59th IEEE/ACM International Symposium on Microarchitecture (MICRO 2026)
November 3, 2026 - Athens, Greece

DIKTAMO*: Extending CXL for Resilience to CPU Failures

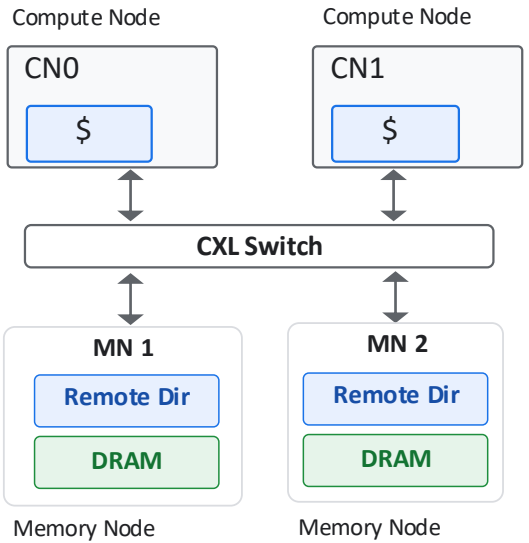
Antonis Psistakis**, Burak Ocalan, Chloe Alverti, Fabien Chaix, Ramnatthan Alagappan, and Josep Torrellas

*Diktamo: Endemic to the island of Crete (Greece), this herb has been known since antiquity for its reputed healing properties.

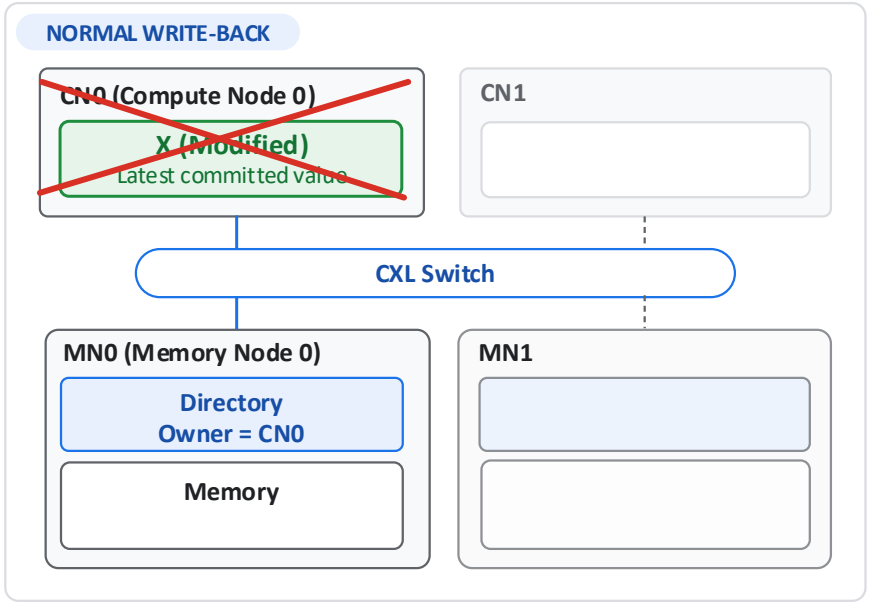
** Currently affiliated with Google

Background

- **Compute Express Link (CXL)**
  - Enables memory pooling and sharing
  - CXL 3.0+ enables compute nodes (CNs) to access remote memory nodes (MNs) with **hardware-enforced cache coherence**
- **Existing Fault Tolerance Specs**
  - **Failure handling**: Addresses link, switch, and device failures through error detection (e.g., timeouts) and packet retransmission
  - **Key limitations**:
    1. **No processor/host (CN) failures considered**
    2. **On a failure, no attempt to recover app to consistent state**



What Happens When a CPU Holding Dirty Shared Data Crashes?



