Kiln: Closing the Performance Gap Between Systems With and Without Persistence Support

Jishen Zhao ¶, Sheng Li*, Doe Hyun Yoon♦, Yuan Xie¶◊, and Norm Jouppi★

¶Pennsylvania State University, *HP Labs, ♦ IBM Research, ◊AMD Research China Lab, ★Google

Abstract: We propose a persistent memory design that adopts a nonvolatile cache and a nonvolatile main memory to enable atomic in-place updates without logging or copy-on-write.

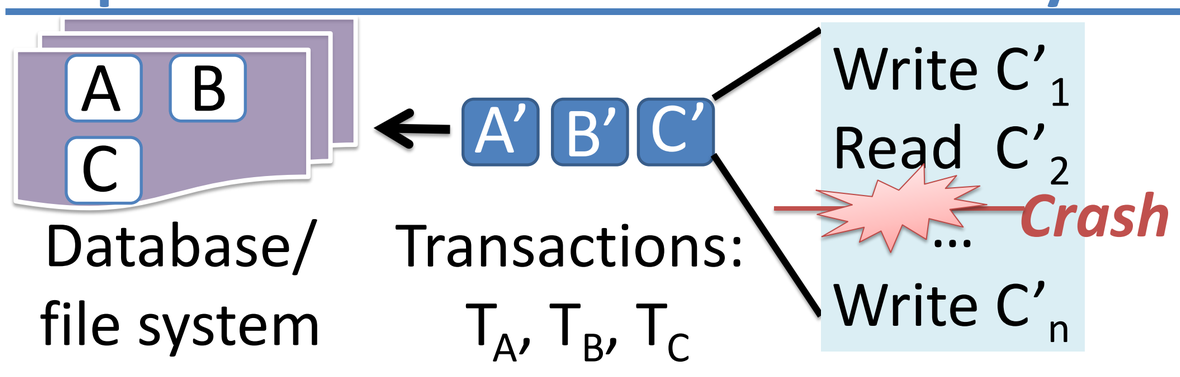
Persistent Memory

DRAM (No Persistence)
- Volatile ☹
- Byte-addressable ☺
- Fast ☺

Disk/Flash (Persistence)
- Block-addressable ☹
- Slow ☹
- Nonvolatile ☺

NVRAM (STT-MRAM, PCM, ReRAM, etc.)

Requirements of Persistent Memory



Atomicity
Consistency
Isolation
Durability

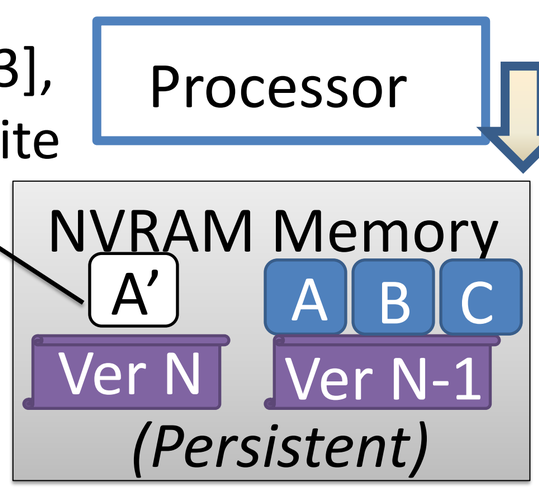
⇐

Multiversioning ✓
Ordering of writes ✓
Concurrency control
Nonvolatility ✓

Prior Work

Logging [2,3], copy-on-write [1,4]

(Atomic 8B writes)



Clflush/mfence or uncacheable writes [2, 4], epoch barriers [1,3]

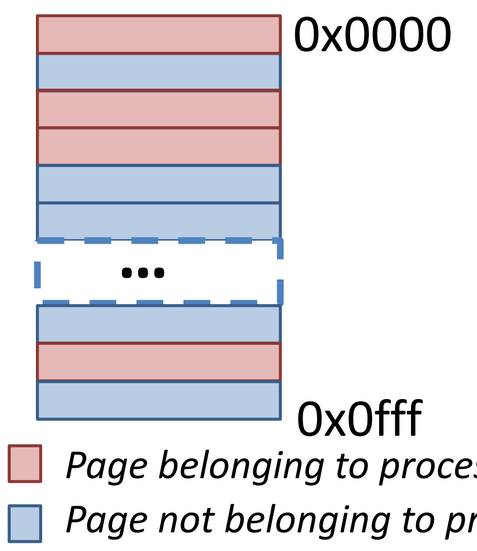
Software Interface

```
persistent {
 read Xa, Xb, Xc;
 some processing;
 write X0, X1;
}
```

