REPRODUCING SPECTRE ATTACK WITH GEM5: HOW TO DO IT RIGHT?

PIERRE AYOUB - EURECOM

PIERRE.AYOUB@EURECOM.FR

CLÉMENTINE MAURICE - UNIV LILLE, CNRS, INRIA

CLEMENTINE.MAURICE@INRIA.FR

EUROSEC'21

26 APRIL, 2021

INTRODUCTION

WHOAMI

WHOAMI

Intern



Intern





Intern



Ph.D. Student



Intern



Ph.D. Student



ABOUT THIS PRESENTATION





• Can we simulate transient execution attacks? \Rightarrow **Spectre**.





- Can we simulate transient execution attacks? \Rightarrow **Spectre**.
- Why?





- Can we simulate transient execution attacks? \Rightarrow **Spectre**.
- Why?
 - Rely on tiny CPU details to work \Rightarrow **difficult to simulate**.





- Can we simulate transient execution attacks? \Rightarrow **Spectre**.
- Why?
 - Rely on tiny CPU details to work \Rightarrow **difficult to simulate**.
 - Hard to study and reproduce \Rightarrow **simulation could be helping**.

SPECTRE



- Can we simulate transient execution attacks? \Rightarrow **Spectre**.
- Why?
 - Rely on tiny CPU details to work \Rightarrow **difficult to simulate**.
 - Hard to study and reproduce ⇒ **simulation could be helping**.
- Comparing a **real system** and a **simulated system**:





- Can we simulate transient execution attacks? \Rightarrow **Spectre**.
- Why?
 - Rely on tiny CPU details to work \Rightarrow **difficult to simulate**.
 - Hard to study and reproduce \Rightarrow **simulation could be helping**.
- Comparing a **real system** and a **simulated system**:
 - **Raspberry Pi**, ARM processor.





- Can we simulate transient execution attacks? \Rightarrow **Spectre**.
- Why?
 - Rely on tiny CPU details to work \Rightarrow **difficult to simulate**.
 - Hard to study and reproduce ⇒ **simulation could be helping**.
- Comparing a **real system** and a **simulated system**:
 - **Raspberry Pi**, ARM processor.
 - **gem5**, micro-architectural simulator.





- Can we simulate transient execution attacks? \Rightarrow **Spectre**.
- Why?
 - Rely on tiny CPU details to work \Rightarrow **difficult to simulate**.
 - Hard to study and reproduce \Rightarrow simulation could be helping.
- Comparing a **real system** and a **simulated system**:
 - **Raspberry Pi**, ARM processor.
 - **gem5**, micro-architectural simulator.
- Goals:





- Can we simulate transient execution attacks? \Rightarrow **Spectre**.
- Why?
 - Rely on tiny CPU details to work \Rightarrow **difficult to simulate**.
 - Hard to study and reproduce ⇒ **simulation could be helping**.
- Comparing a **real system** and a **simulated system**:
 - **Raspberry Pi**, ARM processor.
 - **gem5**, micro-architectural simulator.
- Goals:
 - Attack that works similarly on both systems.





- Can we simulate transient execution attacks? \Rightarrow **Spectre**.
- Why?
 - Rely on tiny CPU details to work \Rightarrow **difficult to simulate**.
 - Hard to study and reproduce ⇒ **simulation could be helping**.
- Comparing a **real system** and a **simulated system**:
 - **Raspberry Pi**, ARM processor.
 - **gem5**, micro-architectural simulator.
- Goals:
 - Attack that works similarly on both systems.
 - Compare the **faithfulness of the simulation**.



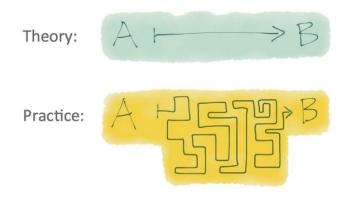


- Can we simulate transient execution attacks? \Rightarrow **Spectre**.
- Why?
 - Rely on tiny CPU details to work \Rightarrow **difficult to simulate**.
 - Hard to study and reproduce ⇒ **simulation could be helping**.
- Comparing a **real system** and a **simulated system**:
 - **Raspberry Pi**, ARM processor.
 - **gem5**, micro-architectural simulator.
- Goals:
 - Attack that works similarly on both systems.
 - Compare the **faithfulness of the simulation**.
 - Discover how gem5 could be helpful.





WHAT REALLY HAPPENED



 Available Spectre implementations failed on our Raspberry Pi ⇒ guidelines and custom implementation.

- Available Spectre implementations failed on our Raspberry Pi ⇒ guidelines and custom implementation.
- 2. Reproducing a micro-architecture is **impossible**.

- Available Spectre implementations failed on our Raspberry Pi ⇒ guidelines and custom implementation.
- 2. Reproducing a micro-architecture is **impossible**.
- 3. gem5 needed some extension to compare it to the real system \Rightarrow **patch**.

• **Guidelines** that are important for micro-architectural security research.

- **Guidelines** that are important for micro-architectural security research.
- Usage of gem5 for helping attack development and understanding.

- **Guidelines** that are important for micro-architectural security research.
- Usage of gem5 for helping attack development and understanding.
- Simulation of Spectre and evaluation of faithfulness.

- **Guidelines** that are important for micro-architectural security research.
- Usage of gem5 for helping attack development and understanding.
- Simulation of Spectre and evaluation of faithfulness.
- **Requirements of gem5** to simulate those attacks.

TABLE OF CONTENT

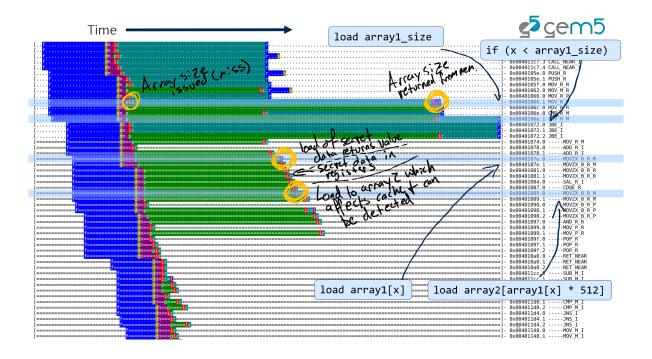
- Related Work
- Spectre Attack
- gem5 Simulator
- Implementation
- Faithfulness
- Conclusion

RELATED WORK

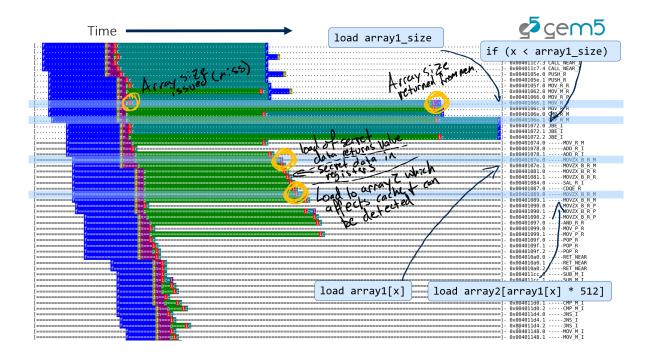
RELATED WORK

• No specific literature about simulation of micro-architectural attack.

