How Close Are We to GPU Data Processing?

Toward Disaggregated Data Systems in the AI Hardware Era

Jigao Luo

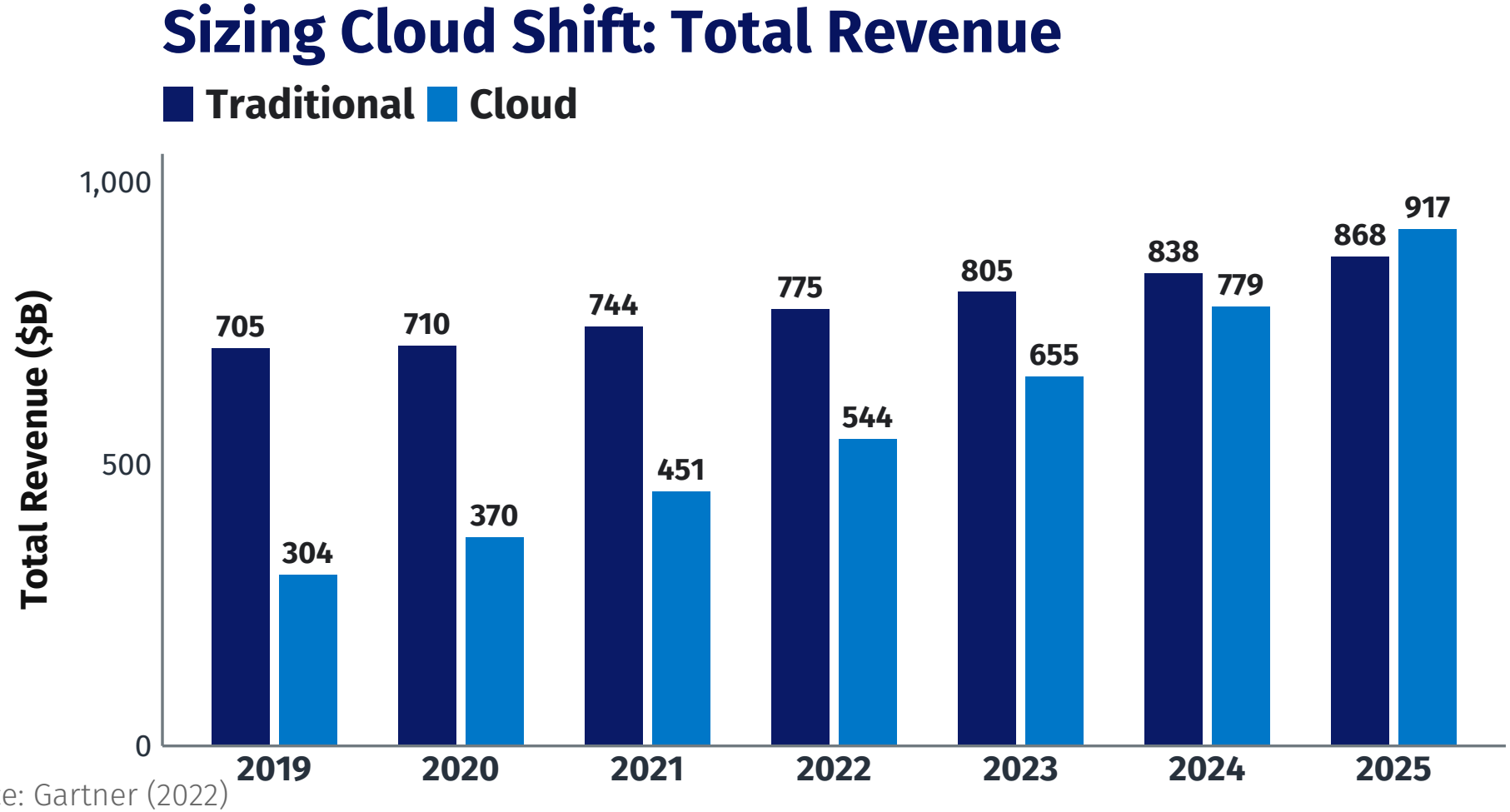
2026

Systems Group, Technische Universität Darmstadt

TECHNISCHE UNIVERSITÄT DARMSTADT

SYSTEMS

Cloud Is a Success Story



Source: Gartner (2022)

Cloud Database Is Also a Success Story

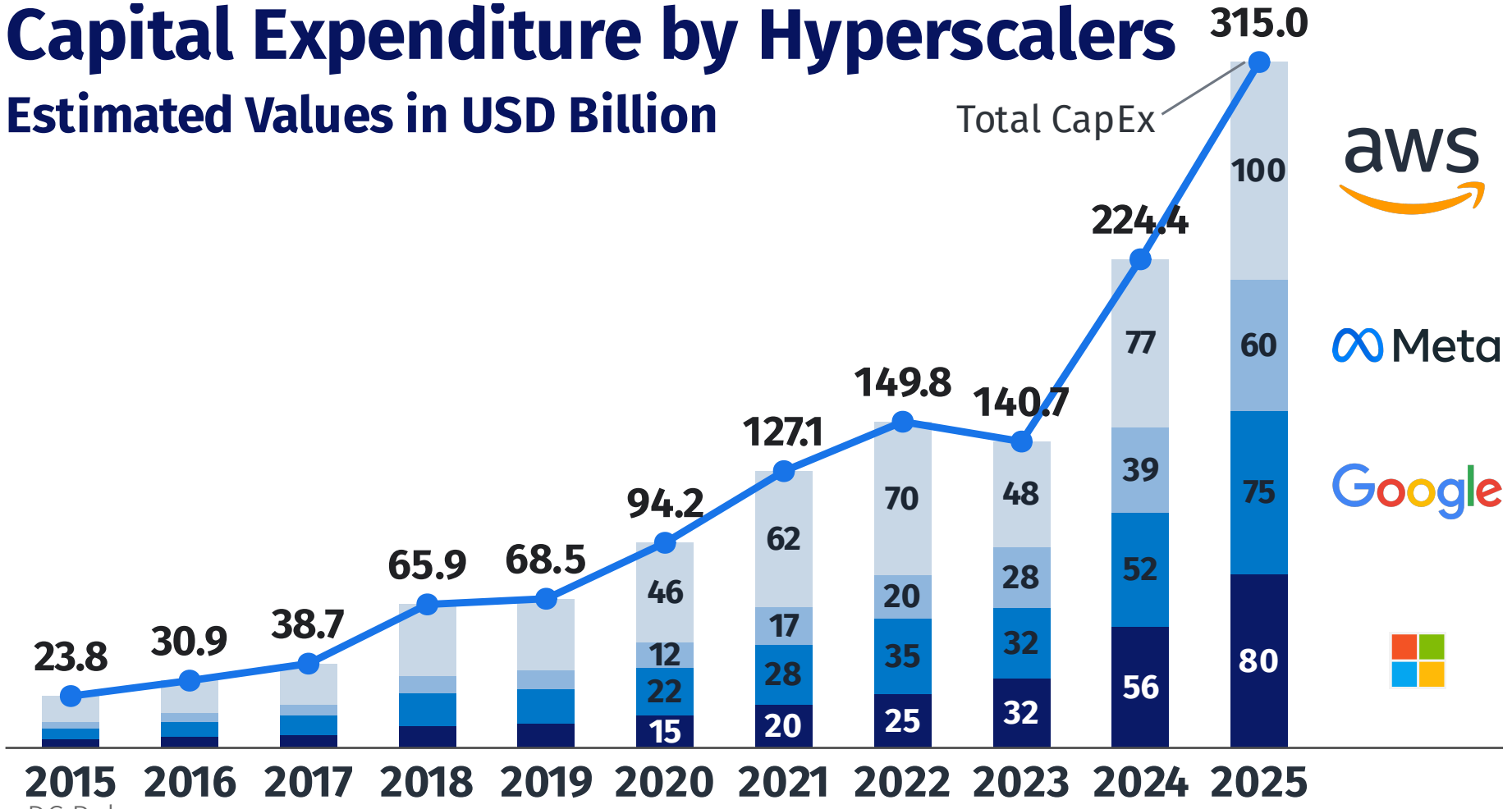
SaaS IPO Ranking

Rank	Company	IPO Year	IPO Size ($B)
1	**Snowflake**	2020	3.4
2	UiPath	2021	1.34
3	Dropbox	2018	0.756
4	Zoom	2019	0.751
5	Datadog	2019	0.648

AI Investment Is Shaping the Future

Capital Expenditure by Hyperscalers

Estimated Values in USD Billion



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?**

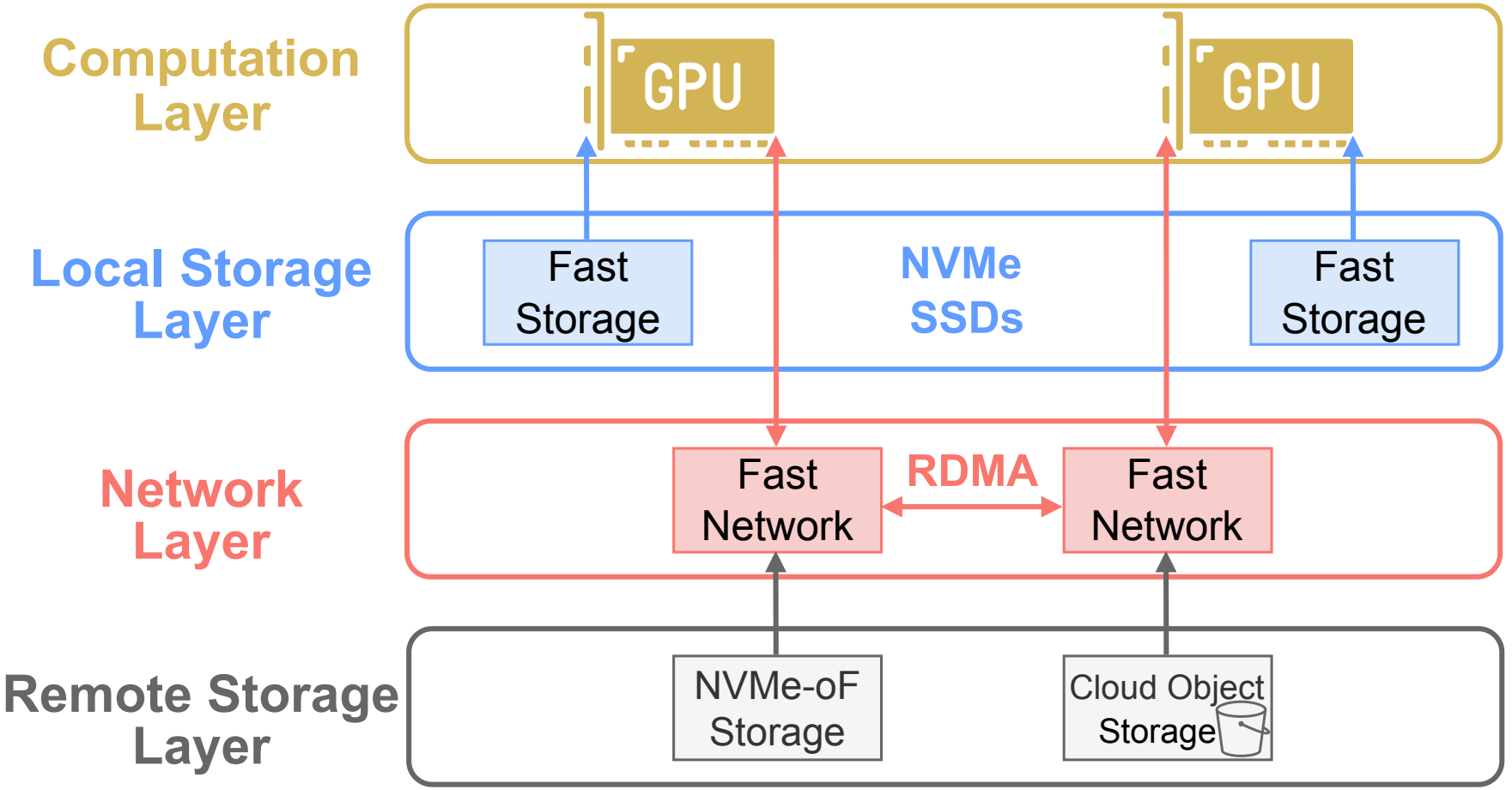
How to Ride the AI Investment Wave for DB?

- 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

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
- Focus shifting from computation to I/O

My Research

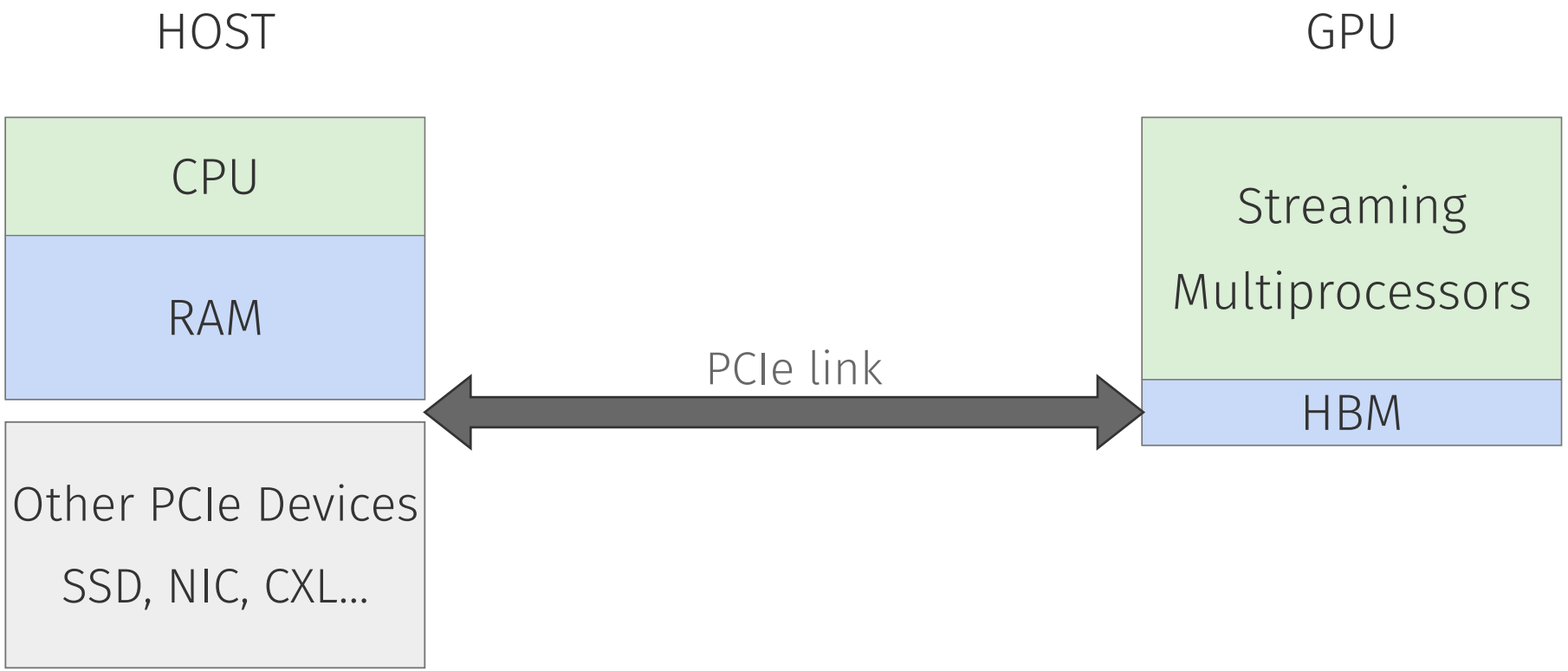

Computation Layer
GPU
GPU
Local Storage Layer
Fast Storage
NVMe SSDs
Fast Storage
Network Layer
Fast Network
RDMA
Fast Network
Remote Storage Layer
NVMe-oF Storage
Cloud Object Storage

Talk Structure

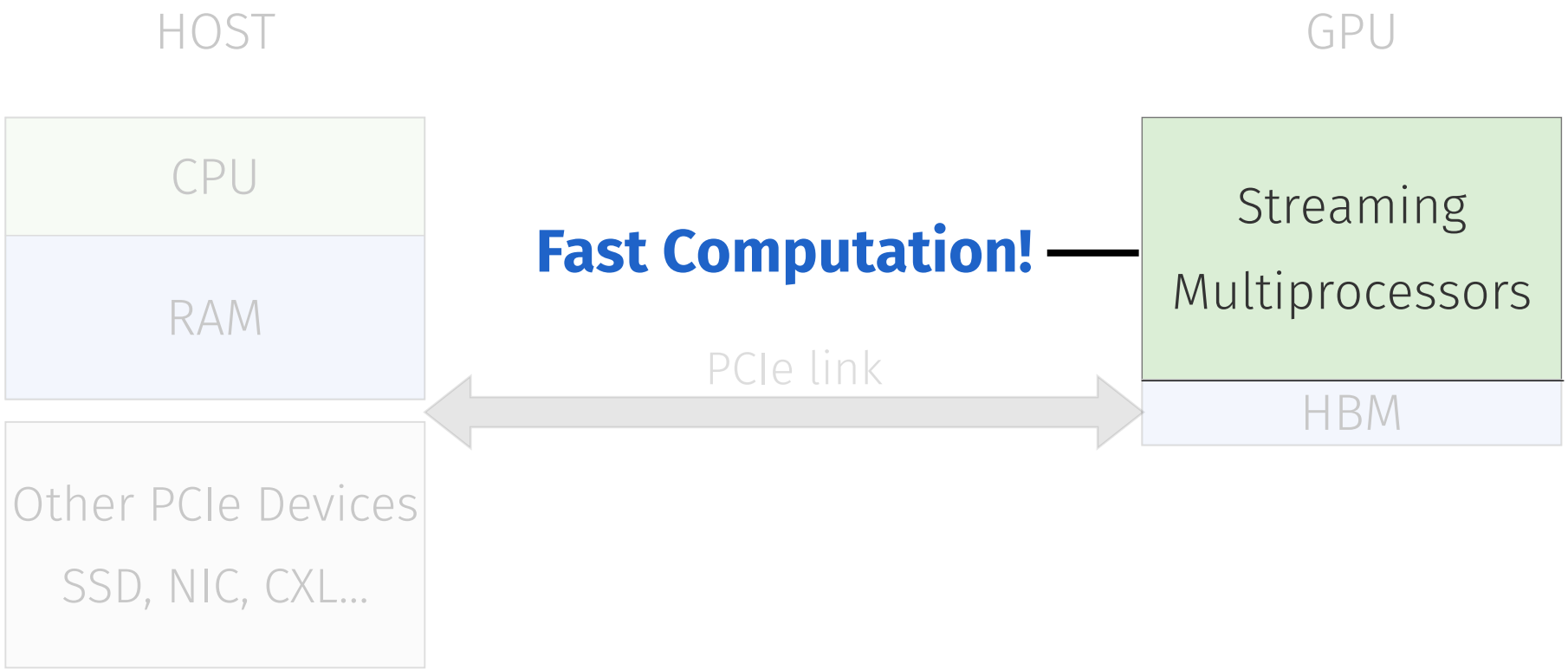
- Background: GPU Data Processing
- My Research
  - Overview: Datenbank-Spektrum 26
  - GPU & Parquet: DaMoN 26 🏆
  - PystachIO: VLDB 26
- The Road Ahead

