

Hands-Off Persistence System (HOPS)

Swapnil Haria¹, Sanketh Nalli¹, Haris Volos²,
Kim Keeton², Mark D. Hill¹, Mike M. Swift¹

¹ University of Wisconsin-Madison

² Hewlett Packard Labs

WHISPER Analysis

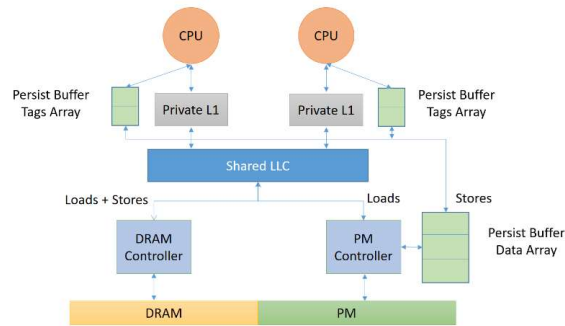
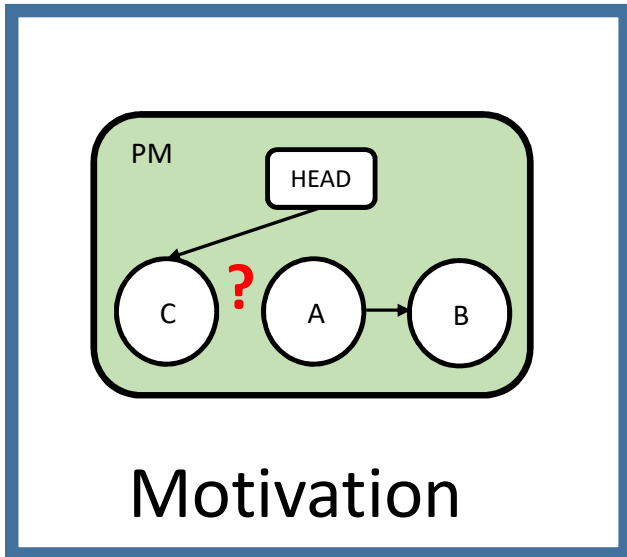
- 4% accesses to PM, 96% to DRAM
- 5-50 epochs/transaction
- Self-dependencies common
- Cross-dependencies rare



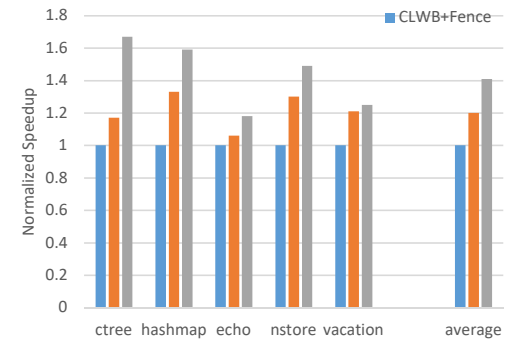
HOPS Design

- Volatile memory hierarchy (almost) unchanged
- Order epochs without flushing
- Allows multiple copies of same cacheline
- Correct, conservative method based on coherence

Outline

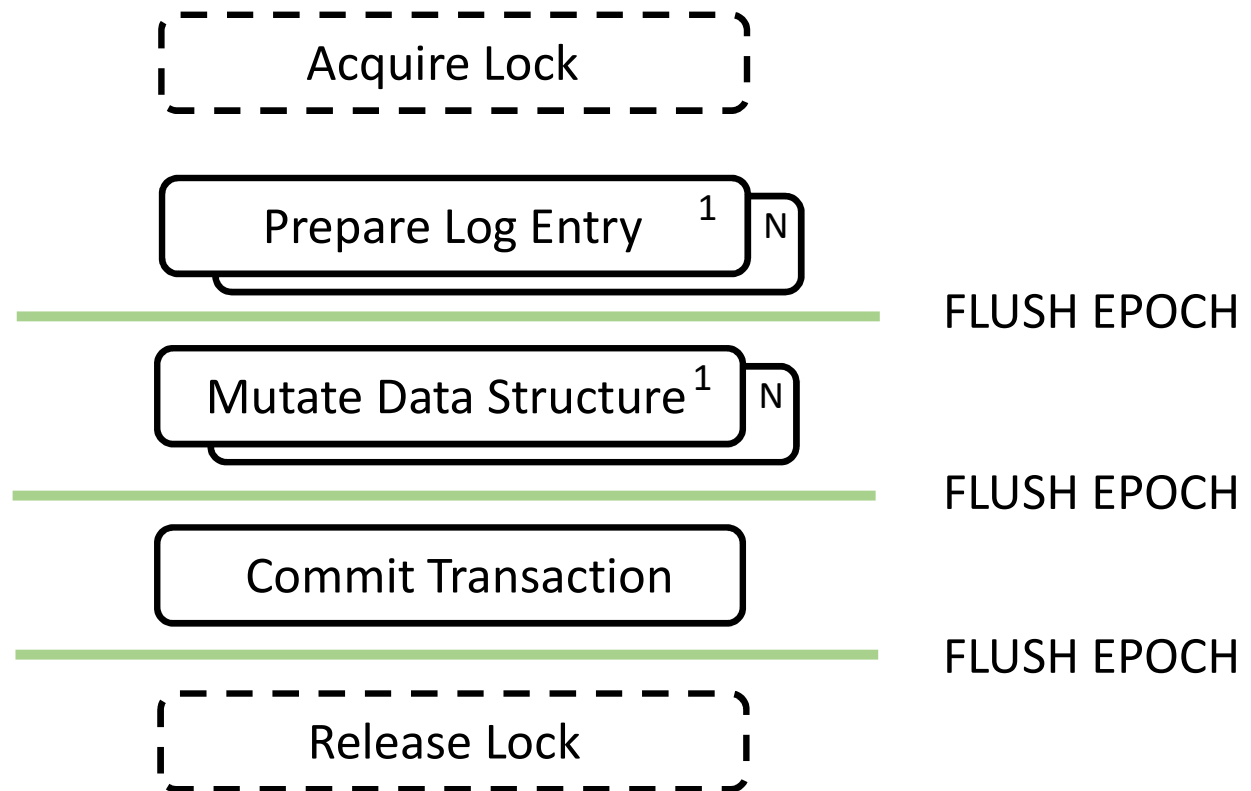


HOPS Design

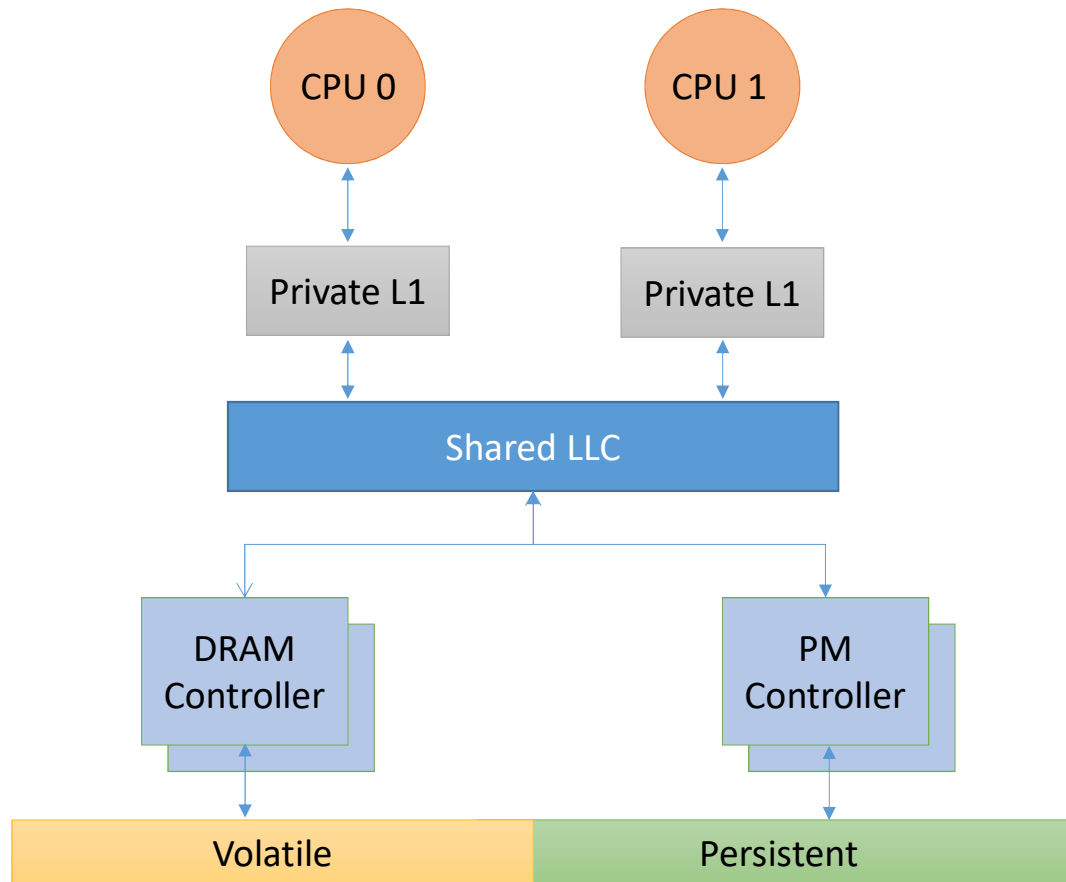


Evaluation

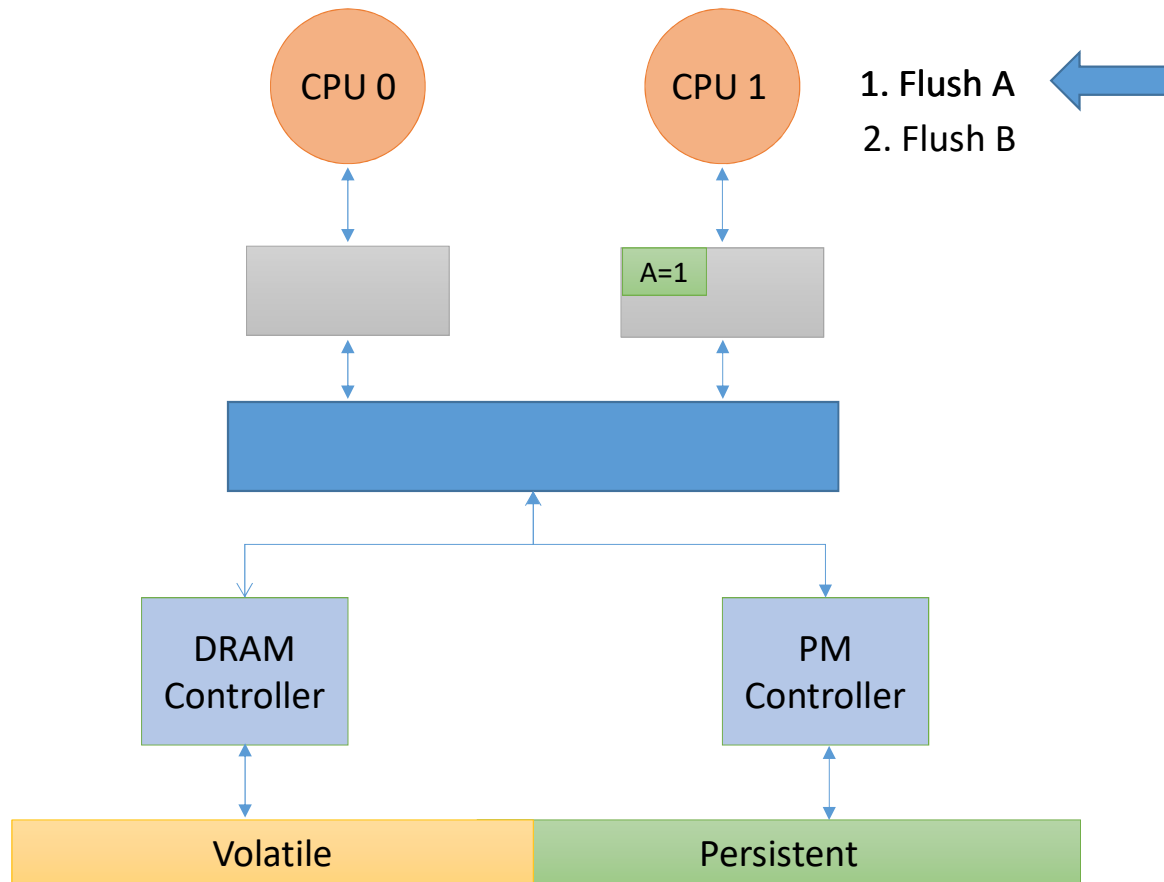
ACID Transactions (currently)



