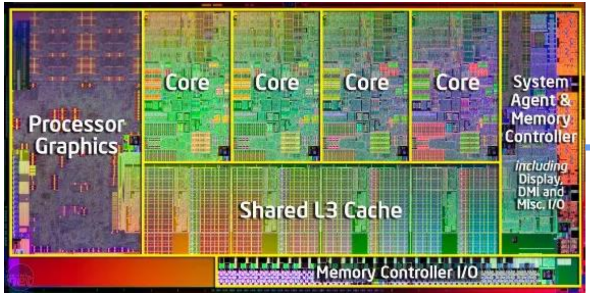
Denial-of-Service Attacks on Shared Cache in Multicore: Analysis and Prevention

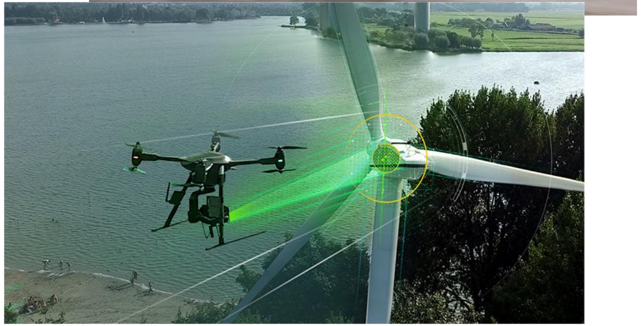
Michael Bechtel and Heechul Yun

University of Kansas

Multicore Platforms

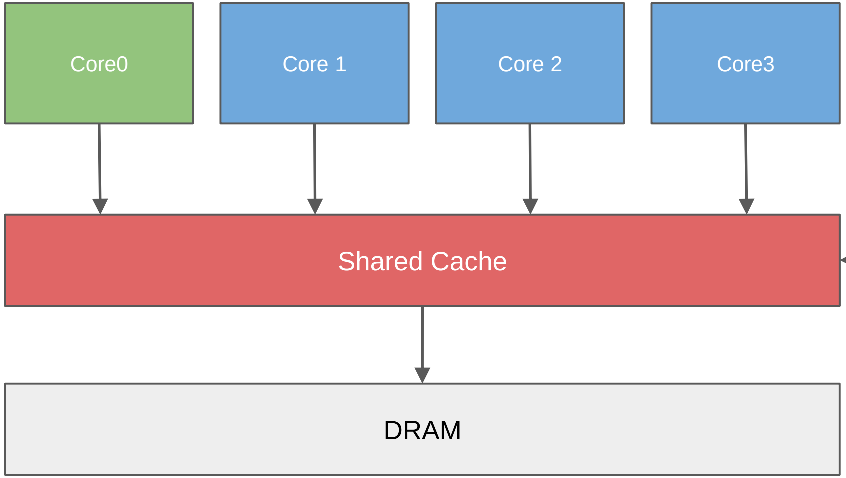
- Increasingly demanded in embedded real-time systems.
  - Provide improved performance.
  - Better satisfy size, weight and power (SWaP) constraints.





Multicore Platforms

- Worst case performance is unpredictable.
- Many resources are shared by all cores.



Shared caches are important resources.

Shared Cache

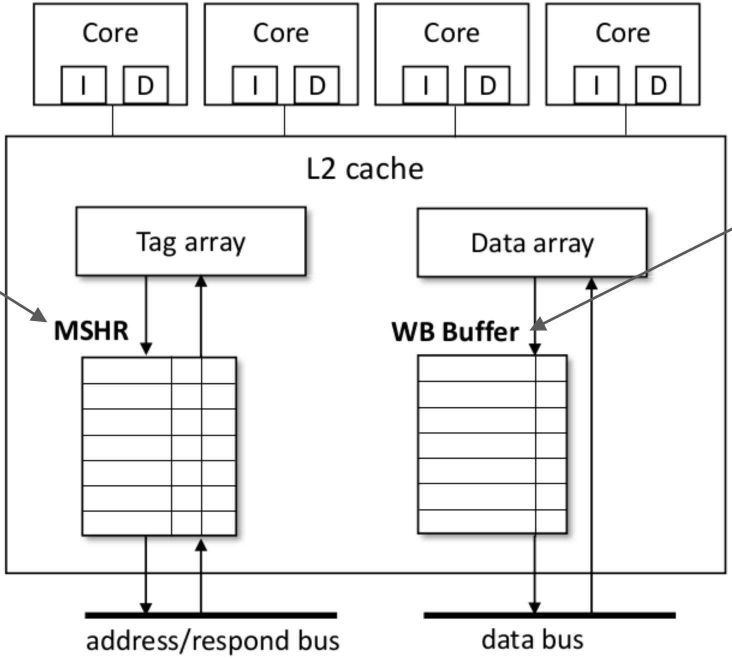
- Must handle requests from all cores.
- Support for **concurrent accesses** is vital for performance.
- Achieved through ***Non-Blocking Caches***.

