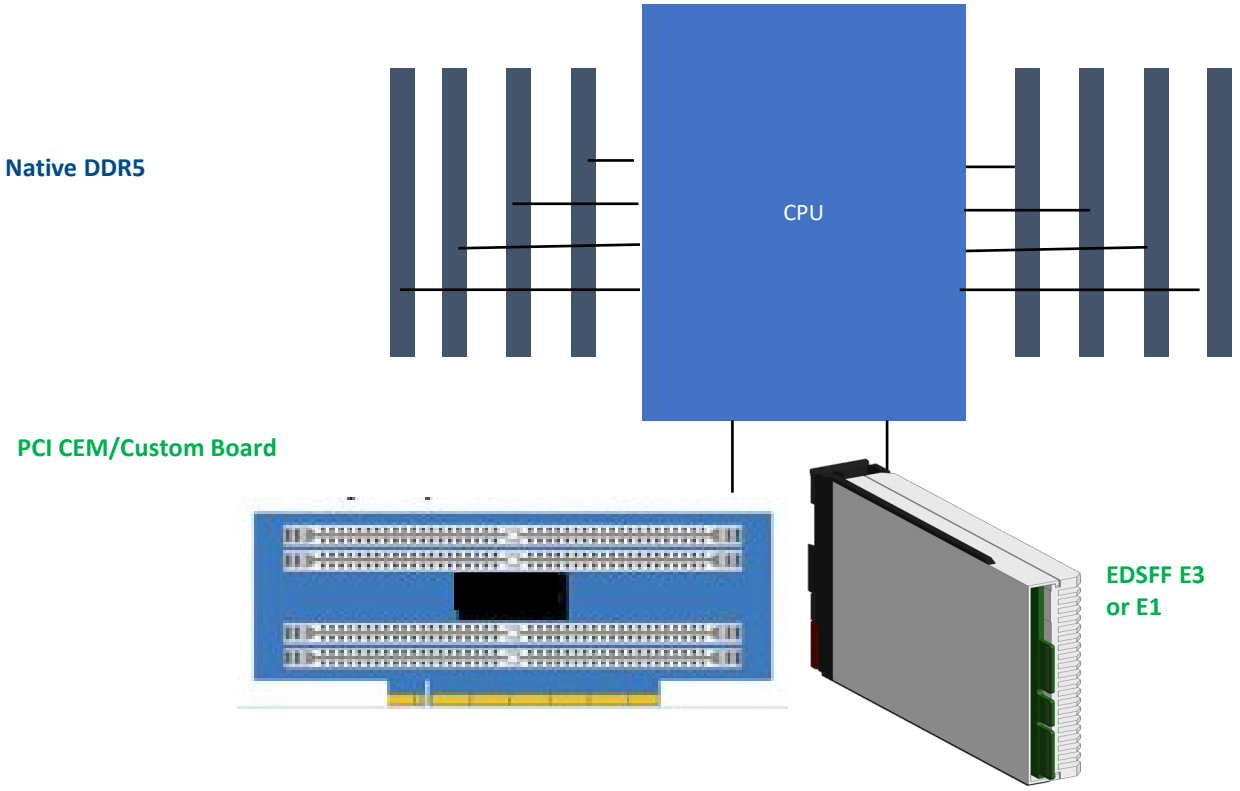
Boost In-memory Database Performance with CXL Memory

Anil Godbole

Datacenter Marketing @Intel

Boost In-memory Database Performance with CXL Memory

Today memory is one of the highest BOM cost item in a Datacenter Server/Rack. Extensive deployment for In-memory database & AI workloads is the primary reason. Multi-multi core CPUs are now available to support the needed compute power. Besides the native DRAM memory channels these CPUs also sport CXL interfaces to augment the memory capacity & bandwidth of the server. This presentation will focus on innovative ways CXL memory can be used to boost performance of the workloads mentioned above.



Value Prop of CXL-attached Memory

Increased Memory Capacity

Improve processor perf

- Reduced SSD accesses

Benefitting Workloads

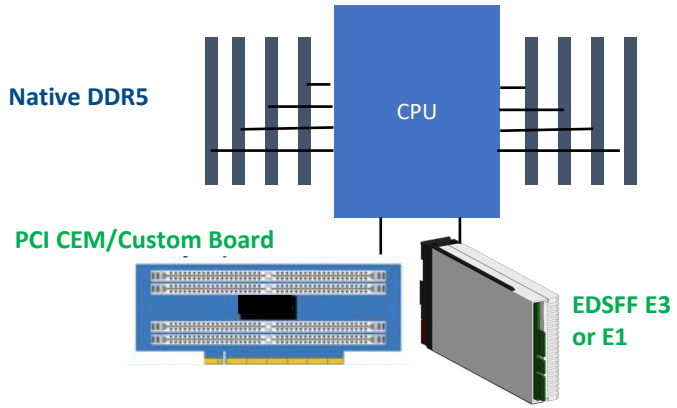
- Virtualized Servers
- In-memory Databases
- AI / ML
- HPC (High Perf Computing)
- Media (CDN, Video 8K)
- Medical (Genomics)

Increased Memory Bandwidth

Improve processor's memory bandwidth using address interleaving

Benefitting Workloads

- AI/ML
- HPC
- Non-relational Databases



Lower Memory TCO

Avoid expensive 3DS DIMMs

- Use standard DIMM capacities for native & CXLmemory
- Use lower-cost memory media on CXL
  - DDR4

