**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 or the main memory bus. However, SCM is not as widely used yet due to higher cost and lower density. WrAP automatically and implicitly catches cache evictions to SCM between a WrAP open and close to prevent overwriting previous values.

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

### 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.

### RESULTS
Data intensive applications utilizing Storage Class Memories for persistency will benefit from Write Aside Persistence.

---

**PROBLEM**

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

<table>
<thead>
<tr>
<th>Crash Cause</th>
<th>Persistent Memory in Inconsistent State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Store A, 1</td>
<td>A 1</td>
</tr>
<tr>
<td>Store B, 2</td>
<td>B 2</td>
</tr>
<tr>
<td>Store C, 3</td>
<td>C 3</td>
</tr>
<tr>
<td>Store D, 4</td>
<td>D 4</td>
</tr>
</tbody>
</table>

---

**RELATED WORK**

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

**CONTROL**

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

**Background Path**

- Buffer
- All writes must flush to Log

**Victim Persistence Cache**

- Contains excited entries from open WrAPs

**Log**

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

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

---

**EXPERIMENTAL SETUP**

Experimental Setup:

- For A Dynamic Binary Instrumentation tool.
- DRAMSim2 – A cycle accurate memory simulator.
- WrAP API implementation

Log Cleanup Options:

- Lazy
- DRAM Alias Table
- SCM Log Area

**RESULTS**

- WrAP can lazily clean up the log area from the SCM.
- The lazy software WrAP can perform with close to the same speedup as the hardware WrAP implementation.

---

**CONCLUSIONS**

- WRAP lets the cache hierarchy continue doing what it does best:
  - Reducing memory access latency
  - Performing lightweight inter-process 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.

---

**Overview**

- WrAP automatically and implicitly catches cache evictions to SCM between a WrAP open and close to prevent overwriting previous values.
- WrAP time approaches the non-transactive time where there are no atomic guarantees in the transaction by the application.
- Related Work Atomic 8-byte write to Storage Class Memory is needed.