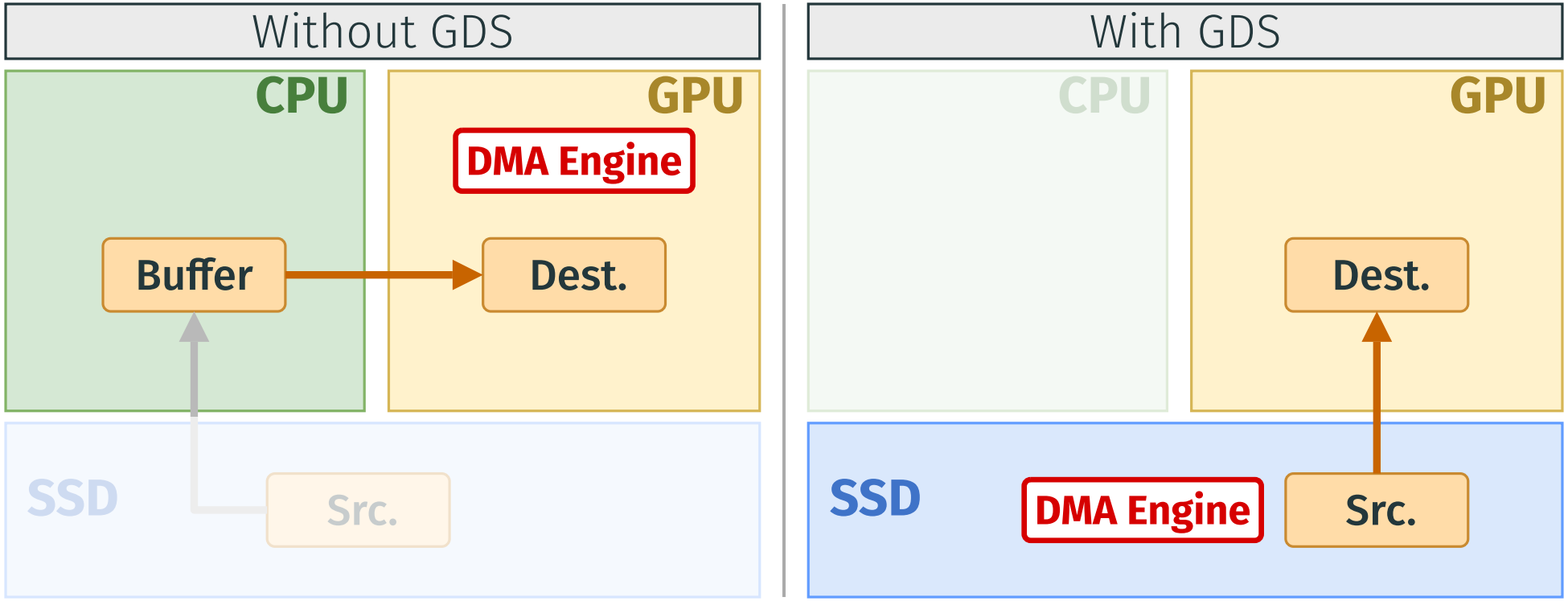
With 4 × PCIe 5.0 NVMe SSDs.

