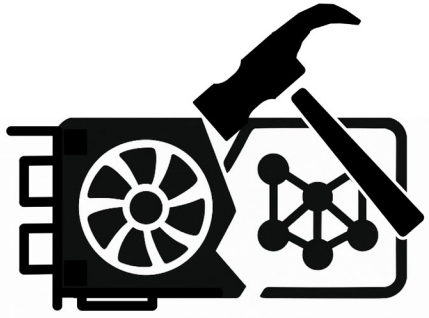


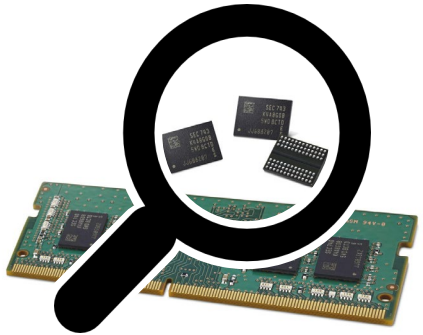
GPUHammer: Rowhammer Attacks on GPU Memories are Practical

Chris S. Lin†, **Joyce Qu**†, Gururaj Saileshwar

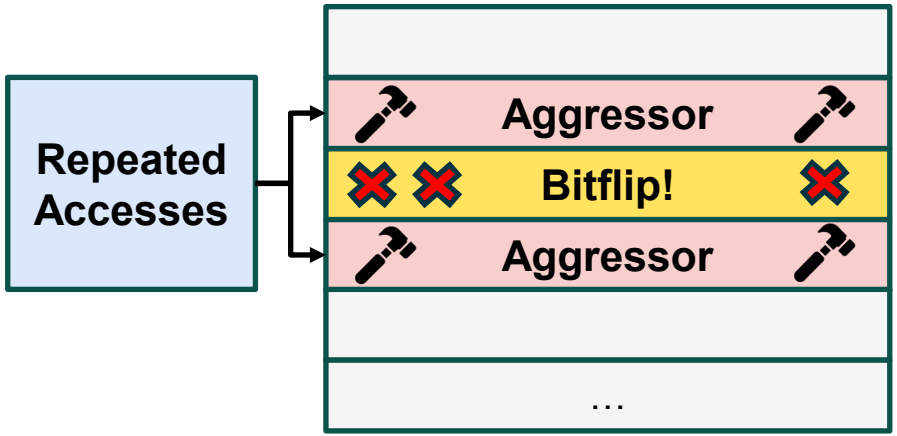
UNIVERSITY OF TORONTO

†Equal contribution

The Rowhammer Vulnerability[1]



Inside these memory chips …



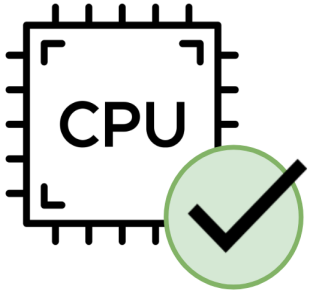
Rapid Accesses to a DRAM Row Can Flip Bits in Neighbouring Rows.

UNIVERSITY OF TORONTO

[1]Flipping bits in memory without accessing them: an experimental study of DRAM disturbance errors (*ISCA '14)*

CPU Rowhammer Exploits

Exploits Targeting CPU-based DRAMs are Well-Studied

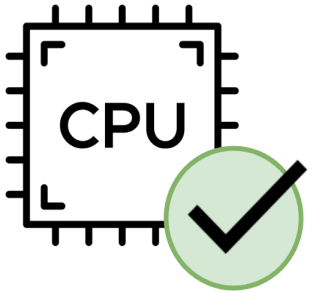


DDR3-5 **LPDDR3&4X**

UNIVERSITY OF TORONTO

CPU Rowhammer Exploits

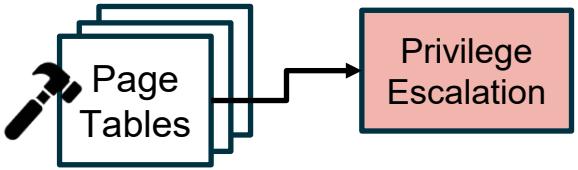
Exploits Targeting CPU-based DRAMs are Well-Studied



DDR3-5 LPDDR3&4X

Instruction Modification[1]: `jmp rax` ⇨ `jmp rcx`

Page Table Tampering[1]:

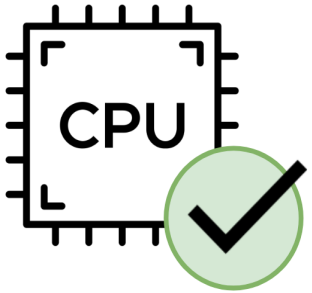


UNIVERSITY OF TORONTO

[1]Exploiting the DRAM rowhammer bug to gain kernel privileges (Google Project Zero 2015)

CPU Rowhammer Exploits

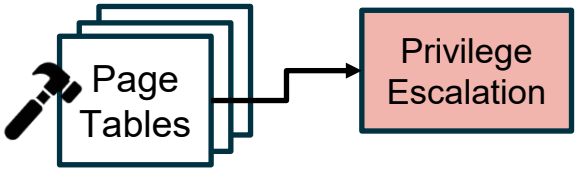
Exploits Targeting CPU-based DRAMs are Well-Studied



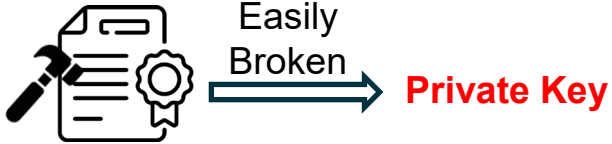
DDR3-5 LPDDR3&4X

Instruction Modification[1]: `jmp rax` ⇨ `jmp rcx`

Page Table Tampering[1]:



Break Crypto-Protocols[2]:



Weakened Pub Key

UNIVERSITY OF TORONTO

[1]Exploiting the DRAM rowhammer bug to gain kernel privileges (Google Project Zero 2015)

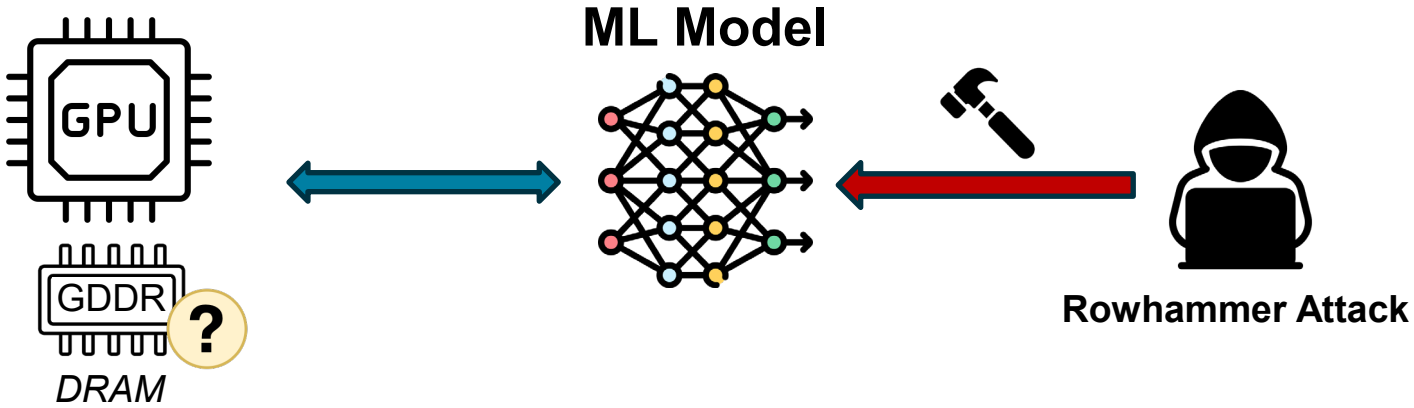
[2]Flip Feng Shui: Hammering a Needle in the Software Stack (SEC '16)

The Rowhammer Attack

Who Else Uses DRAM?

The Rowhammer Attack

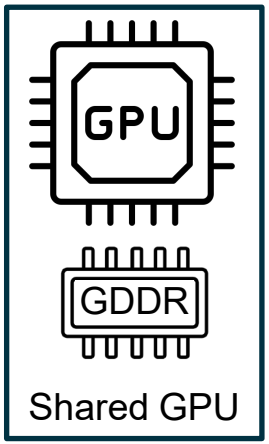
Who Else Uses DRAM? **GPUs!**



GPUs Hold Critical Data. Are they Secure Against Rowhammer Attacks?

