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

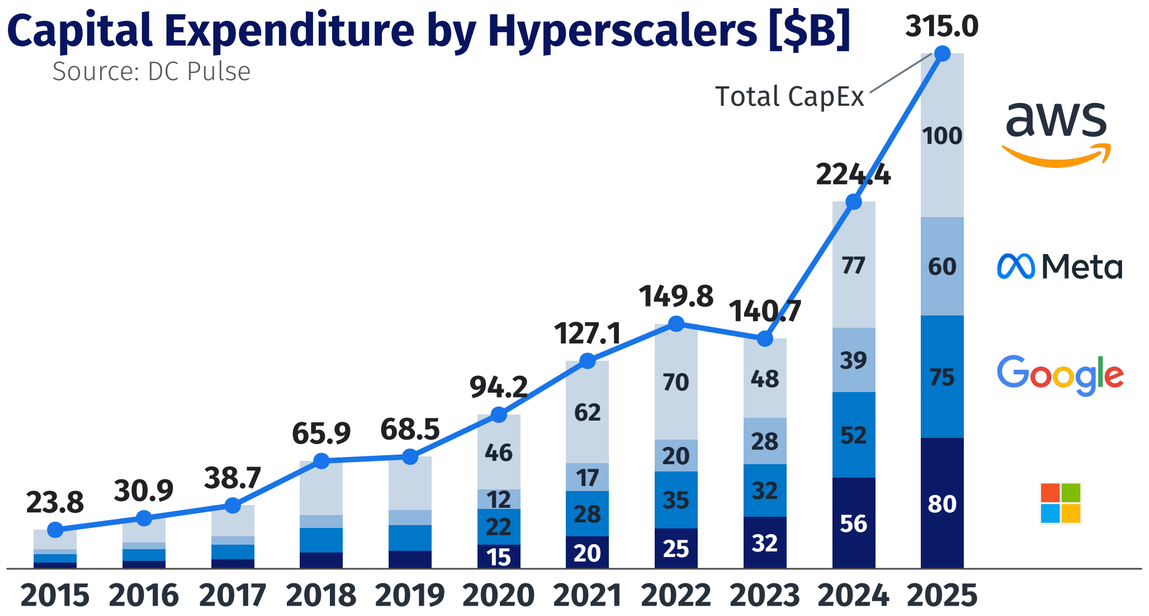
Jigao Luo*, Nils Boeschen*, Muhammad El-Hindi, Carsten Binnig