GDS Benefits: Host Memory BW

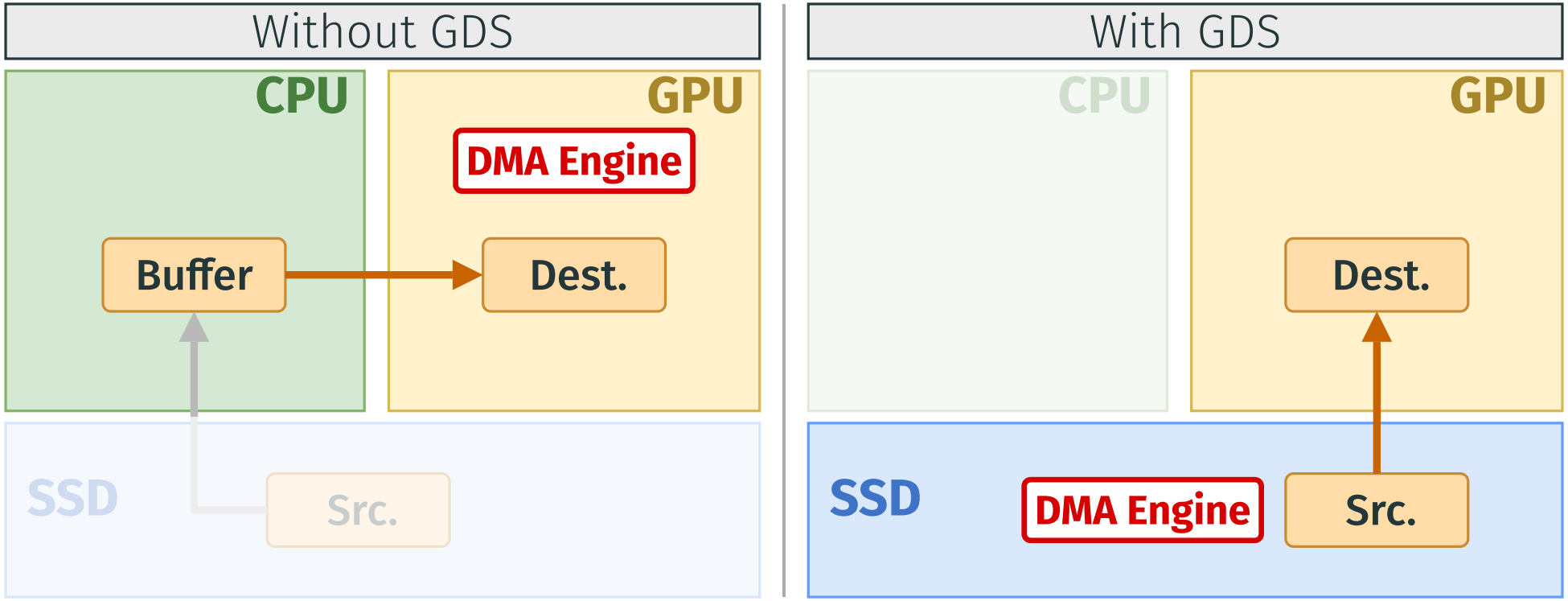


With 4 × PCIe 5.0 NVMe SSDs.