Non-Blocking Cache

- Allow for multiple concurrent cache accesses.
  - Greatly improves performance.

Miss Status Holding Registers.

- Track outstanding cache misses.



Writeback Buffer.

- Holds evicted dirty lines (writebacks).
- Prevents cache refills from waiting.

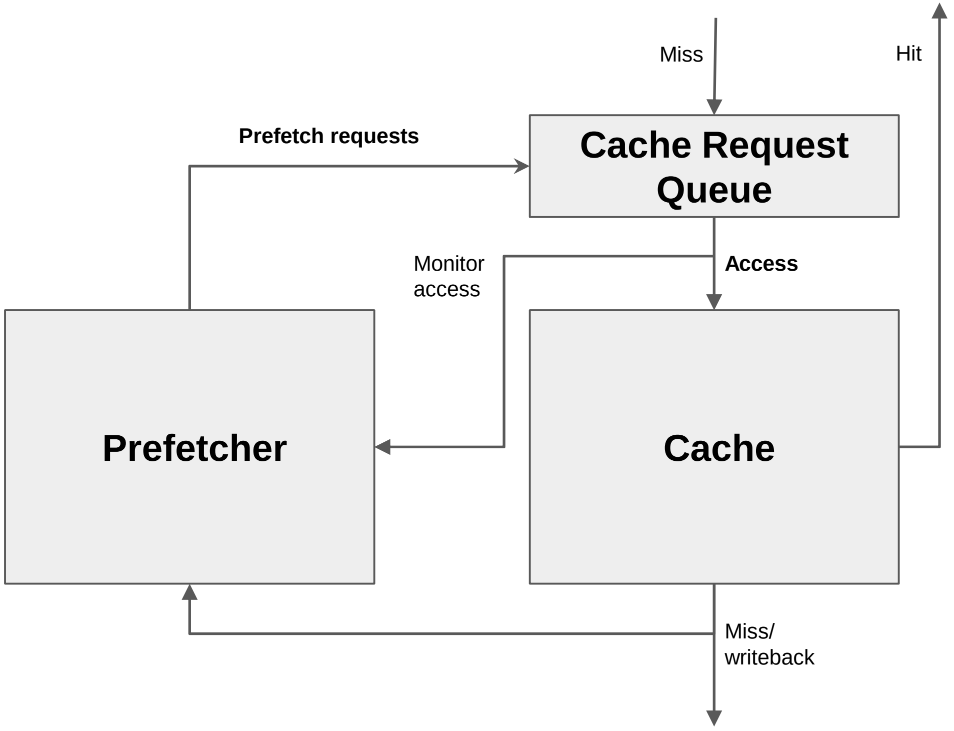
- **If either structure is full → cache block**

Shared Cache Blocking

- Cache blocking on a shared cache affects **all** cores.
  - No cores can access the cache.
  - Can significantly affect application timings.
- Unblocks when MSHRs and Writeback buffer have free entries.
  - **Unblocking can take a long time (memory access).**
- **Can be maliciously used by attackers.**

Hardware Prefetcher

- Predicts and loads future memory addresses into the cache.



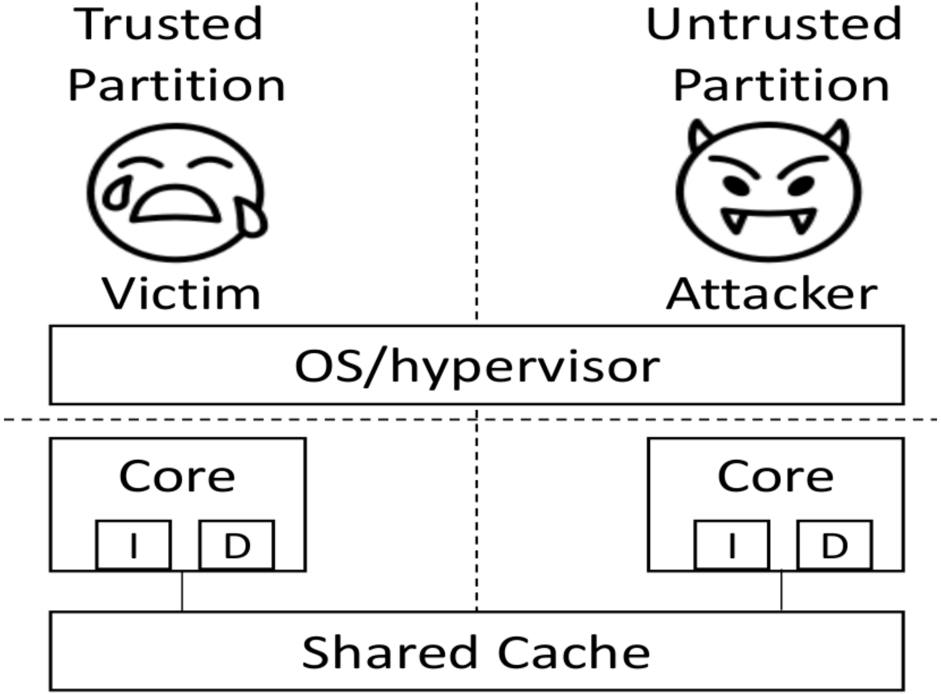
- Increases concurrent cache accesses.
- **Exacerbates cache blocking.**

