Flash Memory Summit

Memory Tiering with CXL

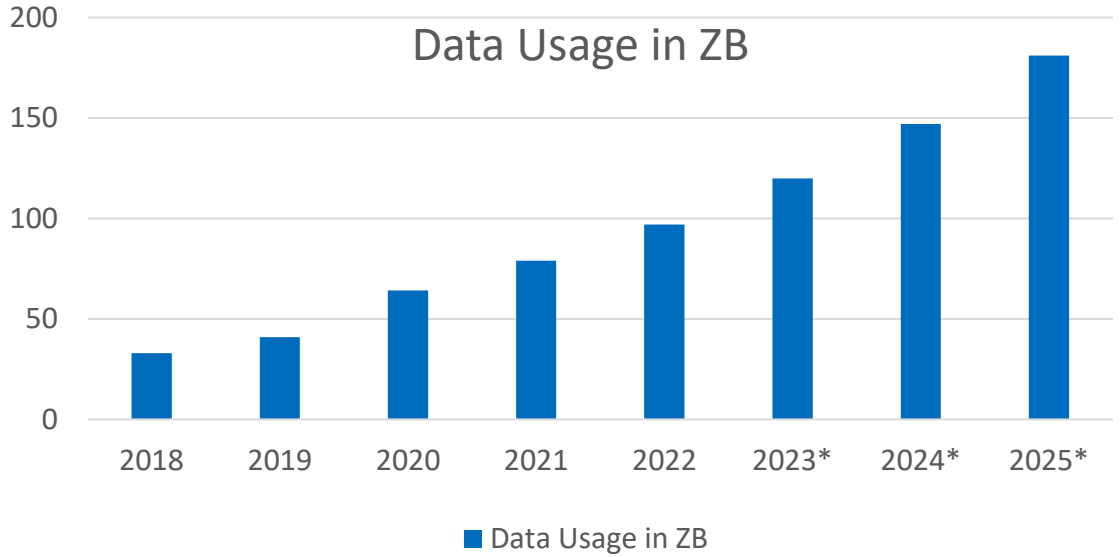
Ravi Kiran Gummaluri

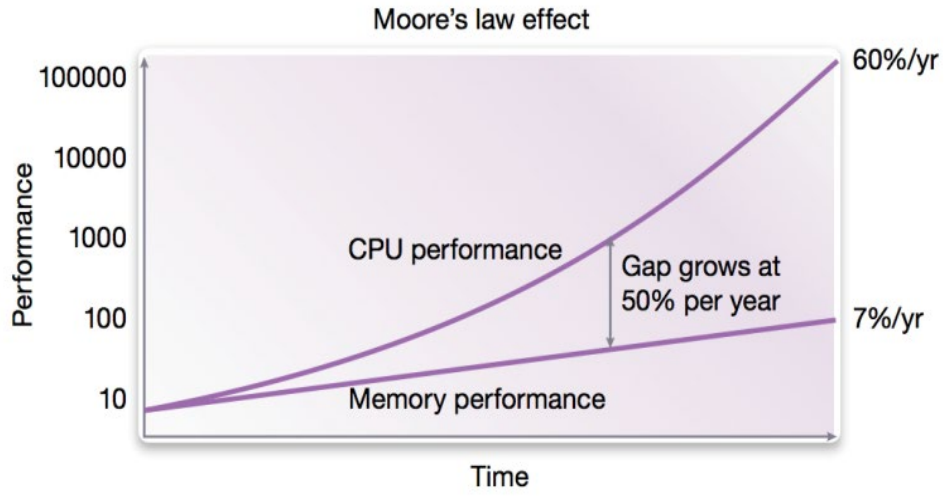
35 YEARS OF NAND FLASH MEMORY INNOVATION

Appendix

- Memory Demand and scaling challenges
- NUMA architecture
- CXL expansion memory
- CXL memory usage model
- Memory Tiering : Kernel managed
- Memory Tiering : Application/service managed
- Next steps