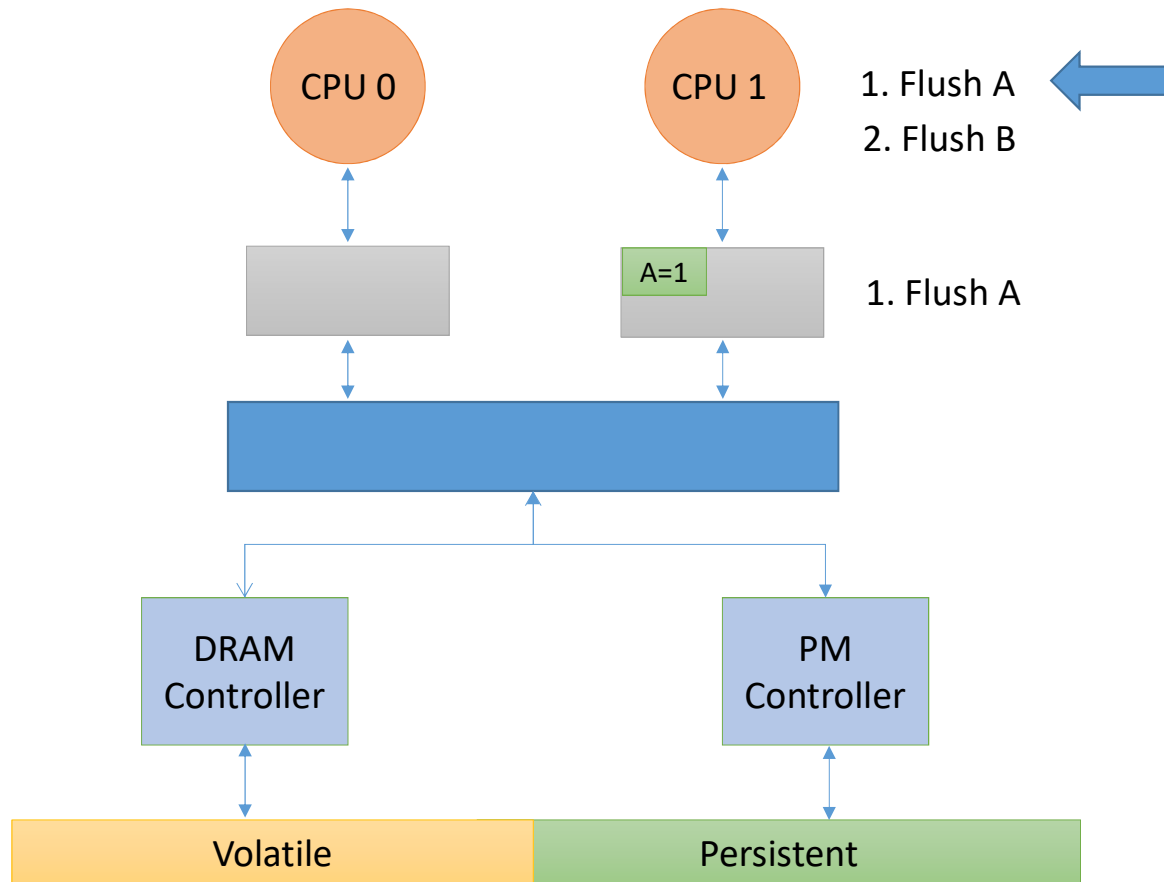
Base System



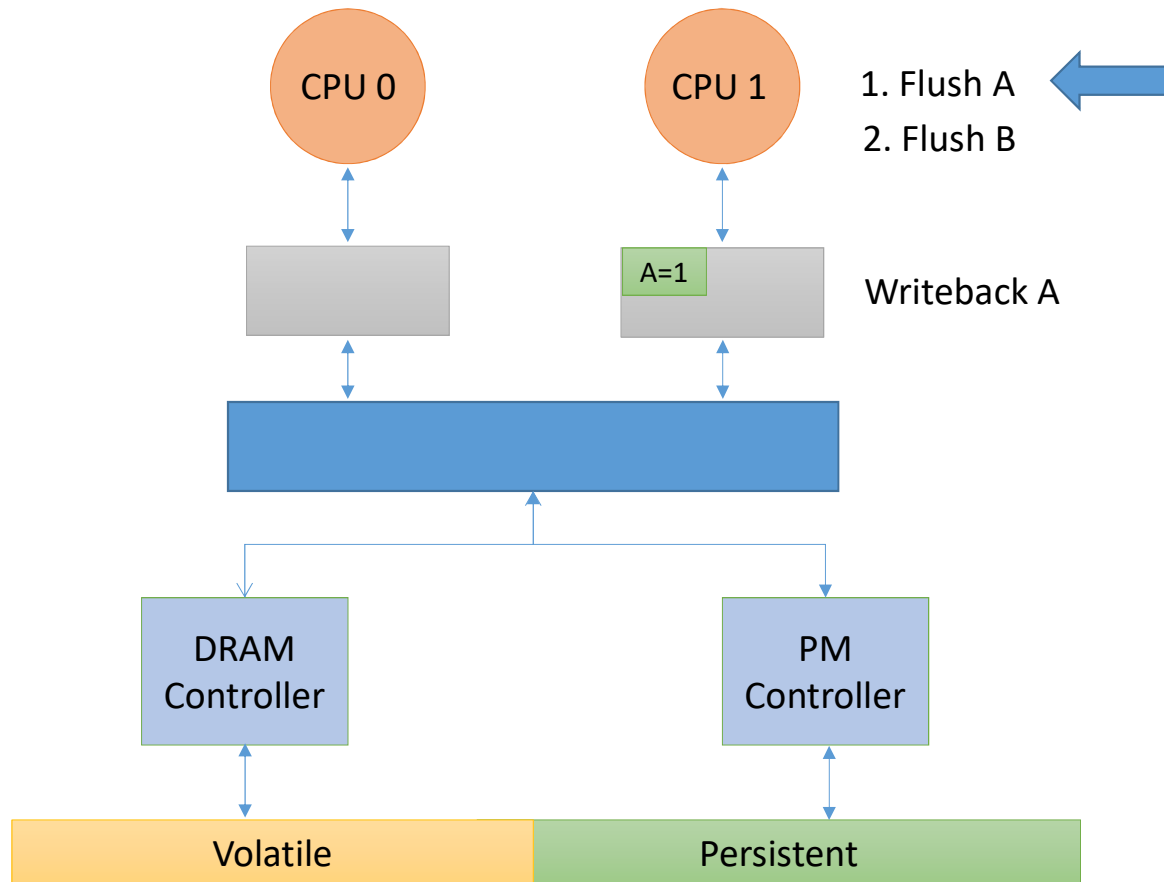
Base System: Flush



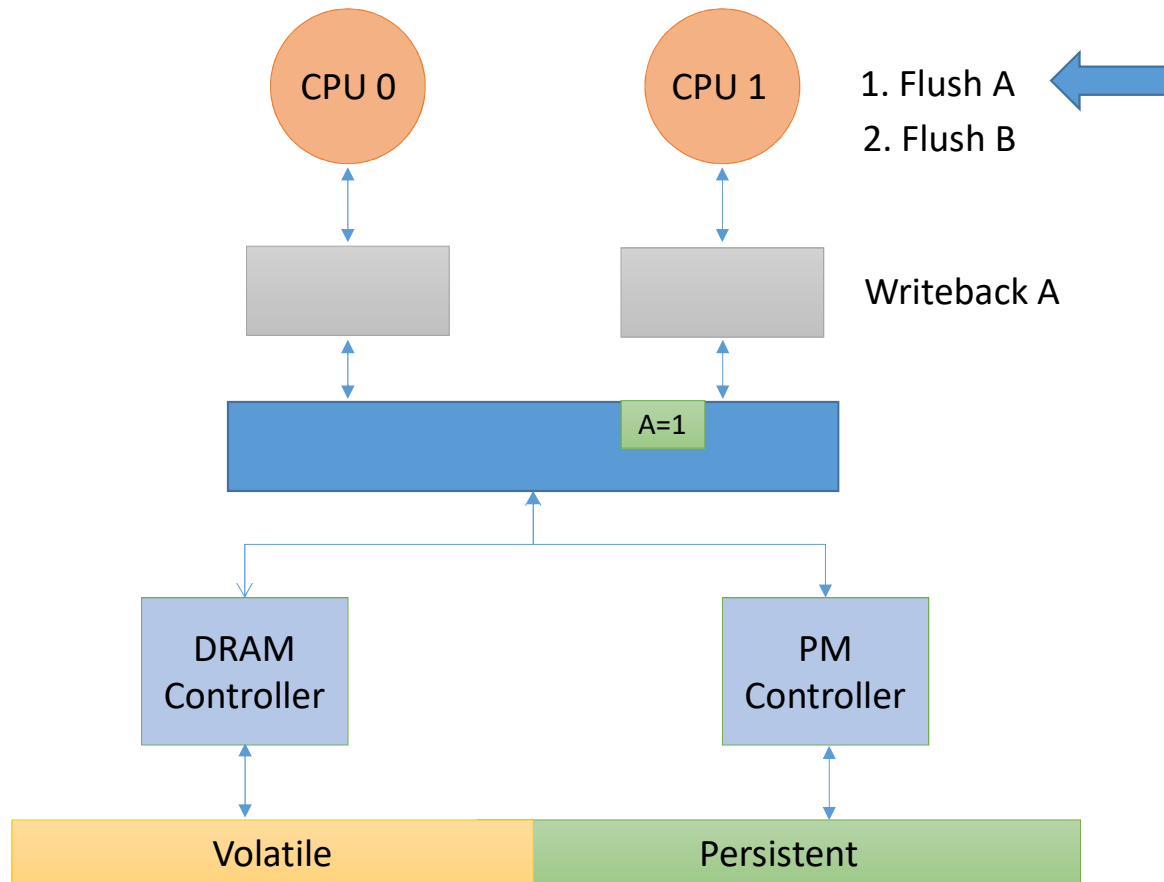
Base System: Flush



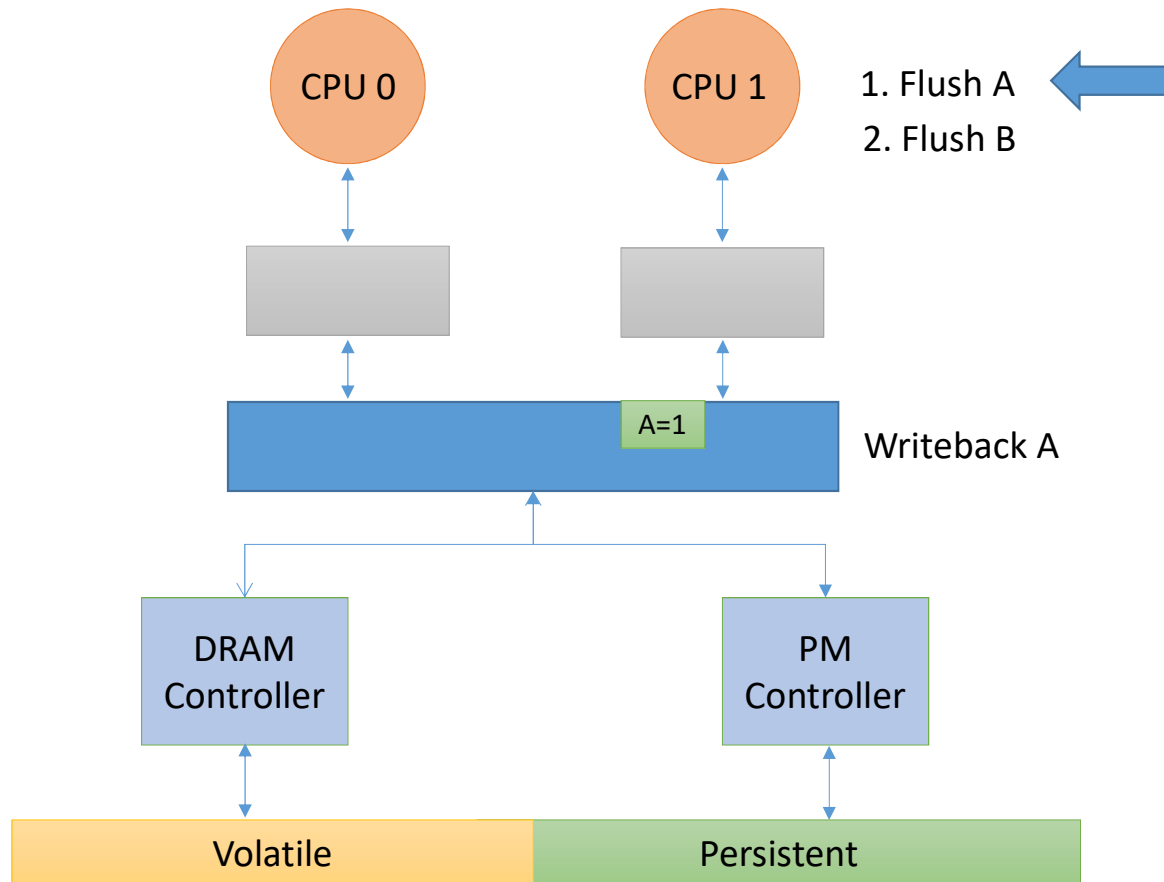
Base System: Flush



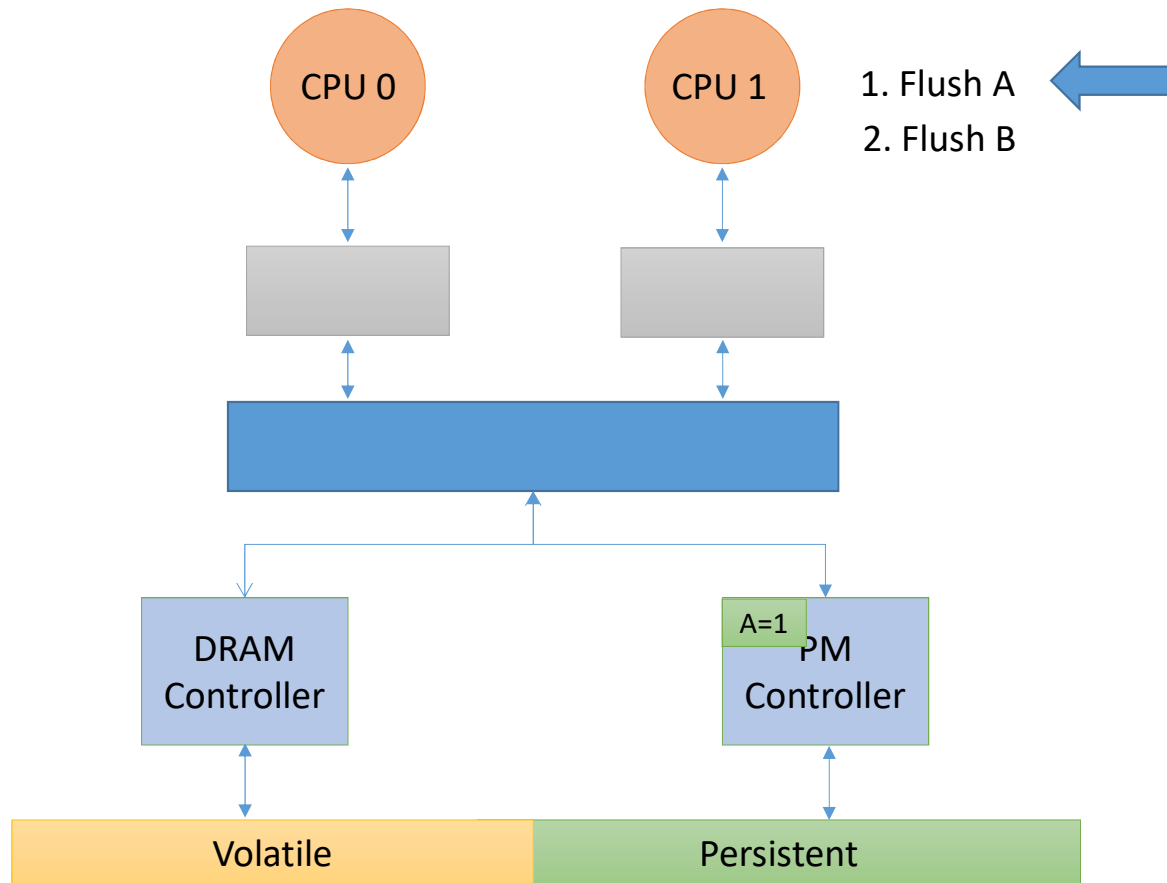
Base System: Flush



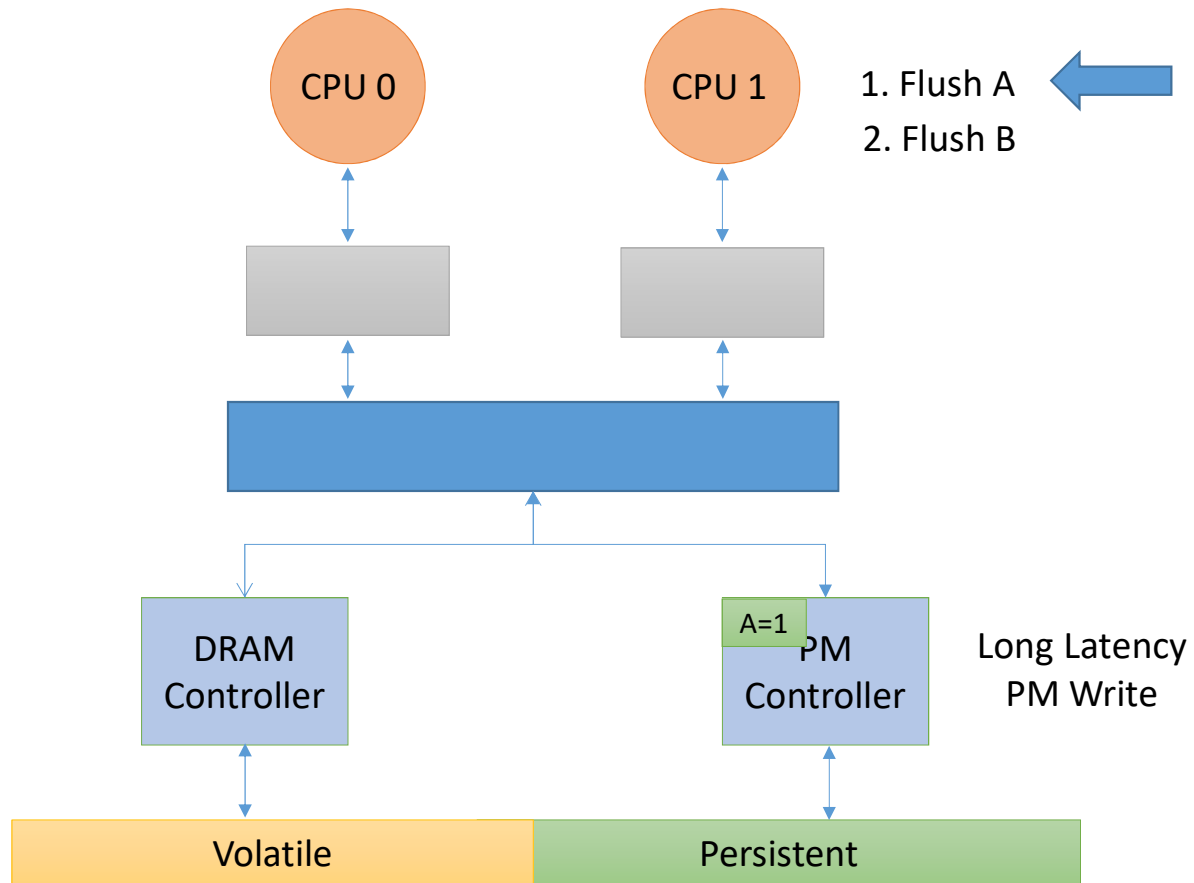