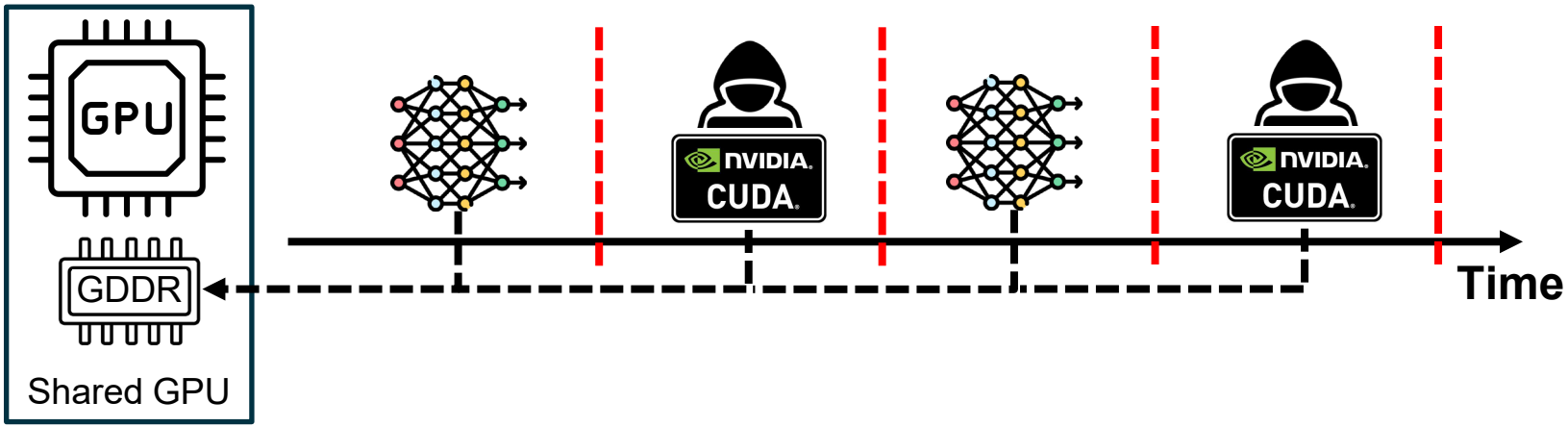
Threat Model for GPUs

Time-Sliced Setting for Serverless GPUs



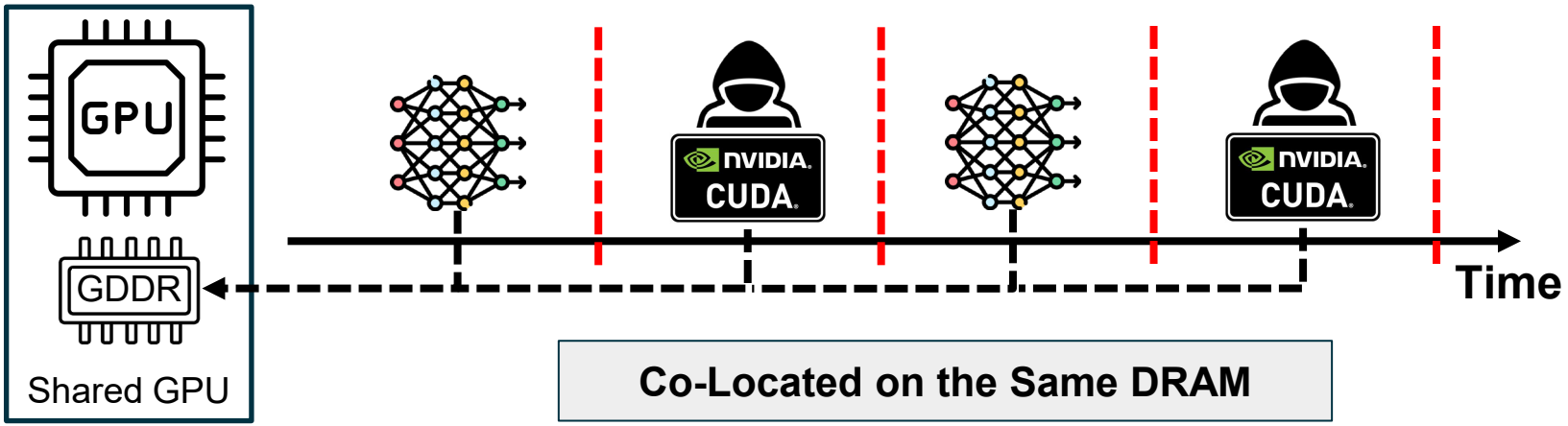
Threat Model for GPUs

Time-Sliced Setting for Serverless GPUs



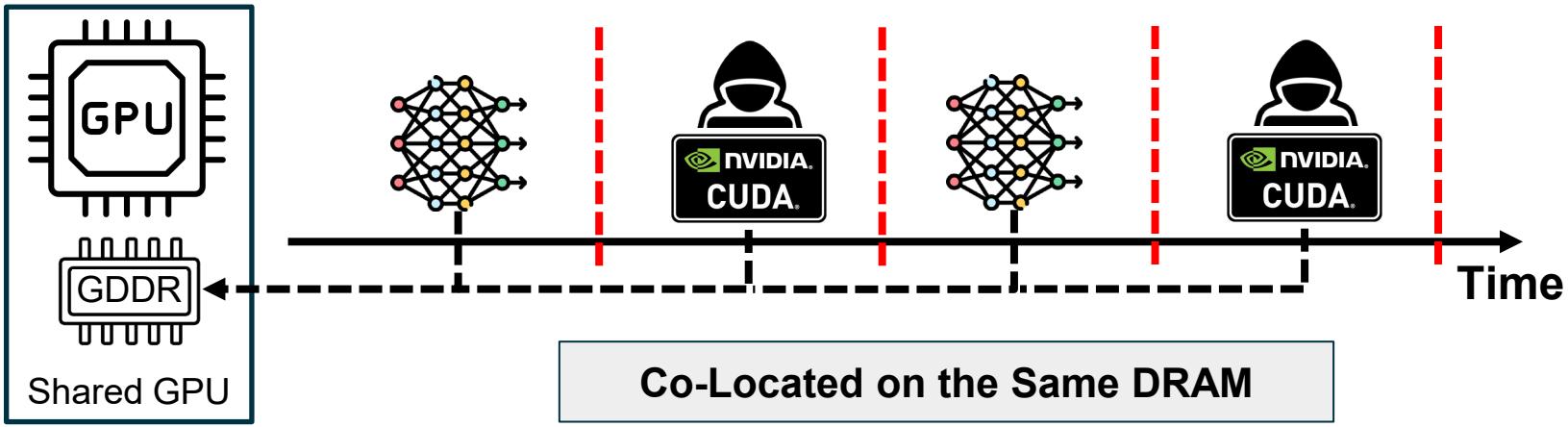
Threat Model for GPUs

Time-Sliced Setting for Serverless GPUs



Threat Model for GPUs

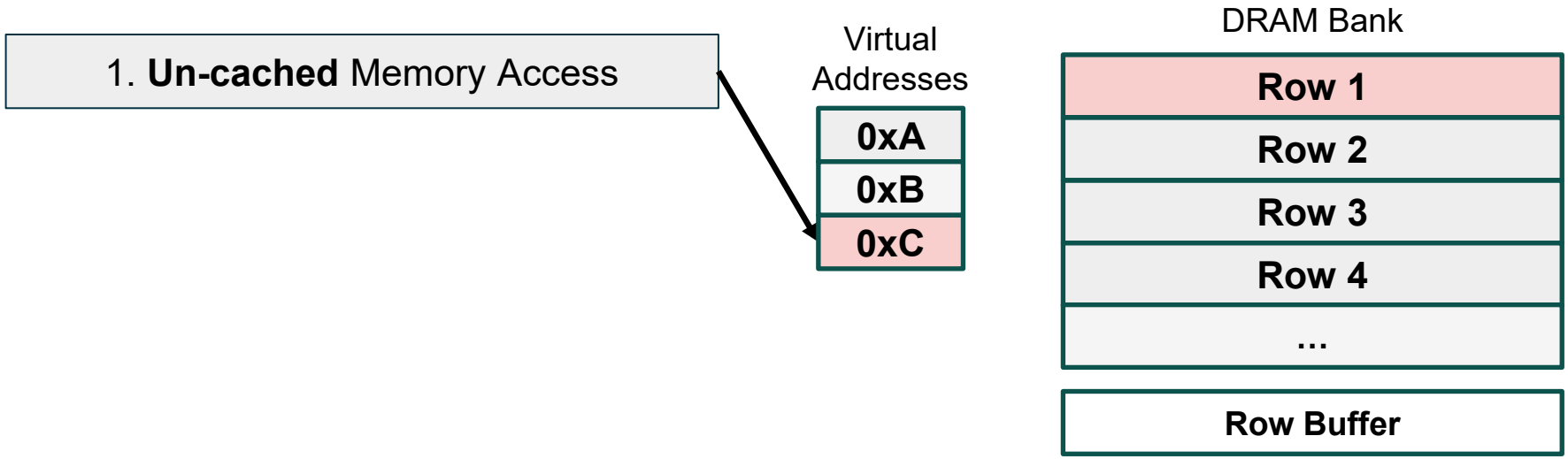
Time-Sliced Setting for Serverless GPUs



How to Launch Rowhammer Attacks on GPUs?

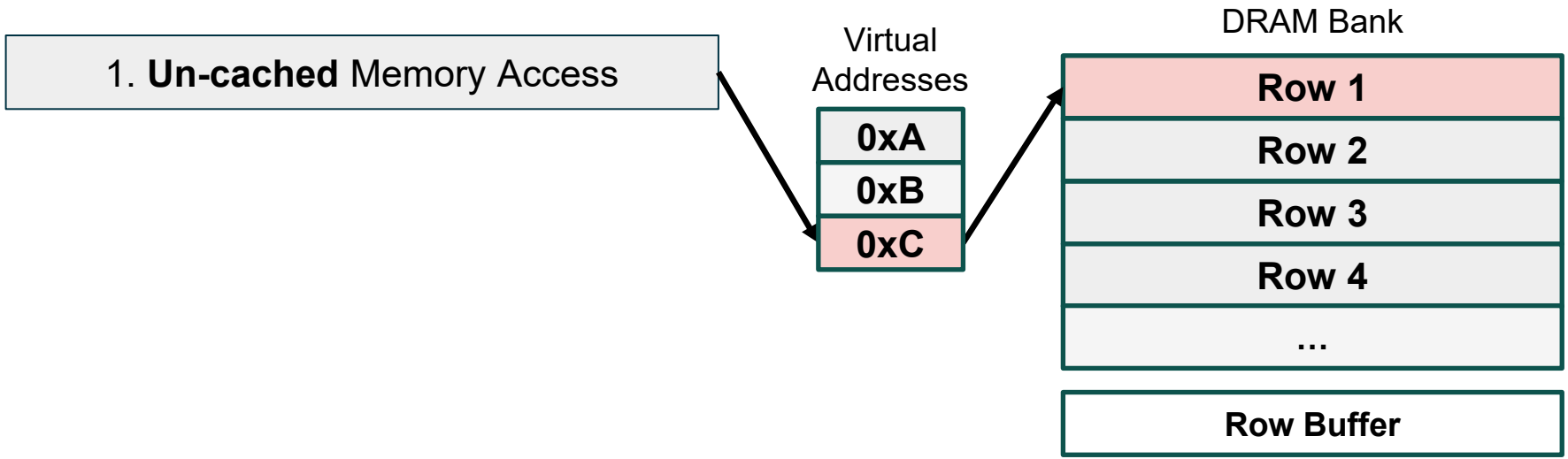
Basic Rowhammer Attack

Launching Naïve Rowhammer Attack on GPUs



Basic Rowhammer Attack

Launching Naïve Rowhammer Attack on GPUs

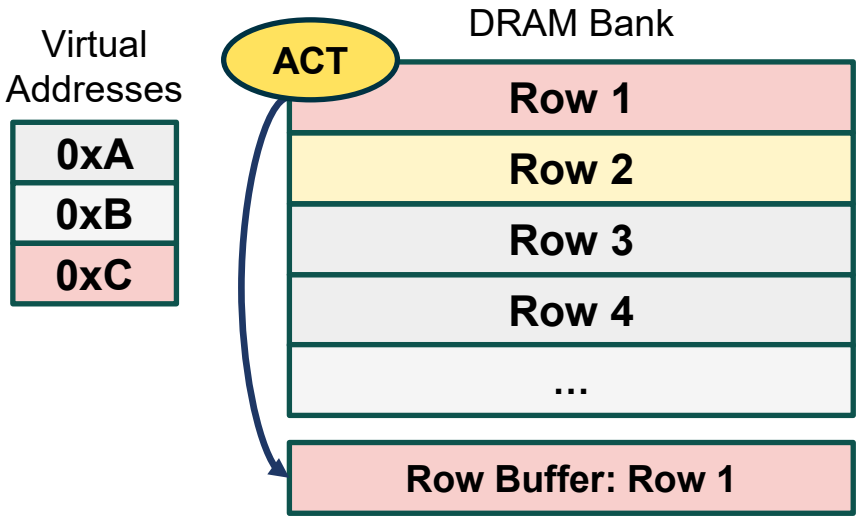


Basic Rowhammer Attack

Launching Naïve Rowhammer Attack on GPUs

1. **Un-cached** Memory Access

2. Row Activated **(ACT)**, Bring to Buffer

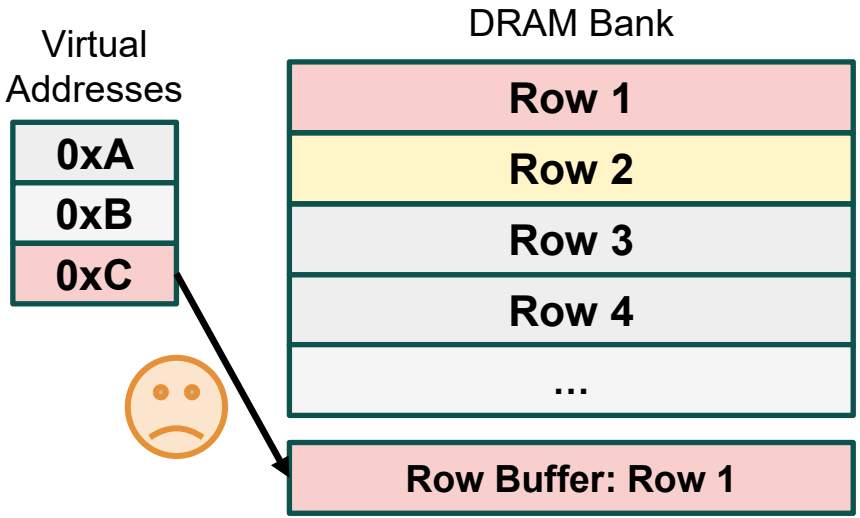


Basic Rowhammer Attack

Launching Naïve Rowhammer Attack on GPUs

1. **Un-cached** Memory Access

2. Row Activated **(ACT)**, Bring to Buffer



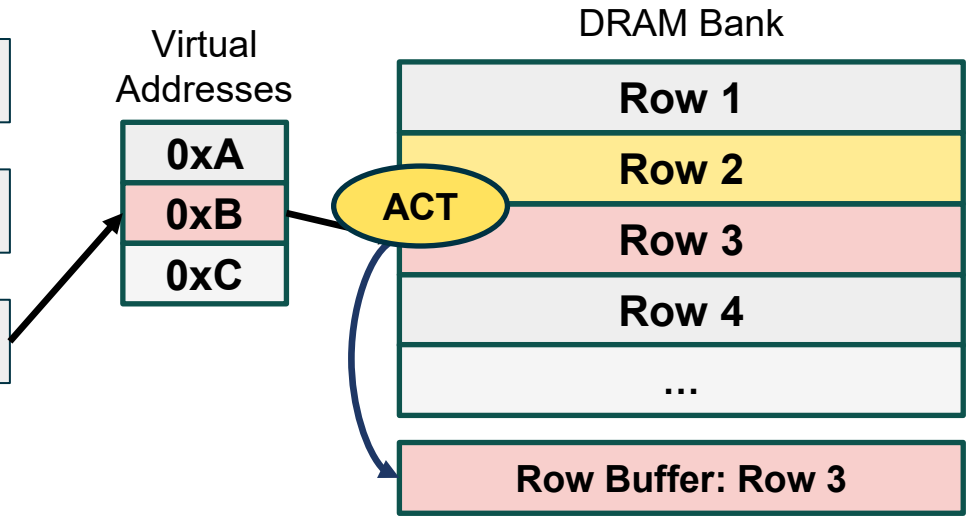
Basic Rowhammer Attack

Launching Naïve Rowhammer Attack on GPUs

1. **Un-cached** Memory Access

2. Row Activated **(ACT)**, Bring to Buffer

3. ACT Diff. Row, Close Prior Row



Basic Rowhammer Attack

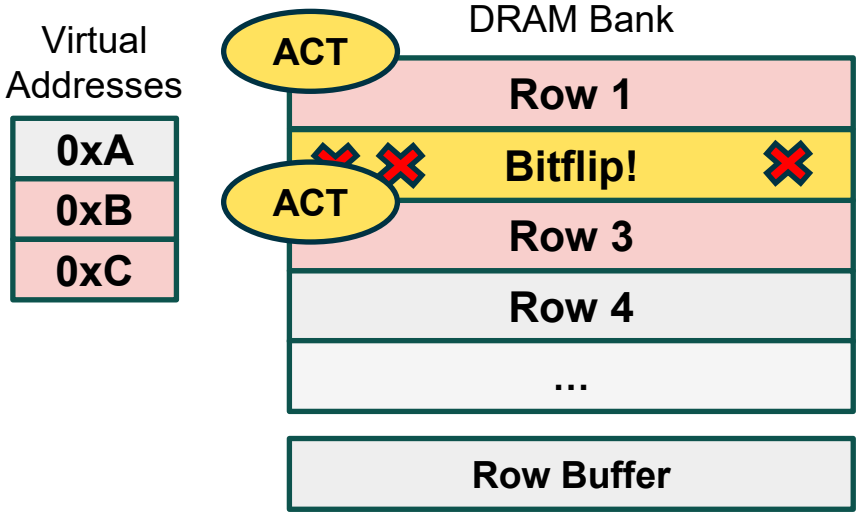
Launching Naïve Rowhammer Attack on GPUs