1. Dirty (new) data are lost
2. Crashed CN is unreachable

One Approach to Attain Resilience: Write-Through Protocol

Option A — Write-Back (WB)

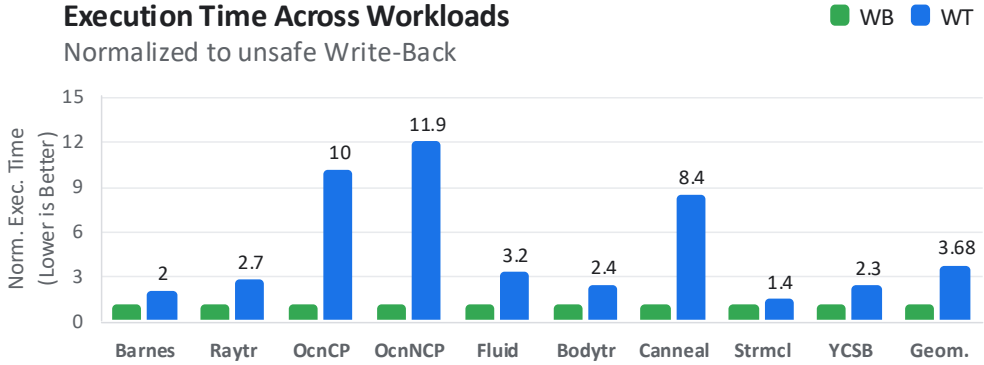
Unsafe

Local L1 cache absorbs stores



Execution Time Across Workloads

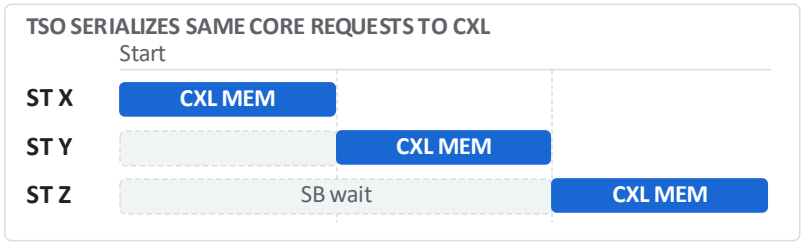
Normalized to unsafe Write-Back



Option B — Write-Through (WT)

Safe

Every store writes synchronously to resilient MN



GEOMEAN SLOWDOWN

3.68x

Contribution: DIKTAMO



What? Extends CXL to make it resilient to processor failures



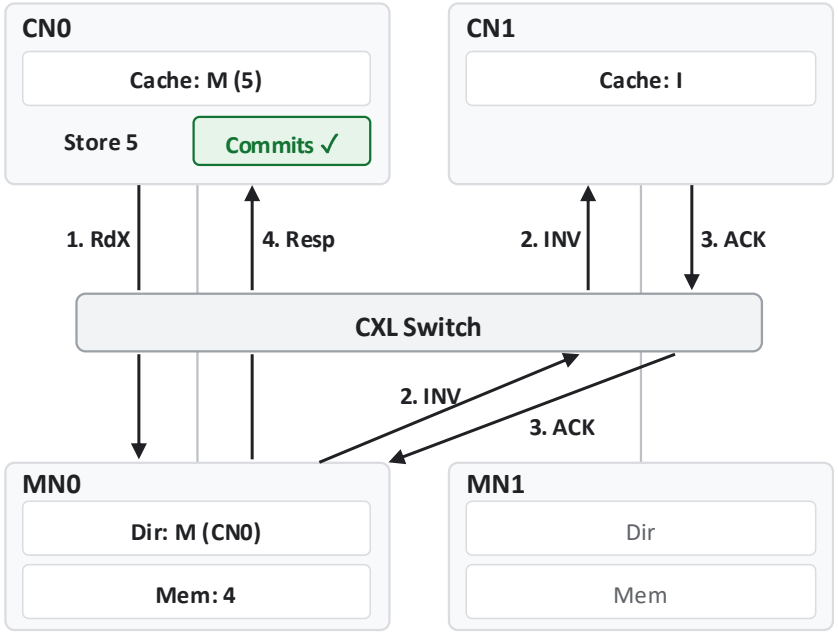
How? Extends traditional WB MESI coherence protocol with replication messages