Base System: Flush



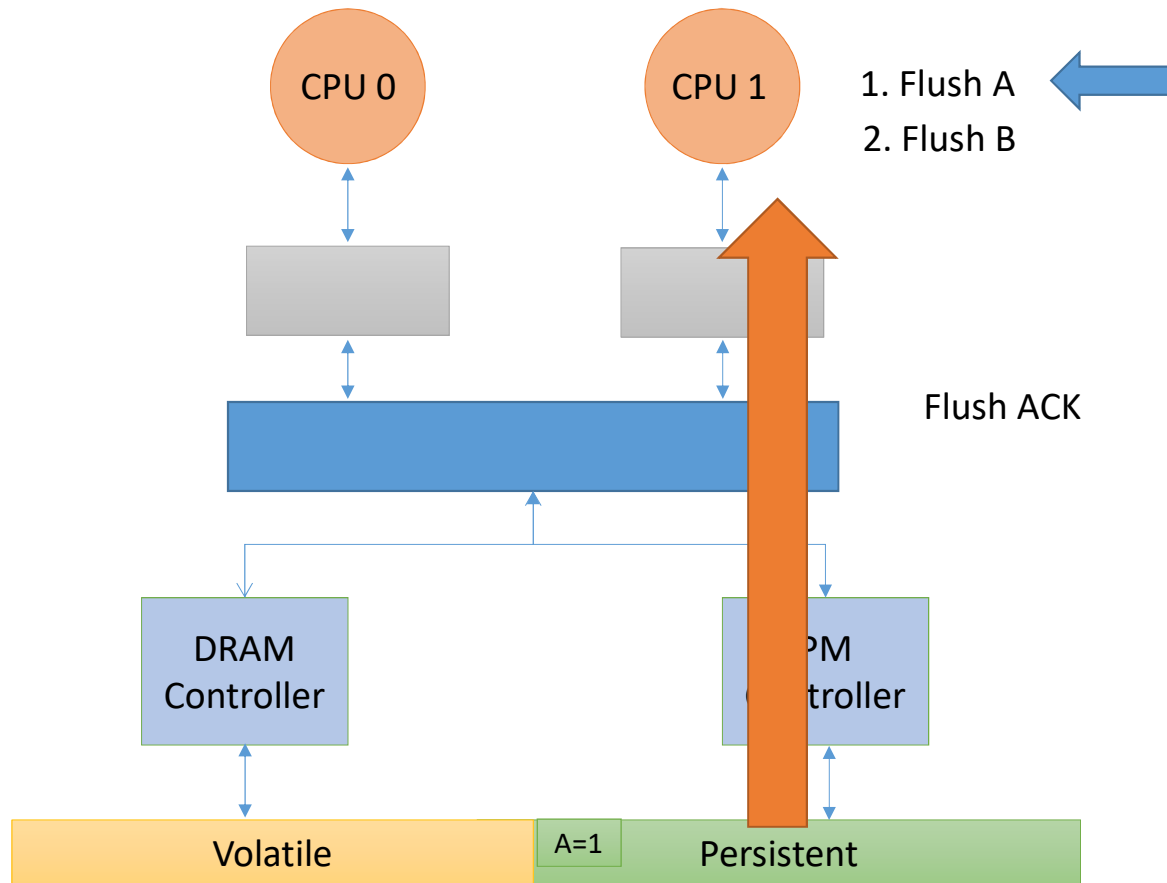
Base System: Flush



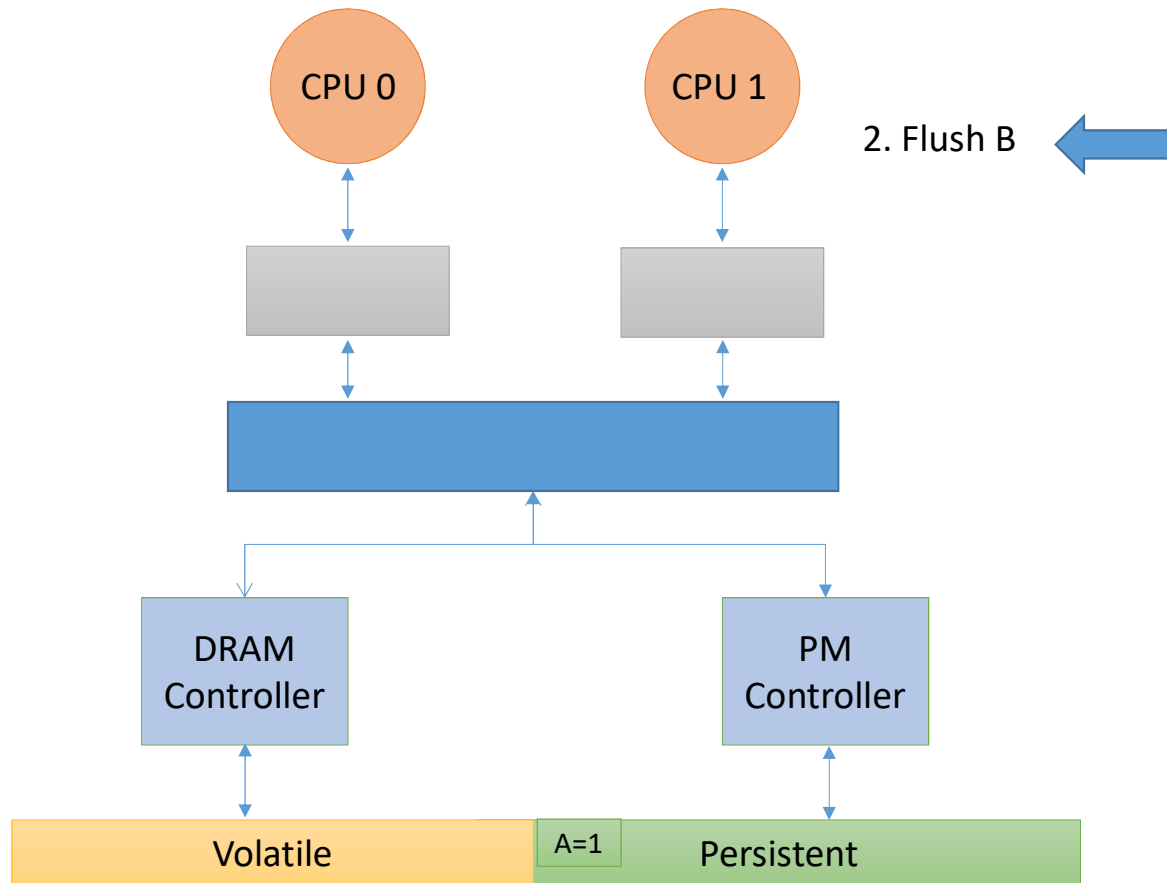
Base System: Flush



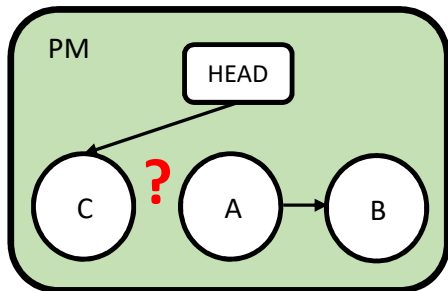
Base System: Flush



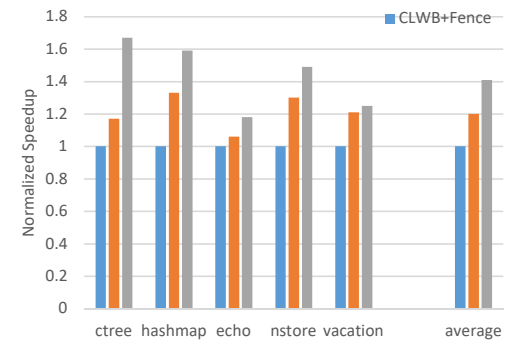
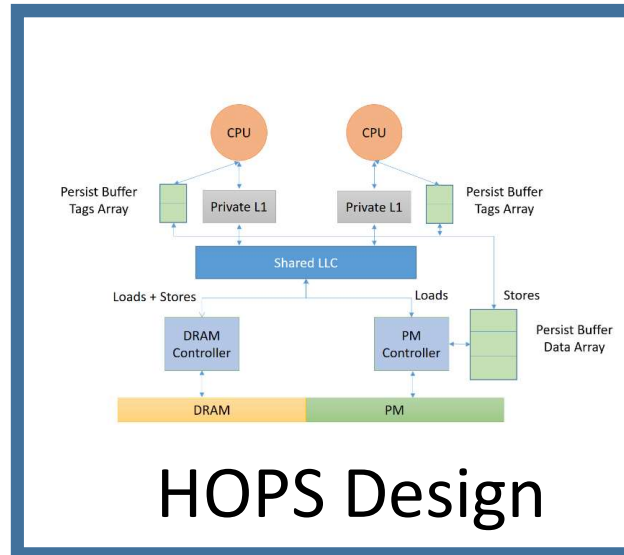
Base System: Flush