1. **Un-cached** Memory Access

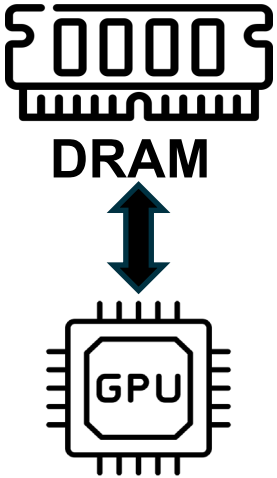
2. Row Activated **(ACT)**, Bring to Buffer

3. ACT Diff. Row, Close Prior Row

4. Repeat 1-3

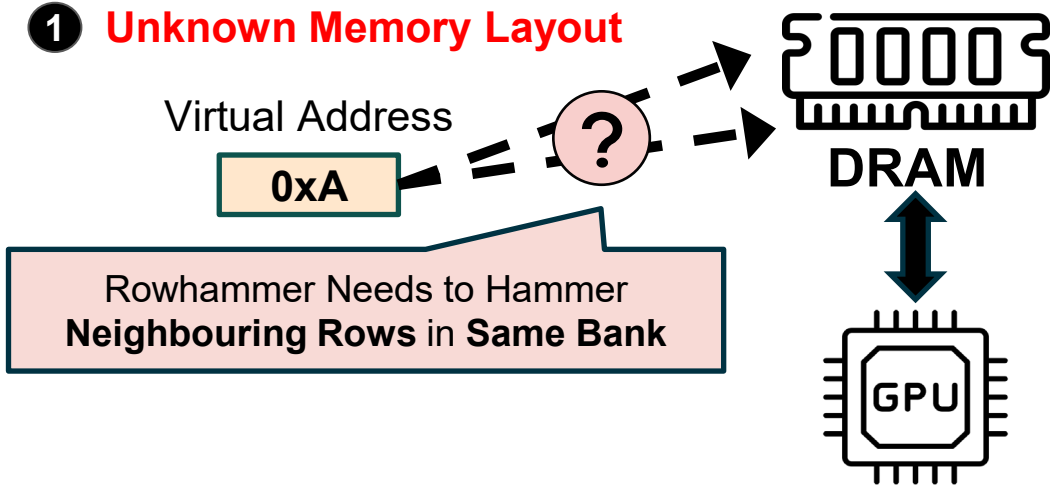


Challenges



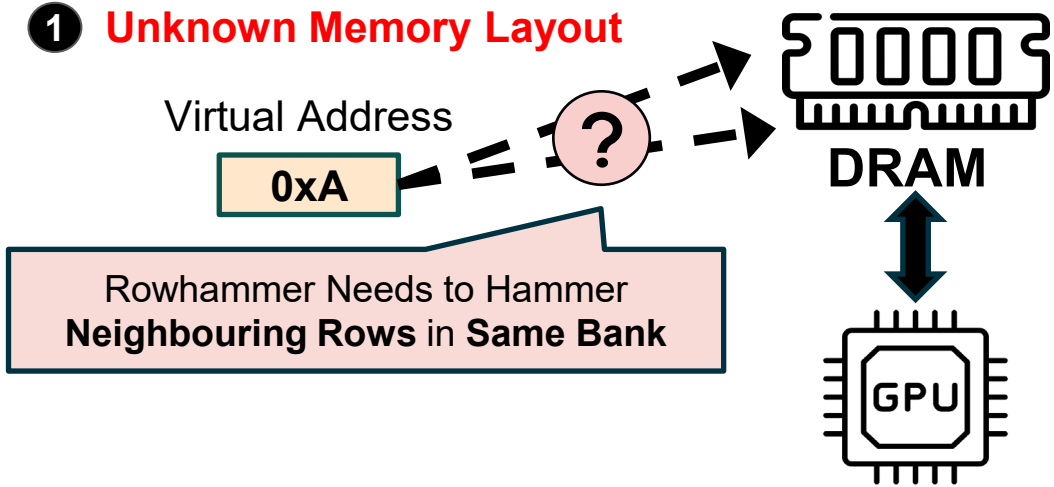
Challenges

1 **Unknown Memory Layout**



Challenges

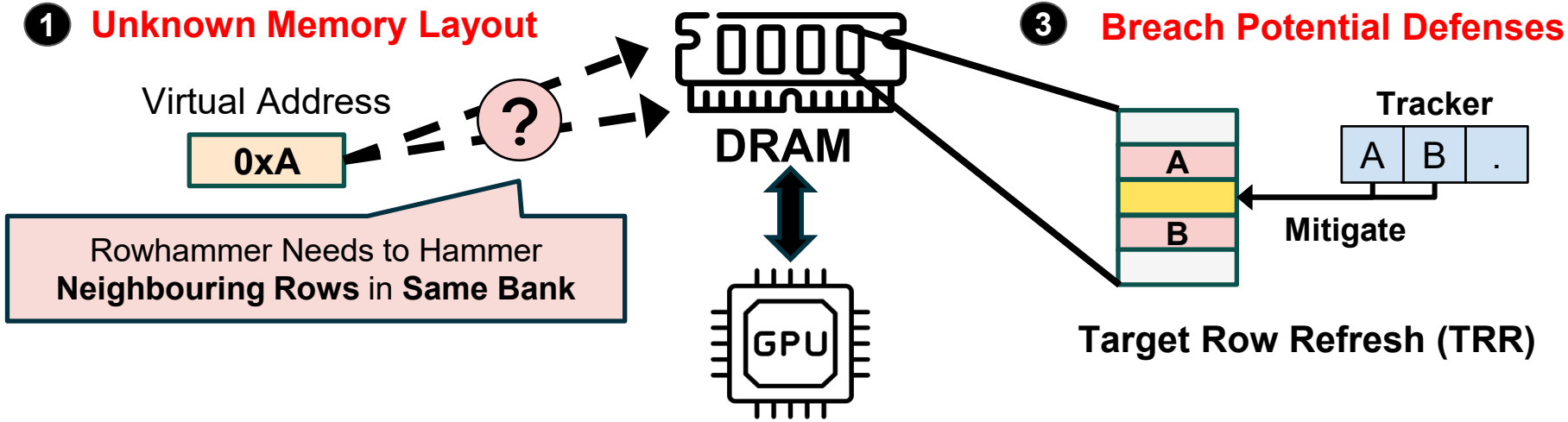
1 **Unknown Memory Layout**



4X-5X Higher Latency Than CPU, Limited Hammering Intensity

2 **High Memory Latency**

Challenges



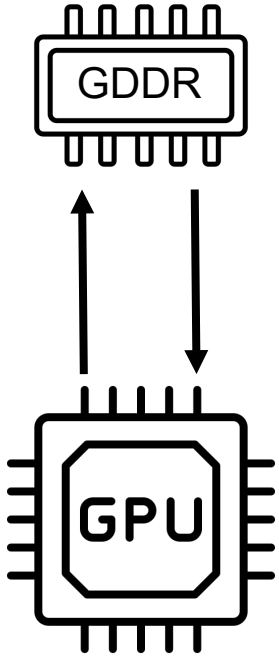
4X-5X Higher Latency Than CPU, Limited Hammering Intensity

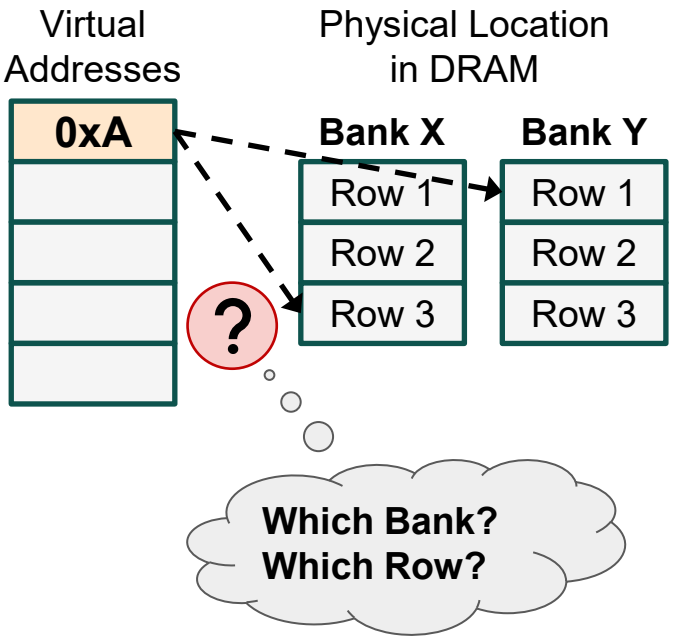
2 **High Memory Latency**

Challenges

1 Unknown Memory Layout

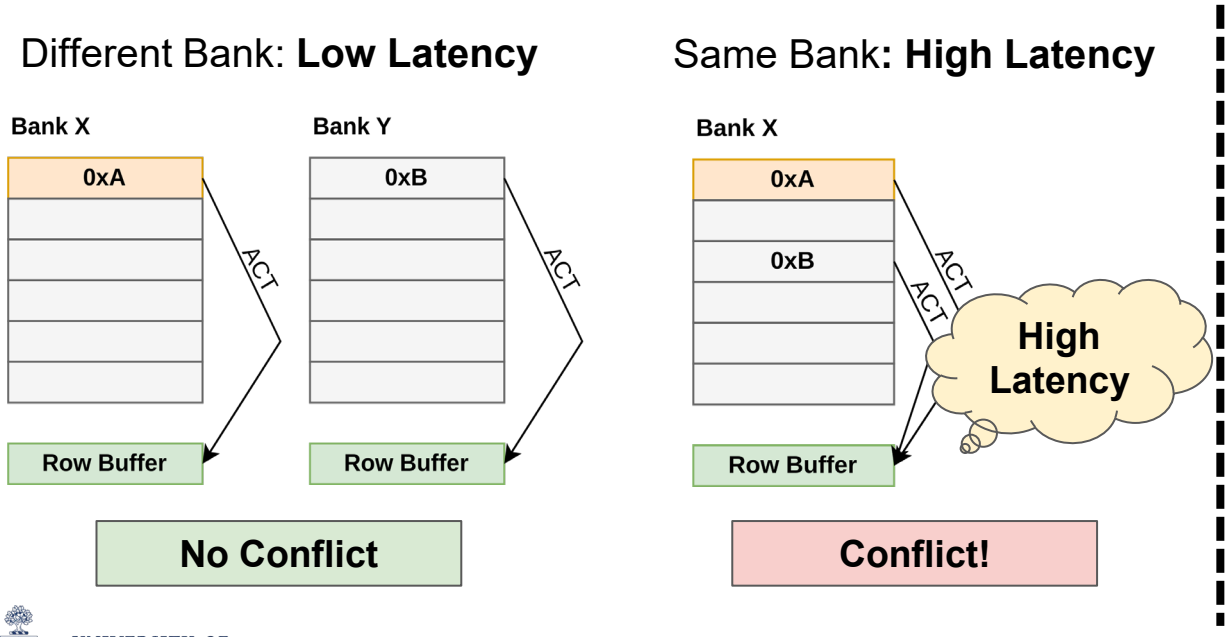
Rowhammer Needs to Hammer Neighbouring Rows in Same Bank





Challenge 1: Uncovering Memory Layout

Side Channel for Reverse-Engineering **Address Mapping**[1]



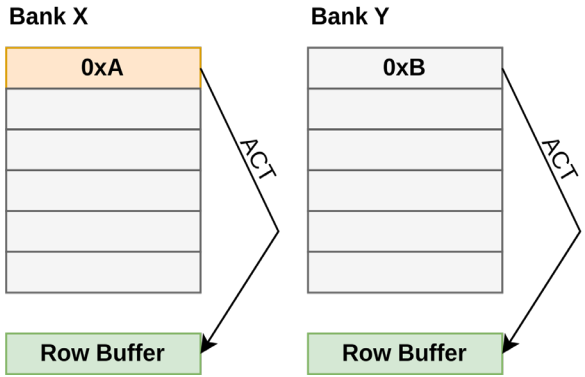
UNIVERSITY OF TORONTO

[1]DRAMA: Exploiting DRAM Addressing for Cross-CPU Attacks (SEC '16)

Challenge 1: Uncovering Memory Layout

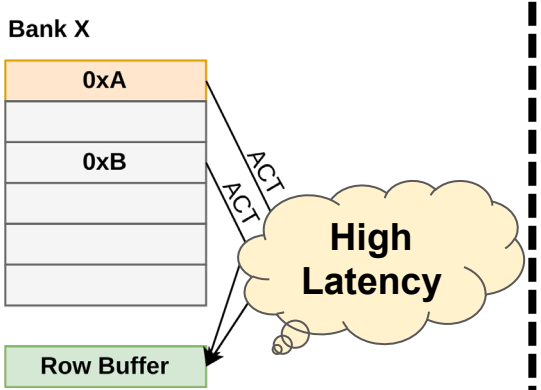
Side Channel for Reverse-Engineering **Address Mapping**[1]

Different Bank: **Low Latency**



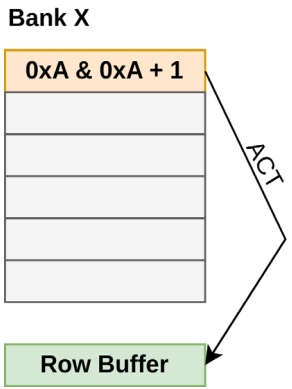
No Conflict

Same Bank**: High Latency**



Conflict!