Result: Enables recovery of program state after a processor failure

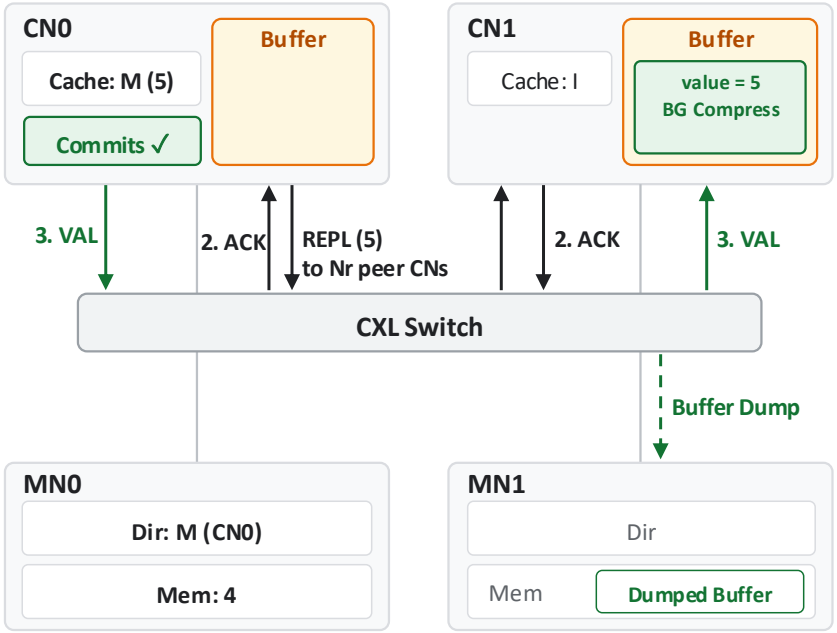
High-Level Protocol: WB Coherence vs. DIKTAMO Approach

WB Coherence



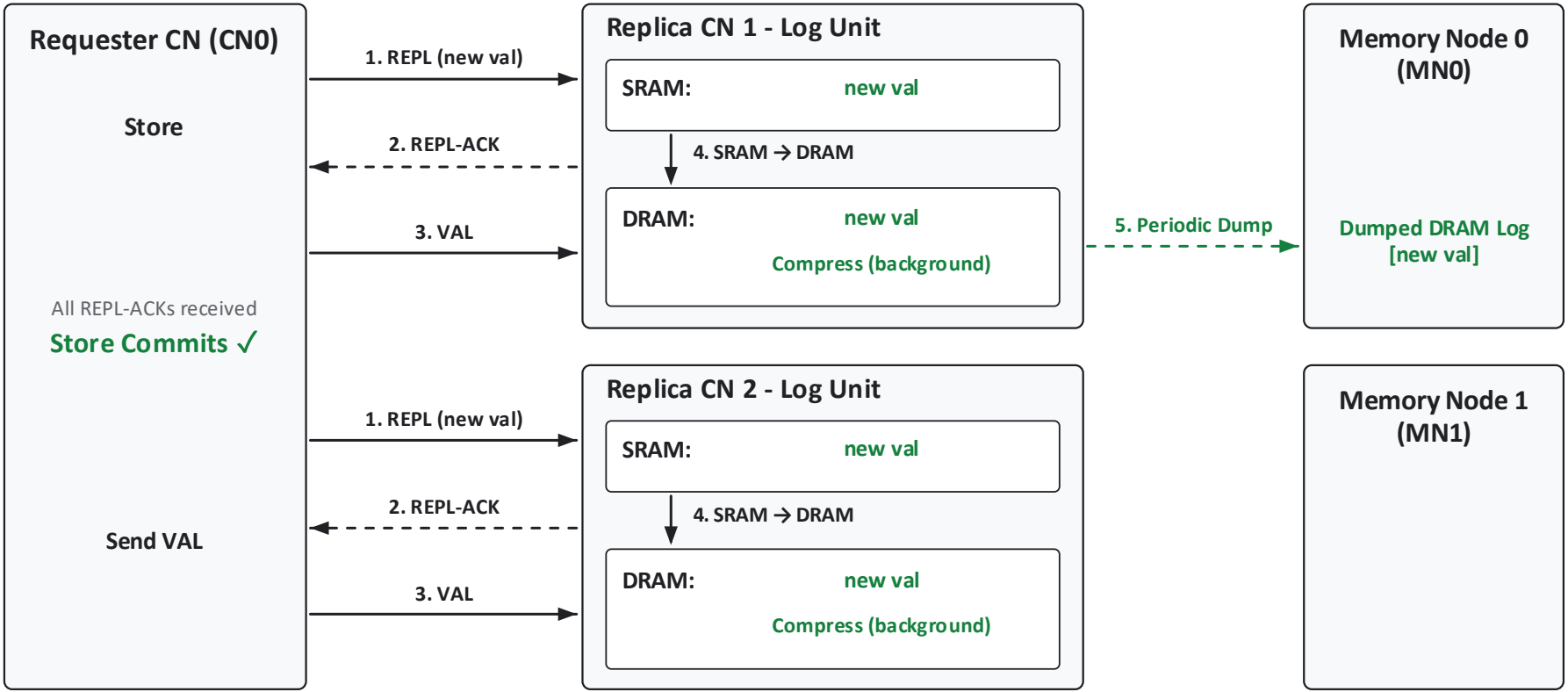
- Store commits as soon as coherence finishes