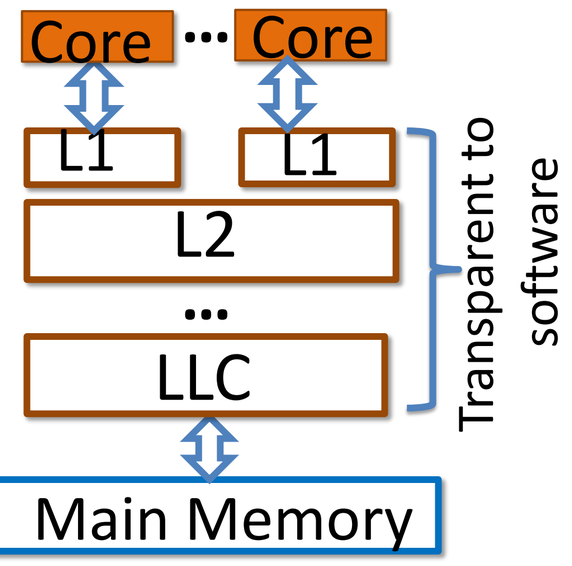
```
Persistent(inorder) {
 write Xa, Xb, Xc;
}
Persistent(inorder) {
 write Pa, Pb, Pc;
}
```

Different Views of Memory System

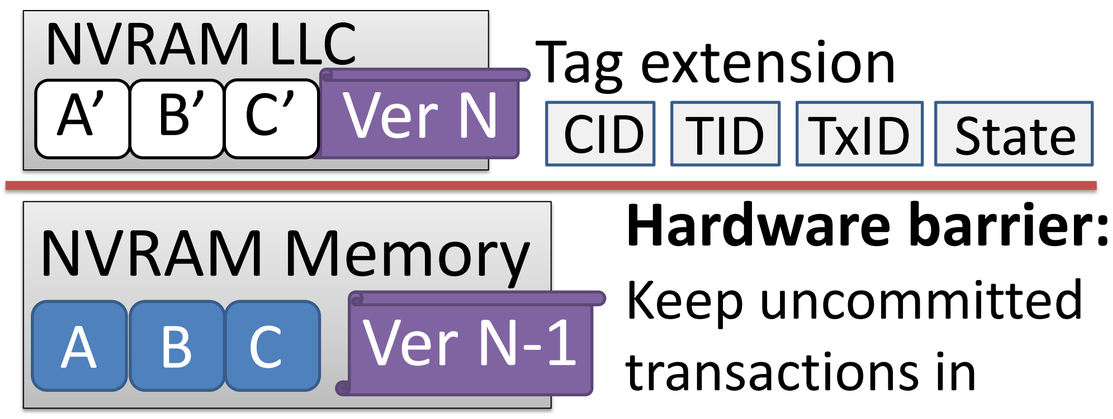
Software's view
A flat address space



Hardware's view
A hierarchy

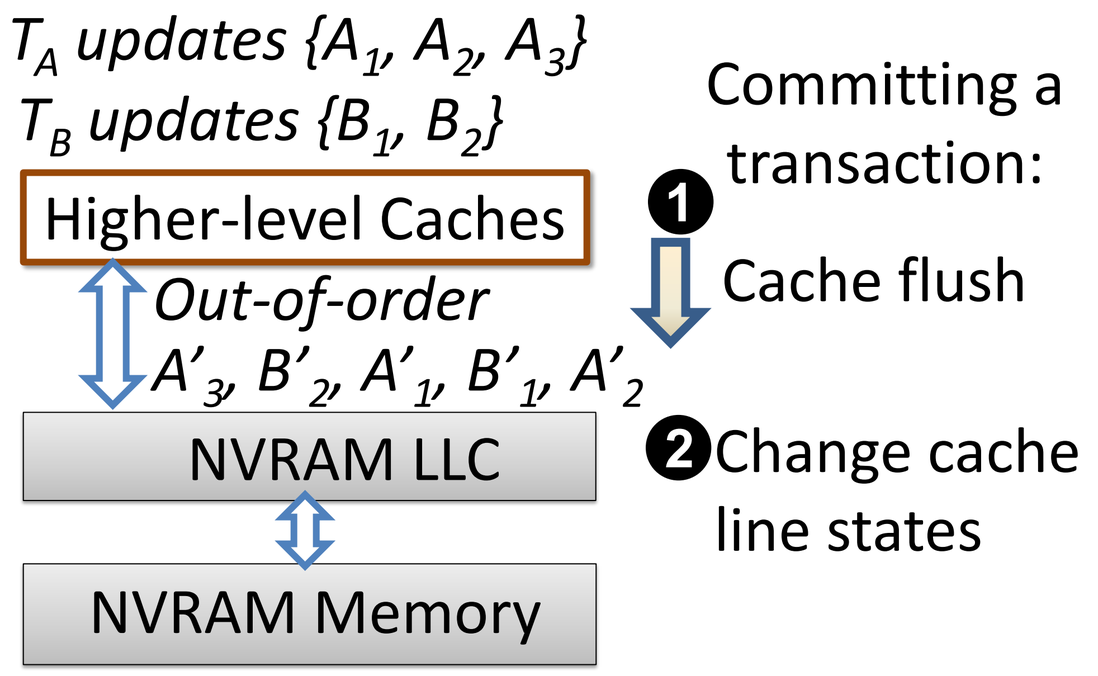


Leveraging Caching for In-place Updates



Hardware barrier:
Keep uncommitted transactions in NVRAM LLC

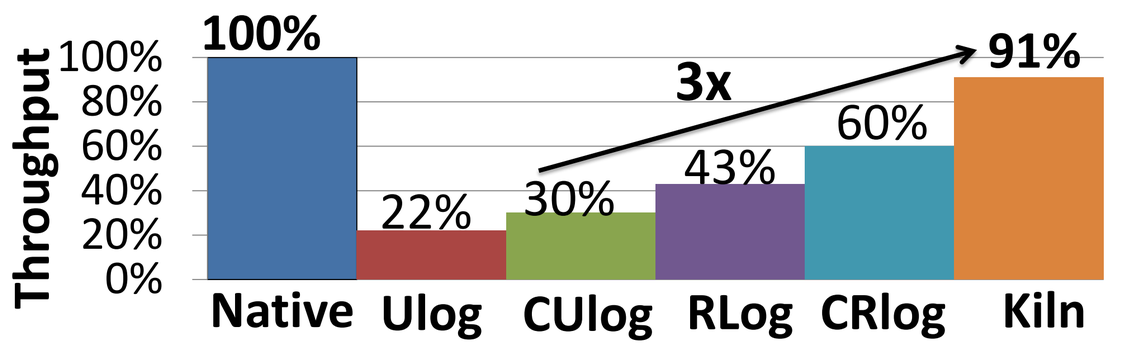
Out-of-order Writebacks, In-order Commits

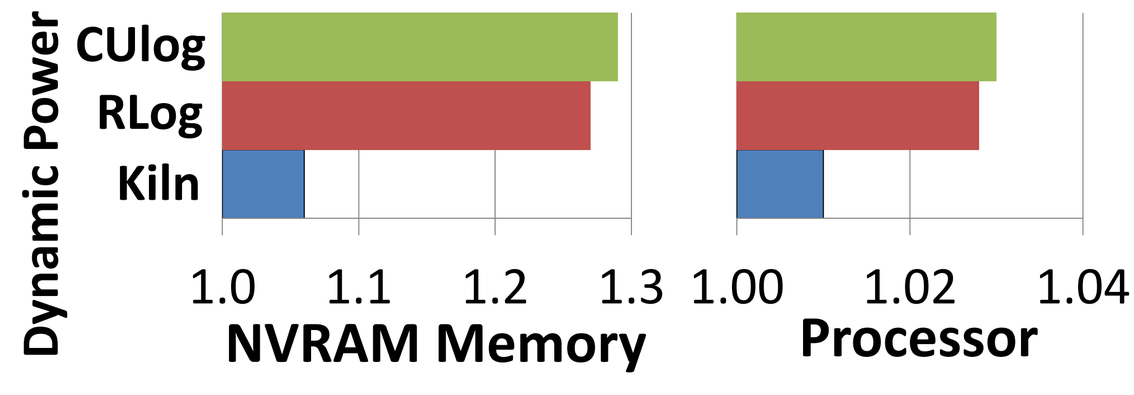


Committing a transaction:

1. Cache flush
2. Change cache line states

Key Results





References

[1] J. Condit, E. B. Nightingale, C. Frost, E. Ipek, B. Lee, D. Burger, and D. Coetzee. Better I/O through byte-addressable, persistent memory. In SOSP'09.

[2] H. Volos, A. J. Tack, and M. M. Swift. Mnemosyne: lightweight persistent memory. In ASPLOS'11.

[3] J. Coburn, A. M. Caulfield, A. Akel, L. M. Grupp, R. K. Gupta, R. Jhala, and S. Swanson NV-heaps: making persistent objects fast and safe with next-generation, non-volatile memories. In ASPLOS'11.

[4] S. Venkataraman, N. Tolia, P. Ranganathan, and R. H. Campbell. Consistent and durable data structures for non-volatile byte-addressable memory. In FAST'11.