TU Darmstadt · TU Munich

SYSTEMS

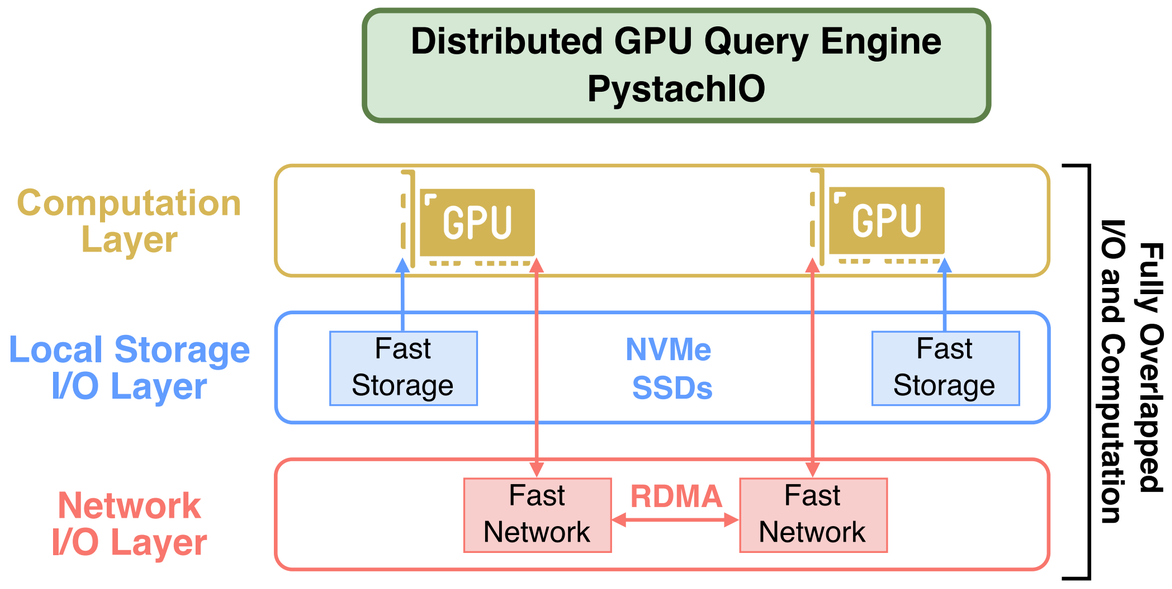
* Equal contribution

AI Investment Shapes Modern & Future HW



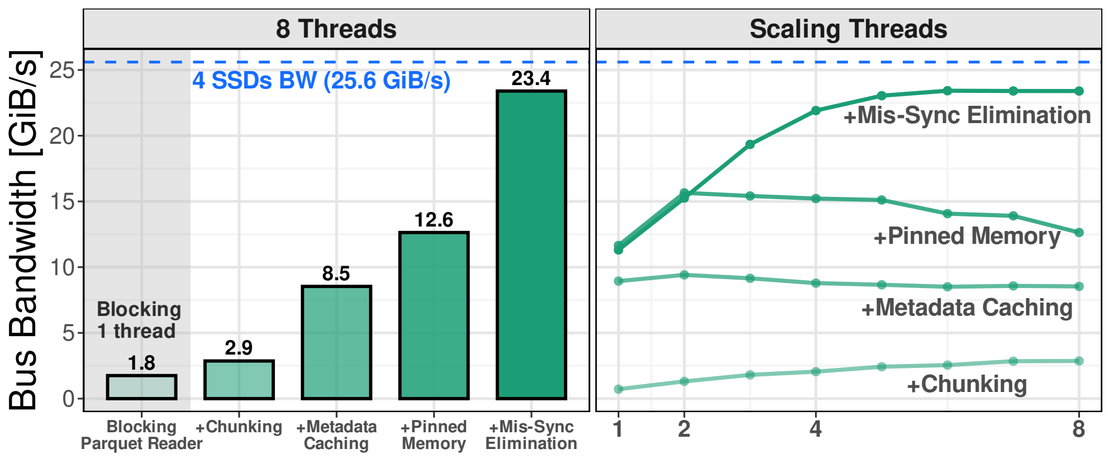
- CPU performance plateau → outside the AI spotlight
- Data centers shifting toward GPUs, storage and networks
- **How do we ride the AI investment wave for DB?**

Need for Re-Design & Contributions



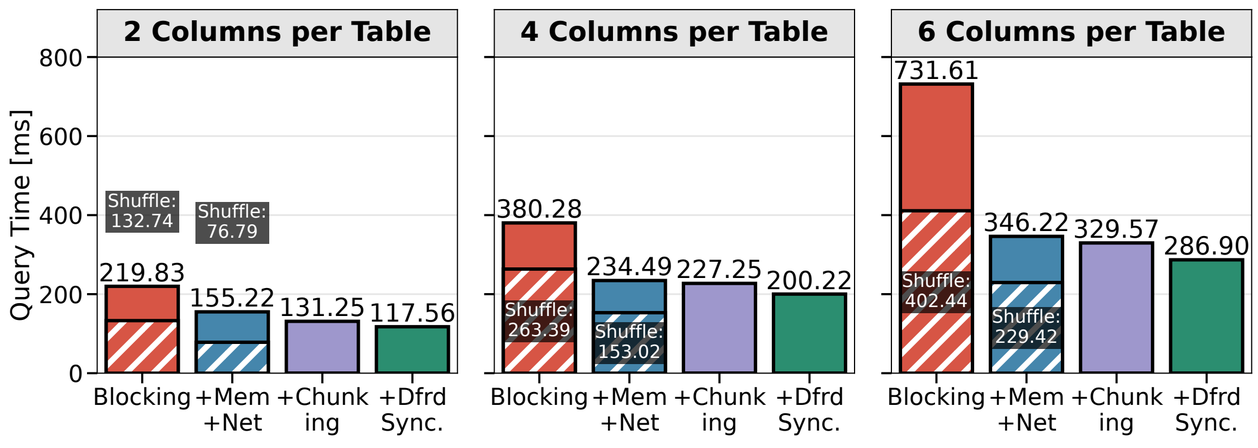
- Issue: Existing GPU DBs remain single-node only
- **Research Question: How to build a distributed GPU DB?**
- Challenges shifting from computation to storage & network I/O