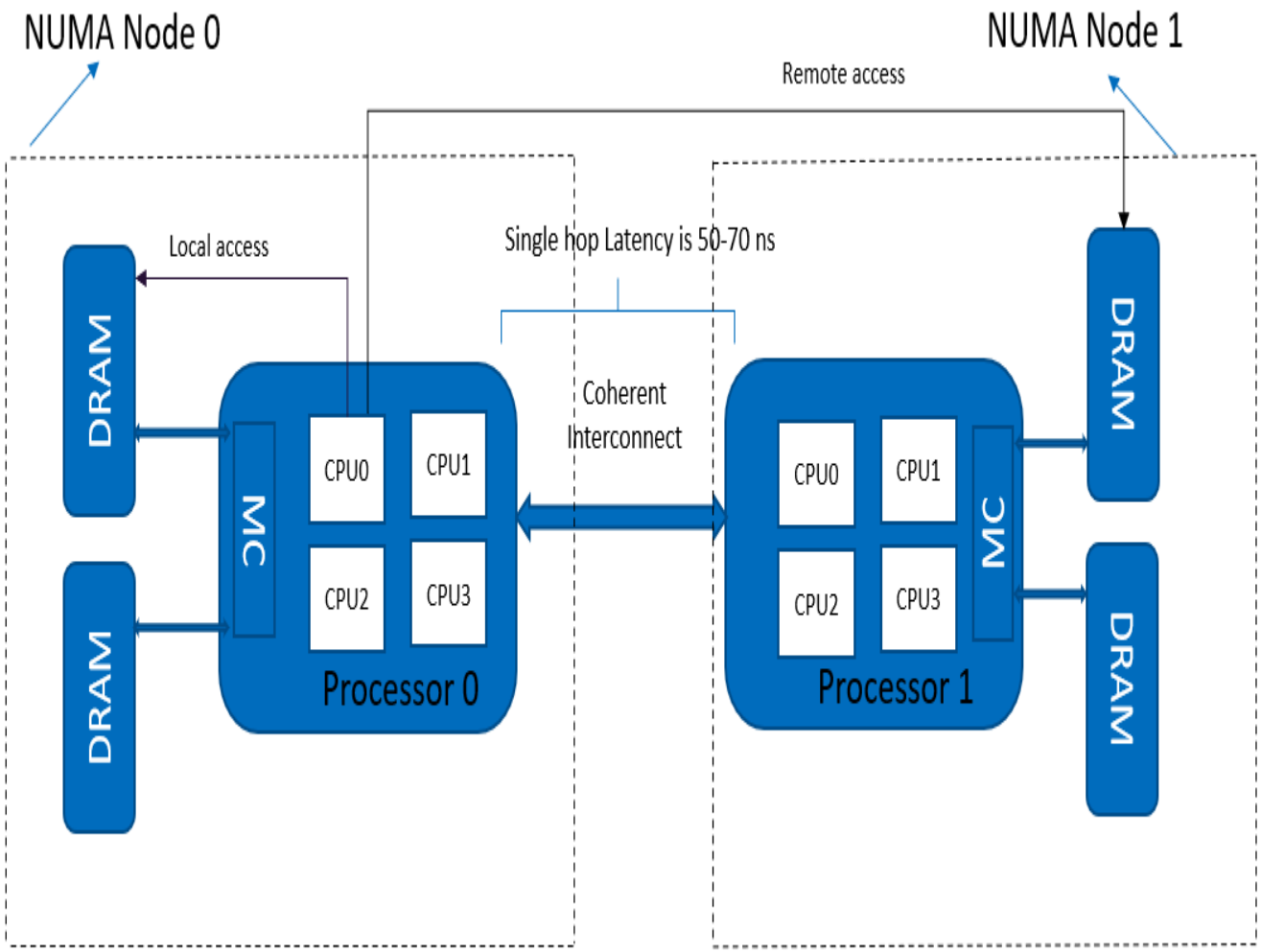
Memory Demand and Scaling challenges





- Growing demand for Memory need in data center applications . -> 26% y/y
- DRAM is not scaling -> Memory Capacity is doubling every four years
- Processor speed -> has been doubling every two years.
- Memory Latency -> is only improving 1.1 times every two years.
- How do we solve increased memory BW and capacity requirements ?