Adopted from Professor Onur Mutlu's (CMU/ETHZ) Comp. Arch. lecture notes.

Outline

- Background
- **Threat Model/Code**
- Embedded Platform Evaluation
- Simulation
- OS-based Solution
- Conclusions

Threat Model



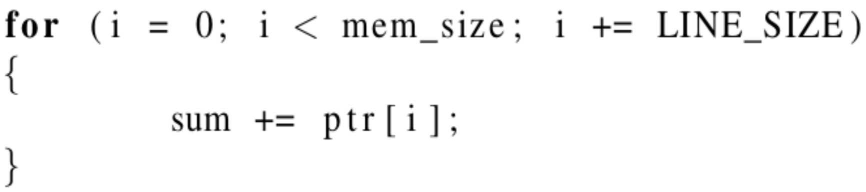
- Attackers can't directly affect the victim.
    - Core/memory isolation.
- Attackers can't run privileged code.
- **System has a shared cache.**

Cache DoS Attack

- Attackers can perform Denial-of-Service (DoS) attacks on the shared cache.
- MSHRs are a known attack vector[1].
- **Writeback buffer is also an attack vector.**

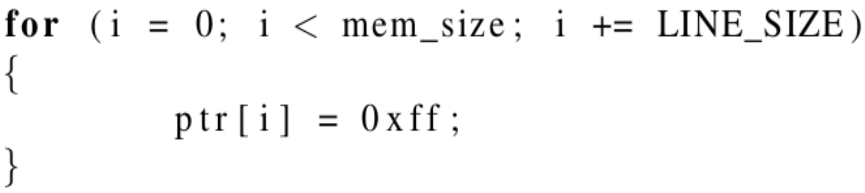
[1] Prathap Kumar Valsan, Heechul Yun, Farzad Farshchi. Taming Non-blocking Caches to Improve Isolation in Multicore Real-Time Systems. *IEEE Intl. Conference on Real-Time and Embedded Technology and Applications Symposium (RTAS)*, IEEE, 2016.

Cache DoS Attack Code



```
for (i = 0; i < mem_size; i += LINE_SIZE)
{
        sum += ptr[i];
}
```

Read Attacker
(BwRead)



```
for (i = 0; i < mem_size; i += LINE_SIZE)
{
        ptr[i] = 0xff;
}
```

Write Attacker
(BwWrite)

- Synthetic benchmarks that read from or write to a 1D array.
  - Generate continuous loads or stores.
- Working set size denoted in ():
  - BwRead(LLC): fits inside the LLC.
  - BwRead(DRAM): doesn’t fit inside the LLC.

Outline

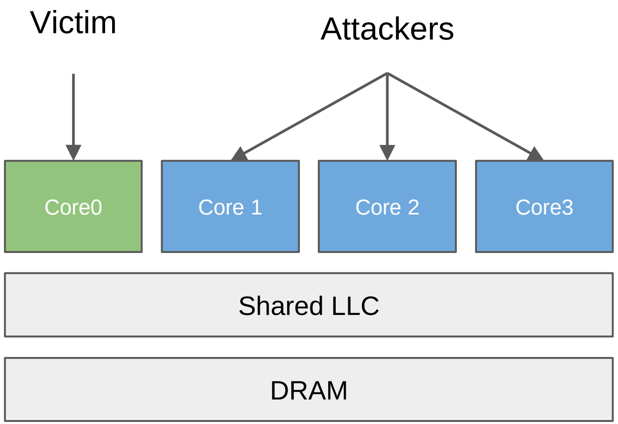
- Background
- Threat Model/Code
- **Embedded Platform Evaluation**
- Simulation
- OS-based Solution
- Conclusions

