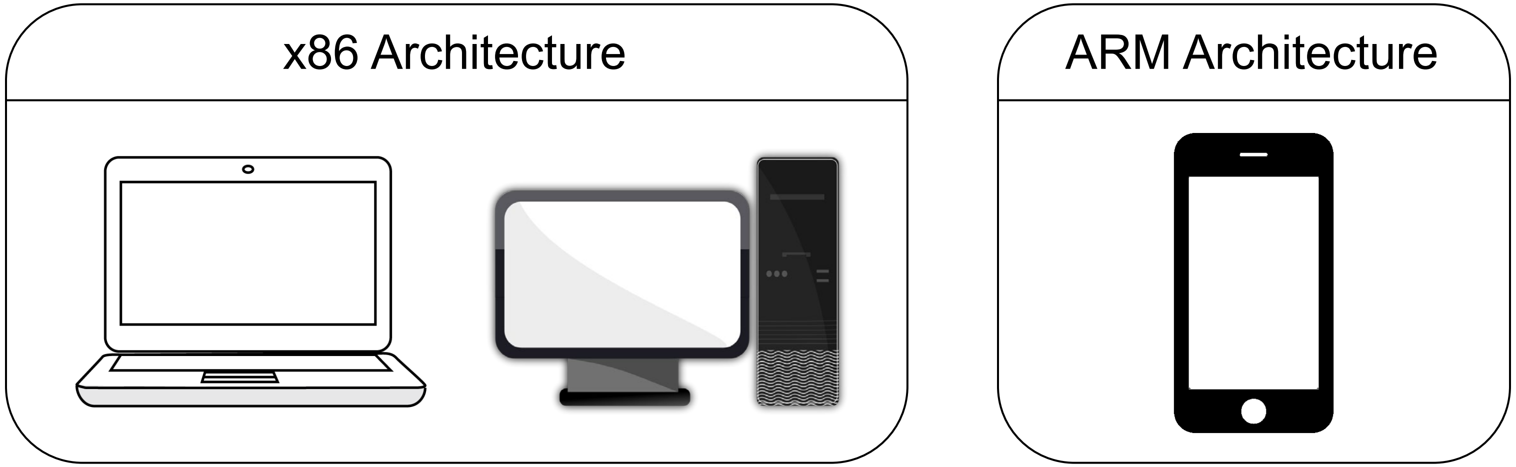
An Exploration of ARM System-Level Cache and GPU Side Channels

Patrick Cronin[*], Xing Gao[*], Haining Wang[+], Chase Cotton[*]

University of Delaware[*], Virginia Tech[+]

Computer Architecture – Then

- For many years laptop and desktops have been dominated by x86 while mobile devices are dominated by ARM

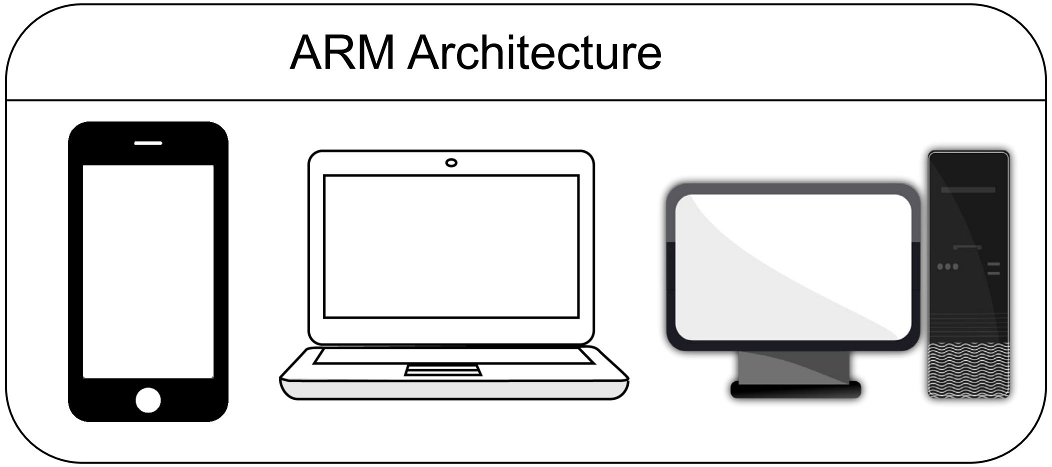


x86 Architecture

ARM Architecture

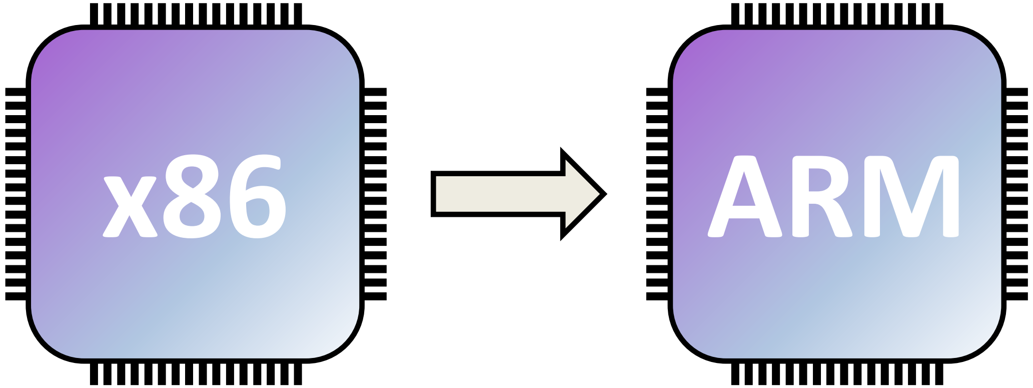
Sharing Too Much

- Apple has switched all of their new products to ARM based devices and Windows vendors are starting to follow suit



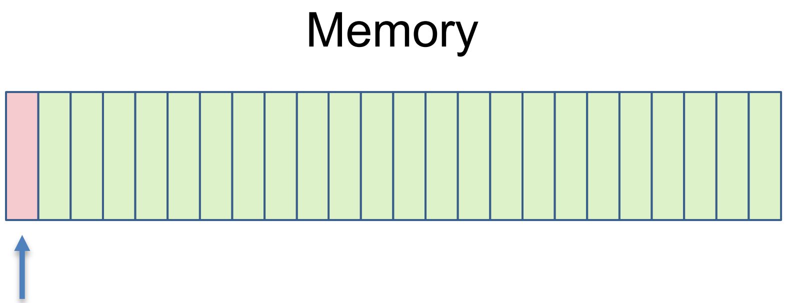
Sharing Too Much

- ARM processor architecture rapidly gaining popularity and acceptance in consumer systems
  - Provides new vectors and easier access to previously x86 only side channel attacks
  - Examine whether same mistakes from previous systems carry over to new ARM devices



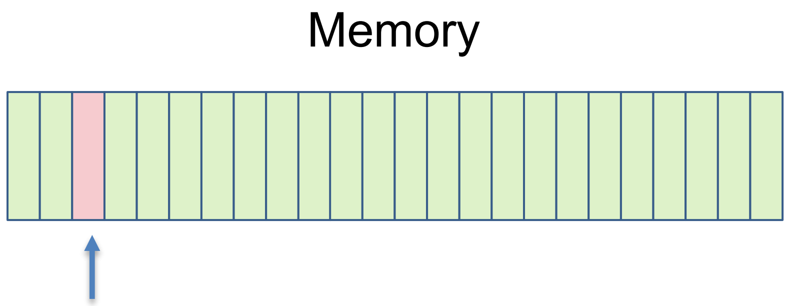
Attacking CPUs – Cache Side Channels

- Computer systems operate on memory
- Memory accesses can be very slow
- Many operations are in a pattern or predictable



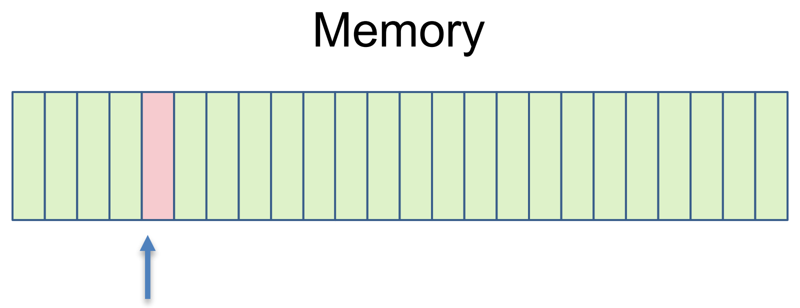
Attacking CPUs – Cache Side Channels

- Computer systems operate on memory
- Memory accesses can be very slow
- Many operations are in a pattern or predictable