J. LOWE-POWER - VISUALIZING SPECTRE WITH GEM5

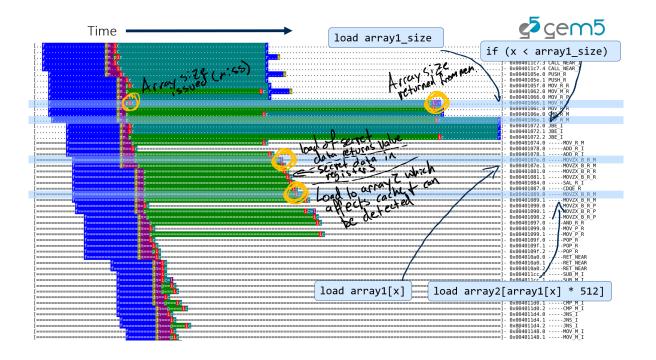


J. LOWE-POWER - VISUALIZING SPECTRE WITH GEM5



• Blog post

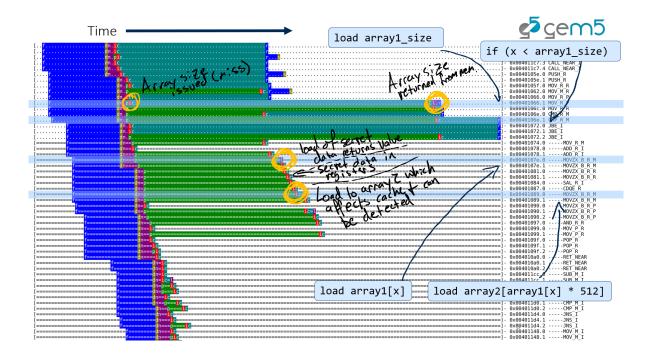
J. LOWE-POWER - VISUALIZING SPECTRE WITH GEM5





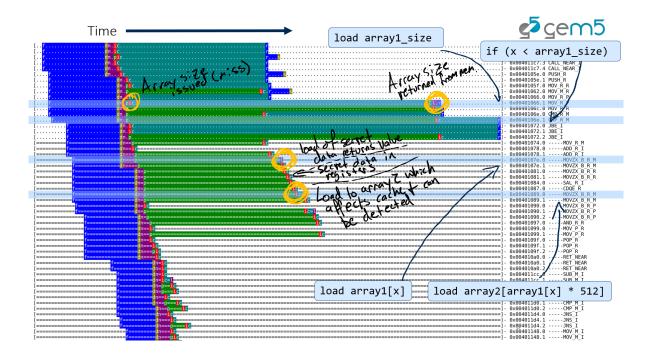
• x86

J. LOWE-POWER - VISUALIZING SPECTRE WITH GEM5



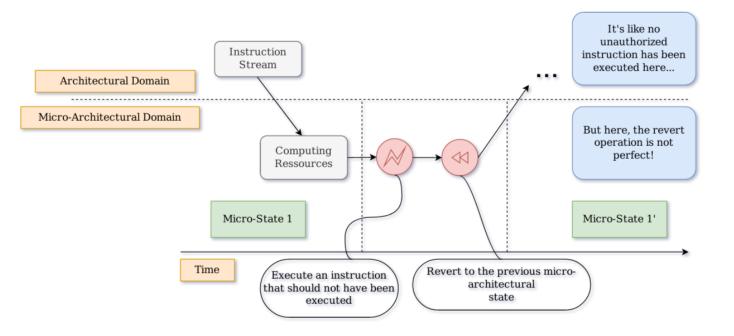
- Blog post
- x86
- Default gem5 configuration

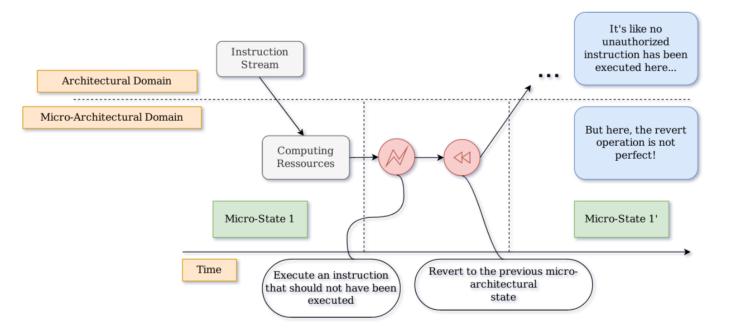
J. LOWE-POWER - VISUALIZING SPECTRE WITH GEM5



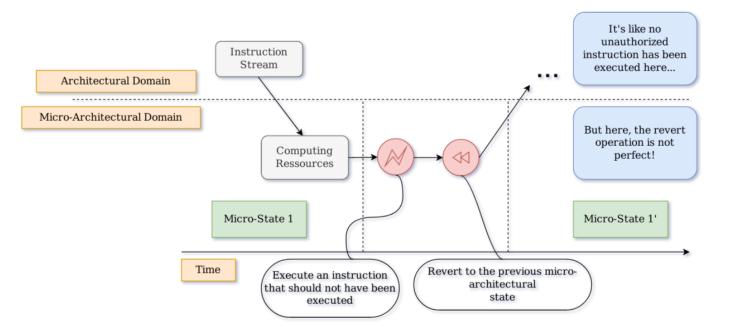
- Blog post
- x86
- Default gem5 configuration
- We wanted to go deeper!