Background: GPU Data Processing

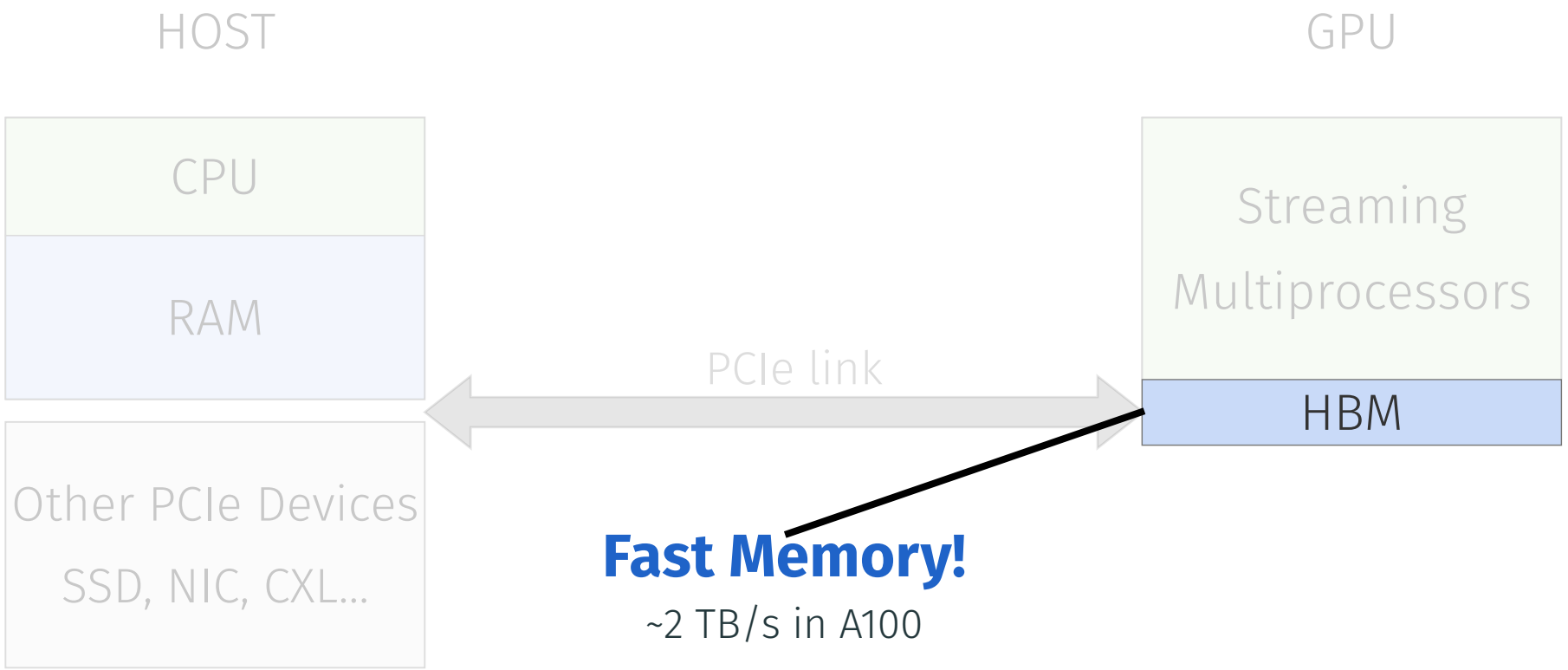
GPU System Primitives



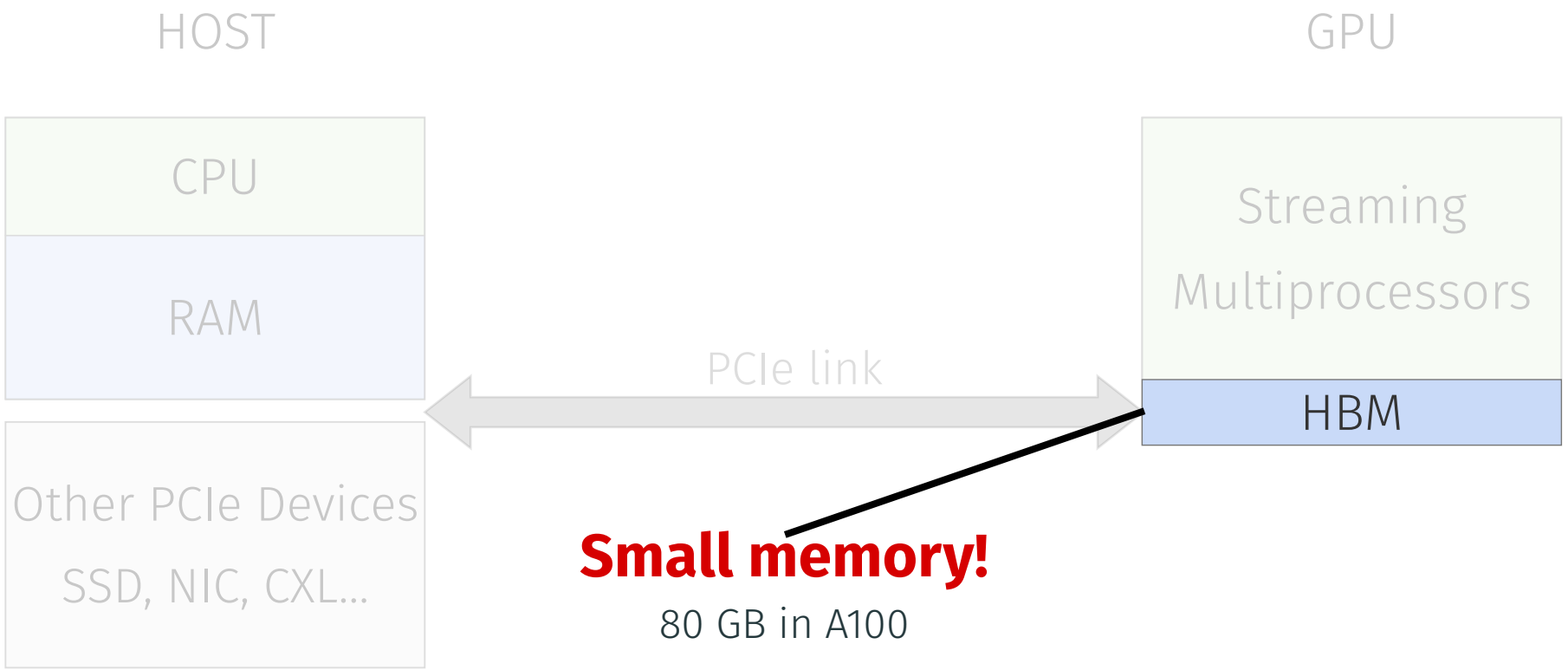
GPU System Primitives



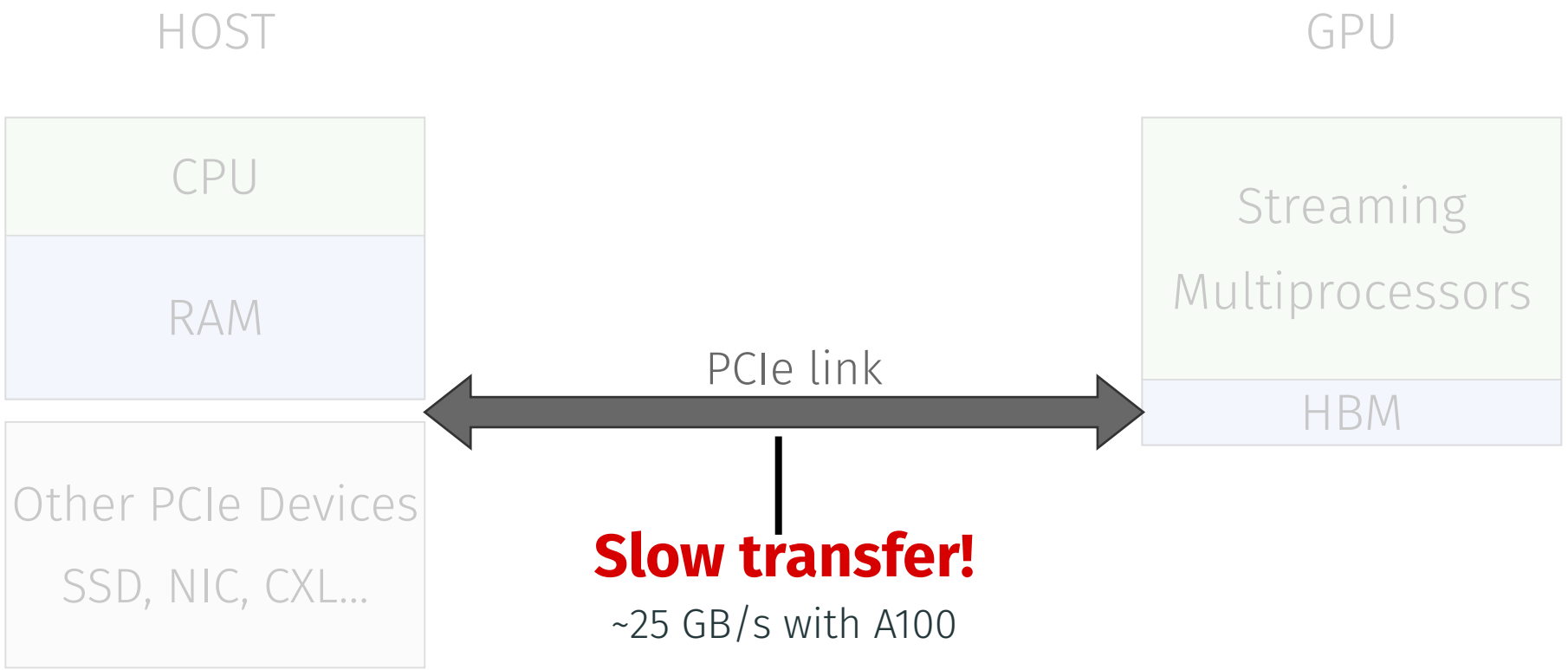
GPU System Primitives



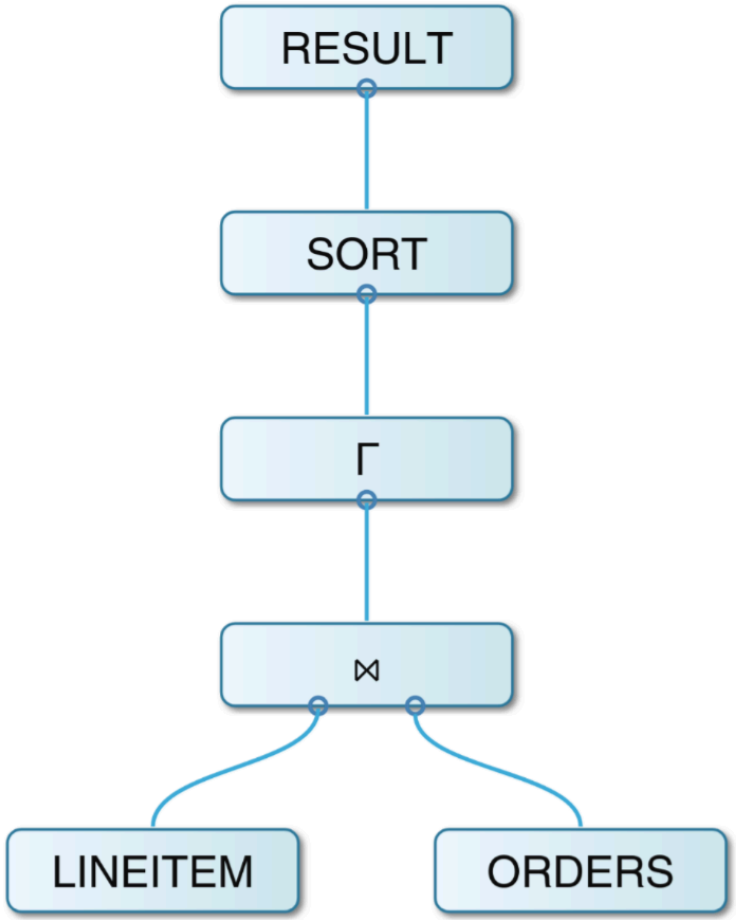
GPU System Primitives: Problems & Challenges



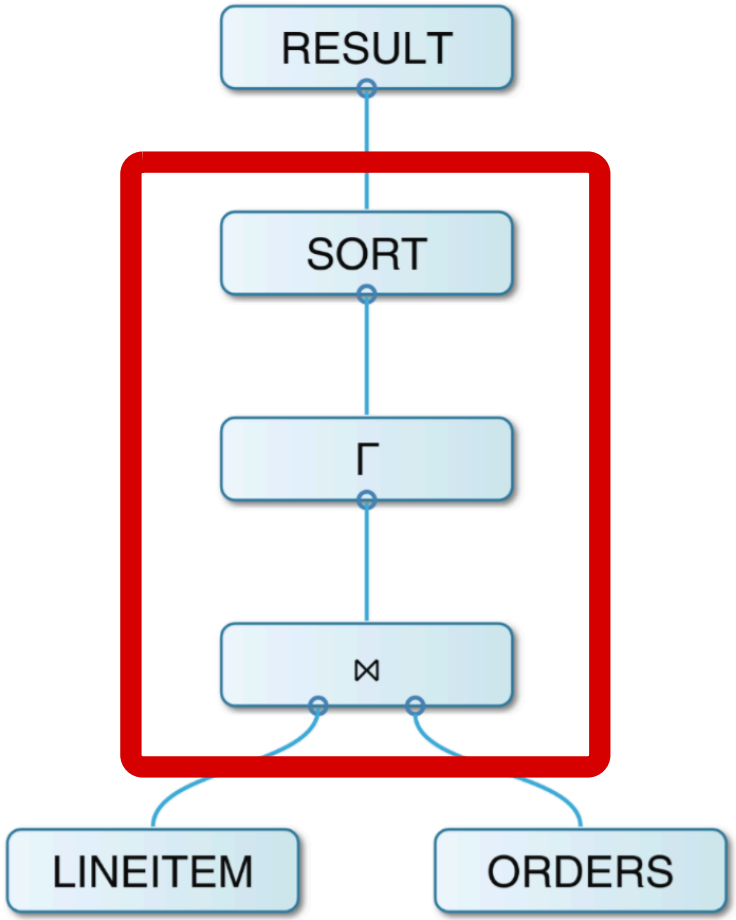
GPU System Primitives: Problems & Challenges



GPU Data Systems

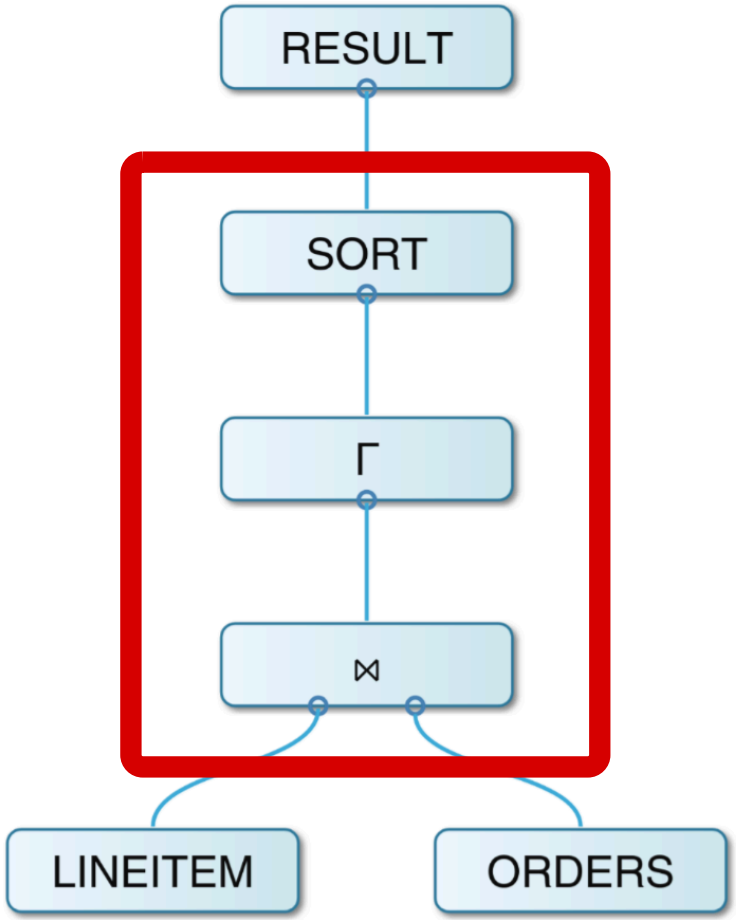


GPU Data Systems



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

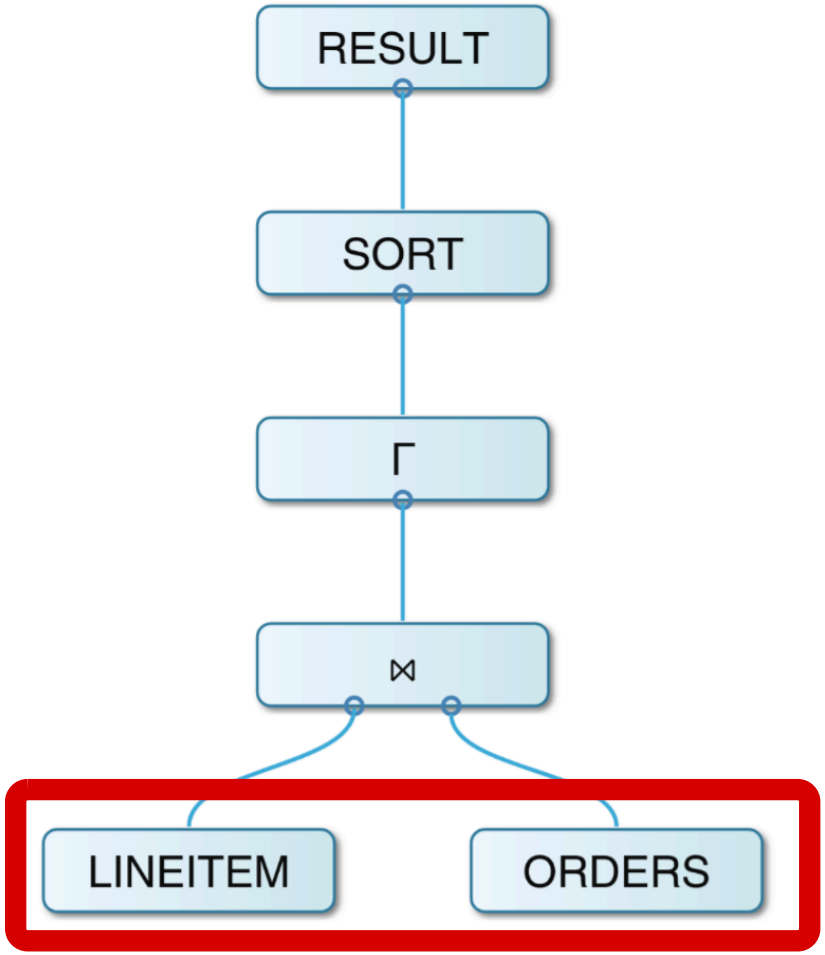
GPU Data Systems



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

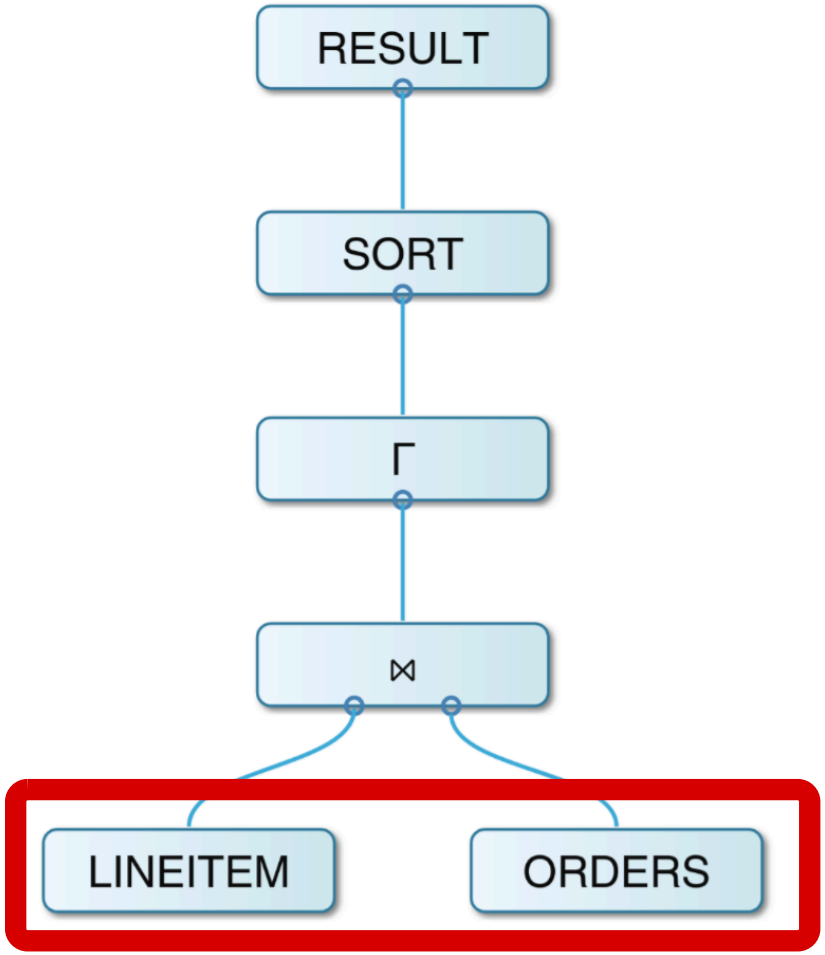
Not in this talk

GPU Data Systems



Where to put base tables?

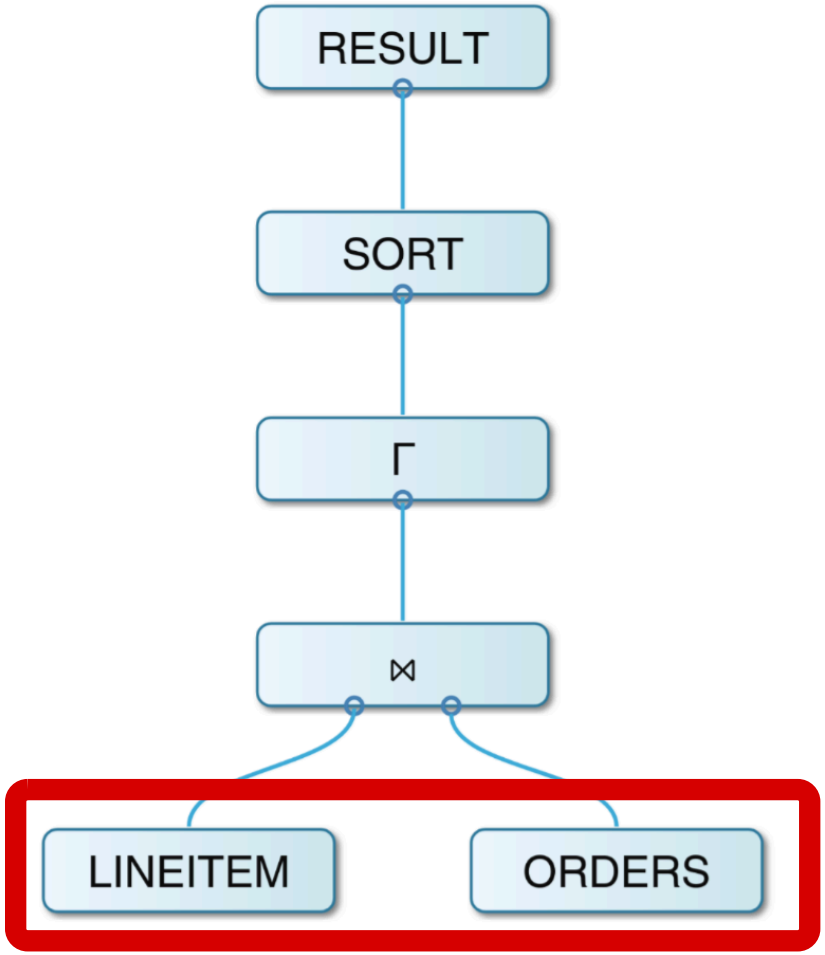
GPU Data Systems



Where to put base tables?

- GPU Memory: HBM

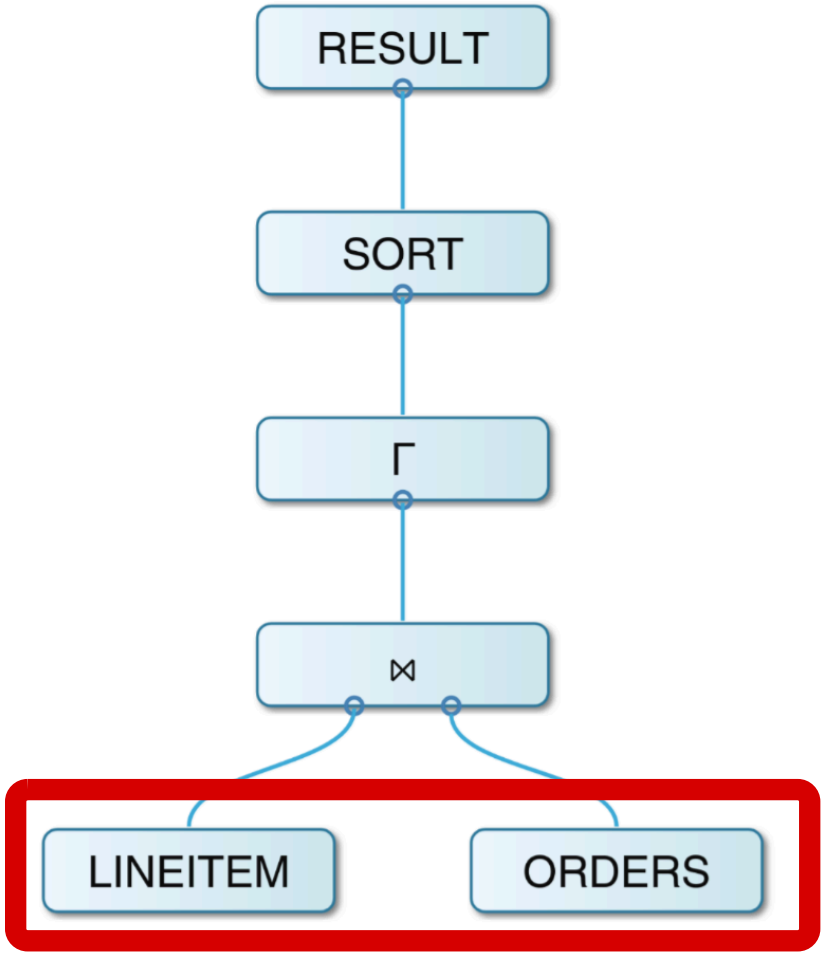
GPU Data Systems



Where to put base tables?