Tested Multicore Platforms

Platform	Raspberry Pi 3	Odroid C2	Raspberry Pi 2	Odroid XU4	
SoC	BCM2837	AmlogicS905	BCM2836	Exynos5422	
CPU	4x Cortex-A53	4x Cortex-A53	4x Cortex-A7	4x Cortex-A7	4x Cortex-A15
	in-order	**in-order**	**in-order**	**in-order**	**out-of-order**
	1.2GHz	1.5GHz	900MHz	1.4GHz	2.0GHz
Private Cache	32/32KB	32/32KB	32/32KB	32/32KB	32/32KB
Shared Cache	512KB (16-way)	512KB (16-way)	512KB (16-way)	512KB (16-way)	2MB (16-way)
Memory	1GB LPDDR2	2GB DDR3	1GB LPDDR2	2GB LPDDR3	
(Peak BW)	(8.5GB/s)	(12.8GB/s)	(8.5GB/s)	(14.9GB/s)	

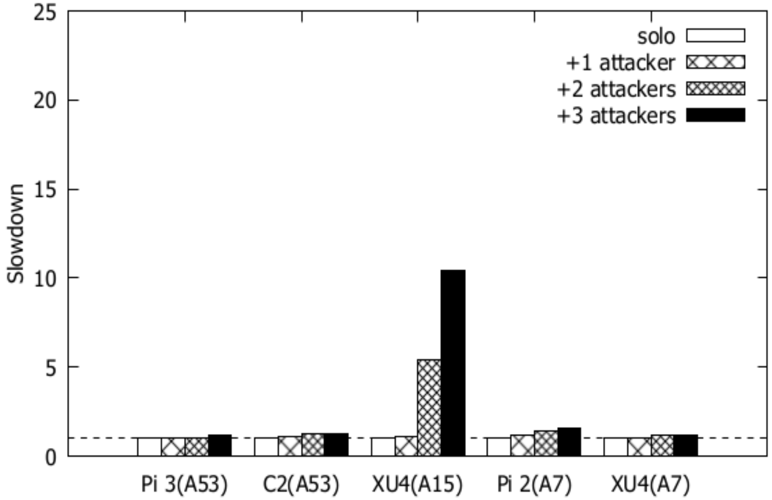
- Tests run across four platforms:
    - 3 CPU architectures: A53(in-order), A7(in-order), A15(OoO).

Cache DoS Attacks



- Measure the performance of the 'Victim'.
  - (1) Solo, and (2) with attackers.
- 'Victim' tasks:
  - BwRead(LLC).
  - EEMBC(L1) and SD-VBS(LLC).

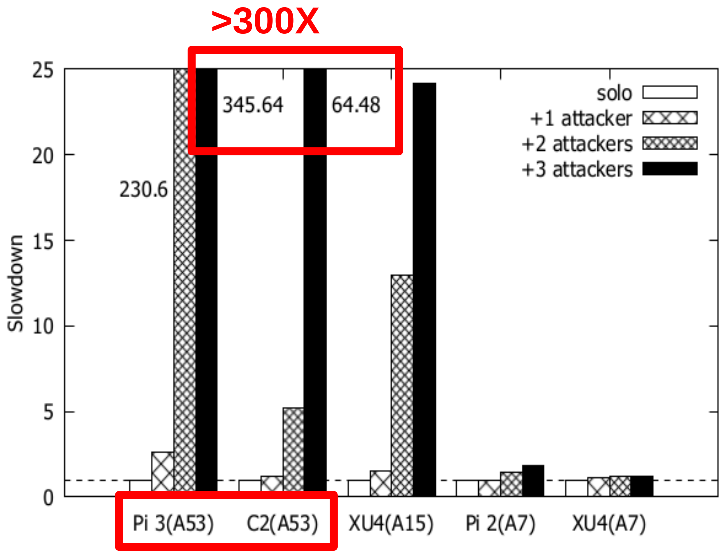
Effects of Cache Read DoS Attacks



- No effect on A53 or A7.
- Only A15 experiences slowdown.
  - MSHR contention[1].

[1] Prathap Kumar Valsan, Heechul Yun, Farzad Farshchi. Taming Non-blocking Caches to Improve Isolation in Multicore Real-Time Systems. *IEEE Intl. Conference on Real-Time and Embedded Technology and Applications Symposium (RTAS)*, IEEE, 2016.