Memory Pooling

- For optimal provision of local DRAM on servers

TCO Savings Examples with CXL Memory

- 128GB and 256GB DIMMs have high price premiums
- CXL Add-in cards with DIMM slots
  - Provide more total channels per CPU socket
  - Allowing same system capacity with lower priced DIMMs
- CXL EDSFF modules can also be used

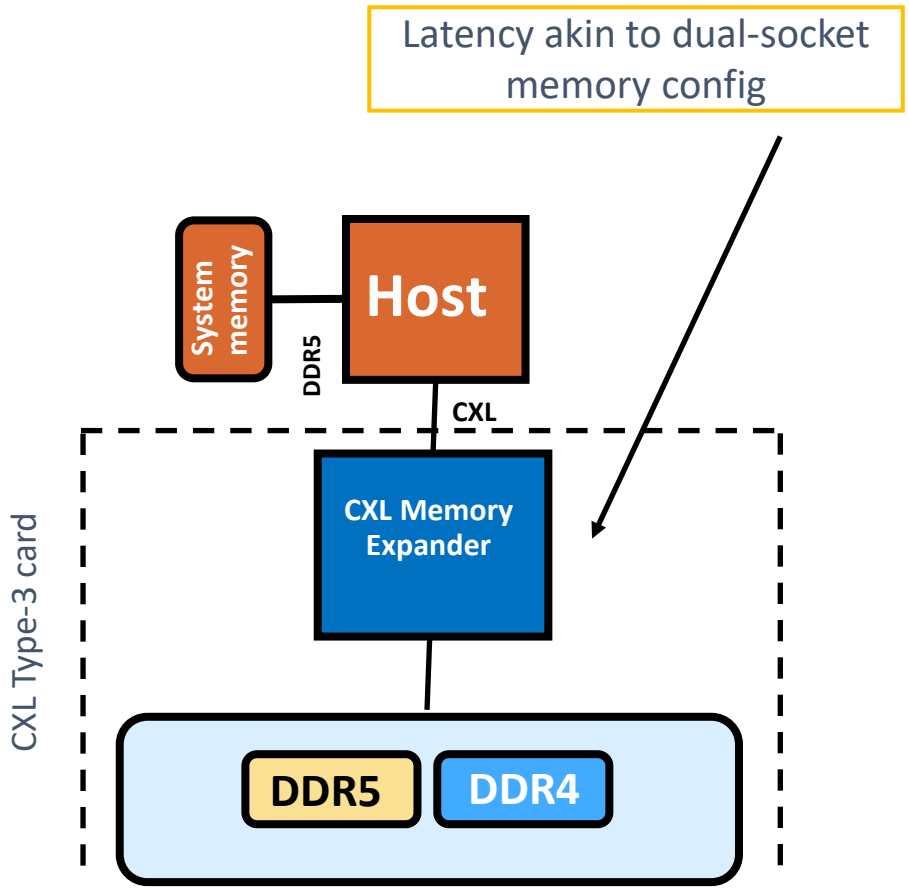
Memory per socket	Socket-attached DIMMs only	With CXL	Memory TCO Savings*
1TB	8x 128GB (1DPC)	8x 64GB + 8x 64GB on CXL	**24%**
2TB	16x 128GB	16x 64GB + 16x 64GB on CXL	**27%**
4TB	16x 256GB	16x 128GB + 16x 128GB on CXL	**39%**

*TCO savings based on Intel modeling using projected DIMM and CXL pricing for 2025

*TCO savings even more if DDR4 RDIMMs are reused using AICs

Use CXL-attached memory to achieve high system memory capacity
Avoid expensive high-capacity DIMMs

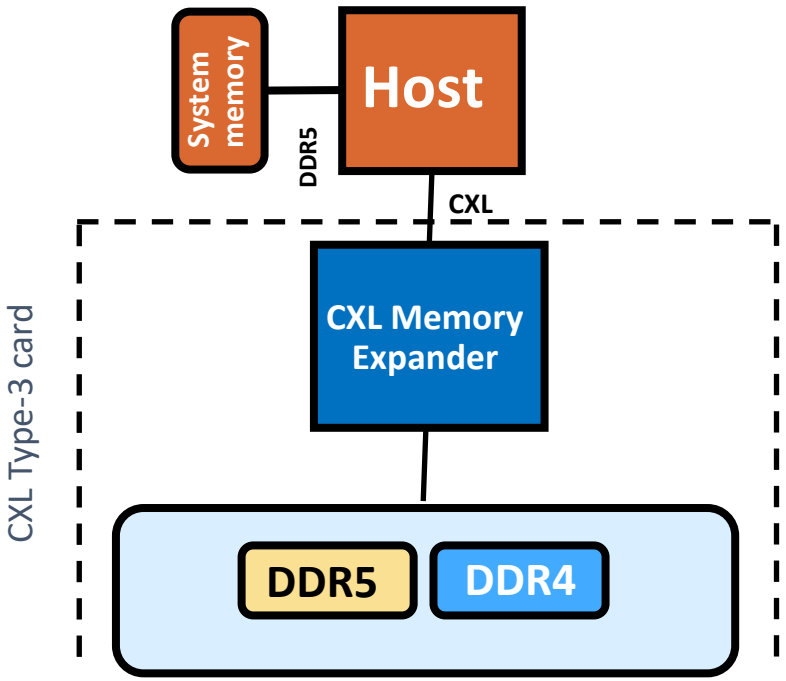
Memory Expansion: S/W-Controlled Tiering

