Cloud-Scale Deployment with CXL Memory

MEMORY FABRIC FORUM

OCP GLOBAL SUMMIT

OCT 15-17, 2024
SAN JOSE, CA

Cloud-Scale Deployment with CXL Memory

Samir Rajadnya, Microsoft

Ahmad Danesh, Astera Labs

Application-Specific Memory Requirements

- Every application has different memory requirements
- General Purpose Compute must accommodate all requirements
  - Example: in-memory database requires max capacity with moderate BW/latency

	General Purpose Compute (CPU Centric Architecture))	Example Application (In-memory database)
Latency	1-3	2
Bandwidth	1-3	3
Capacity	1-3	1
Cost	1	1

1 = Highest Priority

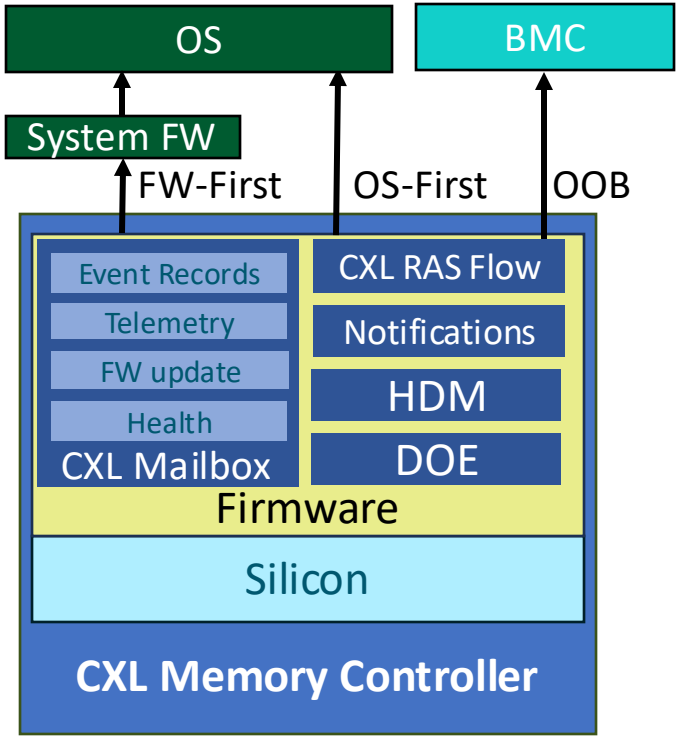
General Purpose Compute Architecture

- **Problem Statements**
  - As CPU cores increase, more memory is required (maintain core:memory ratios)
  - Memory is not scaling at the same rate as CPU cores
  - Higher capacity DIMMs (3DS) cost ~2x of lower capacity DIMMs
  - Sometimes we need more capacity than possible through locally attached
  - Long term, CPUs may move away from multi-socket architectures
- **Solution**
  - CXL provides cost-effective and performant solution to expand memory capacity
  - CXL memory makes it possible to reuse DIMMs
  - Advanced CXL features such as pooled/shared memory enable new functionalities not possible with locally attached memory

CXL Solution Requirements

CXL Solution Requirements

- Application Performance
- RAS
- Security
- In-band System Management and Telemetry
- Out-of-band System Management and Telemetry
- CPU Interoperability
- DIMM Interoperability



Platform Solution Stack

- Application
- Management Tools
- OS/Kernel/BMC
- Device drivers
- System
- Firmware
- Silicon

Holistic Cloud-Scale CXL Memory Solution

CXL Solution Requirements

- Application Performance
- RAS
- Security
- In-band System Management and Telemetry
- Out-of-band System Management and Telemetry
- CPU Interoperability
- DIMM Interoperability

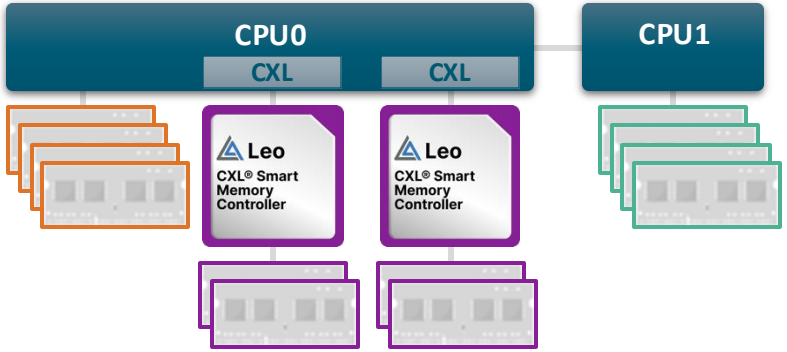
AsteraLabs.