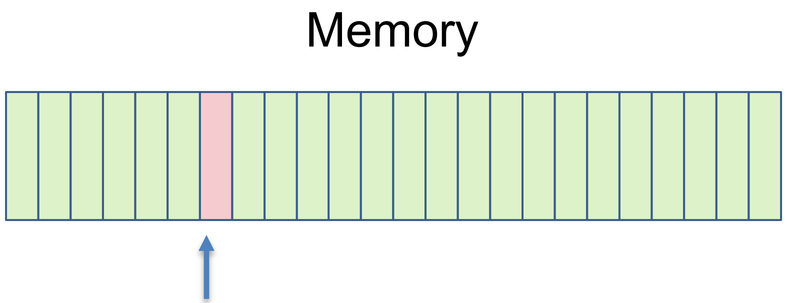
Attacking CPUs – Cache Side Channels

- Computer systems operate on memory
- Memory accesses can be very slow
- Many operations are in a pattern or predictable



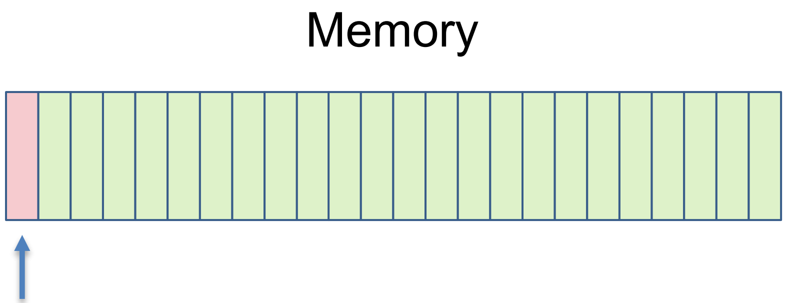
Attacking CPUs – Cache Side Channels

- Computer systems operate on memory
- Memory accesses can be very slow
- Many operations are in a pattern or predictable



Attacking CPUs – Cache Side Channels

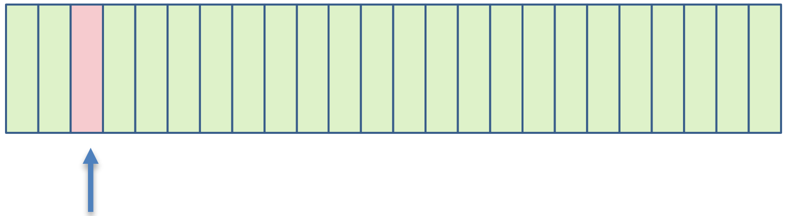
- Computer systems operate on memory
- Memory accesses can be very slow
- Many operations are in a pattern or predictable



Attacking CPUs – Cache Side Channels

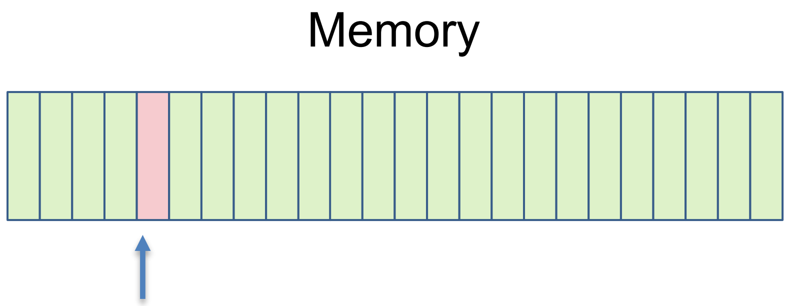
- Computer systems operate on memory
- Memory accesses can be very slow
- Many operations are in a pattern or predictable

Memory



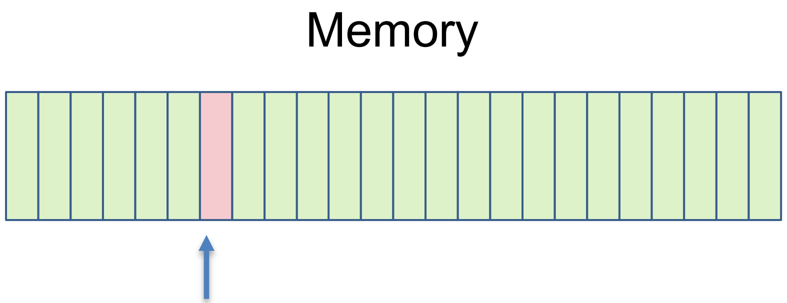
Attacking CPUs – Cache Side Channels

- Computer systems operate on memory
- Memory accesses can be very slow
- Many operations are in a pattern or predictable



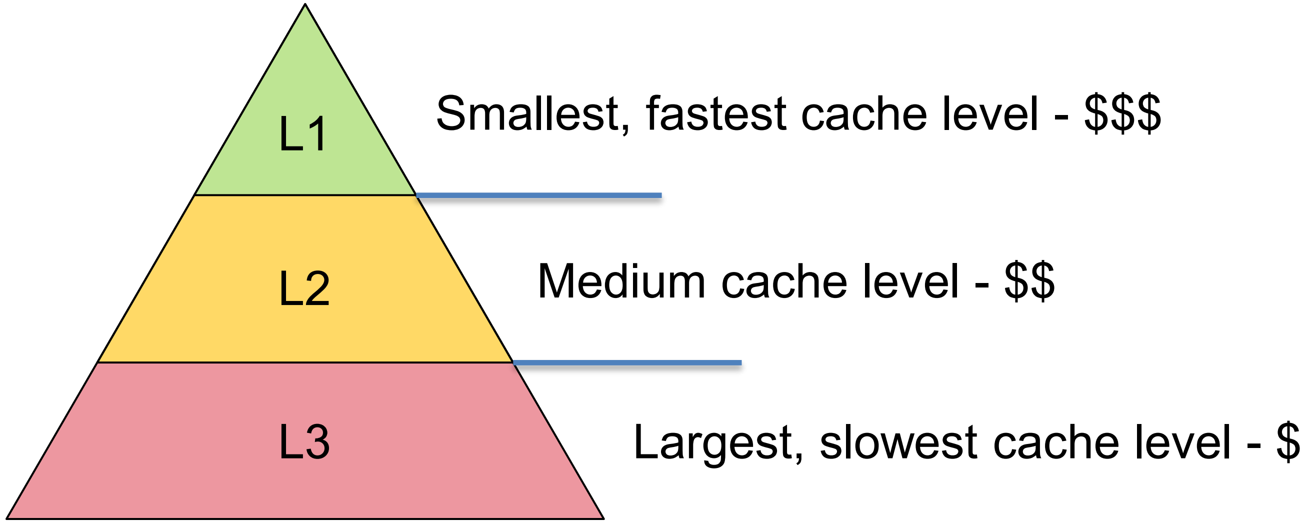
Attacking CPUs – Cache Side Channels

- Computer systems operate on memory
- Memory accesses can be very slow
- Many operations are in a pattern or predictable



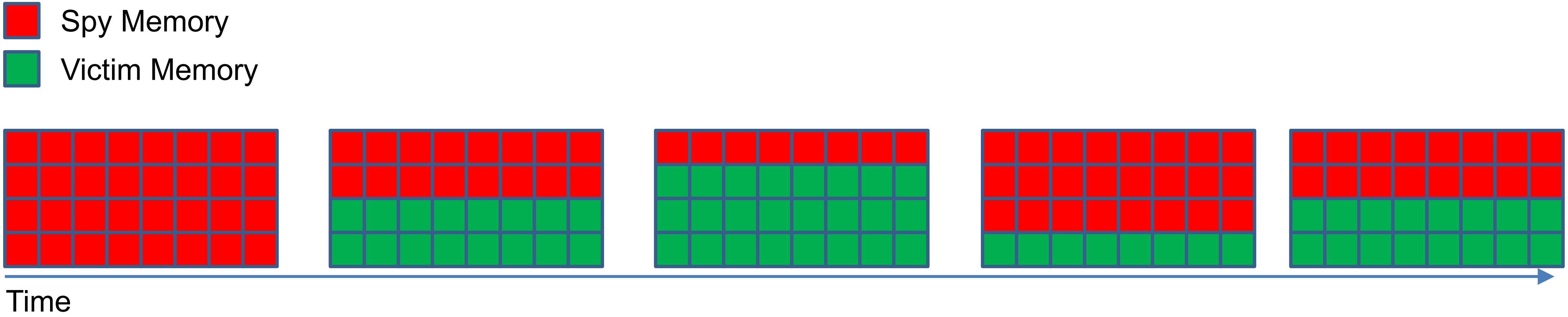
Attacking CPUs – Cache Side Channels

- Caches exploit the patterns in memory access
- Increase speed of the system at reasonable cost


L1
Smallest, fastest cache level - $$$
L2
Medium cache level - $$
L3
Largest, slowest cache level - $

