Bede: Exploiting CXL-Memory for Job Scheduling

Speaker: Pooneh Safayenikoo

Job scheduling has emerged as a preeminent approach for allocating compute cluster resources to compute tasks. In job scheduling, users submit compute jobs together with descriptions of the jobs' resource requirements(e.g., the amount of memory that the job needs). A scheduler assigns resources from a compute cluster to each submitted job. Unfortunately, studies show that roughly half of all jobs experience
scheduling delays, in which they must wait to execute until cluster resources become available. In this work, we explore how emerging CXL.mem shared memory pools can improve job performance by reducing scheduling delays. CXL.mem shared memory pools allow multiple machines to share a pool of byte-addressable memory. Our insight is that memory pools increase the flexibility of memory resources which can reduce scheduling delay. Intuitively, when a job arrives that needs more memory than is available locally on any machines in the cluster, the scheduler can assign the job resources from a CXL.mem shared memory pool to begin execution immediately. The downside is that memory accesses to a CXL.mem shared memory pool have much higher latency compared to accesses to local DRAM.

Our research targets three primary challenges associated with implementing CXL technology for memory disaggregation in cloud computing environments. To address these challenges, we developed two distinct simulators: the first focused on the CXL technology itself, and the second on optimizing job scheduling and memory pool sizing using real-world data from major cloud providers like Google, Azure, and Alibaba.