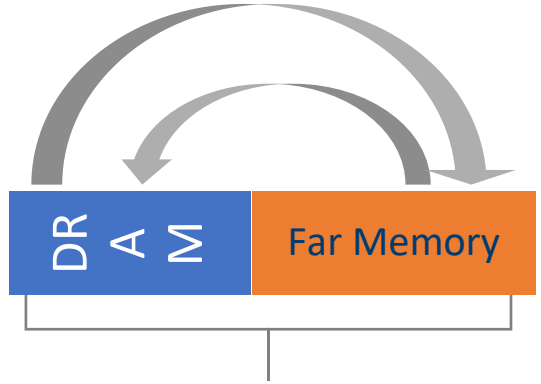


- **Memory presented as two tiers:**
- Main Memory: Lowest latency (‘Near Memory’)
- CXL Memory: Higher latency (‘Far Memory’)
- **How to hide CXL-memory latency?**
- Hot/Cold page tracking telemetry
- Move ‘hot’ data from Far to Near memory
- Move ‘cold’ data from Near to Far memory
- **Net result**
  - Application ‘thinks’ it is running in main memory
  - Minimal latency related perf loss

Workloads with good spatial/temporal locality of execution will benefit

H/W-Controlled CXL Memory Expansion: Intel Flat Memory Mode

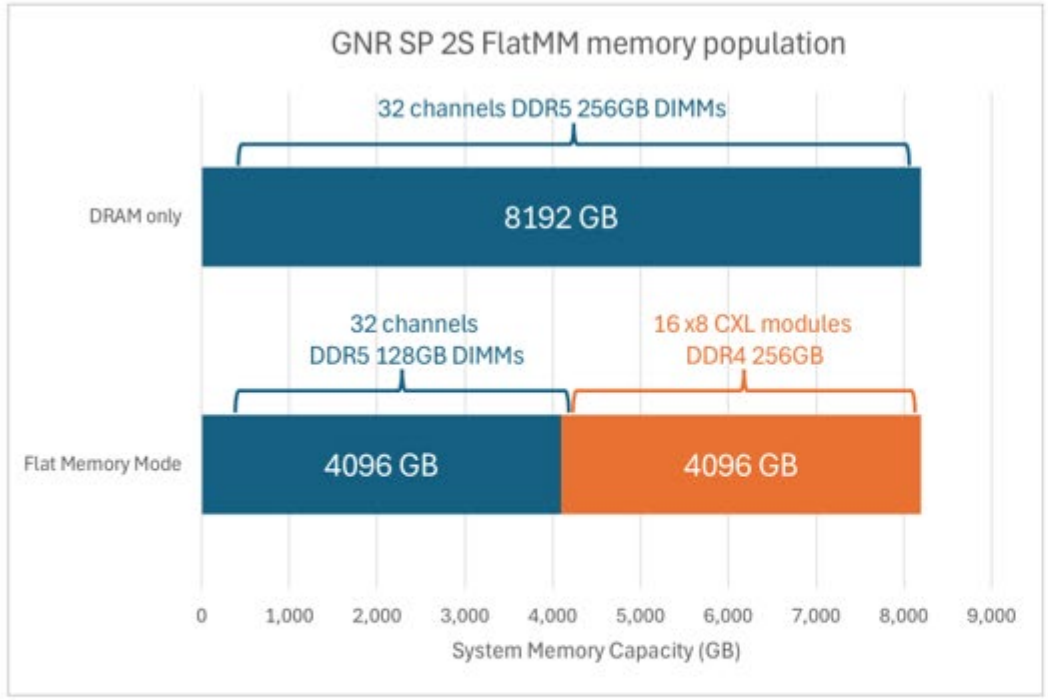


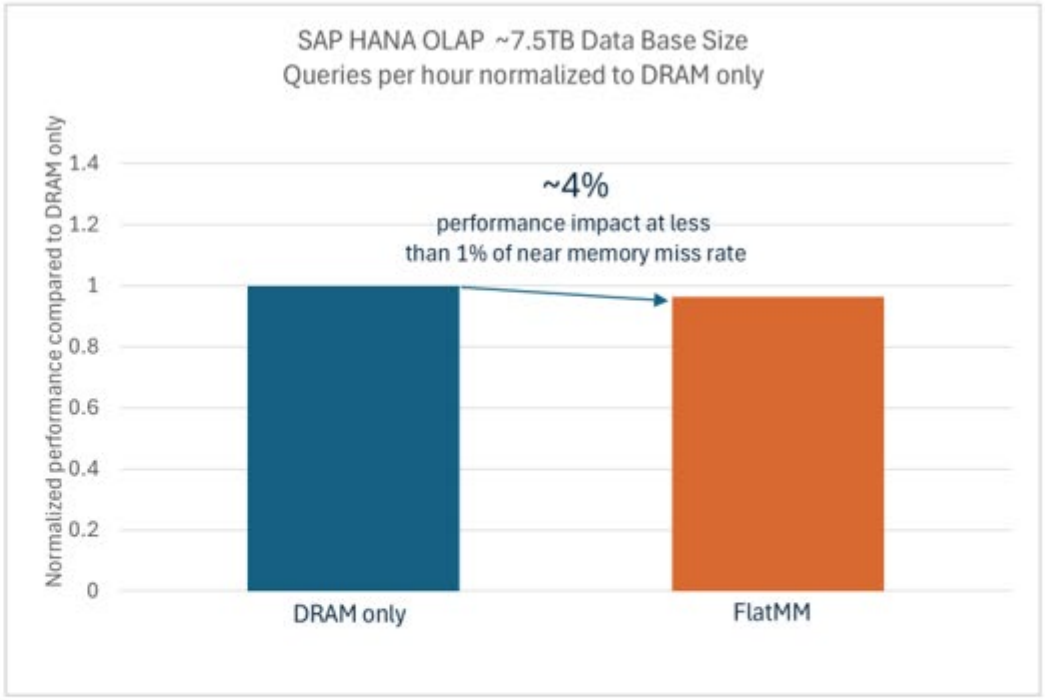


Advantage of Intel Flat Memory Mode over S/w-controlled Tiering

- Single NUMA node – completely transparent to Workloads
- No dependency on O/S version
- Lower Latency Data transfers: Cacheline granularity v/s Pages (4K+)
- No dedicated CPU cores required for Hot/Cold page tracking