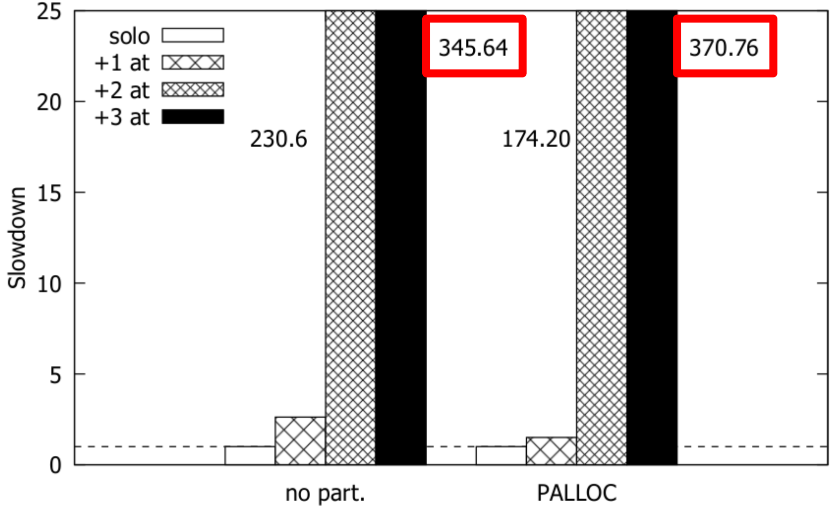
Effects of Cache Write DoS Attacks

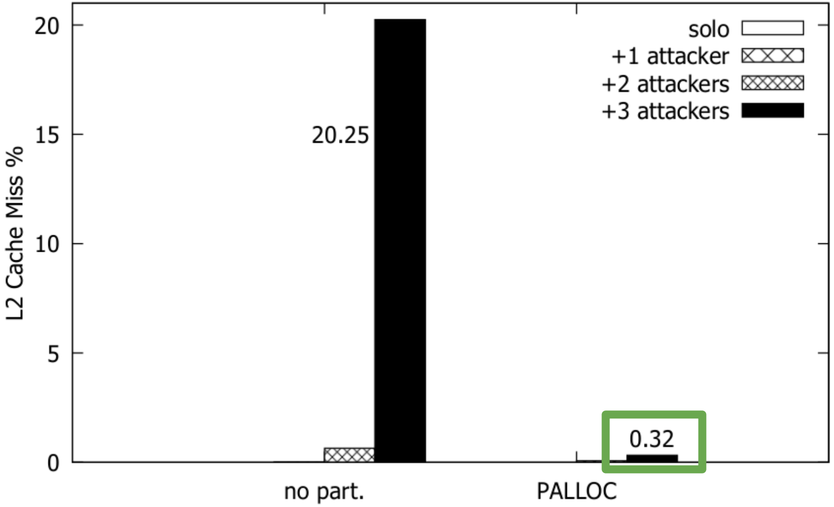


- **A53 experiences massive slowdown.**

Effect of Cache Partitioning (Pi 3)

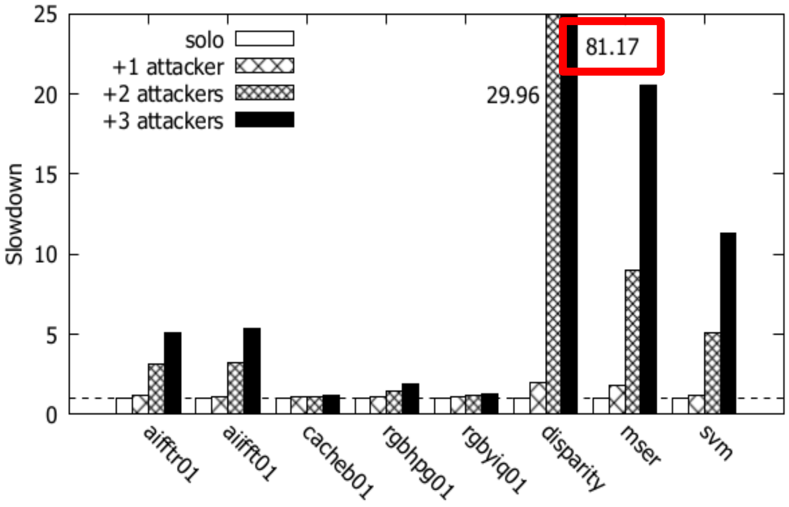
- Give each core a private fourth of the LLC.



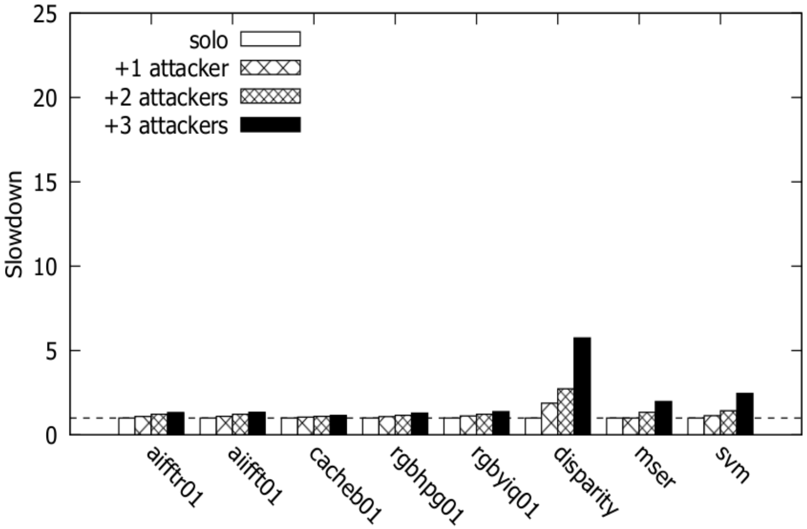


- **Partitioning doesn't protect against DoS attacks.**
  - Internal cache structures are not partitioned.

