MPSoC'15

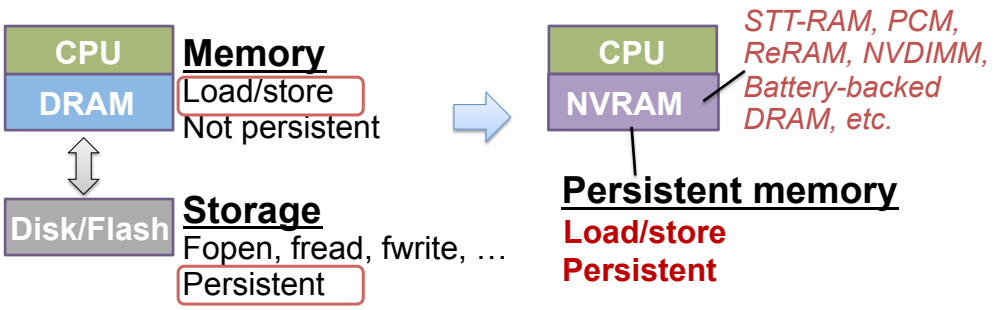
Memory Persistence: A New Dimension in Memory System Design

Jishen Zhao
Computer Engineering, UC Santa Cruz

July, 2015

UNIVERSITY OF CALIFORNIA SANTA CRUZ

New Design Opportunities with NVRAM

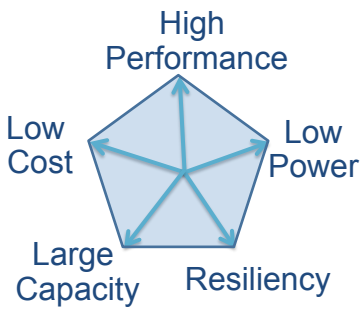


- Allow in-memory data structures to become permanent immediately
- Demonstrated 32x speedup compared with using storage devices
  [Condit+ SOSP'09, Volos+ ASPLOS'11, Coburn+ ASPLOS'11, Venkataraman+ FAST'11]

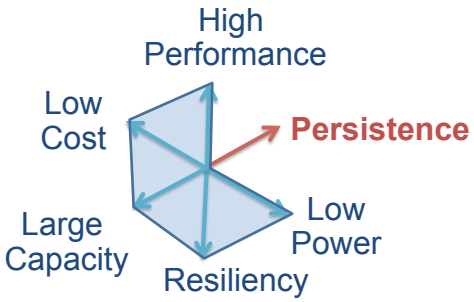
Dry It -- Memory Persistence

Persistence
- Used to be a property of storage systems
- Now needs to be maintained in the memory system

A New Dimension in Memory System Design

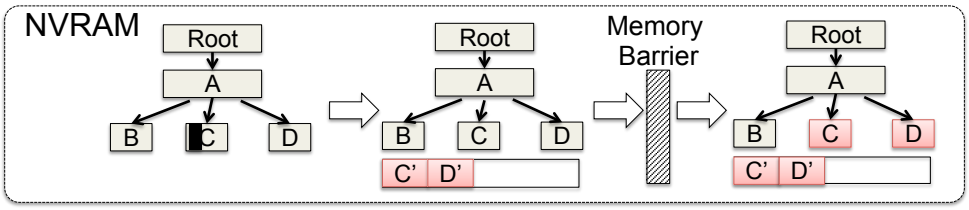


Traditional Memory Design Targets



A New Shape of Memory Design Targets

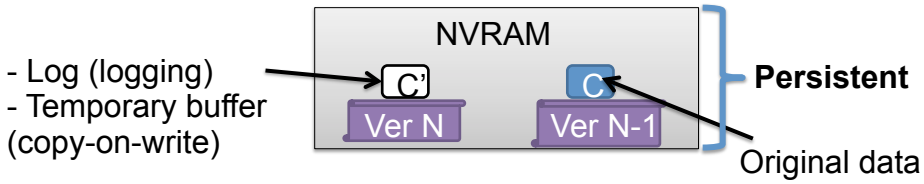
Maintaining Memory Persistence



- Can't just use an in-memory system – ***No persistence support***
  - In-place updates to nonvolatile media may be interrupted without completion in the event of power loss or system crashes
  - Interrupted writes could leave **partially overwritten data** or **missing references**

- Cant' just use a storage system – ***Not optimized for memory***
  - **Overhead** from database or file system interfaces, which assume and are optimized for slow, block-addressable devices
  - Not optimized for fast, byte-addressable memory with a load/store interface

Performance Overhead of Logging/Copy-on-Write



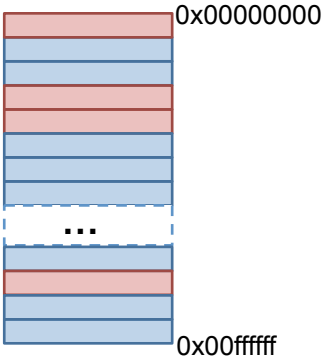
Source of overhead – Multiversioning
- Two writes per one data update
- The memory barrier disables write reordering

Memory traffic increased by 120% !