GDS Benefits: DMA Engines

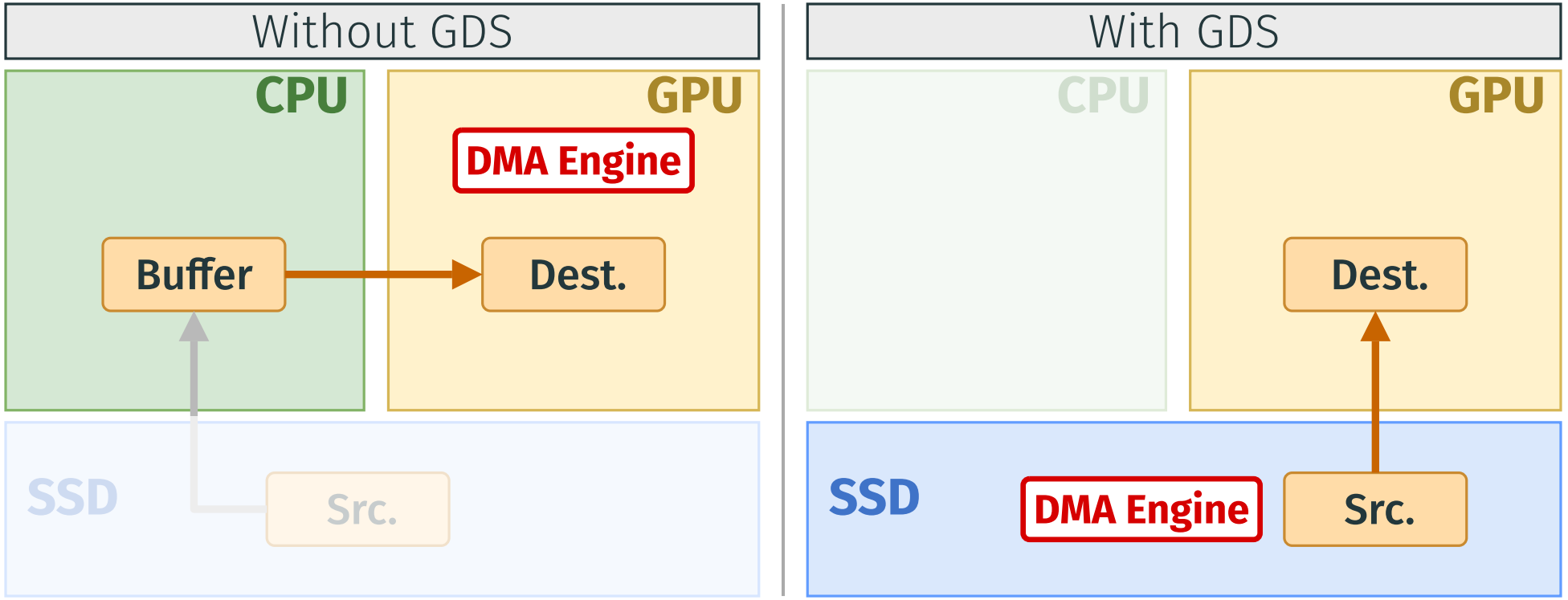


GDS Benefits: DMA Engines



1. DMA Documentation

