Server Long Term RAM

Hui Sub (David) Shim and Katherine Mohr
Advised by Philip Levis

Problem

As DRAM's cost-per-bit has plateaued over the past decade, DRAM has become a major limiting factor in datacenter growth.

A new class of emerging memory technologies, known as **Long-term RAM (LtRAM)**, can be manufactured more cheaply and densely. This density comes with tradeoffs:

- Writes are **slower**
- Write granularity is **coarser** than a cache line
- Endurance is **bounded**

Device	Read lat.	Write lat.	Endurance	Write gran.	Cost/bit
DRAM	80 ns	80 ns	inf.	cache line	high
3D V-RRAM	100 ns	1 µs	10^6	page	low
3D FeFET	200 ns	10 µs	10^5	page	low
PCM	300 ns	1 µs	10^8	256 B	mid
STT-MRAM	20 ns	20 ns	10^{15}	word	high

Prior approaches to integrate LtRAM into servers have **exposed a DRAM-compatible interface** to the software, hiding LtRAM characteristics with **complex memory controllers**. However, hardware complexity **increases cost and performance variability**.

Optane Mechanism	Resulting Overhead	LtRAM Interface Fix
Cache-line Granularity Mismatch	**75%** write throughput drop, **2×** random-read latency	Cache-line reads, page writes; OS coalesces writes
Wear-Leveling	Opaque migration fires every ~3.4 MB written; **~60 µs tail read latency (160×)**	OS-side wear-leveling; hardware has no migration path
Address Indirection Table	**76–200 ns** added to every read	OS-side wear-leveling; no on-module map
Workload-Agnostic Design	70% read bandwidth drop under 50/50 R/W	Read-mostly by design; token allocator paces writes

hsshim@stanford.edu, kmohr@cs.stanford.edu

Solution

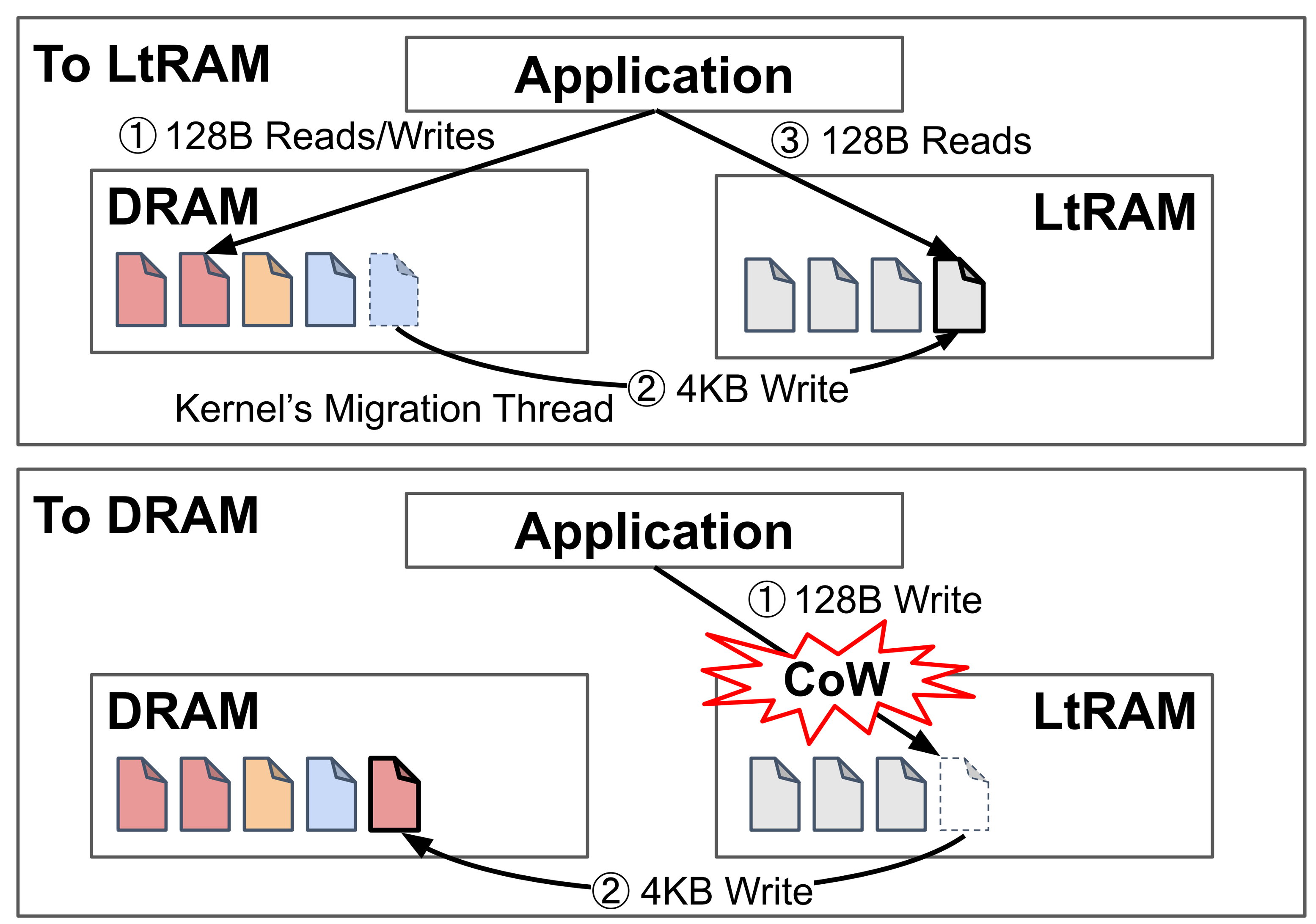
We introduce a novel HW/SW interface that enforces an invariant we call **Application Read-Only Memory (AROM).**

- Only the kernel may write to LtRAM when migrating pages from DRAM.
- LtRAM is **read-only to applications**.
- This design allows us to move LtRAM logic **from the on-DIMM controller to the OS**.