THE SPECTRE ATTACK



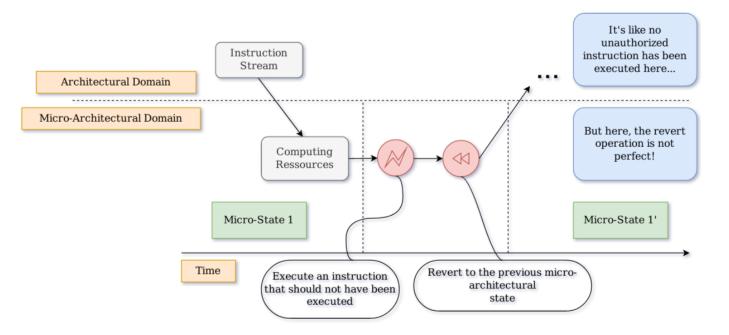


Summary



Summary

• An instruction has been **transiently executed** if it affects the CPU *micro-architectural state* – leaving its architectural state as prior the execution.



Summary

- An instruction has been **transiently executed** if it affects the CPU *micro-architectural state* leaving its architectural state as prior the execution.
- If the the new micro-architectural state depends on a secret and the attacker is able to probe it, he can **recover the secret**.

THE BRANCH PREDICTOR

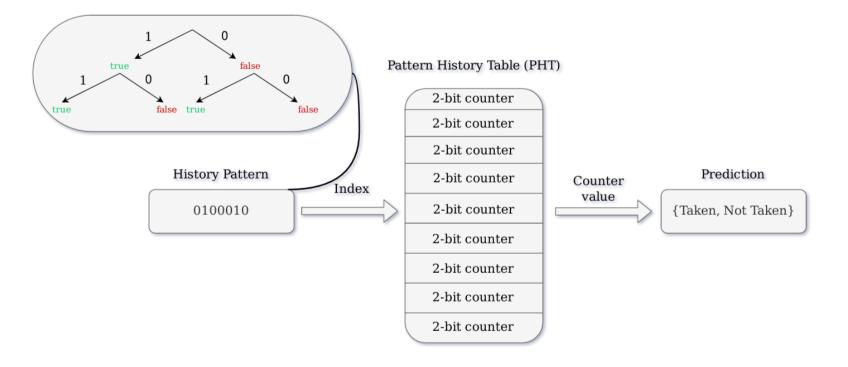
THE BRANCH PREDICTOR

• **Predict instruction flow** when branches are encountered.

THE BRANCH PREDICTOR

- **Predict instruction flow** when branches are encountered.
- Prediction is **dynamic**, it is based on previous execution.

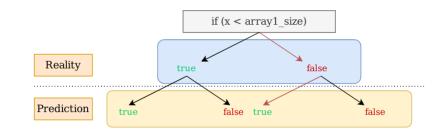
THE PHT, A STRUCTURE USED BY THE BRANCH PREDICTOR



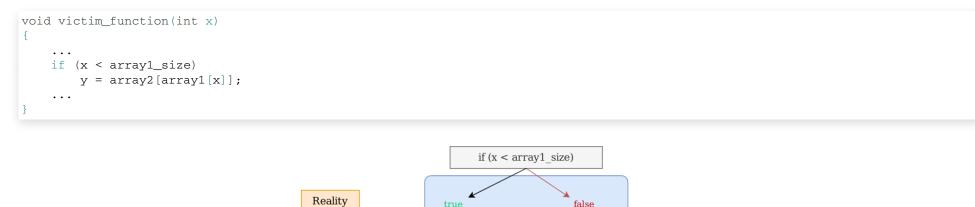
HOW DOES SPECTRE WORK?

WE NEED A TARGET

```
void victim_function(int x)
{
    ...
    if (x < array1_size)
        y = array2[array1[x]];
    ...
}</pre>
```



WE NEED A TARGET



true

true

Prediction



false true

k

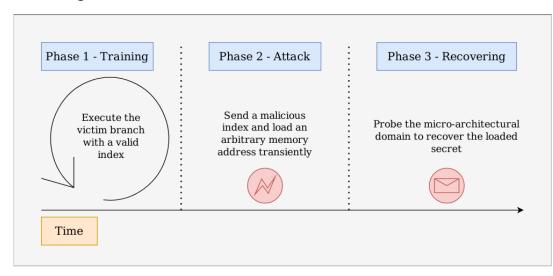
∡

false

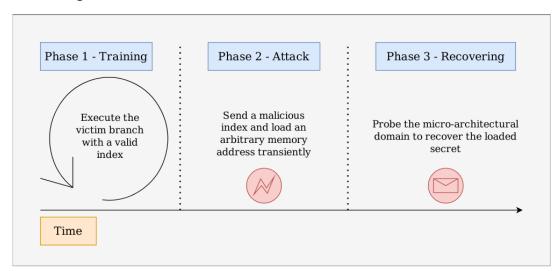
false

Summary

Summary

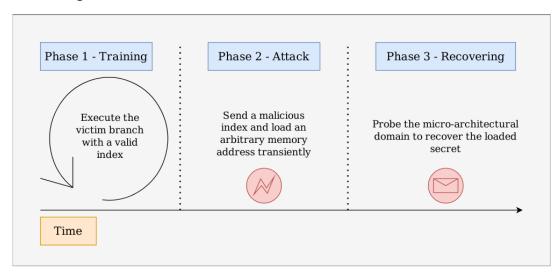


Summary



Details

Summary



Details

Phase 1 - Training 1. Send x < array1_size 2. Repeat until PHT's counters are increased	Phase 2 - Attack 1. Send x such as x = &secret - array1	Phase 3 - Recovering 1. Find the element of array2 that has been cached (array2_cached) 2. Secret is equal to &array2_cached - &array2
Time		·

• Why?

- Why?
 - Hard to reproduce the attack using already existing implementation.

- Why?
 - Hard to reproduce the attack using already existing implementation.
 - Hard to develop a functional attack on a vulnerable processor.

- Why?
 - Hard to reproduce the attack using already existing implementation.
 - Hard to develop a functional attack on a vulnerable processor.
- Refer to the paper for more details.

Compiler version, compiler and manual optimizations

Compiler version, compiler and manual optimizations

Compiler version, compiler and manual optimizations Timer for covert channel

Compiler version, compiler and manual optimizations Timer for covert channel

Compiler version, compiler and manual optimizations Timer for covert channel Prefetcher, re-ordering

Compiler version, compiler and manual optimizations Timer for covert channel Prefetcher, re-ordering

Compiler version, compiler and manual optimizations Timer for covert channel Prefetcher, re-ordering

• Very tricky because we make blind assumptions.