Storage Access: Reading Parquet from SSDs into GPU



- Blocking reading: Metadata → SSD I/O → GPU Kernels
- Our reader: Overlapping computation and storage I/O

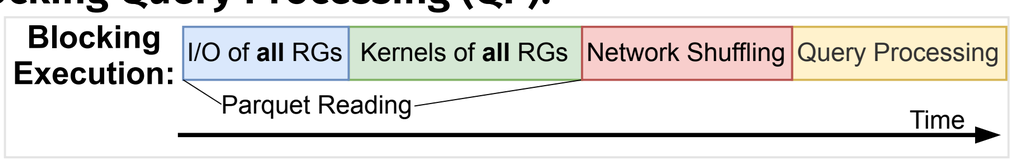
Distributed Execution: Shuffle over RDMA on Multi. GPUs



- Blocking join: Partition → Shuffle → Build → P → S → Probe
- Our distributed join: Overlapping partitioning, shuffle, and join

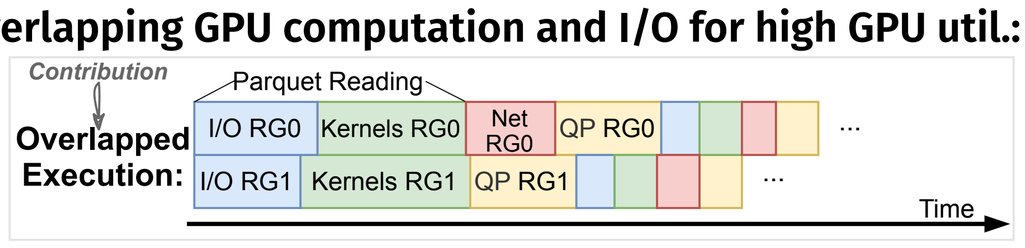
GPU Query Engine: *PystachIO* Over Fast Networks and Fast Storage

Blocking Query Processing (QP):