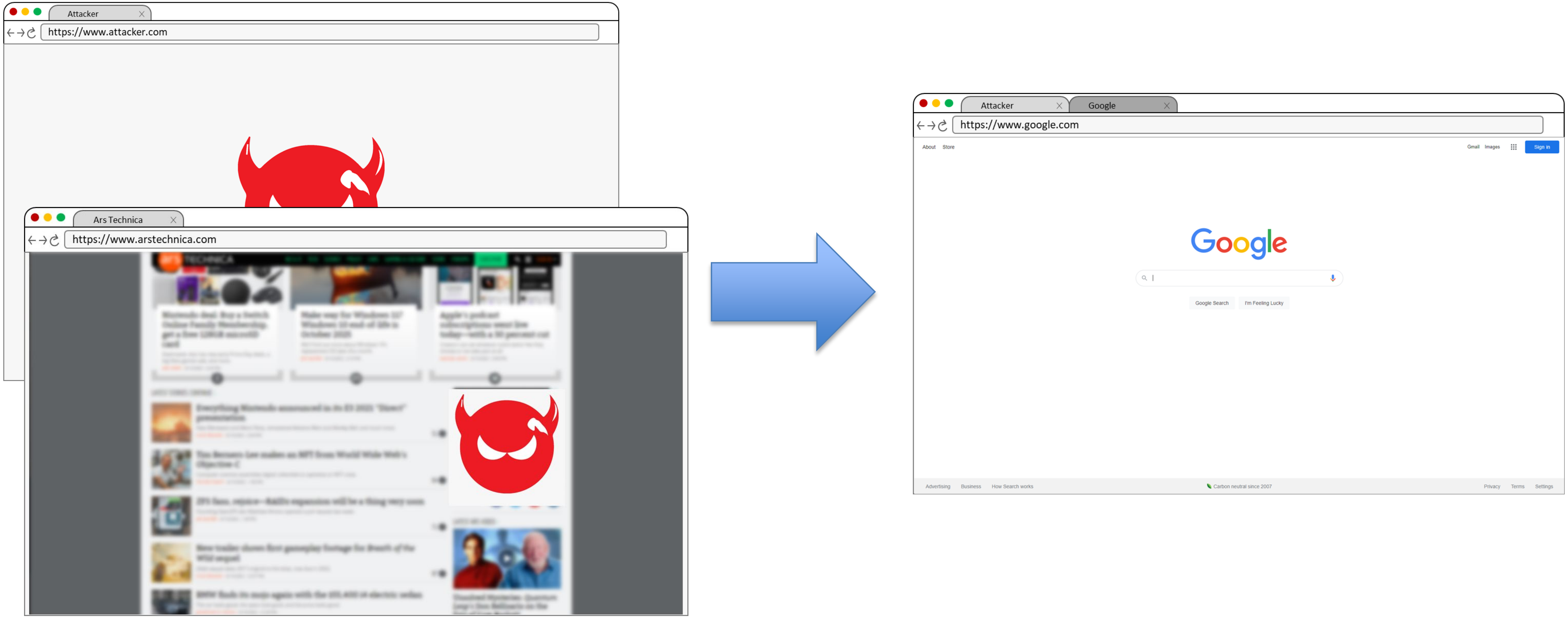
Revisting x86 - Cache Occupancy Channel

- [7] suggests a cache occupancy channel can be utilized to fingerprint websites and study this in x86
- The spy claims the entire cache and times how long it takes to access. As the victim runs, the cache is impacted and a timing feature can be extracted



[7] Shusterman, A., Kang, L., Haskal, Y., Meltser, Y., Mittal, P., Oren, Y., & Yarom, Y. (2019). Robust website fingerprinting through the cache occupancy channel. In *Proceedings of the 28th USENIX Security Symposium*

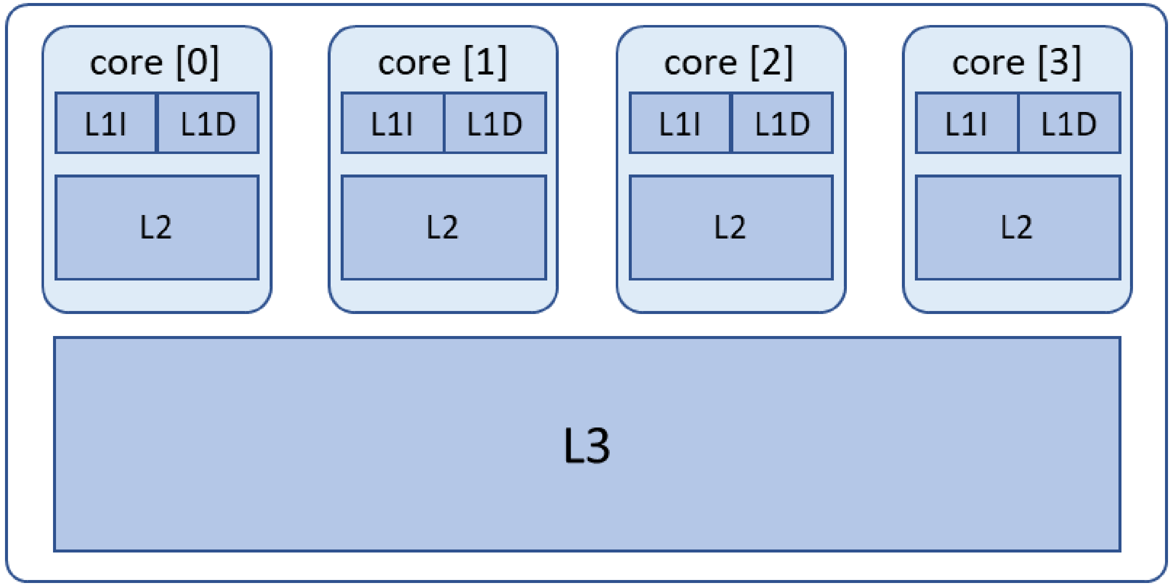
Website Fingerprinting Attack – Process



How is ARM Different?