Compiler version, compiler and manual optimizations Timer for covert channel

Prefetcher, re-ordering

- Very tricky because we make blind assumptions.
- gem5 can help resolve this!

Compiler version, compiler and manual optimizations Timer for covert channel Prefetcher, re-ordering

- Very tricky because we make blind assumptions.
- gem5 can help resolve this!

Transient execution window

Compiler version, compiler and manual optimizations Timer for covert channel

Prefetcher, re-ordering

- Very tricky because we make blind assumptions.
- gem5 can help resolve this!

Transient execution window

• Time during which transient executions can happen.

Compiler version, compiler and manual optimizations Timer for covert channel

Prefetcher, re-ordering

- Very tricky because we make blind assumptions.
- gem5 can help resolve this!

Transient execution window

- Time during which transient executions can happen.
- e.g. if (array1_size < 1) vs. if ((float) x / (float) array1_size < 1)

DEVELOPMENT

Compiler version, compiler and manual optimizations Timer for covert channel

Prefetcher, re-ordering

- Very tricky because we make blind assumptions.
- gem5 can help resolve this!

Transient execution window

- Time during which transient executions can happen.
- e.g. if (array1_size < 1) vs. if ((float) x / (float) array1_size < 1)

Our implementation is also full of tips in the comments, feel free to look at it!

• Pinning

- Pinning Page size

- Pinning
- Page size
- Frequency

- Pinning
- Page size
- Frequency
- Mitigations

THE GEM5 SIMULATOR

• Micro-architectural simulator, **cycle-accurate**.

- Micro-architectural simulator, **cycle-accurate**.
- **State-of-the-art** project, started in 2011.

- Micro-architectural simulator, **cycle-accurate**.
- **State-of-the-art** project, started in 2011.

Parts

- Micro-architectural simulator, **cycle-accurate**.
- **State-of-the-art** project, started in 2011.

Parts

- **C++ core** where logic is programmed.
- **Python interface** where systems are built.

- Micro-architectural simulator, **cycle-accurate**.
- **State-of-the-art** project, started in 2011.

Parts

- **C++ core** where logic is programmed.
- **Python interface** where systems are built.

Architecture

- Micro-architectural simulator, **cycle-accurate**.
- **State-of-the-art** project, started in 2011.

Parts

- **C++ core** where logic is programmed.
- **Python interface** where systems are built.

Architecture

Alpha, **ARM**, Power, SPARC, x86, MIPS, RISC-V.

- Micro-architectural simulator, **cycle-accurate**.
- **State-of-the-art** project, started in 2011.

Parts

- **C++ core** where logic is programmed.
- **Python interface** where systems are built.

Architecture

Alpha, **ARM**, Power, SPARC, x86, MIPS, RISC-V. **Generic Micro-Architecture**

- Micro-architectural simulator, **cycle-accurate**.
- **State-of-the-art** project, started in 2011.

Parts

- **C++ core** where logic is programmed.
- **Python interface** where systems are built.

Architecture

Alpha, **ARM**, Power, SPARC, x86, MIPS, RISC-V.

Generic Micro-Architecture

Very simple ones to a 7-stage out-of-order pipelined processor.

- Micro-architectural simulator, **cycle-accurate**.
- **State-of-the-art** project, started in 2011.

Parts

- **C++ core** where logic is programmed.
- **Python interface** where systems are built.

Architecture

Alpha, **ARM**, Power, SPARC, x86, MIPS, RISC-V.

Generic Micro-Architecture

Very simple ones to a 7-stage out-of-order pipelined processor. **Branch Prediction**

- Micro-architectural simulator, **cycle-accurate**.
- **State-of-the-art** project, started in 2011.

Parts

- **C++ core** where logic is programmed.
- **Python interface** where systems are built.

Architecture

Alpha, **ARM**, Power, SPARC, x86, MIPS, RISC-V.

Generic Micro-Architecture

Very simple ones to a 7-stage out-of-order pipelined processor.

Branch Prediction

Bi-Mode, TAGE, Two-Level, Perceptron, Tournament...

HOW TO USE IT?

Configuring parameters of a cache memory

Instantiating some CPU components

Connecting components together

Configuring parameters of a cache memory

size = '32kB'
assoc = 2
data_latency = 1
mshrs = 4
tgts_per_mshr = 8
write_buffers = 4
prefetcher = StridePrefetcher(queue_size=4, degree=4)

Instantiating some CPU components

Connecting components together

Configuring parameters of a cache memory

size = '32kB'
assoc = 2
data_latency = 1
mshrs = 4
tgts_per_mshr = 8
write_buffers = 4
prefetcher = StridePrefetcher(queue_size=4, degree=4)

Instantiating some CPU components

```
for cpu in self.cpus:
    cpu.createThreads()
    cpu.createInterruptController()
    cpu.branchPredAdd()
if system.getMemoryMode() == "timing":
    self.cacheAddL1()
    self.cacheAddL2()
```

Connecting components together

Configuring parameters of a cache memory

size = '32kB'
assoc = 2
data_latency = 1
mshrs = 4
tgts_per_mshr = 8
write_buffers = 4
prefetcher = StridePrefetcher(gueue_size=4, degree=4)

Instantiating some CPU components

```
for cpu in self.cpus:
    cpu.createThreads()
    cpu.createInterruptController()
    cpu.branchPredAdd()
if system.getMemoryMode() == "timing":
    self.cacheAddL1()
    self.cacheAddL2()
```

Connecting components together

cpu.dtb.walker.port = bus.slave cpu.itb.walker.port = bus.slave cpu.dcache_port = bus.slave cpu.icache_port = bus.slave

Configuring parameters of a cache memory

size = '32kB'
assoc = 2
data_latency = 1
mshrs = 4
tgts_per_mshr = 8
write_buffers = 4
prefetcher = StridePrefetcher(gueue_size=4, degree=4)

Instantiating some CPU components

for cpu in self.cpus: cpu.createThreads() cpu.createInterruptController() cpu.branchPredAdd() if system.getMemoryMode() == "timing": self.cacheAddL1() self.cacheAddL2()

Connecting components together

cpu.dtb.walker.port = bus.slave cpu.itb.walker.port = bus.slave cpu.dcache_port = bus.slave cpu.icache_port = bus.slave