Same Row**: Low Latency**

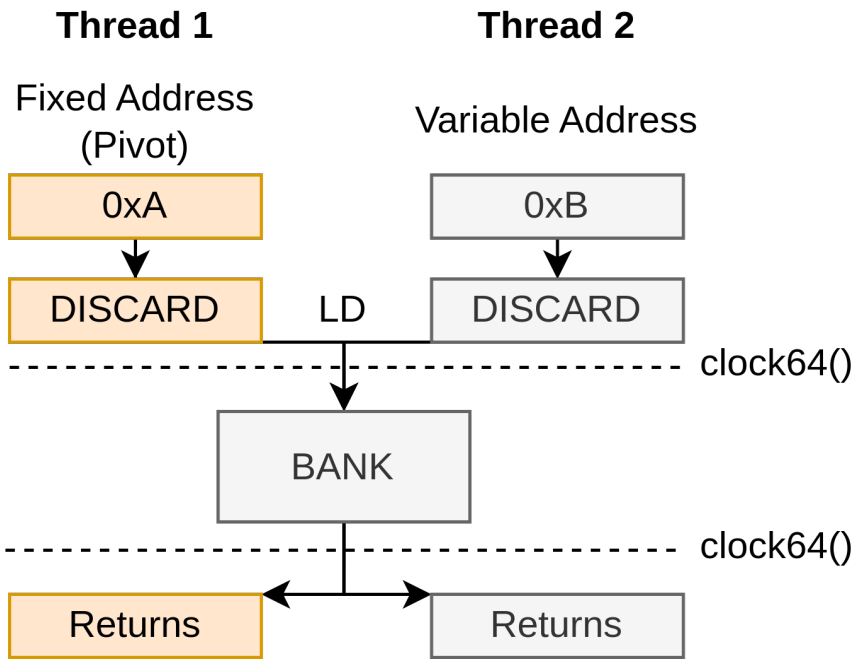


No Conflict

UNIVERSITY OF TORONTO

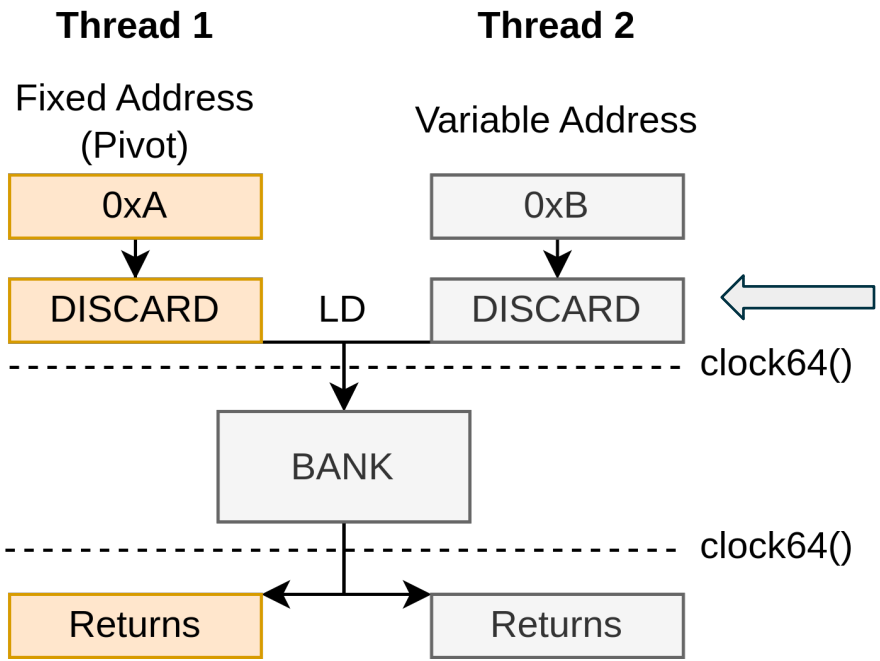
[1]DRAMA: Exploiting DRAM Addressing for Cross-CPU Attacks (SEC '16)

Challenge 1: Uncovering Memory Layout



Use GPU Threads to access Address Pairs, recording latency

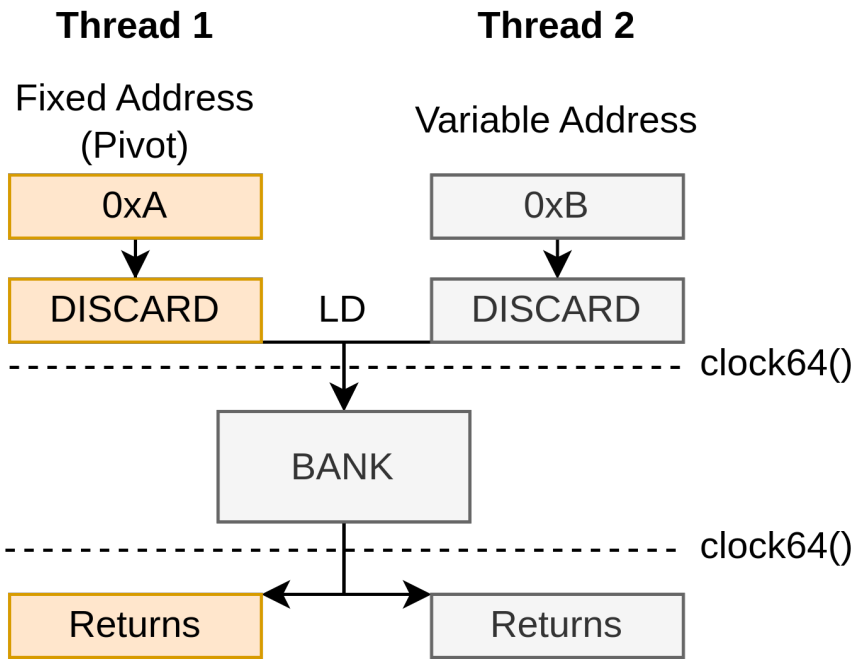
Challenge 1: Uncovering Memory Layout



Use GPU Threads to access Address Pairs, recording latency

Clears cache

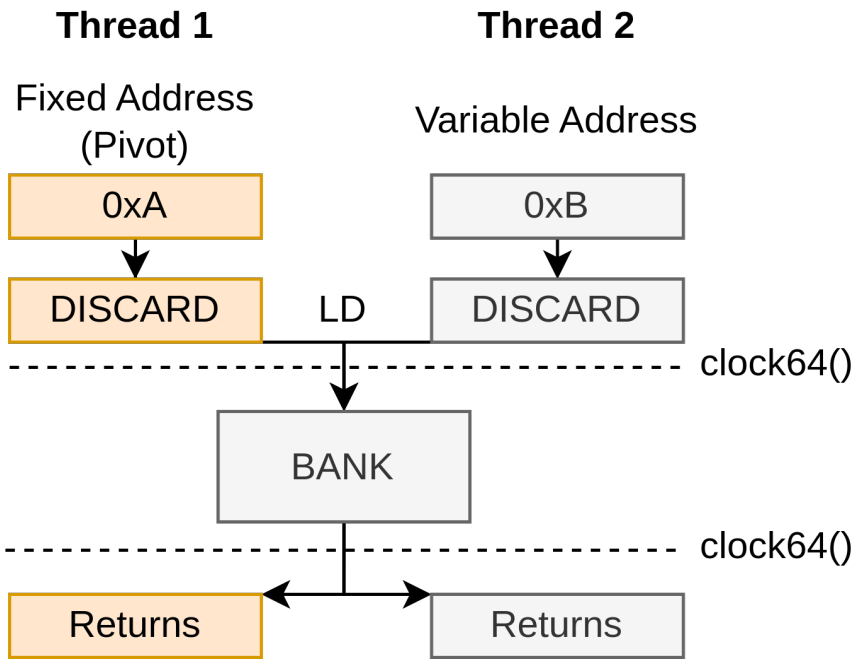
Challenge 1: Uncovering Memory Layout



Use GPU Threads to access Address Pairs, recording latency

The **slowest** access is recorded

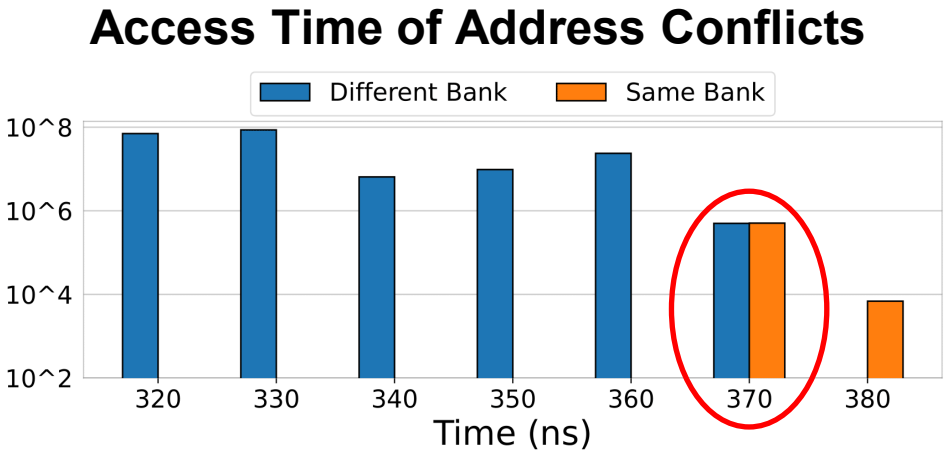
Challenge 1: Uncovering Memory Layout



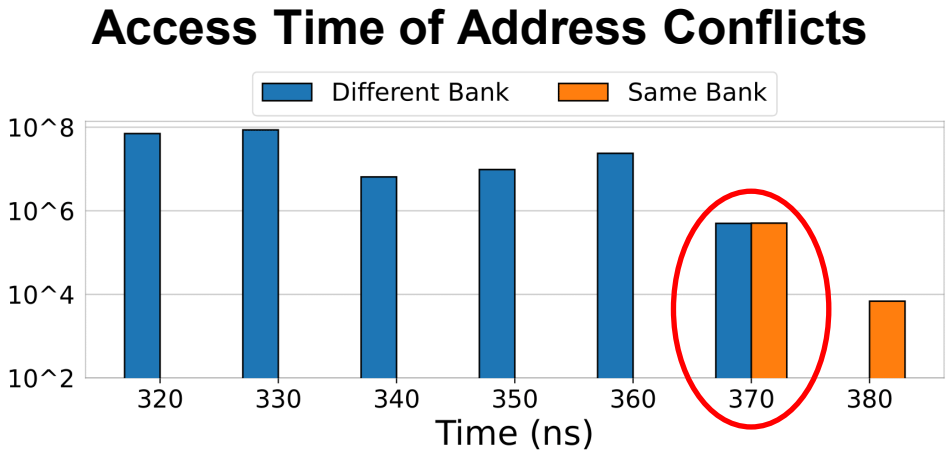
Use GPU Threads to access Address Pairs, recording latency

To avoid system noise,
Take minimum of 10+ attempts

Challenge 1: Uncovering Memory Layout

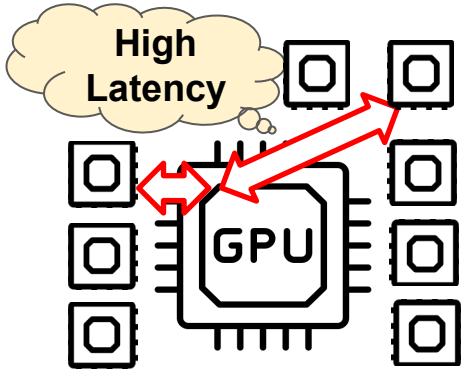


Challenge 1: Uncovering Memory Layout

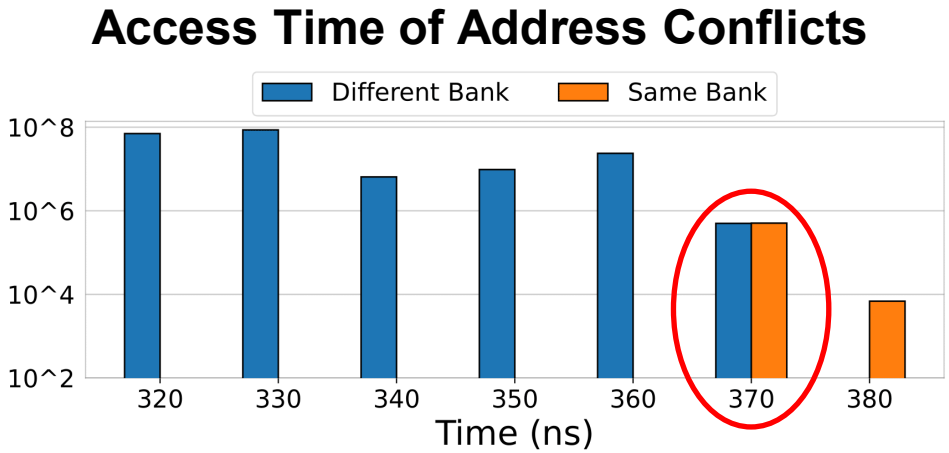


NUMA: **High Latency**
(Non-Uniform Memory Access)