EEMBC and SD-VBS



Raspberry Pi 3 (A53)

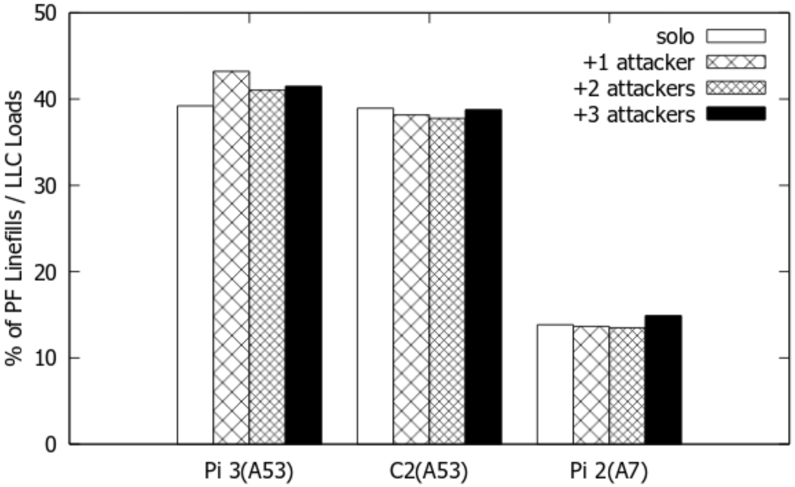


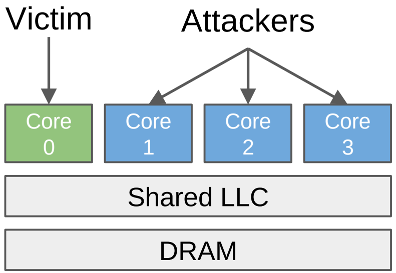
Raspberry Pi 2 (A7)

- **The *Pi 3 (A53)* is more susceptible to write DoS attacks.**
- **DoS attacks are more effective on *LLC sensitive victims* (SD-VBS).**

A53 vs A7

- A53 supports 3 outstanding L1D misses.
  - A7 only supports 1.





Victim

Attackers

Core 0

Core 1

Core 2

Core 3

Shared LLC

DRAM

- **A53 prefetchers generate more concurrent cache accesses.**

Hypothesis

Finding: write cache attackers are effective on A53, but not A7.

Why?

Hypothesis:

- A53 can generate more concurrent cache accesses (hardware prefetcher).
- Concurrent reads (read attacker) → stress MSHR.
- Concurrent writes (write attacker) → stress MSHR and **WB Buffer.**
- **Writeback buffer contention.**

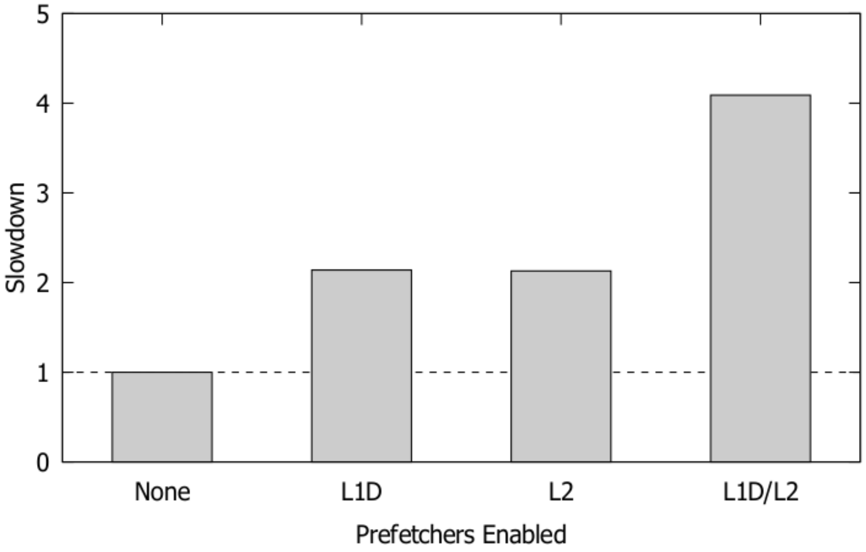
Outline

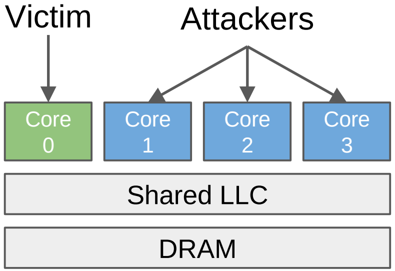
- Background
- Threat Model/Code
- Embedded Platform Evaluation
- **Simulation**
- OS-based Solution
- Conclusions