The on-DIMM controller supports **reads at cache-line granularity** and **writes at 4 KB page granularity**, and it exposes cell health metadata.

The **OS handles policy decisions** such as data placement, page migration, and wear-leveling.

Design

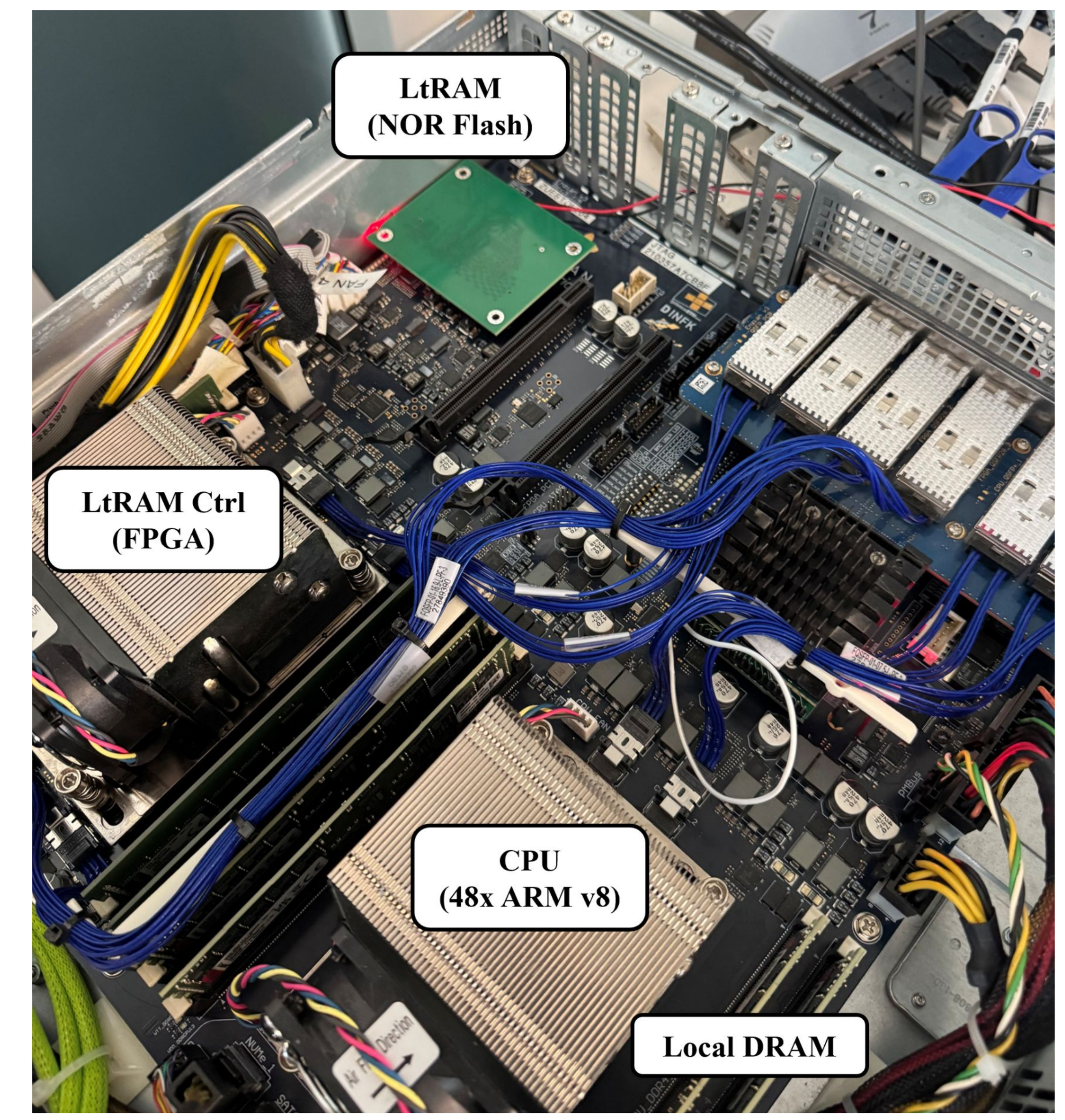


Challenges

- What is the **ideal LtRAM technology** to develop?
- What is the **best migration policy**?
- What is the **best wear-leveling strategy**?
- How can **out-of-memory faults be prevented** during write-heavy periods?

Evaluation

We prototype the interface on the **Enzian platform**, configured so the FPGA acts as the on-DIMM controller. **NOR flash** is chosen as the LtRAM technology due to its cost and availability.



Impact

While NOR flash is inherently slower than DRAM, **emerging LtRAM technologies approach DRAM read performance**, enabling mixed DRAM/LtRAM systems to match pure DRAM systems' performance.