NVLeak: Off-Chip Side-Channel Attacks via Non-Volatile Memory Systems

Zixuan Wang,

Mohammadkazem Taram, Daniel Moghimi,

Steven Swanson, Dean Tullsen, Jishen Zhao

UC San Diego

USENIX Security 2023

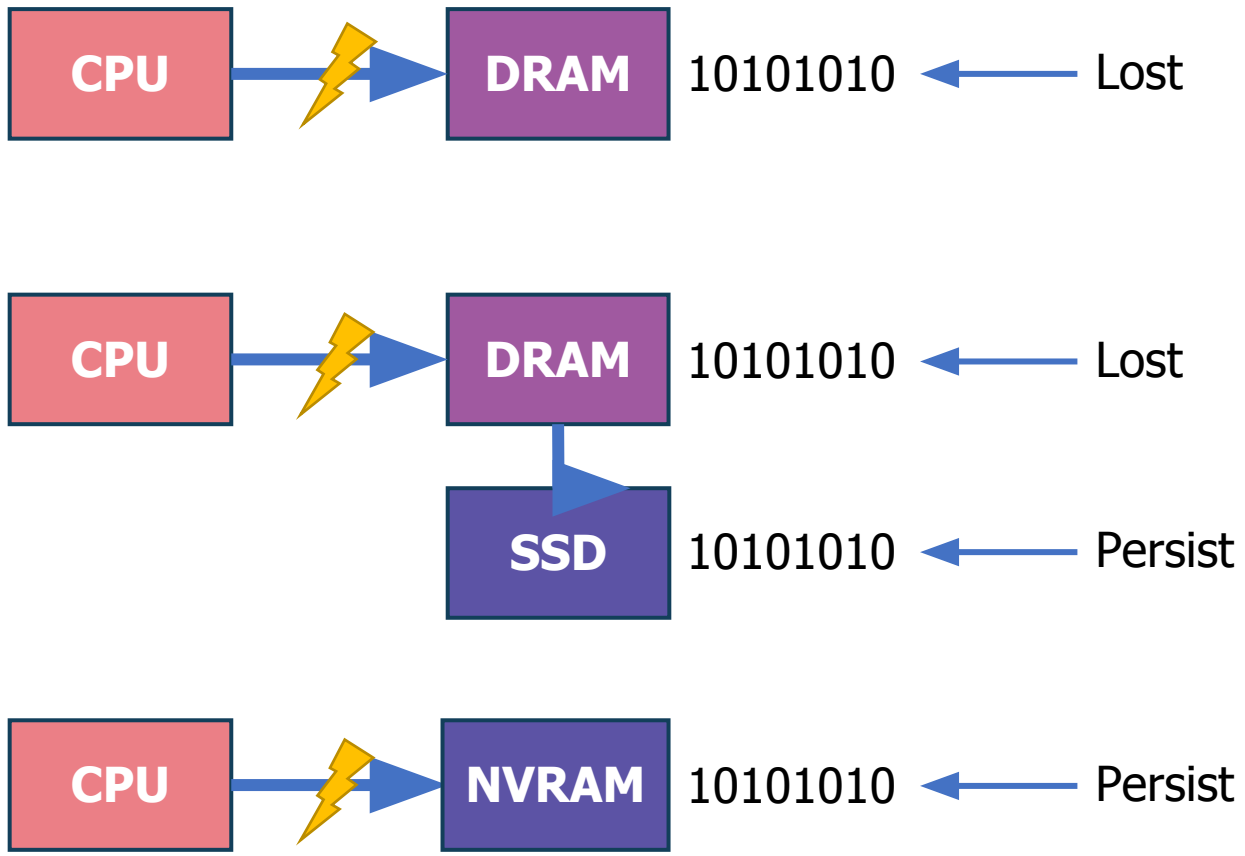
UC San Diego

Outline

- **Introduction**
- Cache
- Wear-Leveling

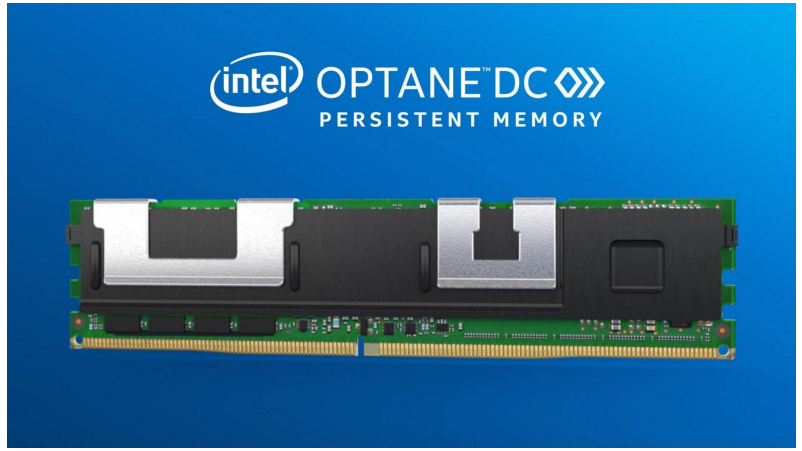
Introduction: Non-Volatile Memory

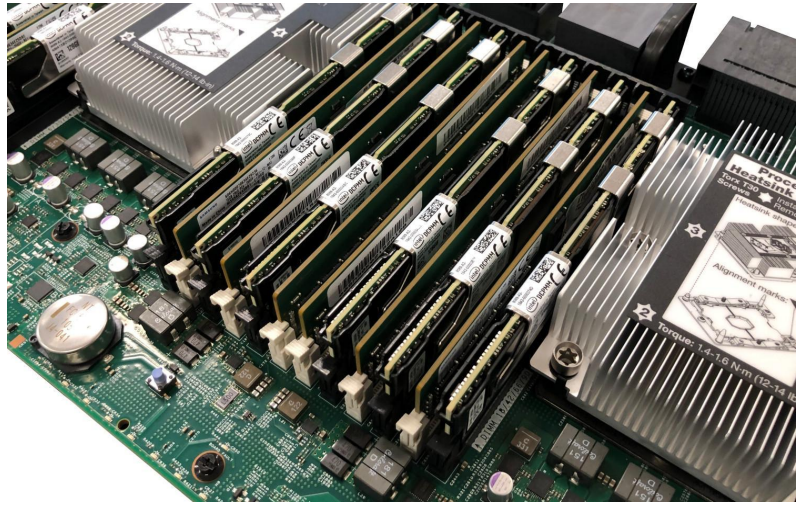
- Volatile Memory
  - DRAM
  - Byte-addressable
  - Data lost after power cycle
- Non-Volatile Storage
  - SSD/HDD
  - Block-addressable
  - Data persistence
- Non-Volatile Memory
  - Intel Optane PM
  - Byte-addressable
  - Data persistence



Introduction: Non-Volatile Memory

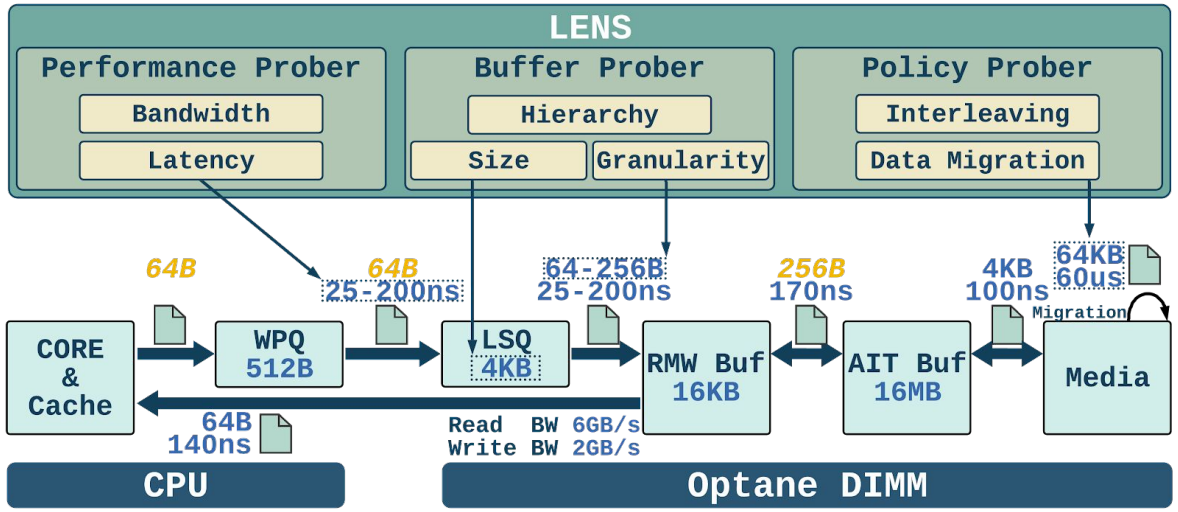
- Non-Volatile Memory
  - Byte-addressable
  - On memory bus
  - Persistent





Introduction: Prior Work

- Intel Optane PM μArch
  - Write pending queue
  - Load store queue
  - Buffers
  - 3D-Xpoint media
    - Data migration

