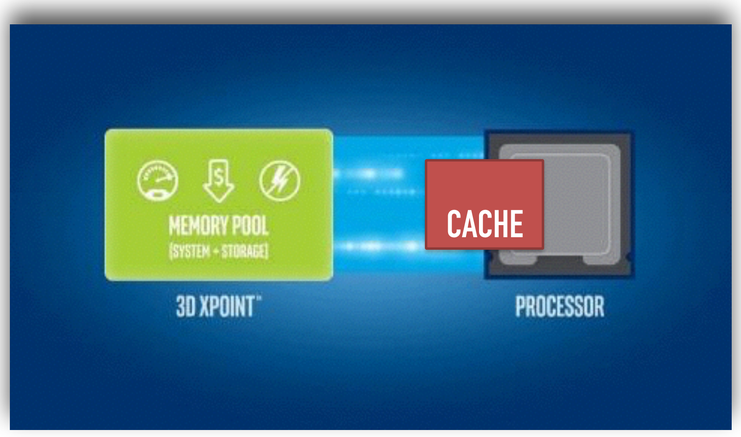
Re-visiting CPU caches in light of NVRAM

Sanketh Nalli, Sean Wilson, Michael Swift, Haris Volos, Kim Keeton

THE UNIVERSITY of WISCONSIN MADISON

HP Labs

Motivation

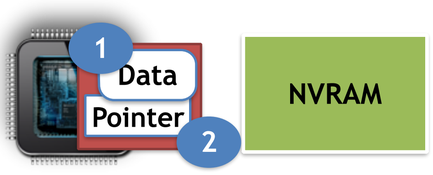


NVRAM is happening !

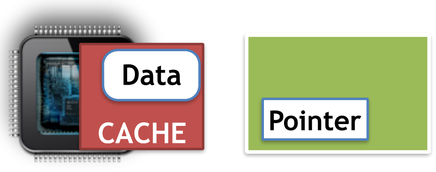
- **3D XPOINT[1]**, an example of NVRAM
- Designed by Intel and Micron, 2015
- Byte-addressable, persistent memory
- Access latency comparable to DRAM
- Much larger capacity than DRAM
- 1000x more durable than NAND

CPU caches are an incomplete extension of NVRAM

Assertion	Implication
CPU caches write-back data to hide memory latency	NVRAM data in CPU caches will be lost on power failure
Cache replacement policy re-orders stores to memory	Inconsistency may arise in NVRAM state



A. Data update followed by pointer update



B. Cache re-orders updates to NVRAM sub-system

C. Data lost on power failure, dangling pointer persists.

Proposed extensions to CPU caches have drawbacks

Extension	Drawback
Epoch barrier [4]	Complex changes to caches; possible burst of stores
x86 clflushopt, clwb & pcommit instructions [2]	Low-level, error-prone management of caches

Analysis

Goal : Characterize NVRAM's access pattern and its CPU cache usage

Methodology

- Using qemu-mtrace, we run workloads
- And collect traces of loads, stores, *movnti*, *mfence*, *clflush* & *pcommit*

Workloads

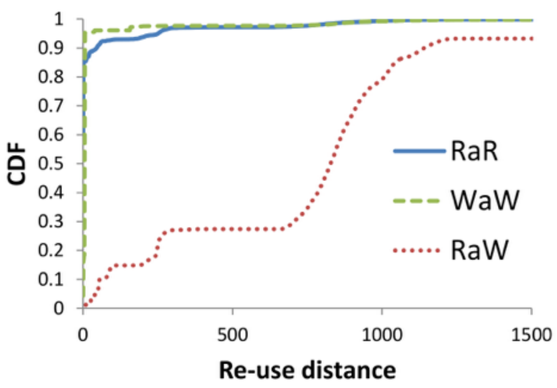
- PMFS [2] - a NVRAM kernel-filesystem
  - File, web and proxy servers on PMFS
- Mnemosyne [3] - a NVRAM user-library
  - Memcached server on Mnemosyne

Metrics

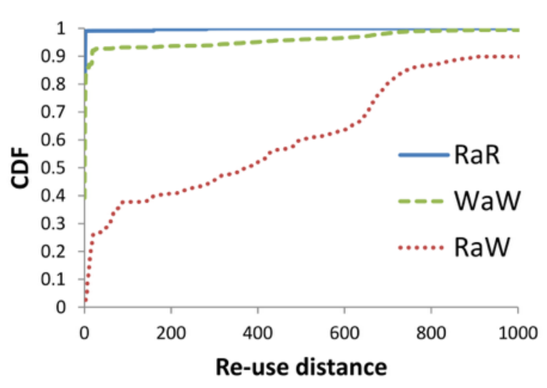
- **Read-after-Read (RaR)** is the number of unique memory addresses between 2 consecutive loads to same NVRAM address
- **Write-after-Write (WaW)**..., for 2 consecutive stores to same NVRAM address
- **Read-after-Write (RaW)**..., for load-after-store to same NVRAM address
- **clflush-to-pcommit** is the number of instructions exec'ed between a clflush & pcommit instruction in 1 NVRAM transaction

Findings

NVRAM has low read-after-write reuse



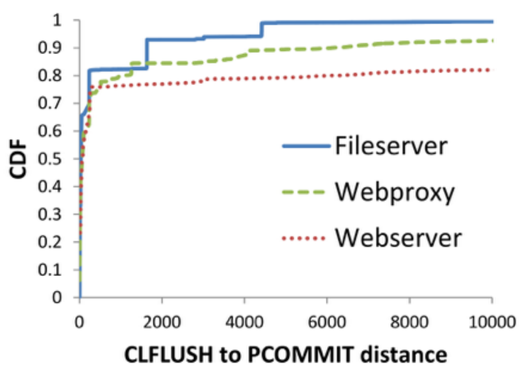
PMFS



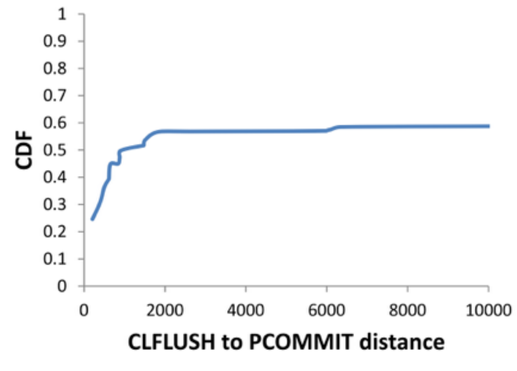
Mnemosyne

- Small RaR & WaW distances, due to streaming nature of reads & writes to NVRAM
- High RaW distances, indicate low re-use
- CPU caches may be ineffective in capturing NVRAM working sets

Huge storm of flushes before commit



PMFS



Mnemosyne

- PMFS & Mnemosyne flush dirty cache lines right before the end of a transaction to ensure crash-consistent NVRAM-state
- Asynchronous write-backs may alleviate high-latency commit and low-level cache mgmt.

[1] http://newsroom.intel.com/community/intel_newsroom/blog/2015/07/28/intel-and-micron-produce-breakthrough-memory-technology
[2] S. R. Dulloor et. al. System software for persistent memory. *9th ACM European Conference on Computer Systems*, April 2014.
[3] H. Volos, A. J. Tack, M.M. Swift. Mnemosyne: lightweight persistent memory. ASPLOS XVI, 2011.
[4] J. Condit, C. Frost et. al. Better I/O through byte-addressable, persistent memory. *22nd ACM SOSP*, October 2009.