WrAP: Write Aside Persistence for Storage Class Memory in High Performance Computing

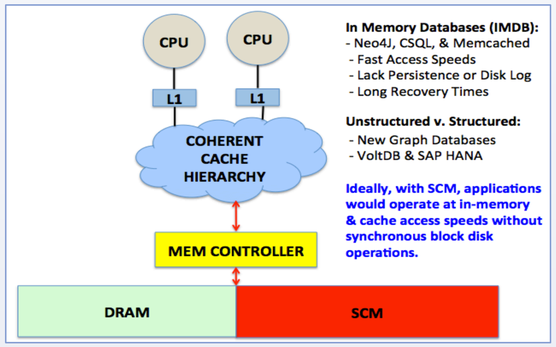
Ellis Giles
Rice University

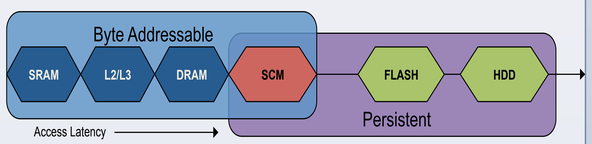
Peter Varman (Advisor)
Rice University

Kshitij Doshi (Advisor)
Intel Corporation

MOTIVATION

Emerging Storage Class Memory (SCM) promises to be a fast, byte-addressable, persistent memory near DRAM on the main memory bus.

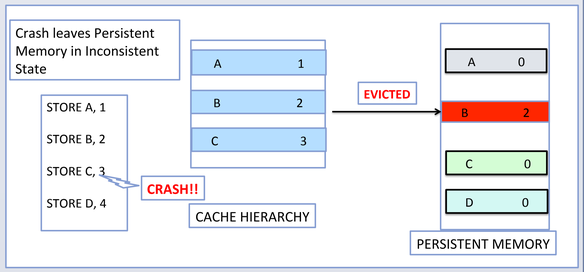


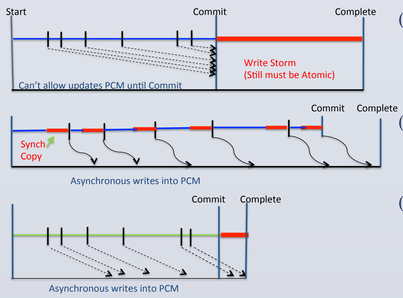


Technology	Density um²/bit	Read / Write Latency (ns)		Read / Write Energy pJ/bit		Write Endurance
HDD	0.00006	3,000,000	3,000,000	2,500	2,500	Near Inf
Flash SSD	0.00210	25,000	200,000	250	250	10^5
DRAM	0.00380	55	55	24	24	10^{18}
PCM	0.00580	48	150	2	20	10^8
Memristor	0.00580	100	100	2	2	10^8

PROBLEM

Hardware and software crashes can leave the system in an inconsistent state.





(a) All writes must be atomic

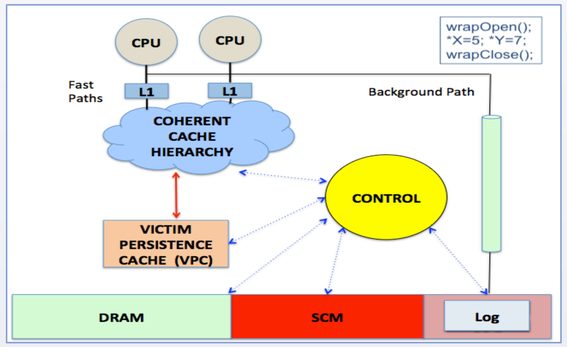
(b) Copy-on-Write

(c) Ideal – asynchronous updates to SCM

Random cache evictions and system crashes cause the programmer to create a slow logging solution to maintain SCM consistency.

HARDWARE WrAP

WrAP automatically and implicitly catches cache evictions to SCM between a WrAP open and close to prevent overwriting previous values.



Controller
- Handles WrAP Tokens
- Cache Evictions
- Prevents SCM from updates
- PCM read priority over Log write

Victim Persistence Cache
- Contains evicted entries from open WrAPs

Background Path
- Buffer
- All writes must flush to Log

Log
- SCM Address/Value Pairs
- Maintains Consistency & Durability
- On wrap close can write to SCM

Giles, E., Doshi, K., and Varman, P. "Bridging the Programming Gap Between Persistent and Volatile Memory Using WrAP." In ACM Computing Frontiers (2013).

WrAP time approaches the non-transactional time where there are no atomic guarantees in a transaction by the application.

RELATED WORK

Atomic 8-byte write to Storage Class Memory is needed.

Ordering of Primitives for Persistent Memory
- Still doesn't solve atomic update problem.

