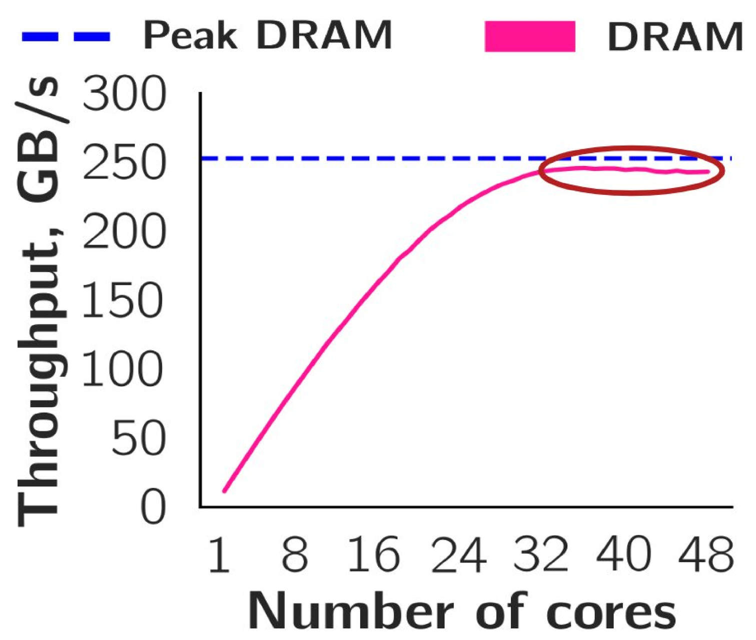
Demystifying CXL Memory Bandwidth Expansion for Analytical Workloads

Georgiy Lebedev, Hamish Nicholson, Musa Ünal, Sanidhya Kashyap and Anastasia Ailamaki
Data-Intensive Applications and Systems Laboratory & Robust Scalable Systems Software Laboratory

Analytics Rely on DRAM for Performance

CPU: 48 cores, 1 socket
DRAM: 253 GB/s
Sequential access: 32 GB array, 8-byte integers

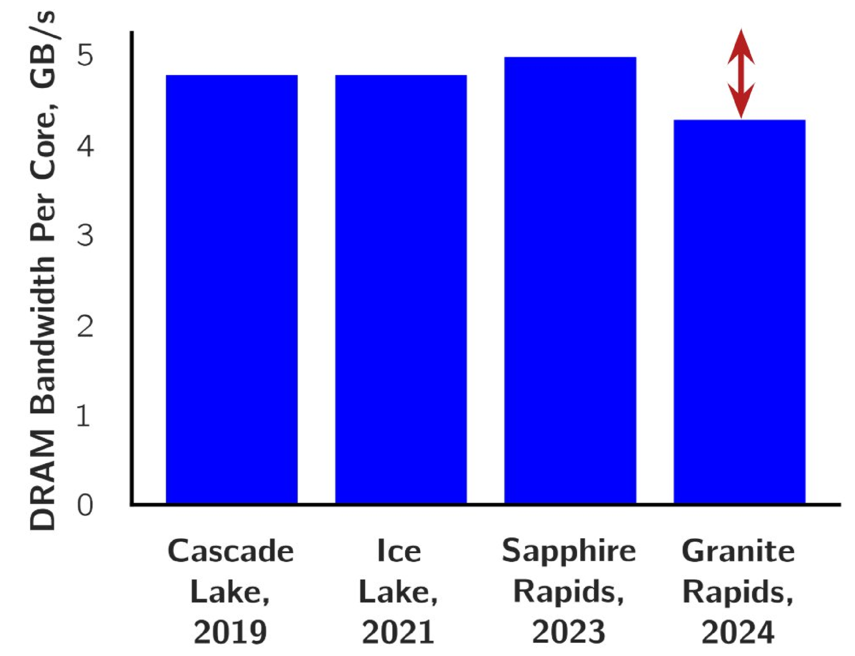
- Bandwidth stalls limit performance
- CPU cores wasted



Bottleneck: Memory Bandwidth

Bandwidth-Per-Core Improvement is Stagnating

- # chips per channel — signal integrity
- # channels per socket — pin budget
- Bandwidth/pin — DDR interconnect