DIKTAMO: WB Coherence + Replication

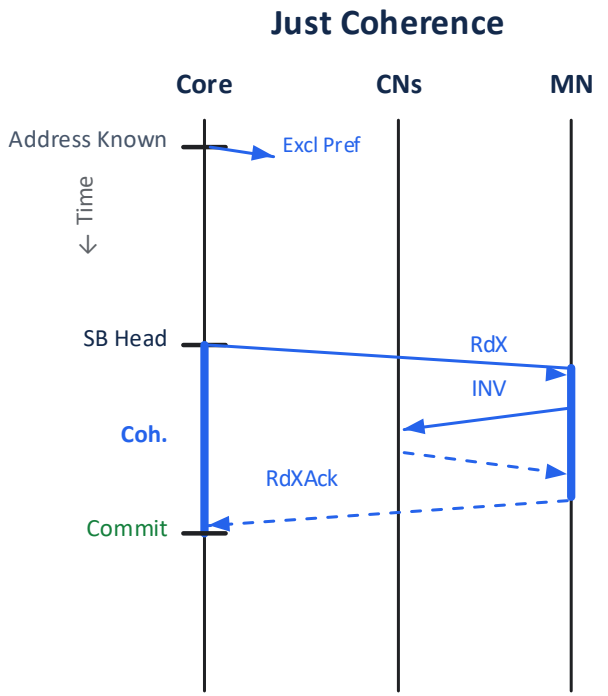


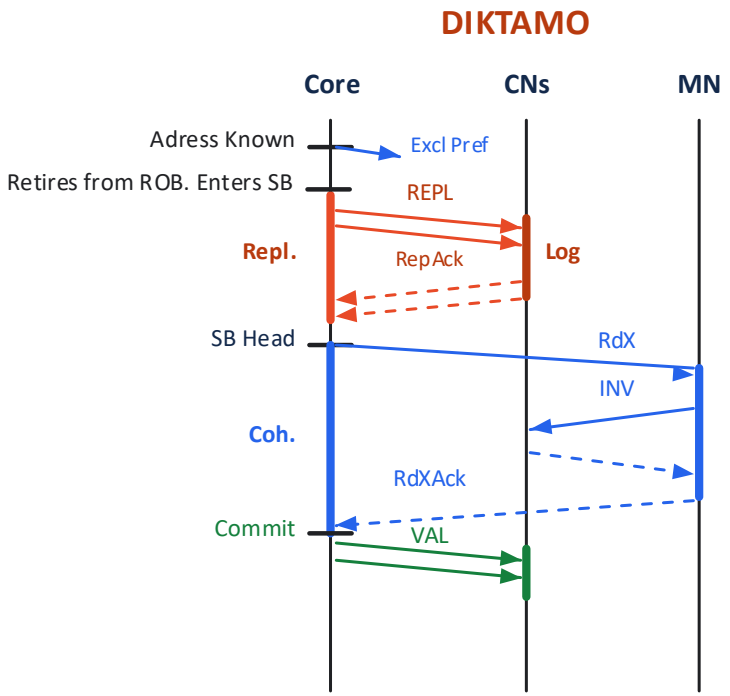
- **After commit:** Entries in the buffers are compressed in the background and periodically dumped at one of the MNs

