**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)  Disk/Flash (Persistence)

- Volatile ☻
- Byte-addressable ☻
- Fast ☻

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

**Requirements of Persistent Memory**

- Atomicity ☑
- Consistency ☑
- Isolation ☑
- Durability ☑

**Prior Work**

- Logging [2,3], copy-on-write [1,4], (Atomic 8B writes)

**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**

**A flat address space**

Software’s view

**A hierarchy**

Hardware’s view

**Leveraging Caching for In-place Updates**

- NVRAM LLC
- Tag extension
- CIB, TID, TxID, State

**Out-of-order Writebacks, In-order Commits**

- `TA` updates `{A1, A2, A3}`
- `TB` updates `{B1, B2}`

**Key Results**

**References**