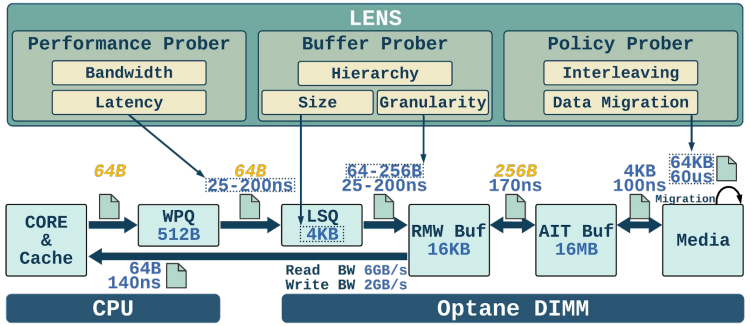


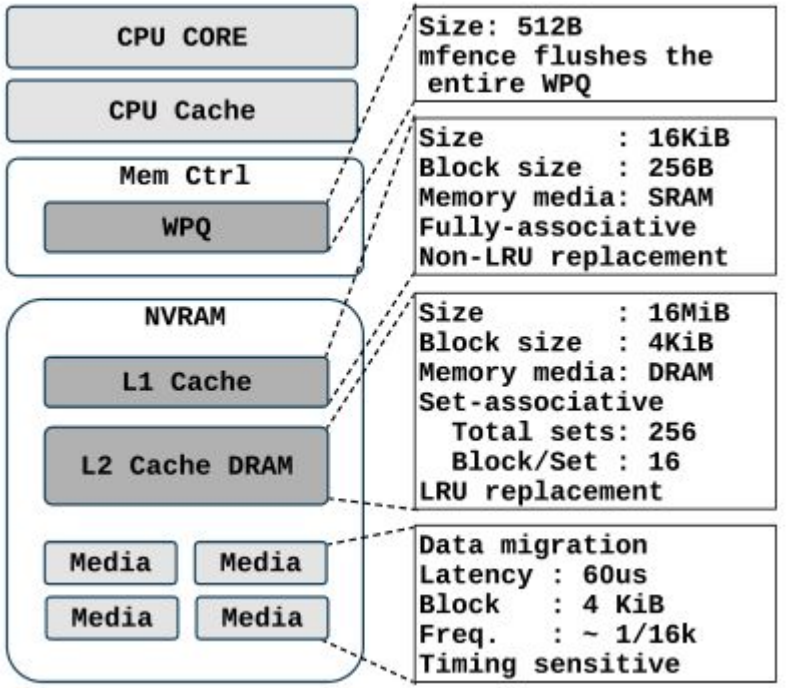
"Characterizing and Modeling Non-Volatile Memory Systems", Wang et al., MICRO 2020.

Introduction: This Work

- Intel Optane PM μArch
  - Write pending queue
  - Load store queue
  - ~~Buffers~~ Caches
    - Set associativity
    - Replacement policy
  - 3D-Xpoint media
    - Data migration
      - Timing-sensitive

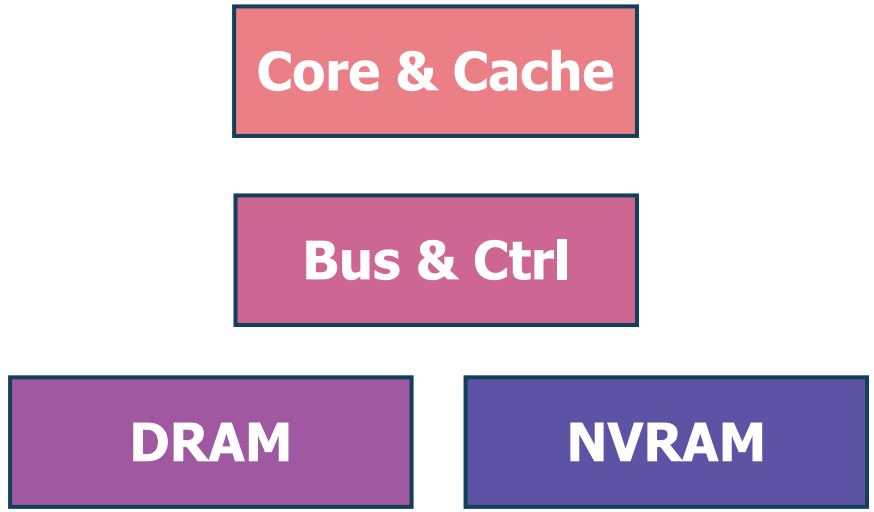
Security Issues (Side/Covert Channel)





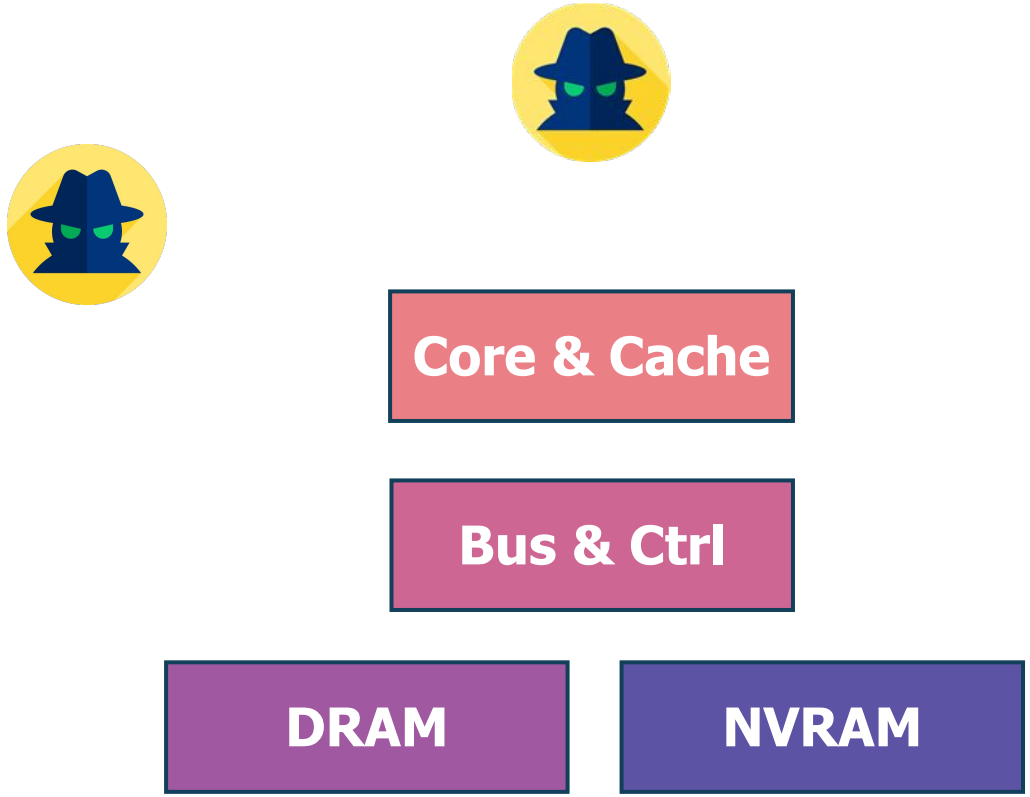
Introduction: Side/Covert Channel Attack

- Side Channel
  - Steals secret data from victim
    - Attacker
      - Measures computation environmental changes
        - Latency, voltage, temperature
      - To detect victim's side effects
      - ☐ Victim's activity
    - Bypass software-based security measurements
      - Hard to defend
        - Hardware redesign, performance loss



Introduction: Side/Covert Channel Attack

- Side Channel
  - Steals secret data from victim
- Covert Channel
  - Transfer secret between attackers
    - A special version of side channel
      - Attack controls both sides of the channel
      - No victim



Introduction: This Work

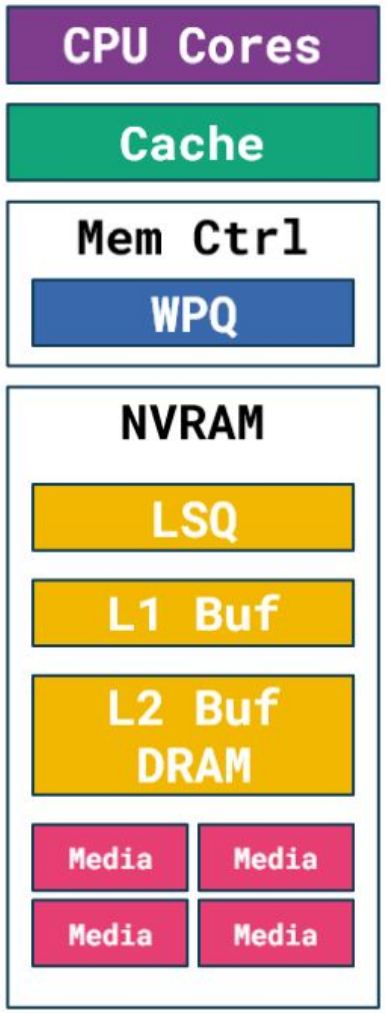
- Detailed μArch Reverse Engineering
  - On-NVRAM caches and data migration
- Attacks
  - Cache-based side/covert channels
    - Break virtual machine isolation
    - Break filesystem isolation
    - Leak database operations
  - Data-migration-based covert channel
    - Break filesystem isolation