DIKTAMO Replication Protocol & Log Lifecycle



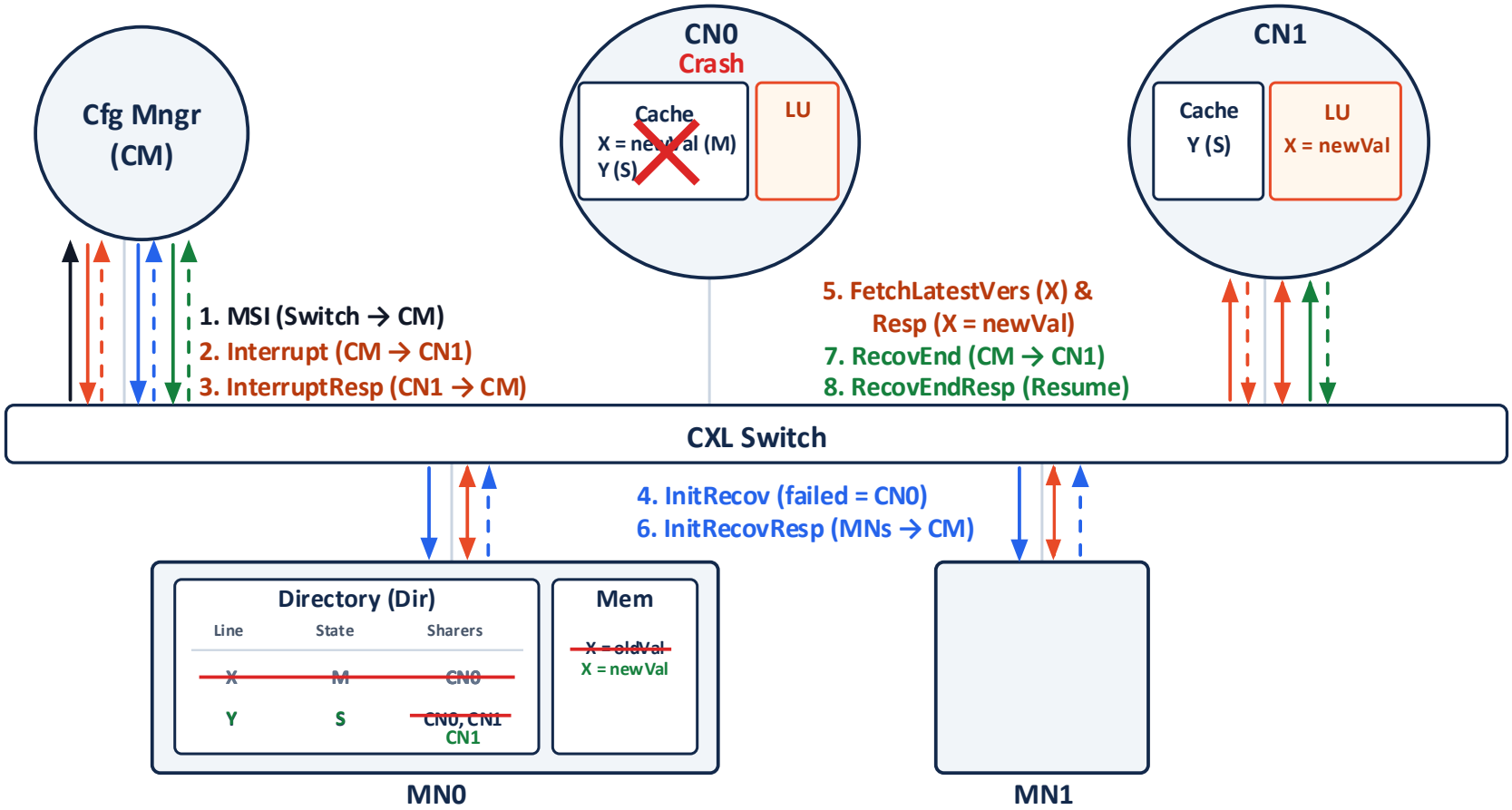
Timing of a Store: Overlapping Coherence and Replication Messages





