Understanding and Improving Persistent Transactions on Optane™ DC Memory

Pantea Zardoshti and Michael Spear, Lehigh University
Aida Vosoughi and Garret Swart, Oracle Corp

IPDPS
May 2020
The Gap Between Promise and Reality

Traditional persistence:
- Expensive
- Slow
- Complex programming
The Gap Between Promise and Reality

Optane™ DC promises:

• High Capacity and Affordable Memory
• High Performance Storage
• Native persistence
The Gap Between Promise and Reality

Does Optane™ DC deliver on these promises?

If not, what can we do about it?
Outline

• Background on Non-Volatile Memory
  • Persistent Programming Models
  • Logging

• Optane™ DC Characteristics and Performance

• Proposals to Improve Optane™ DC Performance

• Conclusions
Persistent Programming Models

- Implement a file system on the NVM
  - Simple approach
  - No changes to the program are needed

- Transactional interface
  - Programmers mark the regions of code that access NVM
  - A run-time system tracks accesses to NVM within those regions
The Persistent Programming Model

- Accessing Persistent Memory:
  - Map a file from NVM to the program virtual address space via DAX
  - Identify NV regions by marking lexically scoped transactions
  - Loads/stores are performed by persistence library
Understanding and Improving Persistent Transactions on Optane™ DC Memory

global int x = 0, y = 0;
...
persistent {
  x = 5; y = x;
}

Logging

Undo Algorithms

Redo Algorithms

Abort

Write-back

Commit

Clean up Log

Write-back

clwb (cl) 

2W + 2

sfence (sf) 

W + 3

cl

X = 0

Y = 0

cl

X = 0

Y = 0

cl

X = 0

Y = 0

sf

X = 0

Y = 0

X = 5

Y = 0

X = 5

Y = 0

X = 5

Y = 0

X = 5

Y = 0

X = 5

Y = 5
Outline

• Background on Non-Volatile Memory

• Optane™ DC Characteristics and Performance
  • Operating Modes
  • Persistent Domains
  • Performance

• Proposals to Improve Optane™ Performance

• Conclusions
Optane™ Operating Modes

Optane DC has two modes:
**App Direct**: Optane is a persistent resource, separate address space from DRAM.
Optane™ Operating Modes

Optane DC has two modes:

**Memory Mode**: Memory controller uses DRAM as cache for Optane
Understanding and Improving Persistent Transactions on Optane™ DC Memory
Optane™ Persistent Domains (eADR)

Add Low-Level ISA for Ordering

N/A
  - caches flush to NVRAM on PLI
  - require an energy source to save data

Understanding and Improving Persistent Transactions on Optane™ DC Memory
Experimental Setup

• Platform
  • LLVM TransMem – works with LLVM 5+
  • Makalu: Fast Recoverable Allocation of Non-volatile Memory

• Environment
  • 16 cores/32 threads 2.30GHz Intel Xeon Gold 5218
  • Linux kernel 4.14, LLVM-Clang 6.0, -O3 optimization.
  • 192GB RAM (not persistent, however, clwb incurs accurate latencies.)
  • 1.5 TB Optane™ DC

• Algorithms
  • Undo-based PTM (U), Redo-based PTM (R)
  • Use a scalable STM under the hood, so that we can saturate the memory bus
  • Dynamic transaction optimizations [PACT 2019]

• Benchmark
  • Write-only: TPCC (New Order), TATP (Location Update), B+-Tree
  • Vacation (travel reservation)
  • Memcached
Do Ideal Results Translate to Optane™ DC Systems?

- Redo logging outperforms undo logging
- eADR still below ideal
- Throughput is consistently lower
  - Optane™ DC latencies higher than DRAM
- Scalability drops earlier
  - Ratio of commits to aborts
  - Write throughput
Optane™ DC Performance Challenges

- Bounded WPQs limit scalability
- Optane has higher write latency than DRAM
- The read latency of random Optane™ DC memory loads is 3x slower than DRAM
The Gap Between Promise and Reality

- Optane™ ADR and eADR do not leverage DRAM to accelerate program reads

<table>
<thead>
<tr>
<th>App Direct Mode</th>
<th>Memory Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Persistent</td>
<td>Volatile</td>
</tr>
<tr>
<td>High Latency</td>
<td>Low Latency</td>
</tr>
</tbody>
</table>

**New Platforms**

- Persistent
- Low Latency
Outline

• Background on Non-Volatile Memory
• Optane™ DC Characteristics and Performance
• Proposals to Improve Optane™ Performance
  • New Operating Modes
  • Evaluation
• Conclusions
New Mode: PDRAM

- Memory controller behaves like eADR
  - Enough reserve power to flush caches / no fences or flushes

- DRAM behaves like Memory Mode
  - All of DRAM serves as a cache of persistent Optane DC

- On system failure
  - Enough reserve power to write back all dirty DRAM pages to Optane DC

- Impact
  - Store entire data set in DRAM, persist in background
New Mode: PDRAM-Lite

- Memory controller behaves like eADR
  - Enough reserve power to flush caches / no fences or flushes

- Small amount of DRAM behaves like Memory Mode
  - A few MB worth of DRAM pages serve as a cache of persistent Optane DC

- On system failure
  - Enough reserve power to write back special DRAM pages to Optane DC

- Impact
  - Less power-hungry than PDRAM
  - Working set stays in DRAM
Evaluation of PDRAM-{Lite}

- PDRAM matches DRAM performance
  - Until Optane scalability bottlenecks

- PDRAM-Lite outperforms Optane eADR

- Optane throughput depends on memory access pattern
  - Moving the redo log into DRAM does not have a significant impact on latency
Impact of Workload Size (Memcached)

Throughput (Kop/s) vs. Workload Size (GB) for Memcached.
Outline

• Background on Non-Volatile Memory
• Optane™ DC Characteristics and Performance
• Proposals to Improve Optane™ Performance
• Conclusions
Conclusions

• Optane™ DC largely delivers on its promises

• Sources of overhead go beyond fences and flushes
  • Limited caching in App Direct mode prevents avoiding these other overheads

• Reserve power has an unexpectedly significant impact
  • Instrumentation
  • Performance
  • Opportunity to create new operating modes
Thank you

zardoshti@lehigh.edu

Video