

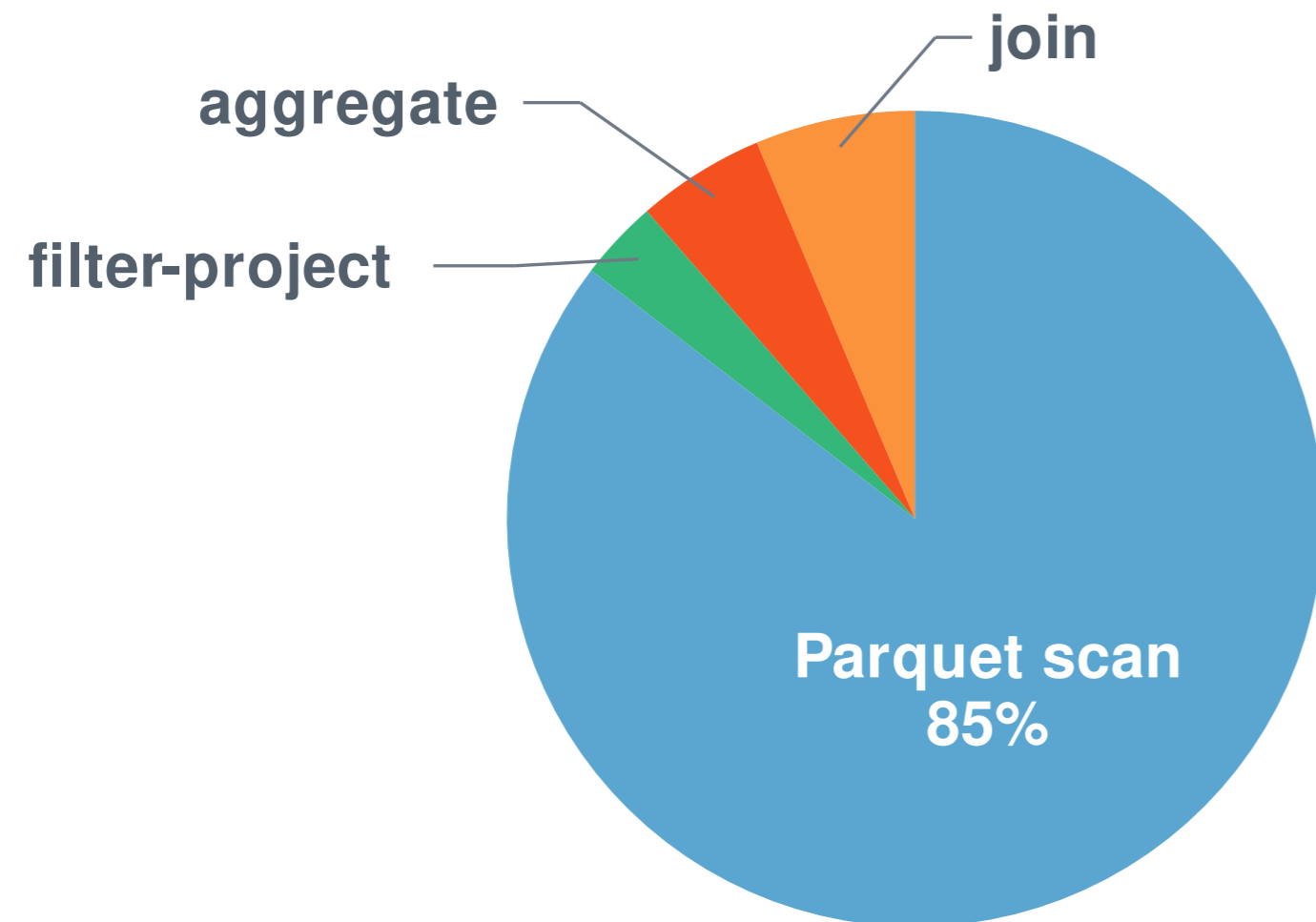
Do GPUs Really Need New Tabular File Formats?