Simulation Environment

- Gem5 + Ramulator.
    - Quad-core CPU.
        - Adapt non-blocking private L1 and shared L2 caches.
    - **Configured to prevent MSHR contention.**
        - L1D misses + L2 prefetcher accesses < L2 MSHRs.
- Workload: cache write DoS attacks.
- **Vary prefetcher configuration and L2 Writeback Buffer size.**

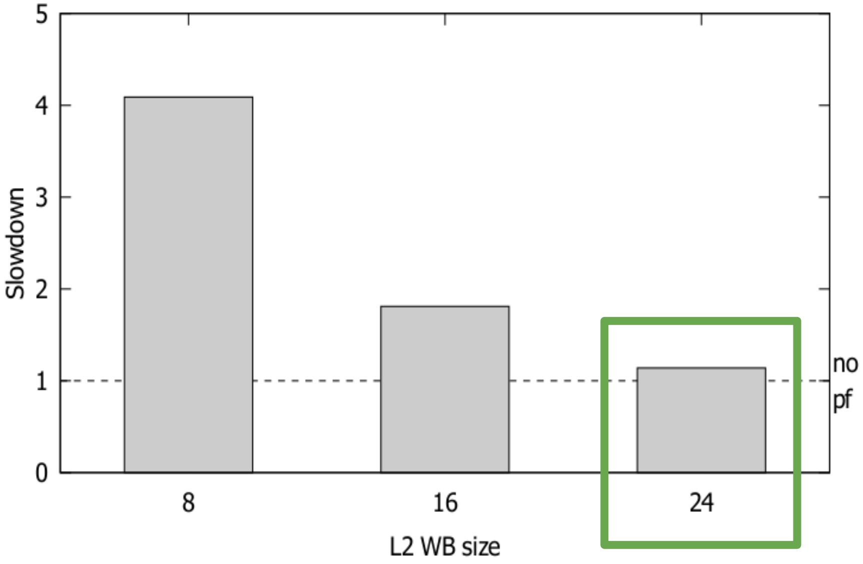
Effect of Hardware Prefetchers

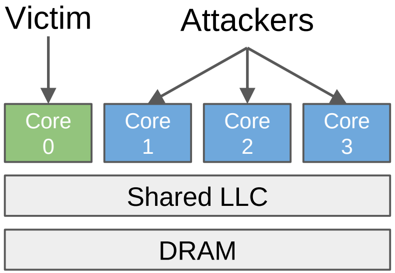




- Hardware prefetchers increase cache blocking.
  - ***Writeback buffer contention.***

Effect of Writeback Buffer Size





- Large WB size decreases cache blocking.
  - ***Reduces writeback buffer contention.***

Outline

- Background
- Threat Model/Code
- Embedded Platform Evaluation
- Simulation
- **OS-based Solution**
- Conclusions

OS-based Solution

- Idea: regulate writes more than reads.

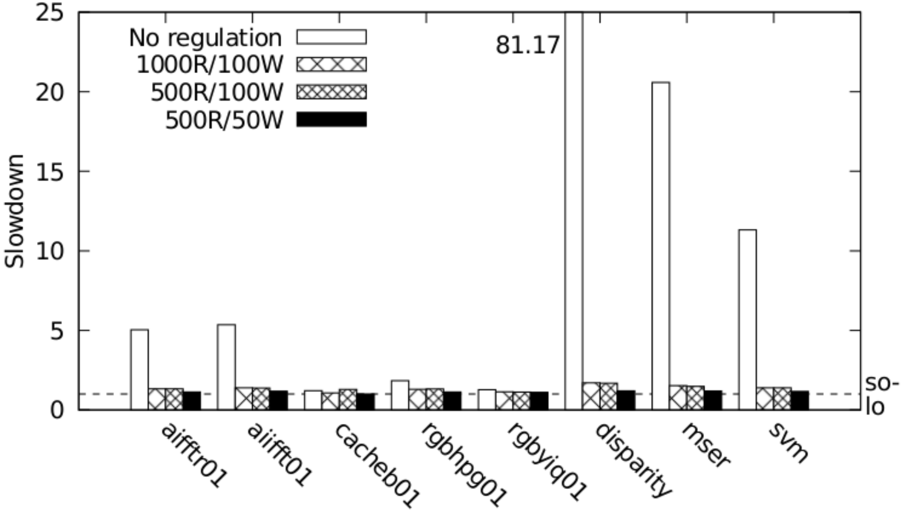
- MemGuard[1].
  - Regulate per-core memory traffic at a regular interval (1 ms).
  - Use LLC miss performance counter.
  - ***Treats reads and writes equally.***

