CXL KV Offload for Multi-Turn Inference

Benchmarking XCENA MX1 with LMCache + vLLM

구현회 (Hyunhoi Koo) · Research Engineer, Lablup

2026.05.28

What Is CXL Memory?

"**DRAM on a PCIe card**"

Byte-addressable memory you mmap from userspace and DMA from a GPU, with capacity beyond CPU DIMM channels.

Agentic Workloads Strain the Memory Hierarchy

Agent characteristics

- Long contexts (100K+ tokens).
- Same prefix replayed every turn.
- Many sessions in flight in parallel.
- KV cache ~16 GiB per context.

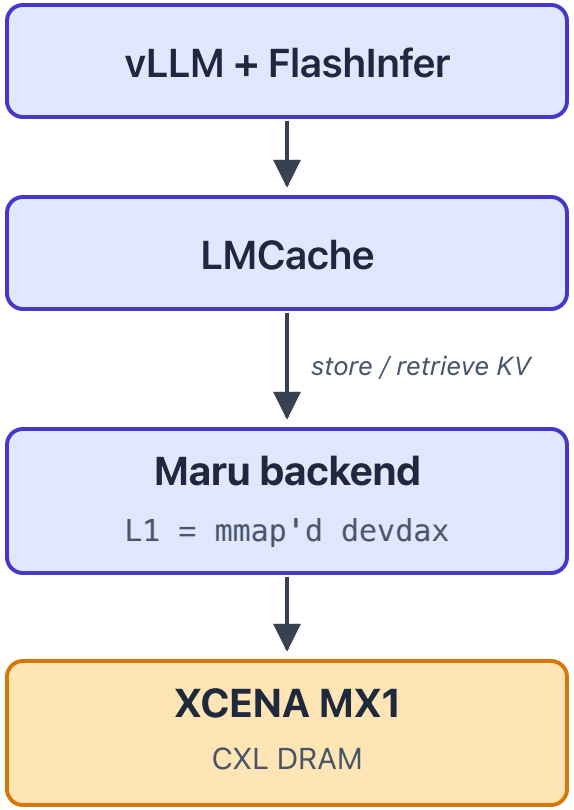
KV cache offload options

- **DRAM**: local, fast, capped by DIMM channels.
- **NFS**: shared across nodes, network latency per fetch.
- **CXL**: local DMA, capacity via PCIe slots.

VRAM alone cannot hold the working KV set. Without an offload tier, every turn pays full re-prefill.

Architecture

- **LMCache** intercepts vLLM's KV cache lifecycle.
- After prefill: async store to CXL.
- On hit: DMA from CXL to GPU VRAM.
- **Maru** backend: L1 mapped onto CXL devdax via `mmap`.
- One device backs multiple replicas.



Experiment Setup

Cache

XCENA MX1, 466 GiB CXL DRAM on `/dev/dax9.0`. LMCache L1 = 256 GiB per replica (Maru backend).