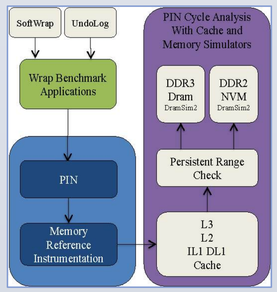
Battery Backup Based Work
- Whole System Persistence
- Recoverable Memory Techniques

Up-Front Cache Line Changes & Data Copying
- BPFS (copy-on-write)
- NV-Heaps (Logging)

Software Control Layer and Compiler Additions
- Mnemosyne

Versioning with Data Copying
- Consistent Durable Data Structures

EXPERIMENTAL SETUP

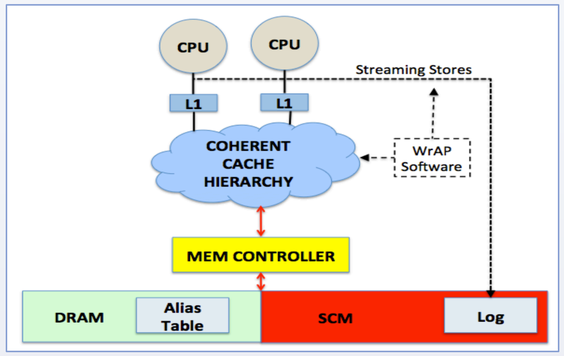


Experimental Setup:
- Pin – A dynamic binary instrumentation tool.
- DRAMSim2 – A cycle accurate memory simulator.
- WrAP API Implementation

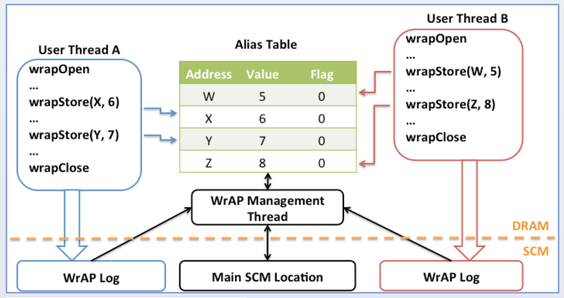
Log Cleanup Options:
- Lazy
- DRAM Alias Table
- SCM Log Area

SOFTWARE WrAP

Programmer explicitly uses WrAP primitives to load and store values. An Alias Table for variables is used to keep working copies in DRAM.



The Alias Table is implemented in DRAM and is checked on a wrapLoad or wrapStore, not requiring synchronous reads or writes to SCM.



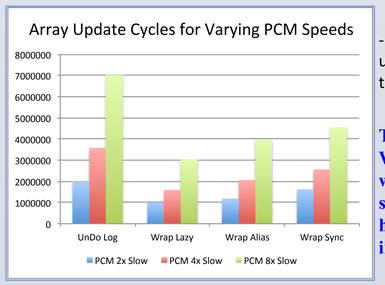

```
wrapToken wid;
wid = wrap_open();
begin
    wrapStore(wid, &x, 1);
    .........
    temp = p_malloc(100);
    wrapStore(wid, &p, temp);
    .........
    for(i = 0; i < 25; i++) ;
    begin
        wrapStore(wid, p+i, i);
    .........
wrap_close(wid);
```

- After all WrAPs are closed, the Alias Table can be flushed directly to the SCM home locations, avoiding the need to examine logs.
- The continuous log area in persistent memory allows for write combining.
- Processing logs can be performed concurrently to free space.

```
WrapStore (wrapToken w, address a, value newVal)
begin
    if No entry for a in AliasTable then
        Allocate DRAM location a' for a;
        Add (a, a') to AliasTable;
    Append (φ(a), newVal) to RedoLog of Wrap[w];
    *a' = newVal; || Write to aliased location in DRAM
```

- wrapRead reads from the Alias Table if entry is present or else SCM.
- -wrapClose performs a persistent mem fence.

RESULTS

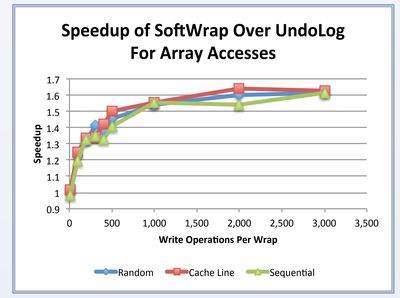


- WrAP can lazily clean up the log area from the Alias Table.

The lazy software WrAP can perform with close to the same speedup as the hardware WrAP implementation.

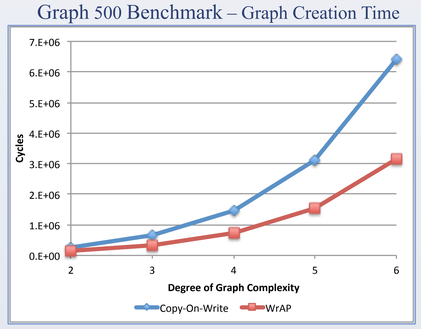
RESULTS

Data intensive applications utilizing Storage Class Memories for persistence will benefit from Write Aside Persistence.

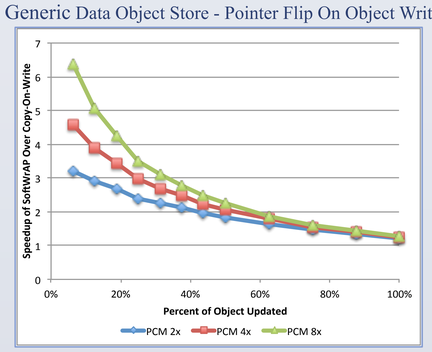


- Graph algorithms are a core to many analytic workloads.
- Graph 500, a large HPC graph benchmark, was announced at ISC2010 and appeared SC2010.

Graph 500 Benchmark – Graph Creation Time



Generic Data Object Store - Pointer Flip On Object Write



CONCLUSIONS

Write Aside Persistence (WrAP) is promising for Storage Class Memory.

- WrAP lets the cache hierarchy continue doing what it does best:
  - Reducing memory access latency
  - Performing lightweight inter-transaction communication
- No disruptive changes to the cache hierarchy
- Avoids front end synchronous operations as with Copy-On-Write
- Allows both Byte Addressability & Persistence of SCM devices
- WrAP has software only solution based on a DRAM Alias Table that shows significant performance gains.