Jigao Luo, Qi Chen, Carsten Binnig
Technische Universität Darmstadt



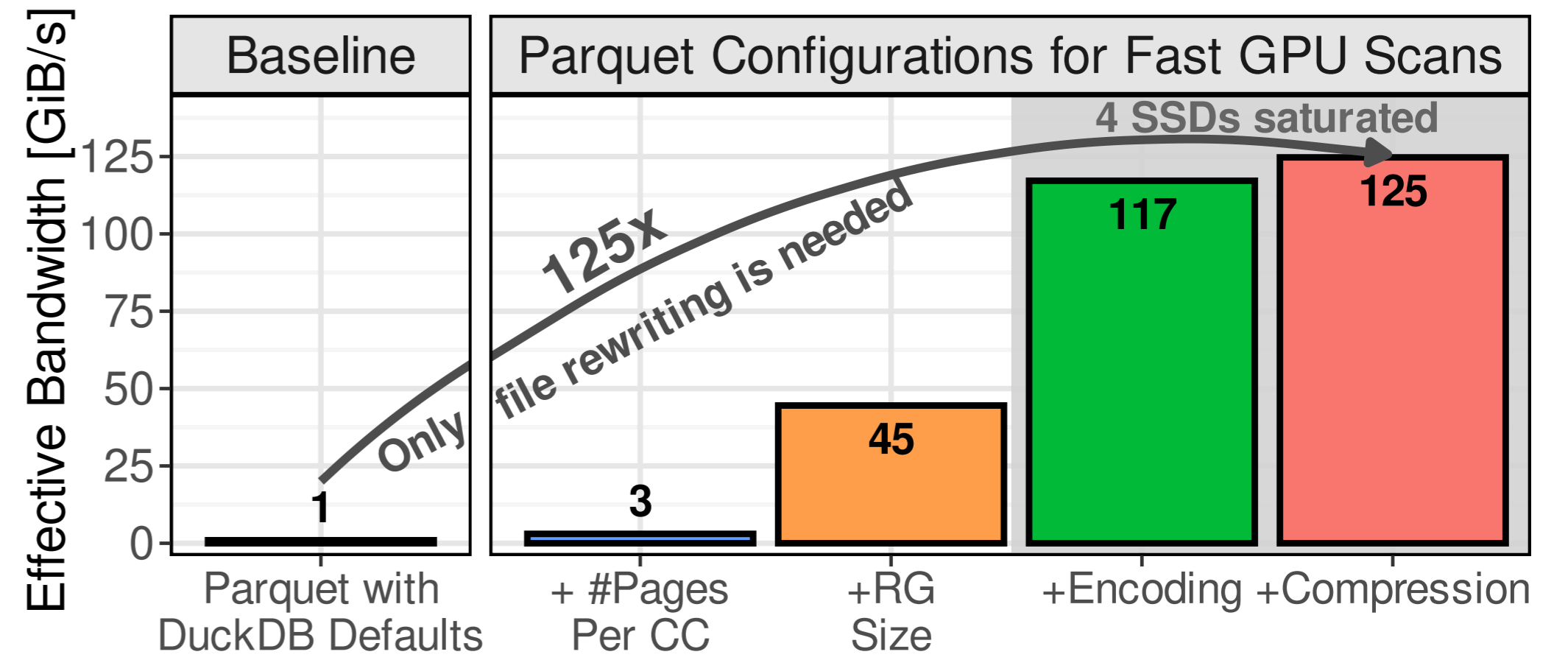
SYSTEMS

Motivation & Research Question



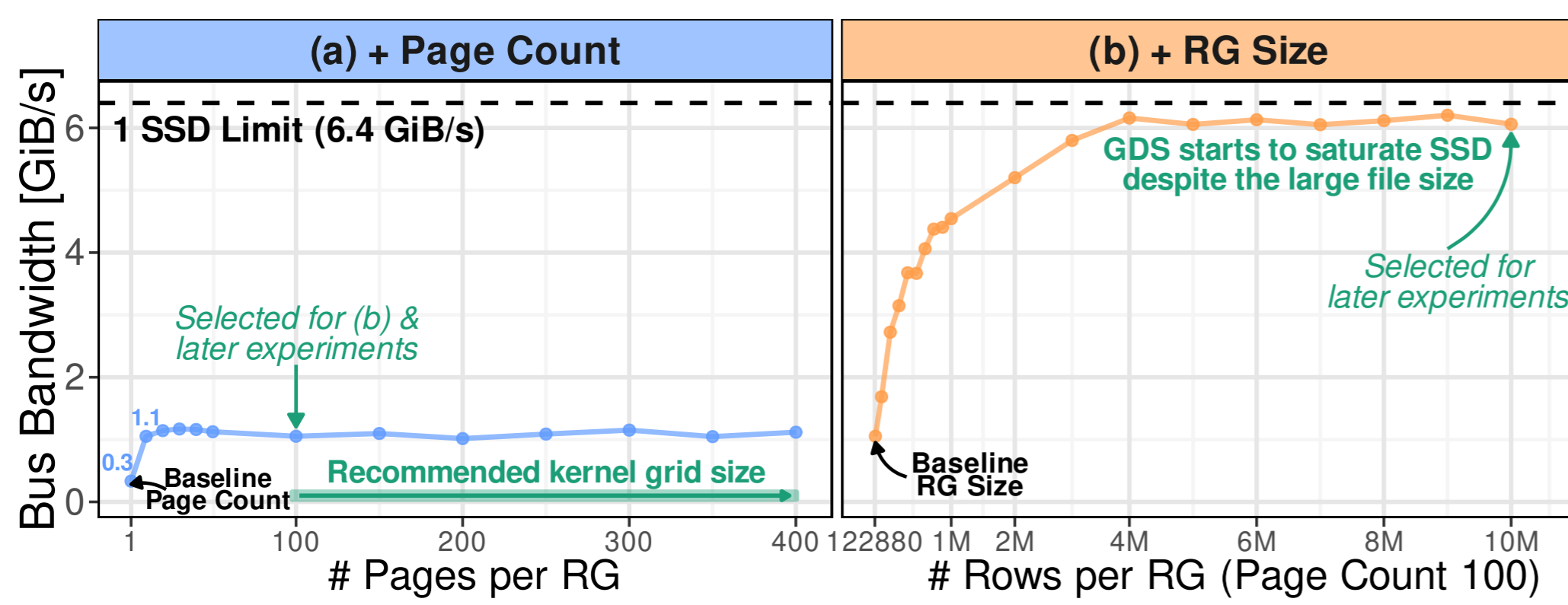
- Parquet Intro: RowGroup → ColumnChunk → Page: encoded & compressed
- NVIDIA RAPIDS: **85%(!)** of TPC-H runtime spent on Parquet scan
- Common perception to challenge: *Parquet is slow*
- Research Question: Do GPUs need new file formats?**