Passing arguments to the Linux kernel

kernel_cmd = [
"console=ttyAMA0"	,
"root=/dev/vda1",	
"rw",	

"mem=2G@0x8000000",

1

Launching a full-system simulation

Connecting to the simulation terminal

Kernel booting up

Launching a full-system simulation

./build/ARM/gem5.opt ./configs/example/arm/starter_fs.py --num-cores=4 --disk-image="aarch64-ubuntu.img" --kernel="vmlinux.arm64"

Connecting to the simulation terminal

Kernel booting up

Launching a full-system simulation

./build/ARM/gem5.opt ./configs/example/arm/starter_fs.py --num-cores=4 --disk-image="aarch64-ubuntu.img" --kernel="vmlinux.arm64"

Connecting to the simulation terminal

gem5/util/term/m5term localhost 3456

Kernel booting up

Launching a full-system simulation

./build/ARM/gem5.opt ./configs/example/arm/starter_fs.py --num-cores=4 --disk-image="aarch64-ubuntu.img" --kernel="vmlinux.arm64"

Connecting to the simulation terminal

gem5/util/term/m5term localhost 3456

Kernel booting up

==== m5 slave terminal: Terminal 0 ====

- [0.000000] Booting Linux on physical CPU 0x000000000 [0x410fd070]
- [0.000000] Linux version 4.18.0+ (arm-employee@arm-computer) (gcc version 7.4.0 (Ubuntu/Linaro 7.4.0-lubuntu1~18.04.1))
- [0.000000] Machine model: V2P-CA15
- [0.000000] Memory limited to 2048MB

• • •

Launching a full-system simulation

./build/ARM/gem5.opt ./configs/example/arm/starter_fs.py --num-cores=4 --disk-image="aarch64-ubuntu.img" --kernel="vmlinux.arm64"

Connecting to the simulation terminal

gem5/util/term/m5term localhost 3456

Kernel booting up

==== m5 slave terminal: Terminal 0 ====

[0.000000] Booting Linux on physical CPU 0x0000000000 [0x410fd070]

[0.000000] Linux version 4.18.0+ (arm-employee@arm-computer) (gcc version 7.4.0 (Ubuntu/Linaro 7.4.0-lubuntu1~18.04.1))

[0.000000] Machine model: V2P-CA15

[0.000000] Memory limited to 2048MB

• • •

. . .

Opening a shell on the simulated system

[0.256634] random: init: uninitialized urandom read (12 bytes read)

[0.271877] init: hwclock main process (684) terminated with status 1

[0.286689] random: mountall: uninitialized urandom read (12 bytes read)

Ubuntu 14.04 LTS aarch64-gem5 ttyAMA0

aarch64-gem5 login: root

Welcome to Ubuntu 14.04 LTS (GNU/Linux 4.18.0+ aarch64)

root@aarch64-gem5:~#

BENEFITS OF (PIPELINE) VISUALIZATION

BENEFITS OF (PIPELINE) VISUALIZATION

• gem5 \Rightarrow output the state of any element in the system.

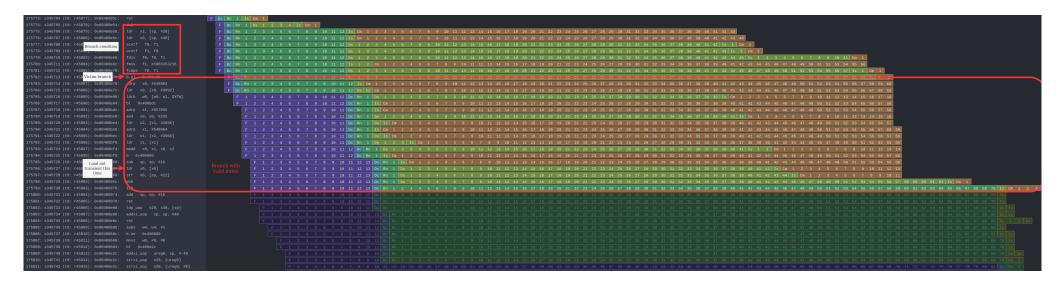
BENEFITS OF (PIPELINE) VISUALIZATION

- gem5 \Rightarrow output the state of any element in the system.
- Konata ⇒ graphically visualize the pipeline of a simulated processor.