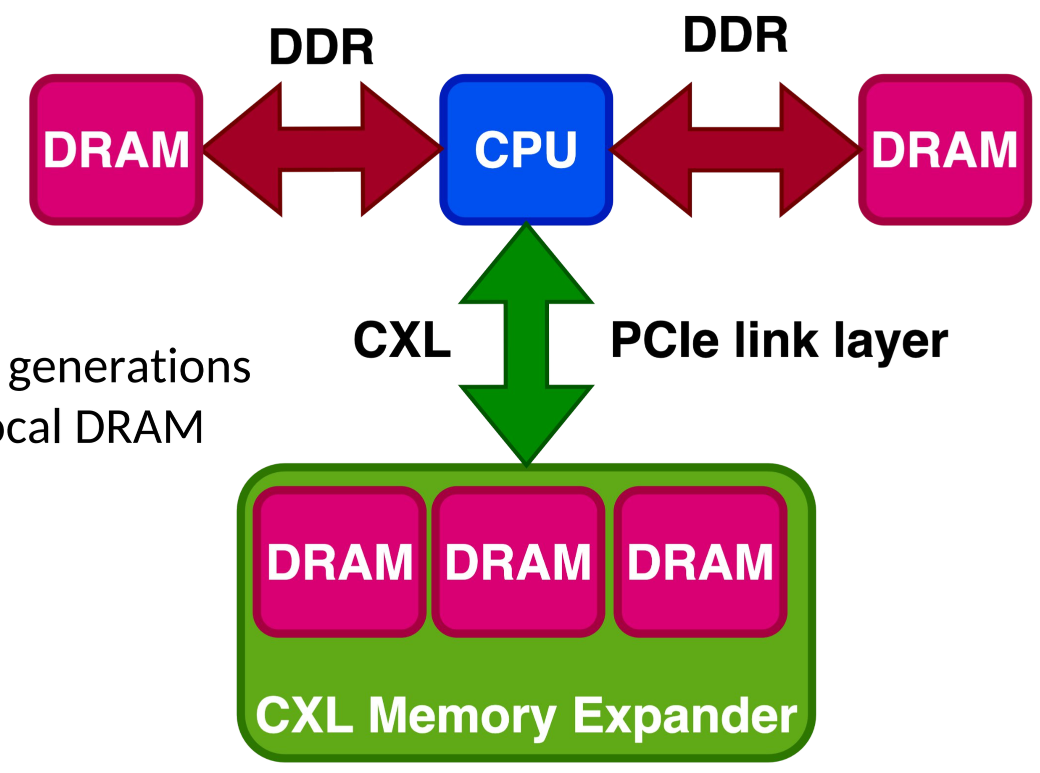


Additional Source of Bandwidth is Needed

CXL Adds Memory Bandwidth from PCIe

DRAM: 253 GB/s
CXL: 41 GB/s

- Bandwidth scales with PCIe generations
- Performance differs from local DRAM

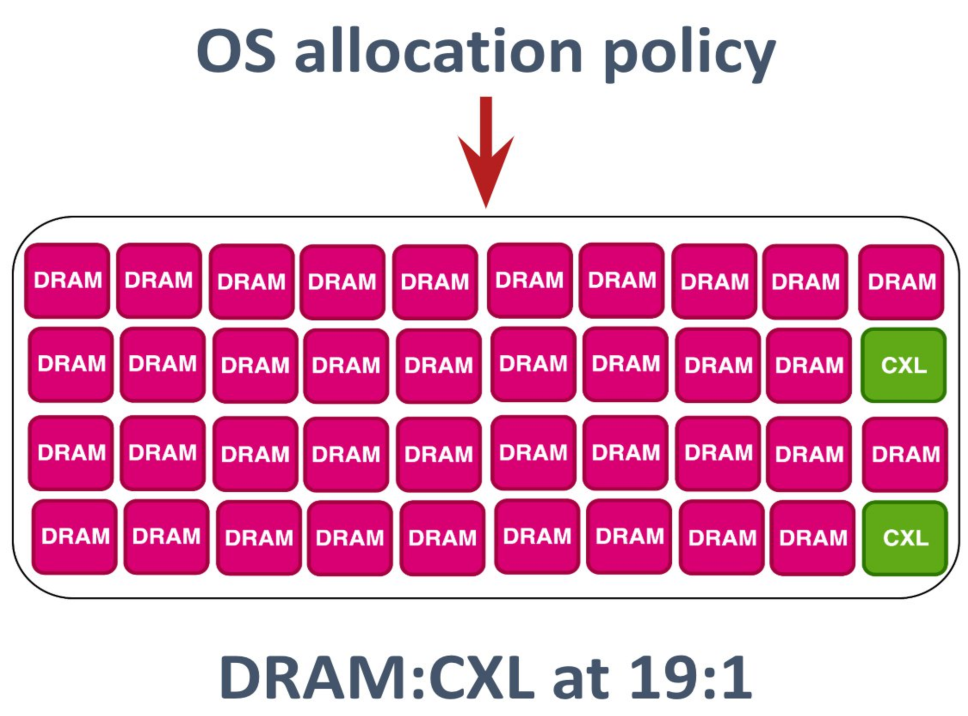


Heterogeneity Requires New Allocation Model

Weighted Heterogeneous Memory Interleaving

DRAM: 253 GB/s
CXL: 41 GB/s

- Avoid software modification
- Adjust to difference in bandwidths
- Maximize system bandwidth

