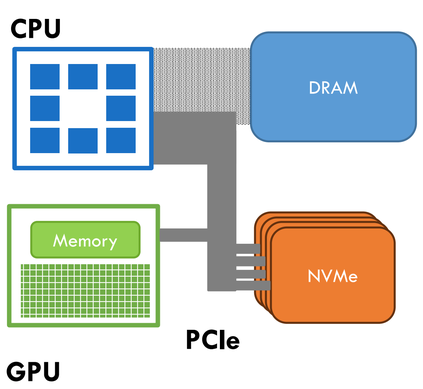
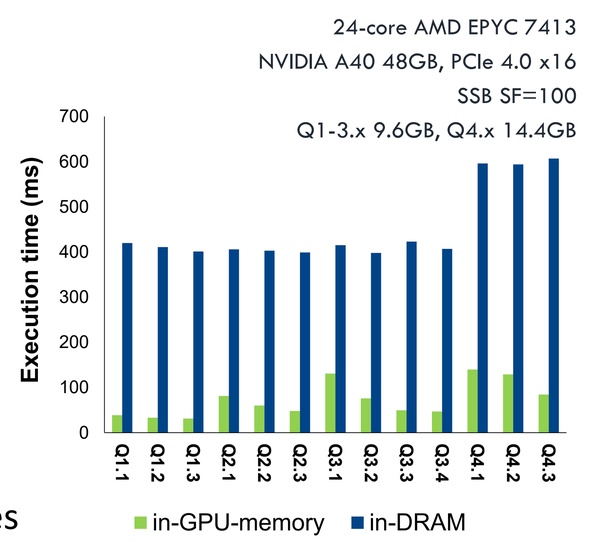
The Effectiveness of Compression for GPU-Accelerated Queries on Out-of-Memory Datasets

Hamish Nicholson, Konstantinos Chasialis, Antonio Boffa, Anastasia Ailamaki
hamish.nicholson@epfl.ch, konstantinos.chasialis@oracle.com, antonio.boffa@epfl.ch, anastasia.ailamaki@epfl.ch

Queries on out-of-memory datasets are limited by interconnect



Interconnect bottleneck causes up to 13× slower queries



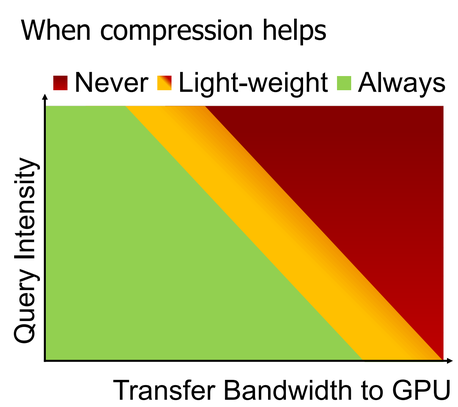
Compression as a Solution to the Interconnect Bottleneck

Recent research:

1) in-memory engines introduce new lightweight encoders

2) out-of-memory systems rely on heavyweight compressors

Miss the full picture:
whether compression helps depends jointly on bandwidth, query intensity, and selectivity