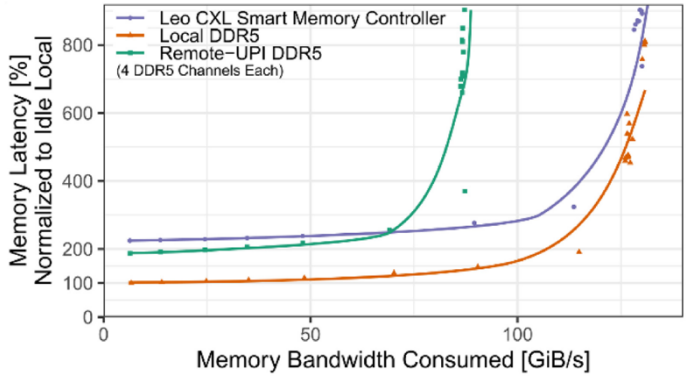
Leo
CXL® Smart Memory Controller

Platform Solution Stack

- Application
- COSMOS Platform APIs
- OS/Kernel/BMC
- Device drivers
- Systems
- COSMOS Embedded Software
- Silicon

Bandwidth and Latency Considerations





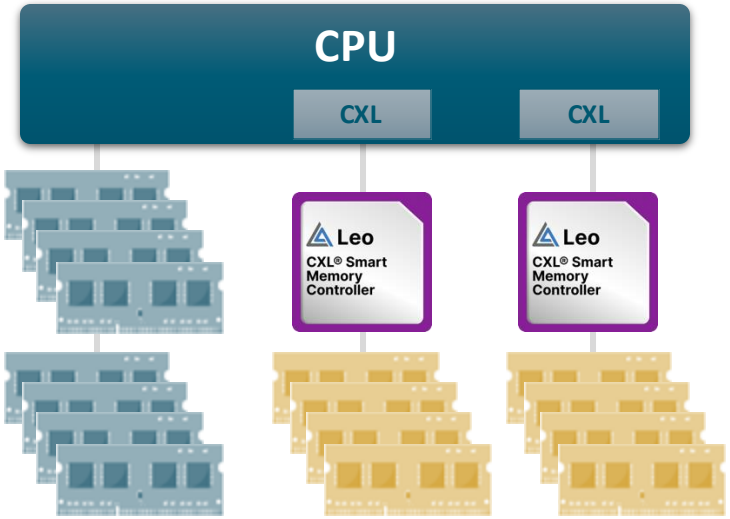
Data from Astera Labs and Intel
https://dl.acm.org/doi/10.1145/3669900

Performance Observations

- Unloaded latency: CXL delivers ~same latency as remote and ~2x local
- Loaded latency: CXL delivers smooth latency/bandwidth response
- Bandwidth: CXL delivers much higher bandwidth than remote; ~same as local