Intel Flat Memory Mode Performance Example: SAP HANA running OLAP;





40%+ TCO Savings with DDR4 RDIMM Reuse

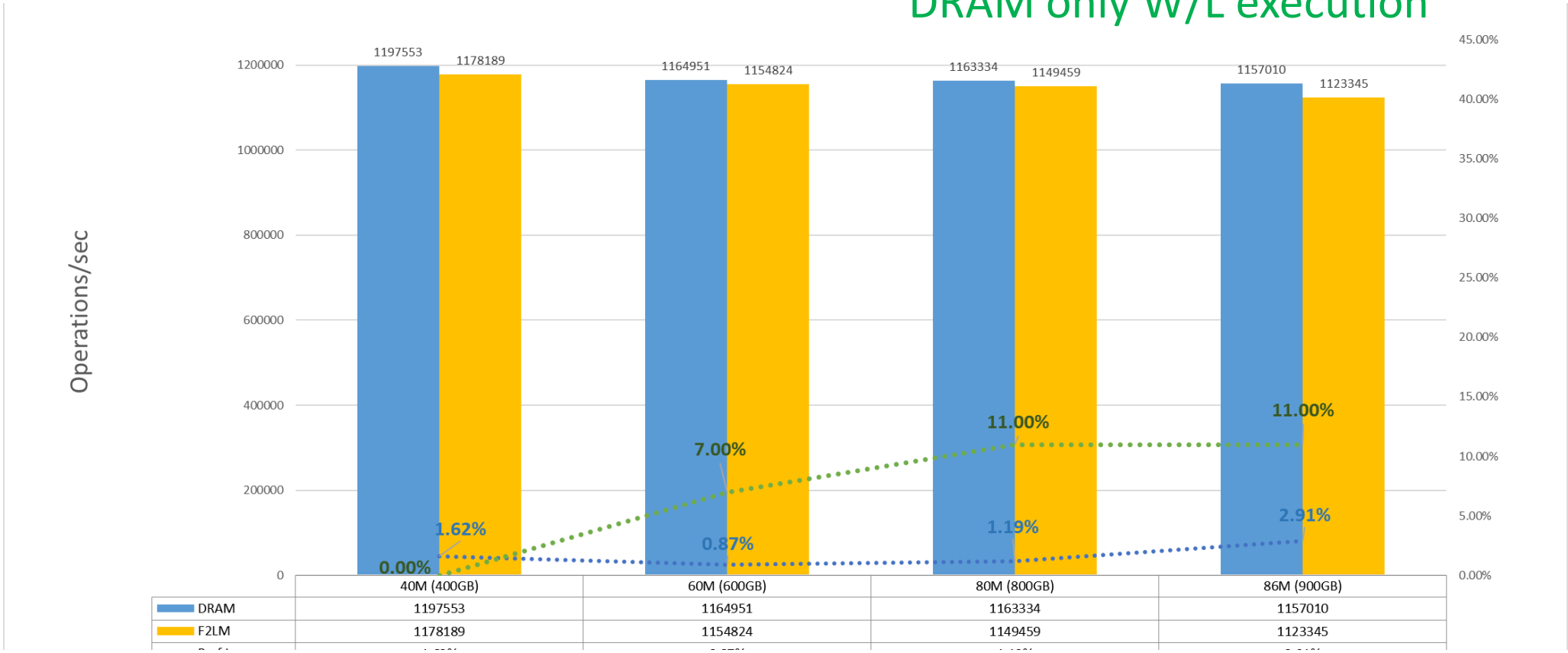
Endorsement by SAP Inc. / Microsoft Azure

https://sapinsider.org/how-intel-xeon-6-and-cxl-are-the-new-secret-weapons-to-tame-sap-data/

