An Analysis of Persistent Memory Use with WHISPER

Sanketh Nalli, Swapnil Haria, Michael M. Swift, Mark D. Hill, Haris Volos, Kimberly Keeton

WISCONSIN
UNIVERSITY OF WISCONSIN – MADISON

HP Labs

MOTIVATION: A STANDARD BENCHMARK SUITE FOR PERSISTENT MEMORY (PM)

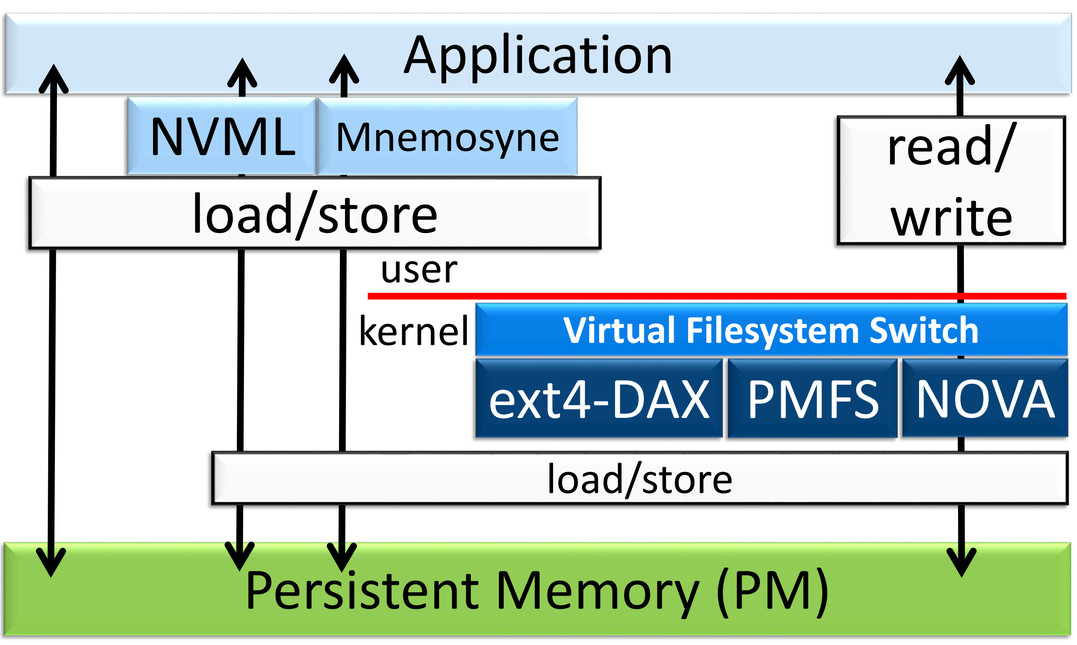
- **Address lack of understanding of PM usage**
- **Minimize use of ad hoc micro-benchmarks**
- **Study diverse, real interfaces for consistency and durability of data and metadata in PM**
- **Aid design of better PM runtimes**
- **Establish a standard for evaluating PM runtimes**