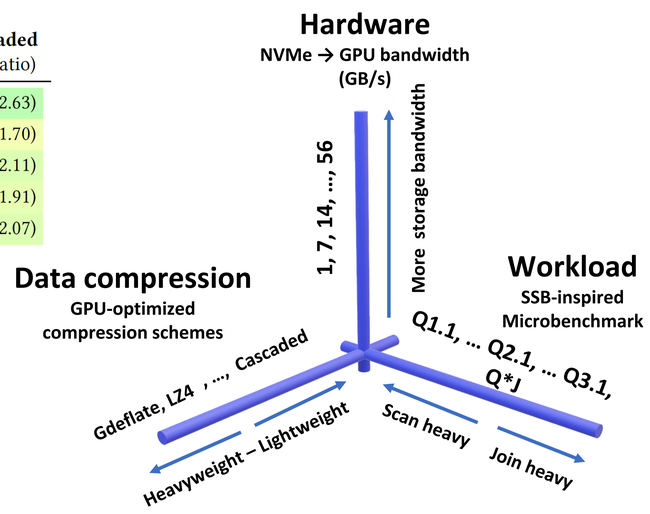


Compression reduces data movement but brings new performance trade-offs

The effectiveness depends on more than just compression ratios

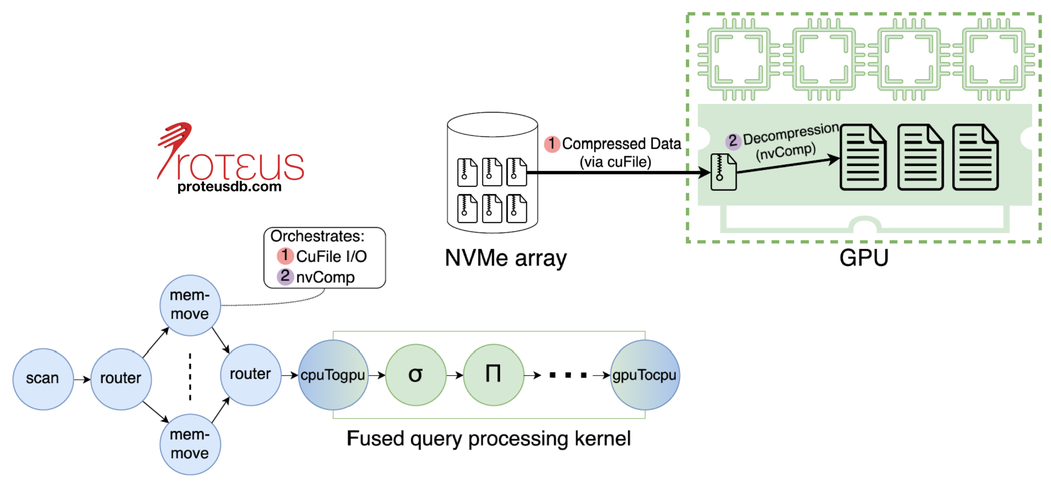
Query	Raw GB	LZ4 GB (ratio)	GDeflate GB (ratio)	Cascaded GB (ratio)
Q1.1	96	65.7 (1.46)	49.0 (1.96)	36.5 (2.63)
Q2.1	96	79.9 (1.20)	75.0 (1.28)	56.5 (1.70)
Q3.1	96	65.5 (1.47)	77.0 (1.25)	45.4 (2.11)
Q4.1	144	110.8 (1.30)	112.7 (1.28)	75.4 (1.91)
Q*J	144	107.3 (1.34)	103.6 (1.39)	69.4 (2.07)

Key factors that govern compression effectiveness?



Prior research overlooks the conditional nature of compression benefits in GPU-accelerated systems