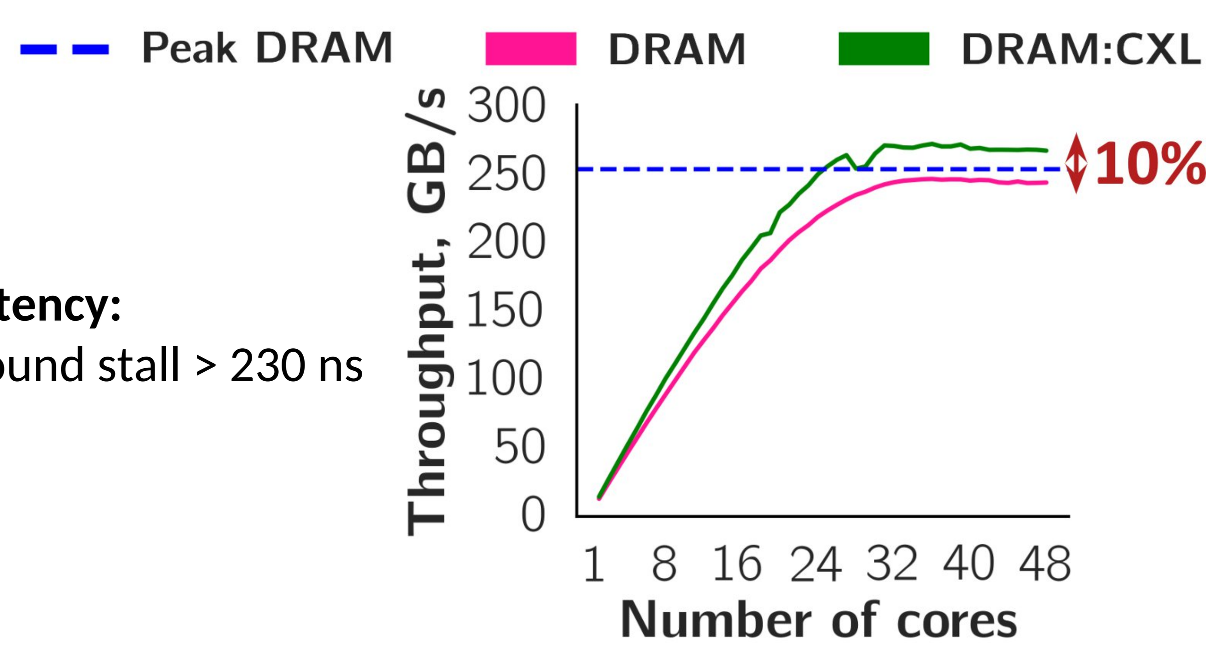


CXL as Another Memory Channel

Performance Improvement from CXL

CPU: 48 cores, 1 socket
DRAM: 253 GB/s
CXL: 41 GB/s
Sequential access: 32 GB array, 8-byte integers