Observation: Due to **NUMA** effect, access time depends on physical location

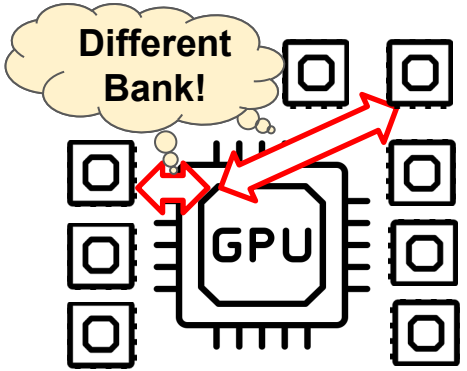


Challenge 1: Uncovering Memory Layout

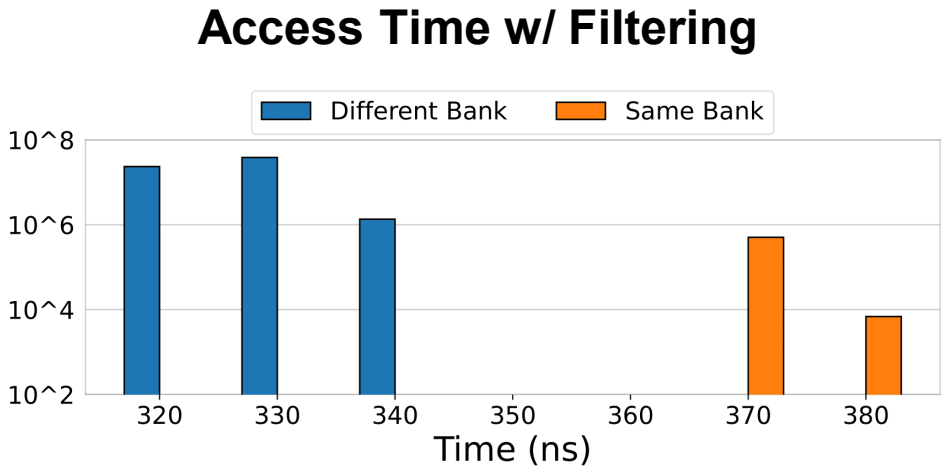


NUMA: **High Latency**
(Non-Uniform Memory Access)

Insight: Same bank addresses are physically close, having similar access latency.



Challenge 1: Uncovering Memory Layout



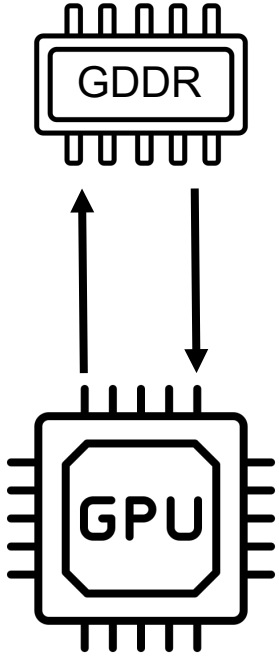
We have Virtual-Physical mapping and ready to hammer!

Challenges

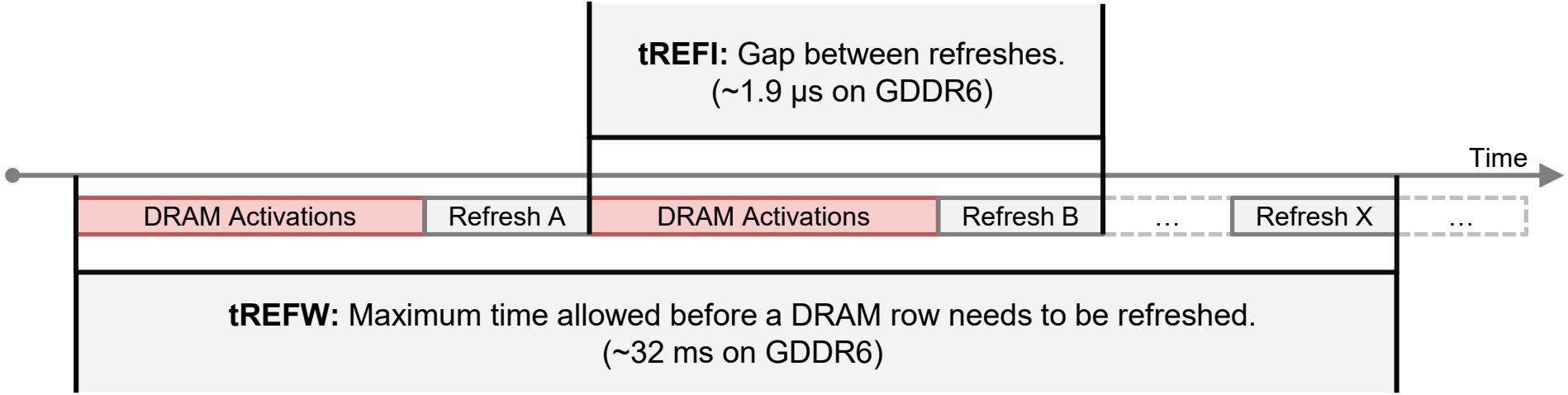
1. Unknown Memory Layout

2. High Memory Latency

4X-5X Higher Memory Access Latency Than CPU!



Challenge 2: Rowhammer Is Time Constrained!



Goal: Trigger bit-flips before refresh

More ACTs per tREFW → Higher chance for bit-flips

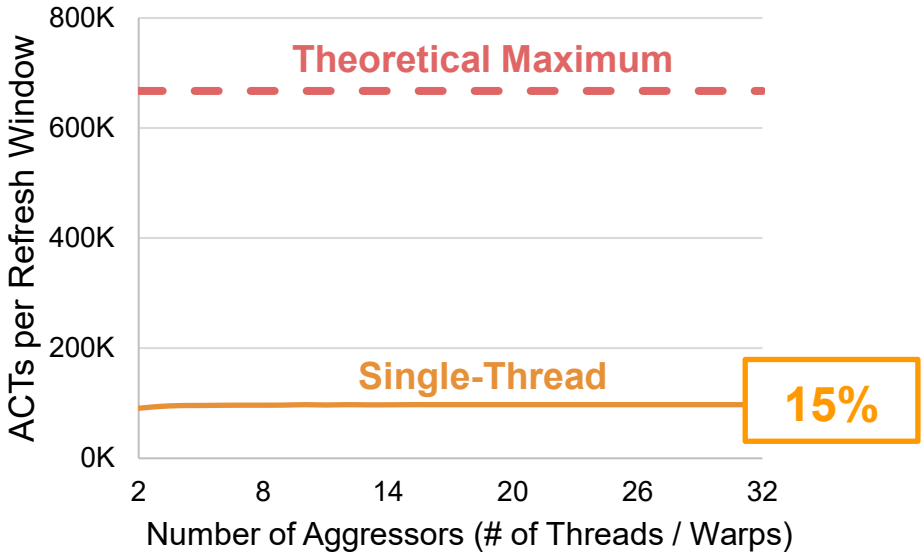
Challenge 2: Increasing Hammering Intensity

Goal: More ACTs per tREFW → Higher chance for bit-flips

❶ Single-Thread Hammering

Large memory access latency.

Low hammering intensity.

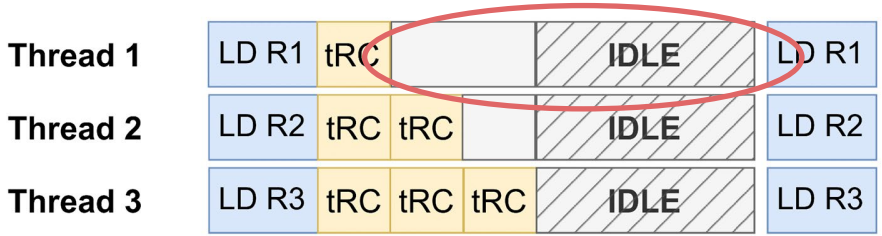


UNIVERSITY OF TORONTO

Challenge 2: Increasing Hammering Intensity

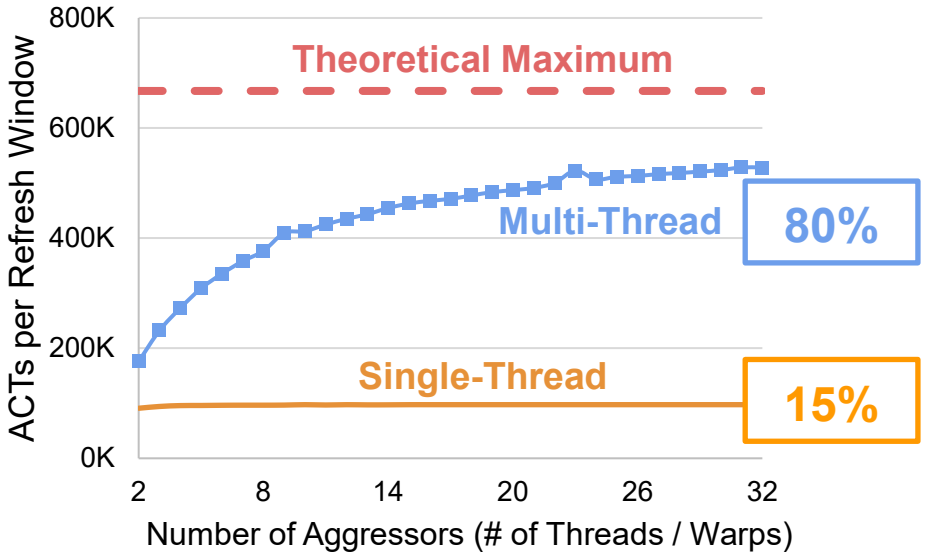
Goal: More ACTs per tREFW → Higher chance for bit-flips

❷ Multi-Thread Hammering



Parallel activations but threads wait.

Medium hammering intensity.

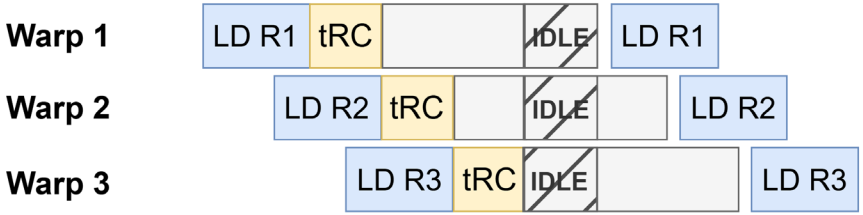


UNIVERSITY OF TORONTO

Challenge 2: Increasing Hammering Intensity

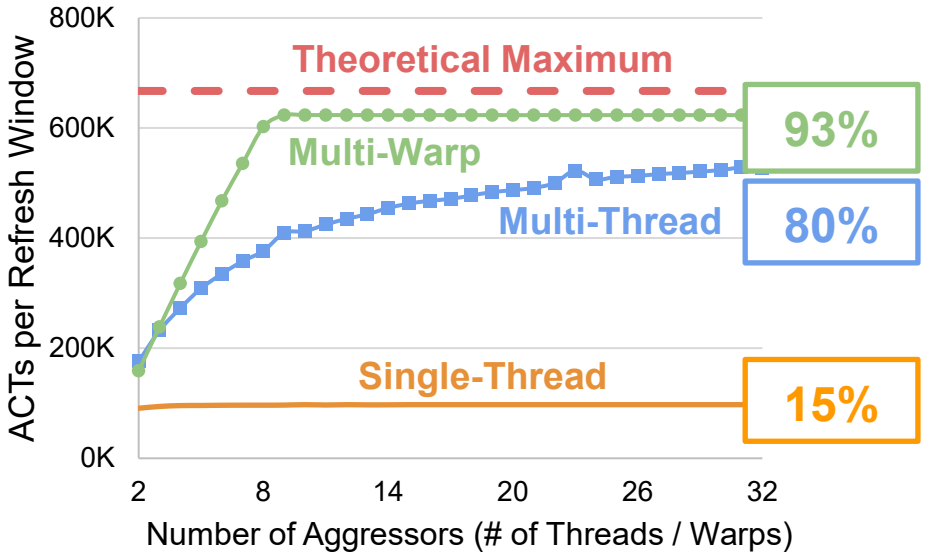
Goal: More ACTs per tREFW → Higher chance for bit-flips

❸ Multi-Warp Hammering



Idle Times Minimized

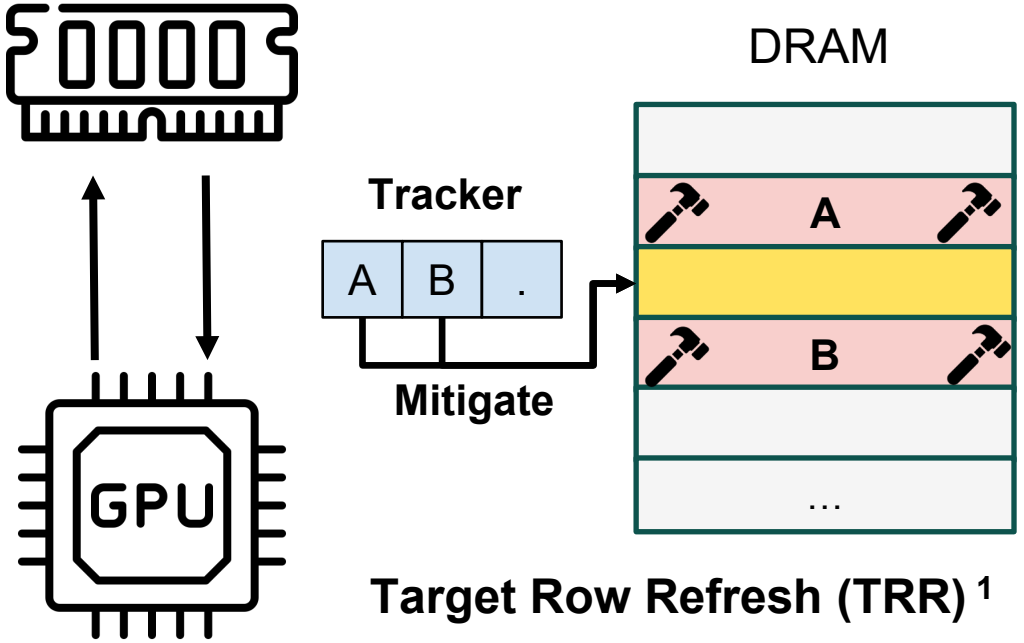
High hammering intensity.