Outline



Motivation

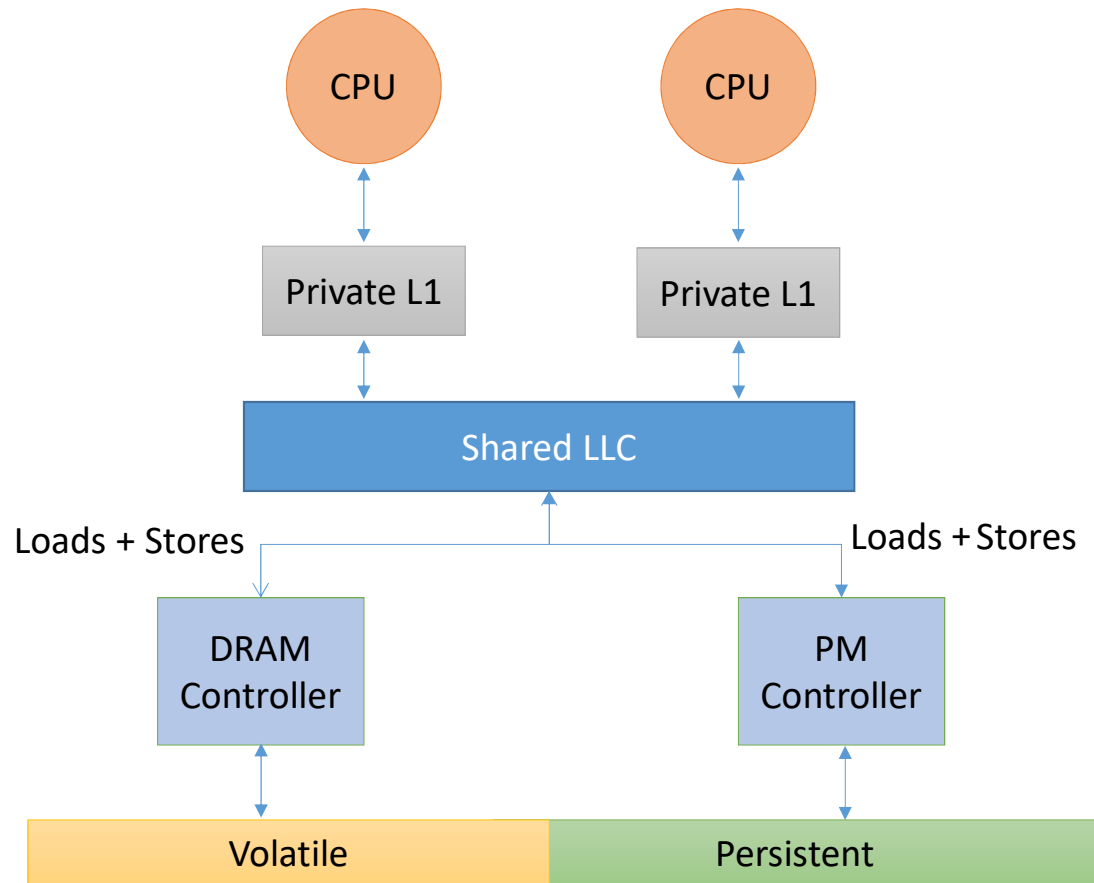


Evaluation

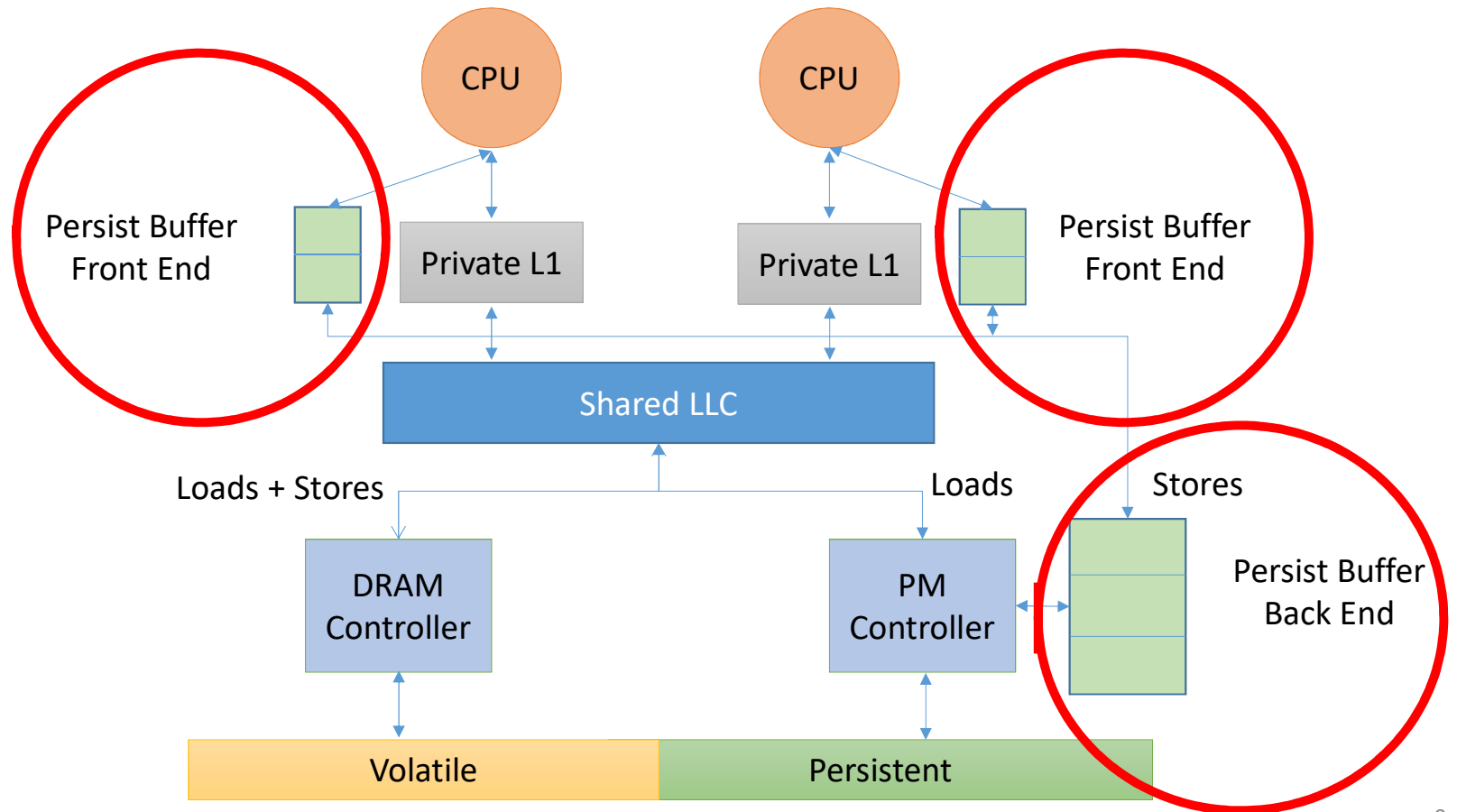
Hands-off Persistence System (HOPS)

- Volatile memory hierarchy (almost) unchanged
- Order epochs without flushing
- Allows multiple copies of same cacheline
- Correct, conservative method for handling cross-dependencies

Base System

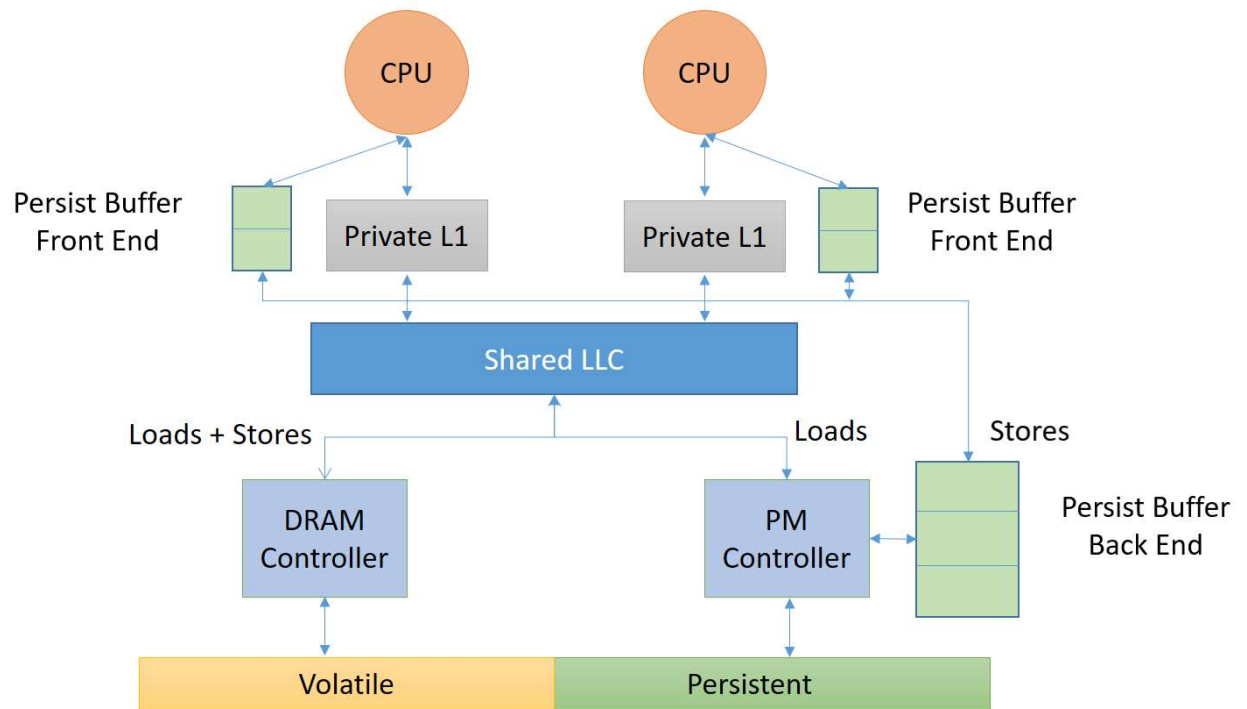


Base System + Persist Buffers



Persist Buffers

- Volatile buffers
- Front End (per-thread)
 - Address, Ordering Info
- Back End (per-MC)
 - Cacheline data
- Enqueue/Dequeue only
 - Not fully-associative

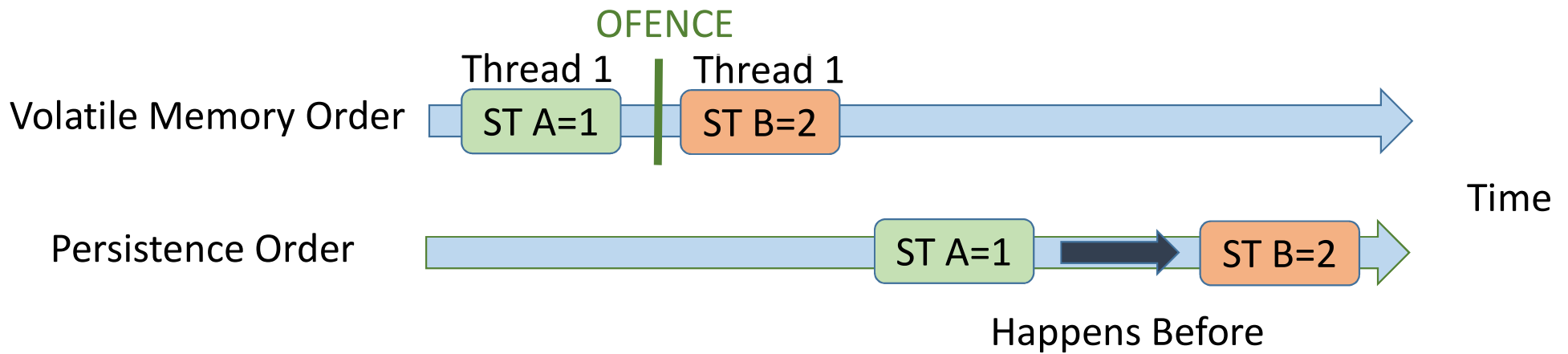


Hands-off Persistence System (HOPS)