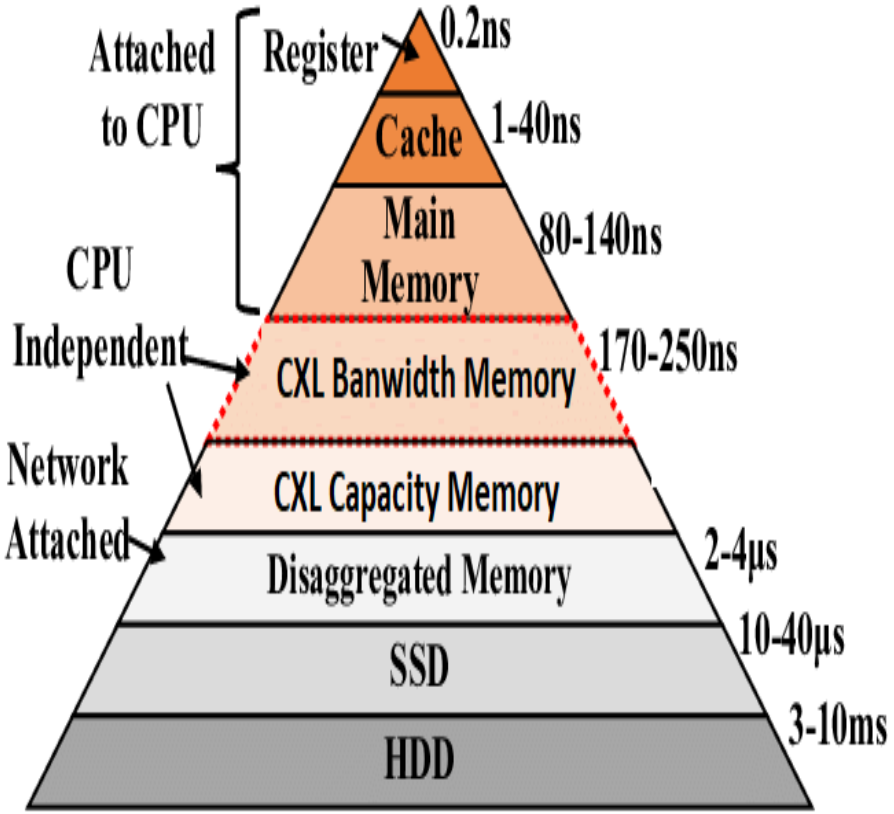
NUMA Architecture and platforms



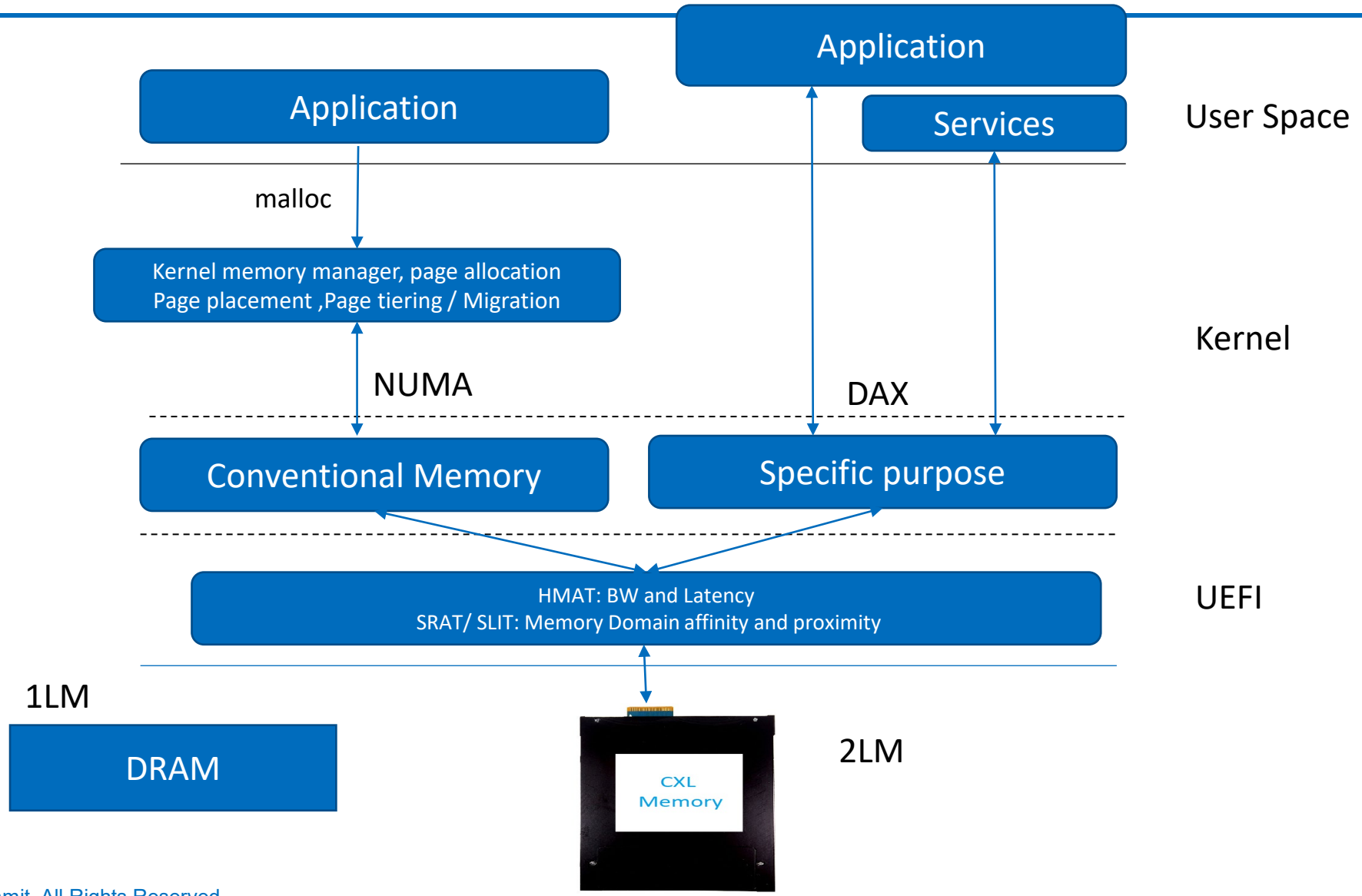
- Multiprocessor systems whose memory is divided into multiple memory nodes to share across system needs .
- Cache Coherent NUMA allows access to memory on other socket for BW and capacity requirements.
- Cons:
- NUMA Arch is tightly coupled with CPU and requires additional sockets to scale memory requirement .
- Existing page management is designed for homogenous DRAM only NUMA node.
- NUMA balancing policies are not effective in reducing access latency by page migration .
- CXL Expansion memory will address these issues with memory tiering .

CXL expansion memory

- CXL expansion memory can provides Latency and BW like 1-hop Remote NUMA.
- Cache-line granular access semantics
- CXL-Memory appears to a system as a CPU-less NUMA node. (Not dependent on CPU Arch)
- Hot Pluggable memory
- Works with various form factors E.1S,E3.S , Add on Card etc
- Interoperable with various memory types (DDR5, LPDDR5, NVM)
- CXL BW memory tier : Move cold pages ,high performance OLTP transaction processing
- CXL Capacity Memory tier : Caching and ML analytics model tasks that need to allocate large amounts of memory but are much less sensitive to bandwidth or latency performance