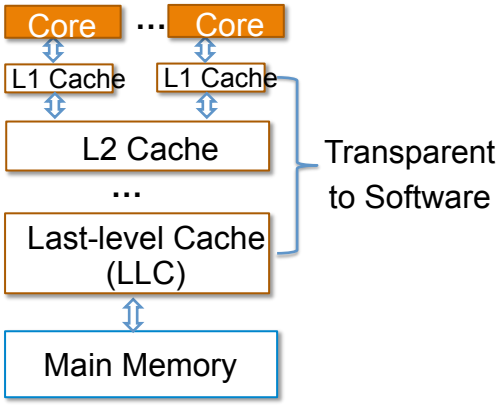
Performance degradation by 3.3x

What Can We Leverage From Hardware

Software's view of memory system
- A flat address space
- Pages

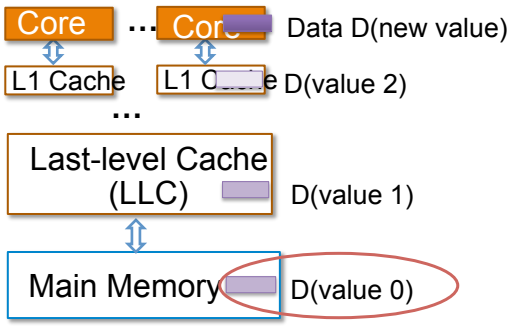


Hardware's view of memory system
- Not flat, a hierarchy
- Cache lines



Leveraging Caching

- How does a cache hierarchy work?
- A multiversioned system by nature!



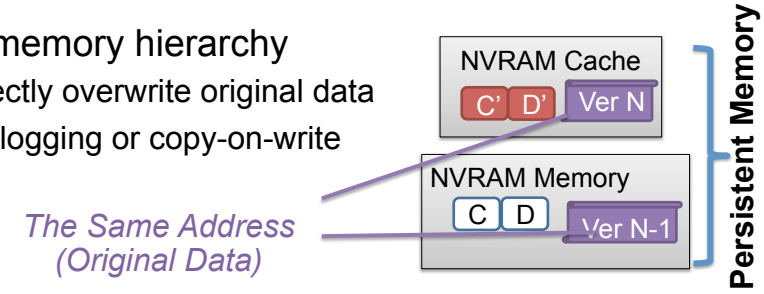
Hardware-based Persistent Memory

Key idea: Maintain multiversioning by hardware

- Leverage caching schemes to automatically maintain multiversioning
- Two levels of NVRAM (phyiscally or logically)

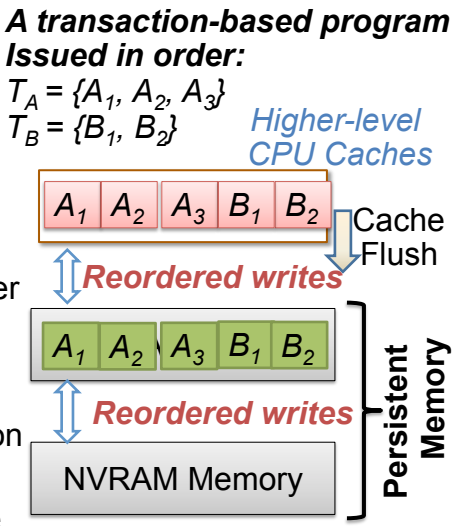
A persistent memory hierarchy

- Updates directly overwrite original data
- No need for logging or copy-on-write

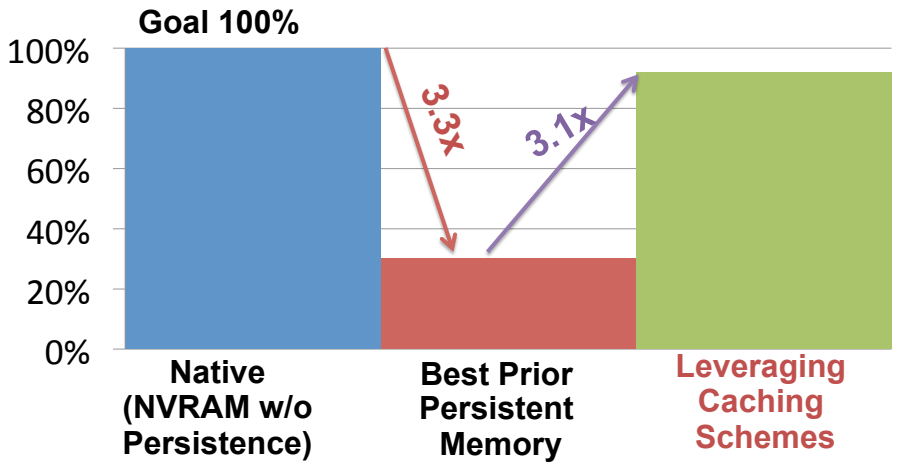


Re-enable Write Reordering

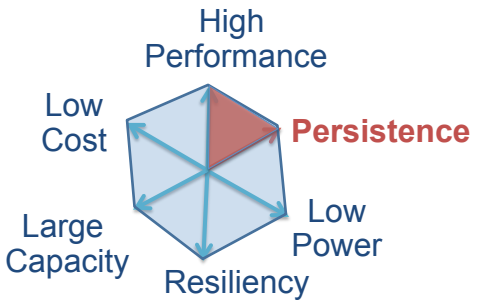
- Appeared in-order writes
  - T_A commits before T_B
- Allow write reordering (no software barrier)
  - Reorder cache writebacks
  - Reorder writes by memory controller
- Write order control
  - Mark each NVRAM cache block
  - Flush higher-level CPU caches upon transaction commits **– Very fast!**
  - Change the state of NVRAM cache blocks in order



Performance Results



Summary



Memory Persistence and Performance

Issue	Opportunity
Memory Controller Design	Hardware-based Persistence Support

Acknowledgments

Coauthors of the paper:

- Kiln: Closing the Performance Gap Between Systems With and Without Persistence Support. Sheng Li (Intel), Doe Hyun Yoon (Google), Yuan Xie (UCSB), Norm Jouppi (Google). MICRO'13. (Best Paper Honorable Mention Award)

MPSoC'15

Memory Persistence: A New Dimension in Memory System Design

Jishen Zhao
Computer Engineering, UC Santa Cruz

July, 2015