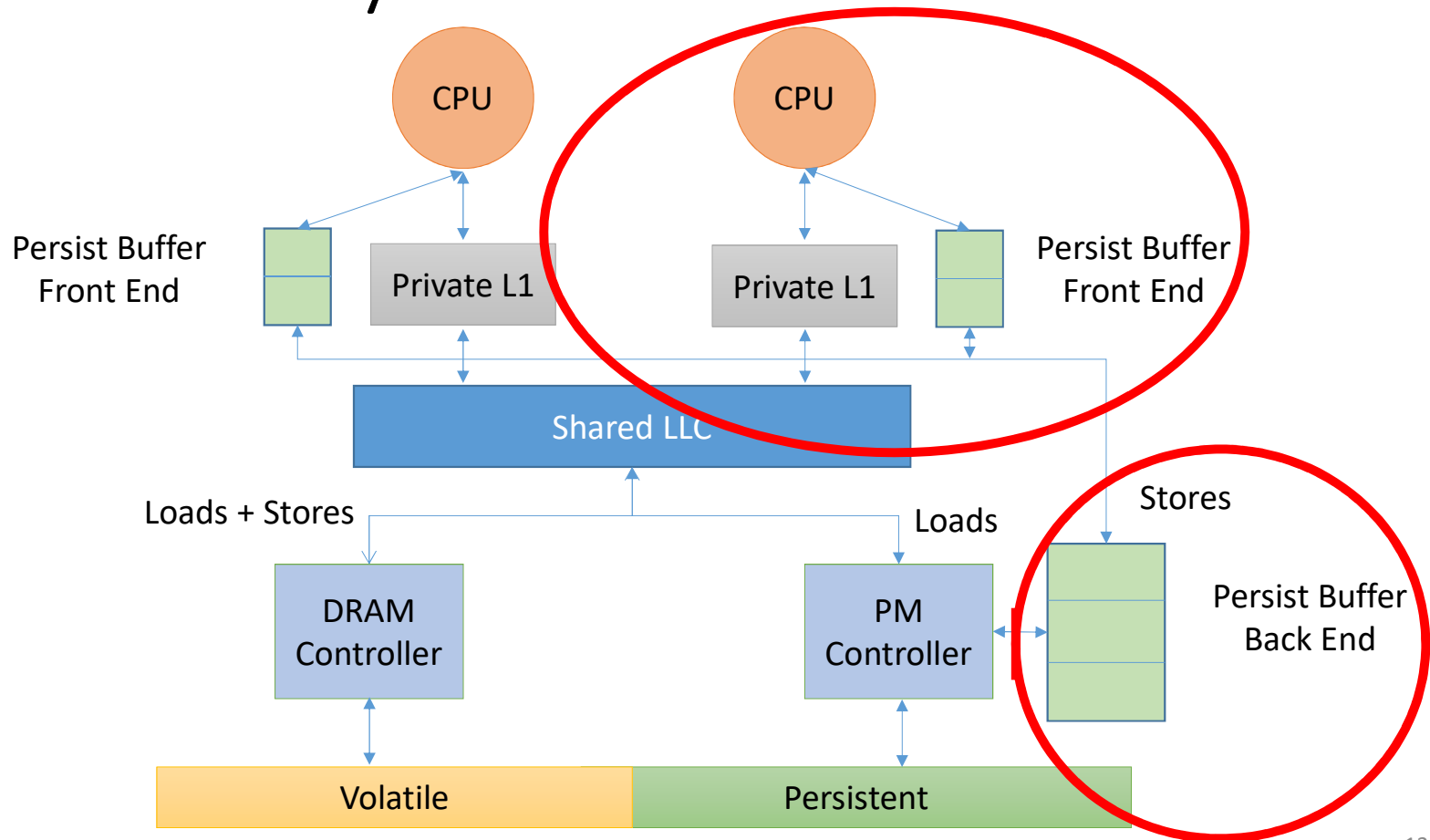
- Volatile memory hierarchy (almost) unchanged
- Order epochs without flushing
- Allows multiple copies of same cacheline
- Correct, conservative method for handling cross-dependencies