Outline

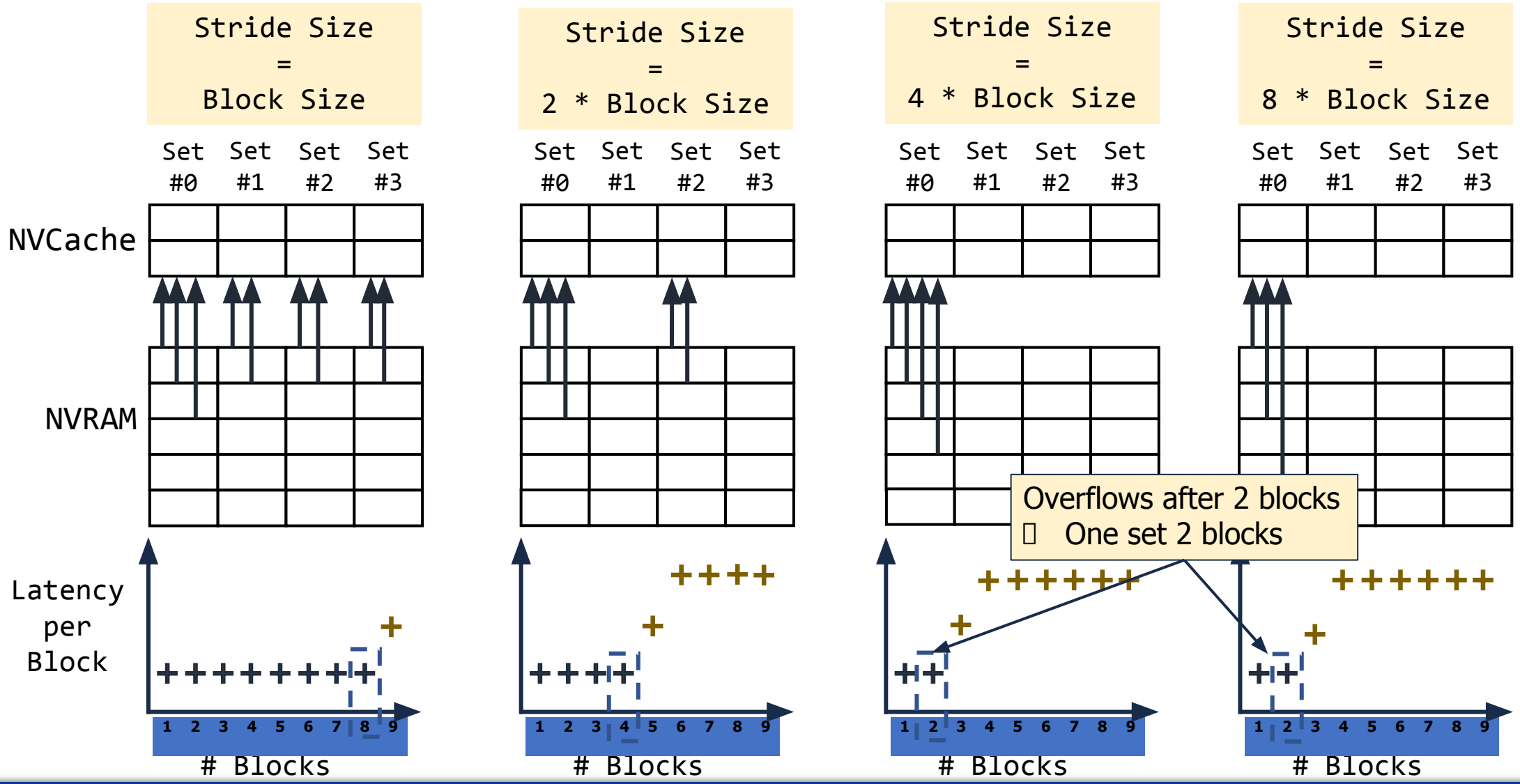
- Introduction
- **Cache**
- Wear-Leveling

Reverse Engineering: NVRAM Caches

- The Problem
  - We know multiple buffers on NVRAM
    - Prior works [MICRO'20]
  - But are they buffers or caches?
- Our Solution
  - Software-based reverse engineering
    - Strided memory access
      - Change stride size
      - Monitor latency changes
      - Detect cache set size