Idea & Contributions



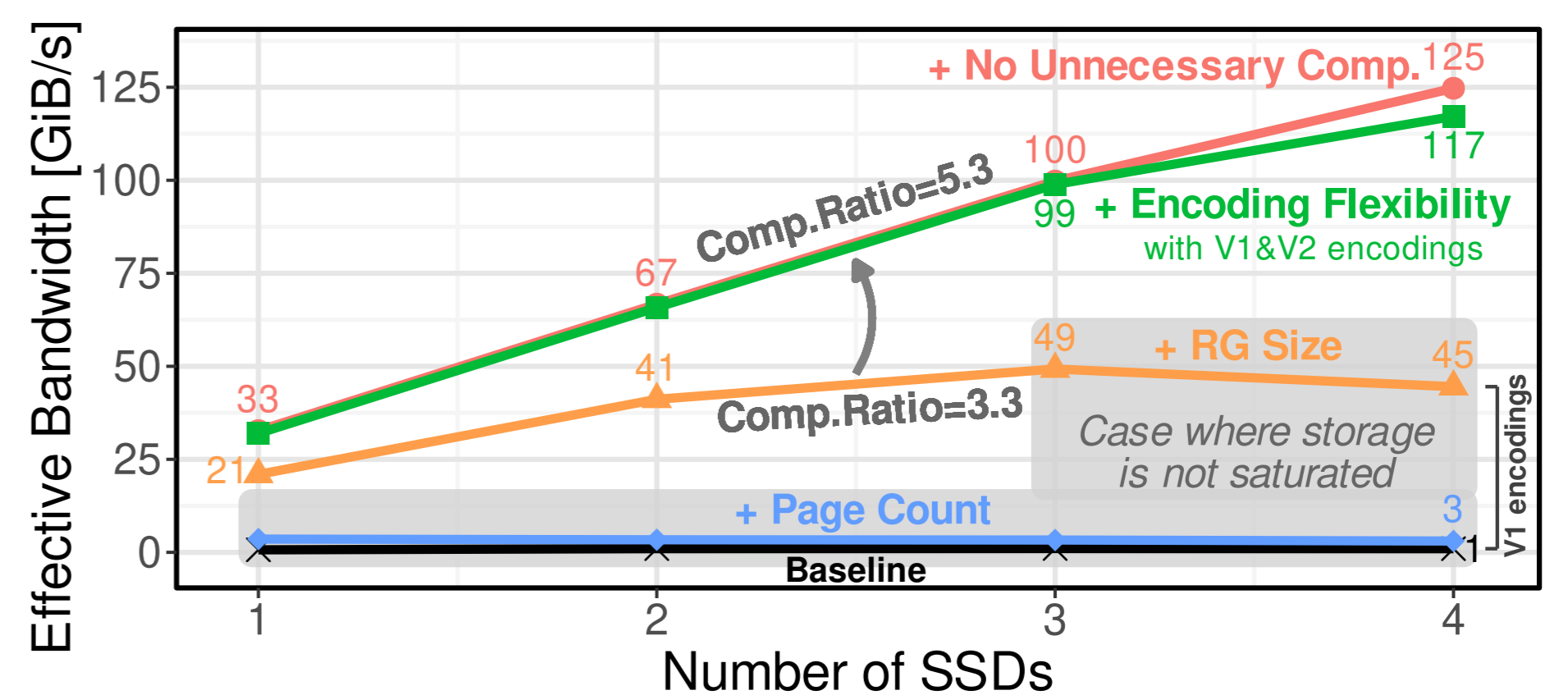
- Configuration defaults: inherited from CPU best practices
- Issue: **CPU defaults ≠ GPU-optimal**
- Misconception: Untuned Parquet is mistaken for slow Parquet
- Up to **125x** speedup after GPU-aware rewriting