UNIVERSITY OF TORONTO

Challenges

1. Unknown Memory Layout
2. High Memory Latency
3. **Breach Potential Defenses**



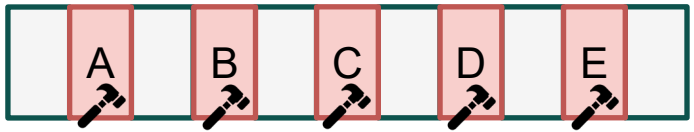
Target Row Refresh (TRR) [1]

UNIVERSITY OF TORONTO

[1]Micron, "DDR4 SDRAM Datasheet," p. 380, 2016.

Challenge 3: Bypassing Defenses

Goal 1: Overflow Tracker

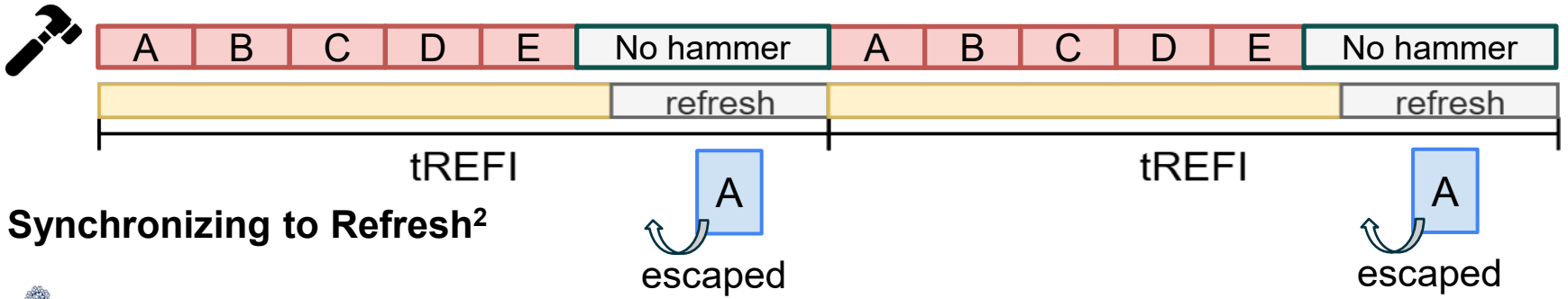


Many-Sided Hammering[1]

Tracker (4-entry)

escaped

Goal 2: Consistent Eviction



Synchronizing to Refresh[2]

UNIVERSITY OF TORONTO

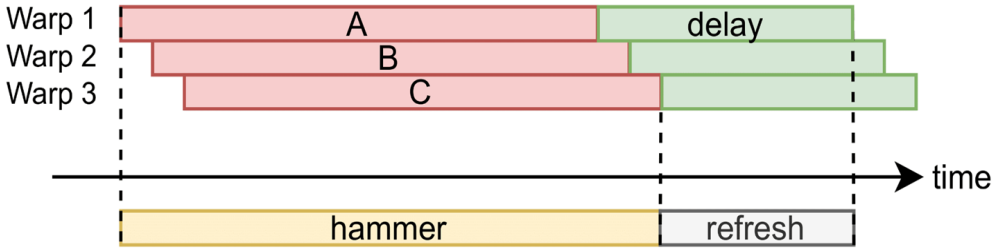
[1]TRRespass: Exploiting the Many Sides of Target Row Refresh

[2]SMASH: Synchronized Many-sided Rowhammer Attacks from JavaScript

Challenge 3: Bypassing Defenses

Prior Work: Synchronized Hammering In **Single-Threaded** Setting

Our Work: Synchronized Hammering In **Multi-Threaded** Setting



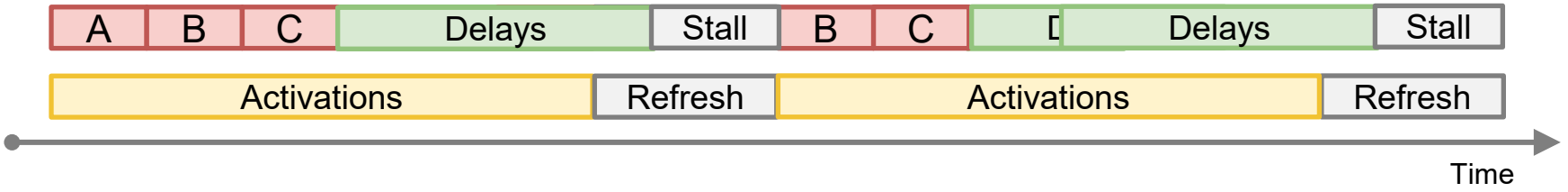
Per-Warp Synchronization:
Add the same amount of delay after each warp's hammering.

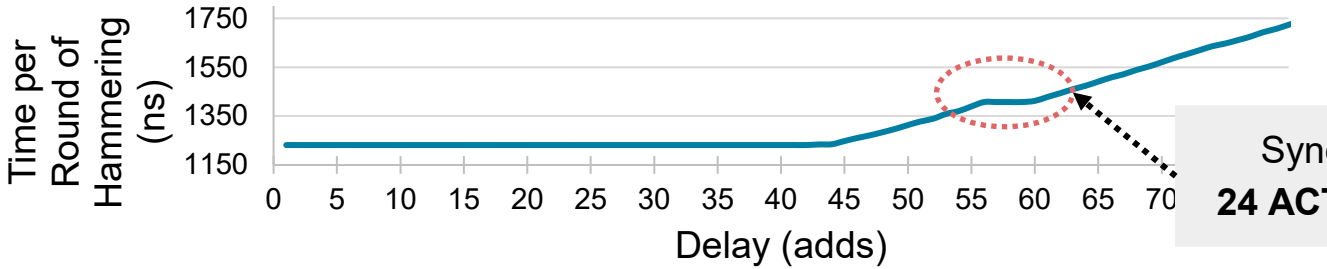
UNIVERSITY OF TORONTO

[1]TRRespass: Exploiting the Many Sides of Target Row Refresh

[2]SMASH: Synchronized Many-sided Rowhammer Attacks from JavaScript

Challenge 3: Bypassing Defenses





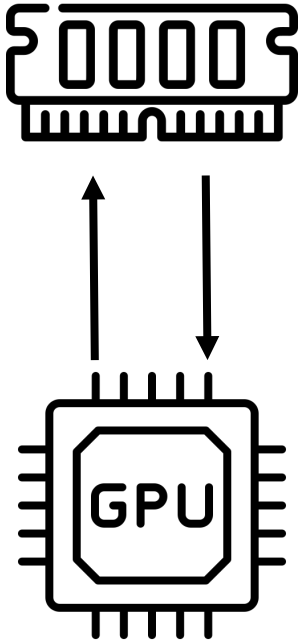
Synchronized to tREFI after **24 ACTs** and **delay of ~56 adds**

Synchronization allows the tracker to evict the same entry every time.

✓ Challenge Solved: **Our attack can accurately synchronize to tREFI.**

Challenges

1. Unknown Memory Layout
2. High Memory Latency
3. Breach Potential Defenses?



- ✓ Timing Reverse-Engineering
- ✓ Multi-Warp Hammering
- ✓ Synchronized Multi-Warp

Bit-flips Observed

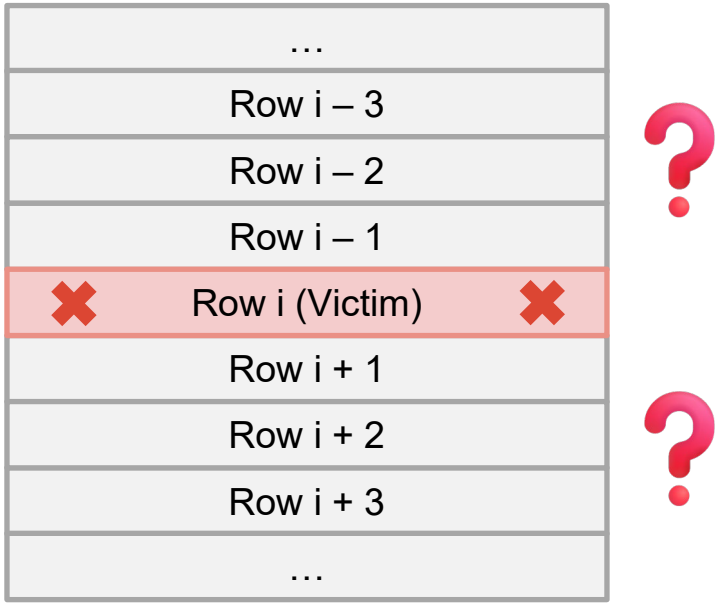
Hammering Campaign on …

- 3 GPUs
- 4 banks each
- 8/12/16/20/24 aggressors
- 2 checkered data patterns
- Took ~30h / bank / GPU

A6000 GDDR6 DRAM is vulnerable to Rowhammer, with bit-flips observed on every bank hammered.
No flips were observed on the A100 or RTX3080.

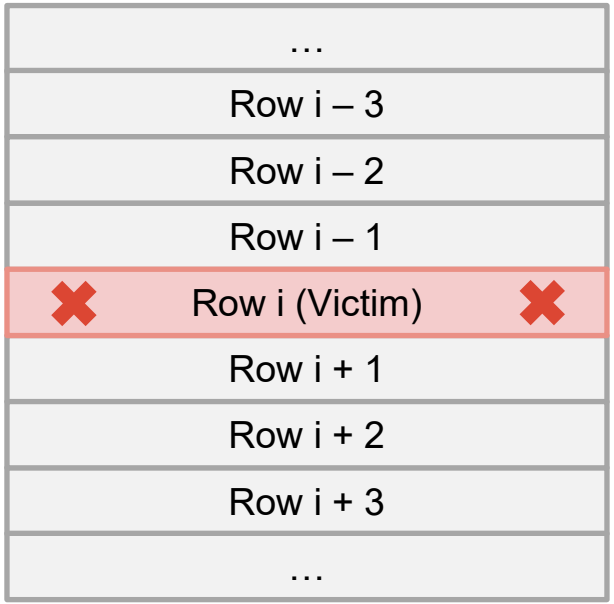
Bit-flips Characterization – Critical Aggressor

Which aggressor row(s) are responsible for the bit-flip?



Bit-flips Characterization – Critical Aggressor

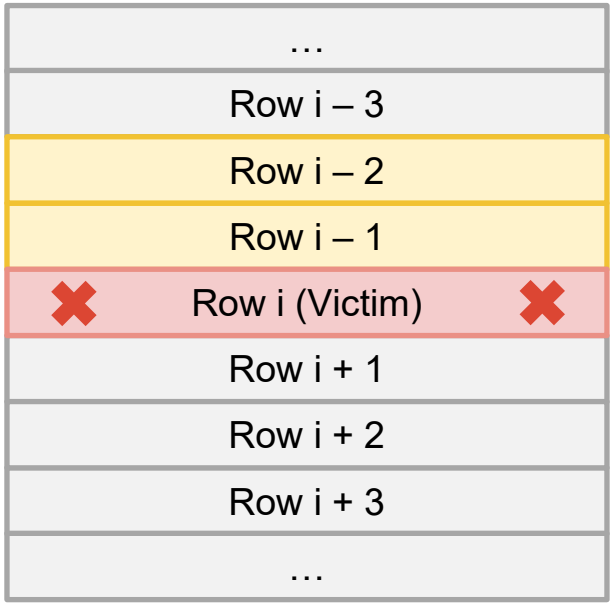
Which aggressor row(s) are responsible for the bit-flip?