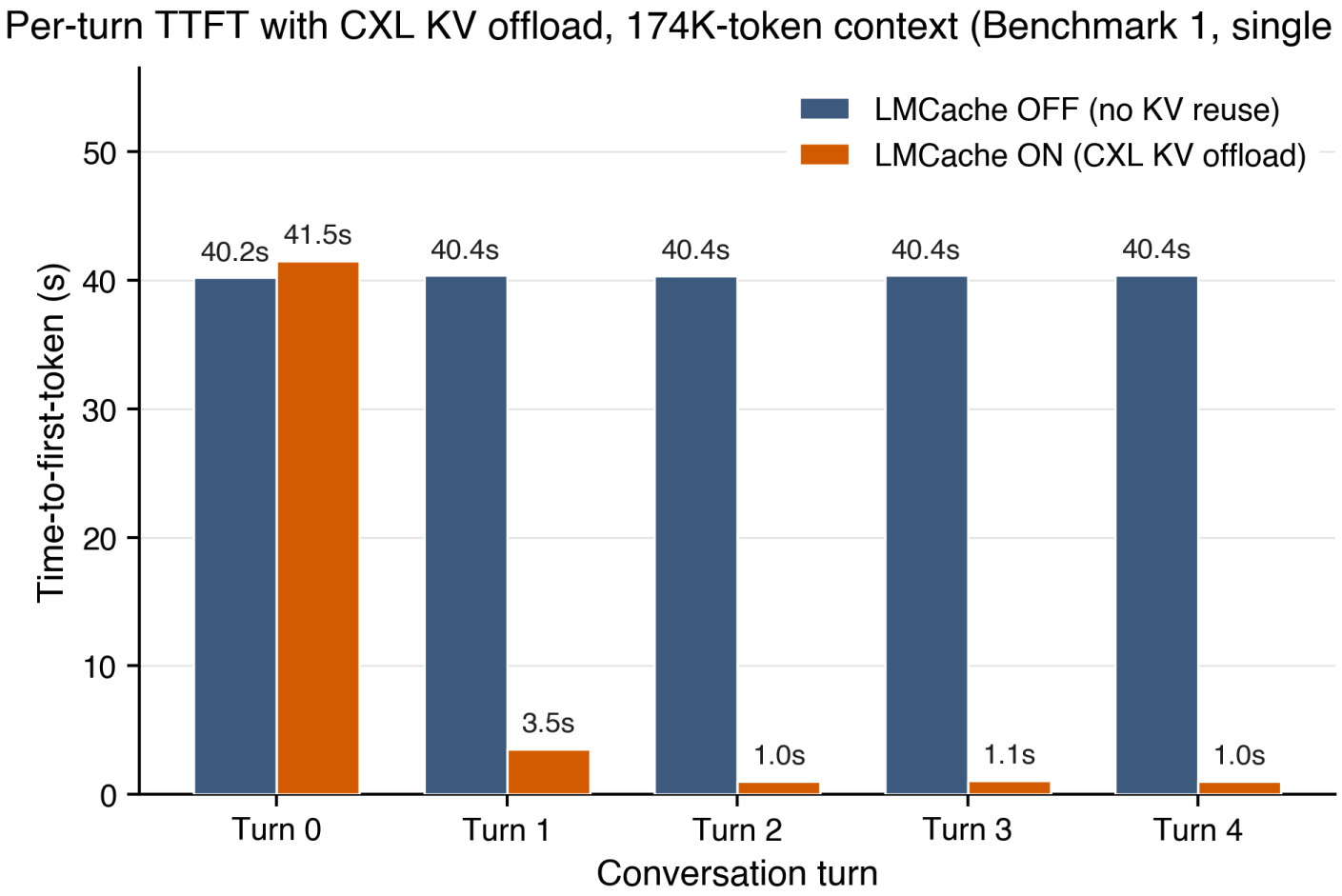
Stack

`Qwen3-30B-A3B` BF16 MoE on 2x `RTX Pro 6000` (96 GB each). `vLLM 0.20.1` + `LMCache`.

Workload

10 contexts x 5 turns. ~174K tokens per context (BFD-packed Python). `MAX_MODEL_LEN=200K`.

