Bede: Exploiting CXL-Memory for Cluster Job Scheduling

Job scheduling is the preeminent approach for allocating compute cluster resources to compute tasks. In job scheduling, users submit jobs, consisting of a workload and a set of resource requirements (e.g., an amount of memory needed by the job); a scheduler assigns resources from a compute cluster to each submitted job. This paper investigates job traces from Google and Microsoft clusters and observes that many jobs experience more than 10 seconds of memory-based scheduling delay, which occurs when the cluster delays a job's execution because it does not have a machine with sufficient idle memory.

We present Bede as a solution to improve compute cluster job scheduling performance by reducing memory-based scheduling delay. At the heart of Bede's design is the CXL.mem memory pool, which allows multiple machines to share a pool of byte-addressable memory. Our key insight is that memory pools allow machines to share idle memory, which makes it more likely that a machine will have sufficient memory for an otherwise memory-delayed job. Bede uses a tiered memory design, in which each server has its own memory and can access a CXL.mem memory pool, to reduce the end-to-end performance impact of CXL.mem memory pools. This tiered memory design imposes two challenges: (1) How should Bede schedule jobs across its tiers? and (2) How should we configure Bede to obtain the best performance improvement? Bede provides two new schedulers to solve the first challenge and a configuration simulator to resolve the second. We show that Bede improves average job execution time relative to state of the art by up to a factor of 30.07 (geometric mean: 5.92).