OFENCE: Ordering Fence

- Orders stores preceding OFENCE before later stores

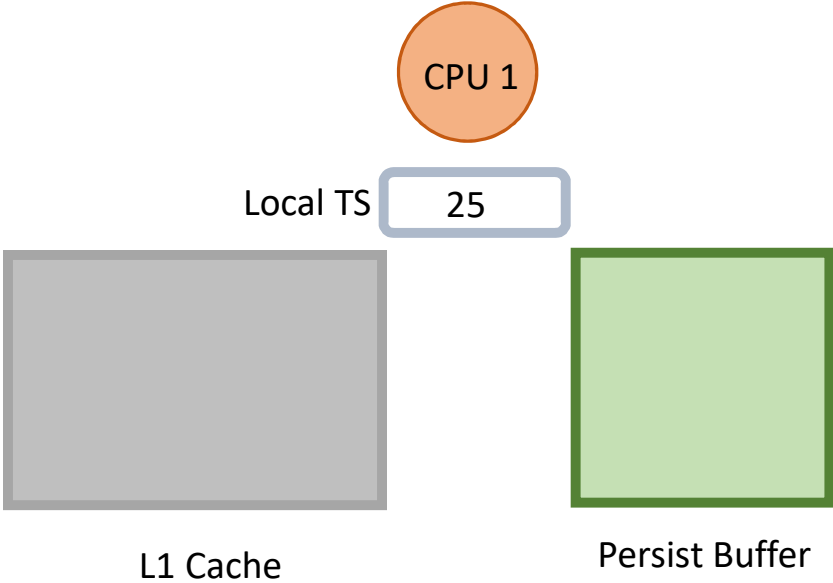


Base System + Persist Buffers



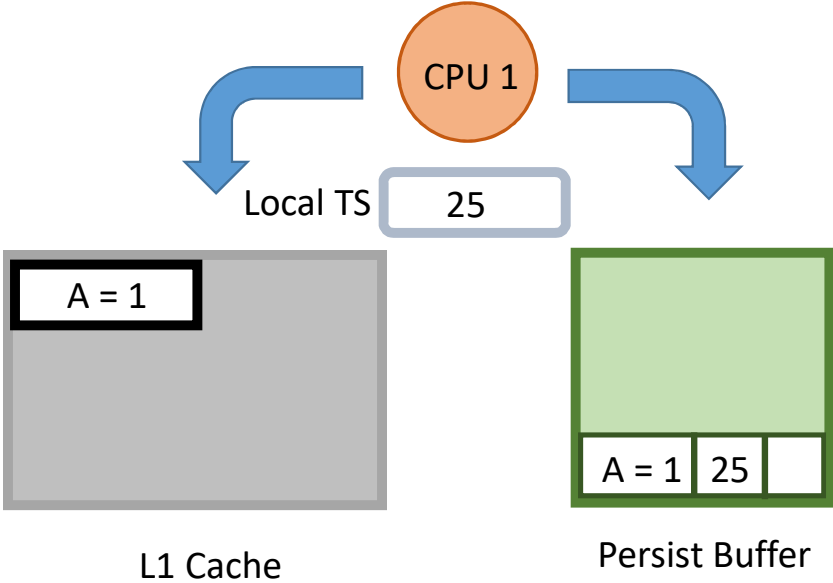
Ordering Epochs without Flushing

- 1. ST A = 1 ←
- 2. ST B = 1
- 3. LD R1 = A
- 4. OFENCE
- 5. ST A = 2



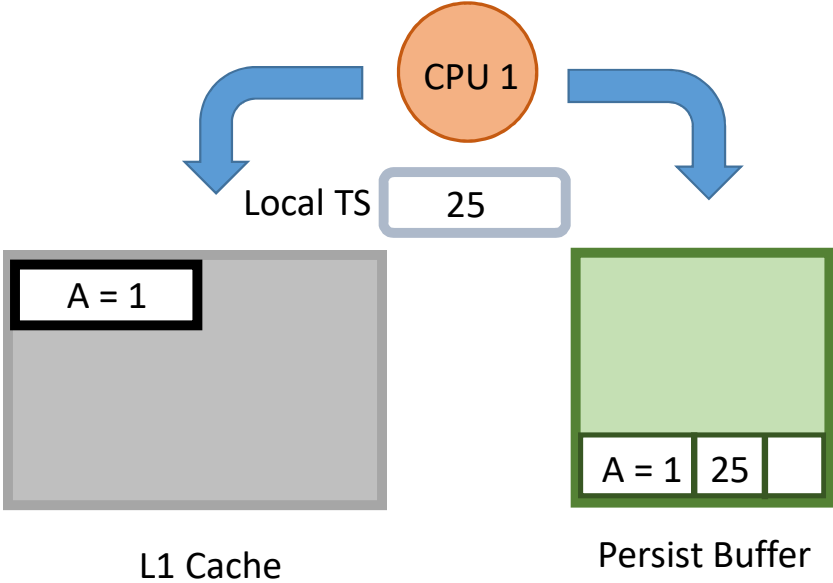
Ordering Epochs without Flushing

- 1. ST A = 1 ←
- 2. ST B = 1
- 3. LD R1 = A
- 4. OFENCE
- 5. ST A = 2



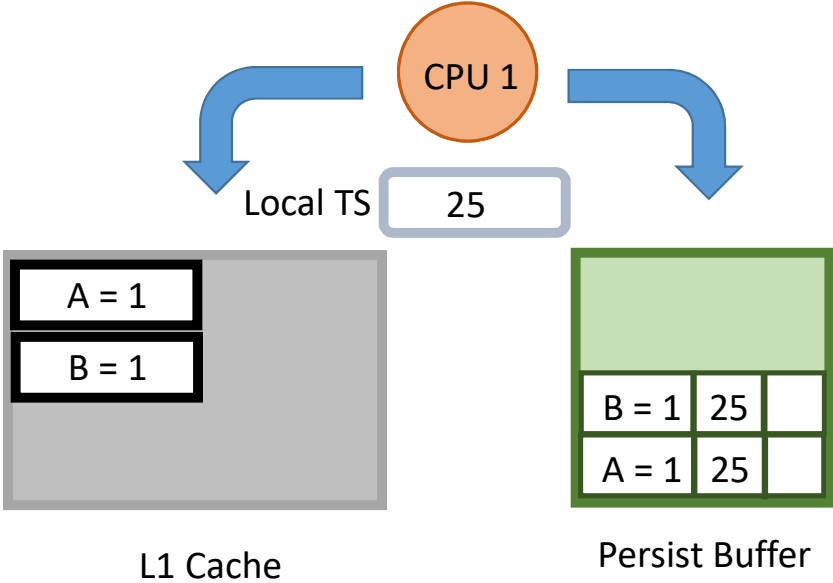
Ordering Epochs without Flushing

- 1. ST A = 1
- 2. ST B = 1 ←
- 3. LD R1 = A
- 4. OFENCE
- 5. ST A = 2



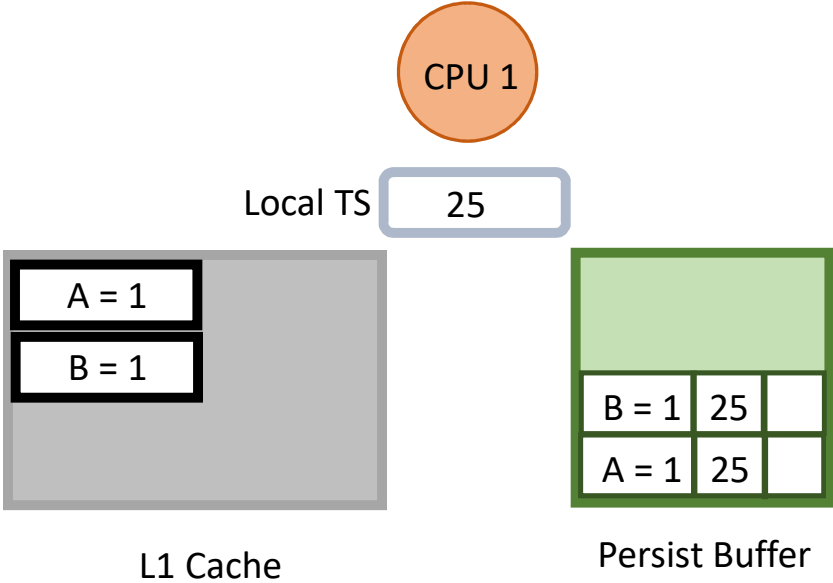
Ordering Epochs without Flushing

- 1. ST A = 1
- 2. ST B = 1 ←
- 3. LD R1 = A
- 4. OFENCE
- 5. ST A = 2



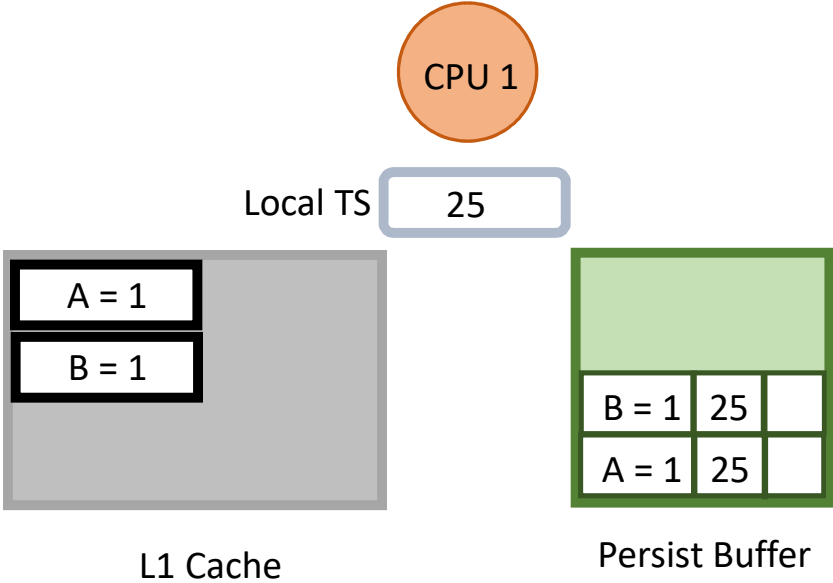
Ordering Epochs without Flushing

- 1. ST A = 1
- 2. ST B = 1 ←
- 3. LD R1 = A
- 4. OFENCE
- 5. ST A = 2



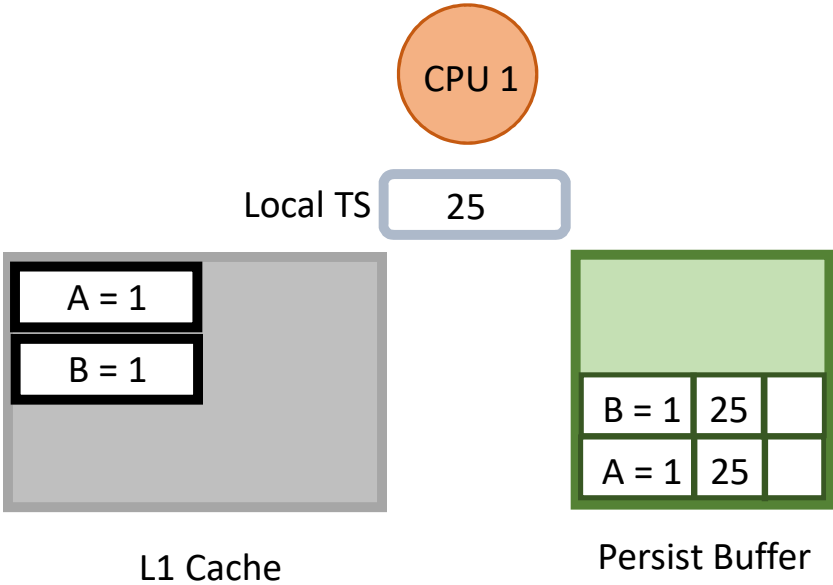
Ordering Epochs without Flushing

- 1. ST A = 1
- 2. ST B = 1
- 3. LD R1 = A ←
- 4. OFENCE
- 5. ST A = 2



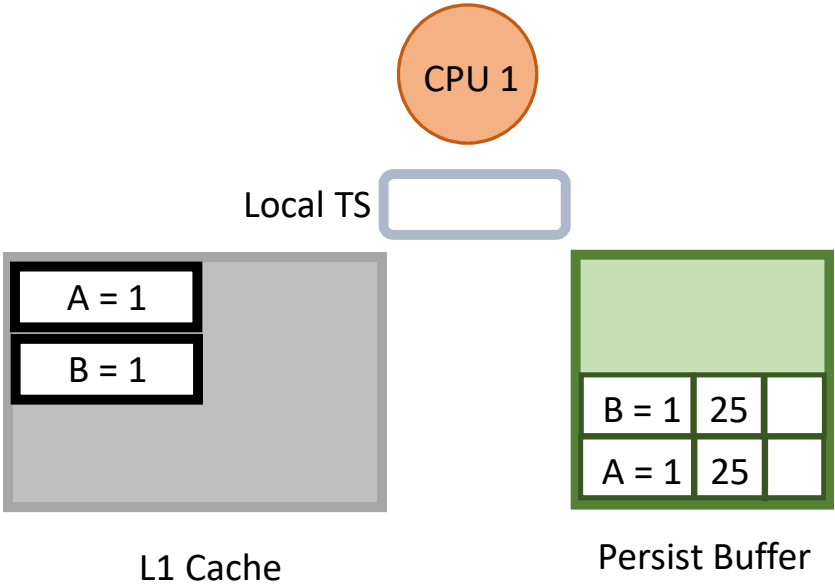