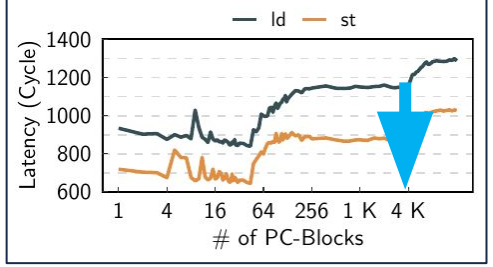


Reverse Engineering: NVRAM Caches

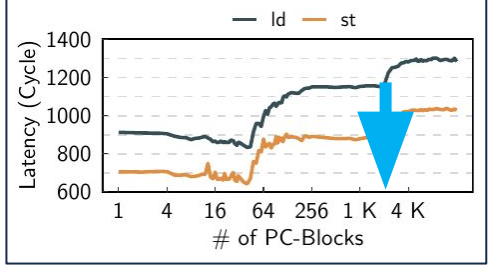


Reverse Engineering: NVRAM Caches

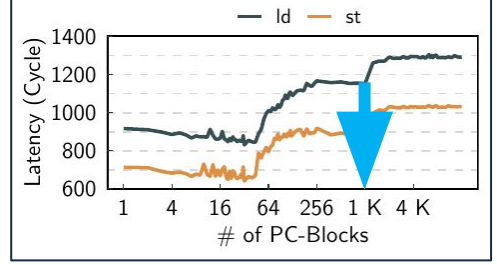
Stride = 1 Block
Overflow = 4K Block



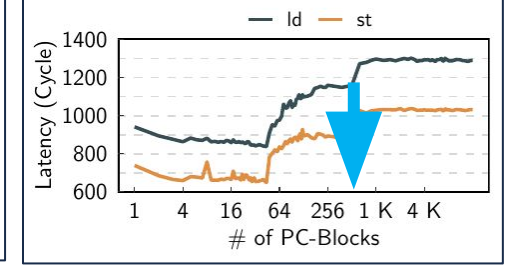
Stride = 2 Block
Overflow = 2K Block



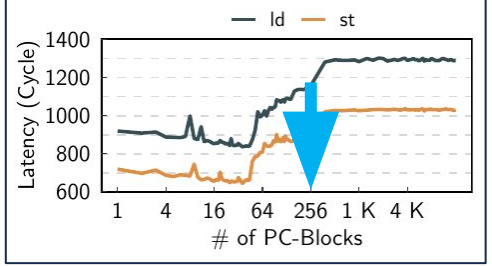
Stride = 4 Block
Overflow = 1K Block



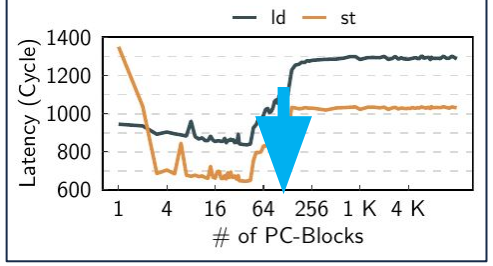
Stride = 8 Block
Overflow = 512 Block



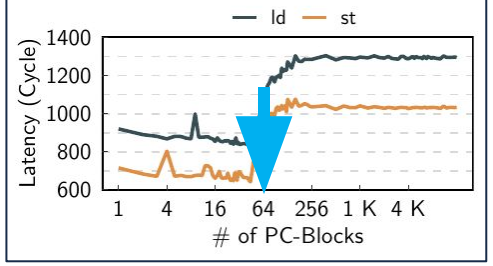
Stride = 16 Block
Overflow = 256 Block



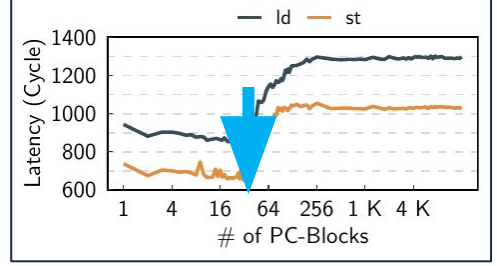
Stride = 32 Block
Overflow = 128 Block



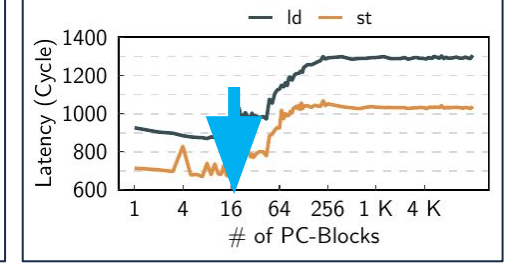
Stride = 64 Block
Overflow = 64 Block



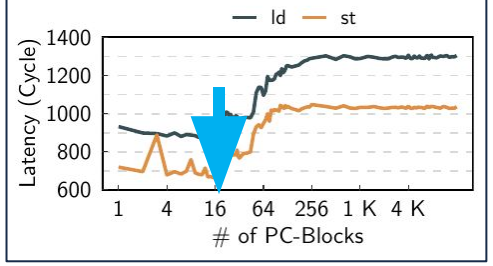
Stride = 128 Block
Overflow = 32 Block



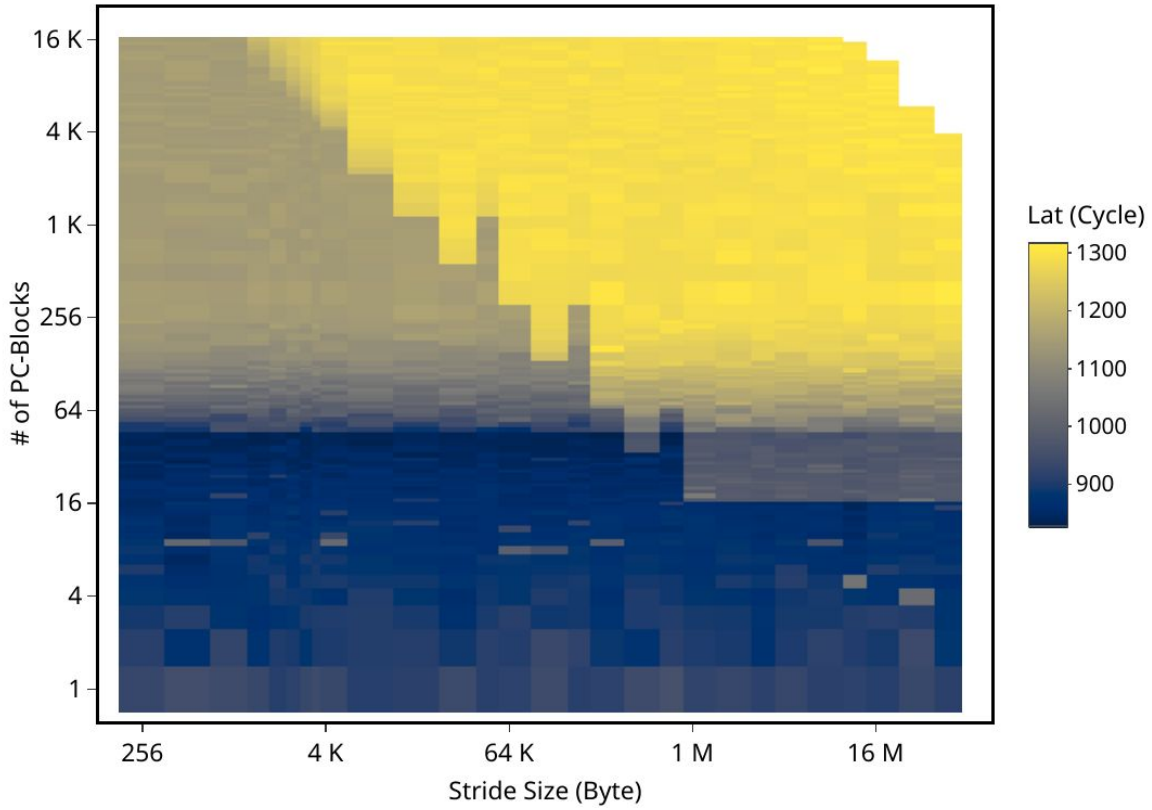
Stride = 256 Block
Overflow = 16 Block

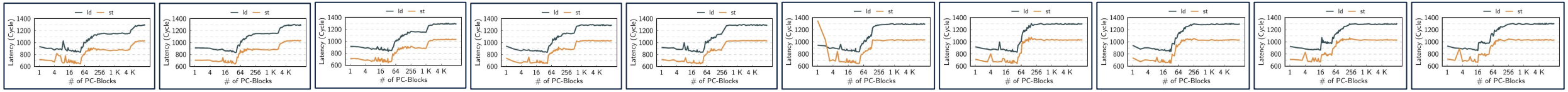


Stride = 512 Block
Overflow = 16 Block



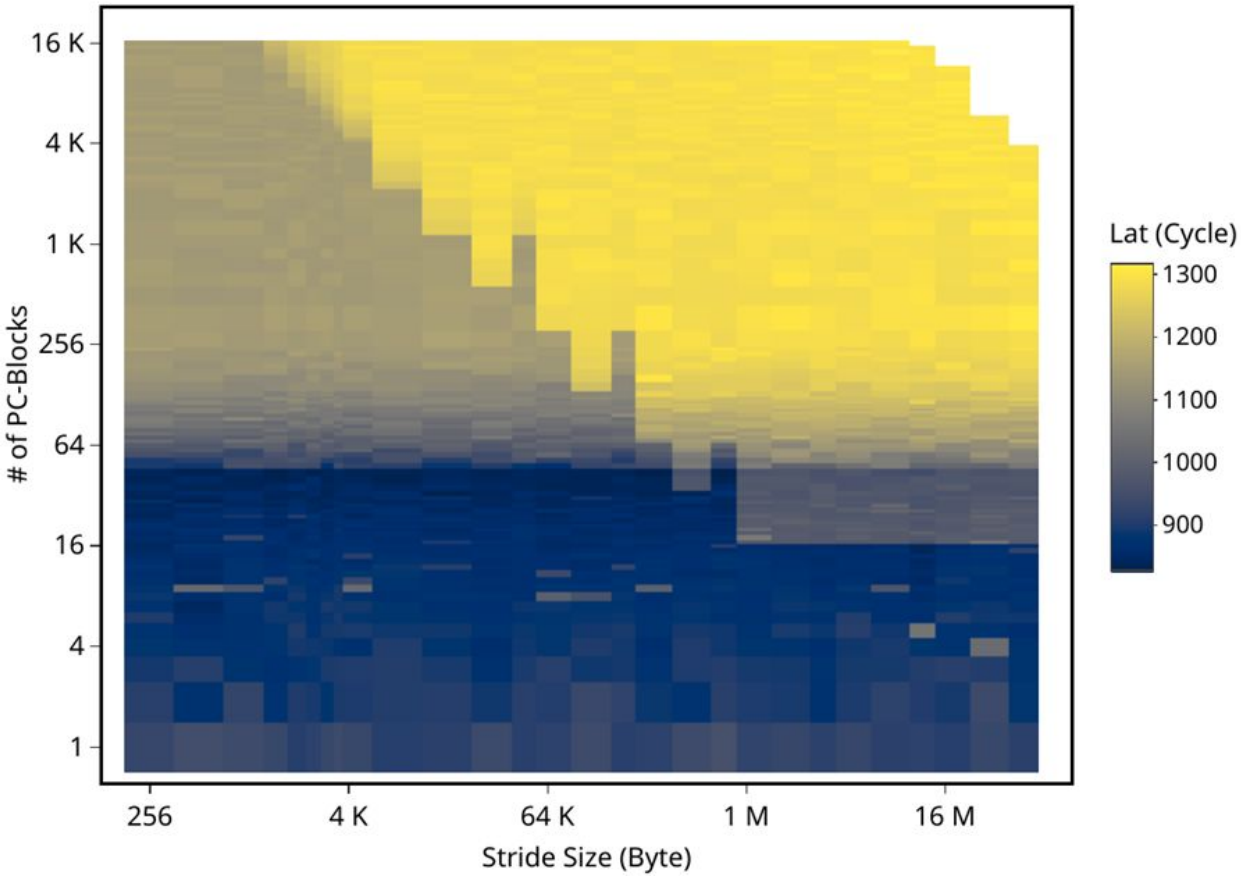
Reverse Engineering: NVRAM Caches





Reverse Engineering: NVRAM Caches

- Results
  - Larger stride size
    - ☐ Smaller overflow point
  - Stride size == 1 MiB
    - Overflow point stable
    - All blocks in the same cache set
    - Cache size / Stride Size
      - ☐ 16 MiB / 1 MiB
      - ☐ 16 ways

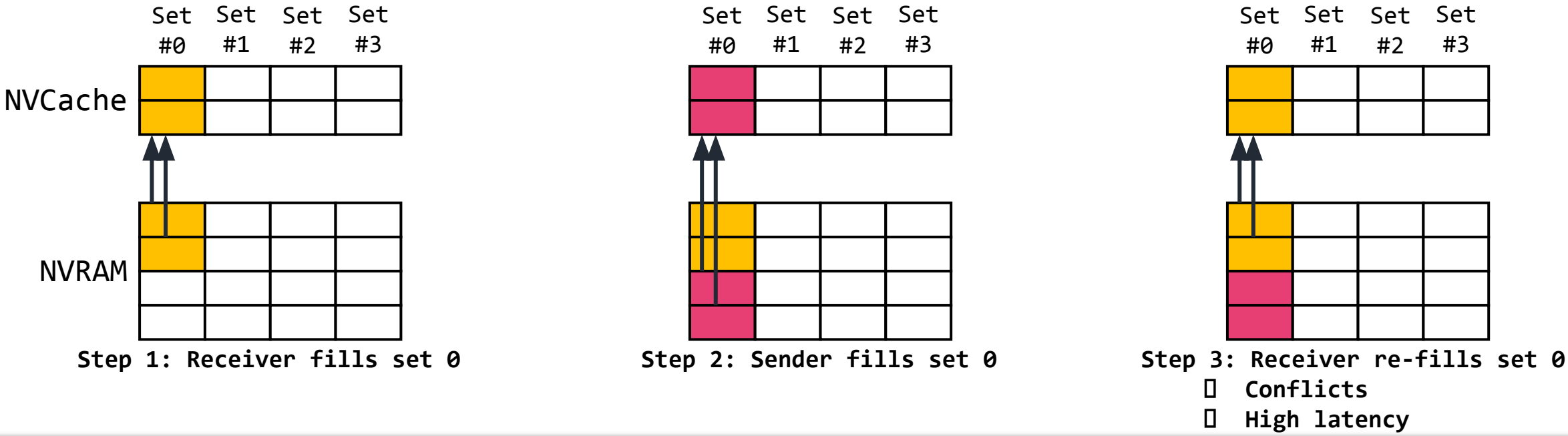


Covert Channel: NVRAM Caches

- Contention in Cache Sets
  - Sender and receiver access the same set
    - ☐ High latency ☐ Sending bit 0
  - Sender and receiver access different sets
    - ☐ Low latency ☐ Sending bit 1

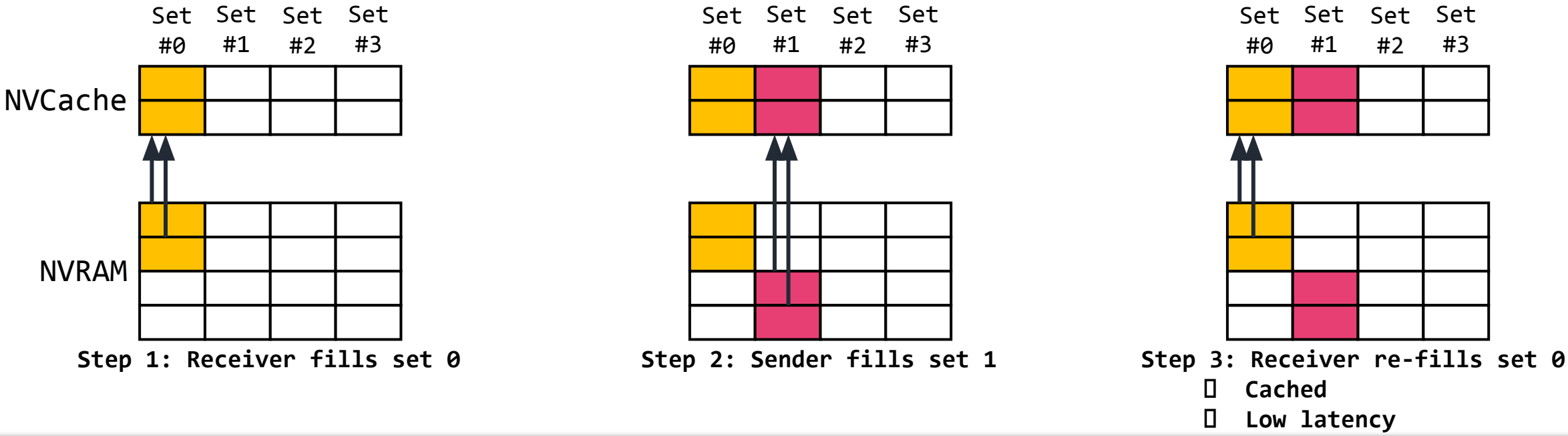
Covert Channel: NVRAM Caches

- Contention in Cache Sets
  - Sender and receiver access the same set
    - ☐ **High latency** ☐ **Sending bit 0**
  - Sender and receiver access different sets
    - ☐ Low latency ☐ Sending bit 1