DRAM latency > CXL latency:
111 ns + bandwidth-bound stall > 230 ns

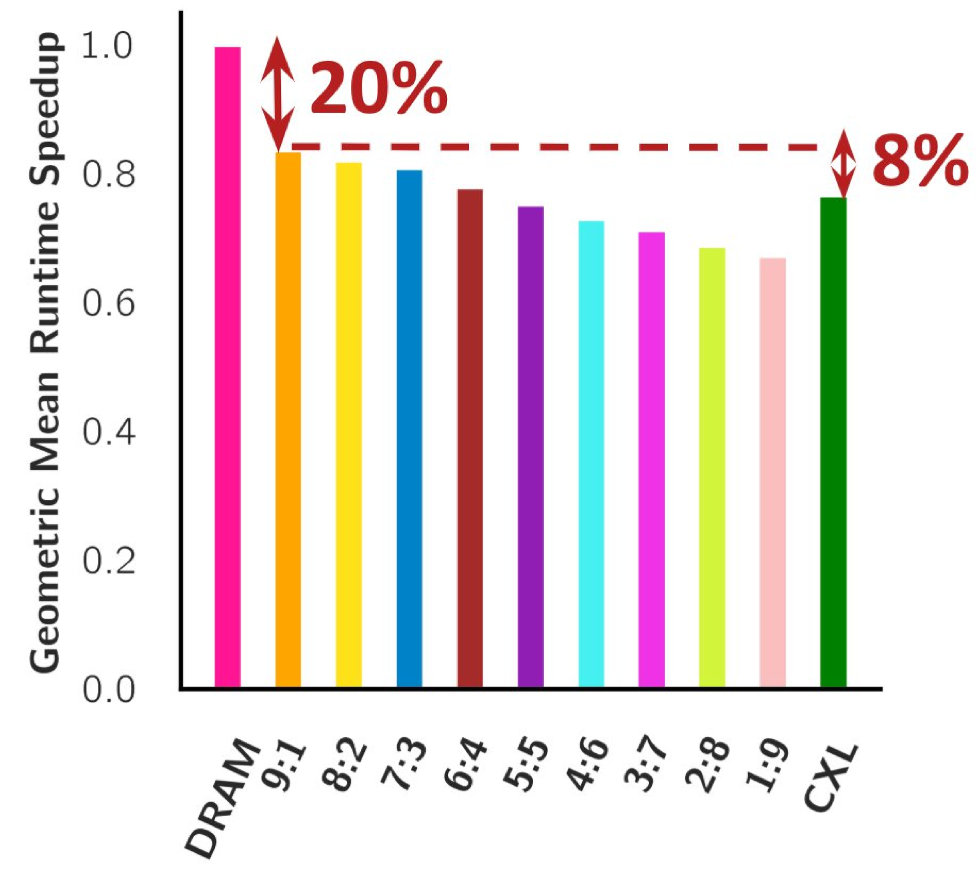


Interleaving CXL Reduces Bandwidth Stalls

Transparent Interleaving Degrades Performance

CPU: 48 cores, 1 socket
DRAM: 253 GB/s
CXL: 41 GB/s
SSB: SF 100, 59GB RSS, Proteus query engine

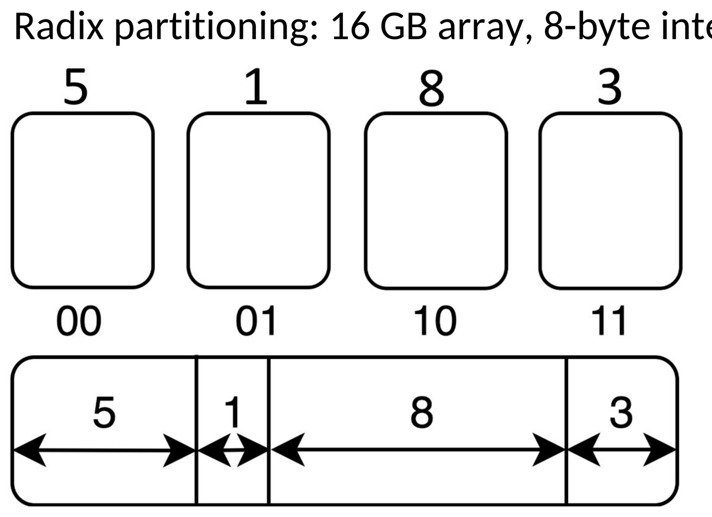
- Latency-critical memory on CXL
- Latency stalls dominate
- % of memory-bound CPU cycles ↑

