

Linux Memory Tiering & CXL 3.0 Optimization Handbook

Compute Express Link Fabrics, DAMON Page Monitoring, AutoNUMA Balancing, and Disaggregated Memory Pools

EXECUTIVE ABSTRACT: Compute Express Link (CXL 3.0) enables disaggregated, pooled memory over PCIe Gen5 interconnects, breaking the physical DRAM channel limits of modern servers. However, CXL memory incurs an additional 150-250ns of PCIe transit latency. This handbook provides an architectural guide to configuring Linux kernel memory tiering, tuning DAMON watermarks, and optimizing AutoNUMA page promotion to keep hot working sets in local DRAM while offloading cold pages to CXL pools.

Memory Architecture Tier	Interconnect Bus	Access Latency (ns)	Per-Socket Bandwidth	Cost per Gigabyte (\$)
Local NUMA Node 0 (DDR5)	Direct Memory Channel	65 - 85 ns (Ultra-fast)	300 - 460 GB/s	\$4.50 - \$6.00 / GB (High)
Remote Cross-Socket DRAM	AMD Infinity Fabric / Intel UPI	120 - 150 ns	150 - 250 GB/s	\$4.50 - \$6.00 / GB
CXL 2.0 / 3.0 Memory Expander	PCIe Gen5 x16 (CXL.mem)	180 - 260 ns (+170ns)	64 GB/s (Unidirectional)	\$2.20 - \$3.00 / GB (-50%)
CXL Pooled Fabric (Multi-host)	PCIe Gen5 Switch / Optical CXL	280 - 380 ns	Shared Fabric	\$1.50 - \$2.20 / GB (-65%)

1. The Server Memory Capacity Wall & The CXL Paradigm

Modern multi-core enterprise CPUs (with 128+ cores per socket) face severe memory starvation: DDR5 signal integrity limits motherboards to 1-2 DIMMs per channel, capping physical server memory at 1.5 to 2.0 TB per socket.

- Compute Express Link (CXL): Operating over physical PCIe Gen5 electrical pins, CXL.mem provides a cache-coherent, load/store interface that expands memory capacity beyond standard DIMM slots.
- The Latency Tax: Accessing CXL memory requires traversing the PCIe controller and CXL ASIC bridge, adding 150 to 250 ns of round-trip latency compared to native DRAM.

2. Linux Kernel Memory Tiering: Top-Tier vs. Secondary Tiers

Linux kernel 6.x introduces **Hierarchical Memory Tiering**:

- Node Categorization: The kernel assigns local NUMA nodes to `Top Tier` (default fast tier) and CXL expanders to `Secondary Tier` (slow tier).
- Demotion Path: When local DRAM becomes full, the kernel background page scanner (`kswapd`) avoids paging out to disk; instead, it executes synchronous in-memory demotion, copying cold physical pages from Top Tier to CXL memory.
- Promotion Path: When an application accesses a page residing on CXL, an AutoNUMA page fault triggers asynchronous migration back to fast local DRAM.

3. Production Tuning: DAMON (Data Access MONitoring) & sysfs Knobs

Default AutoNUMA heuristics react too slowly for latency-critical database engines (Redis, Cassandra, PostgreSQL). Operators must configure **DAMON**:

- Sampling & Aggregation: DAMON dynamically samples process address spaces to build an access frequency histogram with negligible CPU overhead (< 1%).
- Proactive Migration Rules: Configure DAMON sysfs schemes (`/sys/kernel/mm/daemon/`) to demote pages with zero access counts over 10 seconds, ensuring local Top Tier DRAM maintains a 20% free buffer for burst traffic.

4. Production Case Study: Redis & PostgreSQL Cost Optimization

Deploying a 50/50 ratio of local DDR5 and CXL memory pools with DAMON tuning reduces enterprise cloud infrastructure TCO by 38% while keeping p99 query latency within 4% of pure local DRAM.

STRATEGIC ARCHITECTURAL TAKEAWAYS

1. CXL 3.0 provides disaggregated server memory over PCIe Gen5, cutting per-gigabyte RAM costs by up to 50%.
2. CXL memory incurs an additional 150-250ns latency penalty, requiring intelligent operating system memory tiering.
3. Linux kernel DAMON schemes proactively migrate cold pages to CXL while preserving fast local DRAM for hot data.