- GPU Memory: HBM
- HBM of neighbour GPUs: NVLink

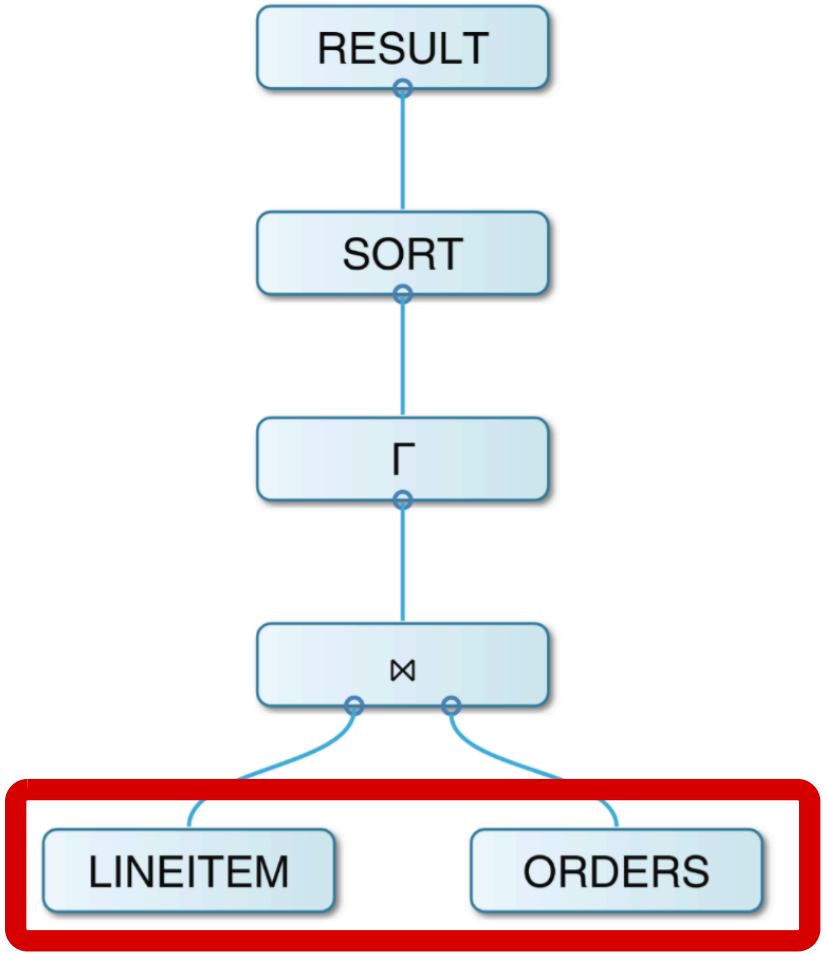
GPU Data Systems



Where to put base tables?

- GPU Memory: HBM
- HBM of neighbour GPUs: NVLink
- CPU Memory

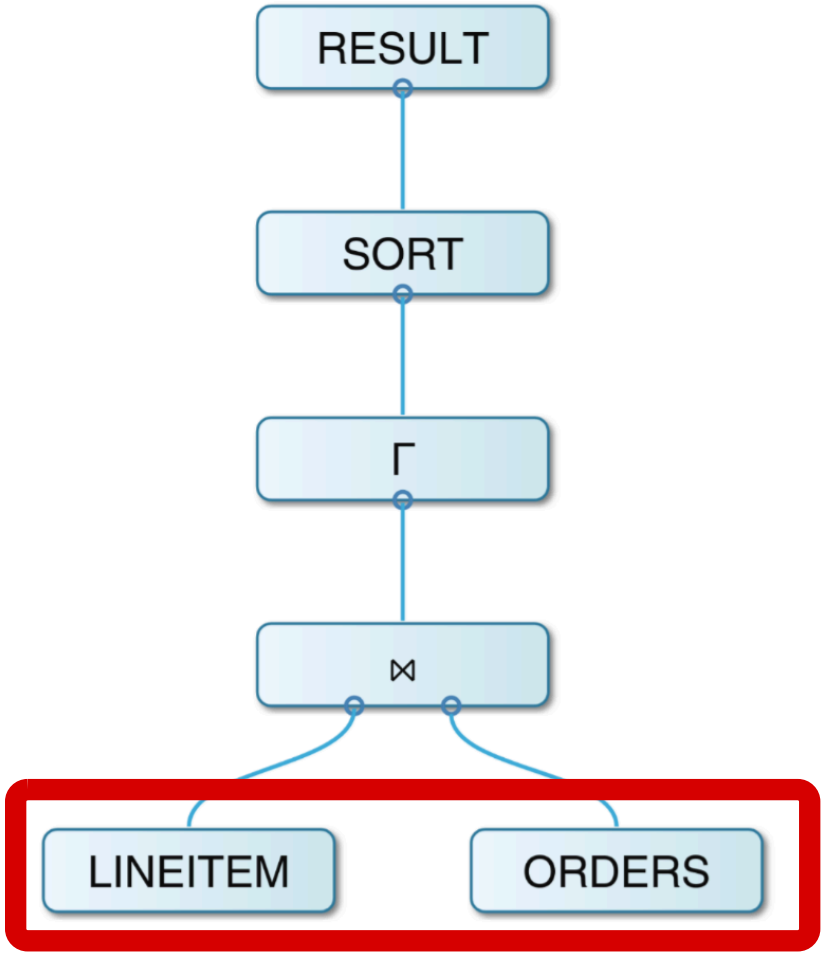
GPU Data Systems



Where to put base tables?

- GPU Memory: HBM
- HBM of neighbour GPUs: NVLink
- CPU Memory
- SSDs

GPU Data Systems



Where to put base tables?

- GPU Memory: HBM
- HBM of neighbour GPUs: NVLink
- CPU Memory
- SSDs
- NICs: remote storage

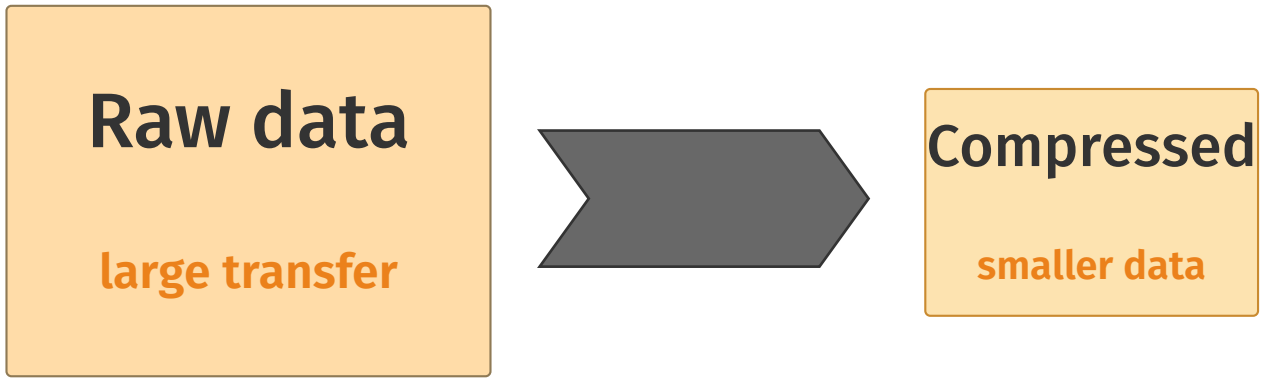
Figure Source: HyPer DB

Common Technique 1: Compression

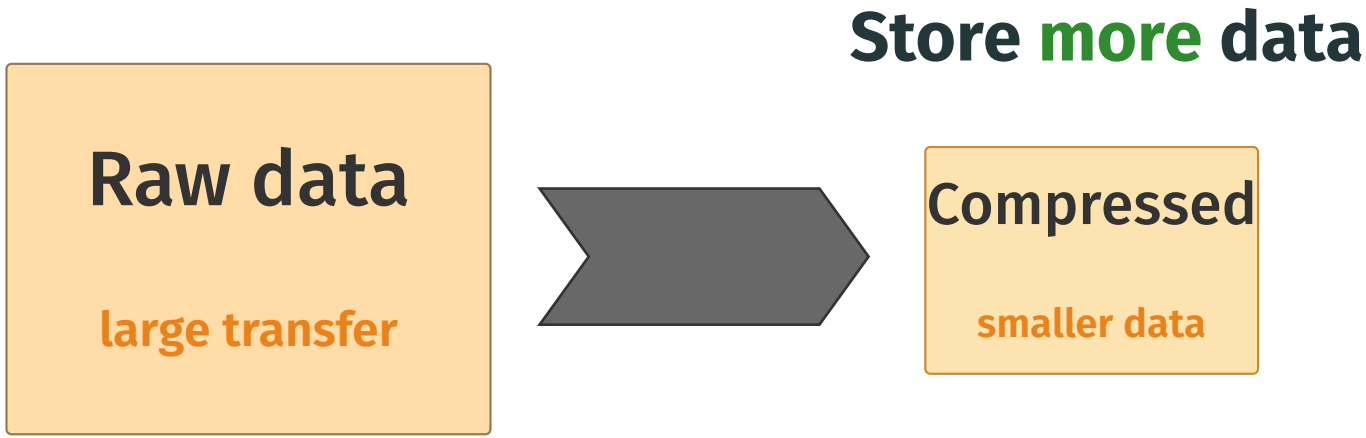
Raw data

large transfer

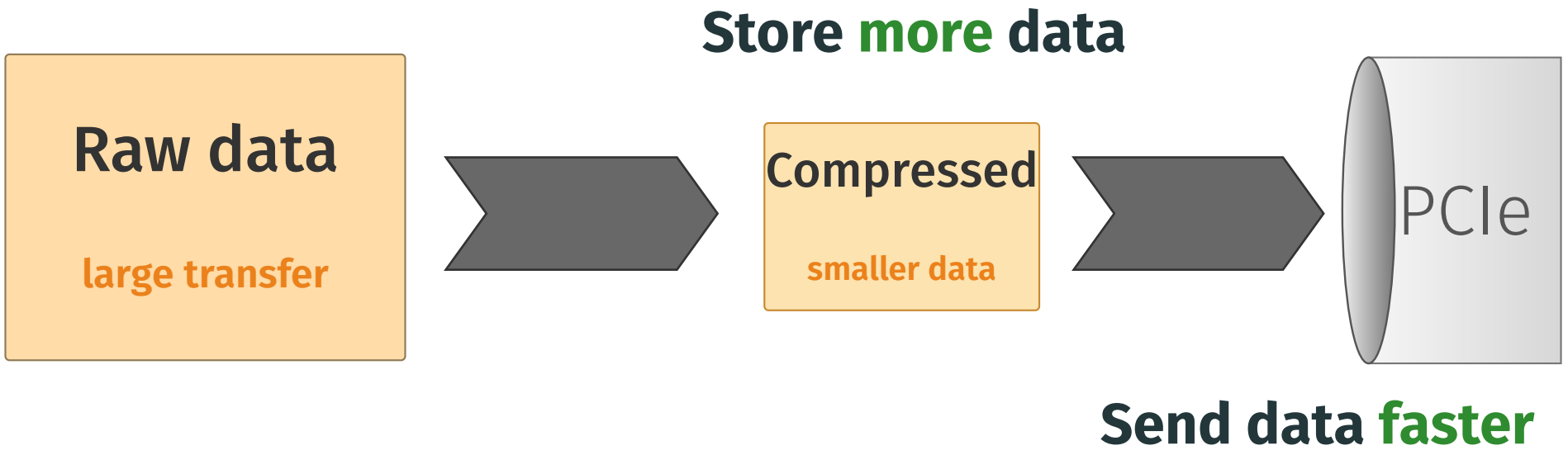
Common Technique 1: Compression



Common Technique 1: Compression

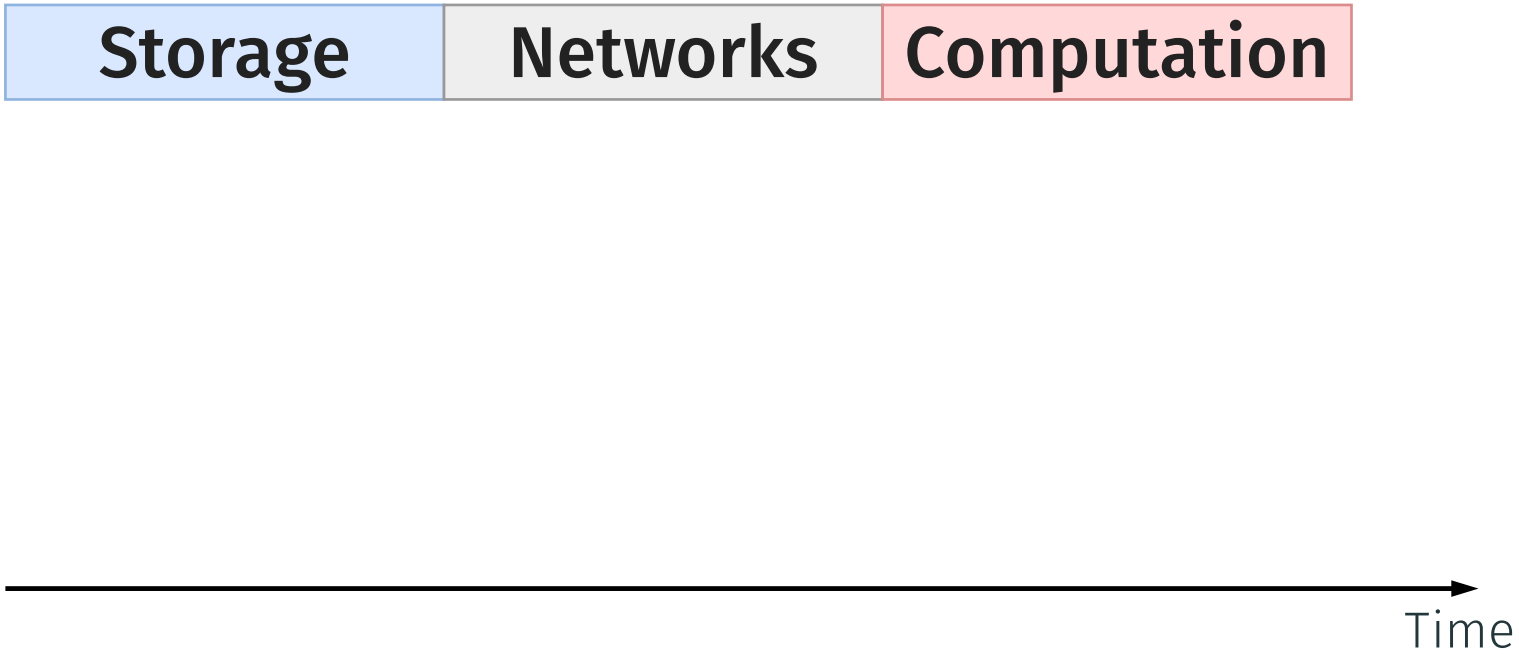


Common Technique 1: Compression



Common Technique 2: Overlapping

Blocking

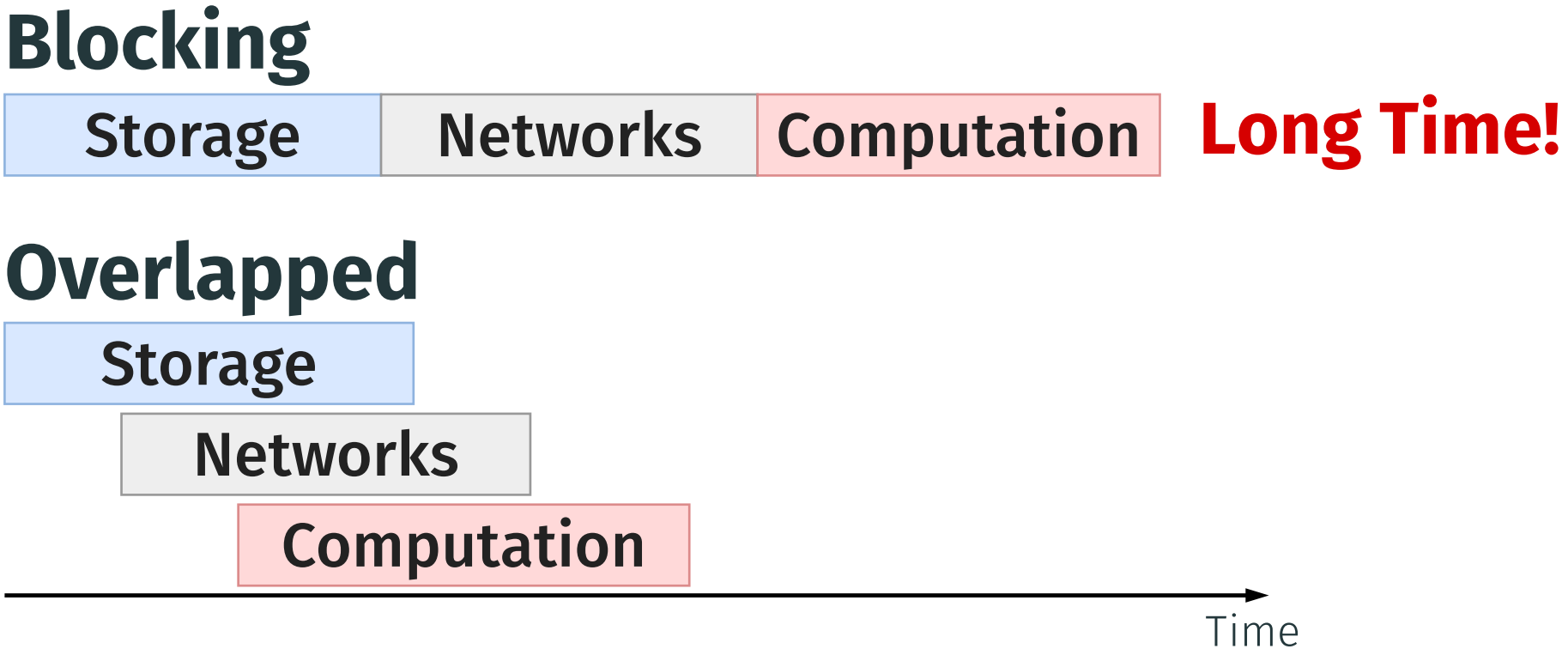


Common Technique 2: Overlapping

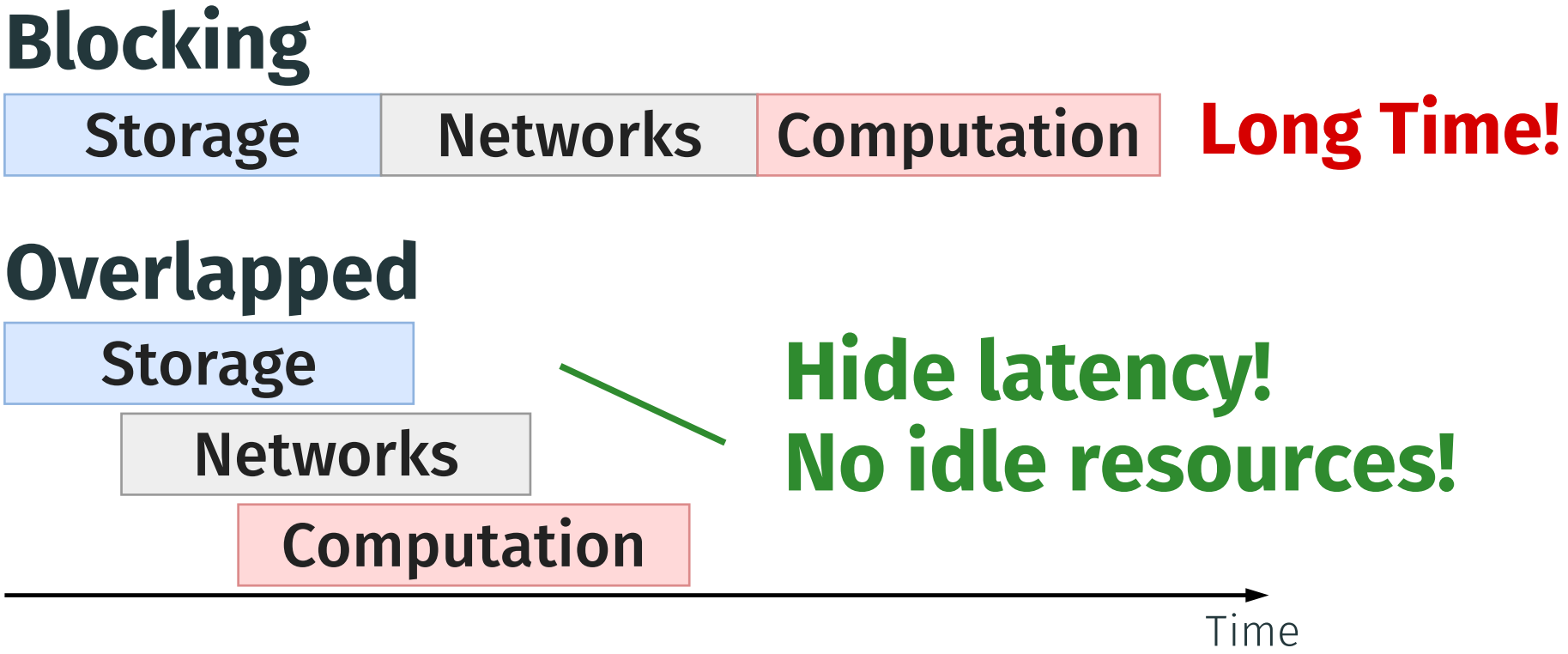
Blocking

Long Time!

Common Technique 2: Overlapping

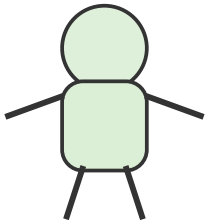


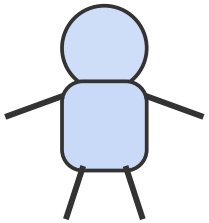
Common Technique 2: Overlapping



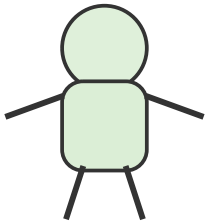
My Research

My First Meeting with Carsten



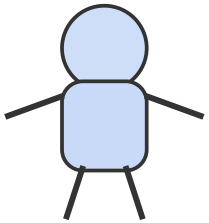


My First Meeting with Carsten



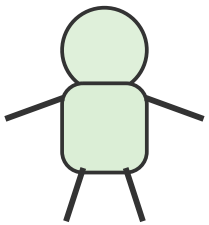
Carsten

Do you know GSL?



Jigao

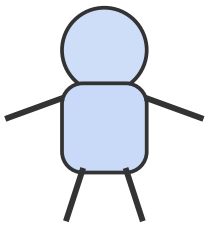
My First Meeting with Carsten



Carsten

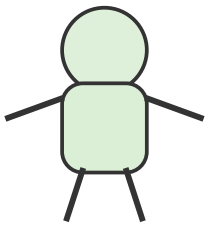
Do you know GSL?

No.



Jigao

My First Meeting with Carsten

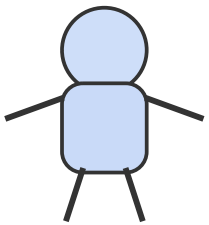


Carsten

Do you know GSL?

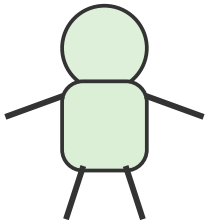
Do you know Matteo Interlandi?

No.



Jigao

My First Meeting with Carsten



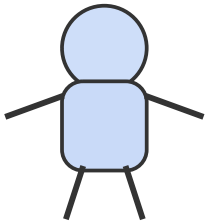
Carsten

Do you know GSL?

Do you know Matteo Interlandi?

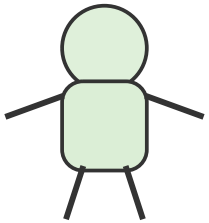
No.

No.



Jigao

My First Meeting with Carsten



Carsten

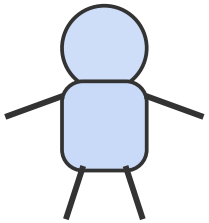
Do you know GSL?

No.

Do you know Matteo Interlandi?

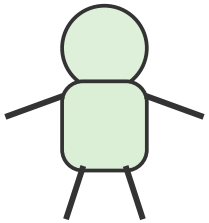
No.

Do you know the TQP paper from Matteo and GSL?



Jigao

My First Meeting with Carsten



Do you know GSL?

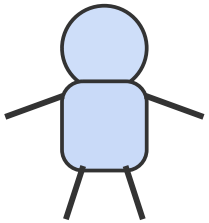
No.

