CXL Memory Tiering for heterogenous computing

Ravi Kiran Gummaluri & John Groves
Micron Technology
Agenda

- Memory demand and scaling challenges
- CXL memory expansion
- HW based Heterogenous Interleave
- SW + HW Heterogenous Interleave
- SW based Heterogenous Interleave
- Enabling SW interleaving
- Next Steps / Call for action
Growing demand -> for memory in data center applications (~26 % yoy).
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?

Figure 1: Growing memory usage
Figure 2: Memory wall
Figure 3: Memory capacity Vs CPU cores

Figure 1 : Source: https://www.statista.com/statistics/871513/worldwide-data-created/
Figure 3 : Source: Based on capacity and core counts from publicly available AMD and Intel datasheets, and public statements.
CXL Memory Expansion
- 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 E3.S, AIC, PMM etc.
- Interoperable with various memory types. (DDR4/DDR5/LPDDR/ NVM ..)

CXL Memory Capacity Expansion
- CXL Direct attached Memory Tiering
  1. Application Transparent
     - OS Managed
     - User Space Library
  2. Application Managed
     - Application Aware (ex: libnuma)
     - Modified (ex: libmemkind)
- CXL Switch / Fabric attached Memory Tiering
  - Another Memory tier added to system with higher latencies.

CXL Memory Bandwidth Expansion
- CXL Heterogenous interleave solutions
  1. Hardware based Interleave
  2. Software and HW Interleave

Figure 4 : Memory Hierarchy
**HW Heterogeneous Interleave**

- System Address map will be interleaved between Local DRAM and CXL memory

**Pros**
- Easy to configure

**Cons**
- Kernel/OS cannot manage memory allocations.
  - Affects kernel memory.
  - Hides the NUMA topology from the OS.
- Fixed configuration : Not scalable for all workloads
  - Capacity expansion workloads will have higher latency
- CMM capacity will be restricted to align with Local DRAM capacity.

Figure 5 : HW Heterogenous interleave
HW + SW Heterogenous Interleave

- **HW**: Supports associating DRAM channels to different NUMA domains.
- **SW**: Interleave 4(Local):1(CXL) NUMA domain using `numactl`.
- **NPS4**: Each socket is partitioned into 4 NUMA domains. Each NUMA domain has 3 memory channels.

**Pros**
- NUMA topology is enabled.
- Kernel/OS can manage the memory allocations.
- Overcomes capacity limitations imposed by HW interleave solution.

**Cons**
- Fixed configuration: Not scalable for all workloads.

Figure 6: HW + SW 4:1 Interleave
SW Heterogenous Interleave

- Memory allocations performed according to per-node weights

Pros
- Scalable: Not fixed configuration
  - Application can configure different weights according to BW requirements.
  - This only applies when explicitly enabled for a job.
- NUMA topology is enabled.
- Kernel/OS can manage the memory allocations
- Overcomes capacity limitations imposed by HW interleave solution.

Cons
- Multiple approaches are proposed
  - Need Community support triage on one.

Figure 7: SW Interleave with weights
Our Journey: Enabling SW Interleaving

- Earlier attempts to enable SW Interleaving in 2022
  - Meta has released M:N patch to provide weight-based interleaving across 2 Tiers.
    - Needed enhancements for multiple tiers.
    - Same weights will be applied to all nodes belonging to a particular tier.

- Current contributions in 2023:
  - Micron/MemVerge released a series of RFCs that provide increasingly flexible weight-based tiering.
    - V1 MemTier based
    - V2 NUMA node based, per feedback
    - V3 cgroups memcg + numa node based (V4 will probably be s/memcg/cpuset/g per feedback)
45% bandwidth gain is observed using best interleave ratio
CXL can provide solutions for increased capacity and bandwidth requirements.

Current solutions (HW based) required various platform configurations to provide bandwidth and capacity solutions. Each Applications/Workload cannot be tuned for best performance.

SW interleaving can provide a single platform configuration for capacity and bandwidth solutions. Flexibility to tune individual application to provide best performance.

- Call for Action:
  - We request Linux community to see this patch set evolve into mainline. The benefits are significant to provide better system level solutions for bandwidth and capacity requirements.

- Various Articles and RFC work under progress
  - Article on weighted interleaving for memory tiering
    - [https://lwn.net/Articles/948037/](https://lwn.net/Articles/948037/)
  - Link for Memory Tier based Interleaving RFC
    - [https://patchwork.kernel.org/project/cxl/cover/20231009204259.875232-1-gregory.price@memverge.com/](https://patchwork.kernel.org/project/cxl/cover/20231009204259.875232-1-gregory.price@memverge.com/)
  - Link for Node based Interleaving RFC
    - [https://patchwork.kernel.org/project/cxl/cover/20231031003810.4532-1-gregory.price@memverge.com/](https://patchwork.kernel.org/project/cxl/cover/20231031003810.4532-1-gregory.price@memverge.com/)
  - Link for cgroups & node based interleaving RFC (getting there!):
    - [https://patchwork.kernel.org/project/cxl/cover/?series=799803](https://patchwork.kernel.org/project/cxl/cover/?series=799803)

- Most recent RFC moves this functionality into cgroups, which is likely where it belongs.
Acknowledgement on SW interleaving patch work and advocacy

MemVerge:
Gregory Price

Micron:
Venkata Ravishankar Jonnalagadda
Srinivasulu Thanneeru
Eishan Mirakhur
John Groves