Ordering Epochs without Flushing

- 1. ST A = 1
- 2. ST B = 1
- 3. LD R1 = A
- 4. OFENCE ←
- 5. ST A = 2



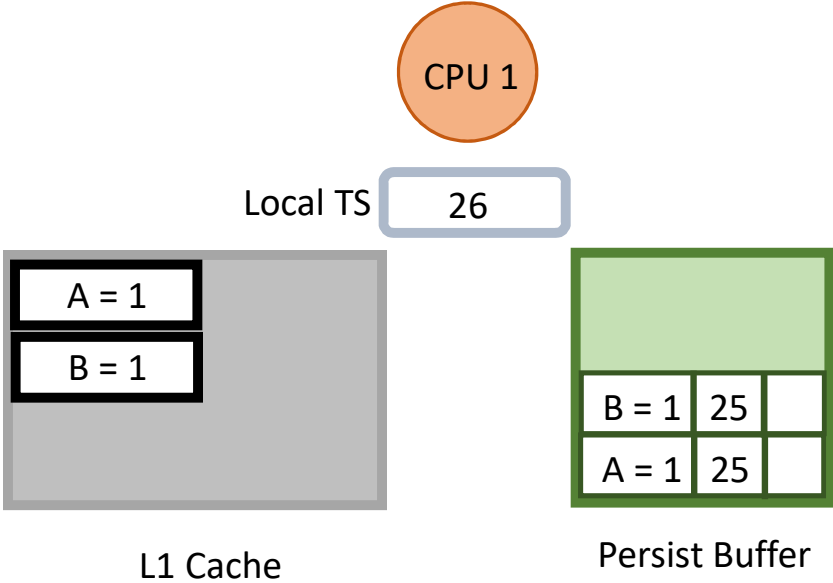
Ordering Epochs without Flushing

- 1. ST A = 1
- 2. ST B = 1
- 3. LD R1 = A
- 4. OFENCE ←
- 5. ST A = 2



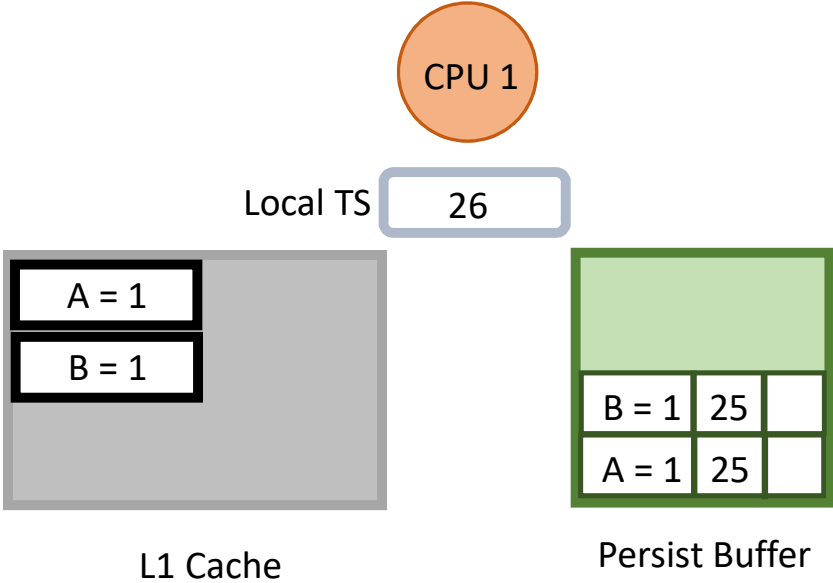
Ordering Epochs without Flushing

- 1. ST A = 1
- 2. ST B = 1
- 3. LD R1 = A
- 4. OFENCE ←
- 5. ST A = 2



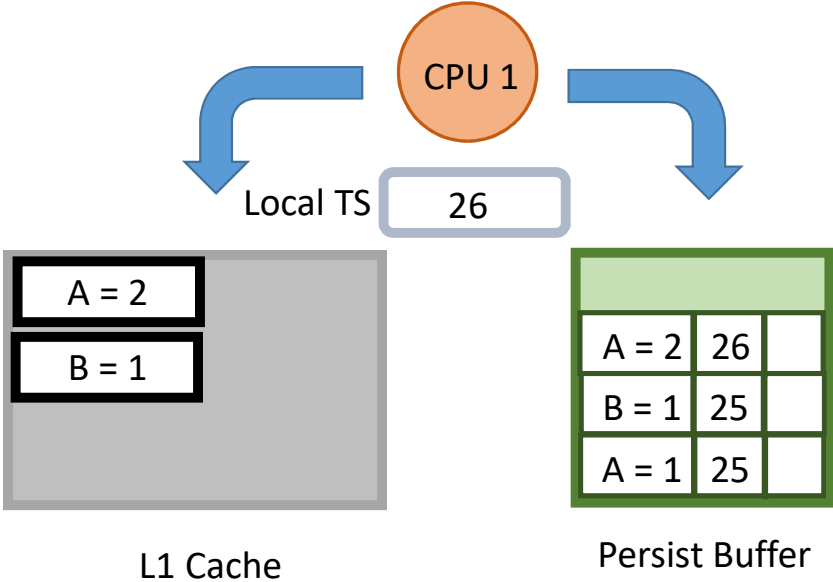
Ordering Epochs without Flushing

- 1. ST A = 1
- 2. ST B = 1
- 3. LD R1 = A
- 4. OFENCE
- 5. ST A = 2 ←

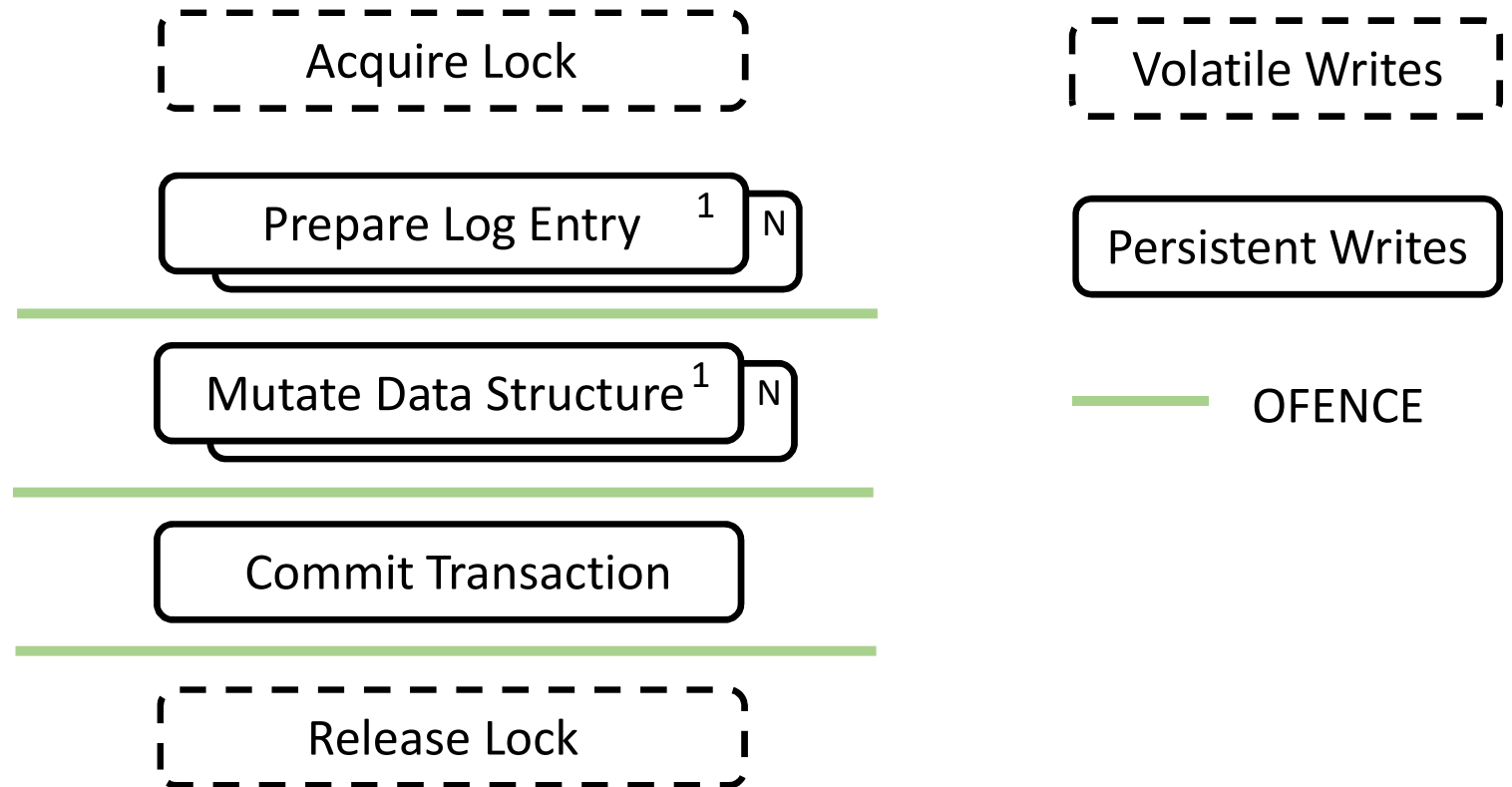


Ordering Epochs without Flushing

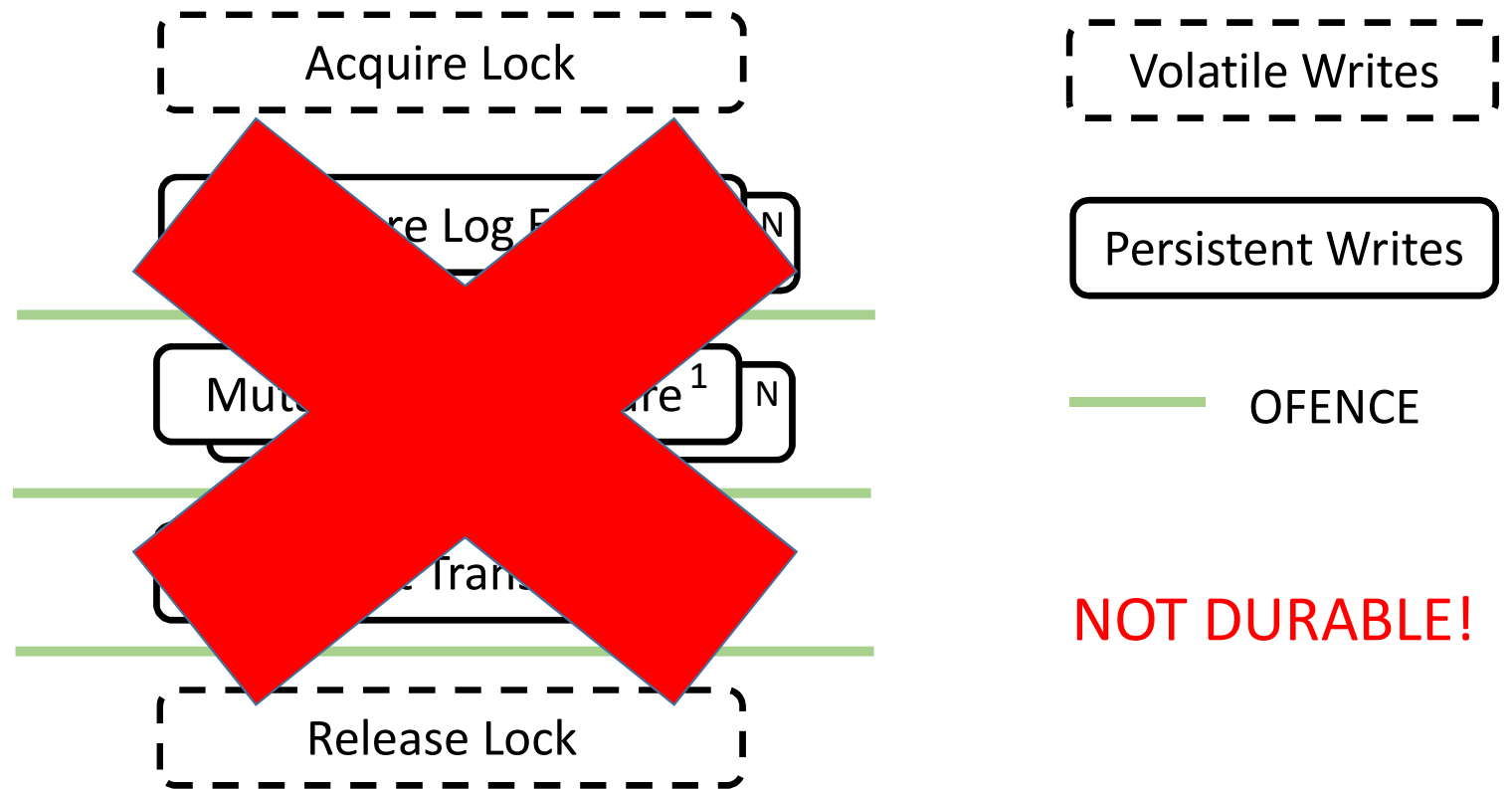
- 1. ST A = 1
- 2. ST B = 1
- 3. LD R1 = A
- 4. OFENCE
- 5. ST A = 2 ←



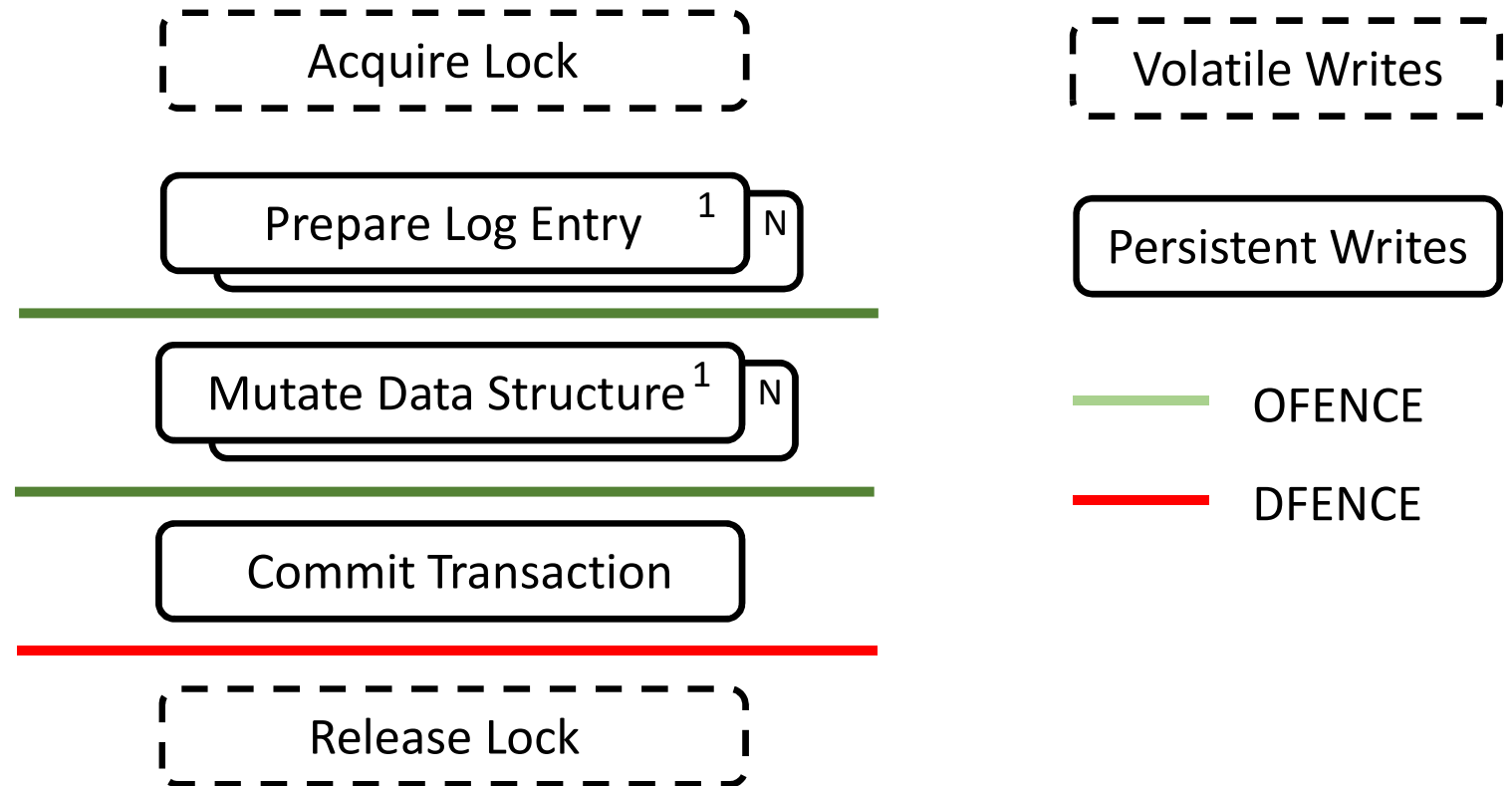
ACID Transactions in HOPS



ACID Transactions in HOPS

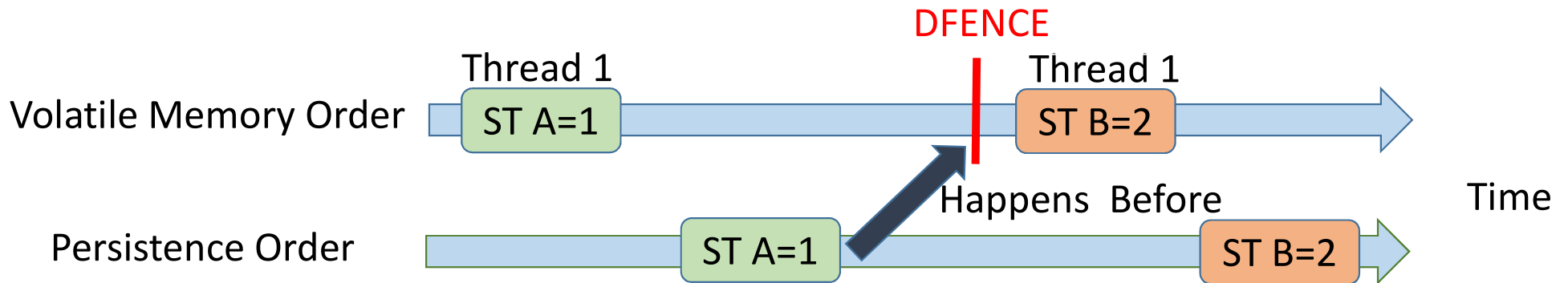


ACID Transactions in HOPS



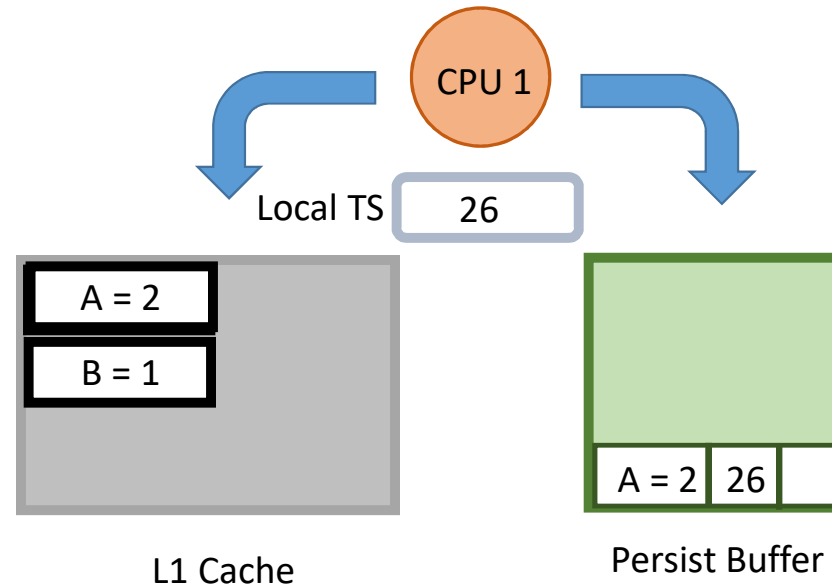
DFENCE: Durability Fence

- Makes the stores preceding DFENCE durable



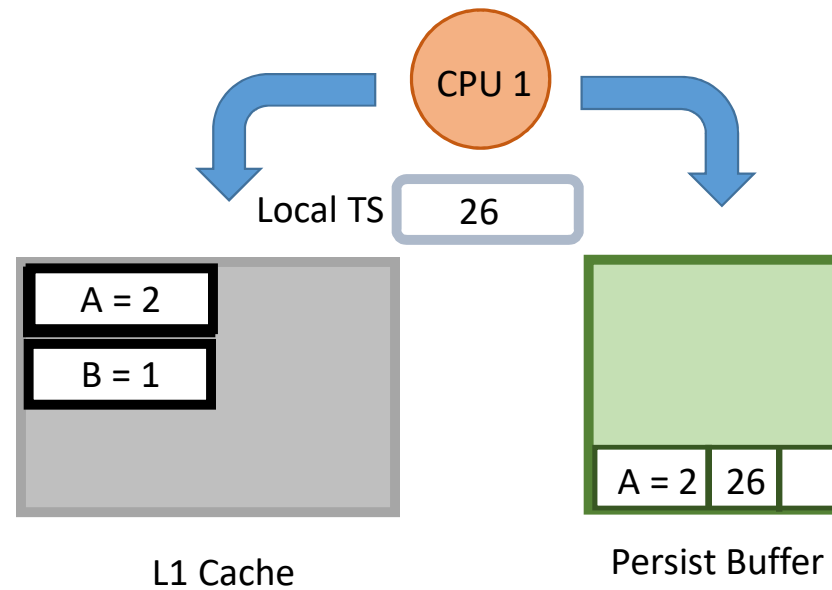
Durability is important too!