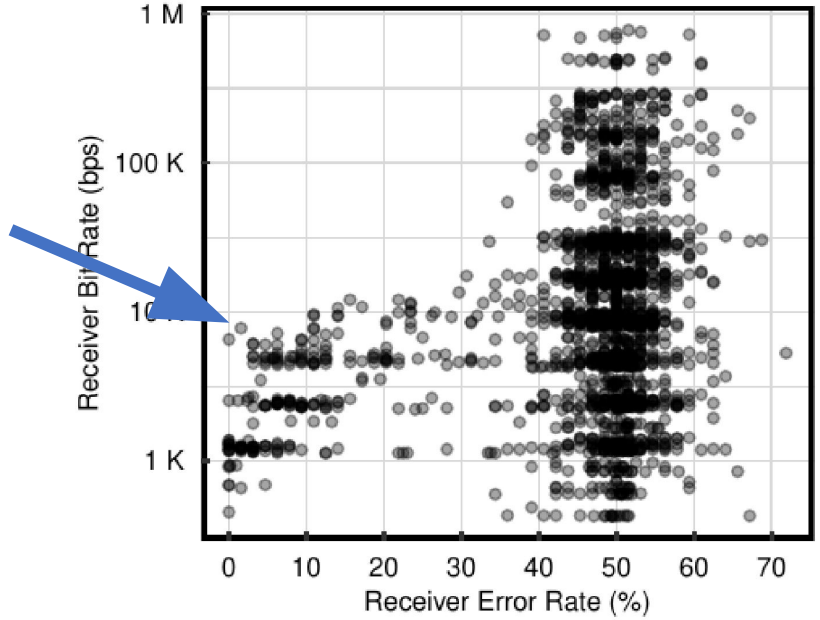
Covert Channel: NVRAM Caches

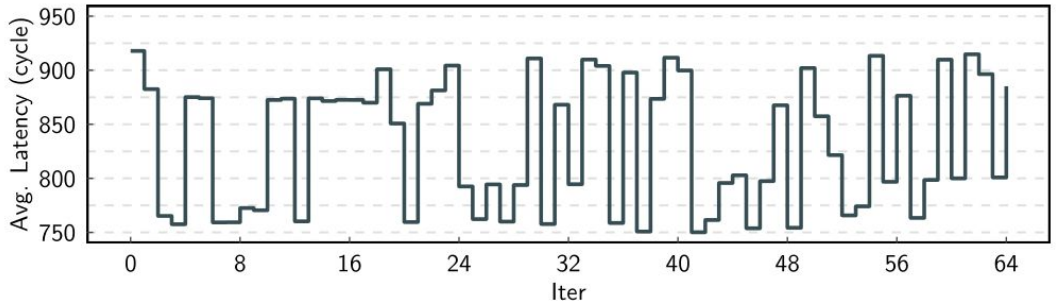
- Contention in Cache Sets
  - Sender and receiver access the same set
    - ☐ High latency ☐ Sending bit 0
  - Sender and receiver access different sets
    - ☐ **Low latency** ☐ **Sending bit 1**



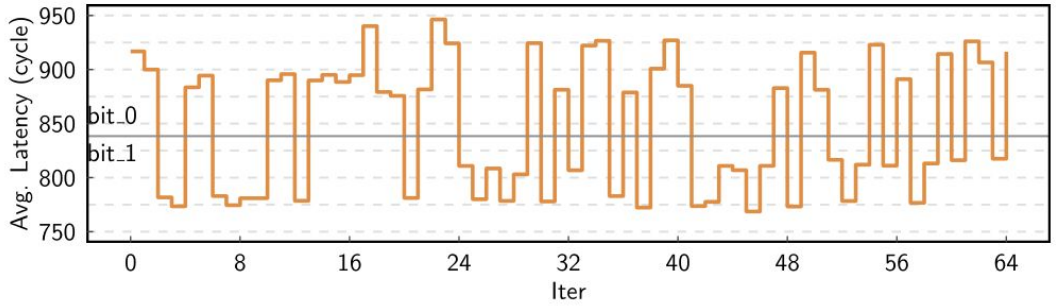
Covert Channel: NVRAM Caches

- Contention in Cache Sets
  - Sender and receiver access the same set
    - ☐ High latency ☐ Sending bit 0
  - Sender and receiver access different sets
    - ☐ Low latency ☐ Sending bit 1





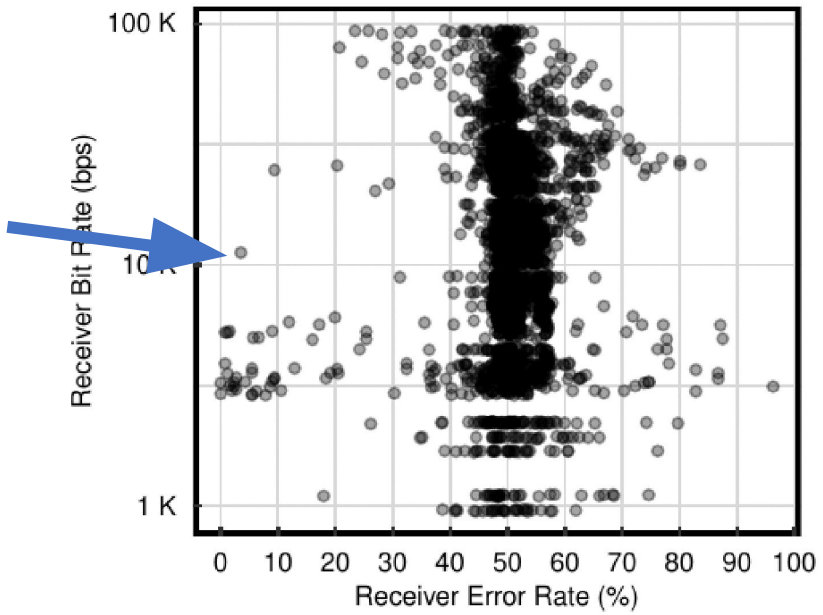
(a) Sender

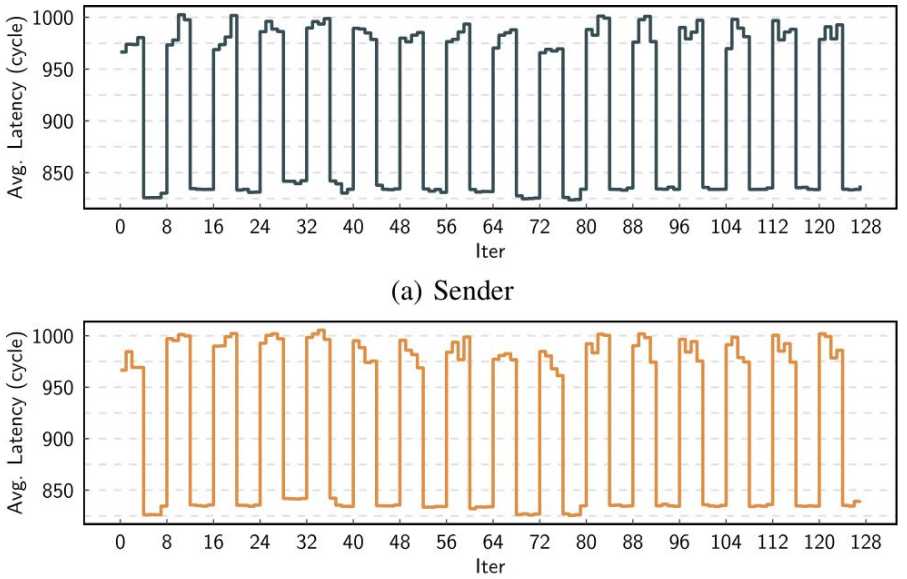


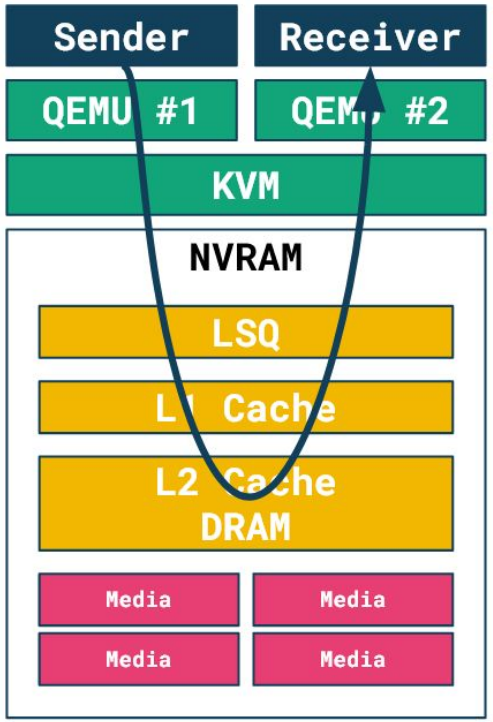
(b) Receiver

Covert Channel: Cross Virtual Machine

- Between Two QEMU VMs
  - Sender and receiver in different VMs
  - Share the same NVRAM DIMM but different regions