- x86 processors utilize straightforward cache design

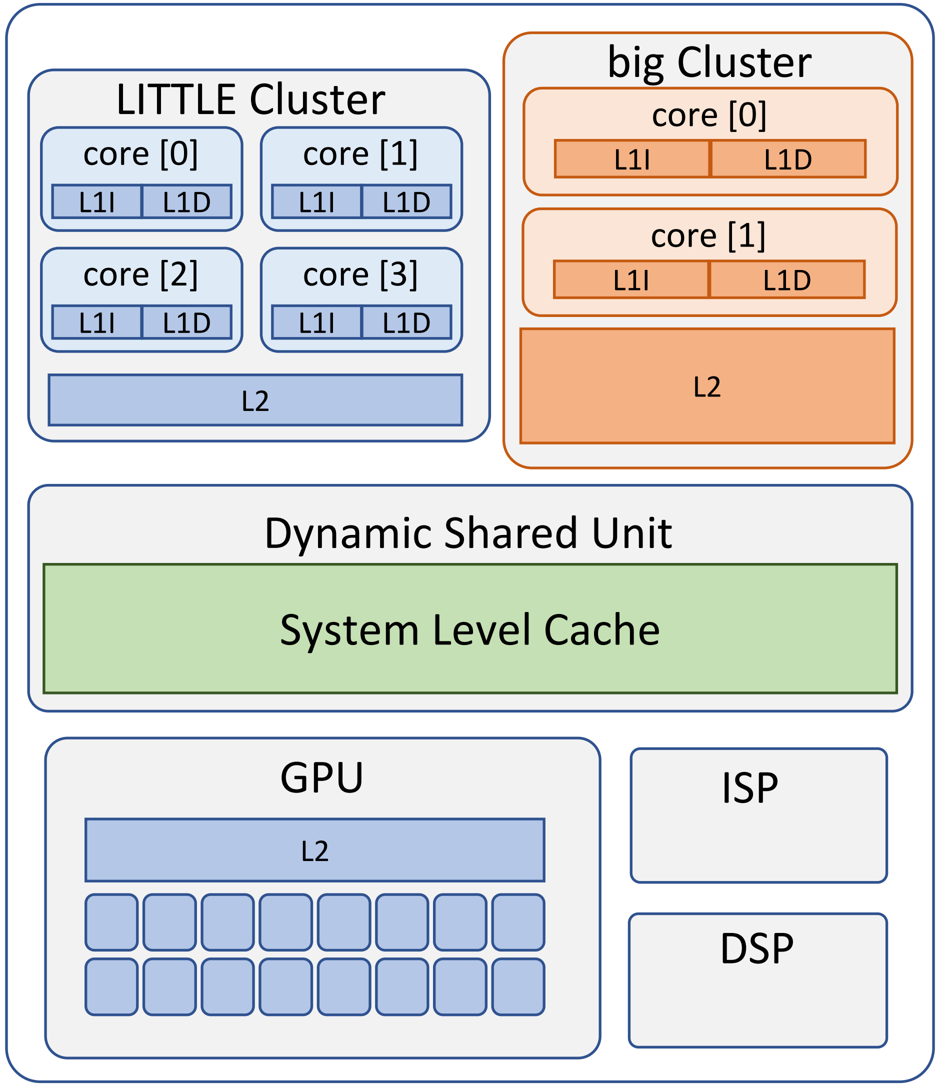
x86 Architecture



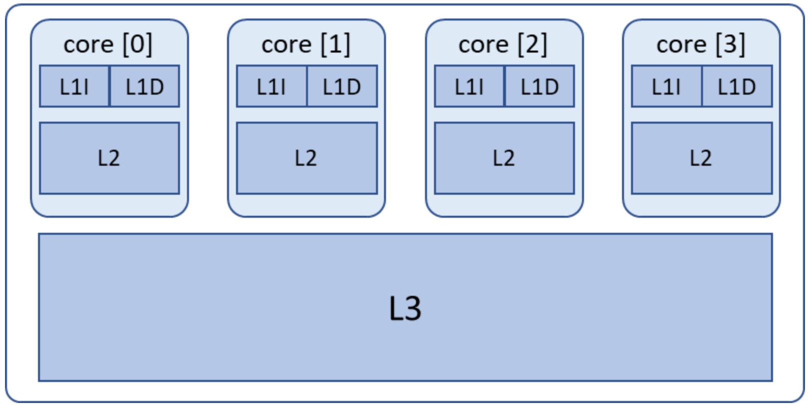
How is ARM Different?

- ARM employs DynamIQ architecture and vastly different cache strategies w/ Integrated Accelerators

ARM big.LITTLE Architecture
ARM DynamIQ Architecture

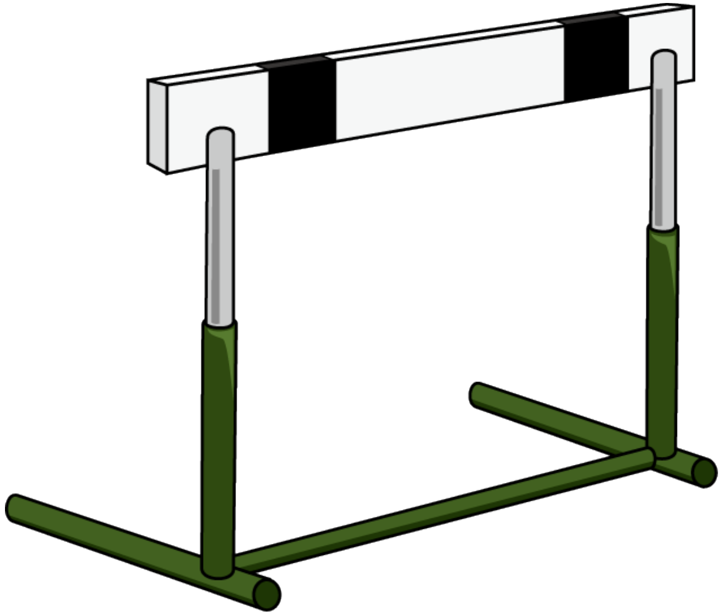


x86 Architecture



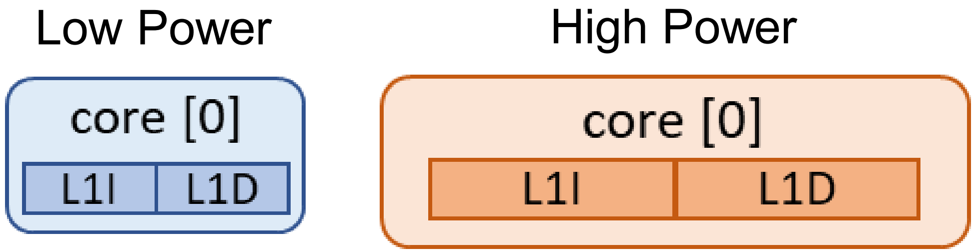
Adjusting the Attack for ARM

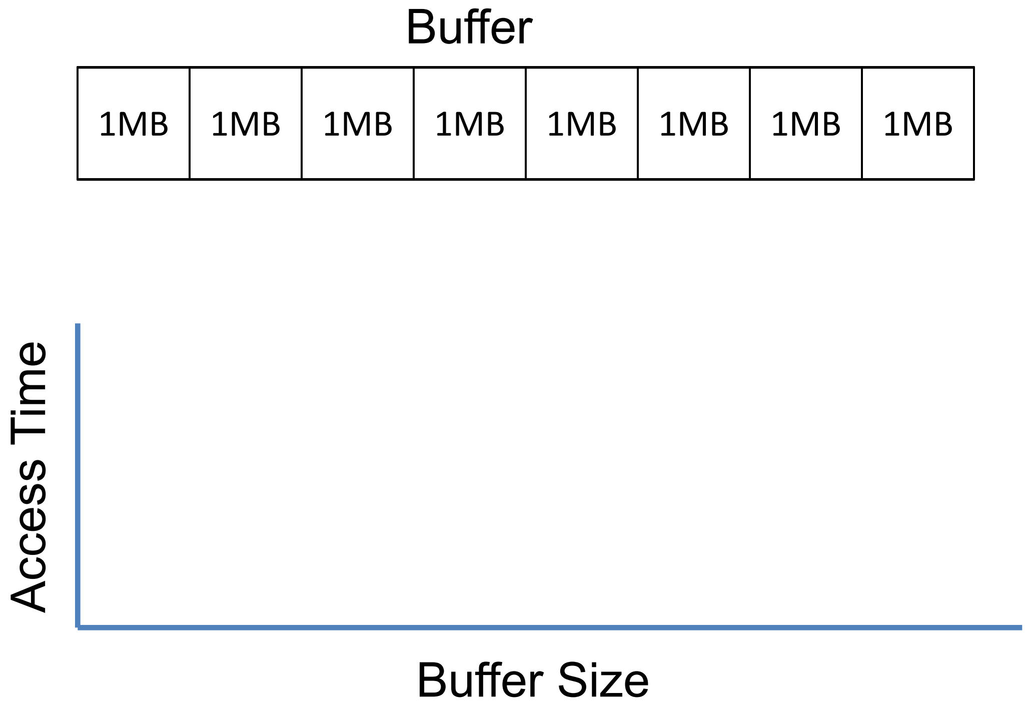
- ARM has heterogeneous processors which run at different frequencies
- ARM caches are designed with different algorithms than their x86 counterparts



Adjusting x86 Attacks to ARM – Core Types

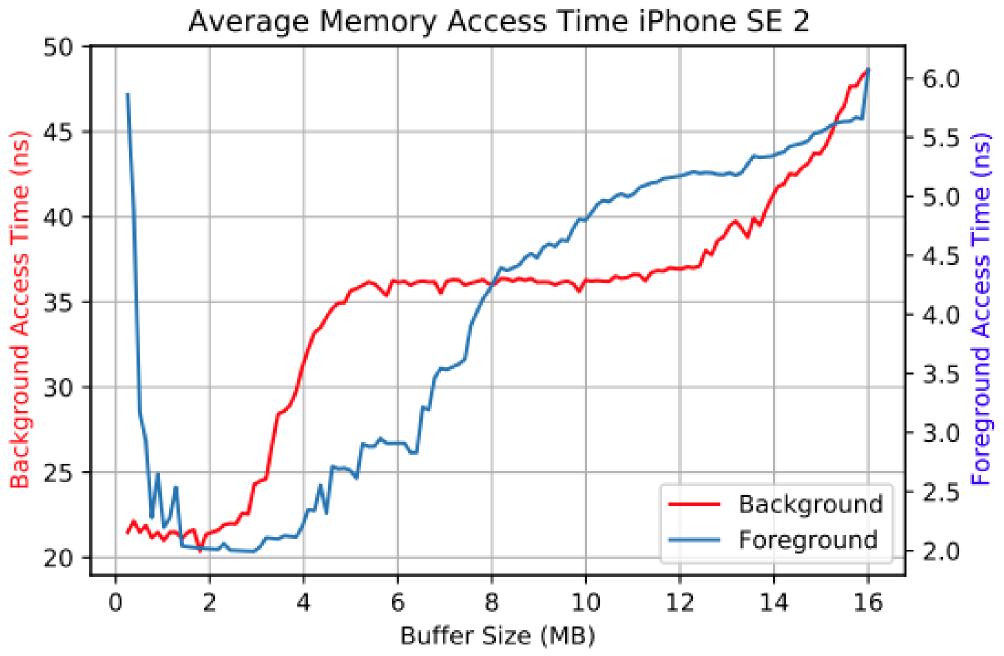
- ARM SoC can contain multiple different core types