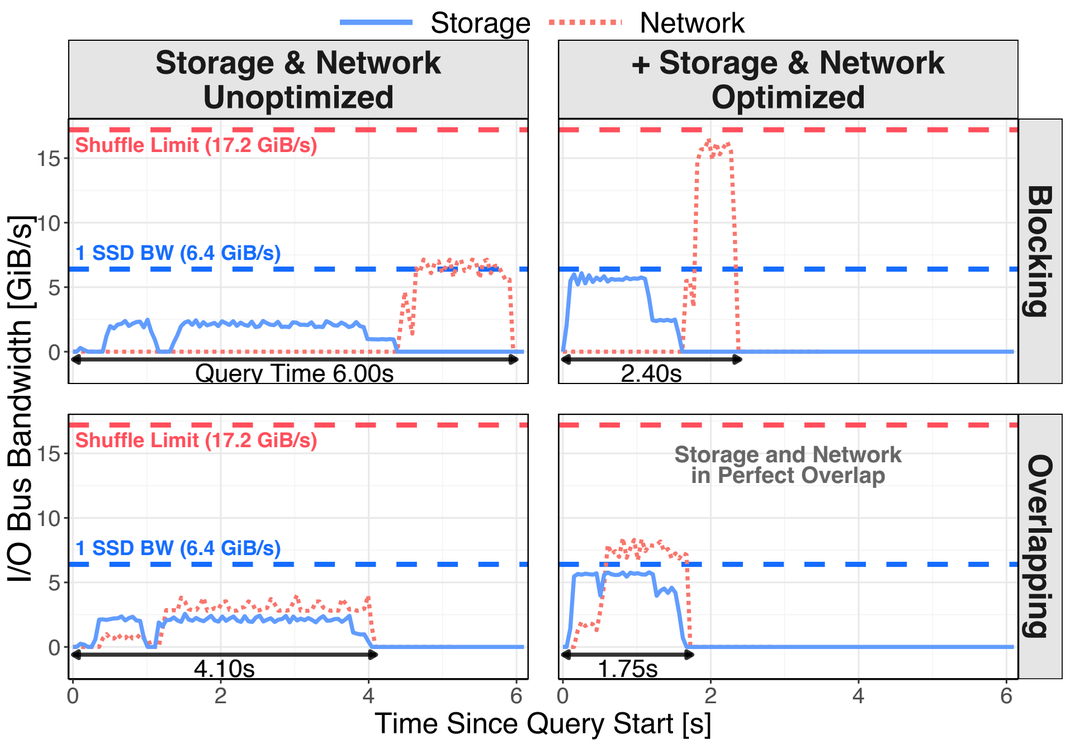


- GPUs idle during I/O: Parquet reading & RDMA shuffling

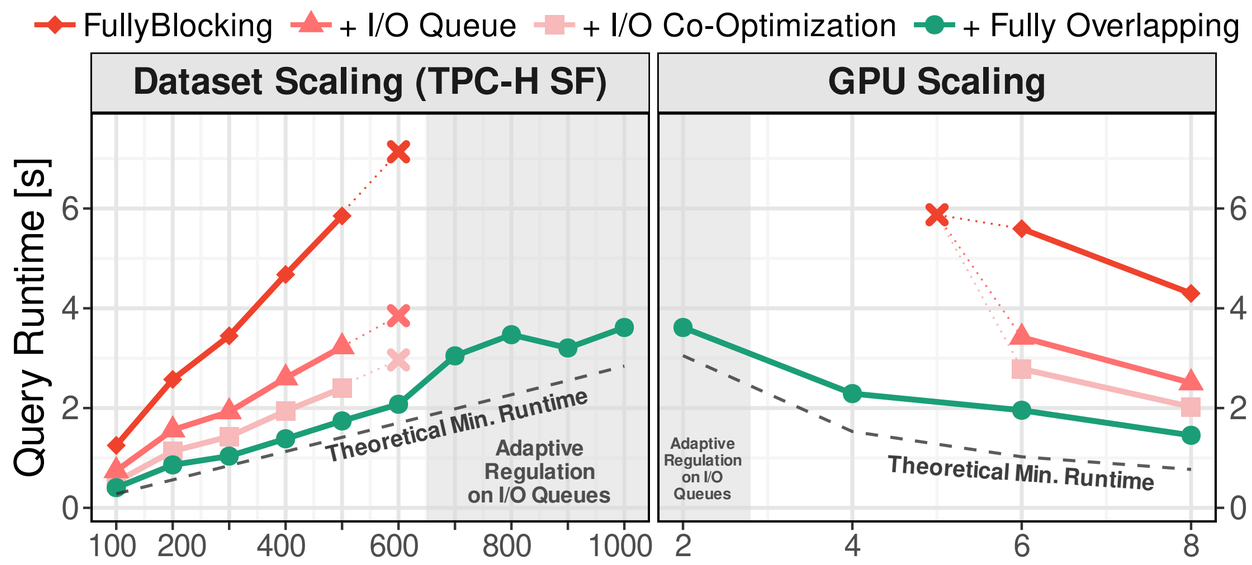
Overlapping GPU computation and I/O for high GPU util.:



- Storage I/O ↔ decoding + decompression in Parquet
- Network I/O ↔ partition + hash join (build & probe)
- Storage + Network I/O ↔ GPU QP (↔ := Overlapping)



Storage & Network with I/O Queue Coordination: TPC-H Query3



- Storage SSD: base tables
- HBM: I/O Queues & operators
- Storage → Queues → Network → QP: Full overlap (→ := Data Flow)

Conclusion

- GPUs and disaggregation reshape the data systems stack
- Focus shifting from computation to **I/O**: storage & network
- **Compute-I/O overlap** is key to I/O-intensive system design

Paper

Code

Personal Website