- Intel Xeon 6787P w/86c
- CXL Memory: Montage MXC,

Intel Flat Memory Mode Performance Example: Mongo DB

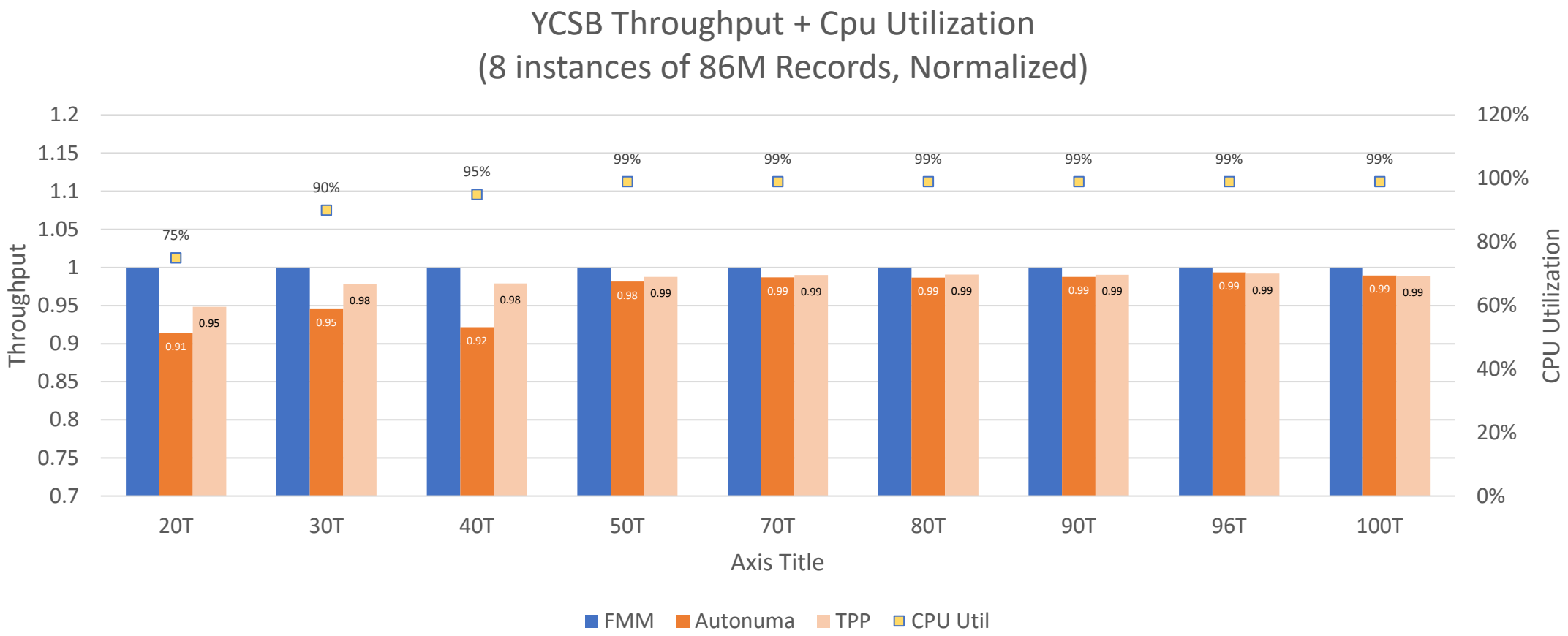
Estimated 27.5% TCO savings compared to DRAM only W/L execution



Less than 3% Perf degradation biggest Dbase footprint (900+GB)

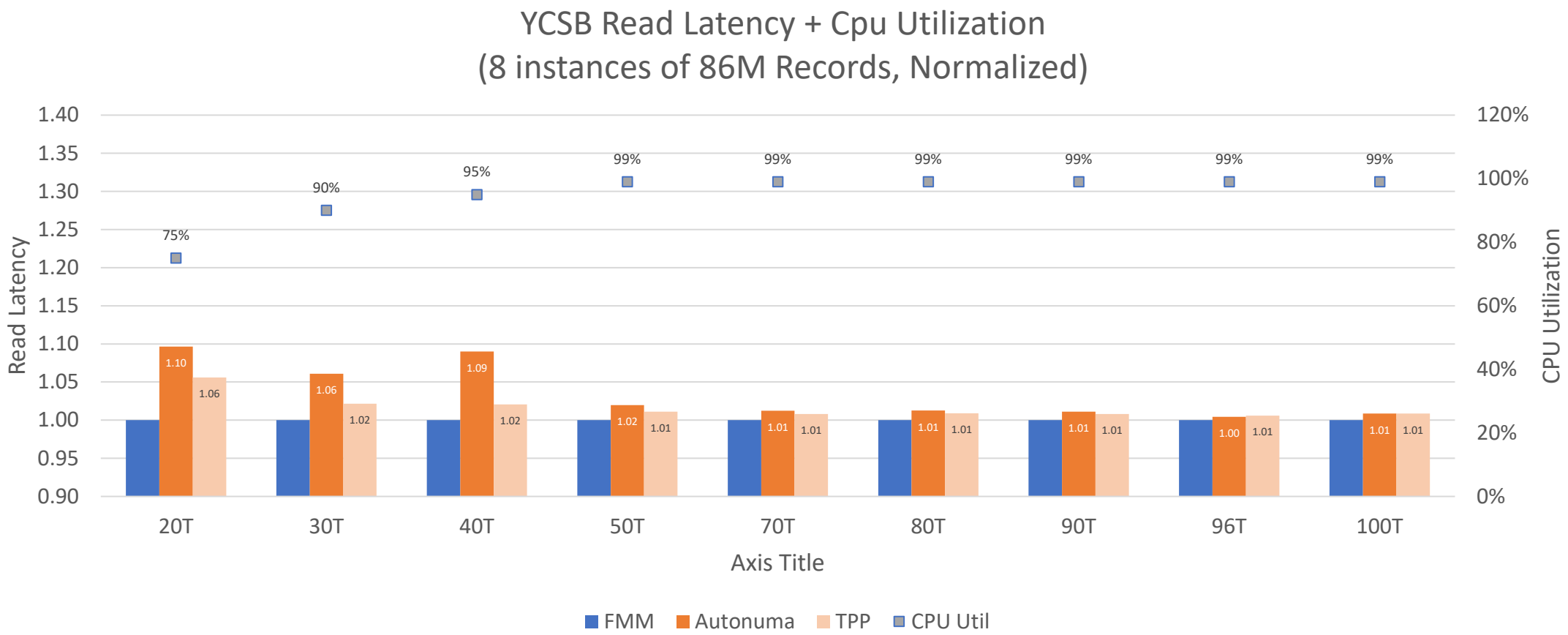
- Intel Xeon 6767P w/64c
- Hynix DDR5 128GB, 64GB DIMMs
- CXL Memory: Astera Leo A1000,