Benchmark 1: Single Replica, Sequential

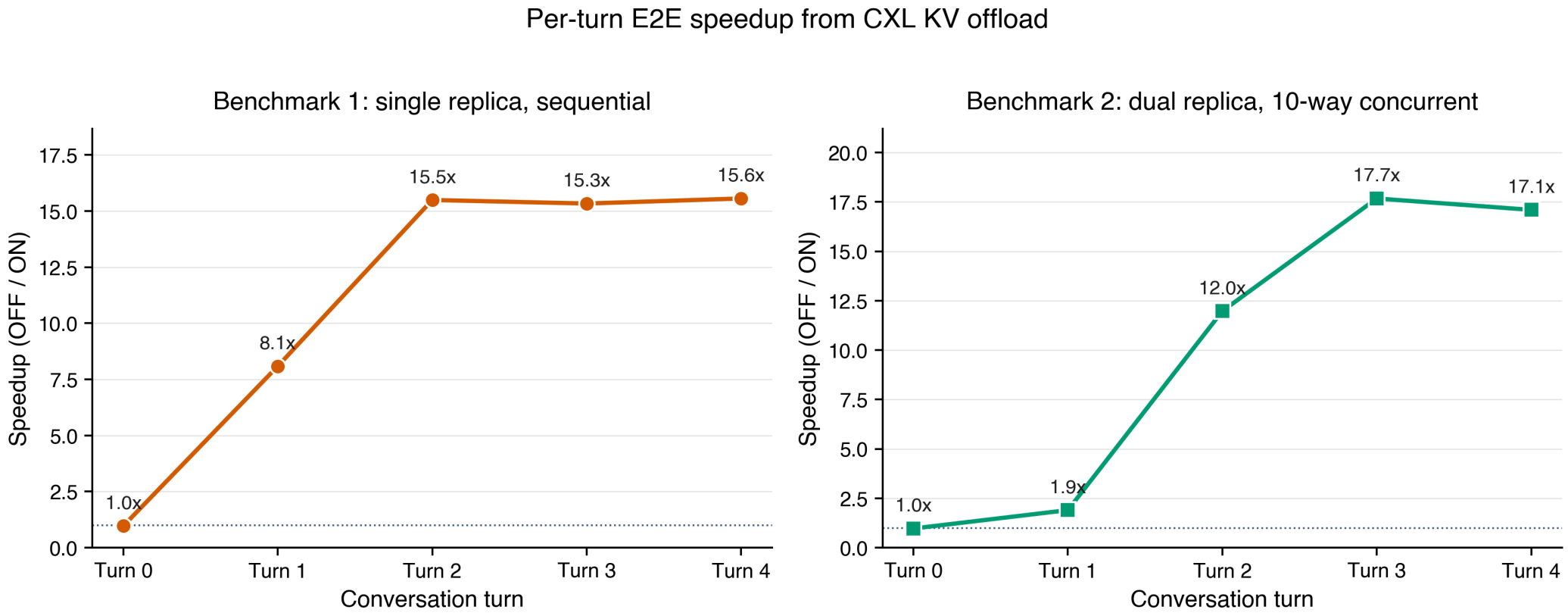


3.7x wall clock

4.2x avg TTFT

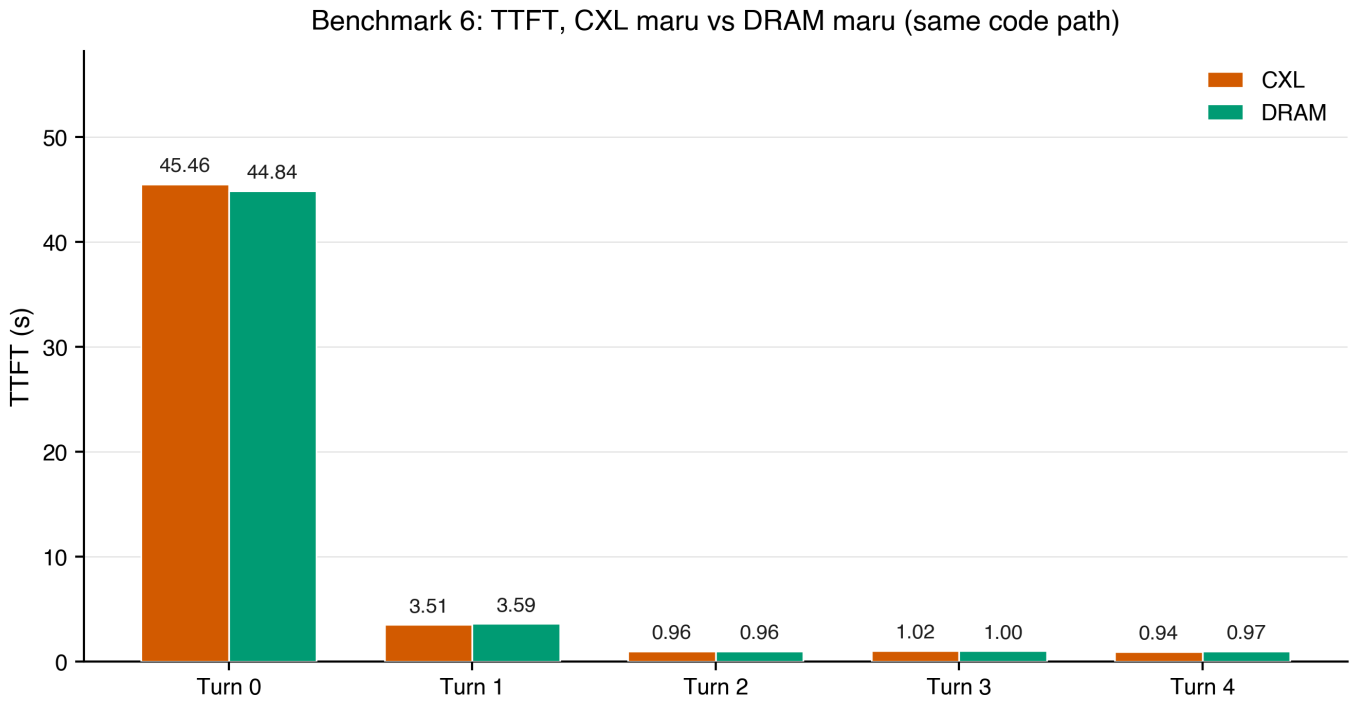
~15x warm turns

Benchmark 2: Dual Replica, 10-Way Concurrent



3.8x wall clock · **~17x** per-request steady state · holds under concurrency stress

CXL Compared



vs DRAM

- Same code path on both tiers (`maru` backend).
- Turn 0 cold prefill: ~45 s on both.
- Turn 1 first retrieval: 3.5 s CXL, 3.6 s DRAM.
- Steady-state (turn 2-4): ~1.0 s on both.

vs NFS

- Cluster-wide capacity.
- Per-fetch network + storage round-trip.

backend AI

Questions?