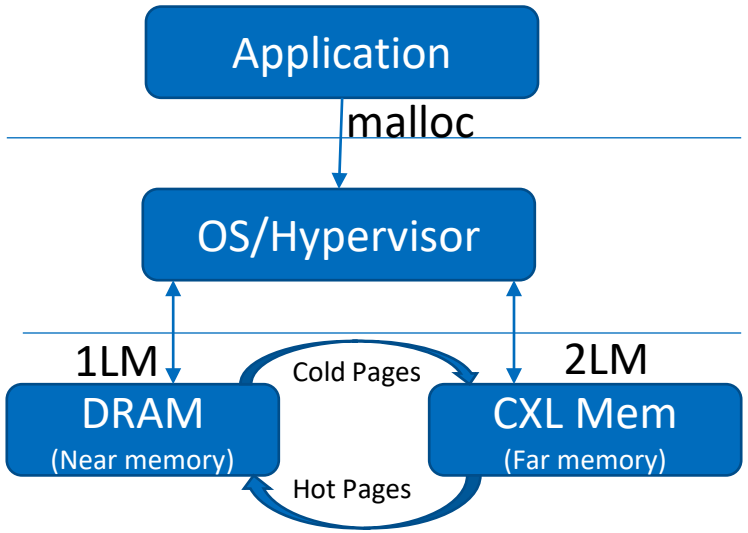


CXL memory usage model



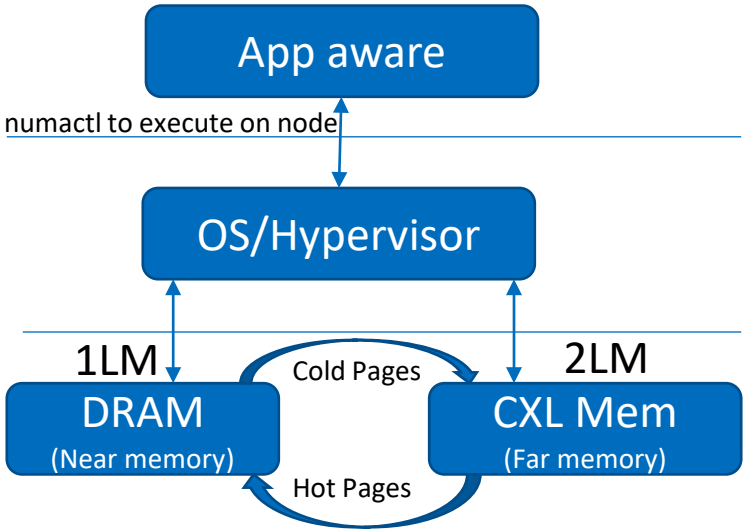
Memory Tiering : Kernel Managed & App Transparent

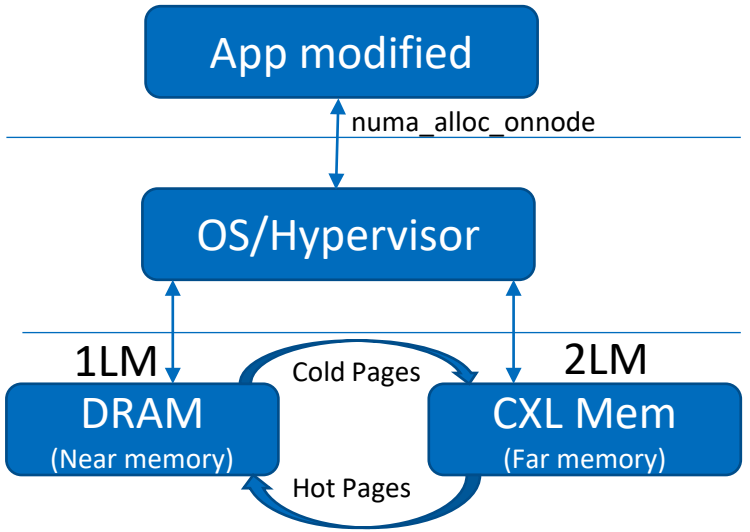
- Applications are not required to modify
- Memory topology is abstracted from user
- Memory placement is controller at kernel .
- SW assisted page migration with improved policies
  - » Memory access sampling , profiling .
  - » Page placement policies with known media characteristics
  - » HW/DMA engine support to migrate pages between different tiers
  - » In virtualized environment additional care need to be take for IOMMU management .