1. ST A = 1
2. ST B = 1
3. LD R1 = A
4. OFENCE
5. ST A = 2

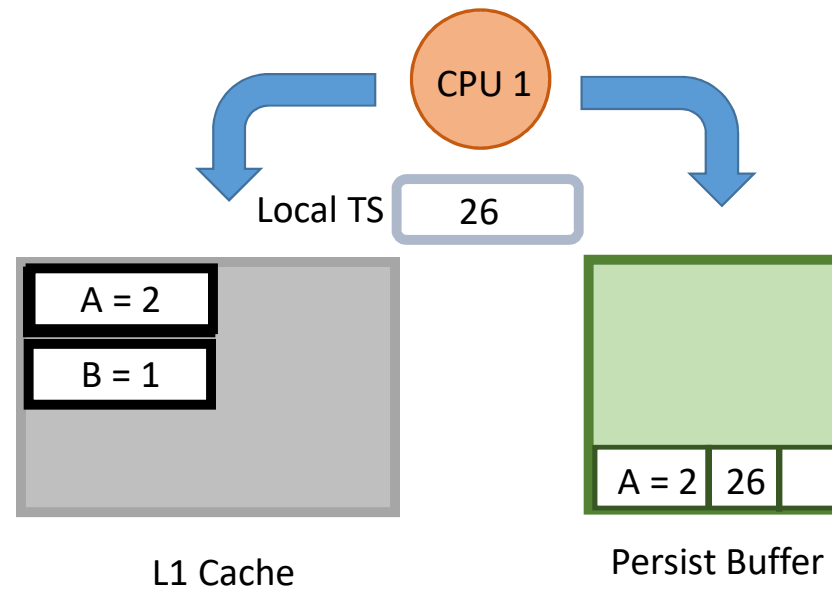


Durability is important too!

1. ST A = 1
2. ST B = 1
3. LD R1 = A
4. OFENCE
5. ST A = 2



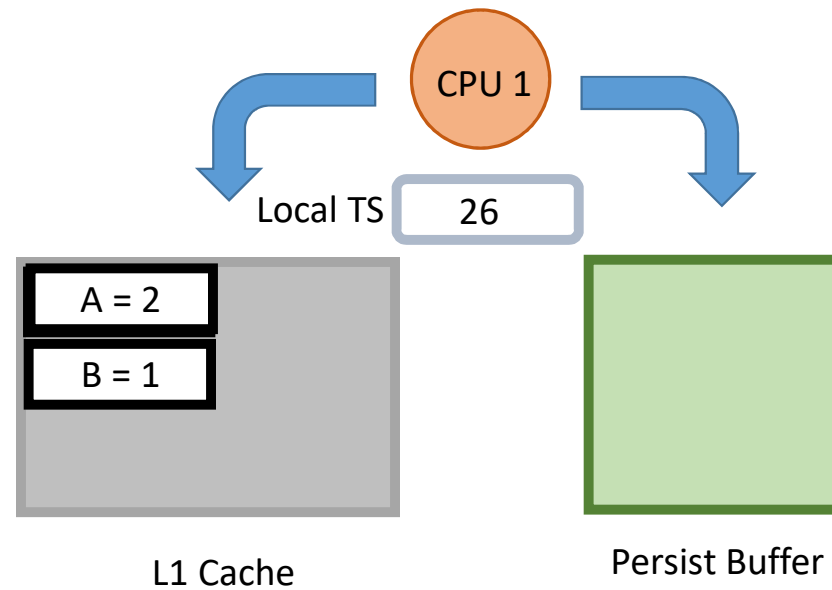
Durability is important too!



1. ST A = 1
2. ST B = 1
3. LD R1 = A
4. OFENCE
5. ST A = 2

N. DFENCE ←

Durability is important too!



1. ST A = 1
2. ST B = 1
3. LD R1 = A
4. OFENCE
5. ST A = 2

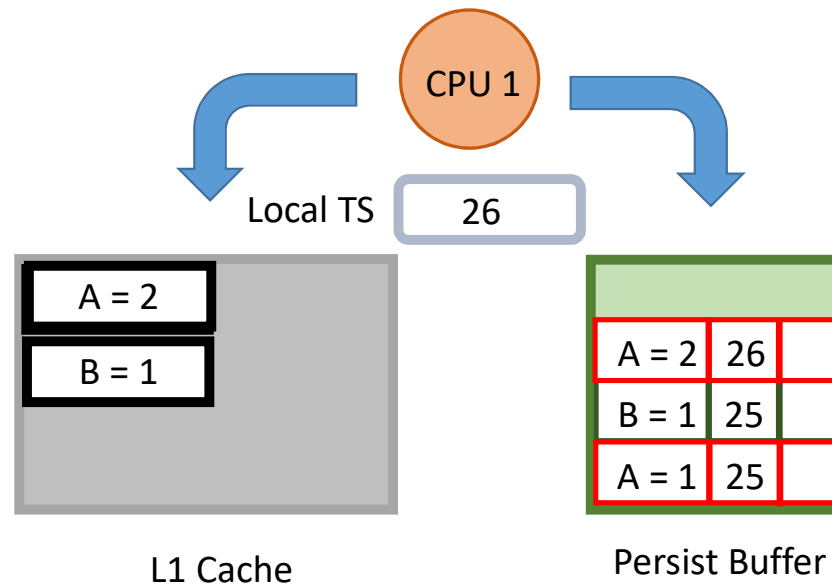
N. DFENCE ←

Hands-off Persistence System (HOPS)

- Volatile memory hierarchy (almost) unchanged
- Order epochs without flushing
- **Allows multiple copies of same cacheline**
- Correct, conservative method for handling cross-dependencies

Preserving multiple copies of cachelines

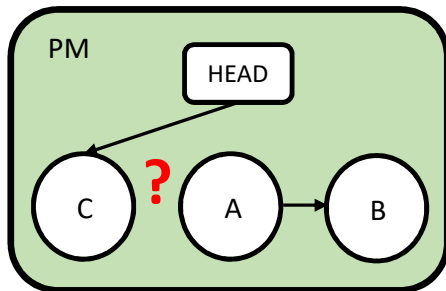
1. **ST A = 1**
2. ST B = 1
3. LD R1 = A
4. OFENCE
5. **ST A = 2**



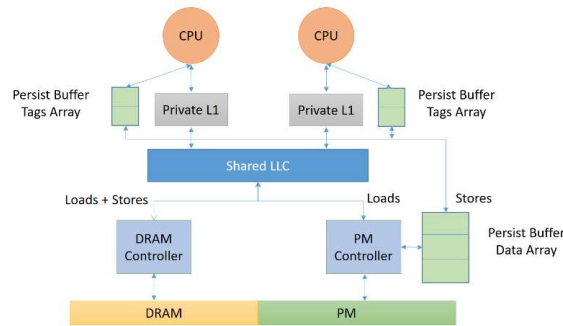
Hands-off Persistence System (HOPS)

- Volatile memory hierarchy (almost) unchanged
- Orders epochs without flushing
- Allows multiple copies of same cacheline
- Correct, conservative method for handling cross-dependencies

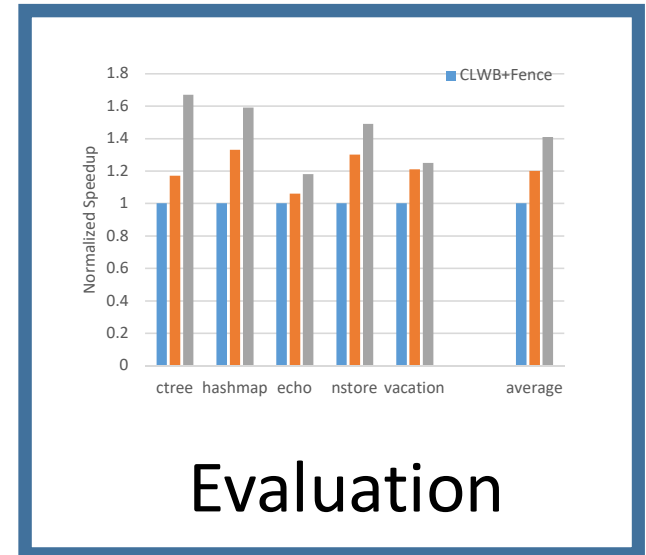
Outline



Motivation



HOPS Design



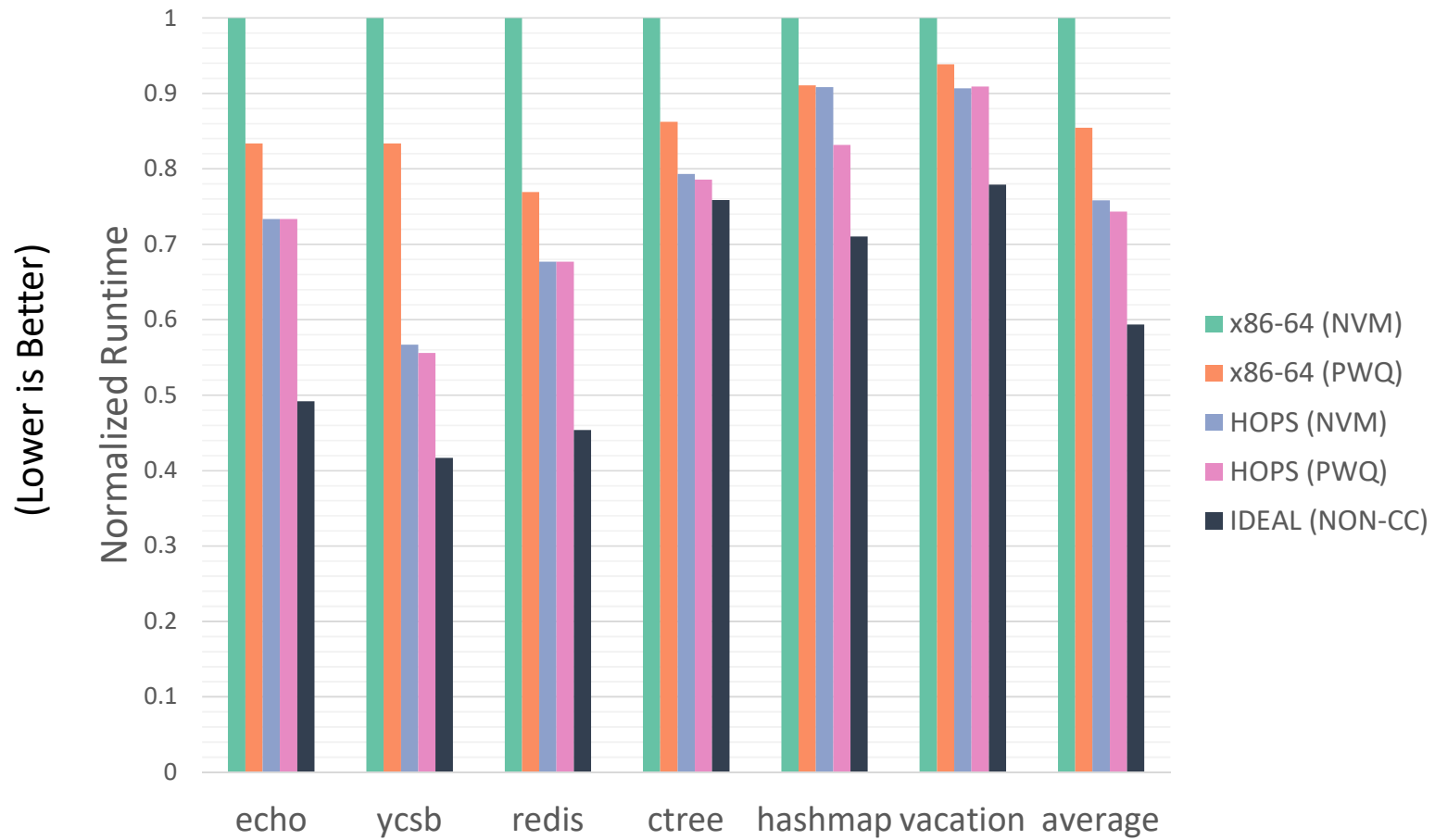
Evaluation

System Configuration

Evaluated using gem5 full-system mode with the Ruby memory model

Parameter	Setting
CPU Cores	4 cores, OOO, 2Ghz
L1 Caches	private, 64 KB, Split I/D
L2 Caches	private, 2 MB
DRAM	4GB, 40 cycles read/write latency
PM	4GB, 160 cycles read/write latency
Persist Buffers	64 entries

Performance Evaluation



WHISPER Analysis

- 4% accesses to PM, 96% to DRAM
- 5-50 epochs/transaction
- Self-dependencies common
- Cross-dependencies rare



HOPS Design

- Volatile memory hierarchy (almost) unchanged
- Order epochs without flushing
- Allows multiple copies of same cacheline
- Correct, conservative method based on coherence

Questions?

Thanks!