Do you know Matteo Interlandi?

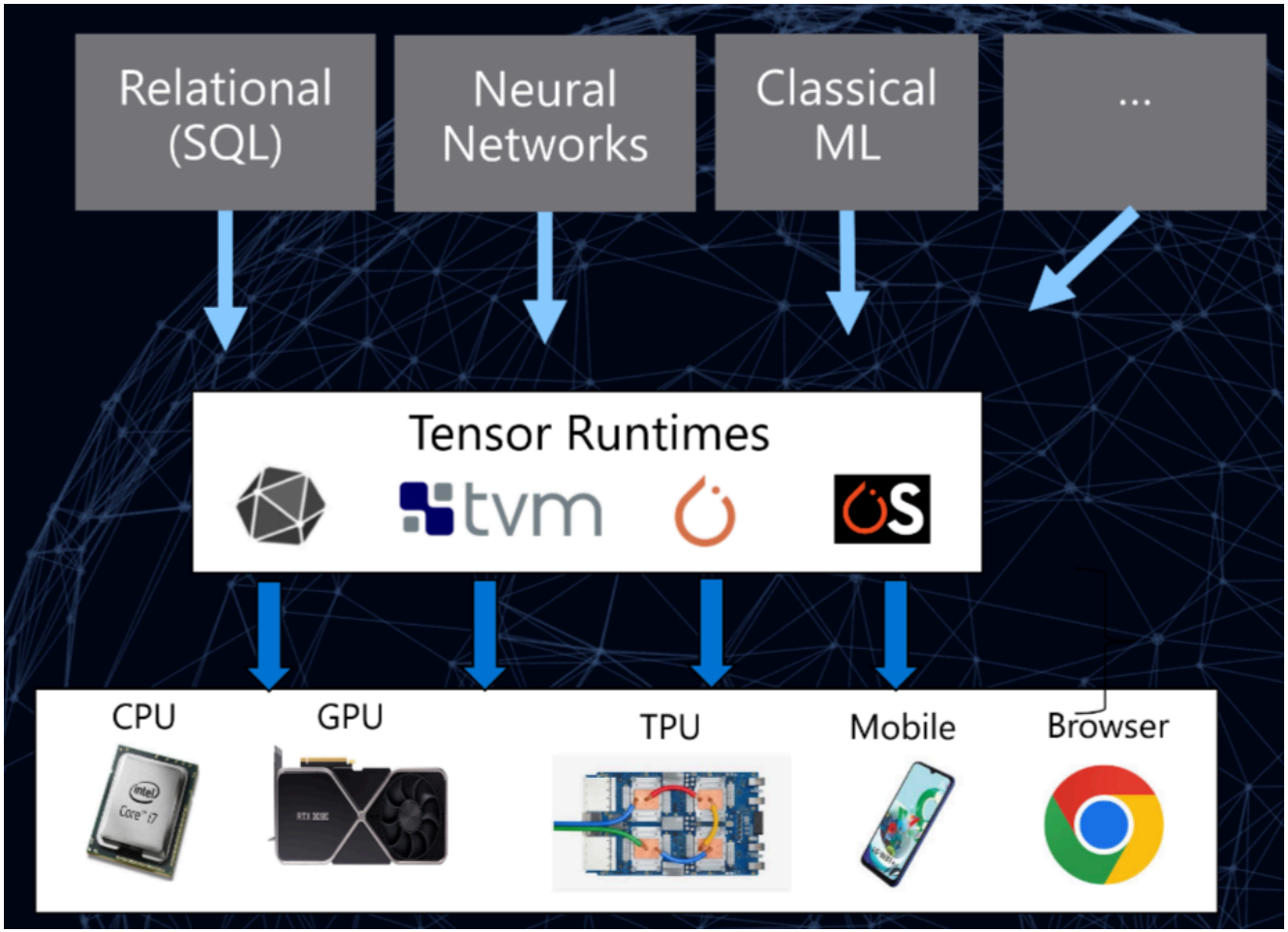
No.

Do you know the TQP paper from Matteo and GSL?

No.



TQP & Matteo



Source: Carlo Curino, “Tensor Query Processing: How Do We Ride the AI Investment Wave for Database Analytics?” (SEBD 2023)

TQP & Matteo

Matteo Interlandi • 1st
Principal Scientist Manager at Microsoft GSL
17h •

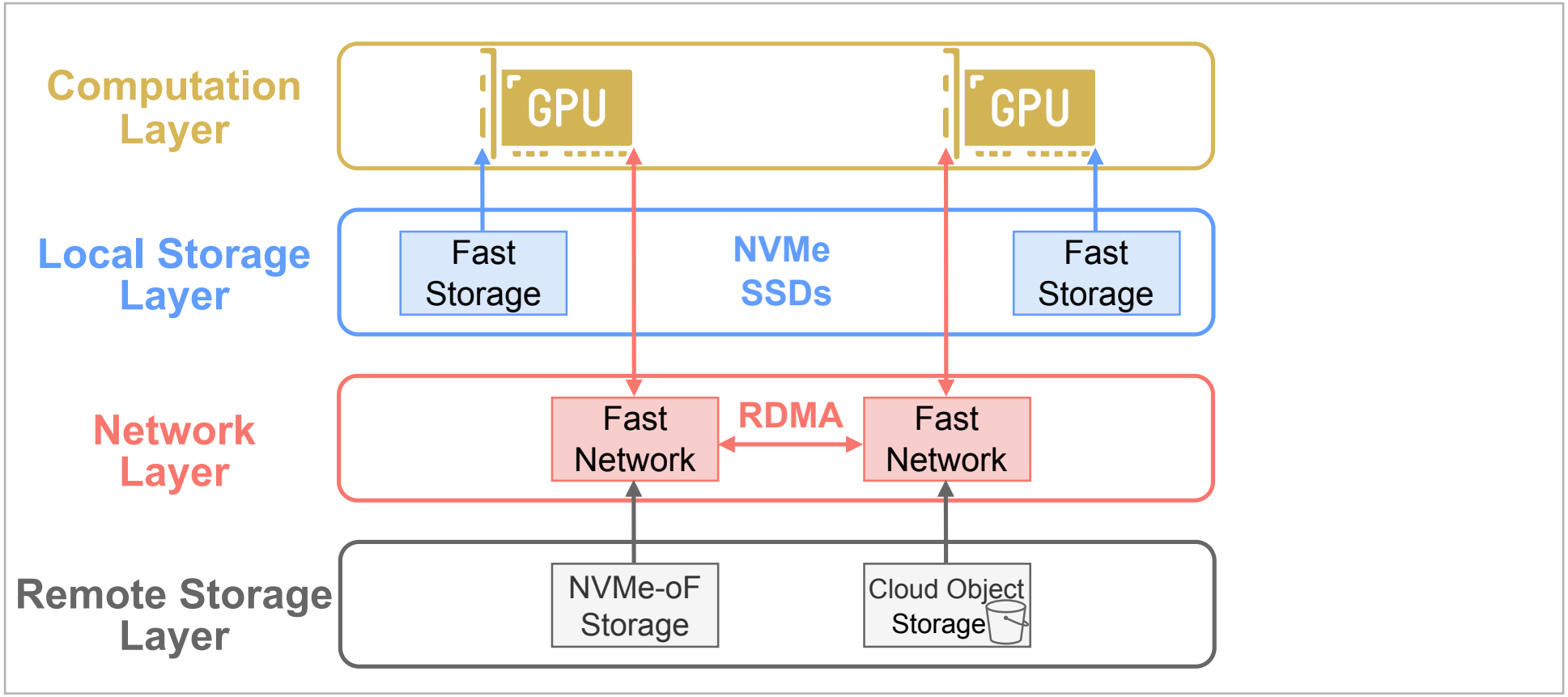
Five years ago, we made a slightly crazy bet: what if SQL could run on the same stack that powers AI?

This week, that bet came full circle.

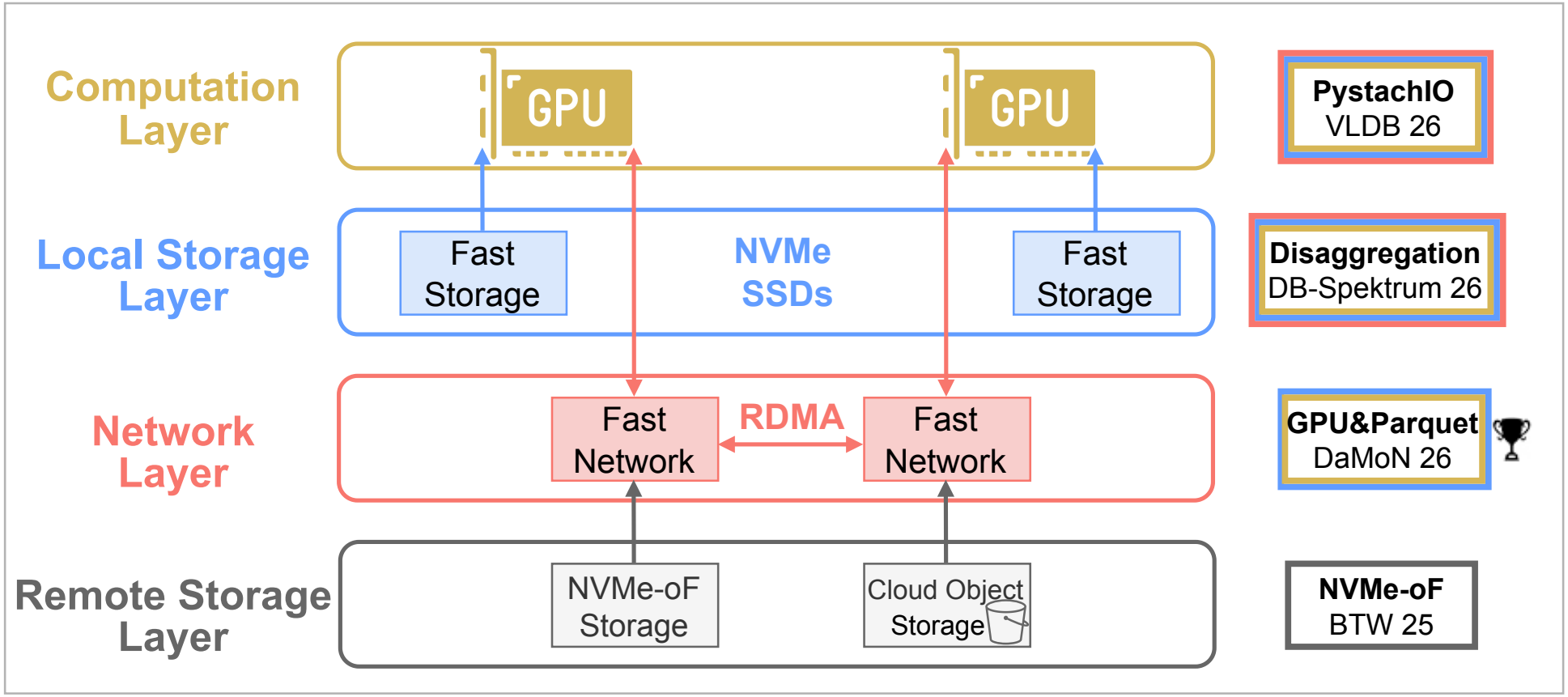
It began 5+ years ago as a research prototype we called the Tensor Query Processor (TQP). The idea was simple: express relational operators on top of the tensor API of PyTorch, so data systems could ride the entire AI wave (GPUs and beyond) for free.

Source: Carlo Curino, “Tensor Query Processing: How Do We Ride the AI Investment Wave for Database Analytics?” (SEBD 2023)

My First Meeting with Carsten: A Roadmap



My Research Status



1. GPU & Parquet, DaMoN 2026

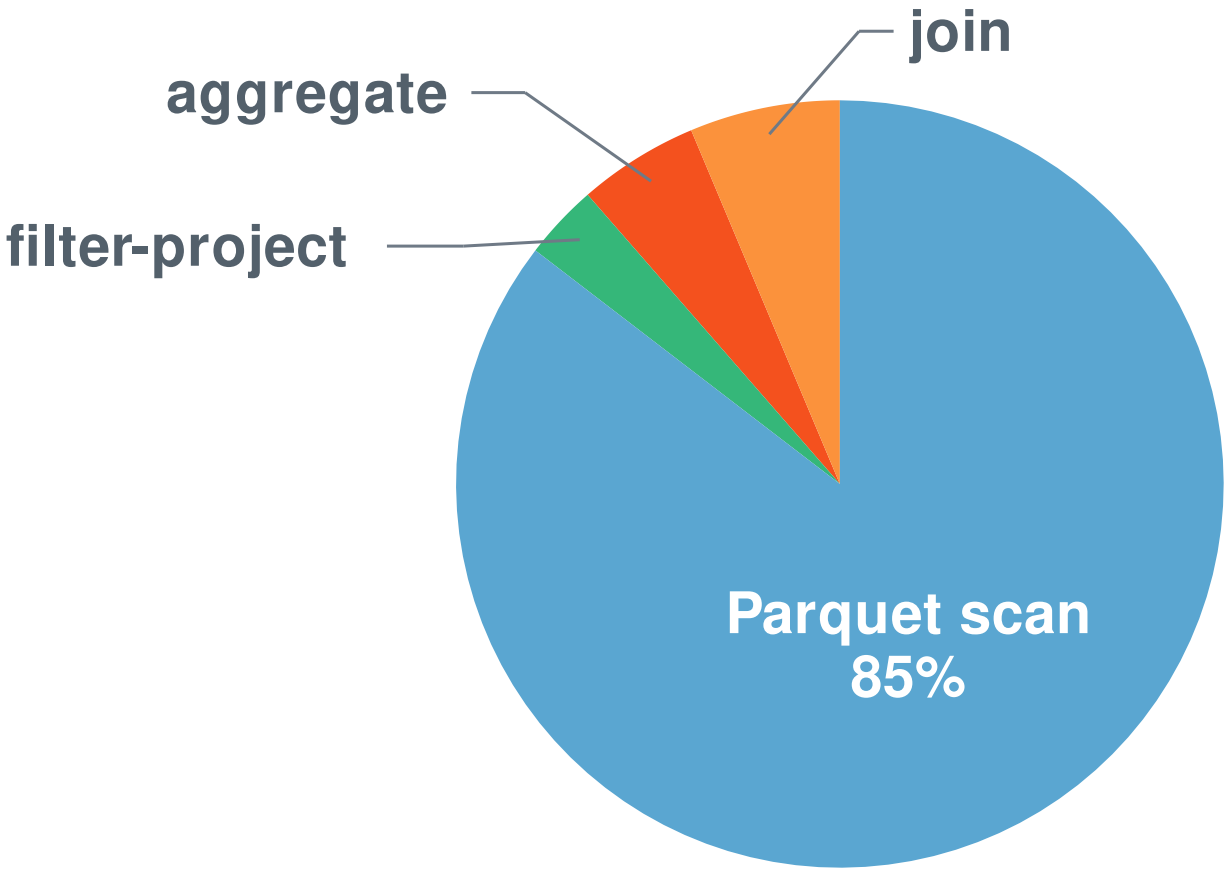
Problem: GPU Bottleneck With Parquet

Problem: GPU Bottleneck With Parquet

Is Parquet Bad for GPUs?

Problem: GPU Bottleneck With Parquet

Is Parquet Bad for GPUs?



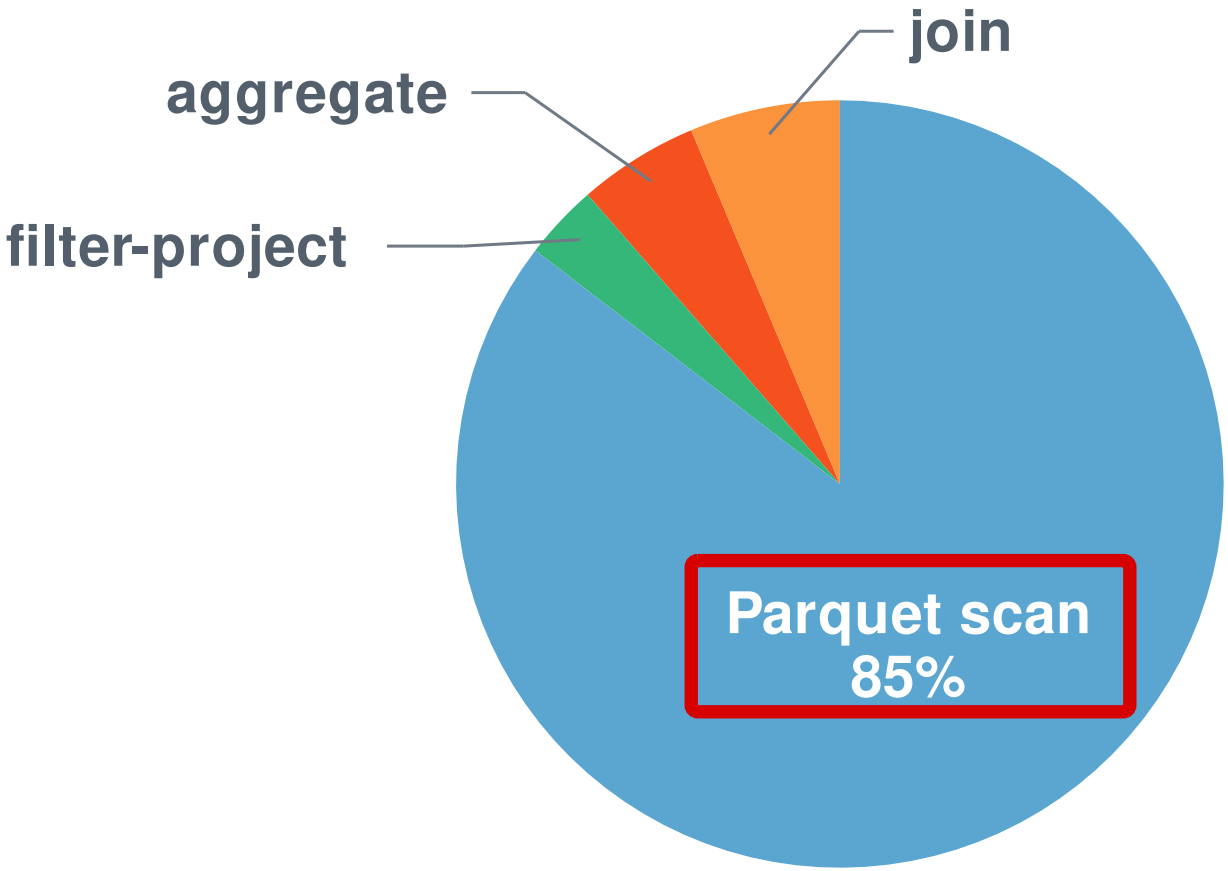
GPU TPC-H SF100 runtime breakdown.

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.

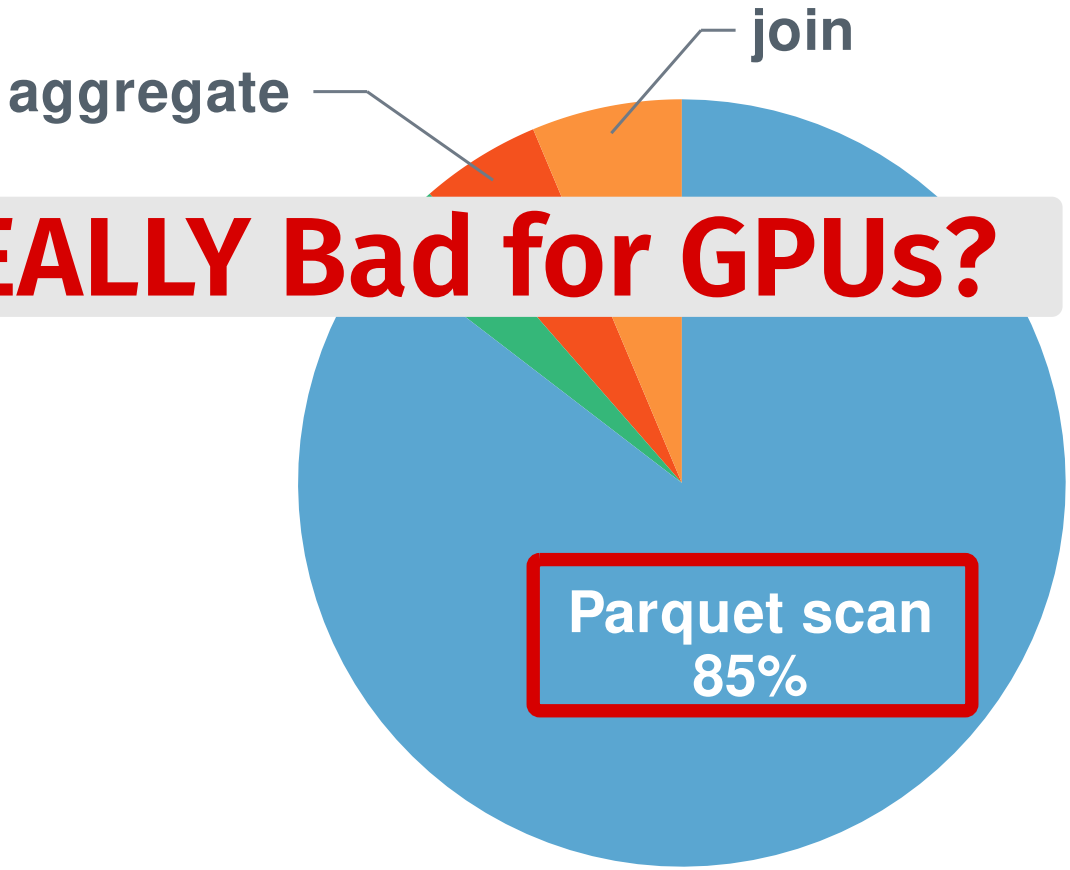
Source: Accelerating Velox with RAPIDS cuDF, Velox-cuDF Team, 2025

Problem: GPU Bottleneck With Parquet

Is Parquet REALLY Bad for GPUs?

Is Parquet Bad for GPUs?