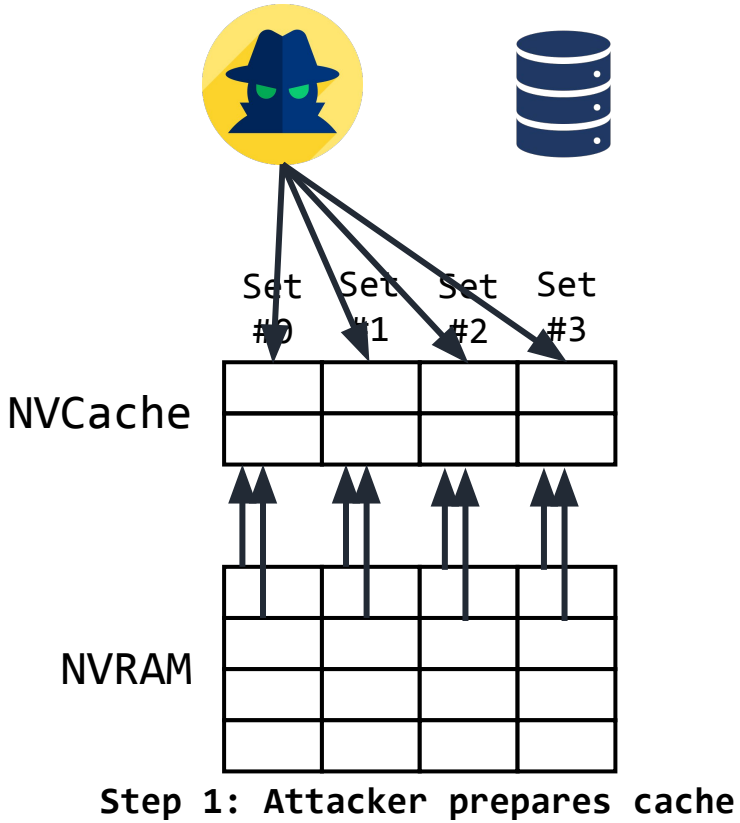




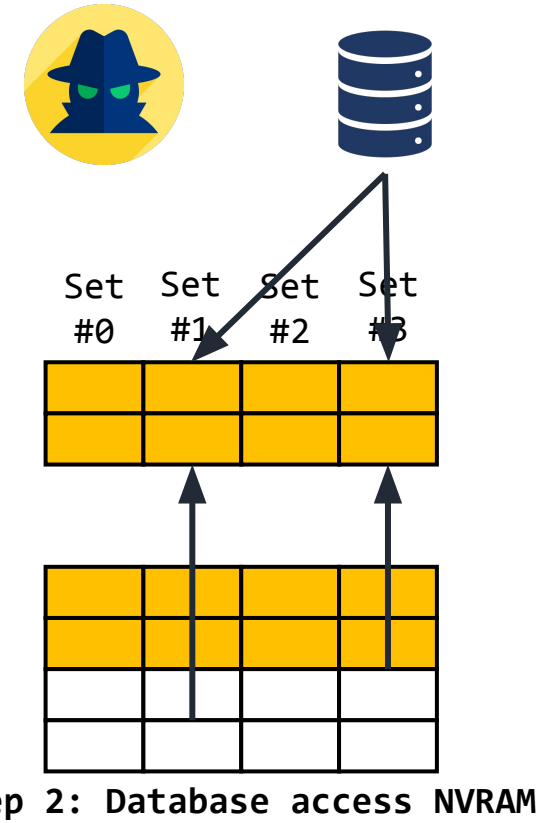
Side Channel: Database Operations

- Threat Model
  - Victim is a database application
    - Executing SQL queries
    - DAX filesystem
  - Attacker is in another process
    - Shares NVRAM with the database
    - Device DAX
  - Attacker monitors NVRAM cache set latencies

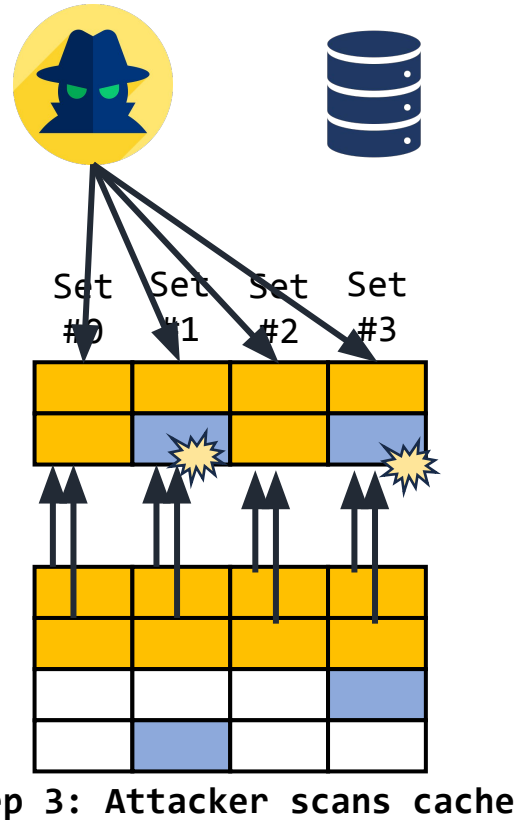
Side Channel: Database Operations



Step 1: Attacker prepares cache



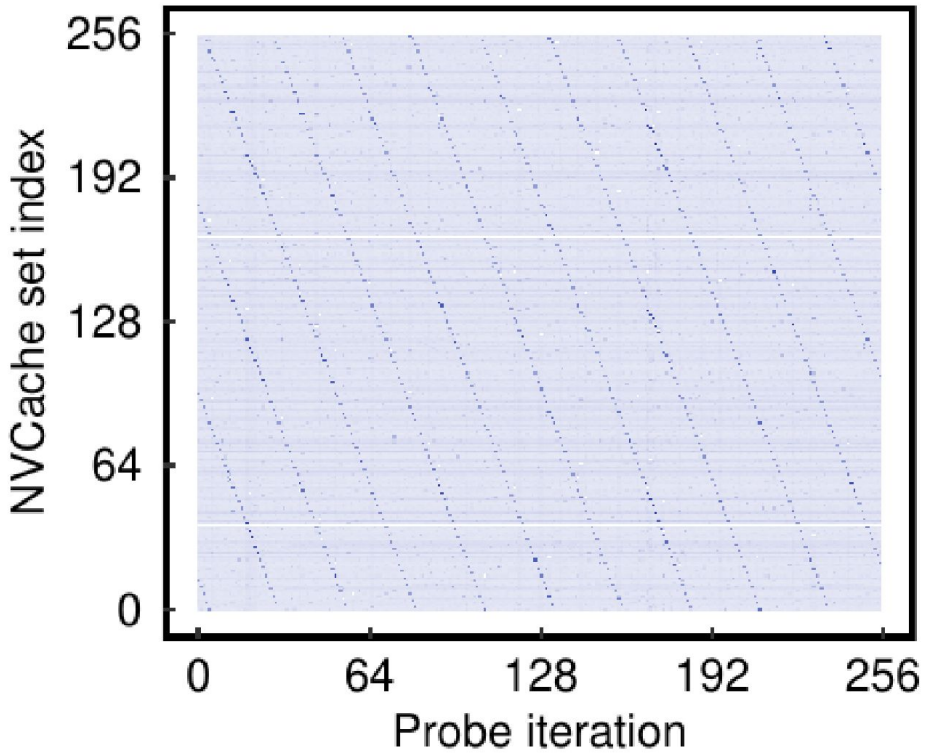
Step 2: Database access NVRAM



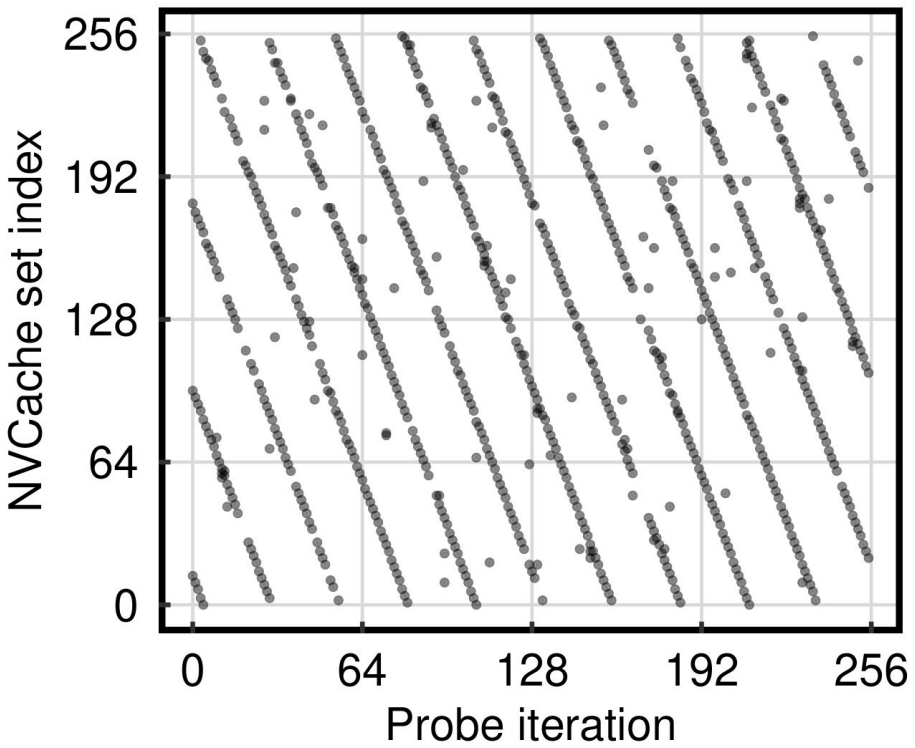
Step 3: Attacker scans cache

Side Channel: Database Operations

- Results
  - Idle



(a) Idle



(c) Idle (filtered)