Intel Flat Memory Mode Performance: Comparison with S/W-controlled Tiering



Up to 10% Better Query Throughput under optimal CPU utilization

Intel Flat Memory Mode Performance: Comparison with S/W-controlled Tiering

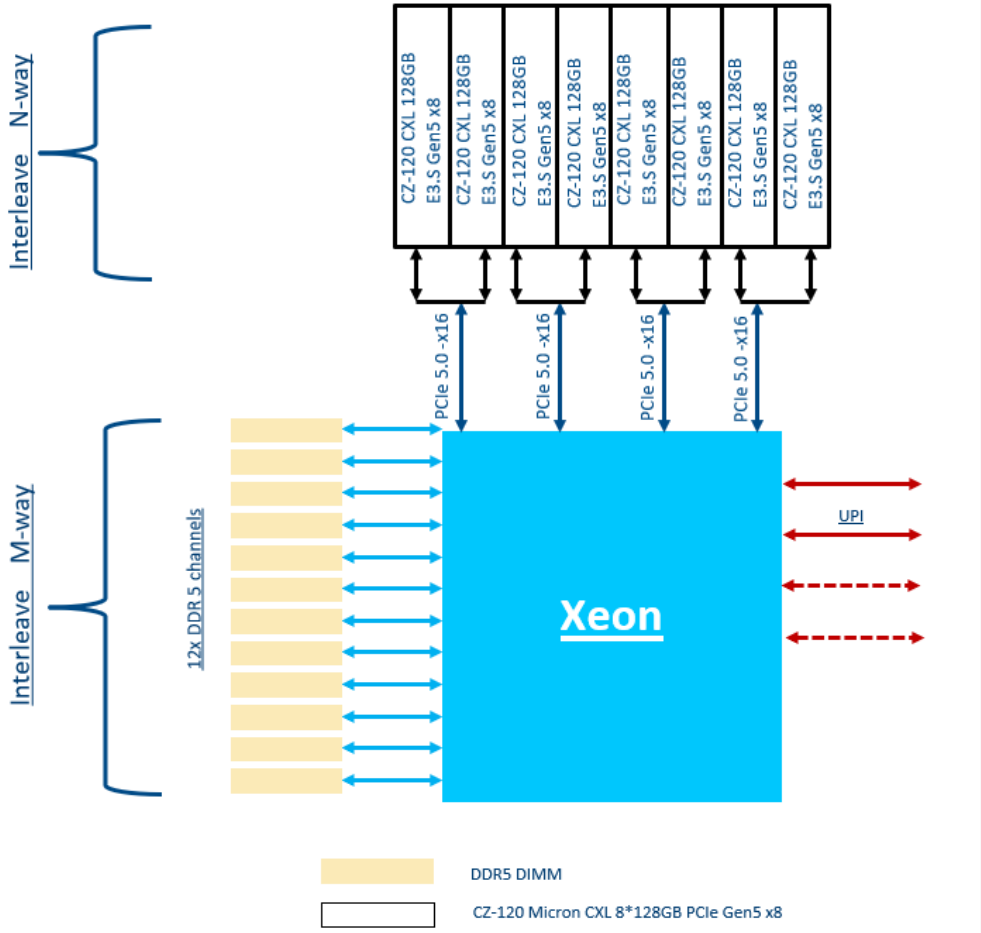


Up to 10% Lower Latency under optimal CPU utilization

B/W Intensive Database W/Ls

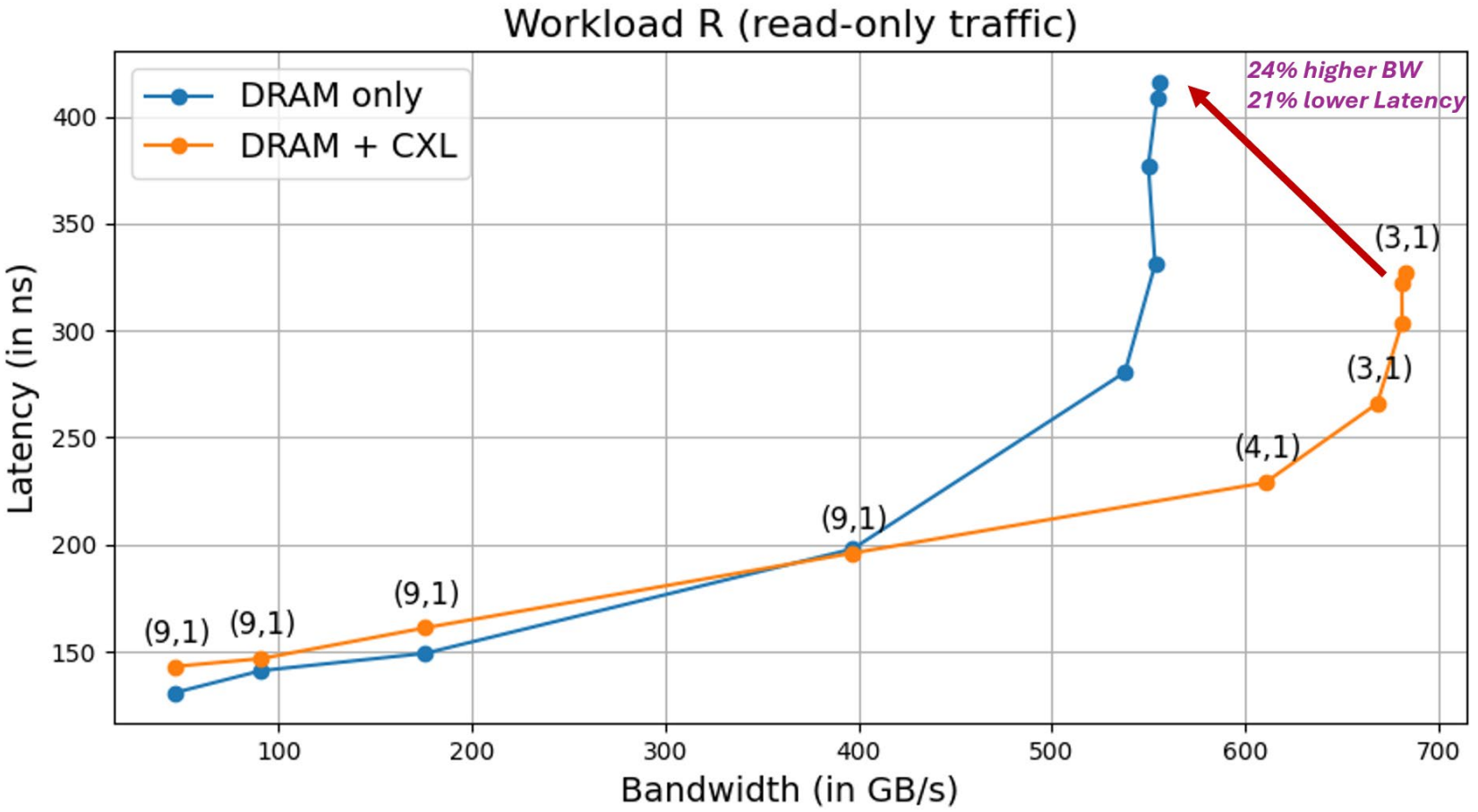
CXL For Memory B/W Expansion: B/W-Weighted Memory Interleaving

- System boots up two-tier memory (Near & Far)
- O/S uses Page-table entries to ‘stripe’ pages between Near & Far memory
- Stripping Ratio (M:N) decided based on B/w of respective tiers
- No page movement involved
- Memory presented to W/Ls as a single NUMA tier



https://community.intel.com/t5/Blogs/Tech-Innovation/Data-Center/Improve-your-HPC-and-AI-workload-performance-by-increasing/post/1647882