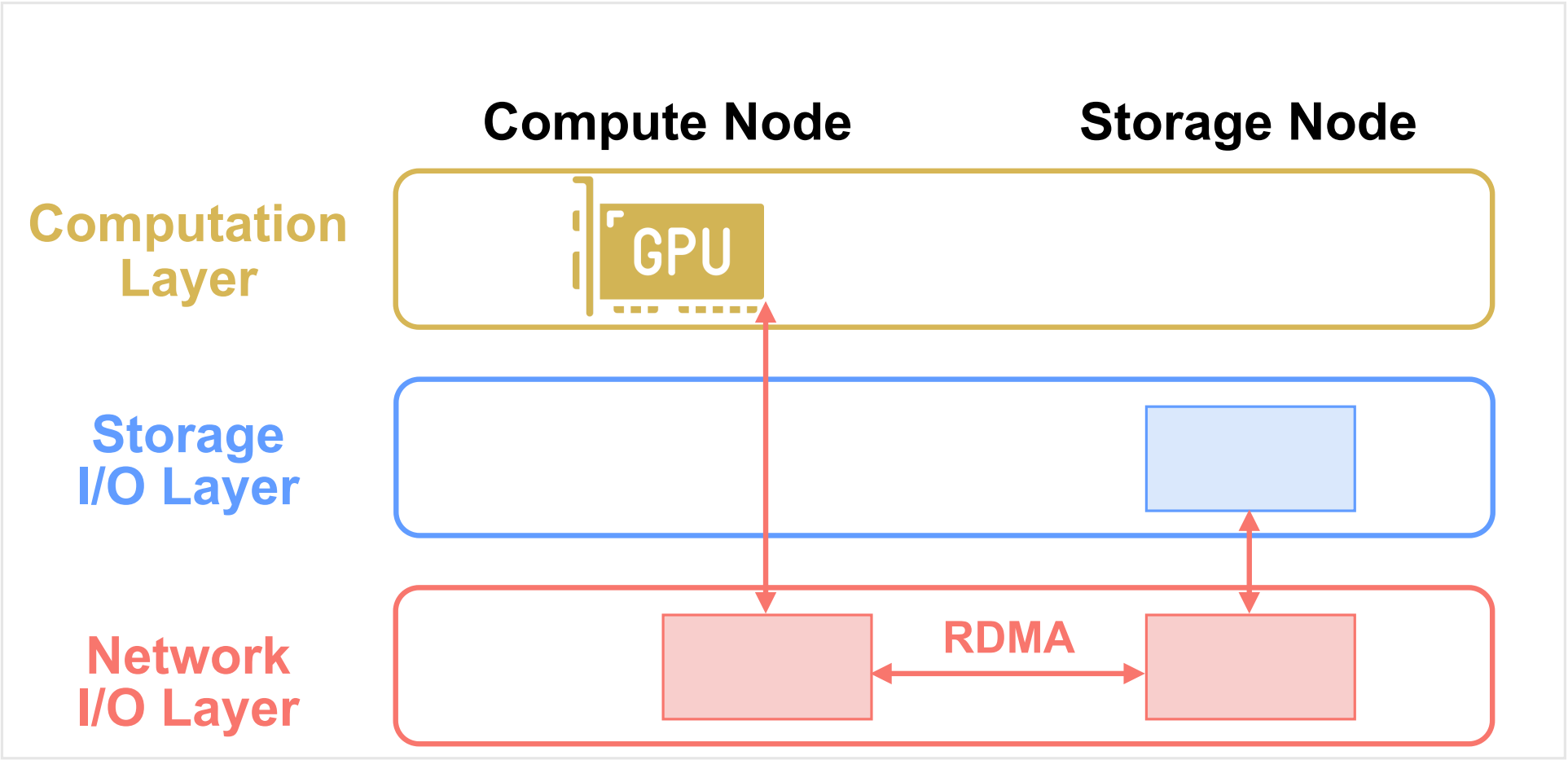
GDS Benefits: DMA Engines



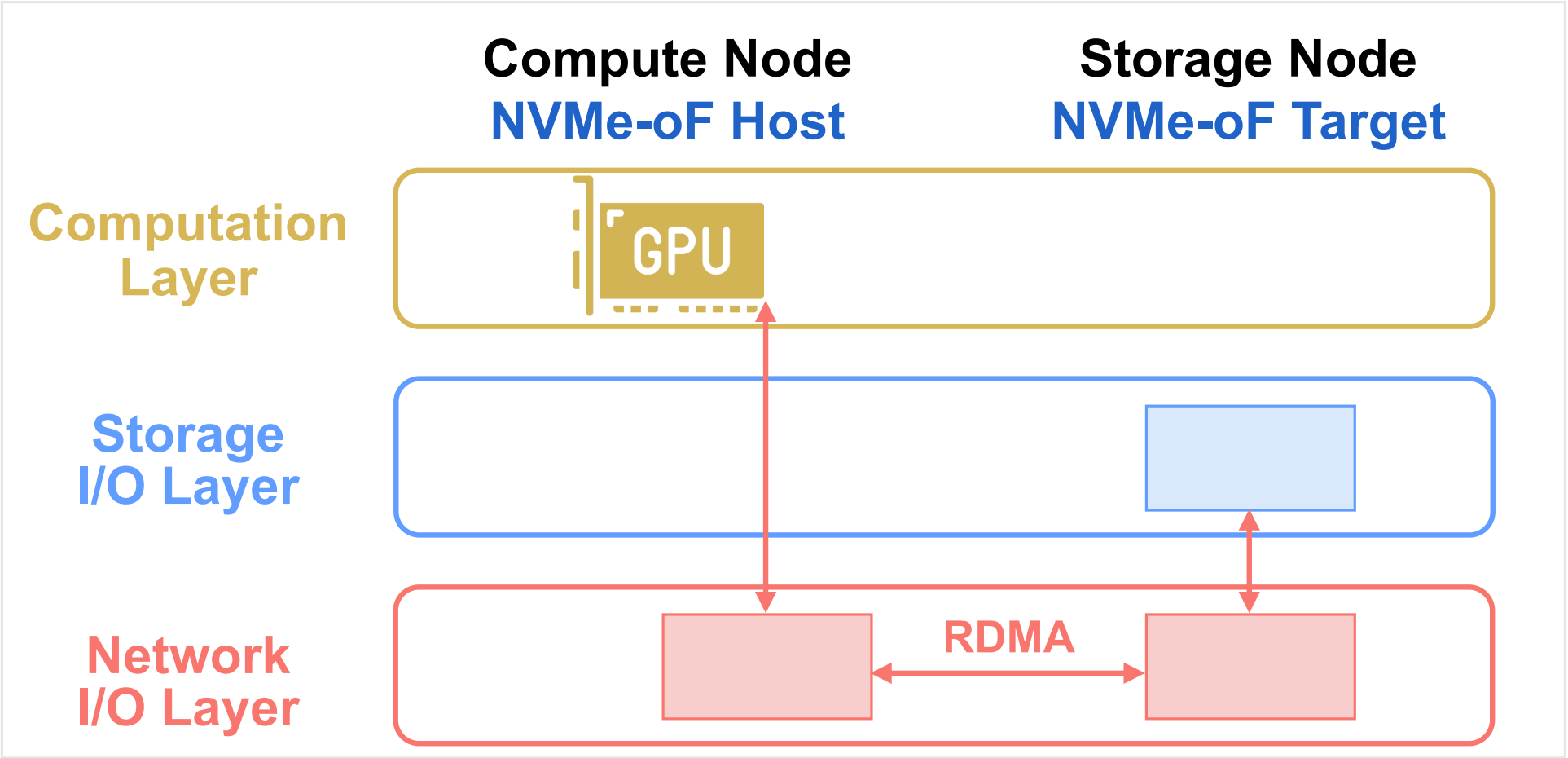