Side Channel: Database Operations

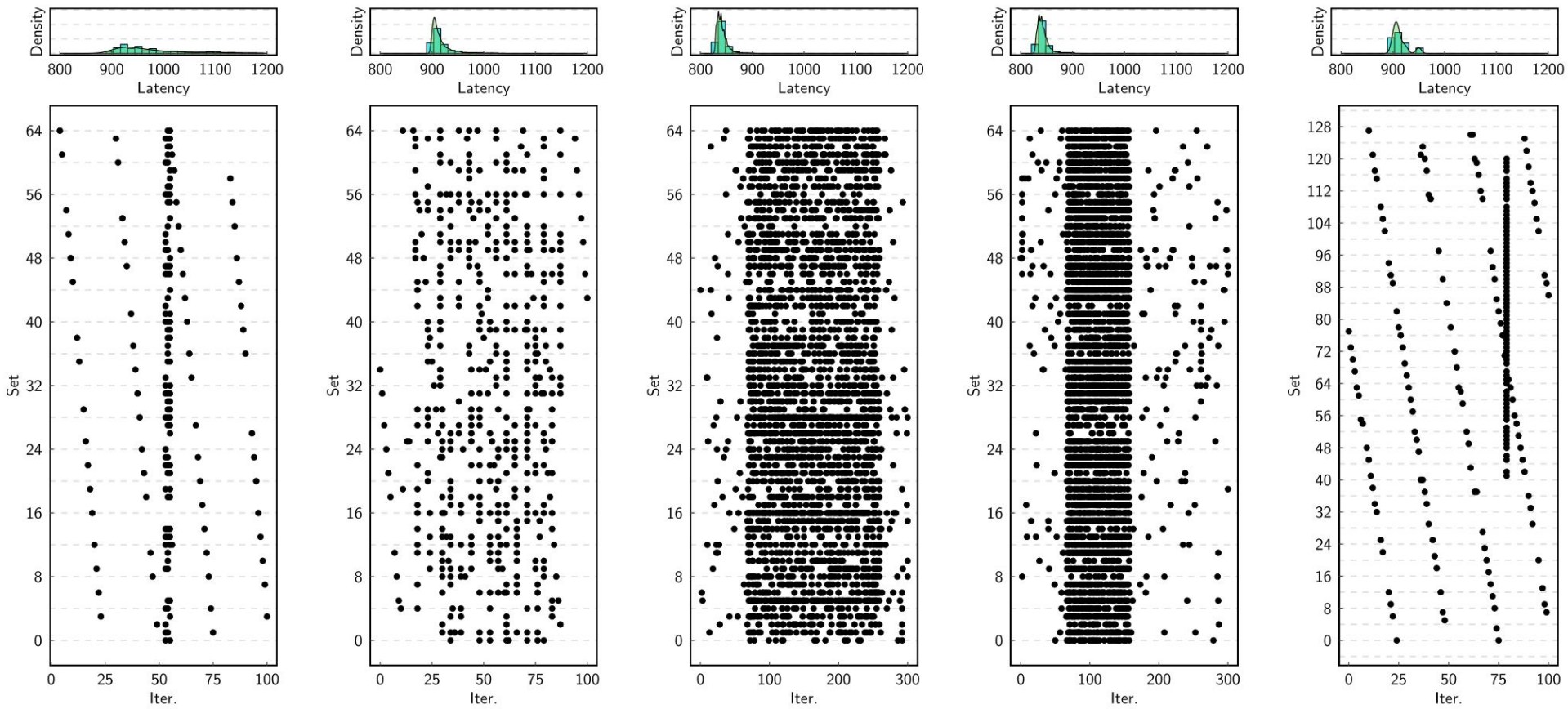
- Results
  - Idle
  - SQL

```
-- U1: Update 100 records in 'info' table
UPDATE info SET name = "New Name" WHERE npi == 1144223363;
-- U2: Update 1 record in 'address' table
UPDATE address SET city = "New City" WHERE npi == 1144223363;
-- C1: Count records in 'address' table
SELECT COUNT(*) FROM address WHERE lower(city)='athens';
-- Q1: Query both tables
SELECT * FROM info, address WHERE
  info.npi = address.npi AND info.npi == 1144223363;
-- I1: Insert 10,000 records in 'info' table
INSERT INTO info VALUES (...), (...);
```

(a) Database operations.

Side Channel: Database Operations

- Results
  - Idle
  - SQL



(a) U1 (b) U2 (c) C1 (d) Q1 (e) I1

Side Channel: Database Operations

- Results
  - Idle
  - SQL

	U1	U2	C1	Q1	I1
U1	10.92	-0.43	0.60	-0.48	-15.45
U2	-0.34	1.81	-0.89	0.99	-14.11
C1	-3.46	-0.39	1.55	0.88	-3.42
Q1	-6.10	0.07	-0.70	1.62	-3.95
I1	-3.99	-0.59	-0.30	-0.72	60.15

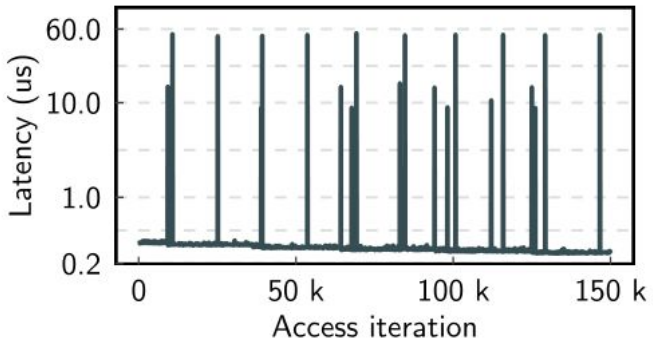
(b) Correlation matrix.

Outline

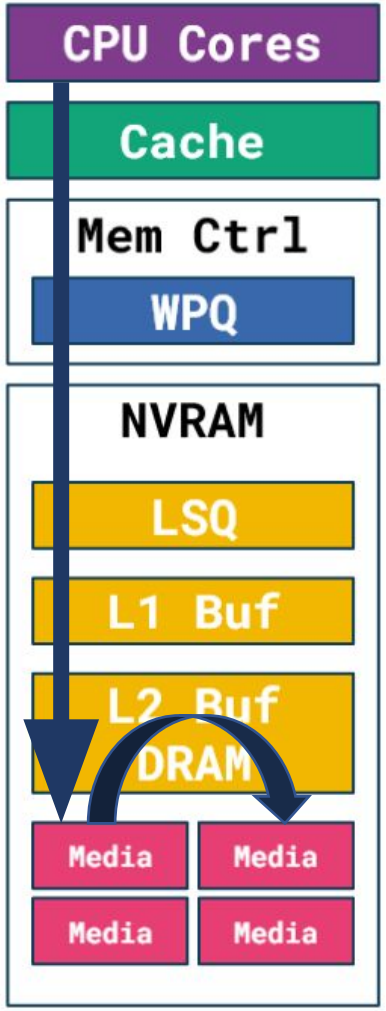
- Introduction
- Cache
- **Wear-Leveling**

Reverse Engineering: Data Migration

- Overwrite Test
  - To detect data migration
  - Write data to the same memory area
    - Inject delays between writes
      - ☐ Check if long latency is related to timings
    - Write to uncacheable memory (to bypass CPU cache)

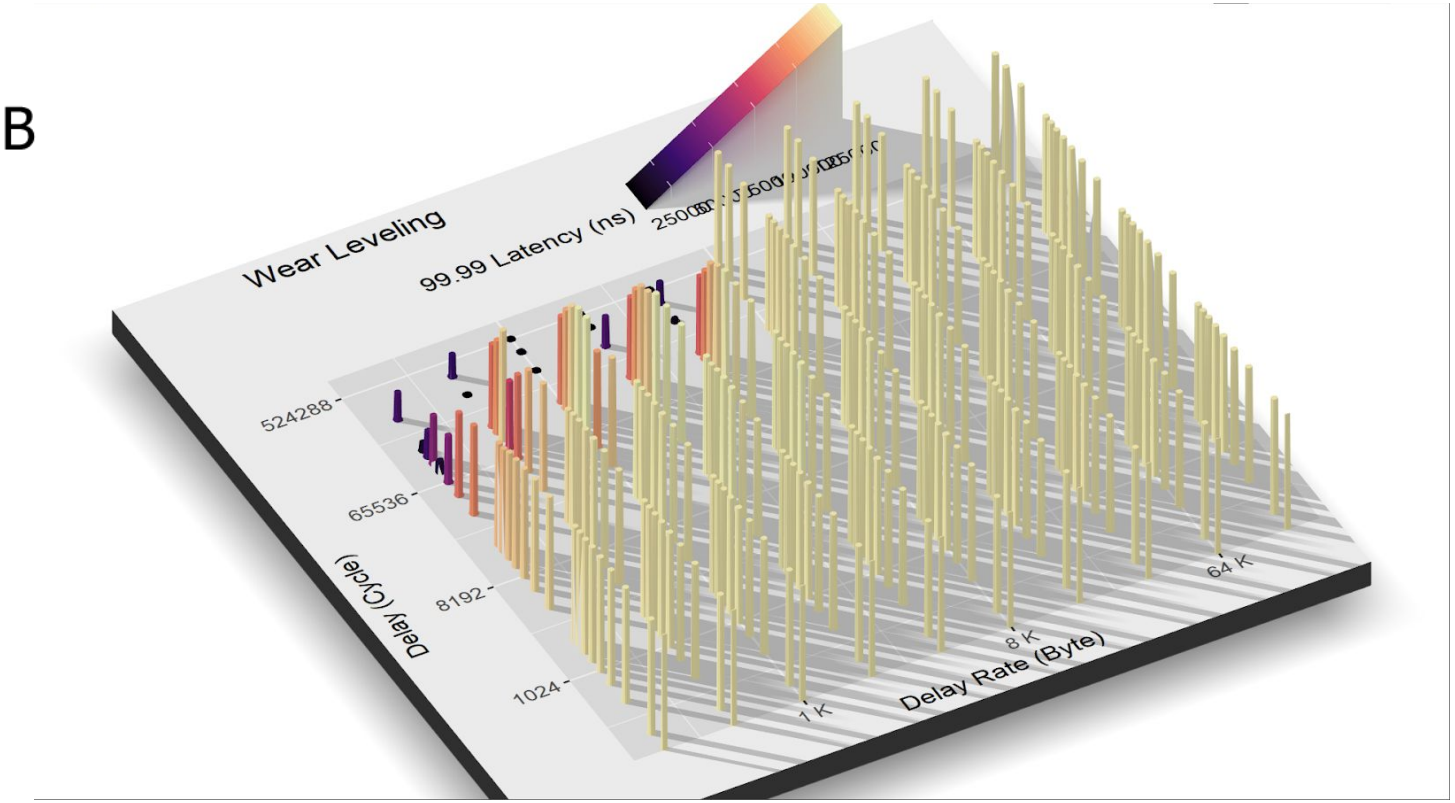


(a) Overwrite per-iter latency.



