Pond: CXL-Based Memory Pooling Systems for Cloud Platforms

Huaicheng Li1, Daniel S. Berger2,3, Lisa Hsu, Daniel Ernst2, Pantea Zardoshti2, Stanko Novakovic, Monish Shah, Samir Rajadnya2, Scott Lee2, Ishwar Agarwal, Mark D. Hill2,4, Marcus Fontoura, Ricardo Bianchini2

The Need for Memory Pooling

(1). DRAM is a major server cost: Azure (50%)
(2). Memory stranding and untouched memory

- Up to 25% stranded memory
- Half of the VMs have 50% of unused memory

Naive CXL Pooling is Inefficient

- CXL latency: one extra NUMA hop
- Workloads suffer from significant performance slowdowns under CXL

Pond Design is Effective in Saving 7-9% DRAM Needs

(1). Pond overpredicts 4% of VMs
(2). Pond saves 7-9% DRAM
(3). Pond with CXL latency at 222%
(4). Pond with CXL latency at 182%
(5). Pond with CXL latency at 158 workloads

Appears in the 28th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS’23)