Bit-flips	Critical Aggressor Row					
	i – 3	i – 2	i – 1	i + 1	i + 2	i + 3
A1		✓				
B1		✓	✓			
B2					✓	✓
B3					✓	
C1		✓	✓			
D1		✓	✓			
D2					✓	
D3				✓	✓	

Bit-flips Characterization – Critical Aggressor

Which aggressor row(s) are responsible for the bit-flip?

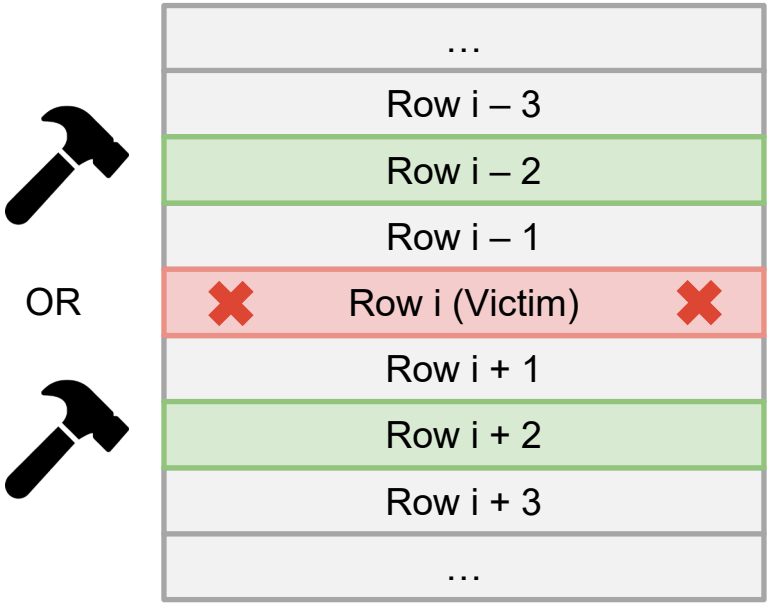


Observations:

Single-sided hammering from one side only

Bit-flips Characterization – Critical Aggressor

Which aggressor row(s) are responsible for the bit-flip?



Observations:

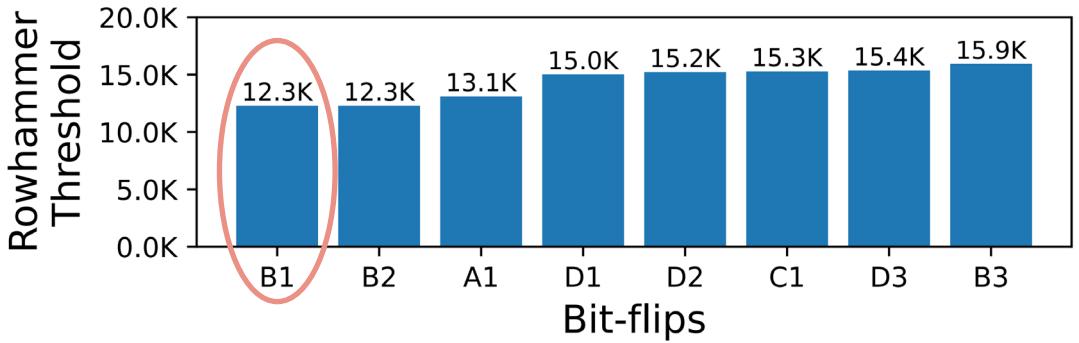
Single-sided hammering from one side only

Row i ± 2 is the most effective

→ **Non-contiguous DRAM row layout**

Bit-flips Characterization – Rowhammer Threshold

Rowhammer Threshold (T_{RH}): The *minimum* number of ACTs to trigger bit-flips.

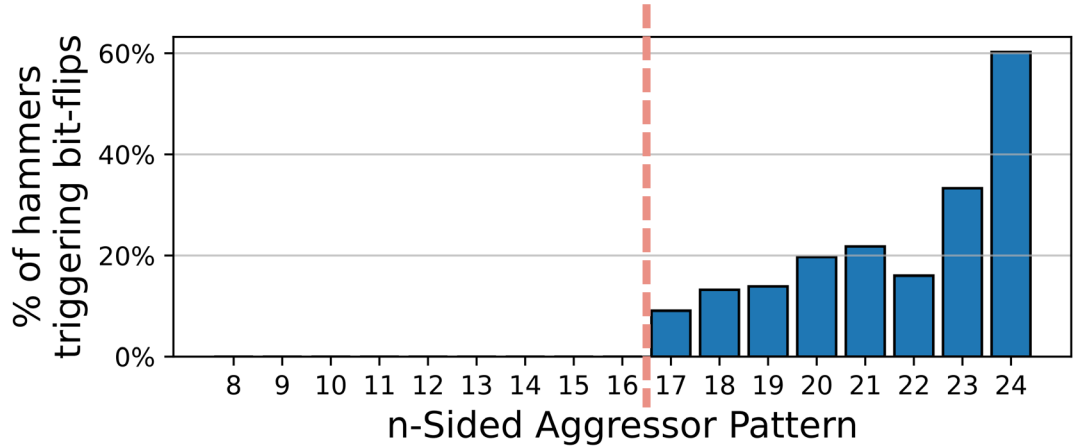


Need >12.3K activations to a single DRAM row on the A6000 GPU to see a bit-flip

Bit-flips Characterization – TRR Tracker Size

Tracker

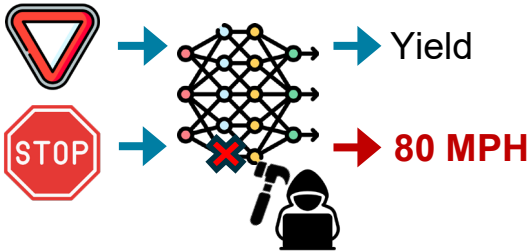
B	C	D	…?



The TRR tracker holds **16** rows per bank on the A6000.

Exploit: Bit-Flipping Attacks on ML Models

Accuracy Degradation
(Terminal Brain Damage[1])



Model Backdoor
(Don't Knock[2])

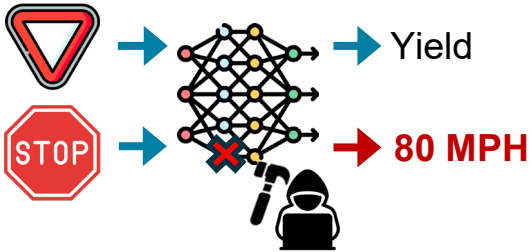
LLM Jailbreak
(PrisonBreak[3])

UNIVERSITY OF TORONTO

[1]Terminal Brain Damage: Exposing the Graceless Degradation in Deep Neural Networks Under Hardware Fault Attacks (SEC '19)
[2]Don't Knock! Rowhammer at the Backdoor of DNN Models (DSN '23)
[3]PrisonBreak: Jailbreaking Large Language Models with Fewer Than Twenty-Five Targeted Bit-flips (Arxiv)

Exploit: Bit-Flipping Attacks on ML Models

Accuracy Degradation
(Terminal Brain Damage[1])



Model Backdoor
(Don't Knock[2])

LLM Jailbreak
(PrisonBreak[3])

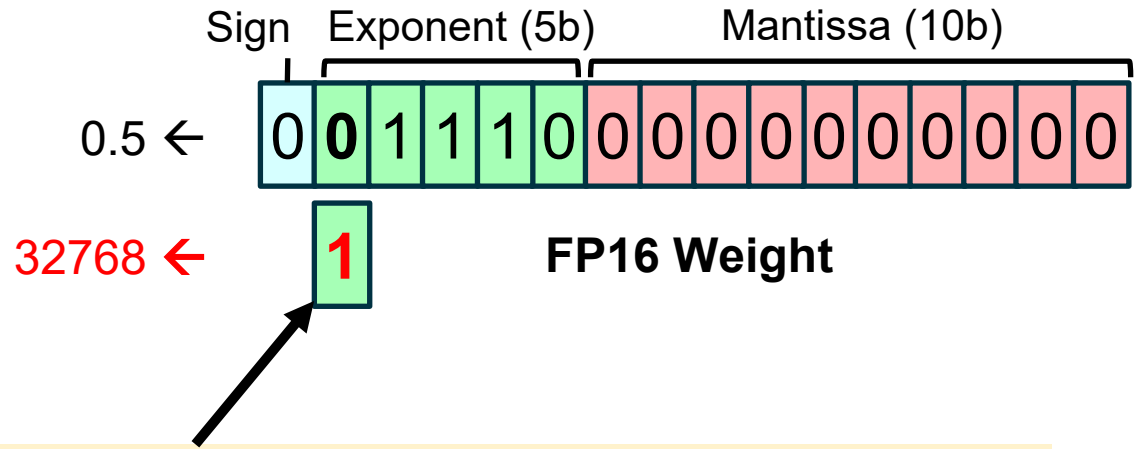
We Demonstrate Accuracy Degradation Attacks on ML Models

UNIVERSITY OF TORONTO

[1]Terminal Brain Damage: Exposing the Graceless Degradation in Deep Neural Networks Under Hardware Fault Attacks (SEC '19)
[2]Don't Knock! Rowhammer at the Backdoor of DNN Models (DSN '23)
[3]PrisonBreak: Jailbreaking Large Language Models with Fewer Than Twenty-Five Targeted Bit-flips (Arxiv)

Exploit: Bit-Flipping Attacks on ML Models

Accuracy Degradation
(Terminal Brain Damage[1])