1. DMA Documentation

2. Less Head-of-Line Blocking

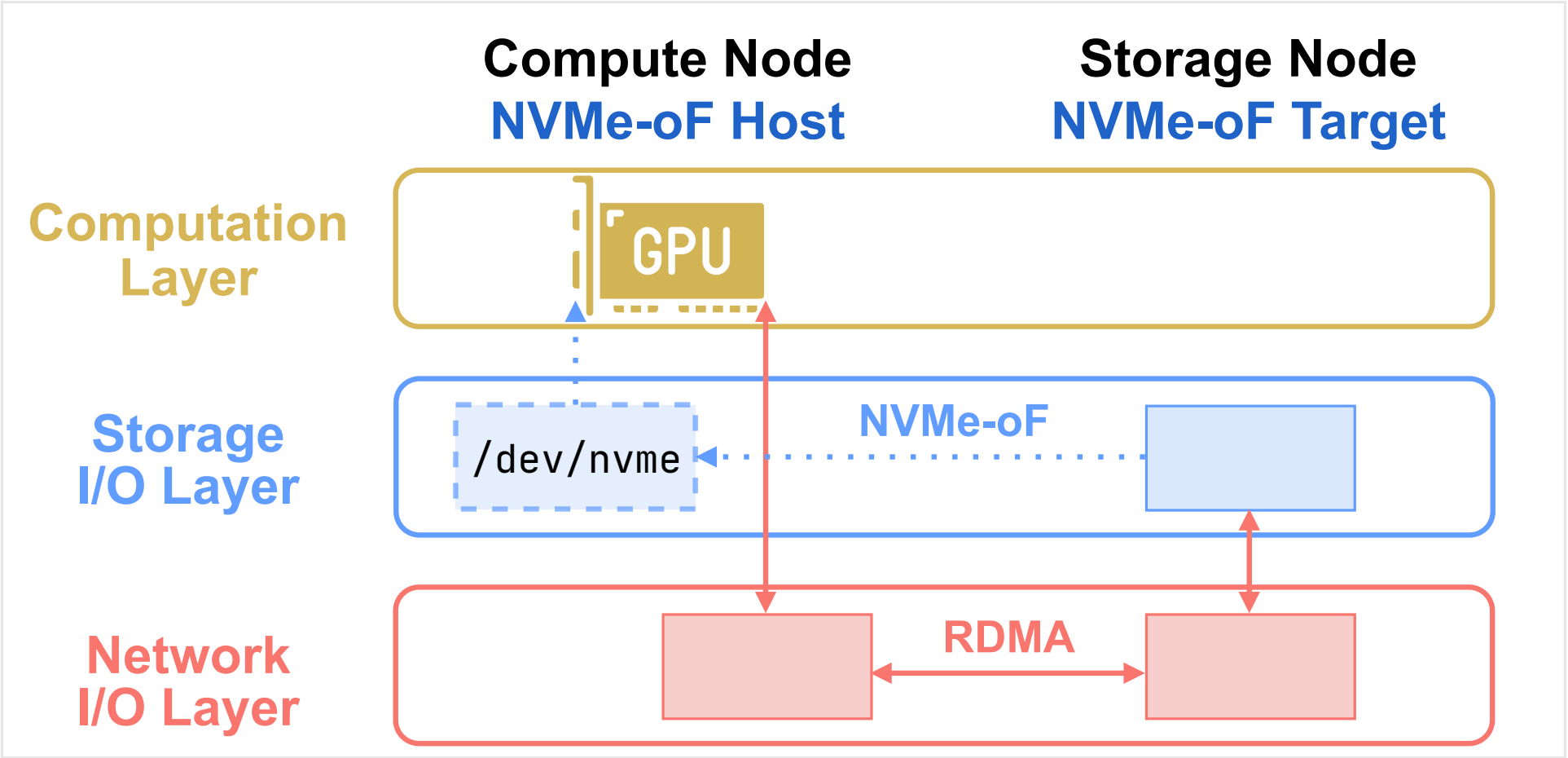
What About Remote Storage?