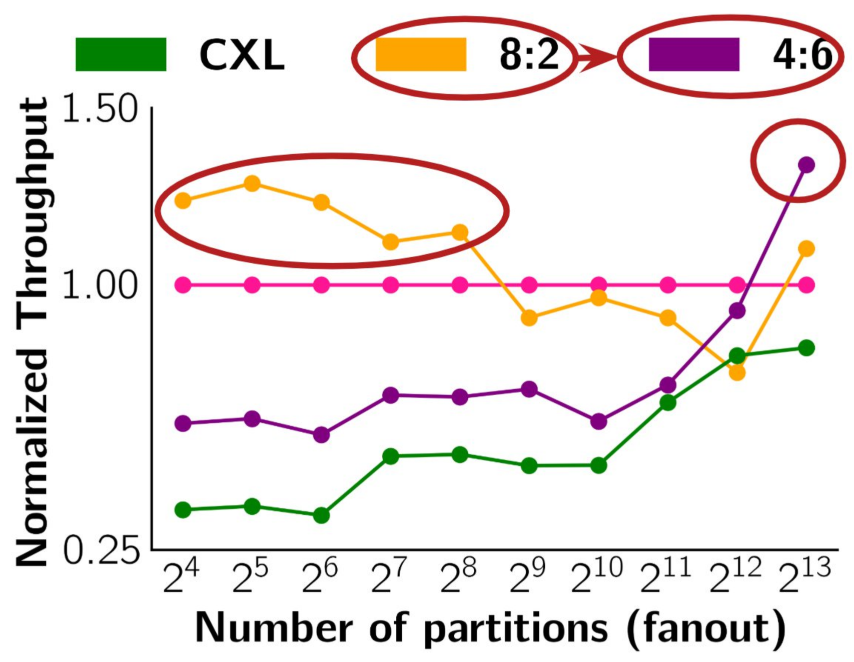


Latency- and Bandwidth- Awareness is Needed

Workload Adaptivity is Essential for Performance

CPU: 48 cores, 1 socket
DRAM: 126 GB/s (SNC-2)
CXL: 41 GB/s
Radix partitioning: 16 GB array, 8-byte integers





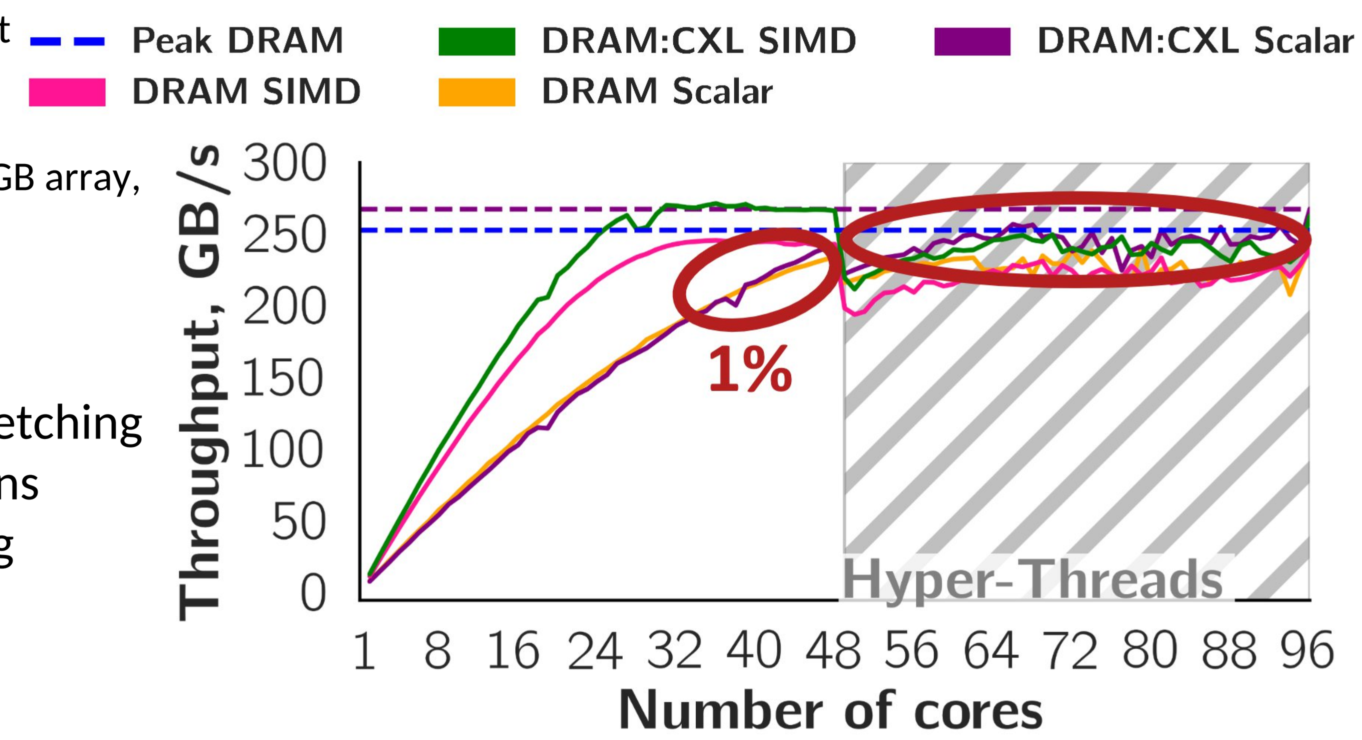
- Read-write ratio moves towards more writes
- Store-bound stalls ↓

Optimal Interleaving Weights Vary by up to 88%

High degree Memory-Level Parallelism is Critical

CPU: 48 cores, 1 socket
DRAM: 253 GB/s
CXL: 41 GB/s
Sequential access: 32 GB array, 8-byte integers

- Aggressive prefetching
- SIMD instructions
- Hyper-threading



Poor MLP Diminishes Gains from 10% to 1%