SuperMem: Enabling Application-transparent Secure Persistent Memory with Low Overheads

Pengfei Zuo\(^1,2\), Yu Hua\(^1\), Yuan Xie\(^2\)

\(^1\) Huazhong University of Science and Technology, China
\(^2\) University of California at Santa Barbara, USA

---

**Background**

- **Persistent memory as main memory**
  - PCM, ReRAM, STT-RAM, 3D-Xpoint
  - High scalability, high density, and low standby power

- **Persistence issue**
  - Crash inconsistency

- **Security issue**
  - Physical-access attacks
    - username, password

- **Consistency guarantee**
  - Cache line flush (cflush)
  - Memory fence (mfence)
  - Logging
  - Copy-on-write

- **Memory encryption**
  - Counter mode encryption

- **The gap between persistence and security**
  - *cflush* and *mfence* cannot operate the counter cache
  - Data and counter cannot reach NVM in the same time

---

**The SuperMem Design**

- **SuperMem: an application-transparent secure persistent memory by leveraging a write-through counter cache.**

  - **A write-through counter cache**
    - No large battery backup
    - No software-level modifications
    - No need to correct counters

  - **Counter write coalescing**
    - Reduce the number of writes

  - **Cross-bank counter storage**
    - Speed up memory writes

---

**Evaluation**

- **Evaluation: 256KB transaction size**
  - Array
  - Queue
  - Btree
  - Hash Table
  - RB-tree

- **Evaluation: 4KB transaction size**
  - Same as above

---

**Comparisons**

- UnSec: An un-encrypted NVM
- WB: A ideal write-back scheme
- WT: A write-through scheme
- WT+CWC: A write-through scheme with CWC
- WT+XBank: A write-through scheme with XBank

---

**The SuperMem Design**

- **Write-through counter cache**
  - Ensure that data and its counter reach the write queue in the same time

- **Cross-bank counter storage (XBank)**
  - Ensure that data and its counter reach the write queue in the same time

---

**Existing solutions (write-back counter cache)**

- **Large Battery Backup**
  - [Awad et al., ASPLOS’16]
  - Expensive
  - Counter cache
  - Limited portability

- **Software-level Modification**
  - [Liu et al., HPCA’18]
  - New programming primitives
  - CounterAtomic
  - Portable

- **Error Correction**
  - [Ye et al., MICRO’18]
  - Low recovery time

---

**Counter Cache**

- Updated Counter

**CPU Cache**

- Encrypted Data

---

**Normative Execution Latency**

- Array
- Queue
- Btree
- Hash Table
- RB-tree

---

**Comparisons**

- UnSec: An un-encrypted NVM
- WB: A ideal write-back scheme
- WT: A write-through scheme
- WT+CWC: A write-through scheme with CWC
- WT+XBank: A write-through scheme with XBank

---

**SuperMem**

---

NVMW 2021