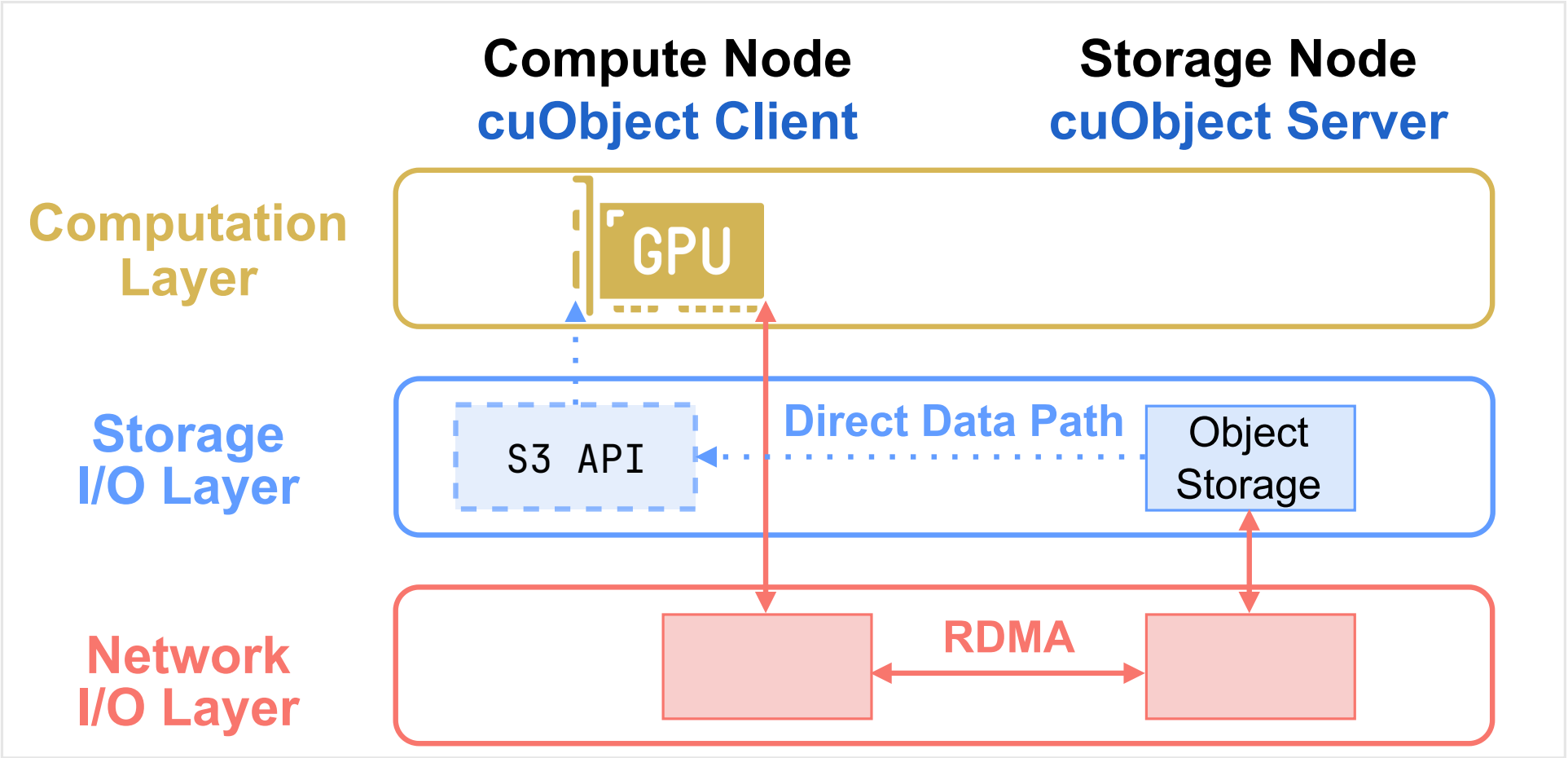
NVMe-oF: NVMe over Fabrics



NVMe-oF: NVMe over Fabrics



cuObject: GDS for On-Prem Objects



The Road Ahead:

How Close Are We to **Speed-of-Light** GPU Data Processing?

1. The Curse of THE REAL S3

What is AWS S3?

- Extremely cheap!

1. The Curse of THE REAL S3

What is AWS S3?