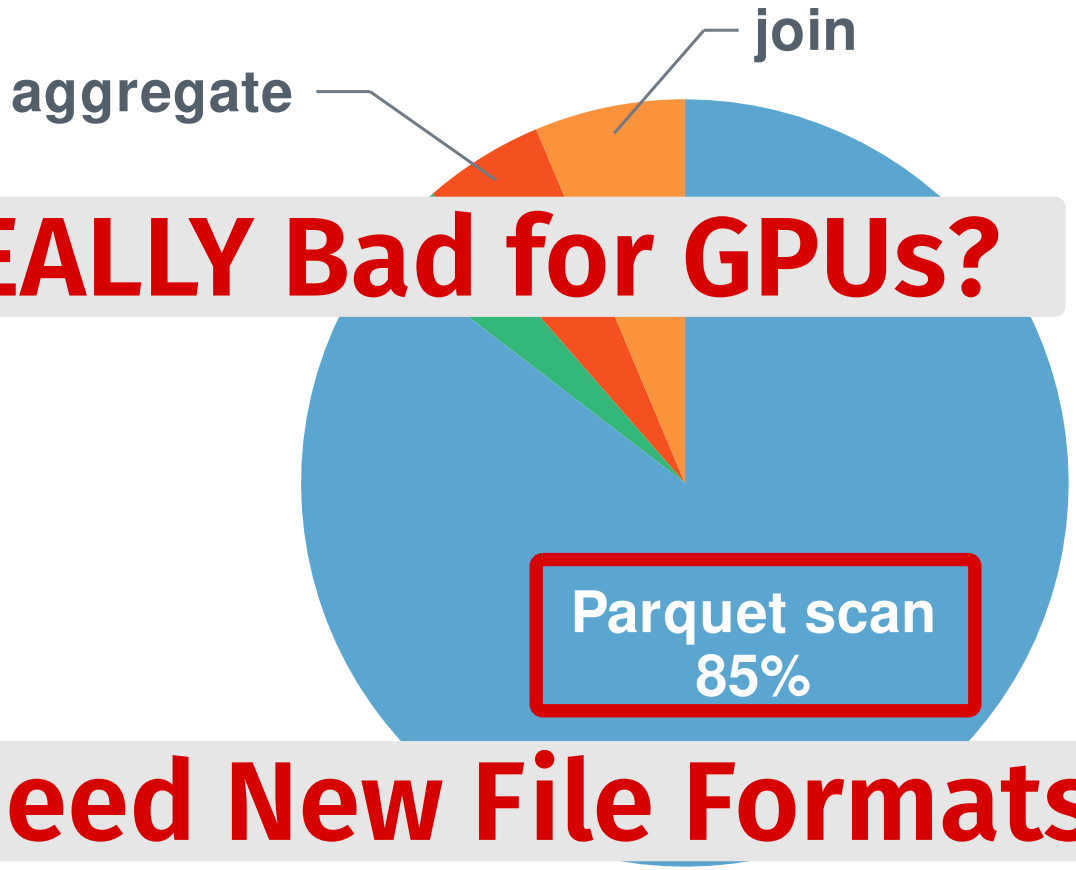
- **85%** in Parquet Scan



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

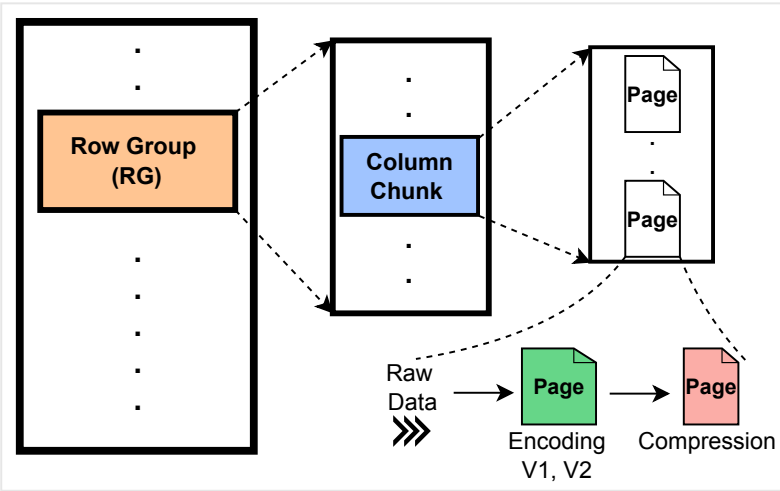
Do GPUs REALLY Need New File Formats?

GPU TPC-H SF100 runtime breakdown.

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

Parquet Configuration Defaults for CPUs

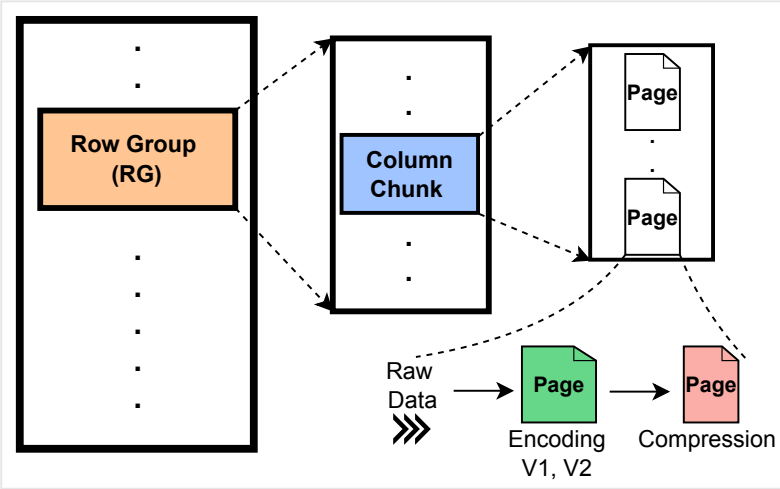
CPU Parquet Config. Defaults in DuckDB:



Parquet Configuration Defaults for CPUs

CPU Parquet Config. Defaults in DuckDB:

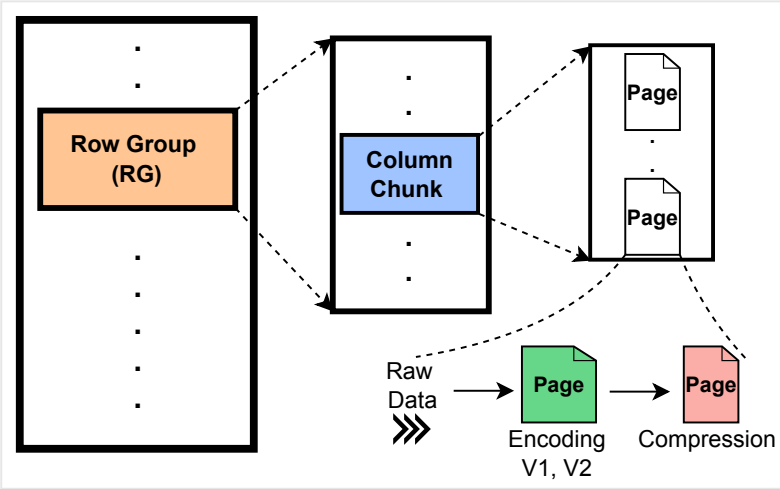
- Row group: 122880 rows



Parquet Configuration Defaults for CPUs

CPU Parquet Config. Defaults in DuckDB:

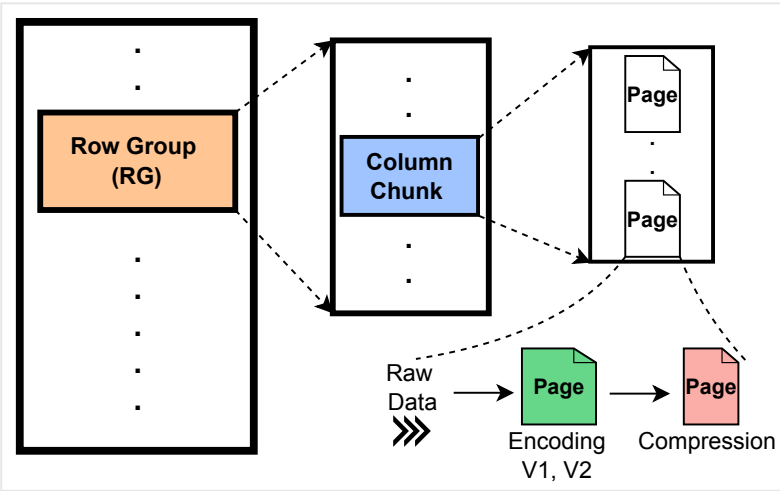
- Row group: 122880 rows
- Column chunk: 1 page only



Parquet Configuration Defaults for CPUs

CPU Parquet Config. Defaults in DuckDB:

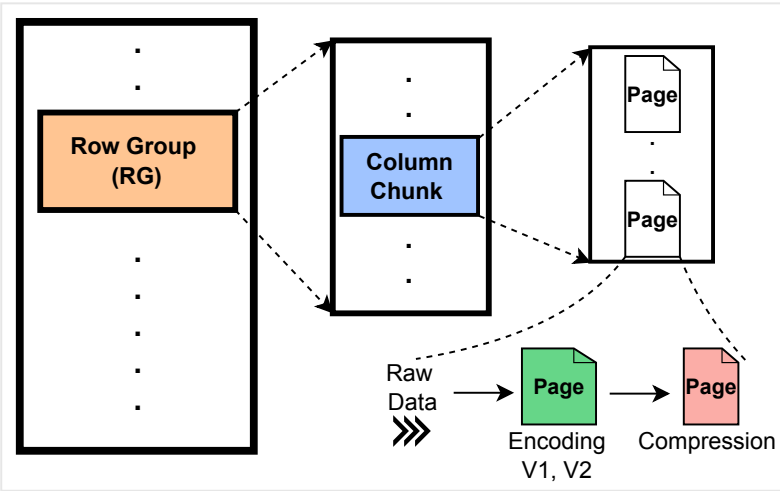
- Row group: 122880 rows
- Column chunk: 1 page only
- Encoding: V1 only



Parquet Configuration Defaults for CPUs

CPU Parquet Config. Defaults in DuckDB:

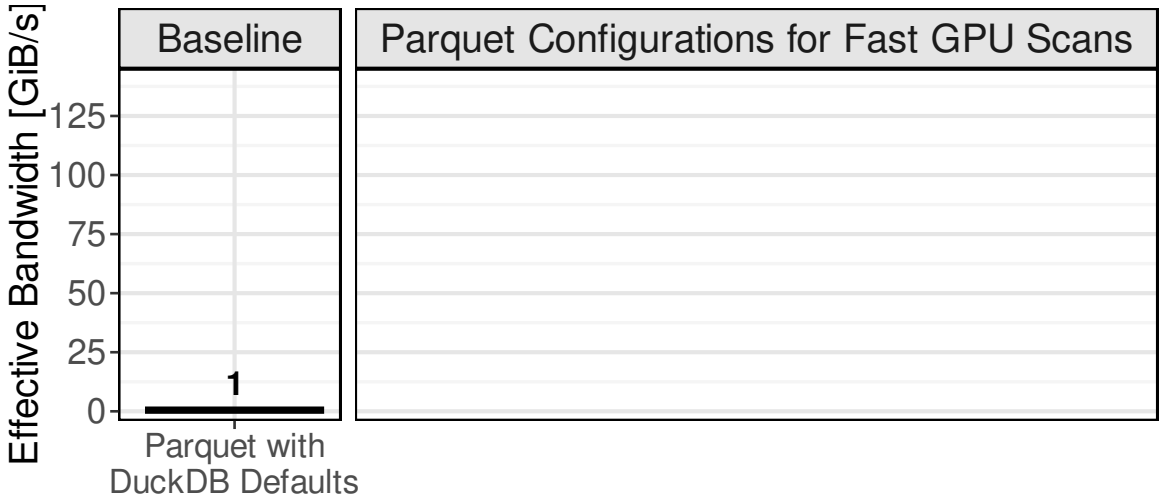
- Row group: 122880 rows
- Column chunk: 1 page only
- Encoding: V1 only
- Compression: used even without size reduction



Issue: CPU Defaults Config. ≠ GPU Optimal Config.

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

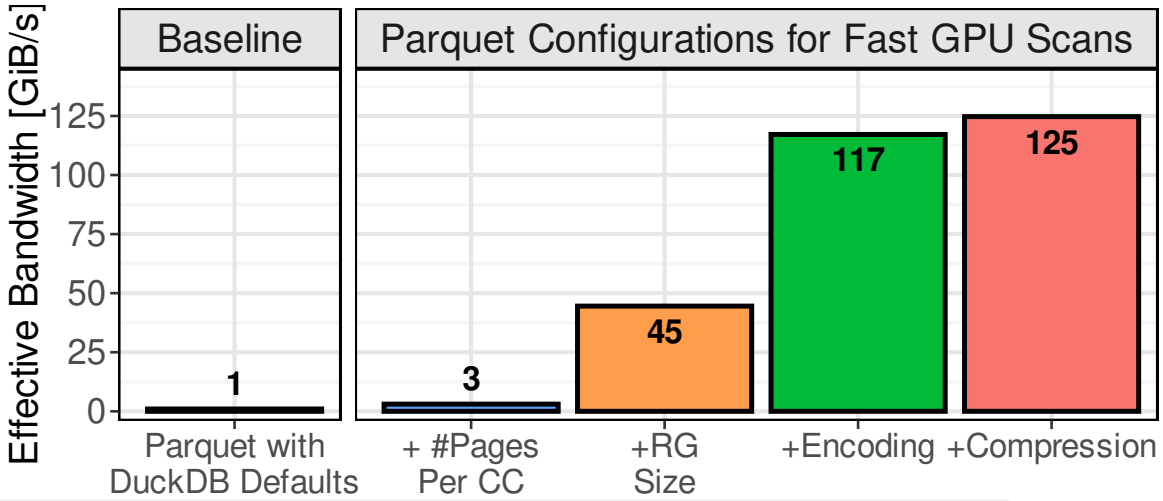
- Row group: 122880 rows
- Column chunk: 1 page only
- Encoding: V1 only
- Compression: used even without size reduction



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

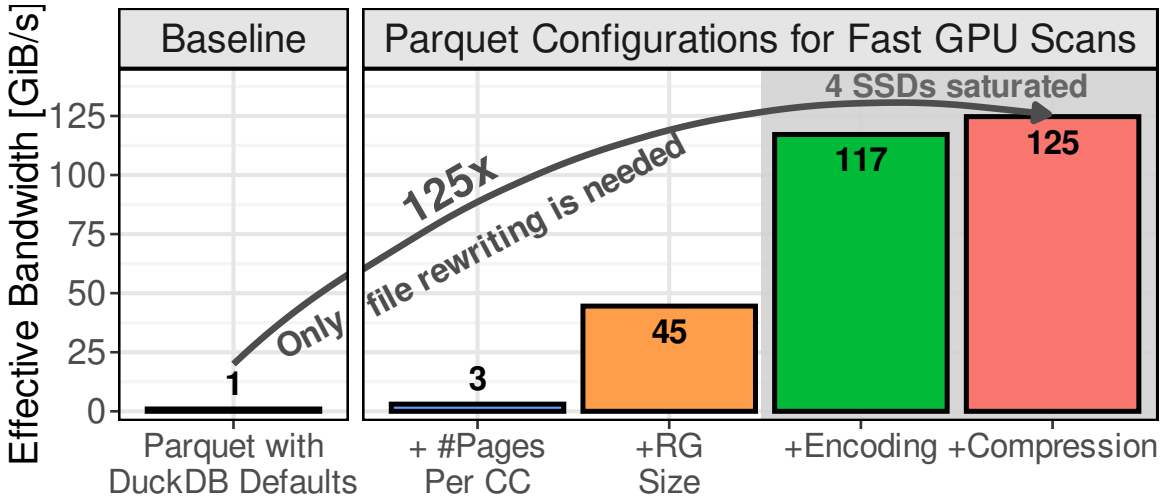


CPU Defaults Config. ≠ GPU Optimal Config.

Issue: CPU Defaults Config. ≠ GPU Optimal Config.

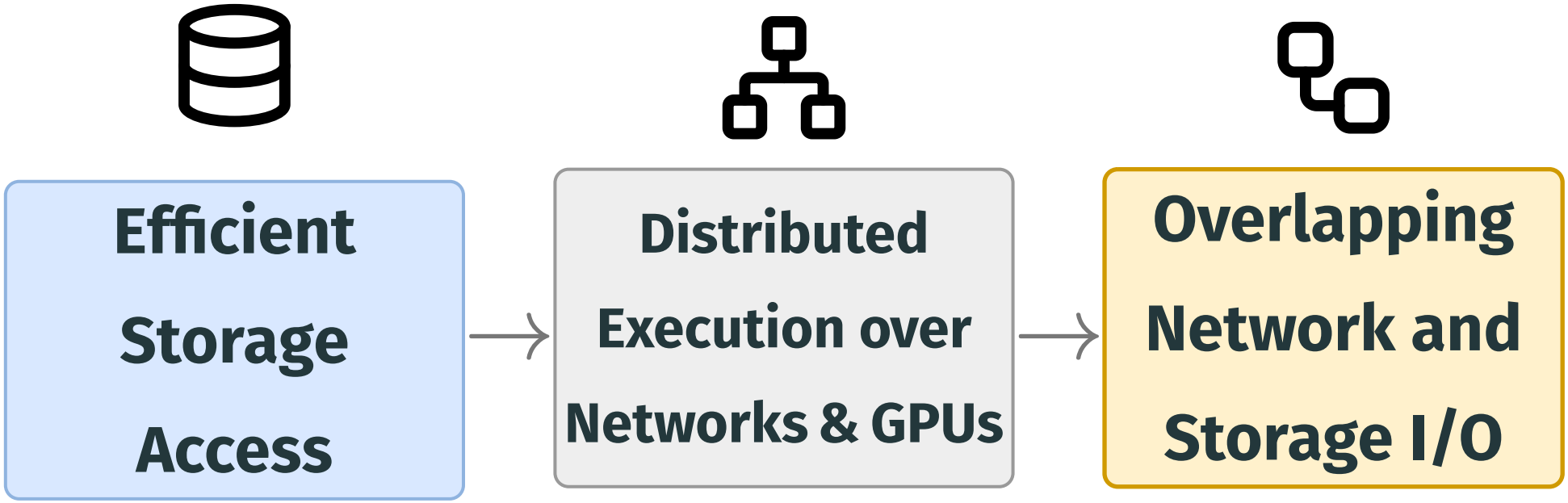
CPU Parquet Config. Defaults in **DuckDB**:

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



2. PystachIO, VLDB 2026

PystachIO Overview



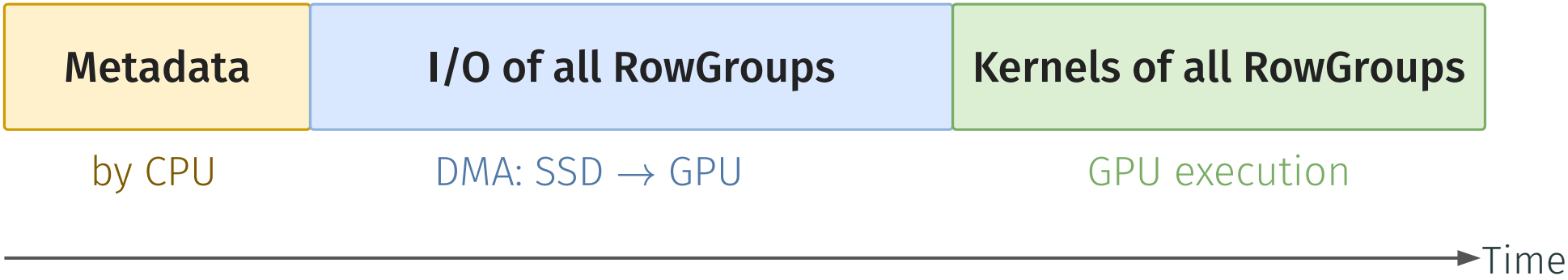
Related work: Do GPUs Really Need New Tabular File Formats?, DaMoN 2026, Best Short Paper Award

PystachIO Overview

Related work: Do GPUs Really Need New Tabular File Formats?, DaMoN 2026, Best Short Paper Award

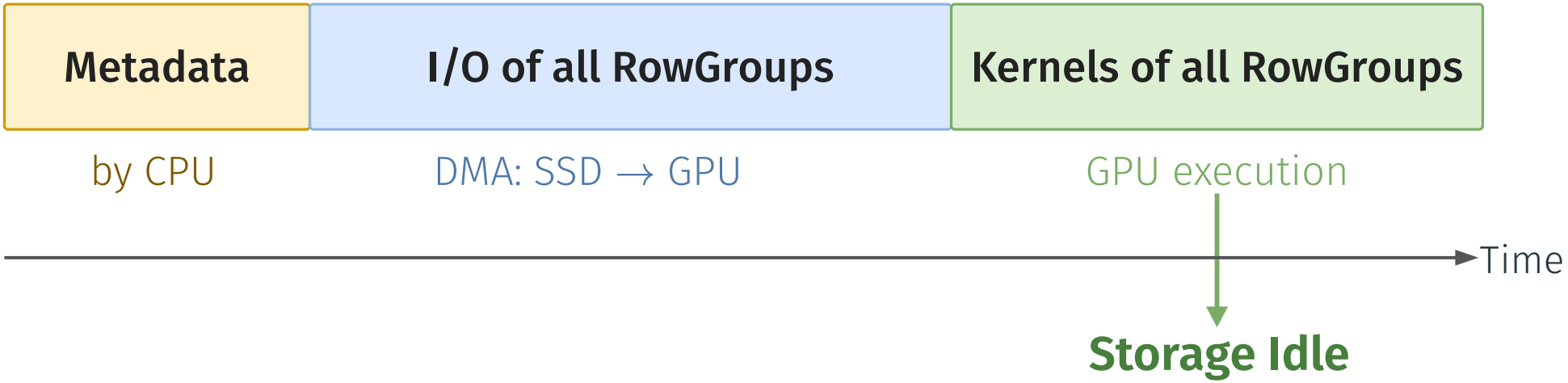
~~Efficient~~ Storage Access: Blocking Parquet Reader

`read_parquet()` in one CPU thread:



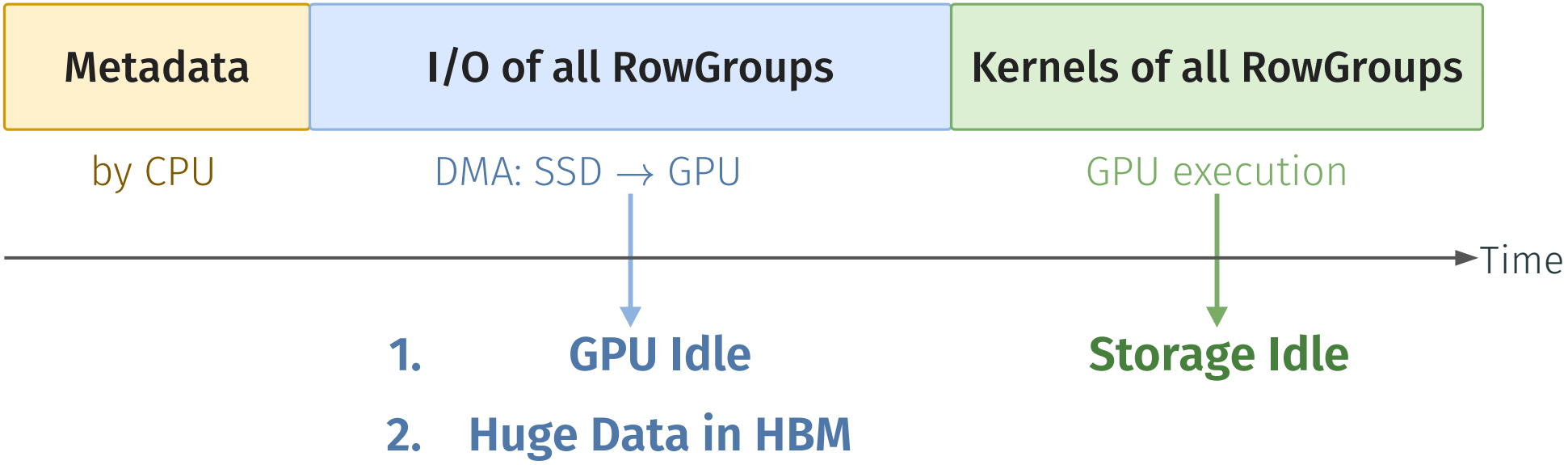
~~Efficient~~ Storage Access: Blocking Parquet Reader

`read_parquet()` in one CPU thread:



~~Efficient~~ Storage Access: Blocking Parquet Reader

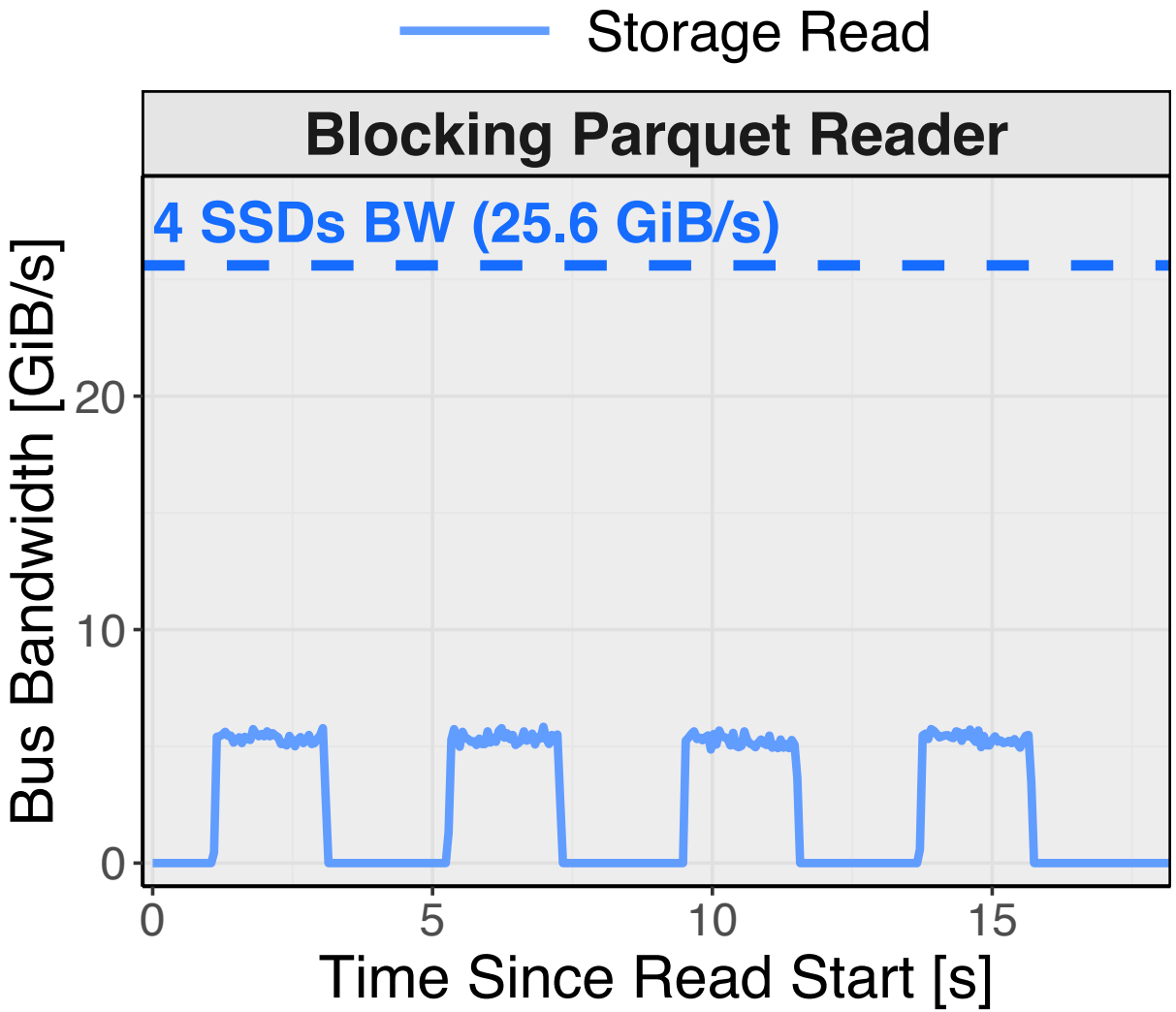
`read_parquet()` in one CPU thread:



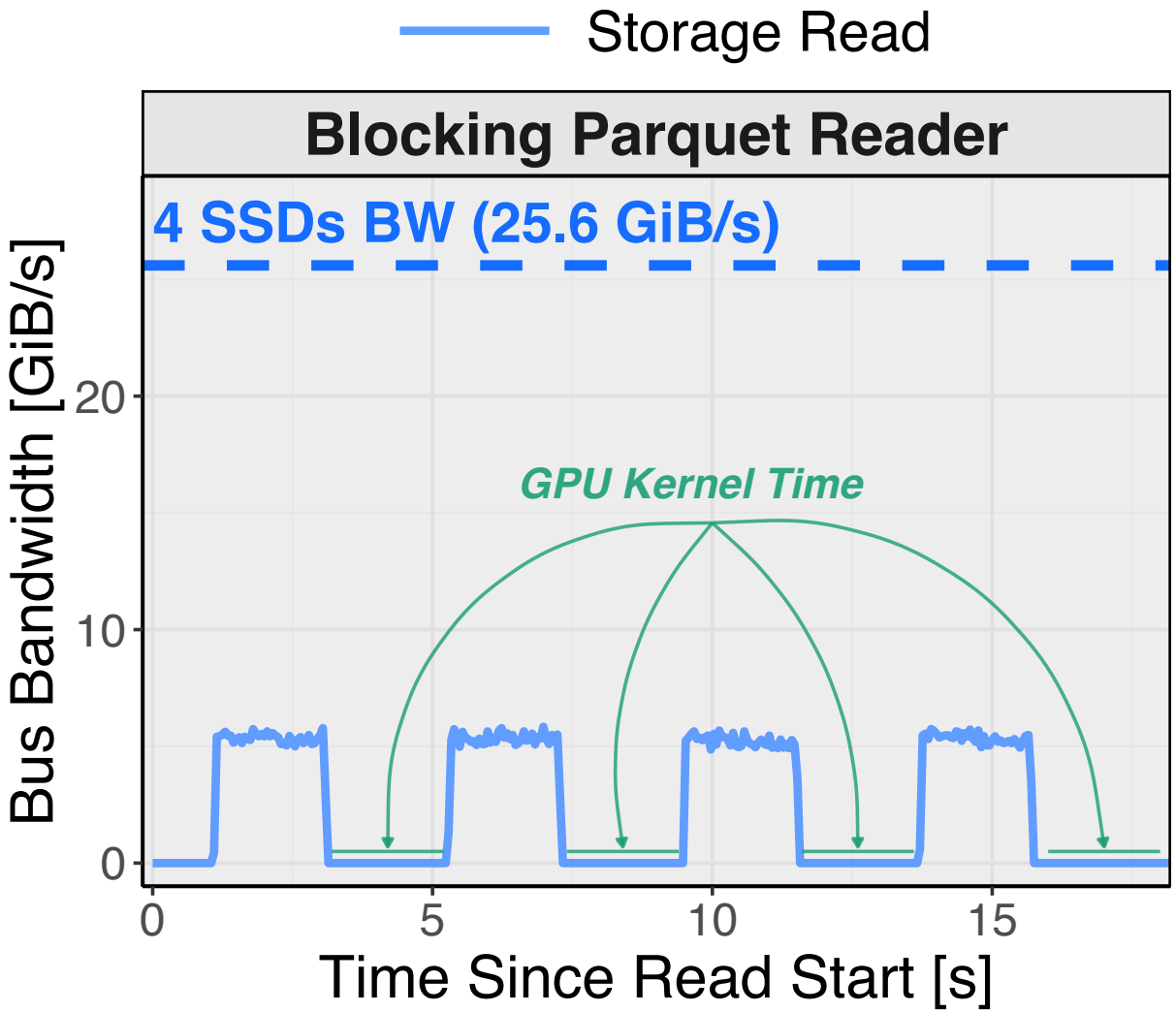
1. **GPU Idle**
2. **Huge Data in HBM**

Storage Idle

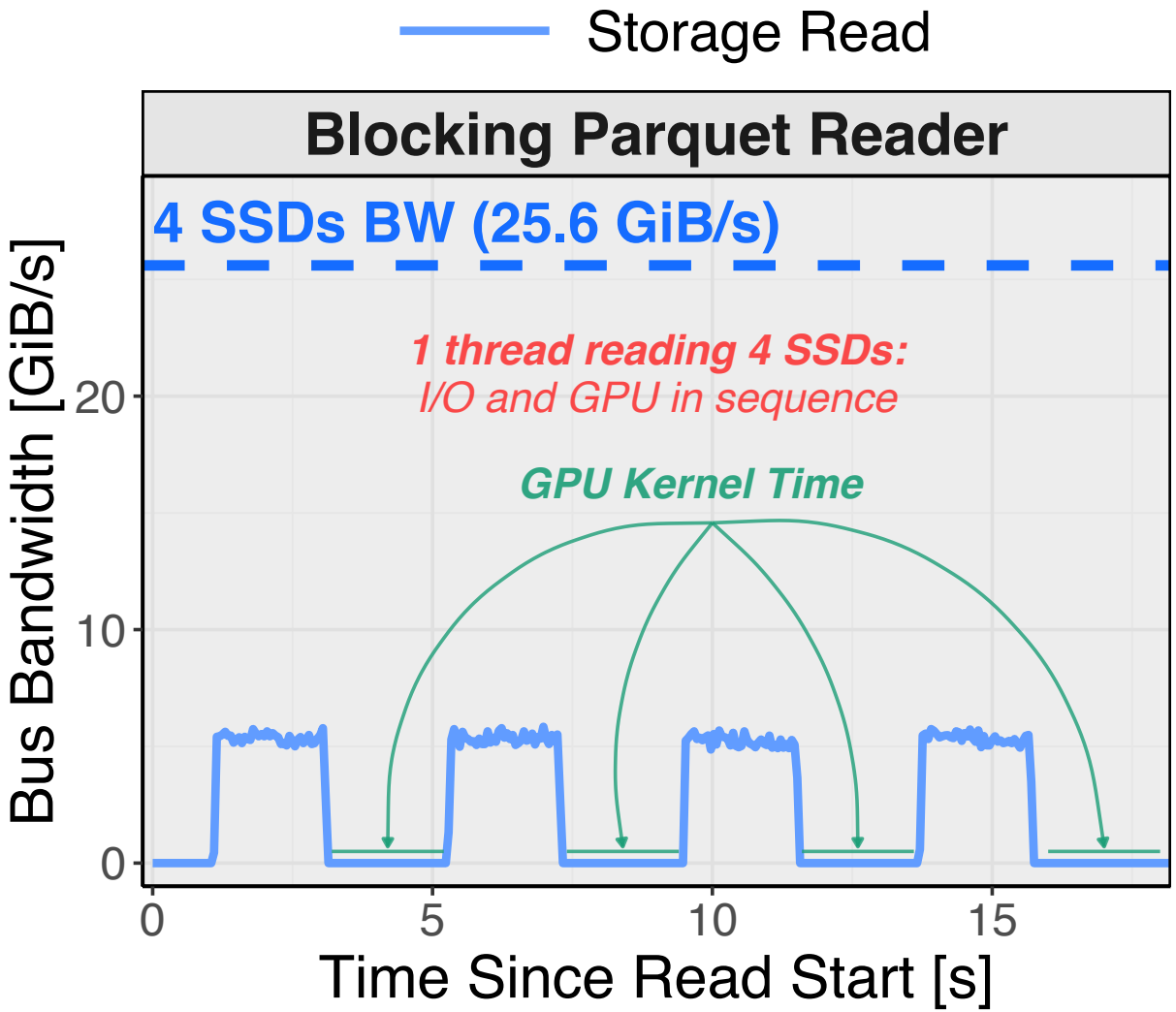
~~Efficient~~ Storage Access: Blocking Parquet Reader



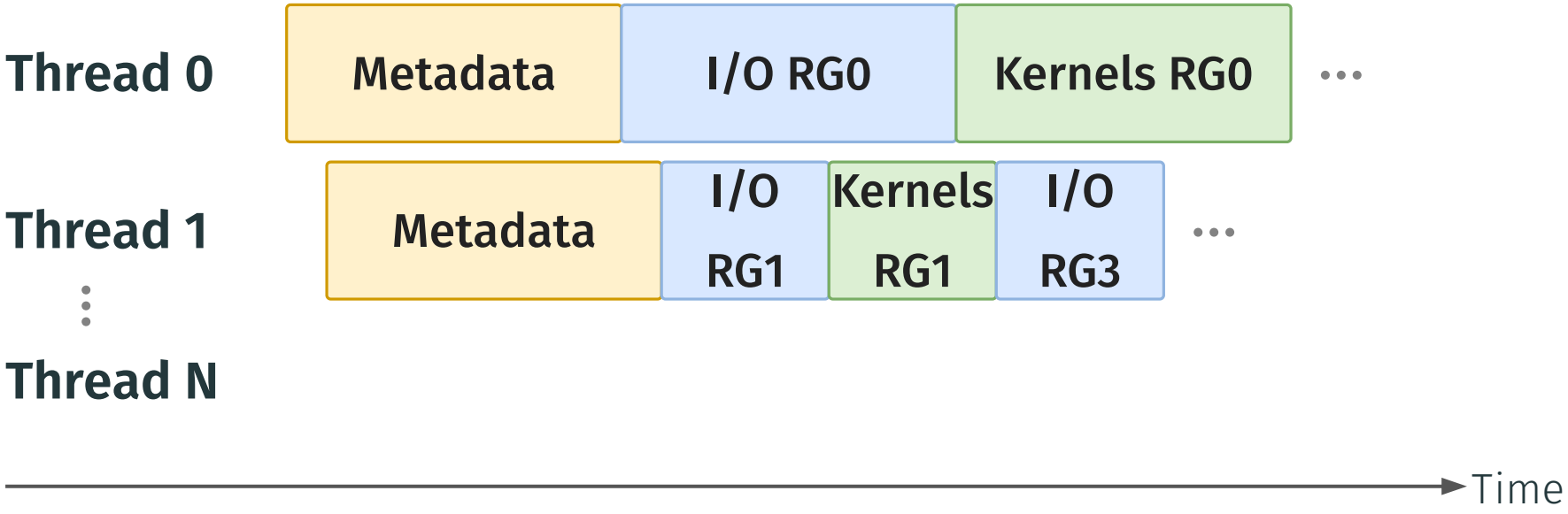
~~Efficient~~ Storage Access: Blocking Parquet Reader



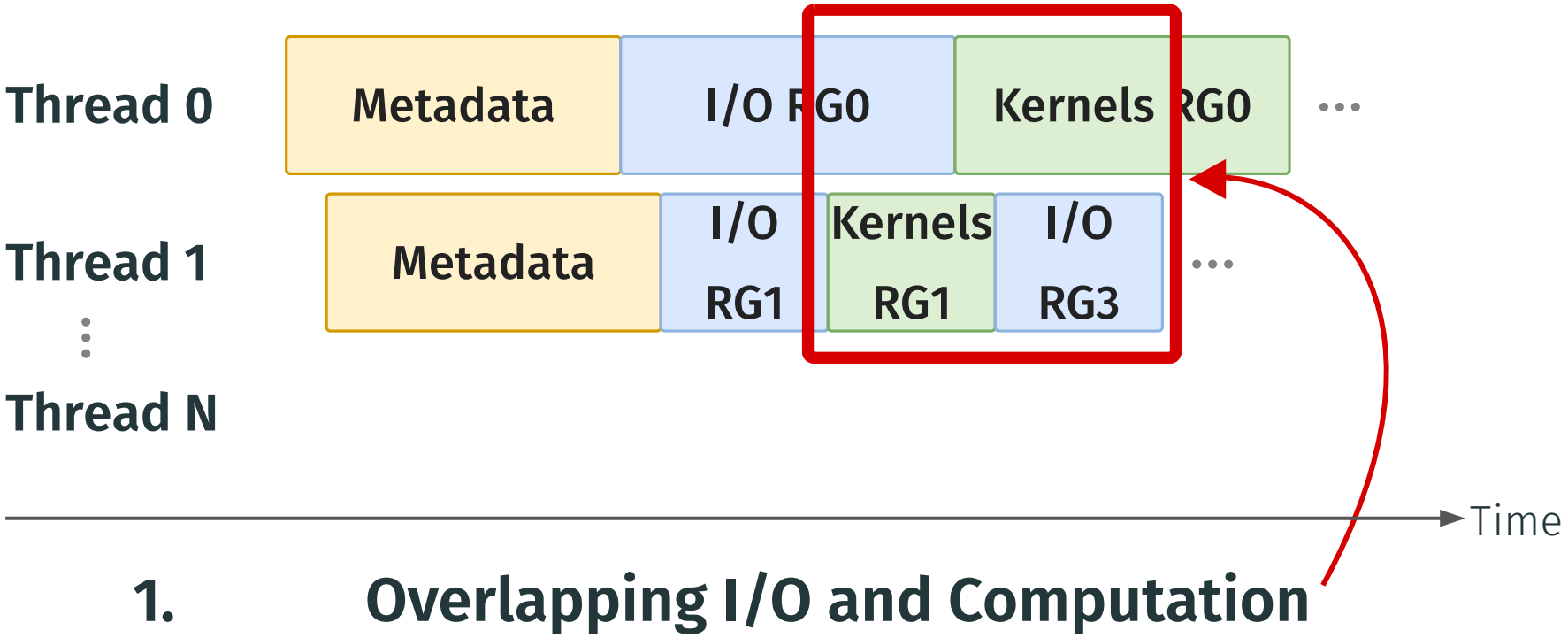
~~Efficient~~ Storage Access: Blocking Parquet Reader