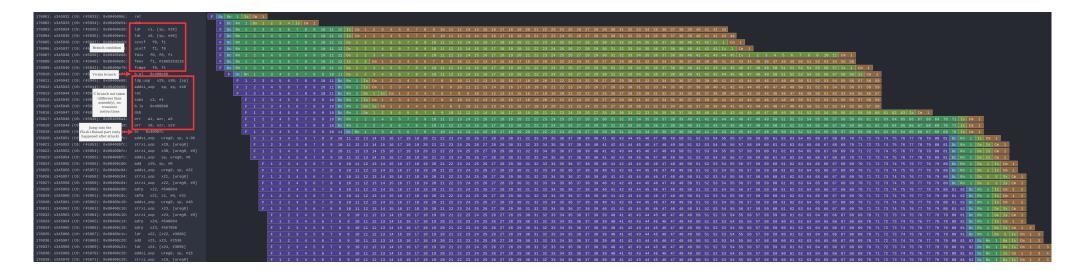
TRANSIENT EXECUTION OF A READ INSTRUCTION WITH A MALICIOUS INDEX

12769: s182700 (t0: r3392): 0x0041b430: b.ne 0x41b500	
12770: s182701 (t0: r3393): 0x0041b500: subs x2, #16	
12771: s182702 (t0: r3394): 0x0041b504: b.cs 0x41b528	
12772: s182703 (t0: r3349): 0x0040095c: ret	F DC Rn 1 IS Cm 1
12773: s182704 (t0: r3350): 0x00400e54:dsh	F DC Rn 1 DS 1 2 3 4 IS Cn 1
12774: s182705 (t0: r3351): 0x00400e58: ldr x1, [sp, #28]	F Dc Rm 1 2 3 4 5 6 7 8 9 10 11 12 Is Cm 1 2 3 4 5 6 7 8 9 10 11 12 Is Cm 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
12775: s182706 (t0: r3352): 0x00400e5c: ldr x0, [sp, #40]	F DC Rn 1 2 3 4 5 6 7 8 9 10 11 12 Is Cn 1 2 3 4 5 6 7 8 9 10 11 12 Is Cn 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
12776: s182707 (t0: r3353): 0x00400e60: scvtf f0, f1	F DC Rn 1 2 3 4 5 6 7 8 9 10 11 12 Ds 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 Is 1 Cm 1
12777: s182708 (t0: r3: Branch condition : ucvtf f1, f0	F DC Rn 1 2 3 4 5 6 7 8 9 10 11 12 Ds 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 Is 1 Ca 1
12778: s182709 (t0: r3 333). oxoo+ooeco : fdiv f0, f0, f1	F DC Rn 1 2 3 4 5 6 7 8 9 10 11 12 Ds 1 2 3 4 5 6 7 8 9 10 11 12 Ds 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 Is 1 2 3 4 5 6 7 8 9 10 11 Cm 1
12779: s182710 (t0: r3356): 0x00400e6c: fmov f1, #1065353216	F DC Rn 1 2 3 4 5 6 7 8 9 10 11 12 Is 1 2 Cm 1 2 3 4 5 6 7 8 9 10 11 12 Is 1 2 Cm 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56
12780: s182711 (t0: r33 <u>57): 0x004800</u> e70: fcmpe f0, f1	F DC Rn 1 2 3 4 5 6 7 8 9 10 11 12 Ds 1 2 3 4 5 6 7 8 9 10 11 12 Ds 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 Is 1 Cm 1 2 3
12781: s182712 (t0: r3: Victim branch 24 D.pl 0x400e88	F DC Rn 1 2 3 4 5 6 7 8 9 10 11 Ds 1 2 3 4 5 6 7 8 9 10 11 Ds 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 Is cn 1 2
12782: s182713 (t0: r3359): 0x00400e78: adrp x0, #548864	F DC Rn 1 2 3 4 5 6 7 8 9 10 11 12 Is
12783: s182714 (t0: r3360): 0x00400e7c: ldr x0, [x0, #3992]	F DC RN 1 2 3 4 5 6 7 8 9 10 11 12 DS IS
12784: s182715 (t0: r3361): 0x00480e80: ldrb w0, [w0, w1, SXTW]	F 1 2 3 4 5 6 7 8 9 10 11 12 Dc Rn 1 Ds 1 2 3 4 5 6 7 8 9 10 11 12 13 14 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 15
12785: s182716 (t0: r3362): 0x00400e84: bl 0x400bdc	F 1 2 3 4 5 6 7 8 9 10 11 12 DC Rn 1 13
12786: s182717 (t0: r3363): 0x00400bdc: adrp x1, #557056	F 1 2 3 4 5 6 7 8 9 10 11 DC Rn 1 Ta
12787: s182718 (t0: r3364): 0x00400be0: and x0, x0, #255	F 1 2 3 4 5 6 7 8 9 10 11 0c Rn 1 0c 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 15
12788: s182719 (t0: r3365): 0x00400be4: ldr x2, [x1, #2656]	F 1 2 3 4 5 6 7 8 9 10 11 DC Rn 1 Dd IS
12789: s182720 (t0: r3366): 0x00400be8: adrp x1, #548864	F 1 2 3 4 5 6 7 8 9 10 11 DC Rn 1 T3
12790: s182721 (t0: r3367): 0x00400bec: ldr x1, [x1, #3968]	F 1 2 3 4 5 6 7 8 9 10 11 Dc Rn 1 05 15
12791: s182722 (t0: r3368): 0x00400bf0: ldr x1, [x1]	F 1 2 3 4 5 6 7 8 9 10 11 DG Rn 1 DS 1 2 3 15
12792: s182723 (t0: r3369): 0x00400bf4: madd x0, x1, x0, x2	F 1 2 3 4 5 6 7 8 9 10 11 12 Dc Rn 1 Ds 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 Is 1 2
12793: s182724 (t0: r3370): 0x00400bf8: b 0x400960	F 1 2 3 4 5 6 7 8 9 10 11 12 00 Rn 1 18
12794: s182725 (t0: r33 [°] _{Transient load} ³⁰ : sub sp, sp, #16	F 1 2 3 4 5 6 7 8 9 10 11 12 13 00 Rn 1 15
12795: s182726 (t0: r33' that leak the	F 1 2 3 4 5 6 7 8 9 10 11 12 13 00 Rm 1 Ds 1 2 3 4 5 6 7 8 9 10 11 12 13 14 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 15
12796: s182727 (t0: r33 s8:	F 1 2 3 4 5 6 7 8 9 10 11 12 13 00 Rm 1
12797: s182728 (t0: r3374): 0x0040096c: dsb	F 1 2 3 4 5 6 7 8 9 10 11 12 13 DC Bn 1
12798: s182729 (t0: r3375): 0x00400970: isb	<u>F 1 2 3 4 5 6 7 8 9 10 11 12 13 02</u>
12799: s182730 (t0: r3376): 0x00400974: add sp, sp, #16	
12800: s182731 (t0: r3377): 0x00400978: ret	
12801: s182732 (t0: r3378): 0x00400e88: ldp_uop x29, x30, [sp]	<u>F 1 2 3 4 5 6 7 8 9 19 11 12 02</u>
12802: s182733 (t0: r3379): 0x00400e88: addxi_uop sp, sp, #48	F 1 2 3 4 5 6 7 8 9 19 11 12 13 DE
12803: s182734 (t0: r3380): 0x00400e8c: ret	F 1 2 3 4 5 6 7 8 9 19 11 12 13 BE
12804: s182735 (t0: r3381): 0x00400698: subs x3, #4	F 1 2 3 4 5 6 7 8 9 10 11 12 BC F 1 2 3 4 5 6 7 8 9 10 11 12 BC F 1 2 3 4 5 7 8 9 10 11 12 BC
12805: s182736 (t0: r3382): 0x0040069c: b.ls 0x4006b0	
12806: s182737 (t0: r3383): 0x004006b0: orr x1, xzr, x24	<u>F 1 2 3 4 5 6 7 8 9 10 11 BC</u>
12807: s182738 (t0: r3384): 0x004006b4: orr x0, xzr, x26	

BRANCH PREDICTOR BEING TRAINED



SPECTRE DEFEATED BY THE BRANCH PREDICTOR



IMPLEMENTING...

• IAIK implementation **failed to perform** the attack **successfully**.