- Extremely cheap!
- Bandwidth: < 1 GB/s per connection
- Latency: ~100ms P99
- CPU-only: DNS, TLS, and HTTP

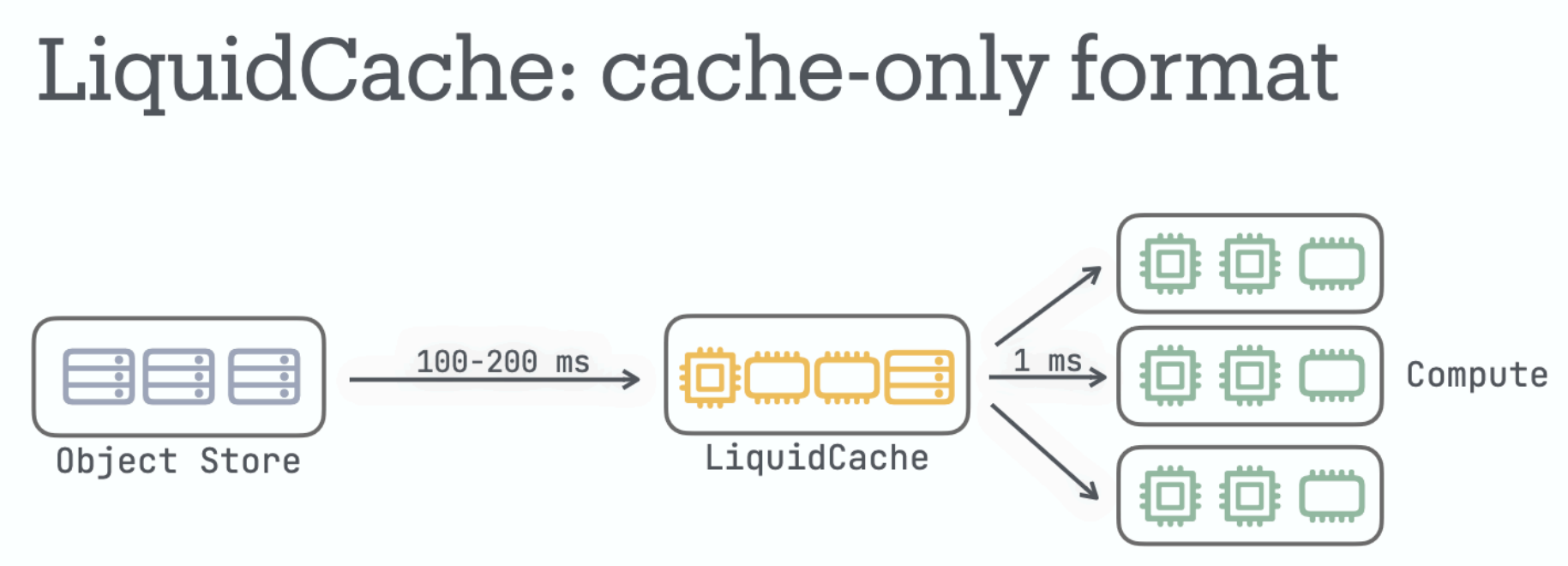
1. The Curse of THE REAL S3

Why AWS S3 is the Curse?

- Tail latency → GPU idle
- Low BW → GPU underutilized
- Data path back to CPU

Wasting expensive GPUs!

1. Caching Layer above AWS S3



Xiangpeng Hao et al., *LiquidCache: Efficient Pushdown Caching for Cloud-Native Data Analytics*, PVLDB 18(13)

2. Team Parquet or Team SomethingElse?

2. Team Parquet or Team SomethingElse?

- Yet Another File Format?

2. Team Parquet or Team SomethingElse?

File Format & GPU

- Hugging Face 🤗 × NVIDIA: acquisition
- **~50%** of Hugging Face 🤗 datasets in Parquet
- Parquet is for DB and AI!