Insight 1 & 2: # Pages in CC & RG Size



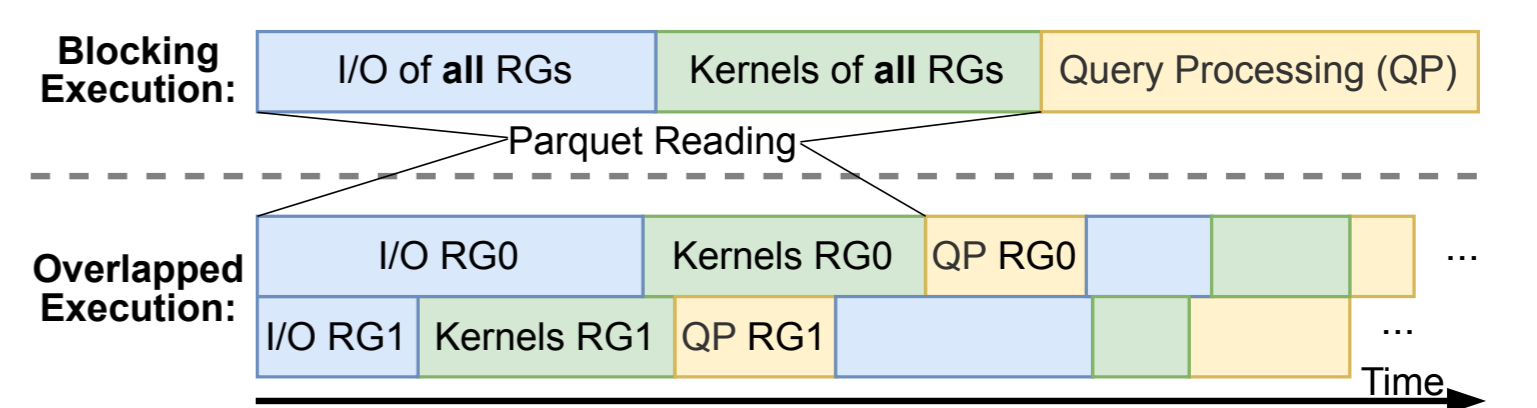
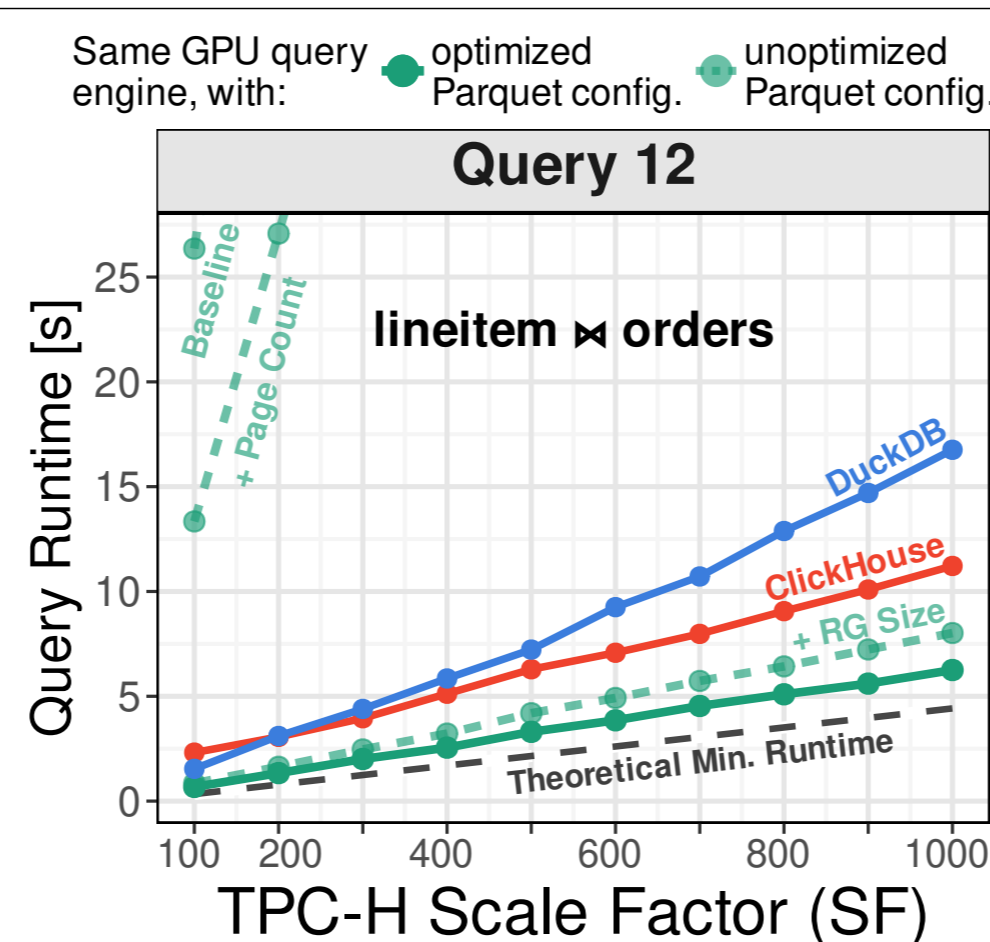
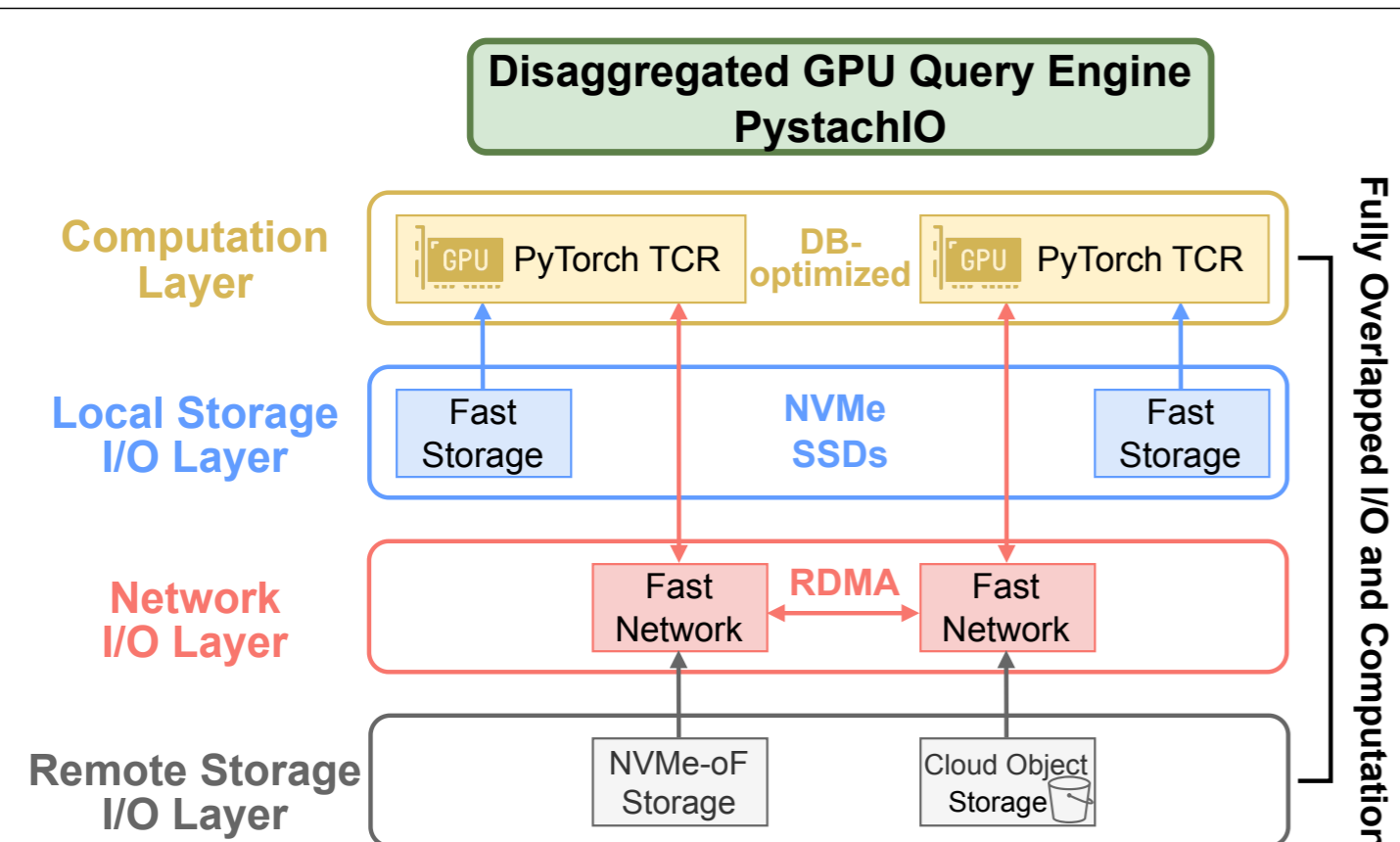
- # Pages in ColumnChunk → GPU decoding kernel grid size
 - Insight 1:** Avoid too few pages to prevent low GPU util.
- GPU I/O stack: MB-level I/O sizes saturate SSDs
 - Insight 2:** Use million-row RG sizes to saturate storage