- IAIK implementation **failed to perform** the attack **successfully**.
- Needed an **implementation** with the following requirements:

- IAIK implementation **failed to perform** the attack **successfully**.
- Needed an **implementation** with the following requirements:
 - 1. Stable results,

- IAIK implementation **failed to perform** the attack **successfully**.
- Needed an **implementation** with the following requirements:
 - 1. Stable results,
 - 2. Follows our **guidelines**,

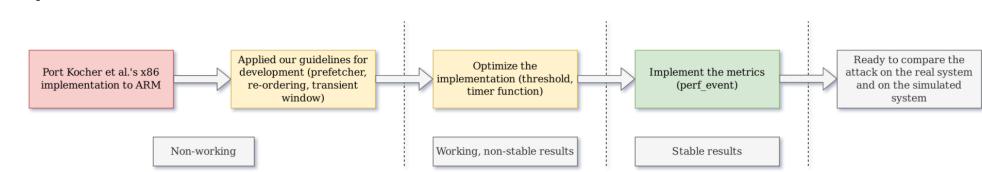
- IAIK implementation **failed to perform** the attack **successfully**.
- Needed an **implementation** with the following requirements:
 - 1. Stable results,
 - 2. Follows our **guidelines**,
 - 3. Usable both on the **Raspberry Pi** and on **gem5**,

- IAIK implementation **failed to perform** the attack **successfully**.
- Needed an **implementation** with the following requirements:
 - 1. Stable results,
 - 2. Follows our **guidelines**,
 - 3. Usable both on the **Raspberry Pi** and on **gem5**,
 - 4. Metrics output.

- IAIK implementation **failed to perform** the attack **successfully**.
- Needed an **implementation** with the following requirements:
 - 1. Stable results,
 - 2. Follows our **guidelines**,
 - 3. Usable both on the **Raspberry Pi** and on **gem5**,
 - 4. Metrics output.

Steps

- IAIK implementation **failed to perform** the attack **successfully**.
- Needed an **implementation** with the following requirements:
 - 1. Stable results,
 - 2. Follows our guidelines,
 - 3. Usable both on the **Raspberry Pi** and on **gem5**,
 - 4. Metrics output.



Steps

• Steps:

- Steps:
 - 1. Syscall emulation system

- Steps:
 - 1. Syscall emulation system
 - 2. Caches

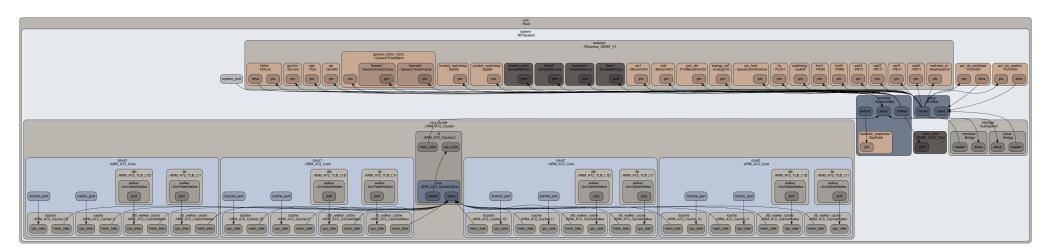
- Steps:
 - 1. Syscall emulation system
 - 2. Caches
 - 3. Branch predictor \Rightarrow Spectre working

- Steps:
 - 1. Syscall emulation system
 - 2. Caches
 - 3. Branch predictor \Rightarrow Spectre working
 - 4. Full-system simulation

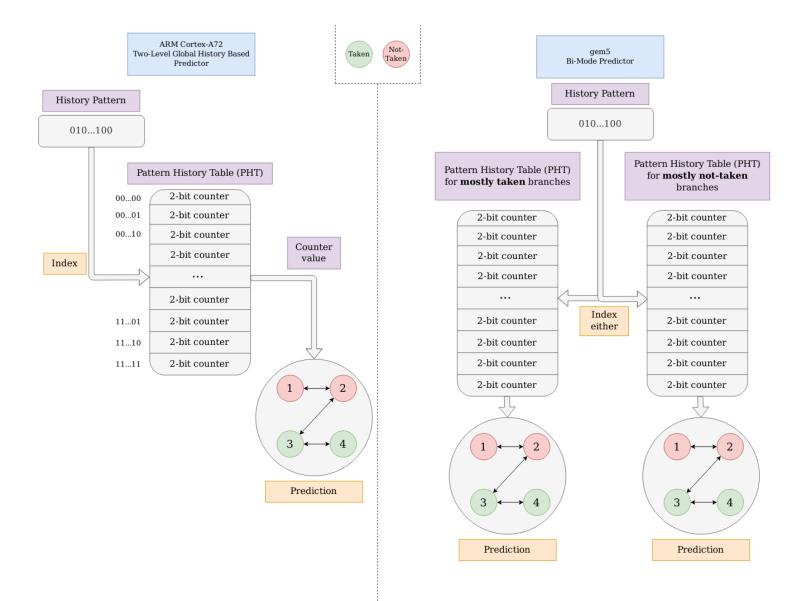
- Steps:
 - 1. Syscall emulation system
 - 2. Caches
 - 3. Branch predictor \Rightarrow Spectre working
 - 4. Full-system simulation
 - 5. Patch for perf_event \Rightarrow measurements working

- Steps:
 - 1. Syscall emulation system
 - 2. Caches
 - 3. Branch predictor \Rightarrow Spectre working
 - 4. Full-system simulation
 - 5. Patch for perf_event \Rightarrow measurements working
- Getting **closer** of the **ARM Cortex-A72** of the Raspberry Pi.

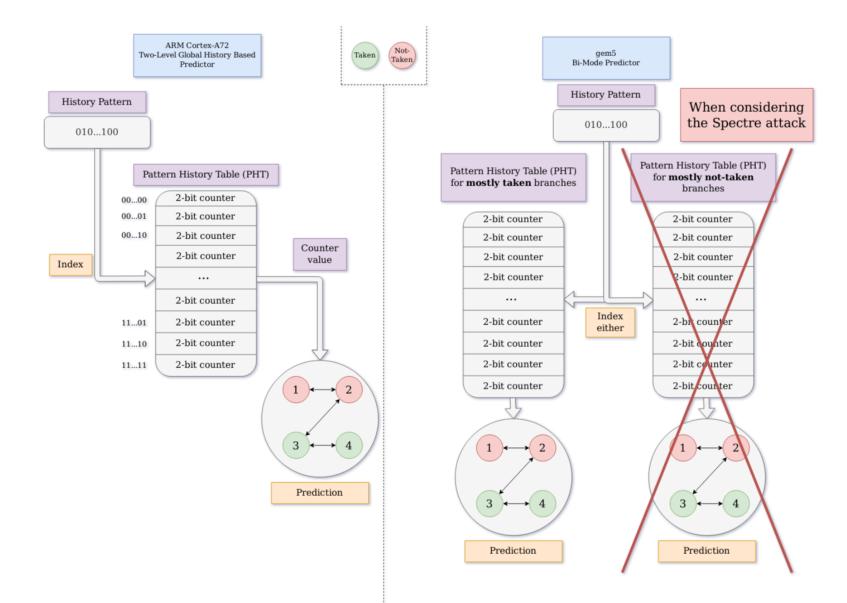
DETAILS OF THE GEM5 SYSTEM



CONFIGURATION OF THE BRANCH PREDICTOR



CONSIDERING SPECTRE, BOTH PREDICTORS ARE EQUIVALENT



• A simulation **not faithful** will not be so **useful**...