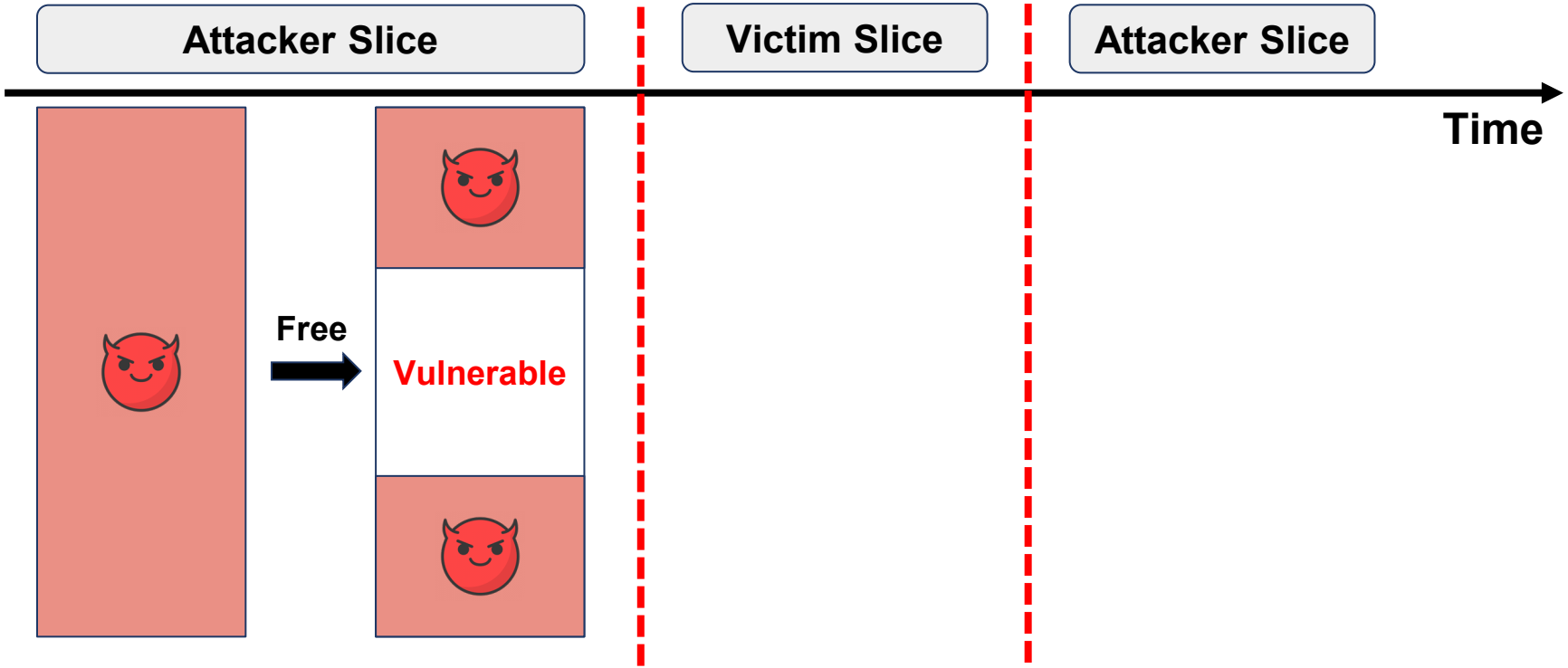


Single Bit-Flip in a Weight Leads to Significant Accuracy Loss

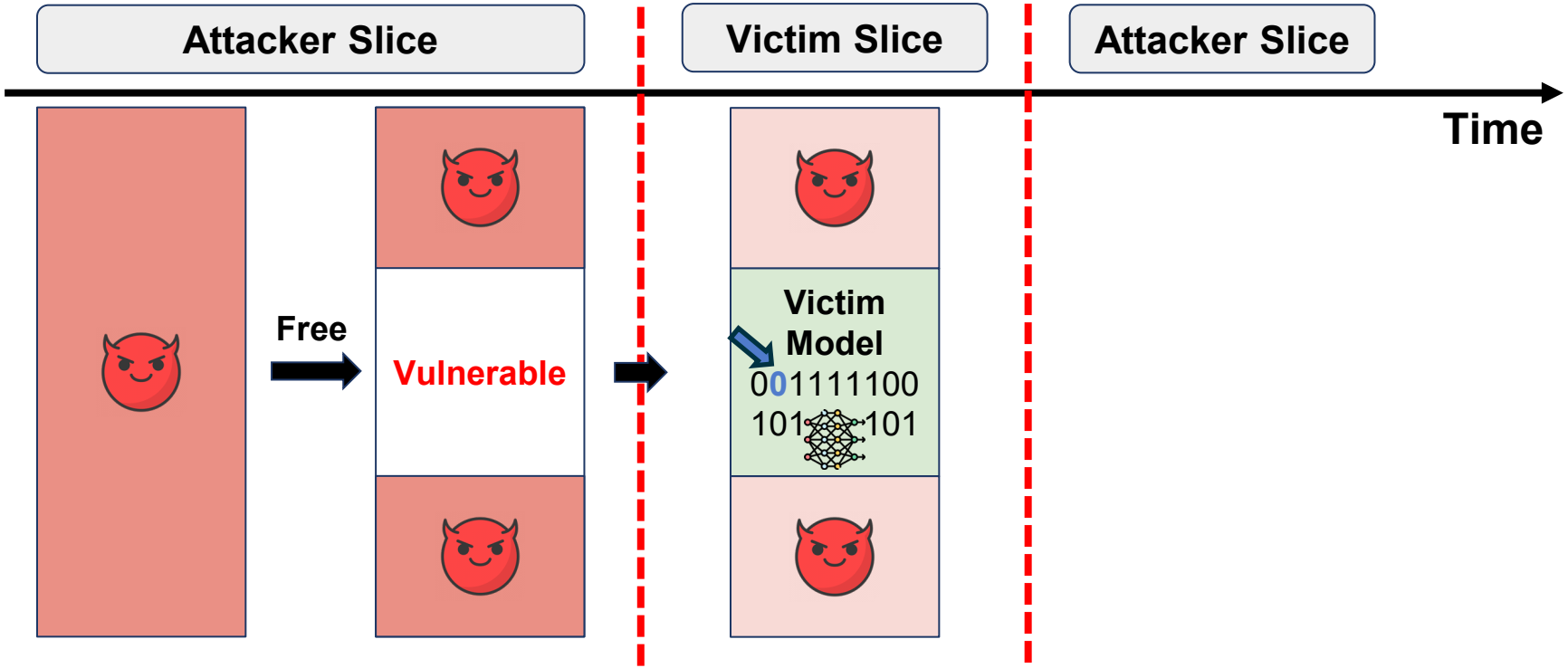
UNIVERSITY OF TORONTO

[1]Terminal Brain Damage: Exposing the Graceless Degradation in Deep Neural Networks Under Hardware Fault Attacks (SEC '19)

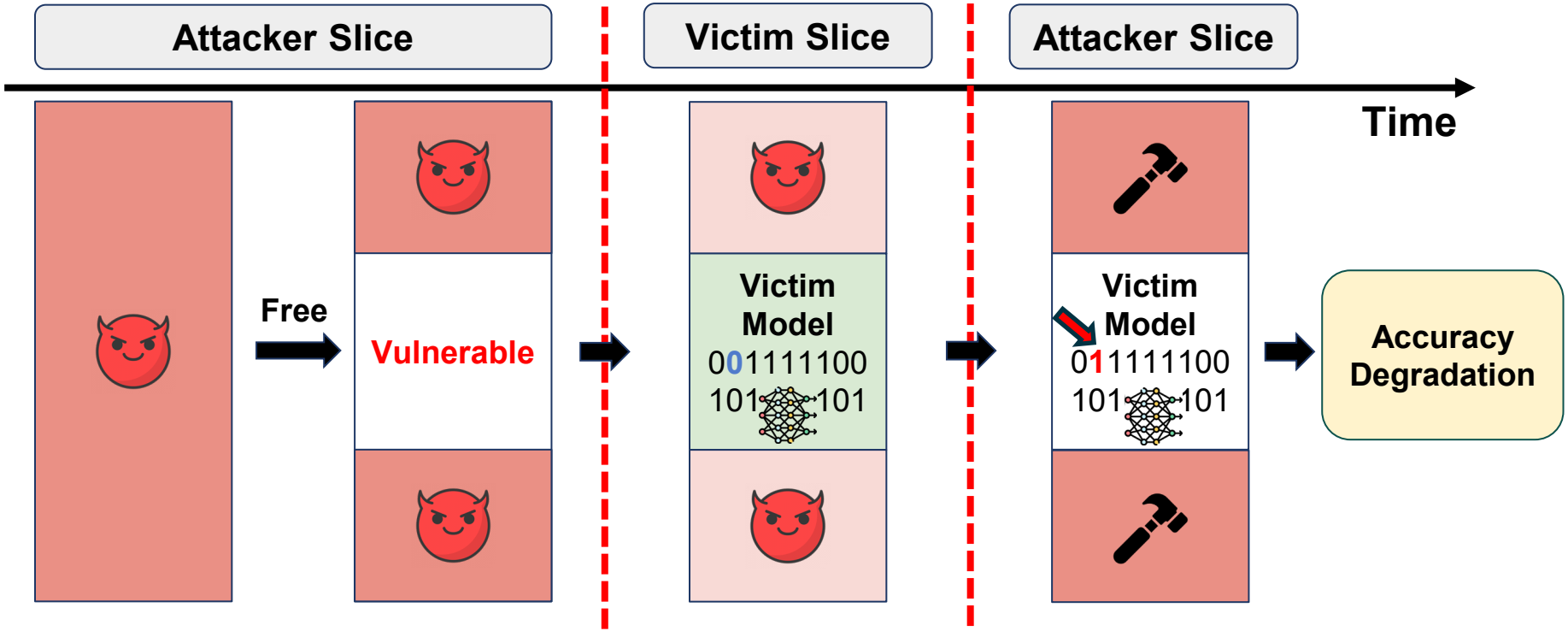
Exploit: Memory Massaging on Time-Sliced GPU



Exploit: Memory Massaging on Time-Sliced GPU



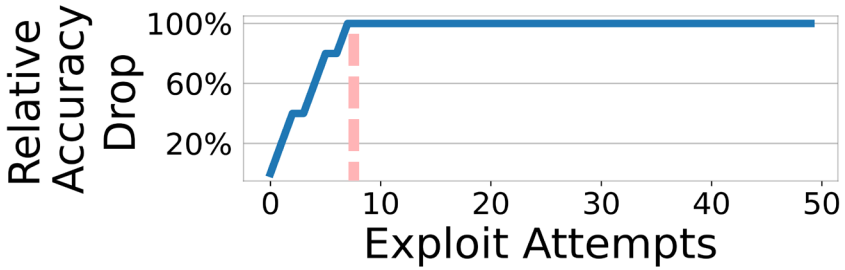
Exploit: Memory Massaging on Time-Sliced GPU



Exploit: Accuracy Degradation on ML Models

Flip Top Bit in Exponent of a Random Weight

ML Models	Base Accuracy	Degraded Accuracy
AlexNet	56.66%	**0.10%**
VGG16	72.22%	**0.08%**
ResNet50	80.26%	**0.08%**
DenseNet161	77.20%	**0.08%**
InceptionV3	69.92%	**0.04%**



Degrade Accuracy from **80%** to **0.1%**
(**< 10** Attempts Targeting Random Weight)

UNIVERSITY OF TORONTO

Mitigations

We disclosed to NVIDIA who confirmed the issue

Recommendation[1]:
Enable **Error Correction Code (ECC)**

NVIDIA Security Notice[1]:

NVIDIA continues to recommend the following existing DRAM mitigations to prevent or lessen the likelihood of Rowhammer attacks including:

- Ensuring System level ECC is enabled across the following NVIDIA products (see "Enabling SYS-ECC" below):
  - **Blackwell**
    - **Data Center: NVIDIA HGX, DGX series (GB200, B200, B100)**
    - **Workstation: NVIDIA RTX PRO series**
  - **Ada**
    - **Data Center: NVIDIA L40S, L40, L4**
    - **Workstation: NVIDIA RTX 6000, 5000, 4500, 4000, 4000 SFF, 2000**
  - **Hopper**
    - **Data Center: NVIDIA HGX, DGX series (H100, H200, GH200, H20, H800)**
  - **Ampere**
    - **Data Center: NVIDIA A100, A40, A30, A16, A10, A2, A800**
    - **Workstation: NVIDIA RTX A6000, A5000, A4500, A4000, A2000, A1000, A400**
  - **Jetson**
    - **Jetson AGX Orin Industrial**
    - **IGX Orin**
  - **Turing**
    - **Data Center: NVIDIA T1000, T600, T400, T4**
    - **Workstation: NVIDIA RTX 8000, RTX 6000, RTX 5000, RTX 4000**
  - **Volta**
    - **Data Center/HPC: Tesla V100, Tesla V100S**
    - **Workstation: Workstation: Quadro GV100**

[1]https://nvidia.custhelp.com/app/answers/detail/a_id/5671

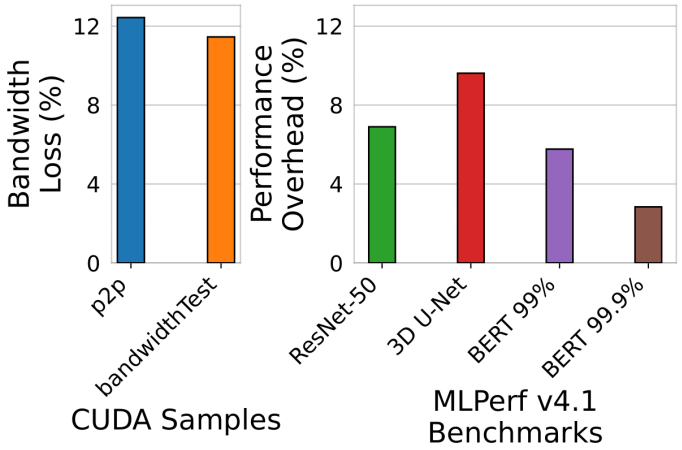
Mitigations

We disclosed to NVIDIA who confirmed the issue

Recommendation[1]:
Enable **Error Correction Code (ECC)**

But…

Incurs up to **10% Slowdown** on A6000



UNIVERSITY OF TORONTO

[1]https://nvidia.custhelp.com/app/answers/detail/a_id/5671

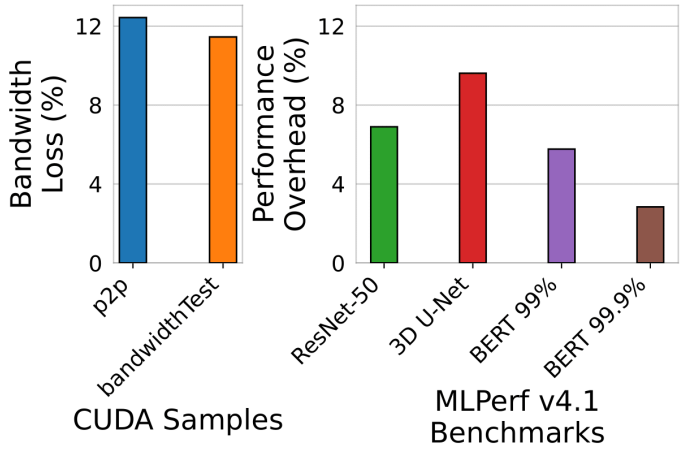
Mitigations

We disclosed to NVIDIA who confirmed the issue

Recommendation[1]:
Enable **Error Correction Code (ECC)**

But…

Incurs up to **10% Slowdown** on A6000



Rowhammer is a Hardware Flaw

We urge adoption of more principled mitigations (e.g., PRAC) in future GPU DRAMs.

UNIVERSITY OF TORONTO

[1]https://nvidia.custhelp.com/app/answers/detail/a_id/5671

Summary

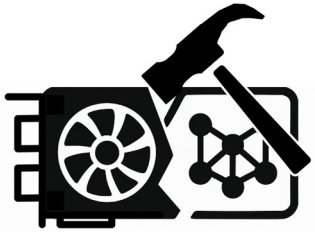
GPUHammer: The First Practical Rowhammer Attack on Discrete GPUs

Novel Techniques to Overcome Challenges for GPU Rowhammer

Exploit Bit-Flips to Degrade DNN Accuracy: **80% → ~0%**

Paper Website

gpuhammer.com



GPUHammer

More in Paper/Poster:

- Expanded Analysis of Reverse-Engineering and Attack Technique
- Detailed Bit-Flip Characterization
- Mitigation Strategies

UNIVERSITY OF TORONTO

shaopenglin@cs.toronto.edu joyce.qu@mail.utoronto.ca gururaj@cs.toronto.edu