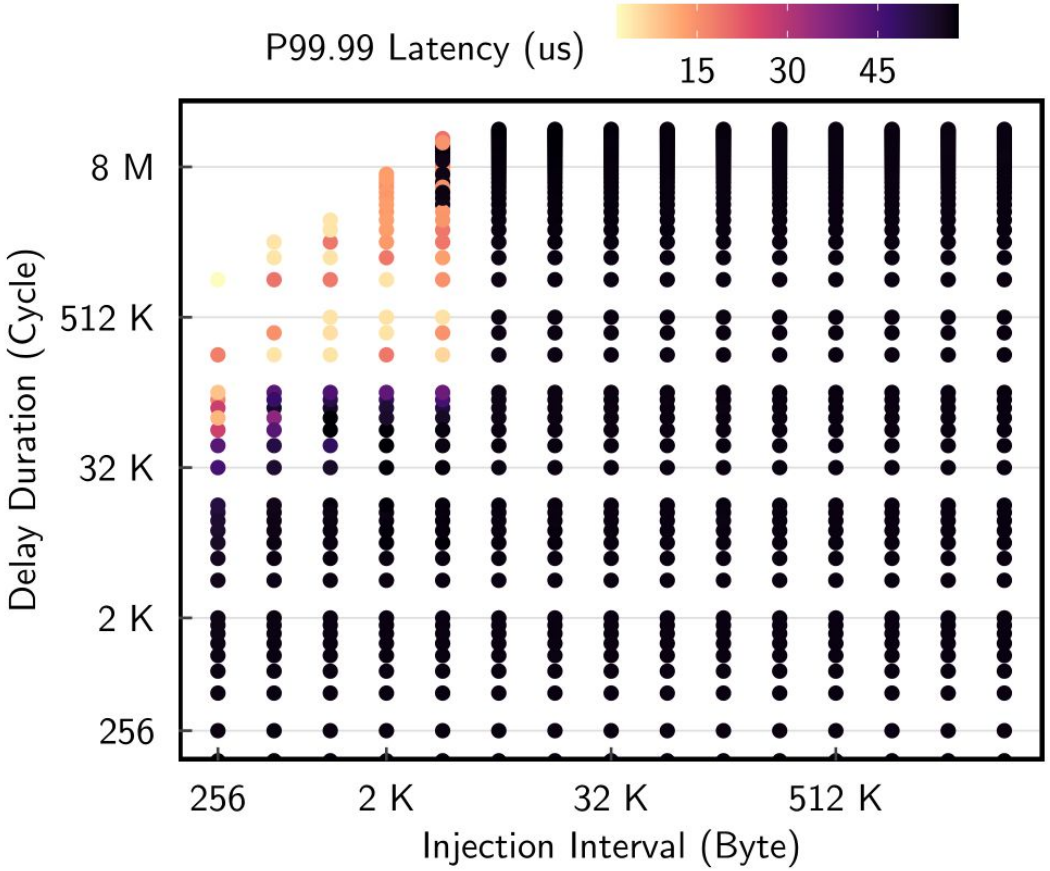
Reverse Engineering: Data Migration

- Results
  - Each bar
    - An overwrite test with 512 MiB write volume
  - X-Axis "Delay per byte"
    - Inject a delay after x bytes written
  - Y-Axis "Delay cycle"
    - Inject y cycles each delay
  - Z-Axis "Longest latency"
    - 99.99 percentile latency



Reverse Engineering: Data Migration

- Results
  - Data migration is timing sensitive
    - More delay cycles (large y value)
      - Less long latencies
    - More frequent delays (small x value)
      - Less long latencies

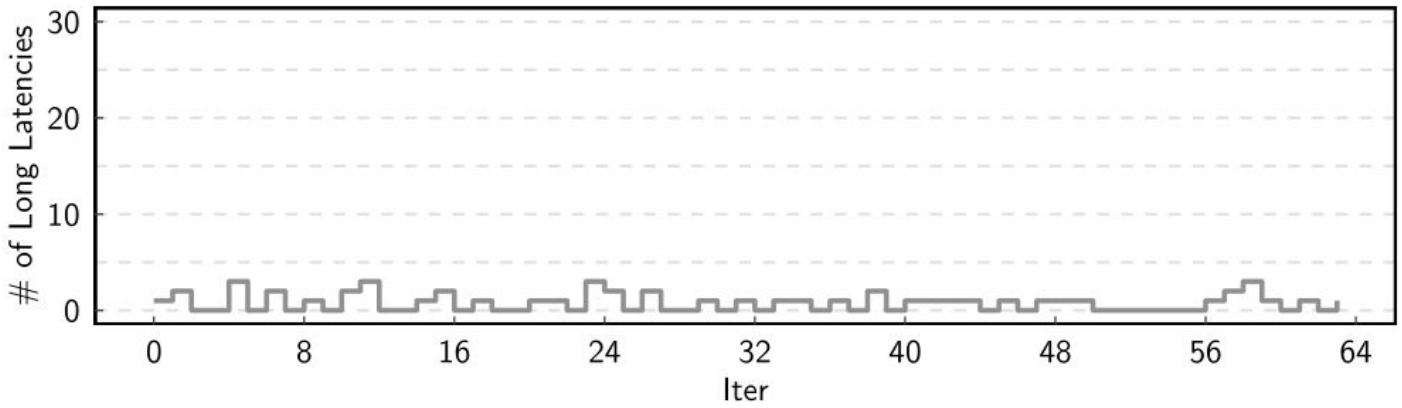


Covert Channel: Filesystem Inode

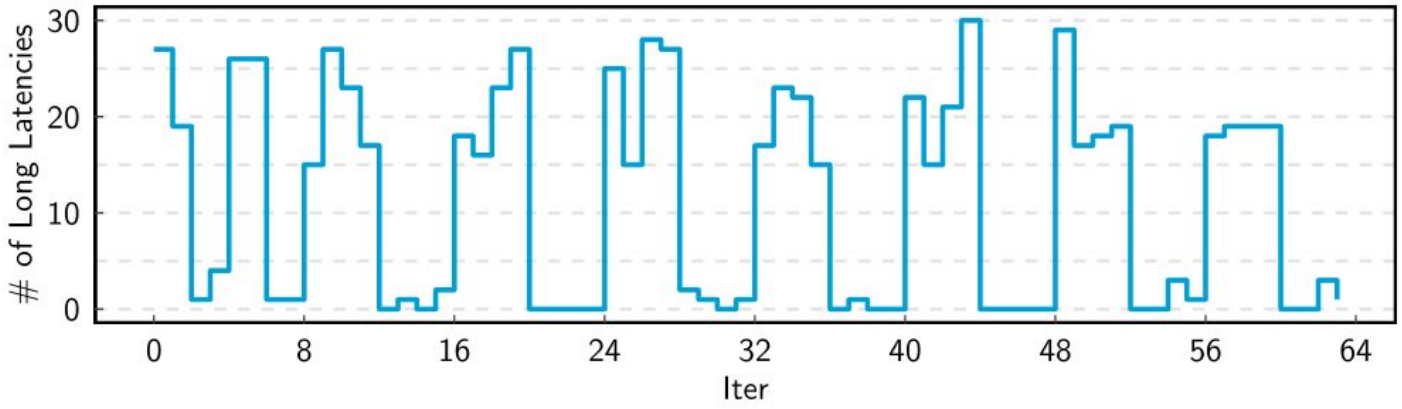
- Between Two File Inodes
  - Sender and receiver access two different files
    - Different permissions and ownerships
    - Their inodes belong to the same memory page
  - The channel
    - Update inode ☐ write NVRAM
    - Measure latency ☐ frequency of long latency

Covert Channel: Filesystem Inode

- Results
  - DRAM
    - Cannot establish channel
  - NVRAM
    - Established channel
    - A bit sequence
      - `0xccf0_f0f0_f0f0_f0f0`



(a) DRAM



(b) NVRAM

Conclusion

- Detailed μArch Reverse Engineering
  - On-NVRAM caches and data migration
- Attacks
  - Cache-based side/covert channels
    - Break virtual machine isolation
    - Break filesystem isolation
    - Leak database operations
  - Data-migration-based covert channel
    - Break filesystem isolation

NVLeak: Leaking Secrets via Non-Volatile Memory Systems

Zixuan Wang,

Mohammadkazem Taram, Steven Swanson, Dean Tullsen, Jishen Zhao

UC San Diego