Adjusting x86 Attacks to ARM – Core Types

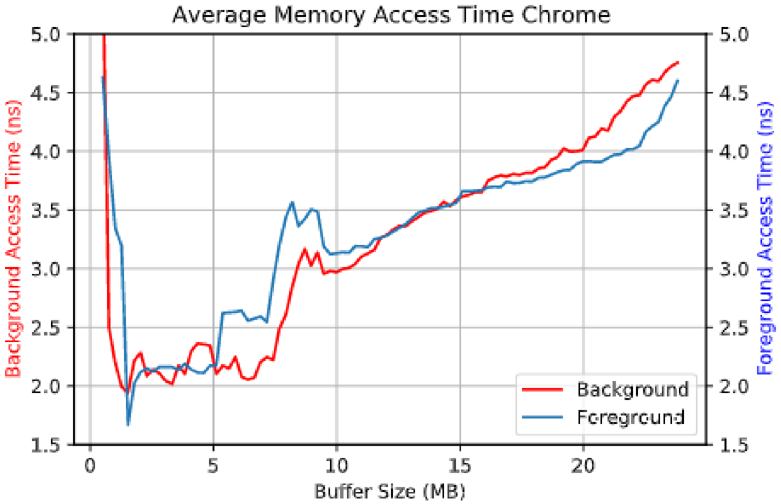
- ARM Schedulers take advantage of High and Low power cores

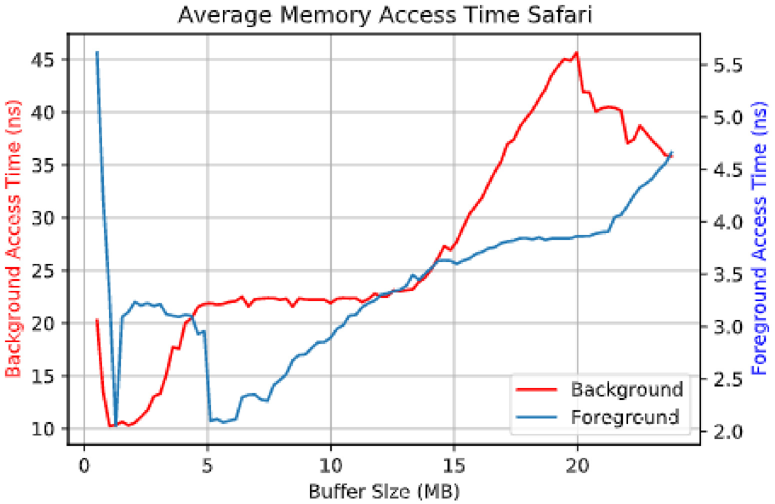


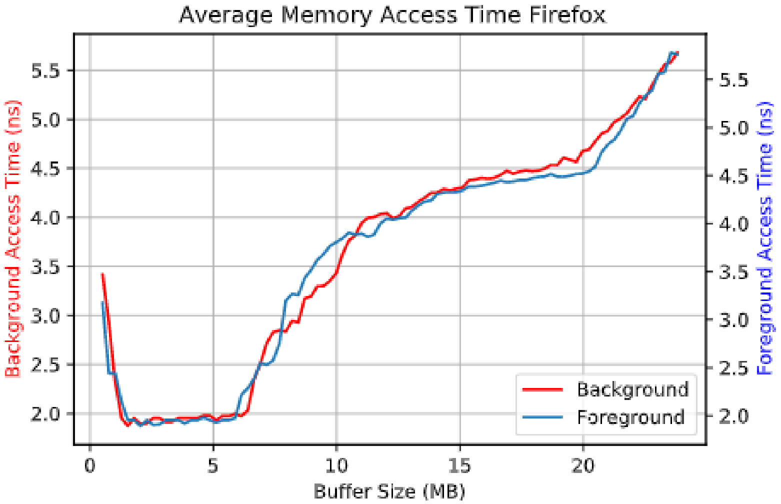
- 10x difference in access speed on iPhone SE2 with foreground vs background web tab
- Differently shaped cache activity
- Caused by energy aware scheduler moving background tab to low cores

Adjusting x86 Attacks for ARM – Browsers

- Each browser has its own JavaScript engine and memory management



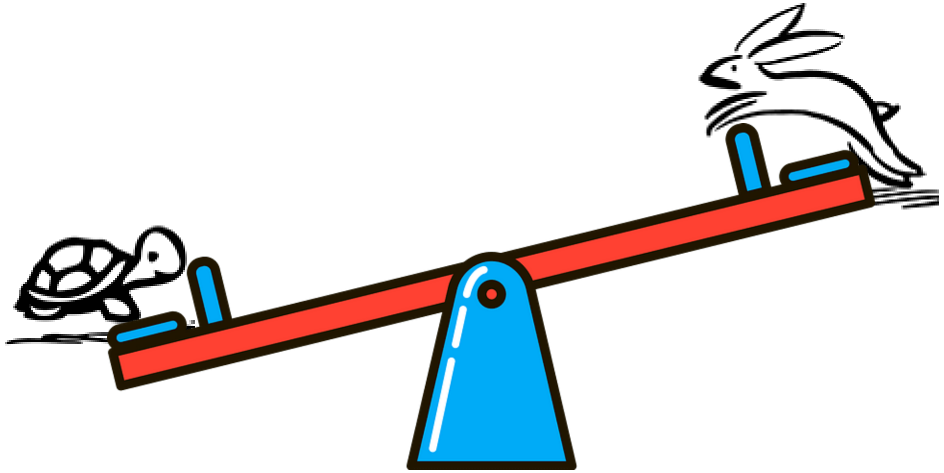




Buffer size must be carefully chosen

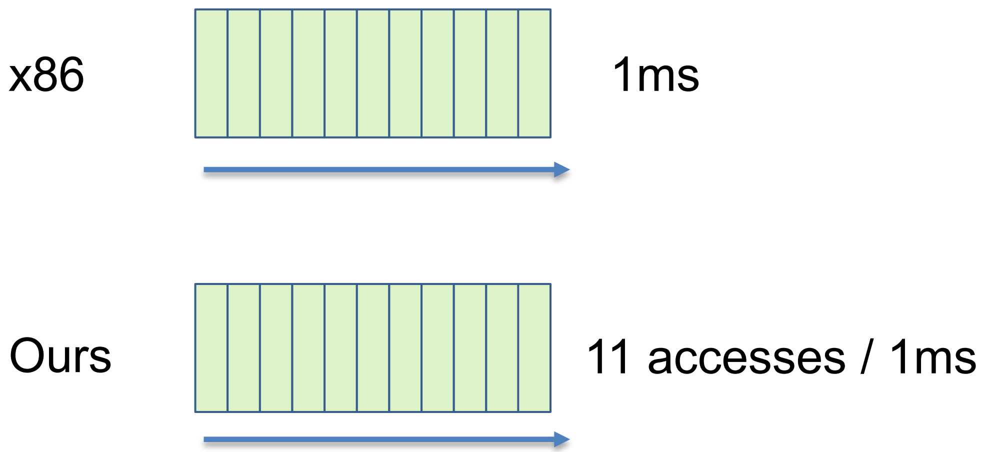
Adjusting x86 Attacks for ARM – Timing

- Constant war between high frequency sampling and access time
- Careful balancing act
  - Too Slow – won’t sample often enough
  - Too Fast – long downtime between samples



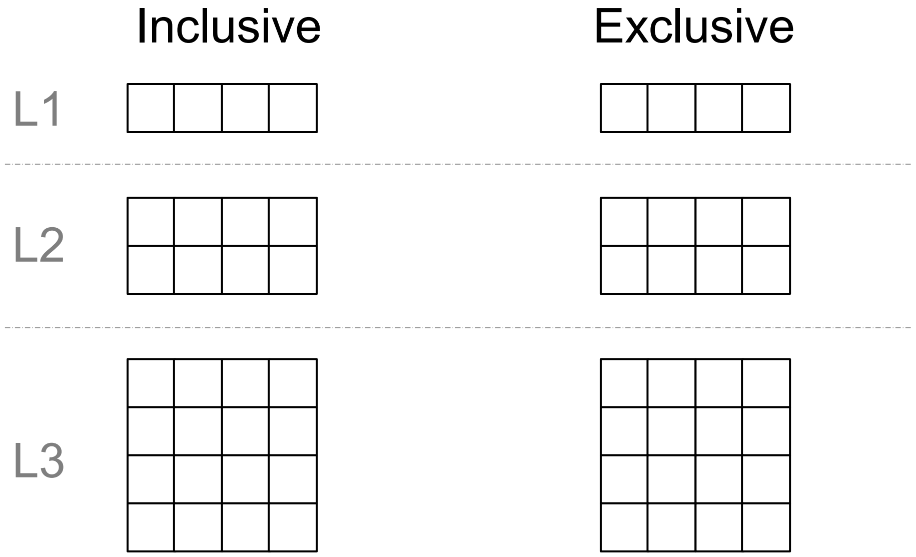
Adjusting x86 Attacks for ARM - Timing

- Invert measurement pattern
- Measure the number of accesses in the time period
- High granularity measurement always!