Efficient Storage Access: Chunking

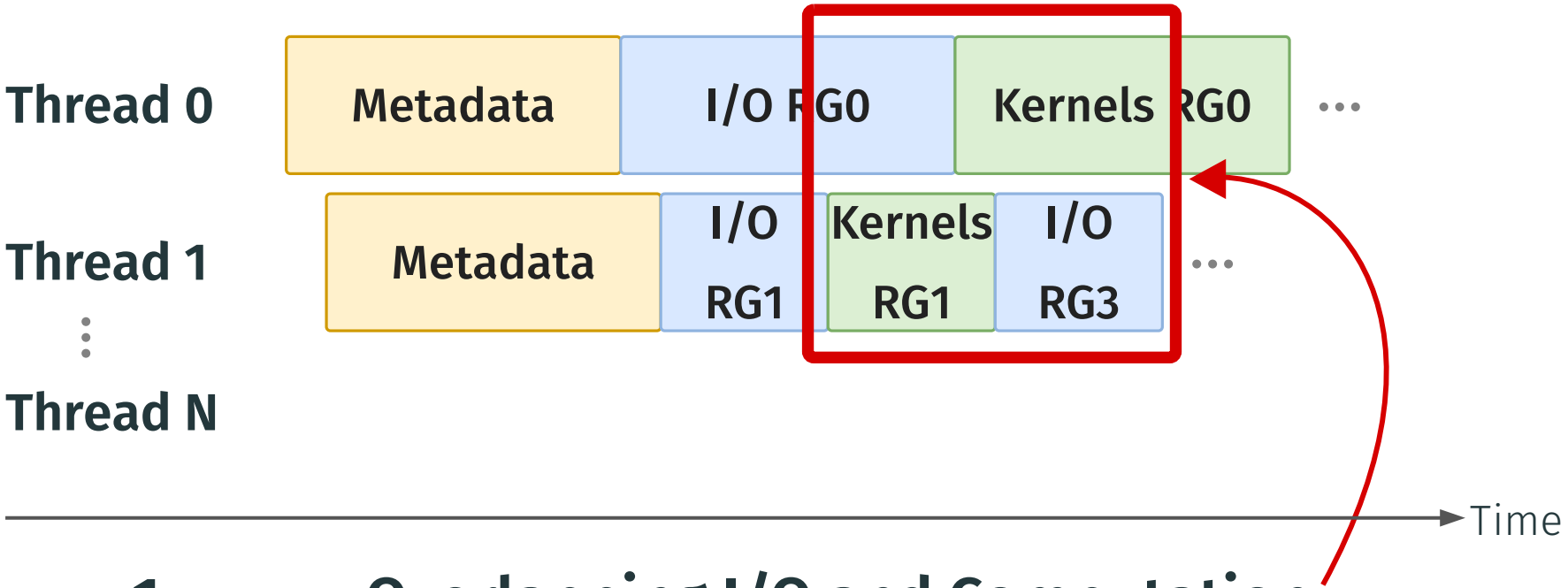


Efficient Storage Access: Chunking



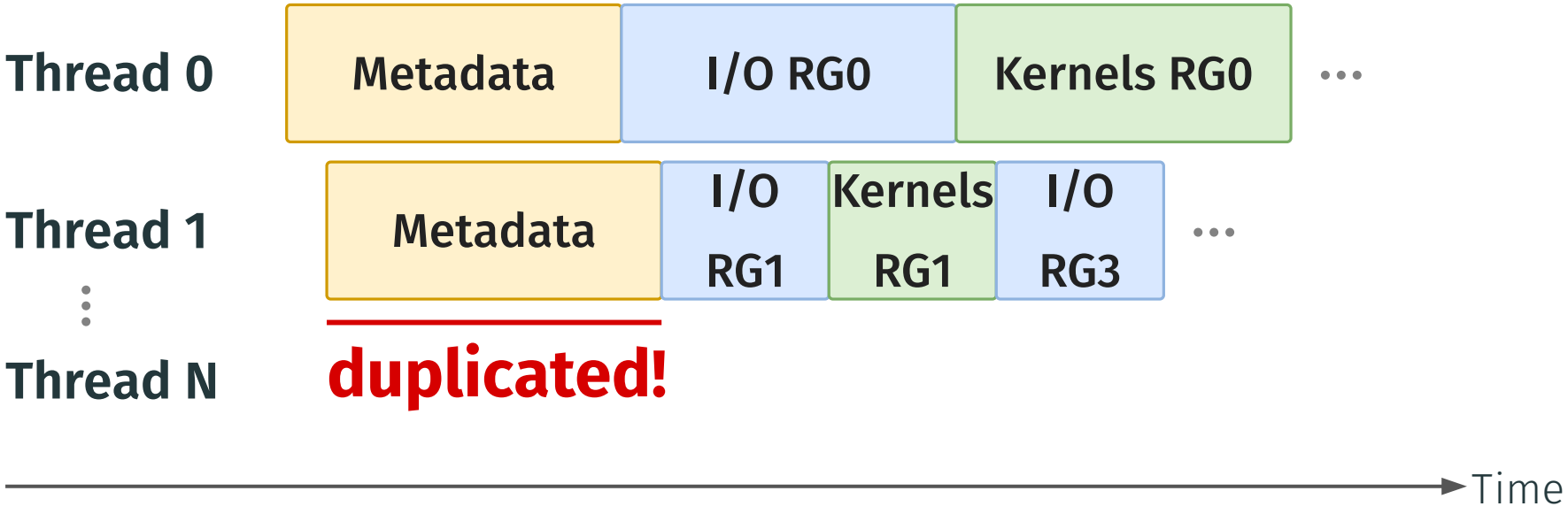
1. Overlapping I/O and Computation

Efficient Storage Access: Chunking

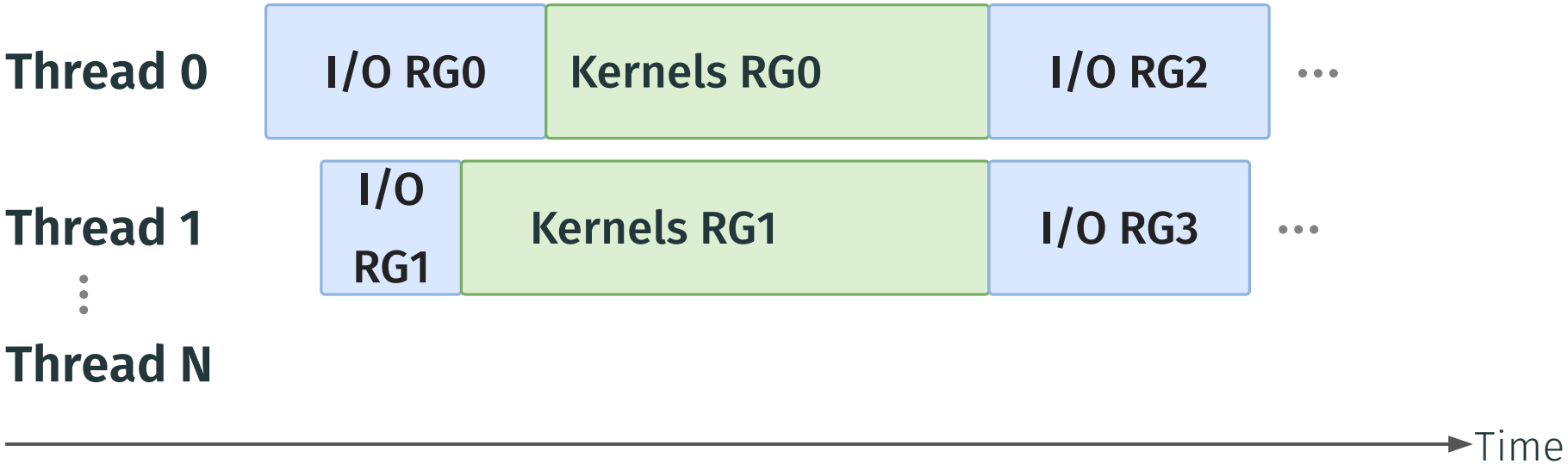


1. Overlapping I/O and Computation
2. Less Memory Pressure

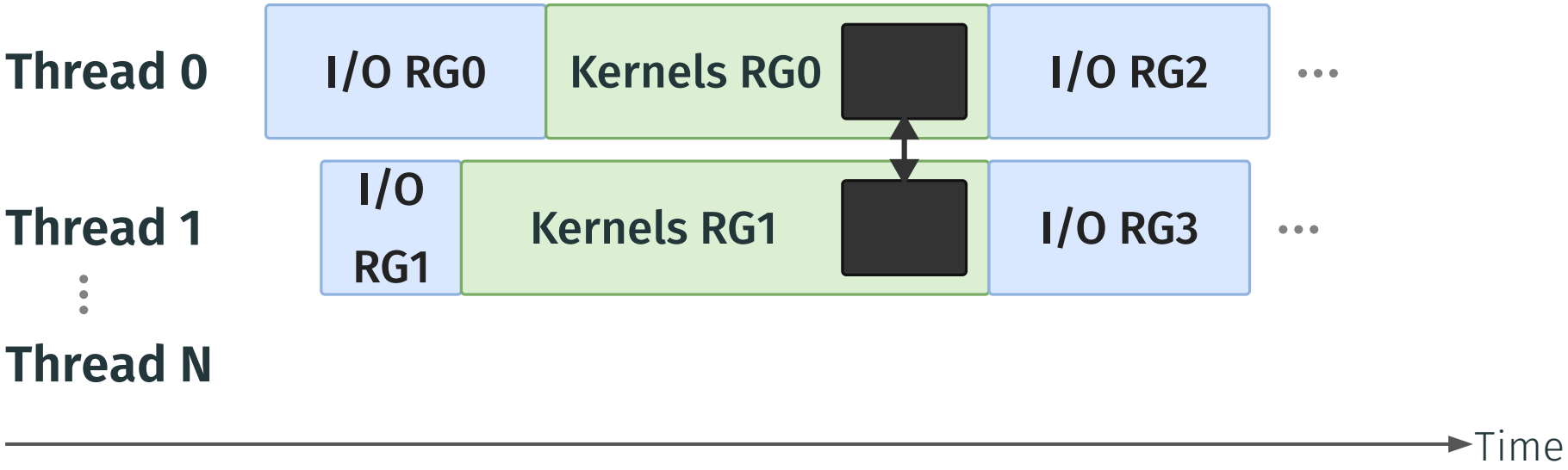
Efficient Storage Access: Chunking



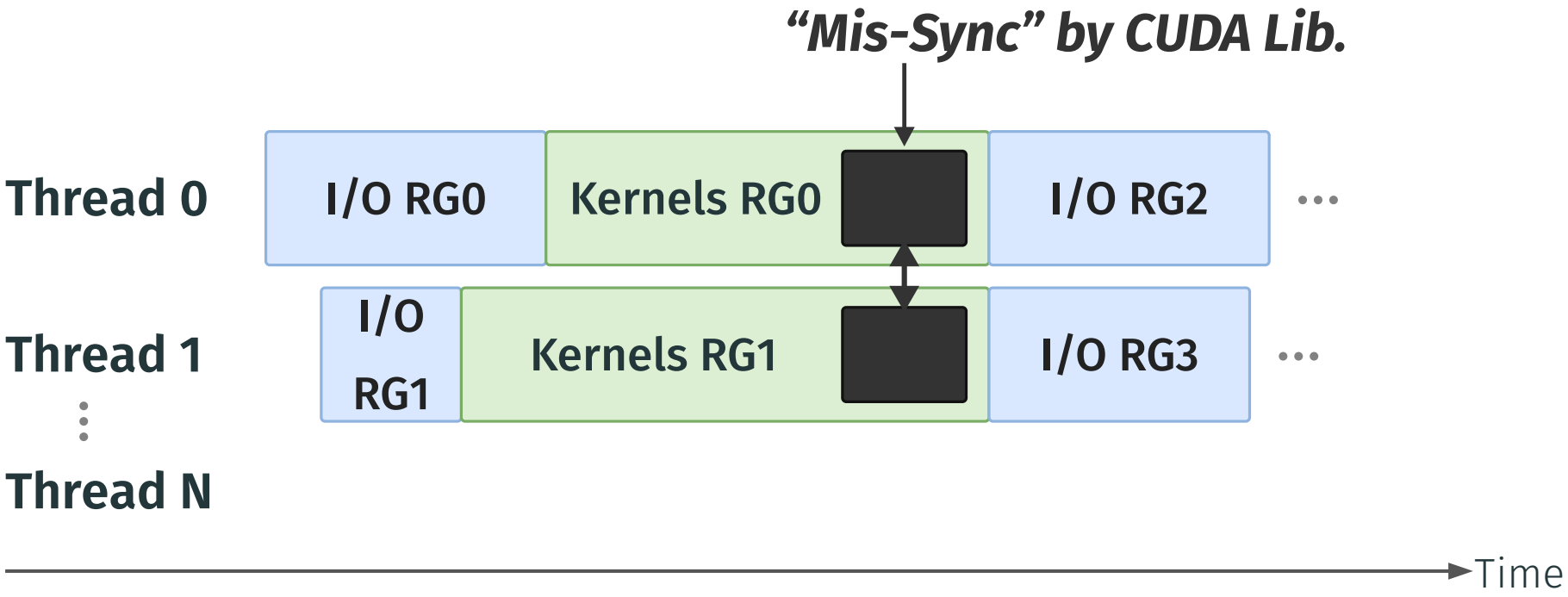
Efficient Storage Access: Metadata Caching



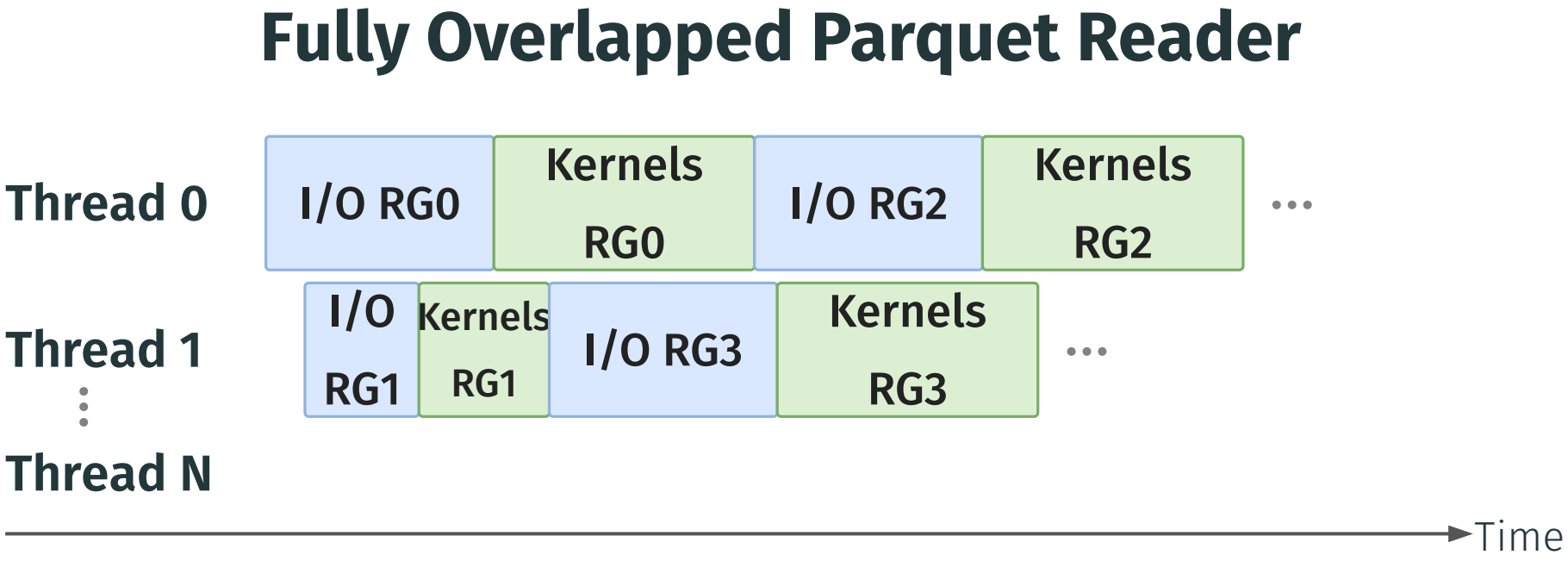
Efficient Storage Access: Metadata Caching



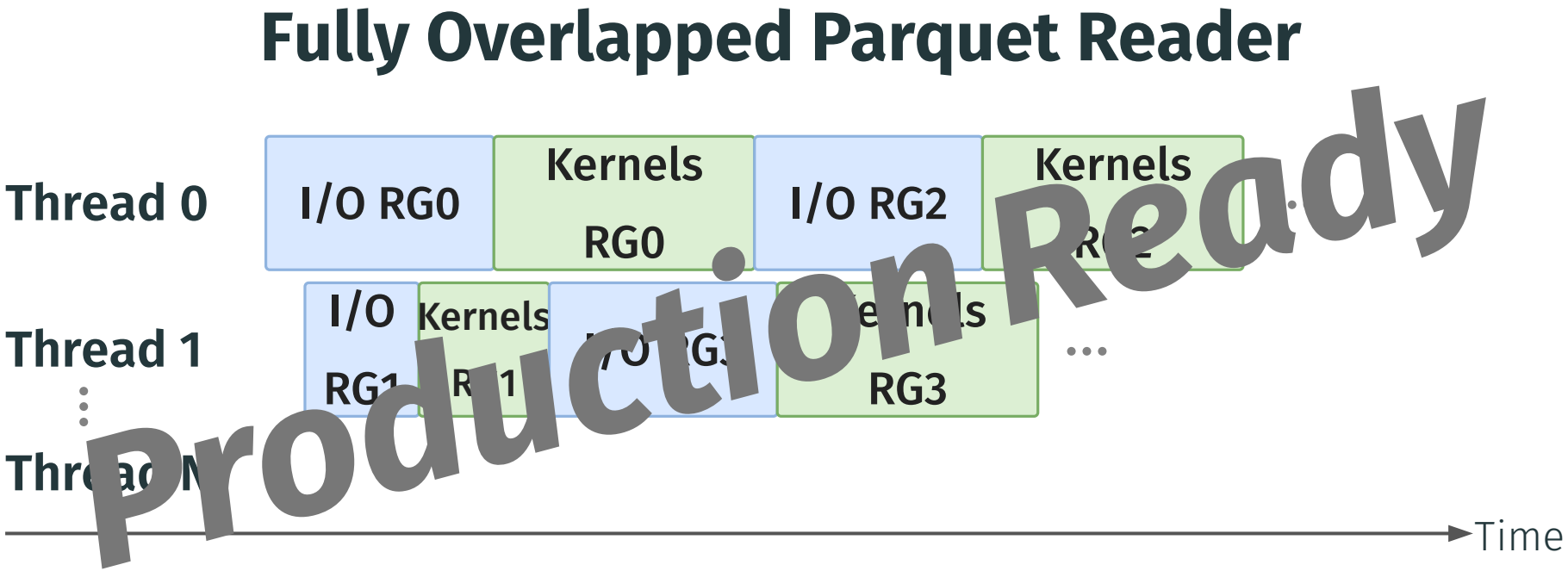
Efficient Storage Access: Metadata Caching



Efficient Storage Access: Mis-Sync Elimination

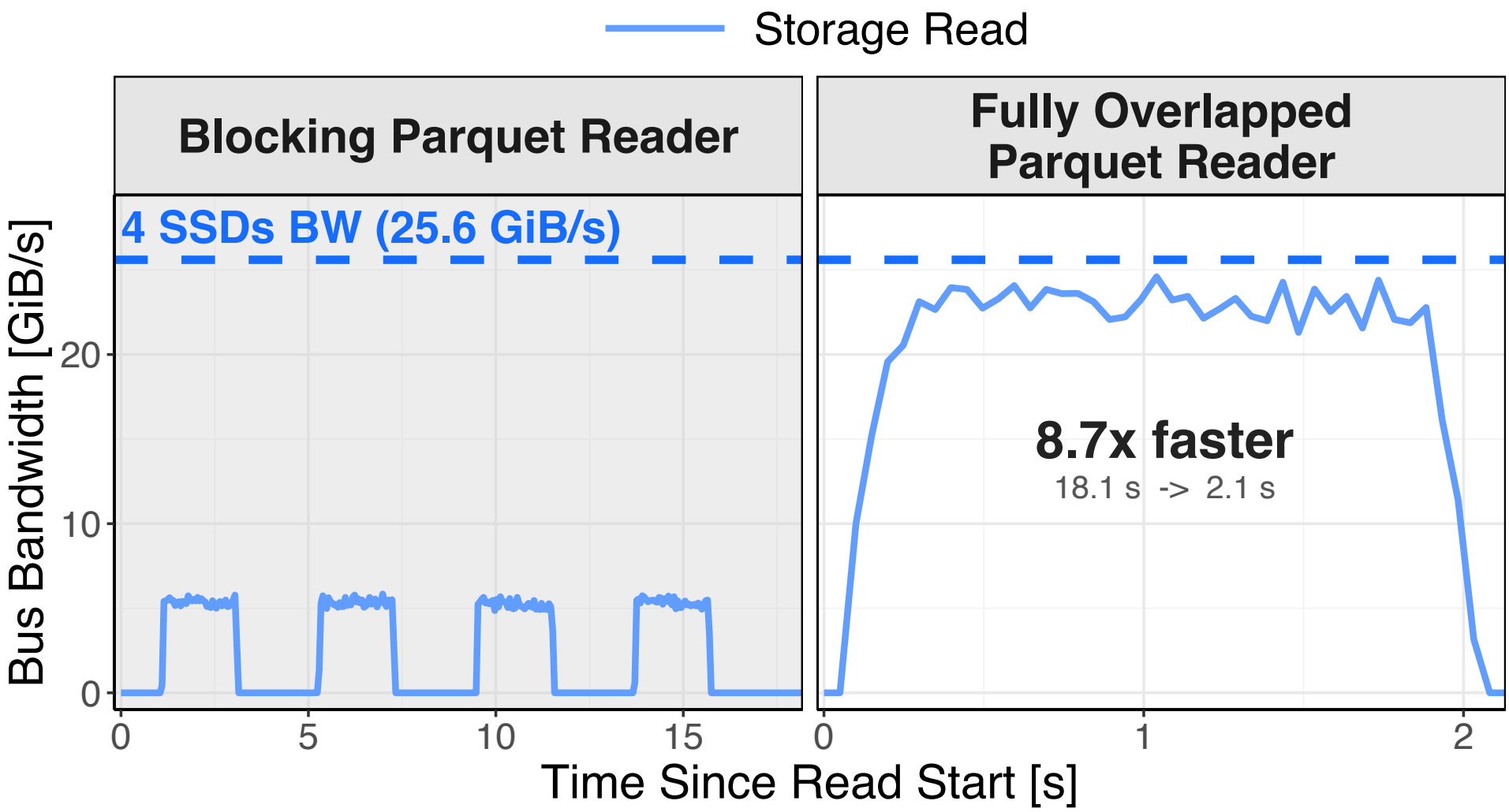


Efficient Storage Access: Mis-Sync Elimination



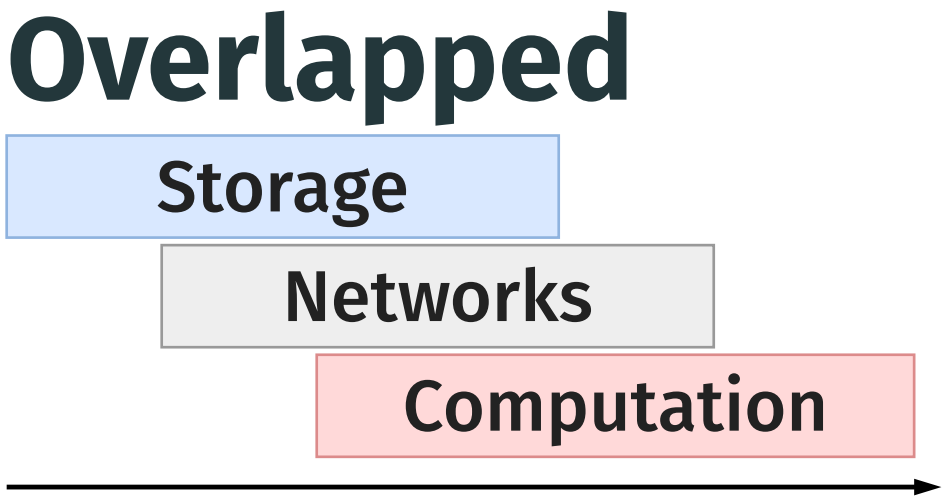
cuDF issue #18892: [Story] Towards a faster Parquet reader with pipelining and multistream optimization, Jigao Luo.
cuDF issue #21272: Migrate libcudf from Thrust to CUB-based algorithms, Bradley Dice.

Efficient Storage Access: Mis-Sync Elimination



Overlapping: Storage, Networks, and Computation

- Storage engine reading Parquet
- Network for distributed join
- **PystachIO**: Storage + Network



PystachIO: Fast Storage and Fast Network

- Storage optimized
- Network optimized
- Next?

PystachIO: Fast Storage and Fast Network

- Storage optimized
- Network opti
- Next?

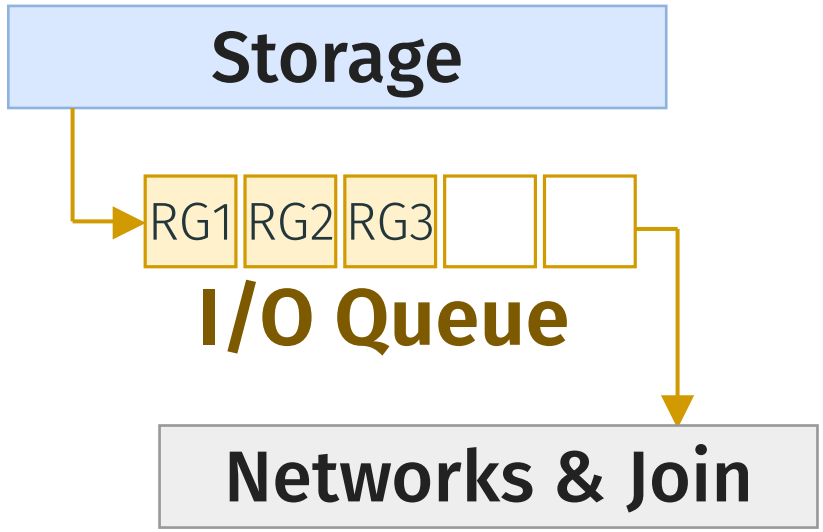
Storage

But how to overlap? **?**

Networks & Join

PystachIO: I/O Queue

- Storage: enqueuing
- Query engine: dequeuing
- Then network shuffling



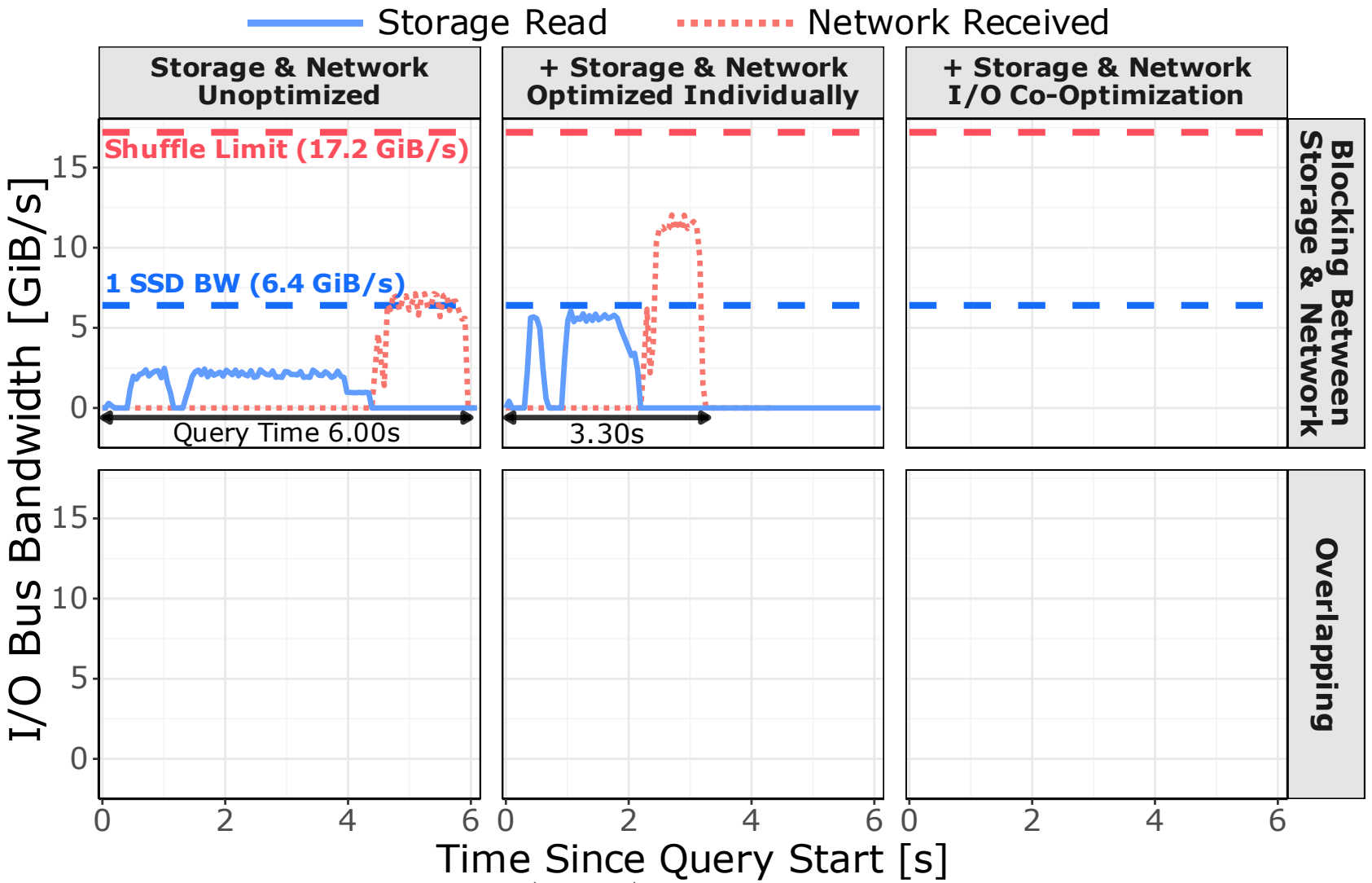
Overlapping Network, Storage, and Computation