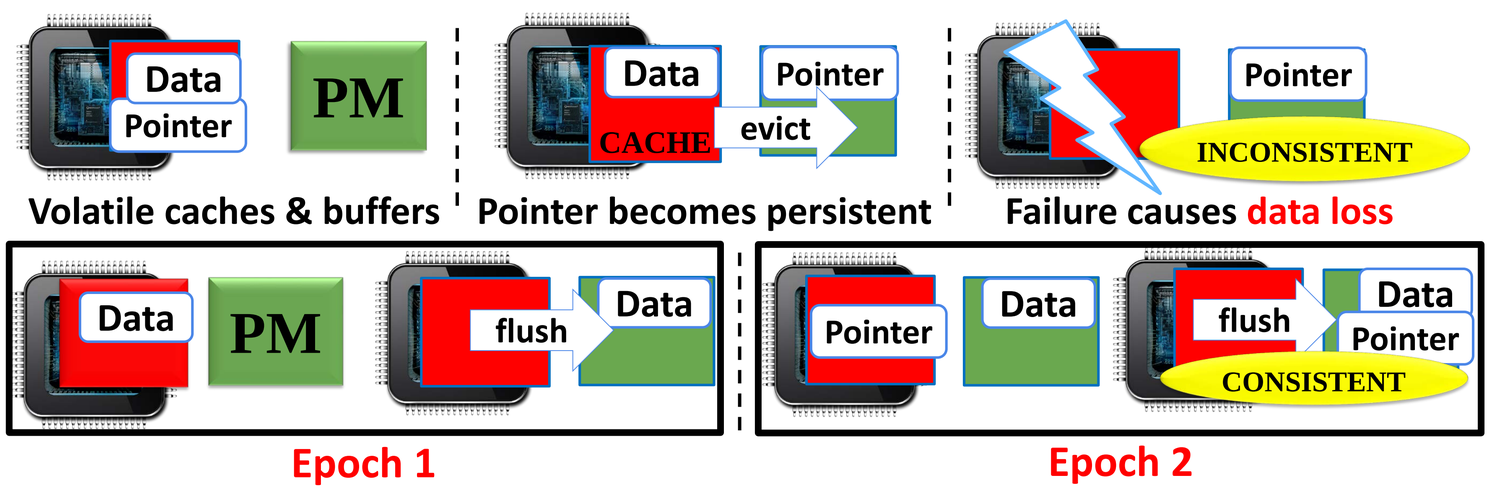
WHISPER = Wisconsin-HP Labs Suite for Persistence

research.cs.wisc.edu/multifacet/whisper/

Application	System	Brief Description	(*Adapted to PM)
Echo*	Native	Multi-version KV store	(U of Washington)
N-store*	Native	Fast, in-memory relational DB	(Carnegie-Mellon)
Redis	NVML	In-memory KV store	(Intel)
C-tree, Hashmap	NVML	Multi-threaded microbenchmarks	(Intel)
Vacation*	Mnemosyne	OLTP travel reservation system	(U of Wisconsin)
Memcached*	Mnemosyne	Distributed in-memory KV store	(U of Wisconsin)
NFS	PMFS	Linux server/client for remote PM access	(Intel)
Exim	PMFS	Popular mail server using mailbox format	
MySQL	PMFS	Widely used RDBMS for OLTP	