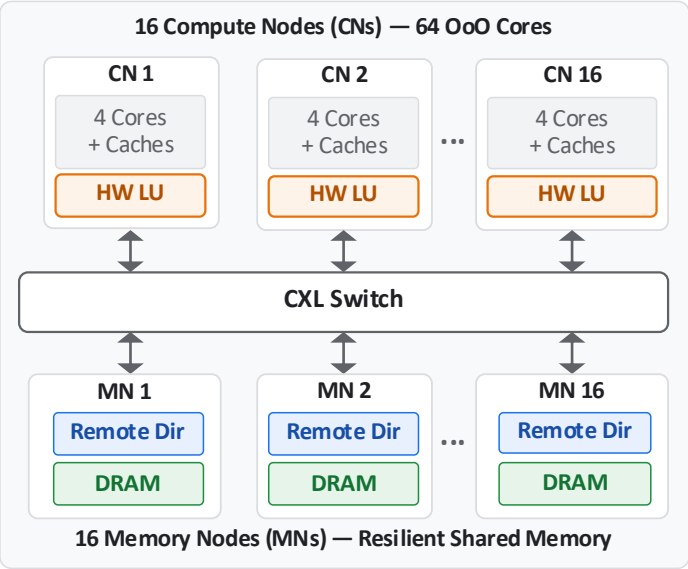
+ Overlap coherence and replication transactions
+ Overlap replication messages of multiple transactions (even in TSO)

DIKTAMO Fault Recovery



Methodology

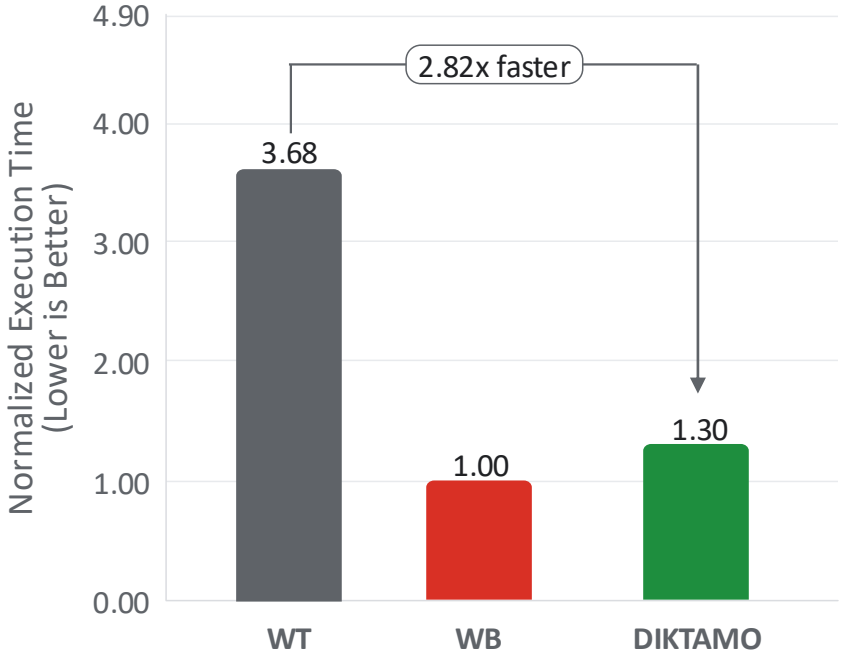
Simulated Cluster Architecture (SST + Pin)



- 9 Workloads across 3 Benchmark Suites
  - PARSEC, SPLASH-2, YCSB
- CXL Network: 200 ns RTT
- Replication factor: Nr = 3

Execution Time

Geomean Execution Time Across Workloads Normalized to unsafe Write-Back



DIKTAMO OVERHEAD VS. WB

1.30x

DIKTAMO SPEEDUP VS. WT

2.82x

Why DIKTAMO Wins?

- **SB overlap:** Replication messages hidden in SB wait time
- **Avoids SB serialization:** Overlapping replication messages from multiple stores without violating TSO (same core)

Conclusion

- CXL specification has incomplete support for **resilience to CN failures**
- **DIKTAMO**
  - **Replicates** stores into hardware Logging Units
    - **Replication messages** overlapped with **SB waiting time**
  - **Recovers** program state
- **Performance**
  - 30% overhead higher than unsafe WB
  - 2.82x speedup over WT

59th IEEE/ACM International Symposium on Microarchitecture (MICRO 2026)
November 3, 2026 - Athens, Greece

DIKTAMO*: Extending CXL for Resilience to CPU Failures

Antonis Psistakis**, Burak Ocalan, Chloe Alverti, Fabien Chaix, Ramnatthan Alagappan, and Josep Torrellas

Thank you! Any questions?

Scan for Paper (PDF)

*Diktamo: Endemic to the island of Crete (Greece), this herb has been known since antiquity for its reputed healing properties.

** Currently affiliated with Google