TPC-H Query 3 (SF 500) runtime executed on 2 GPUs

Overlapping Network, Storage, and Computation

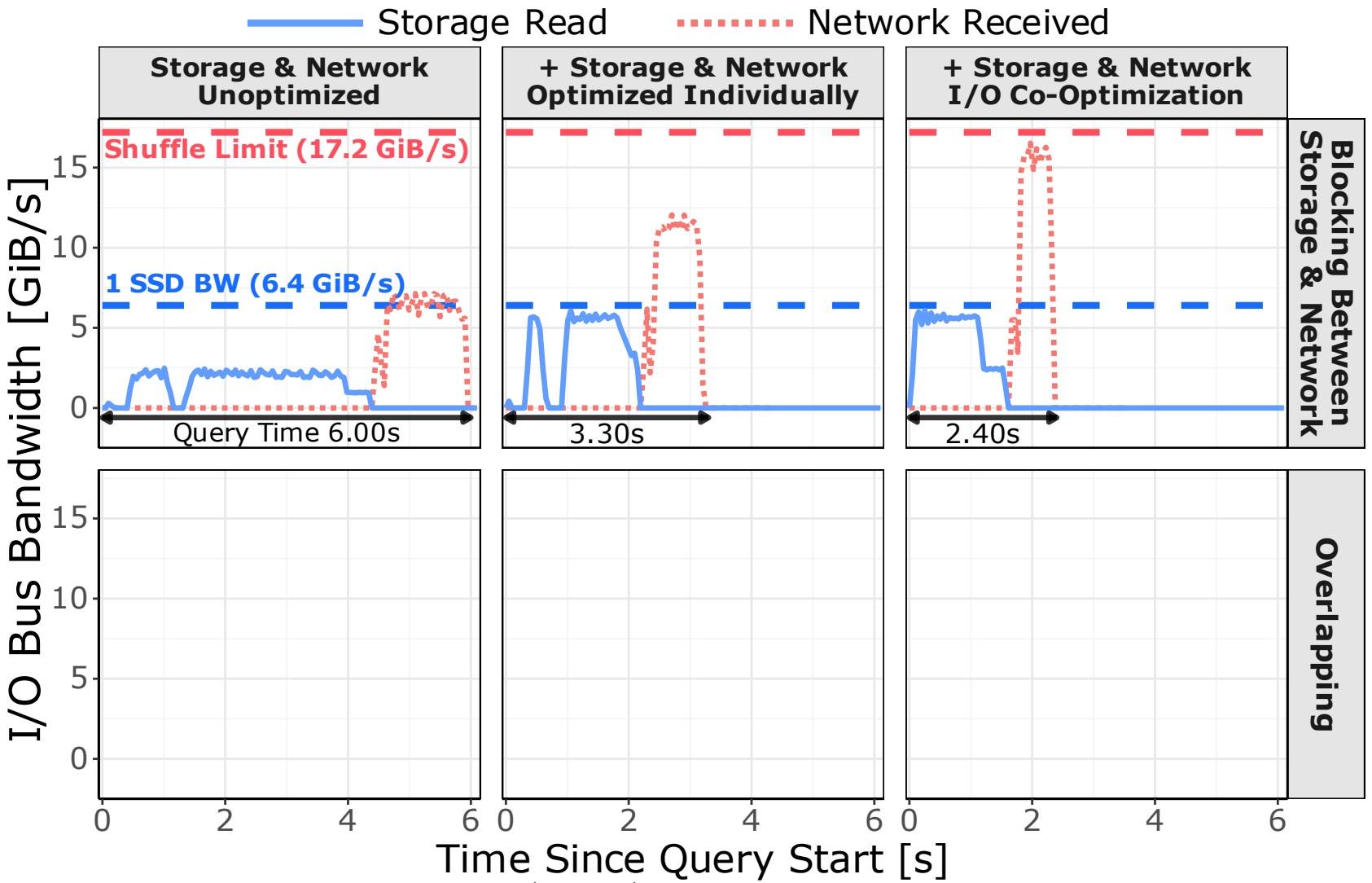
TPC-H Query 3 (SF 500) runtime executed on 2 GPUs

Overlapping Network, Storage, and Computation



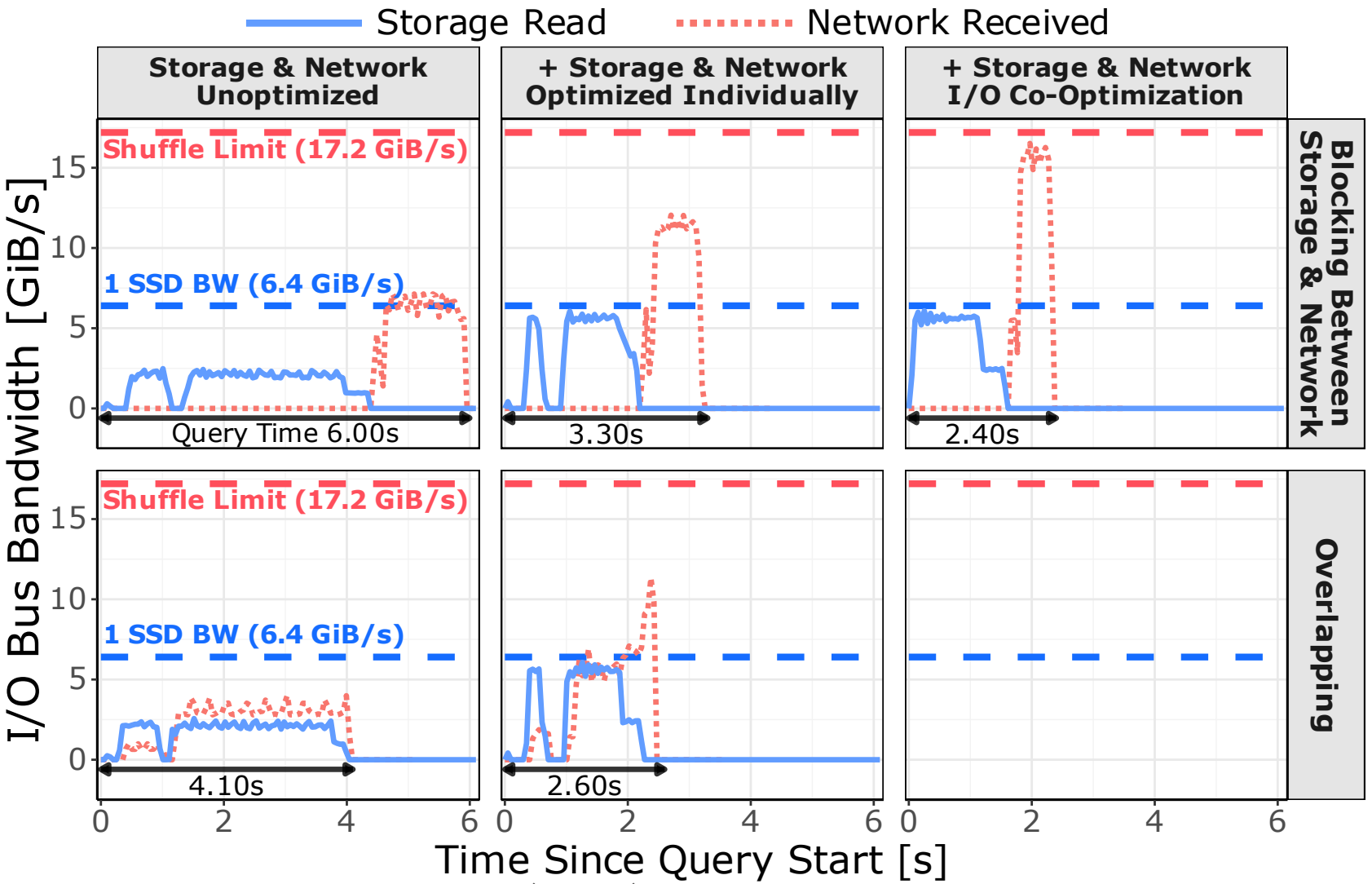
TPC-H Query 3 (SF 500) runtime executed on 2 GPUs

Overlapping Network, Storage, and Computation



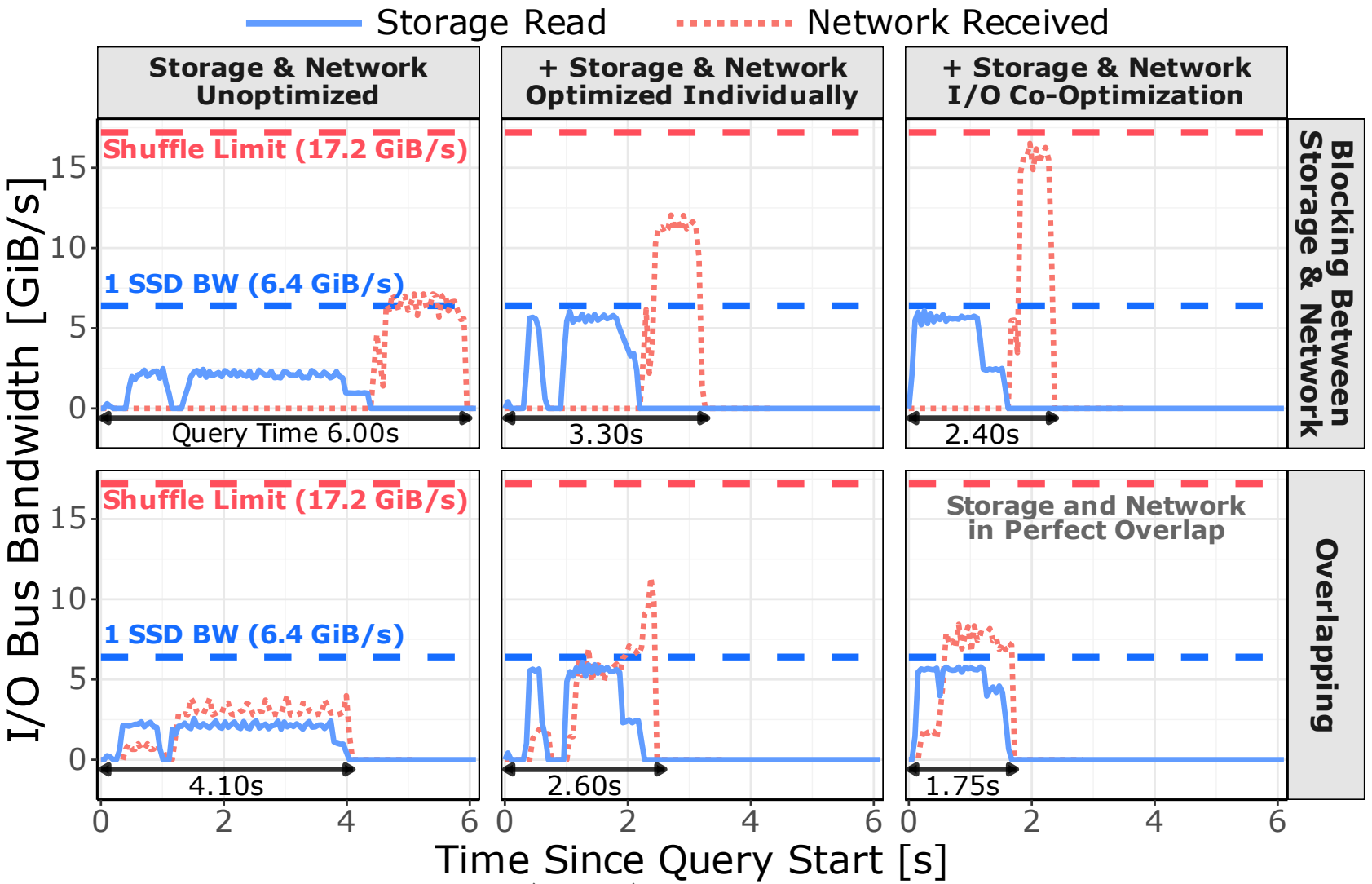
TPC-H Query 3 (SF 500) runtime executed on 2 GPUs

Overlapping Network, Storage, and Computation



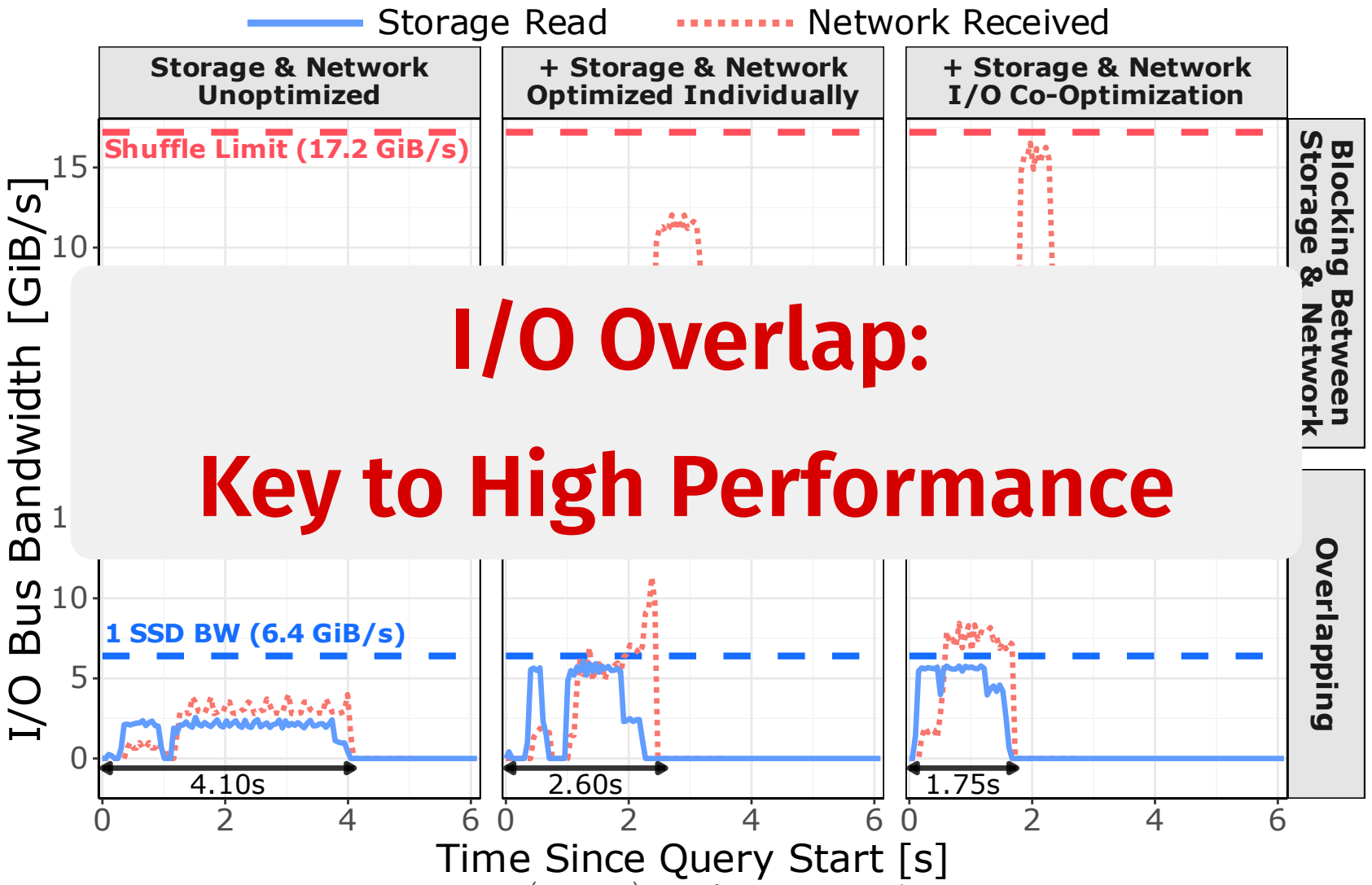
TPC-H Query 3 (SF 500) runtime executed on 2 GPUs

Overlapping Network, Storage, and Computation



TPC-H Query 3 (SF 500) runtime executed on 2 GPUs

Overlapping Network, Storage, and Computation



I/O Overlap:

Key to High Performance

TPC-H Query 3 (SF 500) runtime executed on 2 GPUs

The Road Ahead:

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

1. The Curse of S3

What is S3?

- Extremely cheap!

1. The Curse of S3

What is S3?

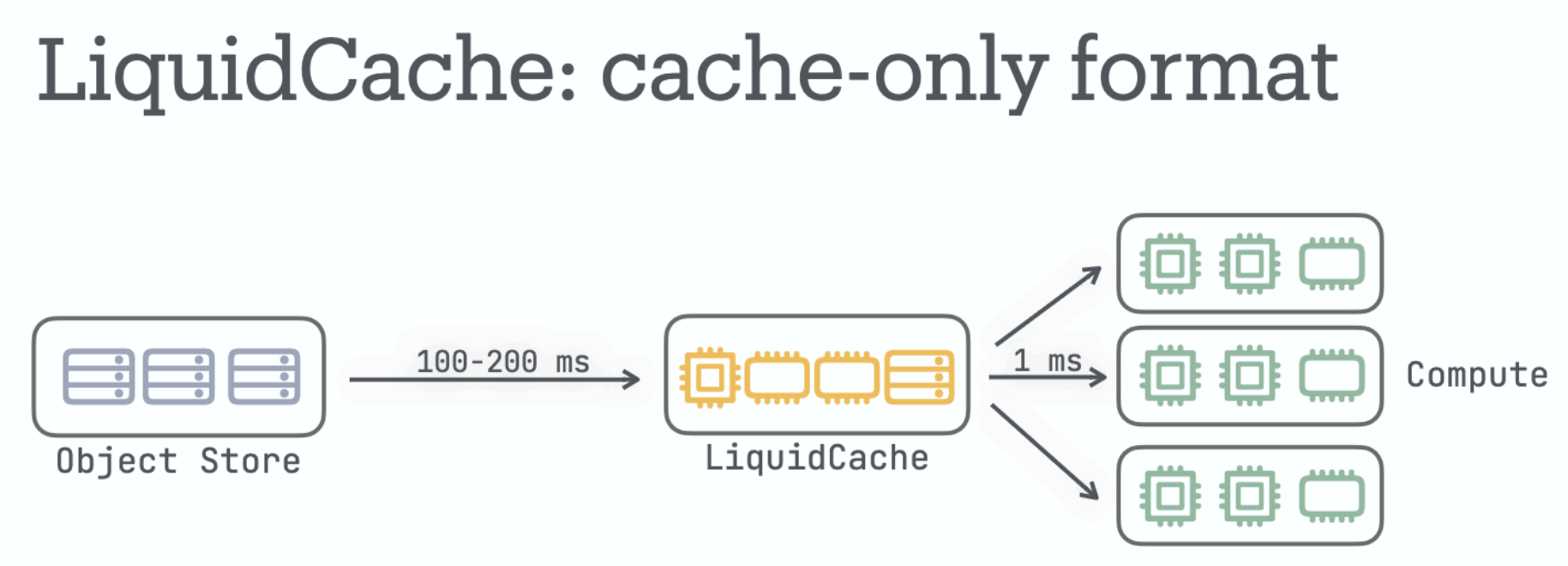
- Extremely cheap!
- Bandwidth: < 1 GB/s per connection
- Latency: ~100ms P99
- CPU-only: DNS, TLS, and HTTP

1. The Curse of S3

Why S3 is the Curse?

- GPUs idle due to S3 latency
- GPUs underutilized due to S3 bandwidth
- Moving the data path back to a CPU-centric system
- GPUs extremely expensive!

1. Caching Layer above 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?

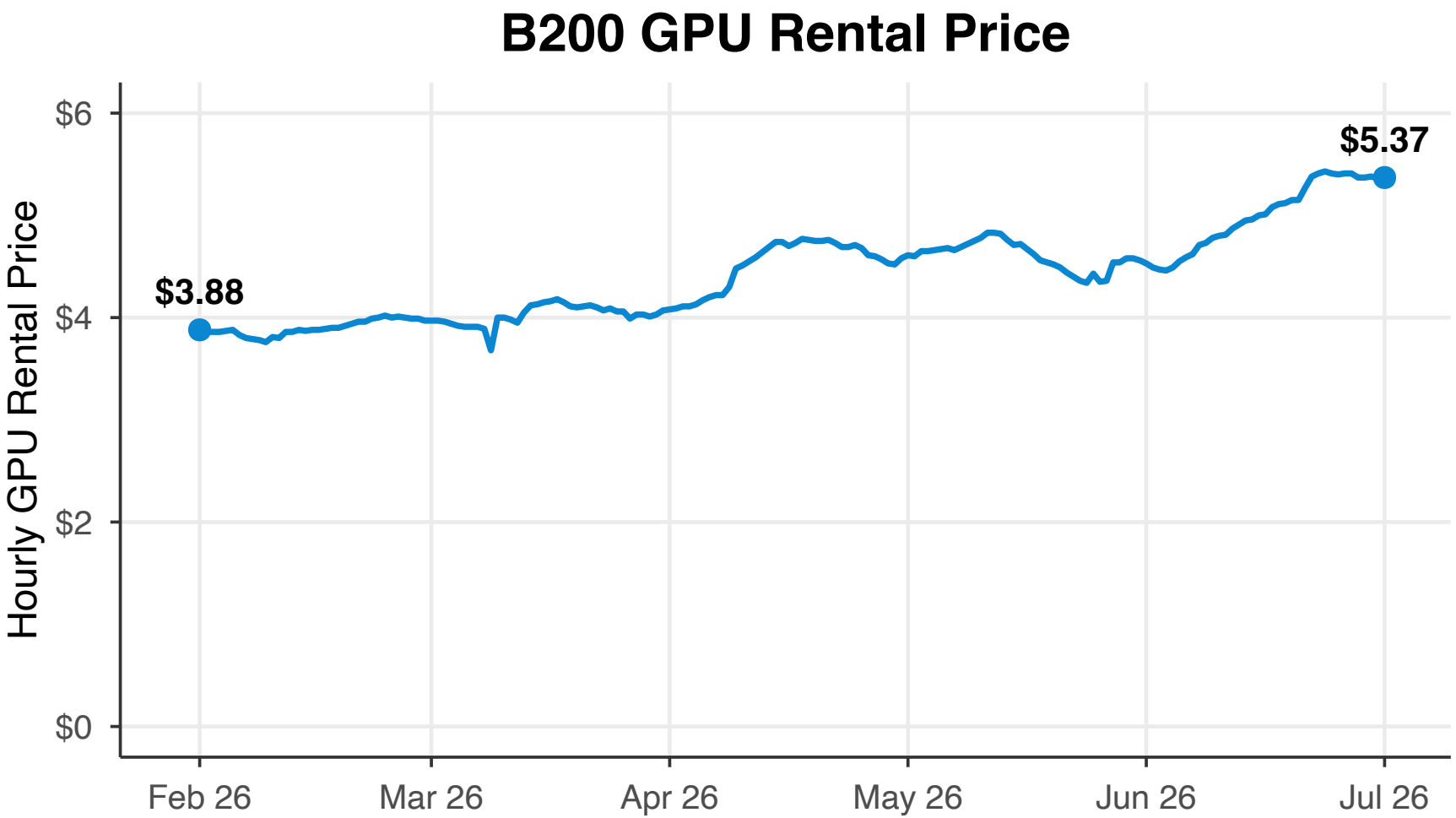
> *"If you want to go fast, go alone;*
> *if you want to go far, go together."*

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

2. Team Parquet or Team SomethingElse?

- File Formats: Software Campus × Snowflake
- Up to €115,000 · two years · from 2027
- 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 -> Old GPU

Thanks For Your Attention!

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