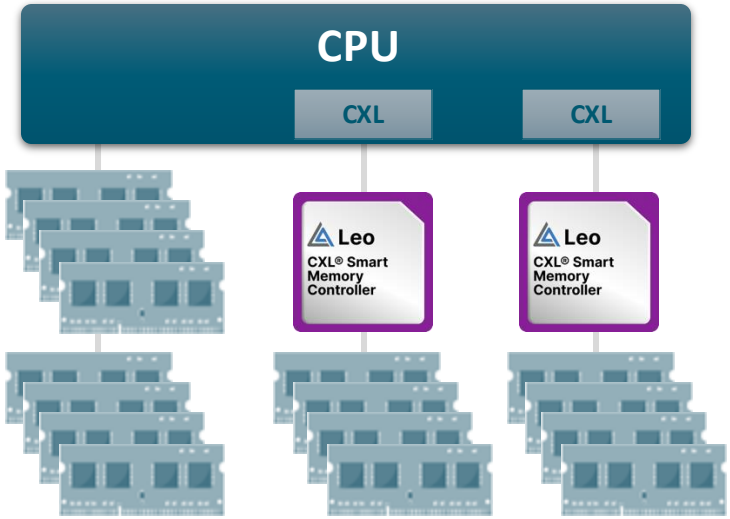
How to Utilize CXL-Attached Memory

Tiering



- Separate NUMA nodes
- Local (hot) + CXL (warm) tiers

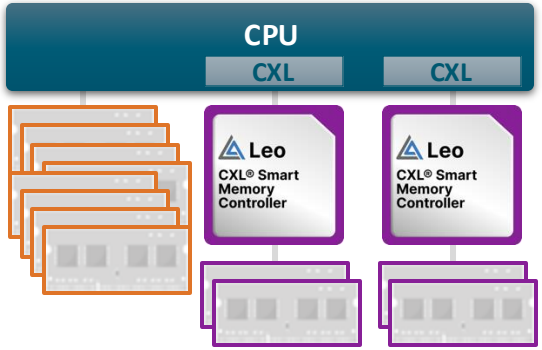
Interleaving



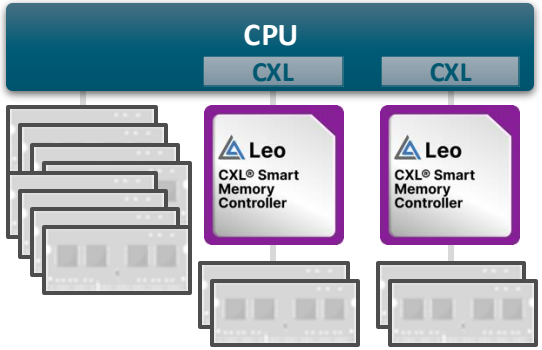
- Single NUMA node
- Data striped across local and CXL

Tiering and Interleaving - Performance Comparison

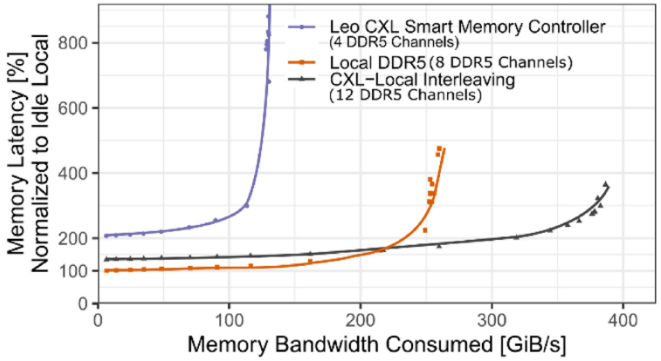
Tiering Setup



Interleaving Setup



Performance Measurements



Data from Astera Labs and Intel
https://dl.acm.org/doi/10.1145/3669900

CXL Performance Observations

- Interleaving delivers higher aggregate bandwidth with lower loaded latency
- Tiering enables flexibility to optimize local and CXL memory access

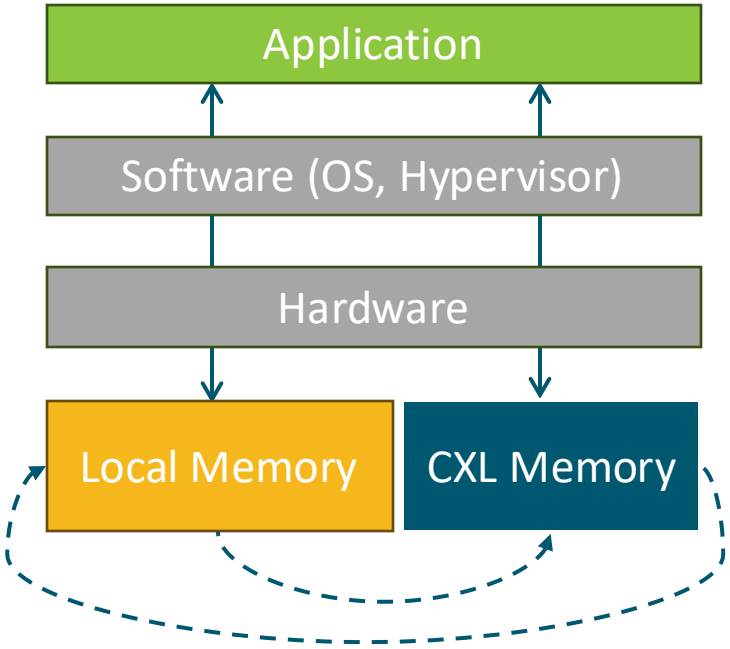
Question: How does this translate to application performance?