- A simulation **not faithful** will not be so **useful**...
- Measurements of **metrics**:

- A simulation **not faithful** will not be so **useful**...
- Measurements of **metrics**:
 - 1. Retrieved bytes: Similar

- A simulation **not faithful** will not be so **useful**...
- Measurements of **metrics**:
 - 1. Retrieved bytes: Similar
 - 2. Iterations: Two times easier on gem5

- A simulation **not faithful** will not be so **useful**...
- Measurements of **metrics**:
 - 1. Retrieved bytes: Similar
 - 2. Iterations: Two times easier on gem5
 - 3. **Cycles**: Three time faster on gem5

- A simulation **not faithful** will not be so **useful**...
- Measurements of **metrics**:
 - 1. Retrieved bytes: Similar
 - 2. Iterations: Two times easier on gem5
 - 3. **Cycles**: Three time faster on gem5
 - 4. Cache misses: Aberrant result

- A simulation **not faithful** will not be so **useful**...
- Measurements of **metrics**:
 - 1. Retrieved bytes: Similar
 - 2. Iterations: Two times easier on gem5
 - 3. **Cycles**: Three time faster on gem5
 - 4. Cache misses: Aberrant result
 - 5. Mispredicted branches: Similar

• If simulation becomes widely used:

- If simulation becomes widely used:
 - Easier to reproduce older attacks for understanding and experimentation.

- If simulation becomes widely used:
 - Easier to reproduce older attacks for understanding and experimentation.
 - With faithful models, researchers could use the simulator itself to discover new vulnerabilities.

- If simulation becomes widely used:
 - Easier to reproduce older attacks for understanding and experimentation.
 - With faithful models, researchers could use the simulator itself to discover new vulnerabilities.
- But...

- If simulation becomes widely used:
 - Easier to reproduce older attacks for understanding and experimentation.
 - With faithful models, researchers could use the simulator itself to discover new vulnerabilities.
- But...
 - Simulation is currently slow.

- If simulation becomes widely used:
 - Easier to reproduce older attacks for understanding and experimentation.
 - With faithful models, researchers could use the simulator itself to discover new vulnerabilities.
- But...
 - Simulation is currently slow.
 - Simulator still needs improvements and extensions.

- If simulation becomes widely used:
 - Easier to reproduce older attacks for understanding and experimentation.
 - With faithful models, researchers could use the simulator itself to discover new vulnerabilities.
- But...
 - Simulation is currently slow.
 - Simulator still needs improvements and extensions.
- In summary:

- If simulation becomes widely used:
 - Easier to reproduce older attacks for understanding and experimentation.
 - With faithful models, researchers could use the simulator itself to discover new vulnerabilities.
- But...
 - Simulation is currently slow.
 - Simulator still needs improvements and extensions.
- In summary:
 - Possible to simulate micro-architectural attacks and being accurate.

- If simulation becomes widely used:
 - Easier to reproduce older attacks for understanding and experimentation.
 - With faithful models, researchers could use the simulator itself to discover new vulnerabilities.
- But...
 - Simulation is currently slow.
 - Simulator still needs improvements and extensions.
- In summary:
 - Possible to simulate micro-architectural attacks and being accurate.
 - Visualization is a very powerful technique to understand the micro-architectural behavior.

WEBSITE

https://pierreay.github.io/reproduce-spectre-gem5/

QUESTIONS?

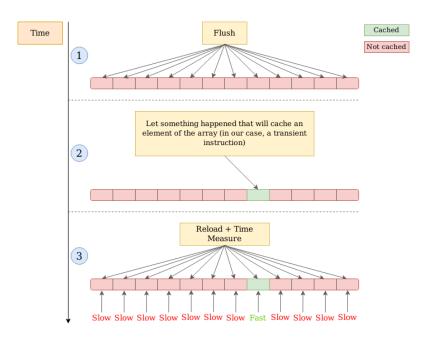
pierre.ayoub@eurecom.fr

APPENDICES

THE MICROARCHITECTURAL DOMAIN

Flush+Reload

Often used to probe the cache state.



Syscall Emulation

Syscall Emulation

gem5 plays the role of the operating system, as it emulates every system calls of a binary over the simulated hardware.

Syscall Emulation

gem5 plays the role of the operating system, as it emulates every system calls of a binary over the simulated hardware.

Full-System Simulation

Syscall Emulation

gem5 plays the role of the operating system, as it emulates every system calls of a binary over the simulated hardware.

Full-System Simulation

gem5 runs an entire operating system over the simulated hardware.

Syscall Emulation

gem5 plays the role of the operating system, as it emulates every system calls of a binary over the simulated hardware.

Full-System Simulation

gem5 runs an entire operating system over the simulated hardware.

Baremetal

Syscall Emulation

gem5 plays the role of the operating system, as it emulates every system calls of a binary over the simulated hardware.

Full-System Simulation

gem5 runs an entire operating system over the simulated hardware.

Baremetal

gem5 runs native assembly code over the simulated hardware, without any operating system layer.

RESULTS

Table 1: Ratio between gem5 and Raspberry Pi runs for each metric. A value below 1 means that gem5's metric is lower than the Raspberry Pi's metric.

	Accuracy Ratio	Accuracy Ratio
Metric	Mean	Standard Deviation
Retrieved Bytes	1.05	NaN
Iterations	0.57	3.81
Cycles	0.31	2.12
Cache Misses	584.08	4581.02
Mispredicted Branches	0.99	2.41