BACKGROUND : PM WORKLOADS, EPOCHS, CONSISTENCY & DURABILITY

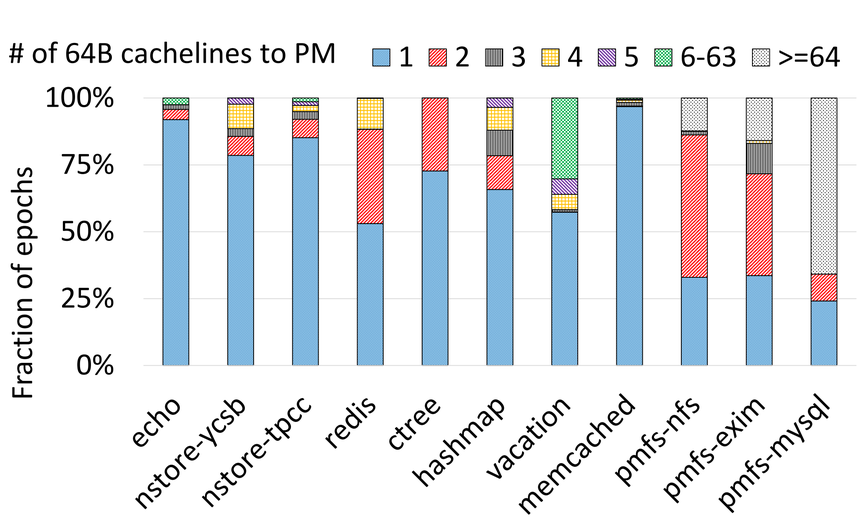


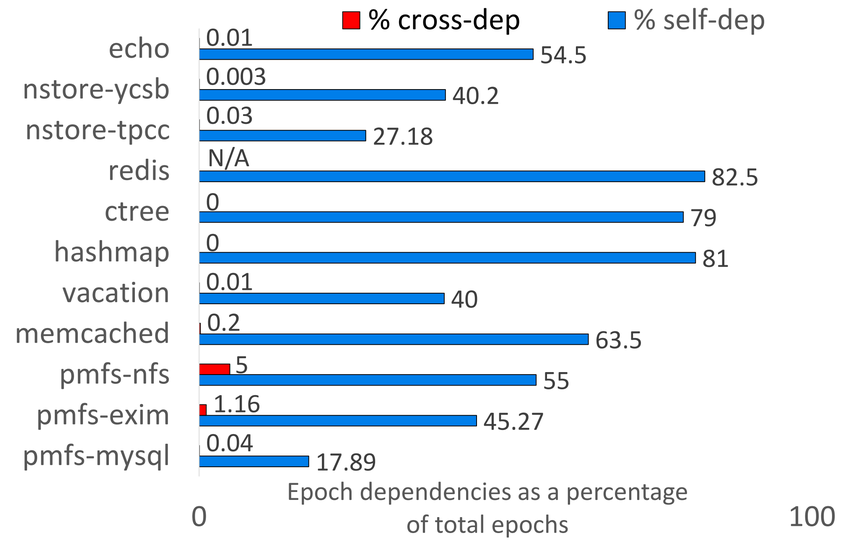


- **Controlling order of writes to PM preserves consistency at all stages.**
- **Epoch = Set of writes to PM guaranteed to be durable before any subsequent writes become durable.**

SELECTED ANALYSIS RESULTS

- **What % of accesses in WHISPER applications are to PM?**
- **How many epochs are there in a transaction to PM?**
  (∵ Durability involves writes to PM & costly if enforced on each epoch)
- **How often do epochs from same or different threads write to the same cacheline to PM ?**
  (∵ Epochs writing to same cache line depend on each other and stall execution)



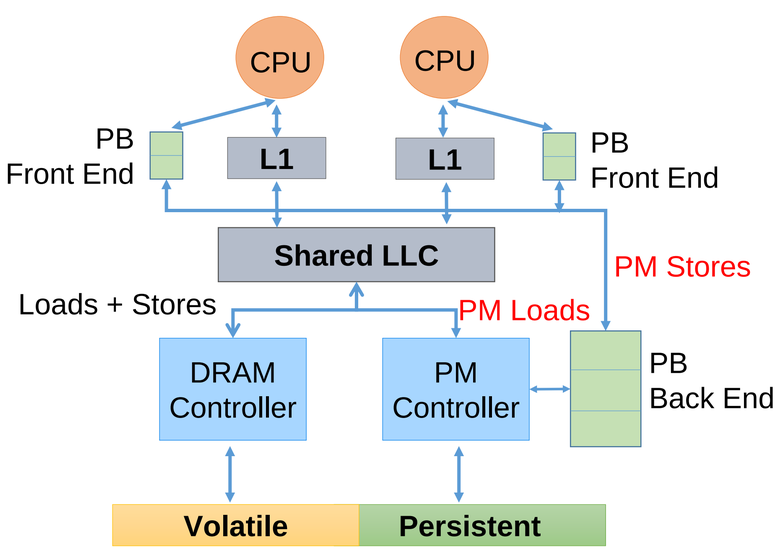


Takeaway: Enforce durability only after last epoch in a transaction as enforcing it on each epoch can slow down execution and impede transaction performance.

Takeaway: Design hardware with multi-versioned buffers and optimize data structures in software to minimize dependencies among epochs.

ANALYSIS SUMMARY → HOPS DESIGN → PERSIST BUFFERS IN HOPS

- **4% accesses are to PM, 96% are to DRAM** → **Volatile memory hierarchy (almost) unchanged by PBs**
- **5-50 epochs per transaction** → **Order Epochs without flushing (OFENCE and DFENCE)**
- **Self-dependencies common** → **Allow multiple copies of same cacheline in PBS via timestamps**
- **Cross-dependencies are rare** → **Correct, conservative method using coherence, timestamps**



24% faster than Intel extensions for PM