- Depends on the application and there's a lot you can optimize

Tiered Memory Deployment Options

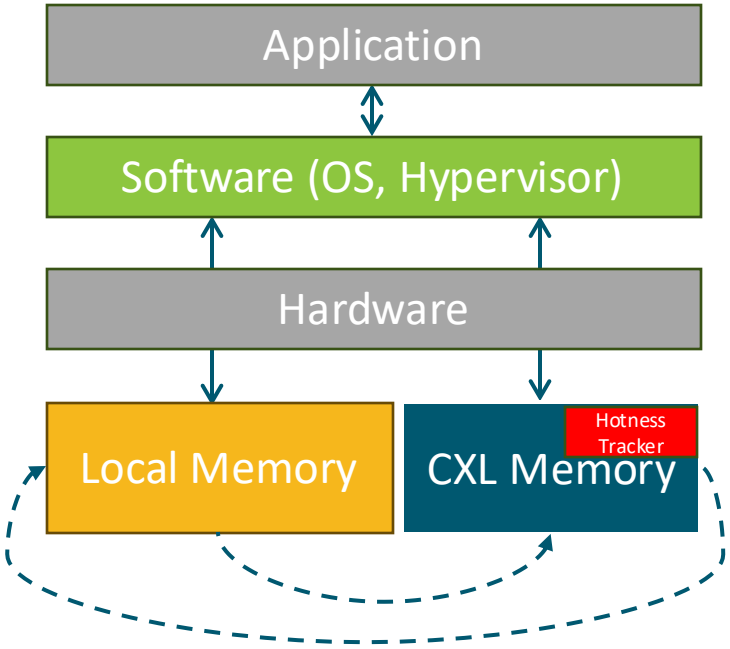
1. Application-managed
   - Application software can be modified to use two memory tiers
   - CXL memory is separately visible to application as ZNUMA (zero core NUMA)
   - Application knows latency sensitivity of different objects and places them accordingly
2. Software-managed
   - Application does not need to be modified
   - SW while working with hot page tracker on CXL controller moves hot pages from CXL tier to local tier.
3. Hardware-managed
   - HW features that moves 64B cache lines from CXL to local memory: Flat memory mode

Case 1: Application-Managed Memory Tiering



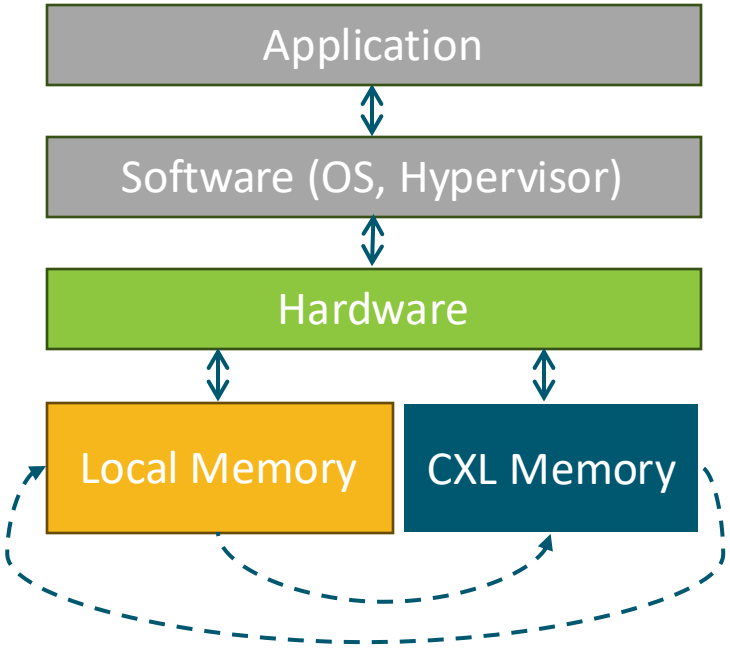
- Application sees two memory tiers
- Application is modified to utilize two tiers
- Application is in control of data placement across two tiers
- Application can promote or demote data between these two tiers depending on hotness of data