2. Team Parquet or Team SomethingElse?

> *"If you want to go fast, go alone;*
> *if you want to go far, go together."*

Julien Le Dem, *Column Storage for the AI Era*, Iceberg Summit 2026

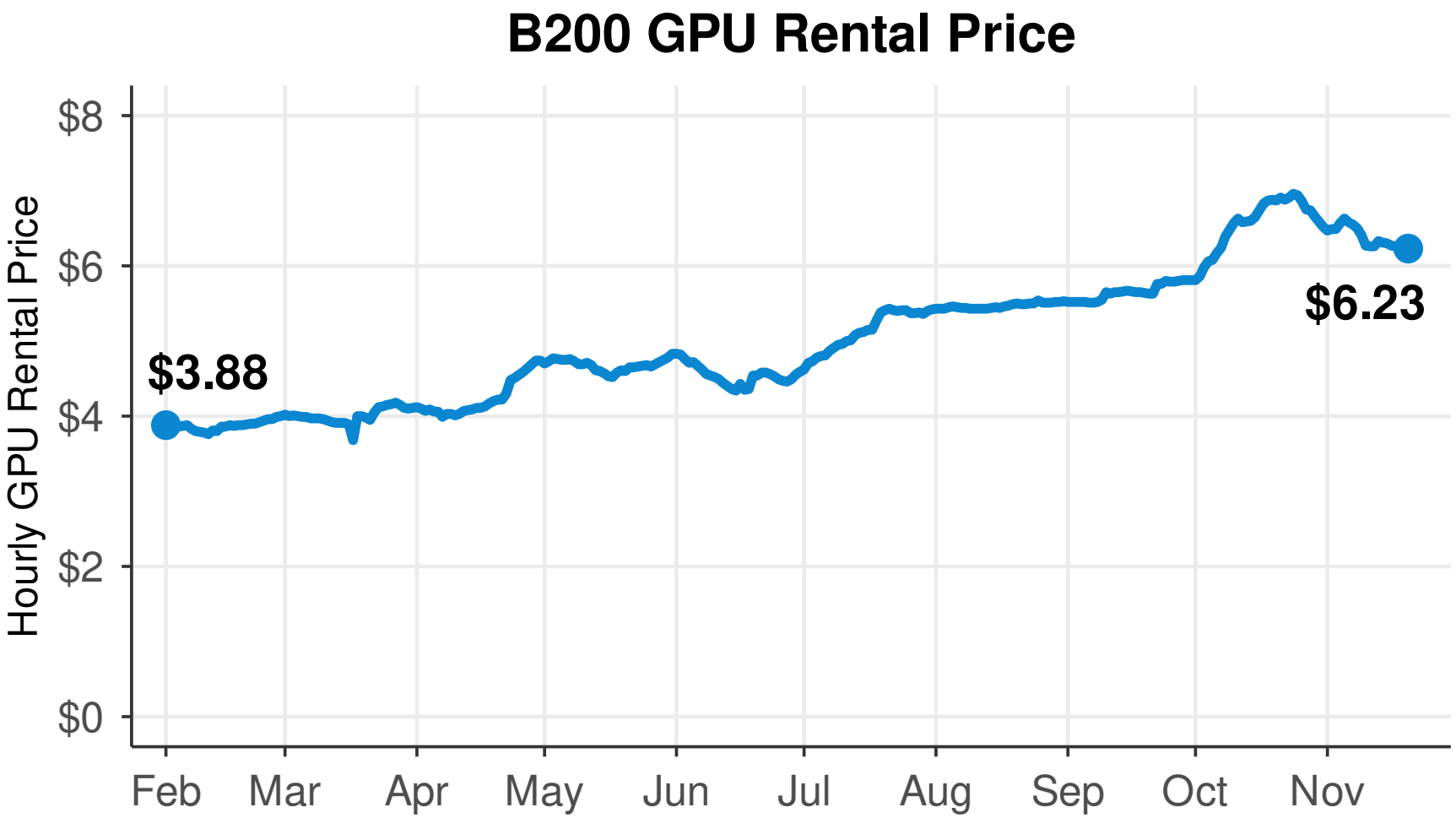
2. Team Parquet or Team SomethingElse?





- File Formats: Software Campus × Snowflake
- Working with Russell Spitzer

3. The Unspeakable Cost



Source: SemiAnalysis GPU Pricing Index

3. The Unspeakable Cost

> *“There’s a lot more money behind LLMs today. So why use GPUs for DB?”*

> *“... would be good to show the GPU compute and memory utilization ...”*

3. The Unspeakable Cost

What is the perfect GPU for DB?

- Large HBM
- Less SMs
- Recent PCIe Gen
- Low monetary cost

Thanks For Your Attention!

Acknowledgement

Special thanks to my colleague Nils for our many fruitful discussions!

Acknowledgement

Special thanks to my colleague Nils for our many fruitful discussions!

These slides are inspired by the following works. Many thanks to the authors:

- **DBs+GPUs: Where are we and what's next?**, Bowen Wu, Systems Group ETHz
- **G-ALP: Rethinking Light-weight Encodings for GPUs** @ DaMoN25, Sven Hepkema, ex-CWI DB & Systems Group ETHz
- **GPU Databases – The New Modality of Data Analytics**, Xiangyao Yu, University of Wisconsin-Madison
- **SIGMOD 26 Keynote: A New Golden Era for Data Management**, Gustavo Alonso, Systems Group ETHz
- **The Composable Codex**, Voltron Data