On-the-fly decompression inside mem-move operator

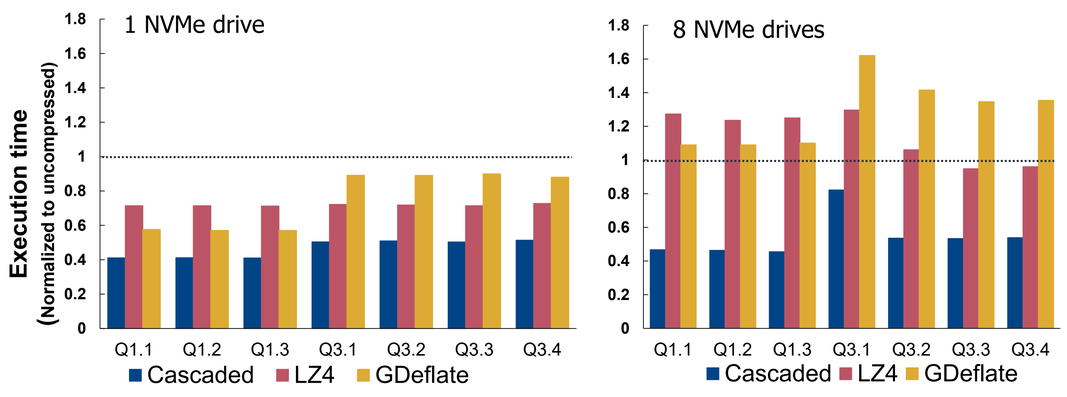


Analysis of compression benefit on out-of-memory datasets along 3 axis

Results on Start Schema Benchmark

Speed up ≃ Compression ratio

SSB SF=1000
1-8x NVMe, PCIe 4 x4



GDeflate and LZ4 degrade performance

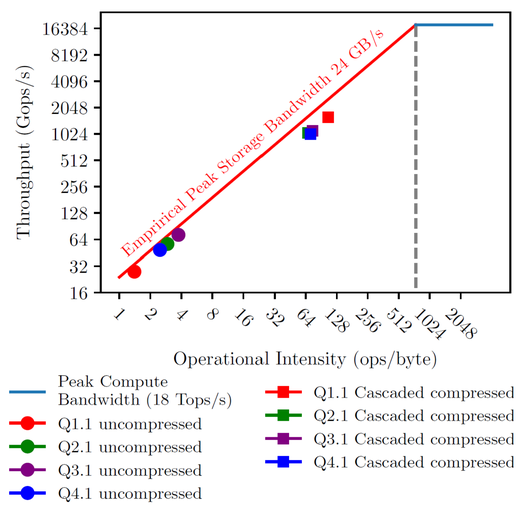
Compression helps most when storage is the bottleneck

Roofline Analysis. Are we still interconnect bound?

From 1-3 ops/byte
To 67-104 ops/byte

28× more instructions executed

2× speed up

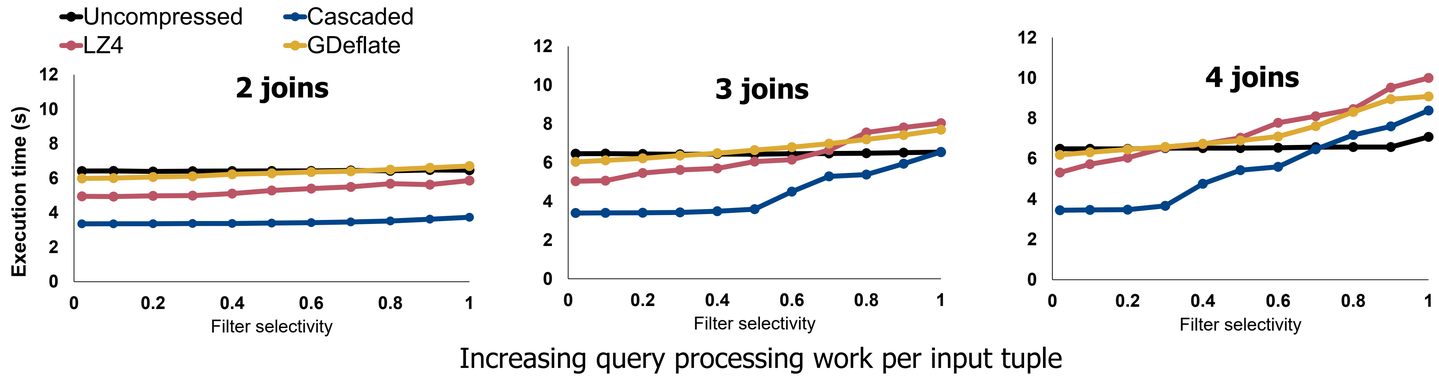


Even with decompression, analytical queries remain far from compute roof
Leaving room for more (efficient) compression

When compression introduces a slow down

With high bandwidth, process intensive queries, high selectivity, decompression can stall the GPU

```
SELECT SUM(lo_revenue)
FROM date, customer, supplier, part, lineorder
WHERE lo_quantity < {FILTER_SELECTIVITY_PARAM}
AND lo_orderdate = d_datekey          ← Q1J, one join
AND lo_suppkey = s_suppkey            ← Q2J, two joins
AND lo_partkey = p_partkey            ← Q3J, three joins
AND lo_custkey = c_custkey            ← Q4J, four joins
```



Increasing query processing work per input tuple