Case 2: Software-Managed Memory Tiering



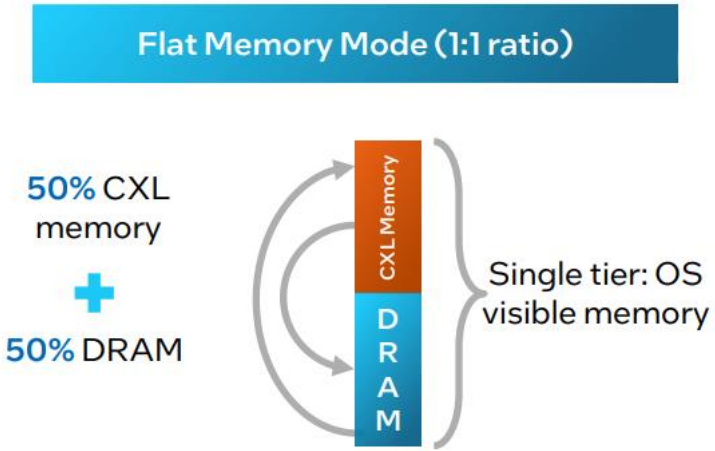
- Application sees single memory tier
- HW (hotness tracker inside CXL controller) tracks hot memory regions on a far CXL memory
- HW provides this information over software interface to OS or guest application
- OS is responsible for migrating hot pages from CXL memory to local memory

Case 3: Hardware-Managed Memory Tiering



HW managed tiering with Intel Flat Memory Mode

- Cache line granular movement
- Hot lines stay in lower latency memory



PowerPoint Presentation (hotchips.org)

Call to Action

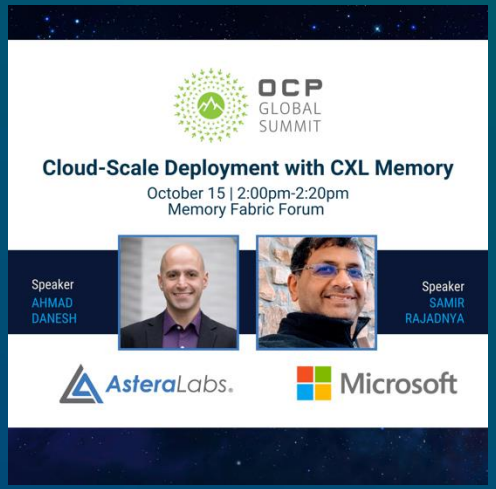
Summary

- CXL provides cost-effective and performant solution to expand memory capacity
- CXL is ready for cloud-scale deployment with multiple deployment options
- Each tiering and interleaving mode have unique performance advantages
- Application-specific performance can be tuned through different tiering modes

Call to Action

- Consider all memory deployment options to optimize application-level performance
- Collaborate in OCP CMS group (e.g., hotness tracking, pooling/sharing, compression…)
- Visit Astera Labs Booth B13 to learn more

Thank you!


OCP GLOBAL SUMMIT
Cloud-Scale Deployment with CXL Memory
October 15 | 2:00pm-2:20pm
Memory Fabric Forum
Speaker
AHMAD
DANESH
Speaker
SAMIR
RAJADNYA
AsteraLabs.
Microsoft


OCP GLOBAL SUMMIT
Optimizing AI Inferencing with CXL Memory
October 17 | 1:10-1:30pm
Speaker
MICHAEL
OCAMPO
Speaker
AHMED
MEDHIOUB
AsteraLabs.


OCP GLOBAL SUMMIT
Launching CXL Memory in Hyperscale Deployments with a Holistic CXL Ecosystem
October 17 | 2:45-3:05pm
Speaker
CHRIS
PETERSEN
Speaker
PRAKASH
CHAUHAN
AsteraLabs.
Meta