Adjusting x86 Attacks for ARM – Invert

- Major Drawback
  - Exclusive caching



Inclusive

Exclusive

L1

L2

L3

Access 1

Access 2

Access 3

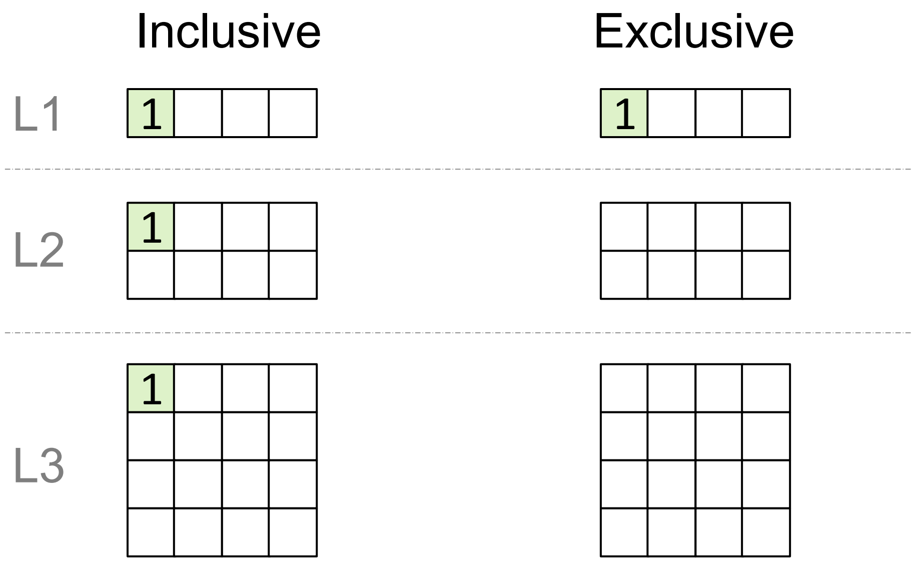
Access 4

Access 5

Access 6

Adjusting x86 Attacks for ARM – Invert

- Major Drawback
  - Exclusive caching



Access 1

Access 2

Access 3

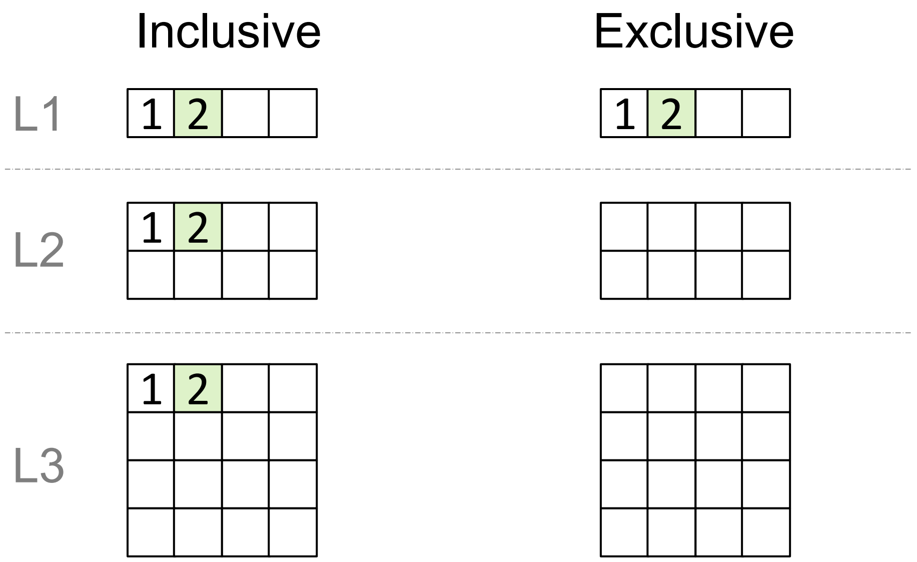
Access 4

Access 5

Access 6

Adjusting x86 Attacks for ARM – Invert

- Major Drawback
  - Exclusive caching



Access 1

Access 2

Access 3

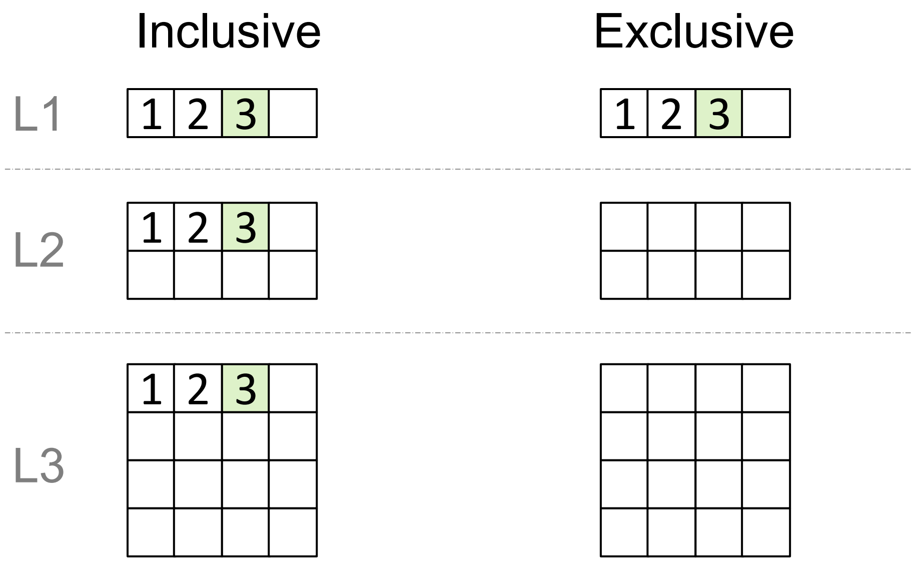
Access 4

Access 5

Access 6

Adjusting x86 Attacks for ARM – Invert

- Major Drawback
  - Exclusive caching



Inclusive

Exclusive

L1

1 2 3

1 2 3

L2

1 2 3

L3

1 2 3

Access 1

Access 2

Access 3

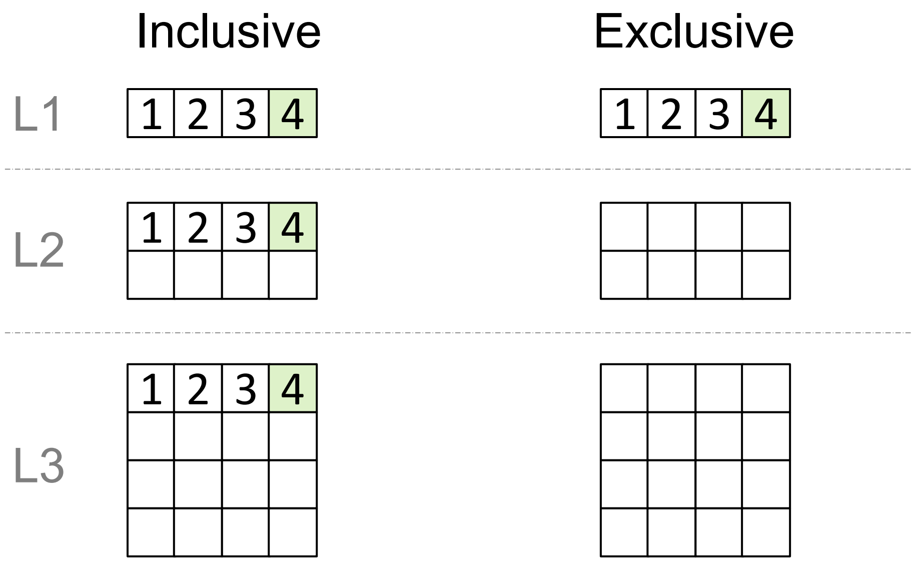
Access 4

Access 5

Access 6

Adjusting x86 Attacks for ARM – Invert

- Major Drawback
  - Exclusive caching



Access 1

Access 2

Access 3

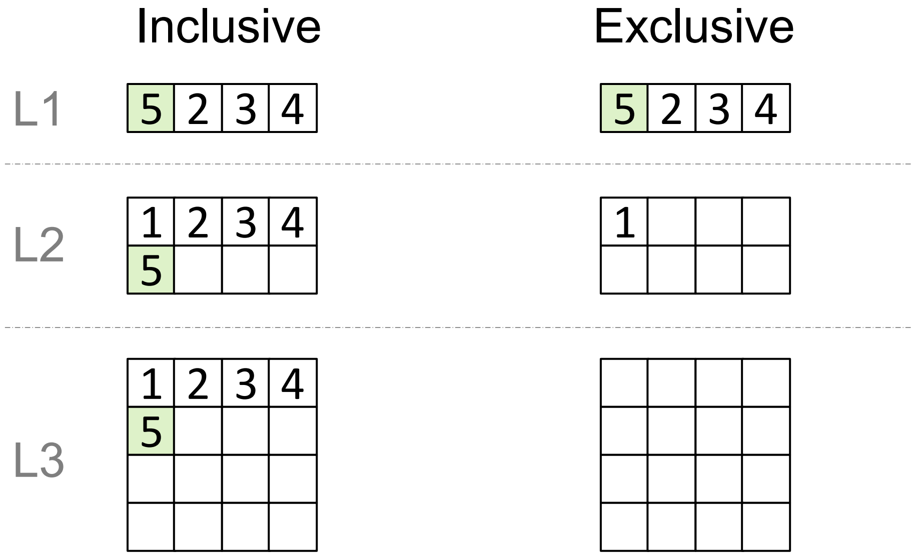
Access 4

Access 5

Access 6

Adjusting x86 Attacks for ARM – Invert

- Major Drawback
  - Exclusive caching



Access 1

Access 2