Insight 3 & 4: Encoding & Compression



- Search for the minimum-size encoding per ColumnChunk
 - Insight 3:** Flexible encoding improves compression ratio
- Some compression cases give zero size reduction
 - Insight 4:** Avoid these compression cases

[Extra] PystachIO, Our Distributed Disaggregated GPU Query Engine

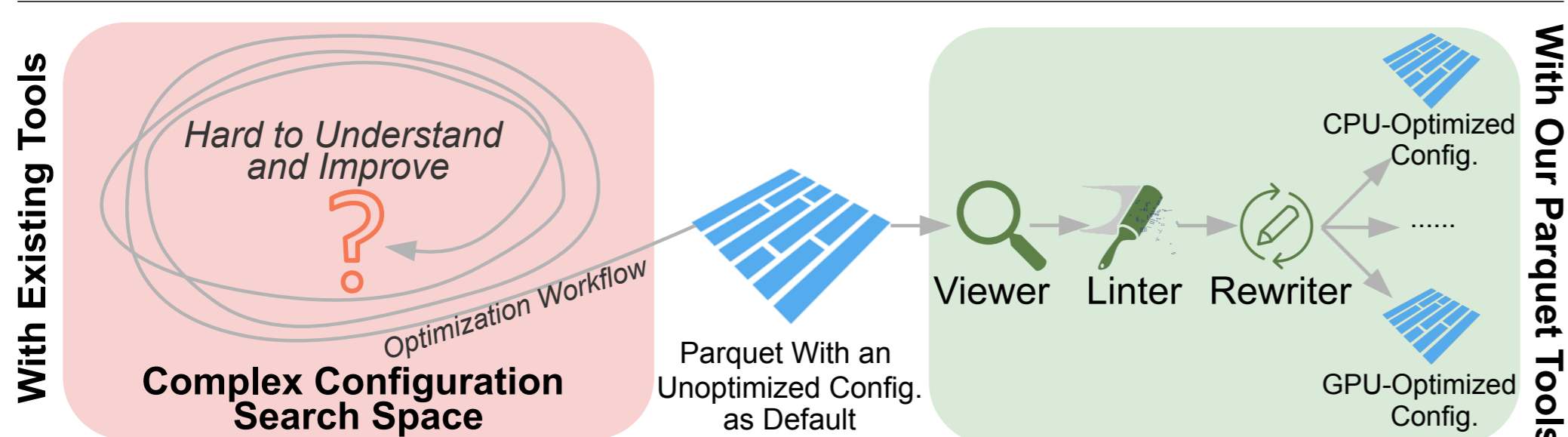


Overlapping:

- Storage I/O ↔ decoding + decomp.
- Storage + Network I/O ↔ query processing

VLDB 2026, Research Track

[Extra] Parquet Tools for Automating Optimization



- Parquet tools:
 - Viewer: understand configurations
 - Linter: gather suggestions
 - Rewriter: apply suggested configurations
- For **new file formats**: Are there any tools available?

Demo (under submission) with Andrew Lamb & Xiangpeng Hao

Conclusion & Takeaway

- 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!