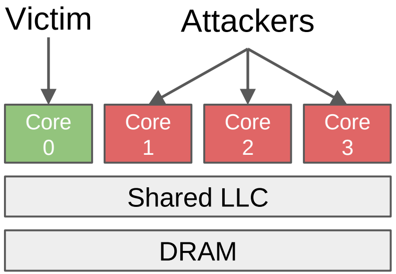
- Our extension
  - Use two performance counters: LLC miss and **LLC writeback.**
    - Separate read and write regulations.
  - **Low threshold for writes, and high threshold for reads.**

[1] Heechul Yun, Gang Yao, Rodolfo Pellizzoni, Marco Caccamo, and Lui Sha. MemGuard: Memory Bandwidth Reservation System for Efficient Performance Isolation in Multi-core Platforms. *IEEE Intl. Conference on Real-Time and Embedded Technology and Applications Symposium (RTAS)*, IEEE, 2013.

Effect of R/W Regulation

- Re-run DoS attacks on EEMBC and SD-VBS with extended solution.





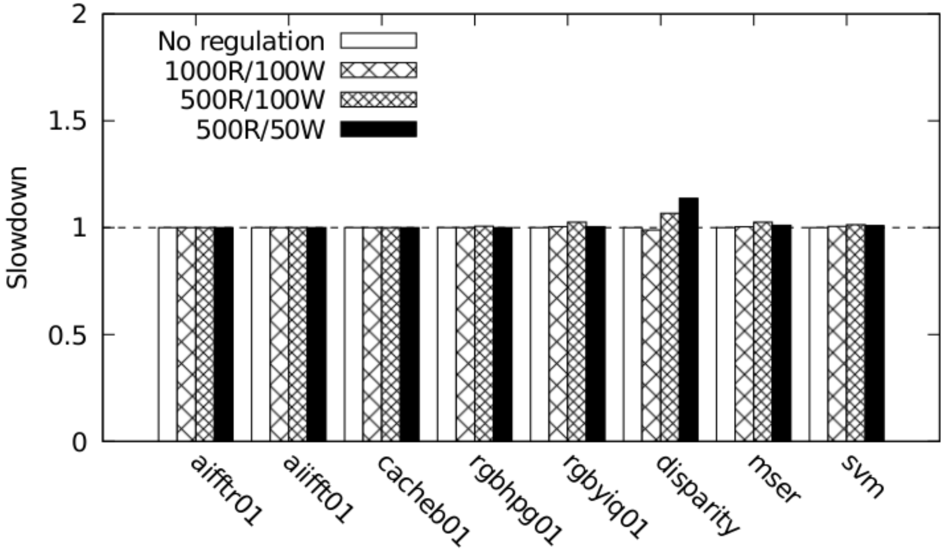
3 R/W values (MB/s):

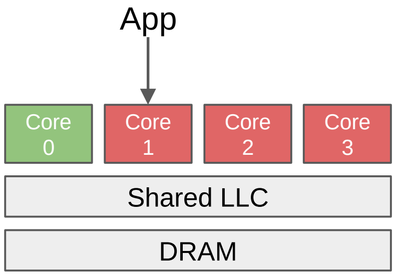
- 1000R / 100W
- 500R / 100W
- 500R / 50W

- **Effectively protects against cache DoS attacks.**

Effect R/W Regulation on Non-attacker Apps

- Run real-world benchmarks on regulated cores.




App

Core 0

Core 1

Core 2

Core 3

Shared LLC

DRAM

3 R/W values (MB/s):

- 1000R / 100W
- 500R / 100W
- 500R / 50W

- **Minimal impacts on normal applications.**

Outline

- Background
- Threat Model/Code
- Embedded Platform Evaluation
- Simulation
- OS-based Solution
- **Conclusions**

Conclusions

- We observe extreme impacts of cache write DoS attacks.
  - Can cause over **300X** slowdown on an actual platform.
- Through simulation, we identify an internal cache structure, the **Writeback buffer**, as a potential attack vector.
- We propose an OS-based solution to mitigate these DoS attacks.
  - Can successfully do so with little to no impact on non-attacking tasks.

Thank you

Disclaimer:

This research is supported by NSF CNS 1718880, CNS 1815959,
and NSA Science of Security initiative contract #H98230-18-D-0009.