Access 3

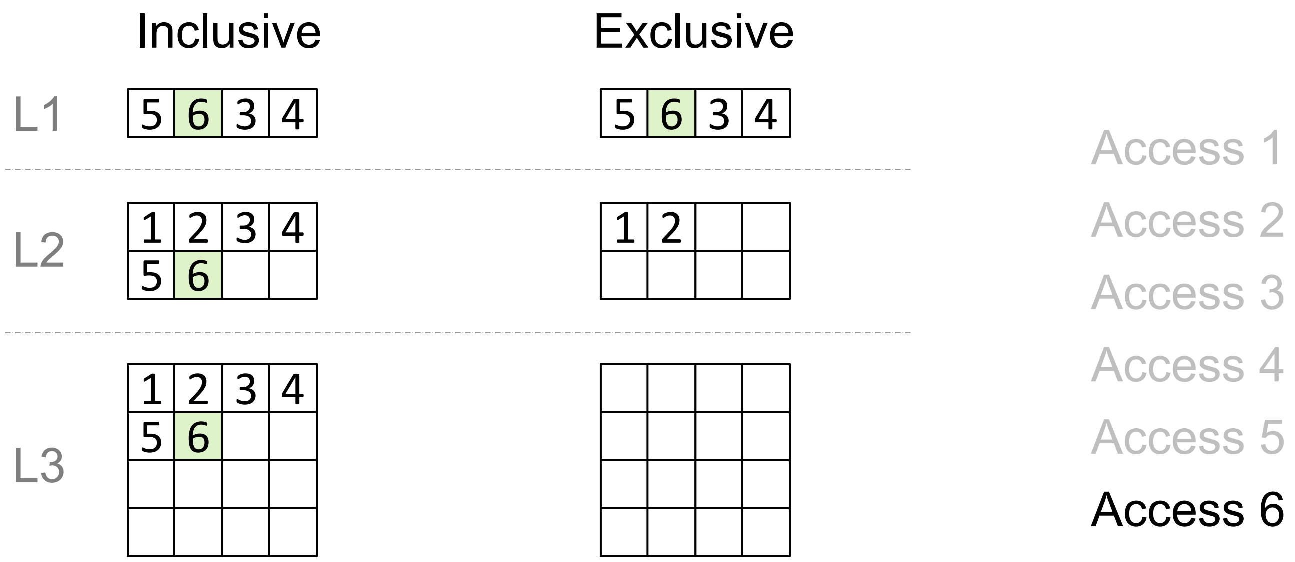
Access 4

Access 5

Access 6

Adjusting x86 Attacks for ARM – Invert

- Major Drawback
  - Exclusive caching



Inclusive

5	6	3	4

L1

1	2	3	4
5	6		

L2

1	2	3	4
5	6		

L3

Exclusive

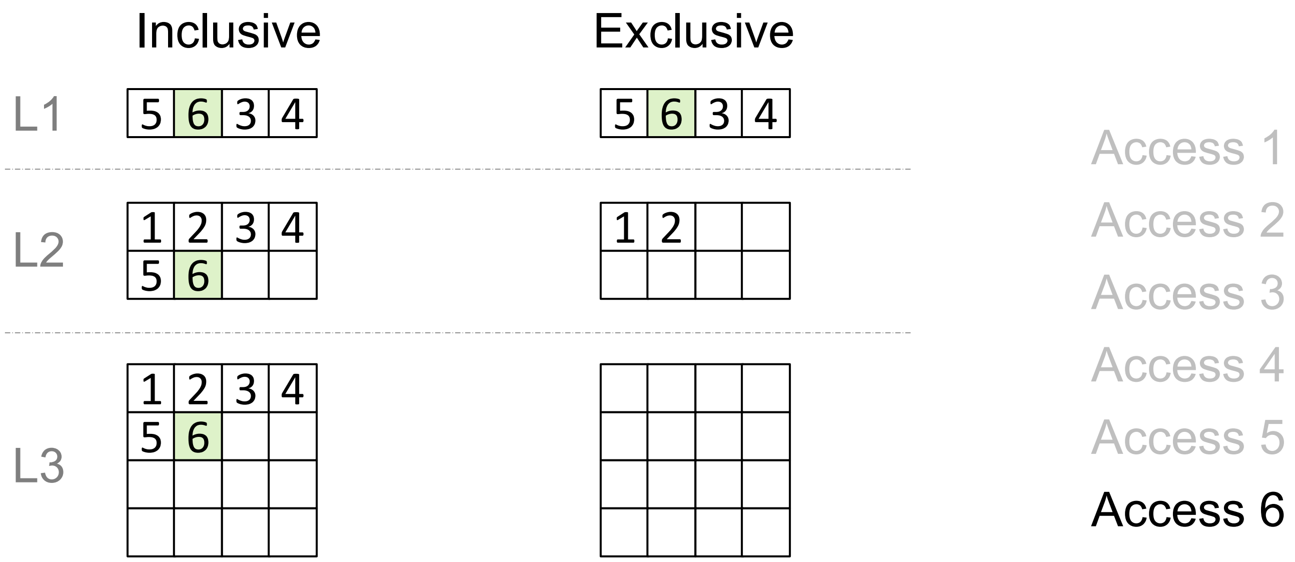
5	6	3	4

1	2		

Access 1
Access 2
Access 3
Access 4
Access 5
Access 6

Adjusting x86 Attacks for ARM – Invert

- Major Drawback
  - Exclusive caching
- Exclusive caching mainly for design density
- If we size our buffer incorrectly, we won't affect the cache!



	Inclusive	Exclusive
L1	5 6 3 4	5 6 3 4
L2	1 2 3 4 5 6	1 2
L3	1 2 3 4 5 6	

Access 1
Access 2
Access 3
Access 4
Access 5
Access 6

Website Fingerprinting Attack

- Closed World
  - Only test against sensitive websites
- Open World
  - Try to identify sensitive websites from many websites

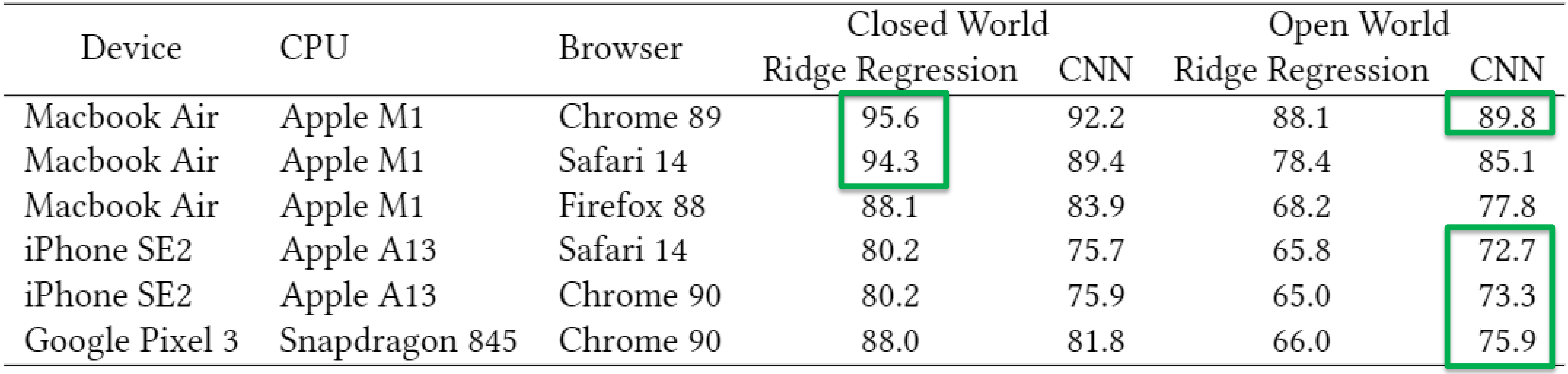
Closed World Experiments

- 100 Accesses to top 100 Websites
- Randomize Access Order to Ensure Fairness

Open World Experiments

- 100 Accesses to top 100 Websites
- 1 Access to 5,000 other Websites
- Randomize Access Order to Ensure Fairness