Memory Tiering : Kernel Managed ,App aware/ App modified NUMA

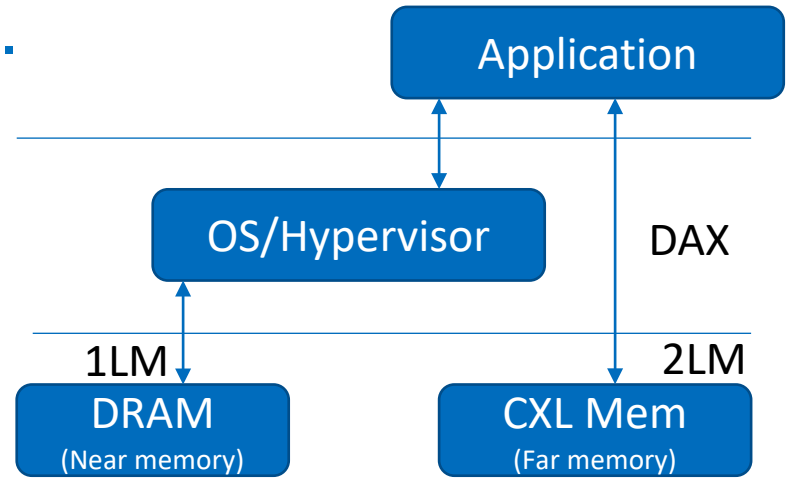
- Applications are aware of NUMA domains
- Application can execute in given NUMA node.
- Memory allocation is requested by application in given NUMA.
- Kernel manages memory CXL memory.





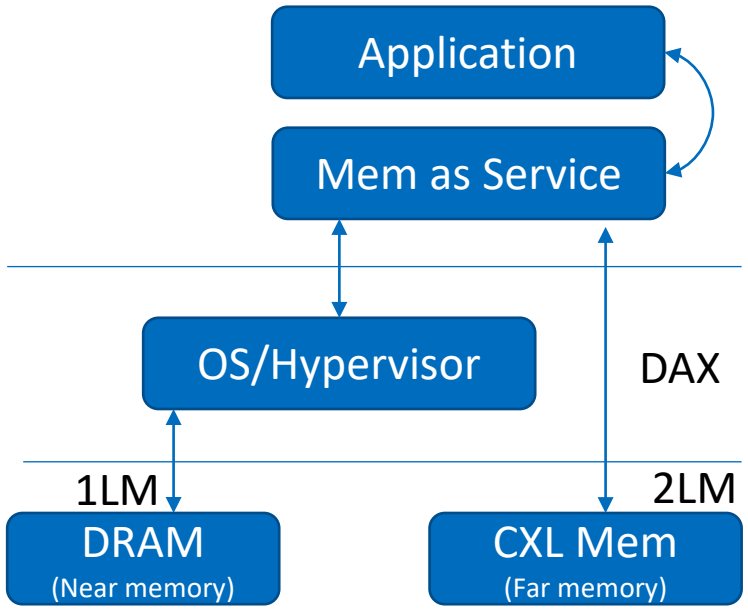
Memory Tiering : Application managed

- CXL memory tier is exposed to application as device direct access file (DevDAX).
- Application is modified to make use of CXL memory allocations in optimized way.
- Memory topology is visible to application.
- Application manages CXL memory .



Memory Tiering : Memory service managed

- Application can use memory service as proxy for allocations.
- Memory topology is presented to memory service .
- Memory allocation and placement controller by memory service .



Next Steps

- CXL memory tiering can solve increasing memory bandwidth and capacity requirements .
- Better page migration and placement policies need to evolve
- Standardization of memory-to-memory data movement
- Work need to be done on virtualized environment and security .
- Industry wide efforts need to be driven on workload analysis with memory tiering.
- Emerging Storage class memory technology need to be evolved for various memory usage models .

