Siloz: Leveraging DRAM Isolation Domains to Prevent Inter-VM Rowhammer

Kevin Loughlin

Jonah Rosenblum, Stefan Saroiu, Alec Wolman, Dimitrios Skarlatos, Baris Kasikci

COMPUTER SCIENCE AND ENGINEERING
UNIVERSITY of MICHIGAN

Microsoft

Carnegie Mellon University
Computer Science Department

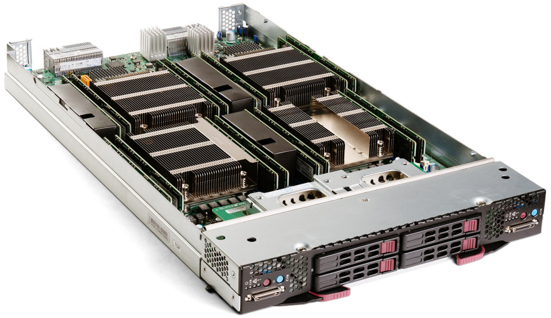
PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

Routine Multi-Tenant Life in the Cloud

Virtual Machine (VM) 0

Virtual Machine (VM) 1

Shared Physical Machine

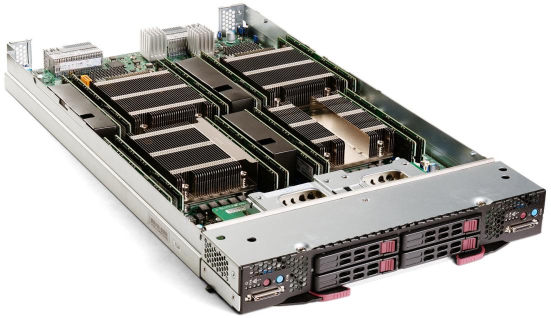


~~Routine~~ Actual Multi-Tenant Life in the Cloud

DATA CORRUPTION OF PROFILE PIC

"How" you ask?
ROWHAMMER BIT FLIPS

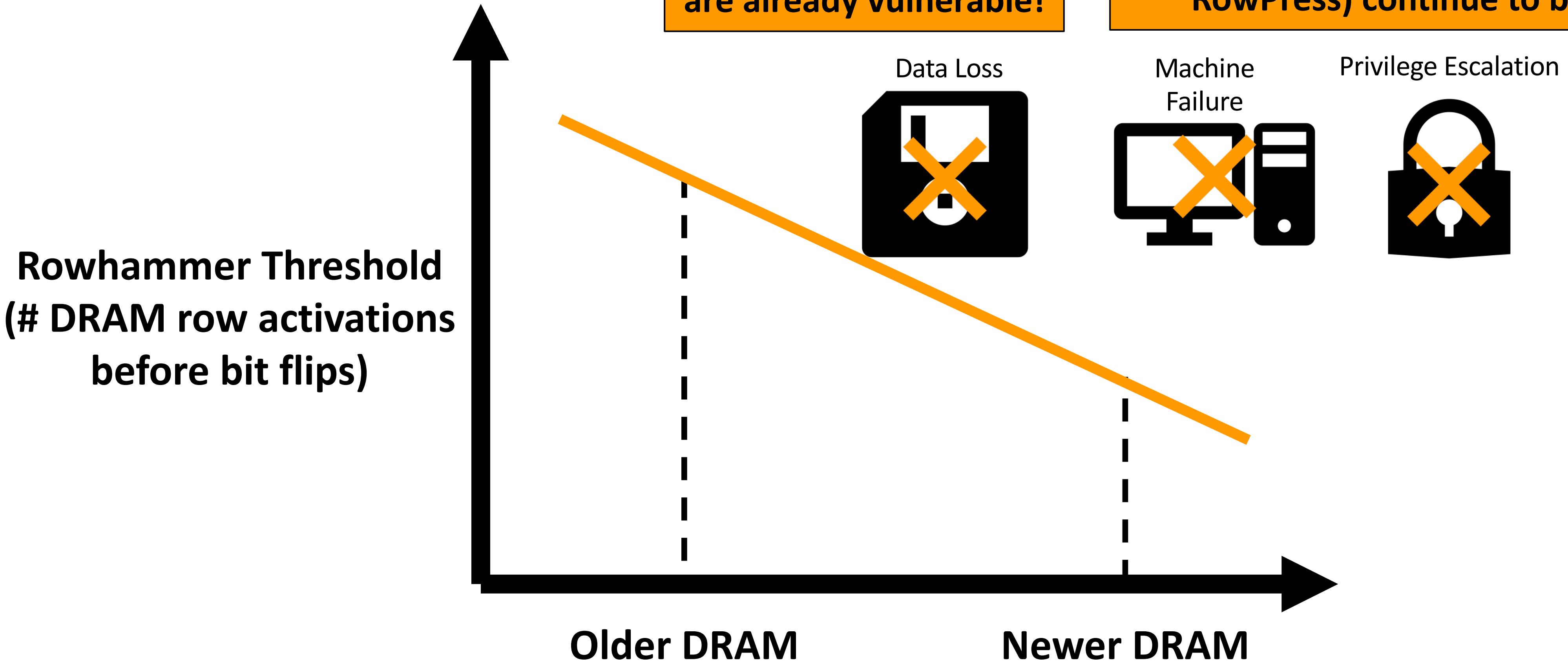
Shared Physical Machine



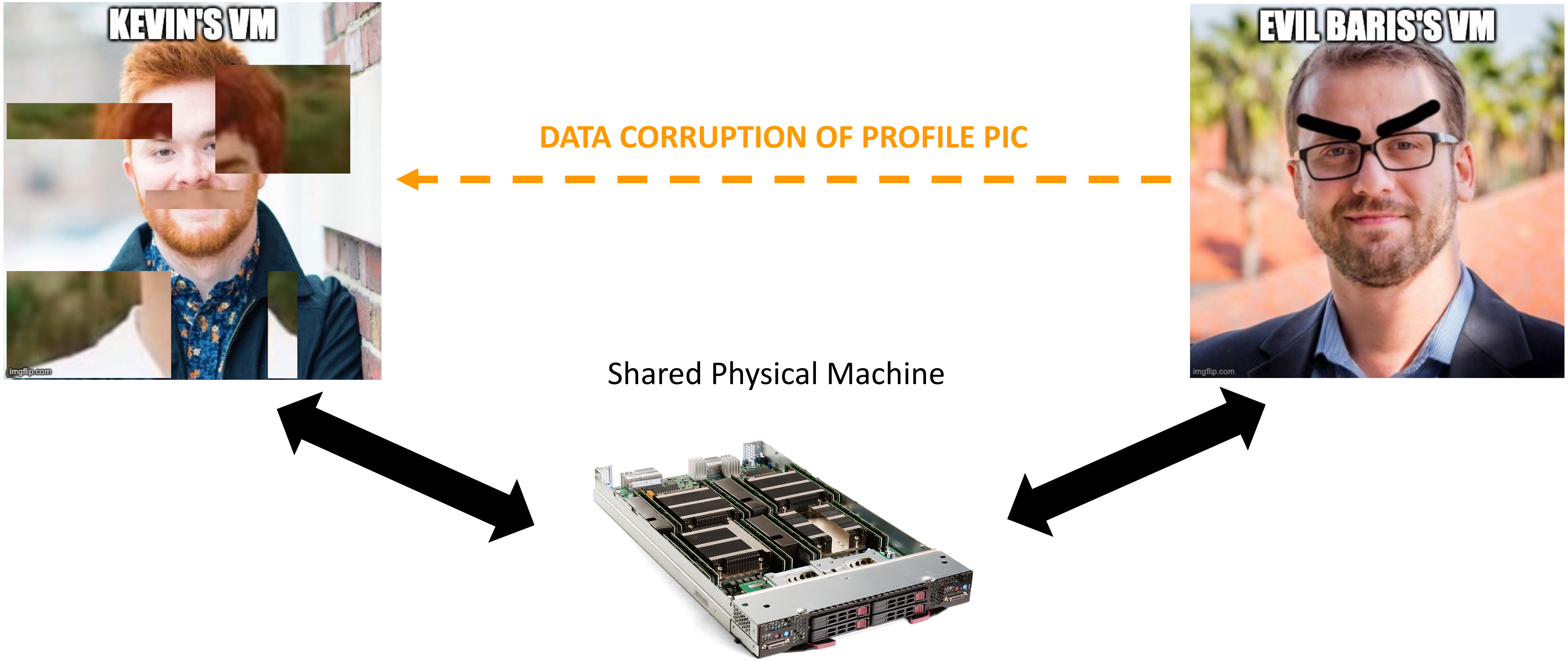
Rowhammer Susceptibility is Increasing

Today's systems are already vulnerable!