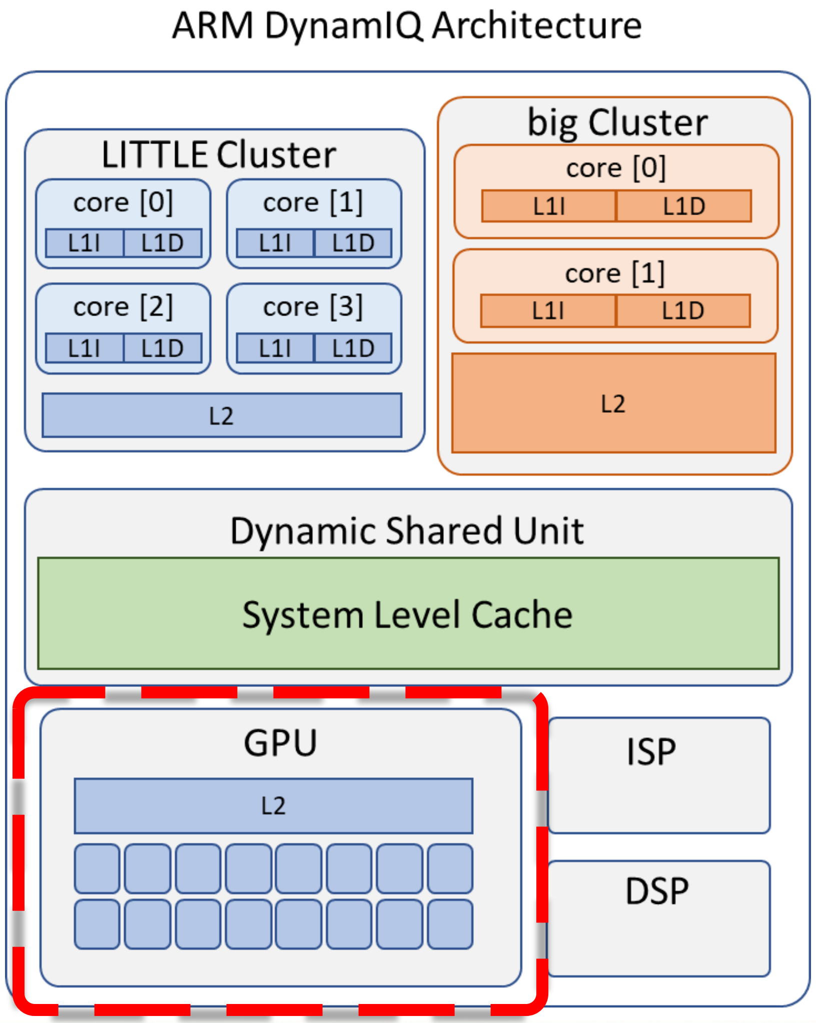
Results – Web-Based



Device	CPU	Browser	Closed World		Open World	
			Ridge Regression	CNN	Ridge Regression	CNN
Macbook Air	Apple M1	Chrome 89	95.6	92.2	88.1	89.8
Macbook Air	Apple M1	Safari 14	94.3	89.4	78.4	85.1
Macbook Air	Apple M1	Firefox 88	88.1	83.9	68.2	77.8
iPhone SE2	Apple A13	Safari 14	80.2	75.7	65.8	72.7
iPhone SE2	Apple A13	Chrome 90	80.2	75.9	65.0	73.3
Google Pixel 3	Snapdragon 845	Chrome 90	88.0	81.8	66.0	75.9

Crafting Another Contention Channel

- The dynamic shared unit interacts with multiple peripherals on the device
- Web content is hardware accelerated by GPU
- Can the GPU act as another channel?



Accessing the GPU from JavaScript

- WebGL/WebGL2
  - Animations, video, 3D experiences
  - Focused on *visuals – 60Hz*
- WebGPU
  - Updates WebGL for computing
  - Supported in beta
- GPU.js
  - Allows quick creation of compute kernels

GPU Contention Challenges

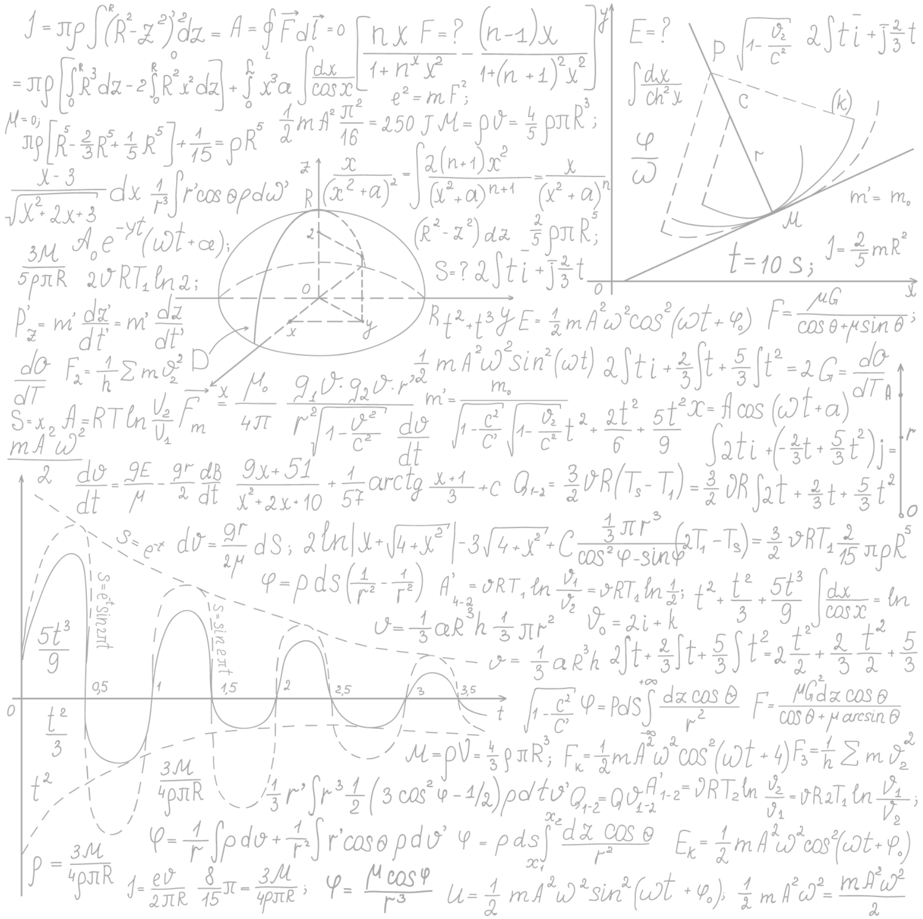
- How do we measure GPU Contention?
- How do we create GPU Contention?

Measuring GPU Contention

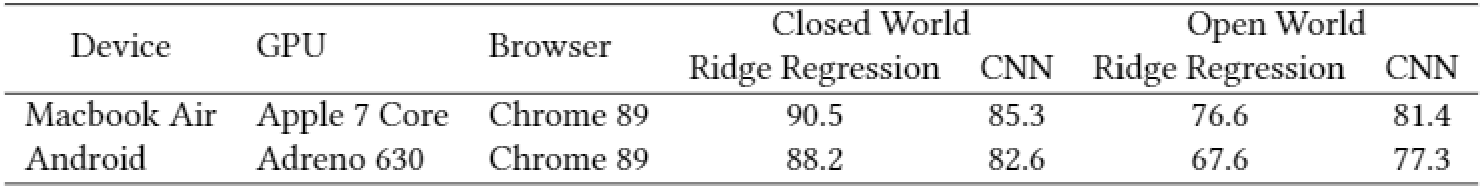
- Cannot interrupt GPU kernel to check time
  - Browser developers removed timing ability due to exploits
- Time completions of kernel instead of interrupting kernel
  - Better granularity if we have very short kernel

Creating GPU Contention

- Matrix Multiplication
  - Very computation heavy
- Dot product
  - Lower complexity, but still lots of multiplication
- Sum array row
  - Minimal complexity
  - Access each element only once



GPU Contention Channel Results



Device	GPU	Browser	Closed World		Open World	
			Ridge Regression	CNN	Ridge Regression	CNN
Macbook Air	Apple 7 Core	Chrome 89	90.5	85.3	76.6	81.4
Android	Adreno 630	Chrome 89	88.2	82.6	67.6	77.3

Better performance on the Google Pixel 3!

Contention – Summary

- Examined 2 contention channels in ARM based devices
- Investigate how the different scheduling of heterogeneous core operating systems effects contention channels
  - Shared cache contention channel demonstrated up to 89% accurate open world attack
  - Novel GPU contention channel performed up to 2% better than cache contention channel on Android open world

Questions?