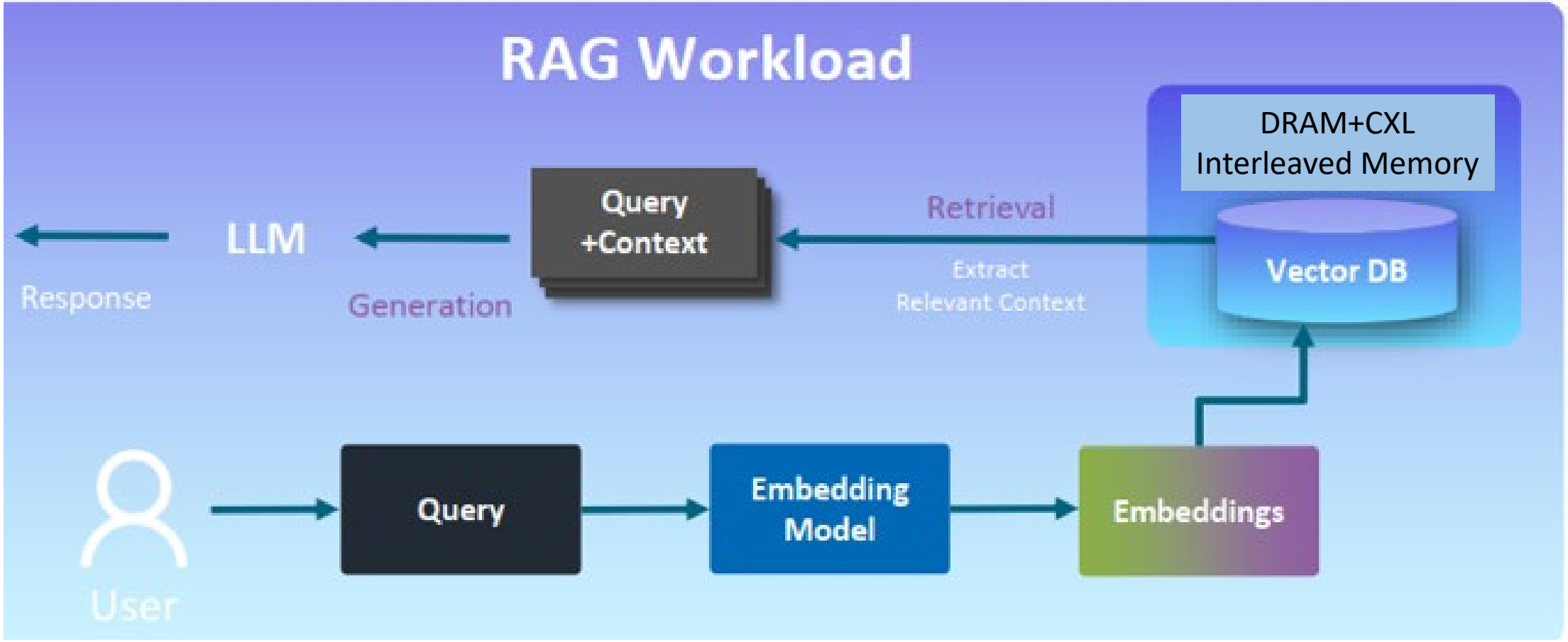
B/W Expansion with DDR5 + CXL Memory



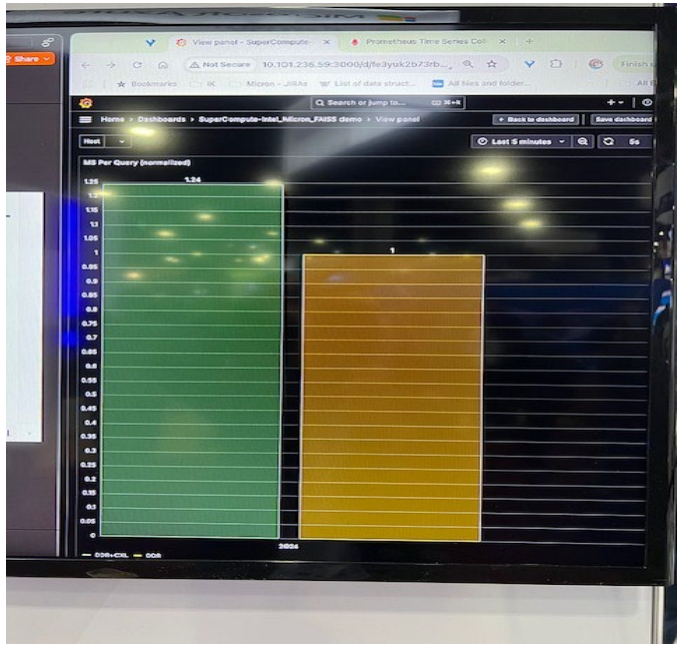
Interleaving weights shown by (M,N) pairs

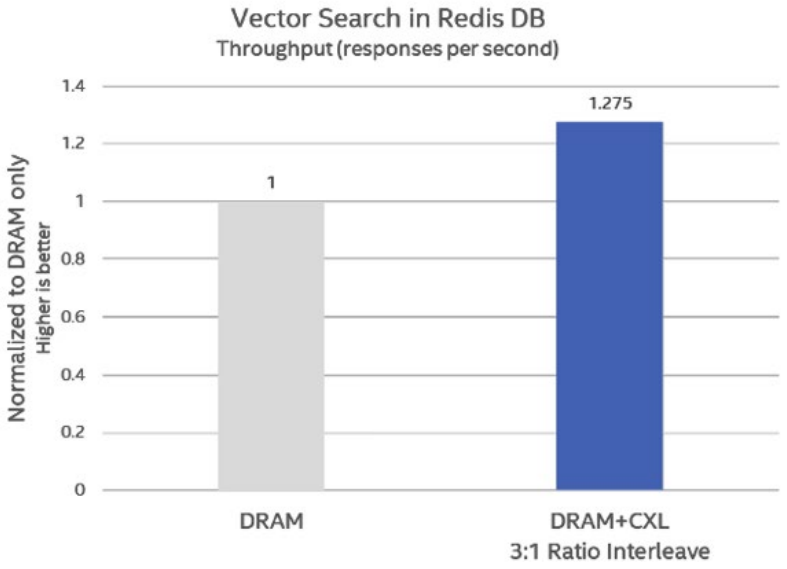
Chart with Intel Xeon 6900 with Micron DDR5 DIMMs & CXL CZ-120 modules

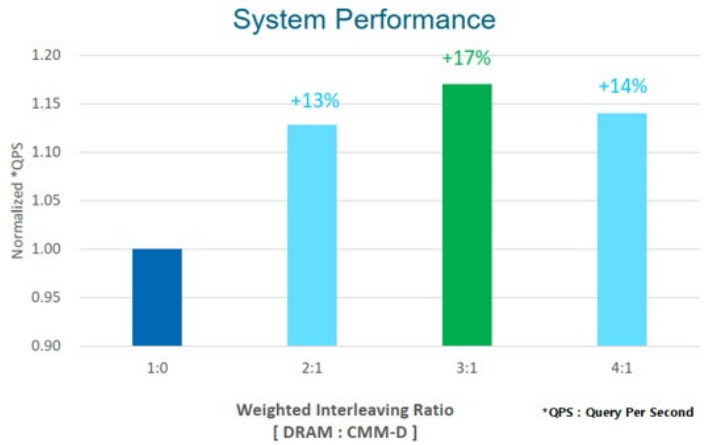
RAG (Retrieval Augmented Generation)



In-Memory RAG Database Performance







23%+ Perf gain for FAISS Vector Db
CPU: Intel Xeon-6 6900 w/ 128 cores
CXL Memory: Micron CZ122 128GB

27%+ Perf gain for Redis Vector Db
CPU: Intel Xeon-6 6787P w/ 86 cores
CXL Memory: 2x x16 Astera Leo D5

10-17%% Put Perf gain for Milvus DB
CPU: Intel Xeon-6 6767 w/ 64 cores
CXL Memory: Samsung CMM-D 256GB

Summary

- Memory Intensive W/Ls like In-memory Databases can benefit significantly with CXL Memory
- For both Memory Capacity expansion & Bandwidth expansion use cases
- Intel Flat Memory Mode has ease of us, performance & TCO advantages over S/w-based tiering techniques
- CXL Device/CPU Ecosystem has significantly matured on recent years

Call-to-Action

- Try out CXL memory for your In-memory Database applications
- Check out CXL Flat Memory Mode on your Intel Xeon CPUs
- Accelerate your AI RAG database query lookup using Intlv-memory tech.