Access patterns that flip bits (ex: Half-Double, RowPress) continue to be unearthed!



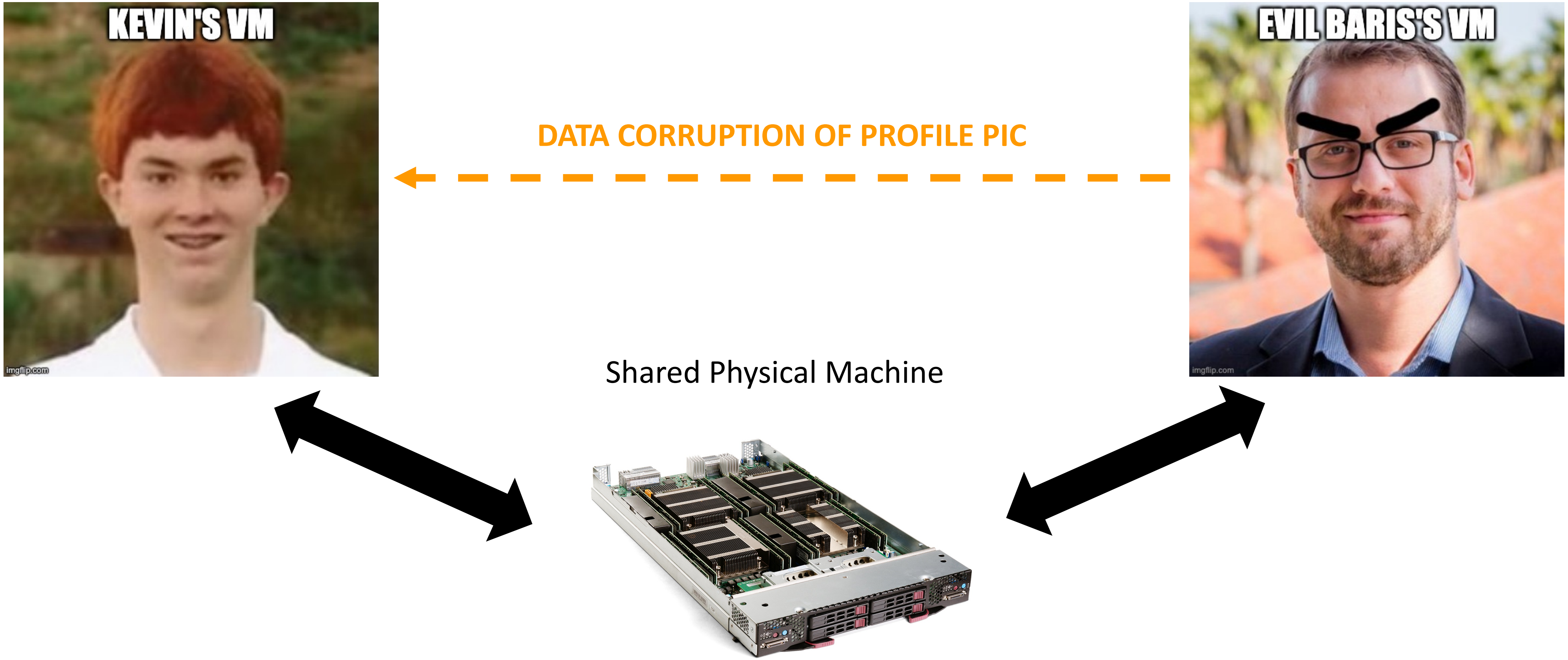
Increasing Susceptibility Risks More Bit Flips



DATA CORRUPTION OF PROFILE PIC

Shared Physical Machine

We All Definitely Want to Prevent the Worst Case



DATA CORRUPTION OF PROFILE PIC

Shared Physical Machine

⚠ Today’s Cloud DRAM Lacks Strong Isolation ⚠

- Motivation Tenants (VMs) can interfere with each other in DRAM
  - Leads to security problems (**Rowhammer**) and performance problems (contention)

- Key Contribution #1 Subarray Groups as DRAM Isolation Domains
  - Prevent inter-VM bit flips ***without sacrificing performance***

- Key Contribution #2 Siloz Hypervisor for Subarray Group Management
  - Provides **first step** towards practical management of DRAM as isolated domains

This Paper = Per-Tenant DRAM Isolation

Siloz Outline

- Background: What We Want, and Why We Don’t Have It
- Subarray Group Primitive
- Siloz Hypervisor
- Evaluation

Siloz Outline

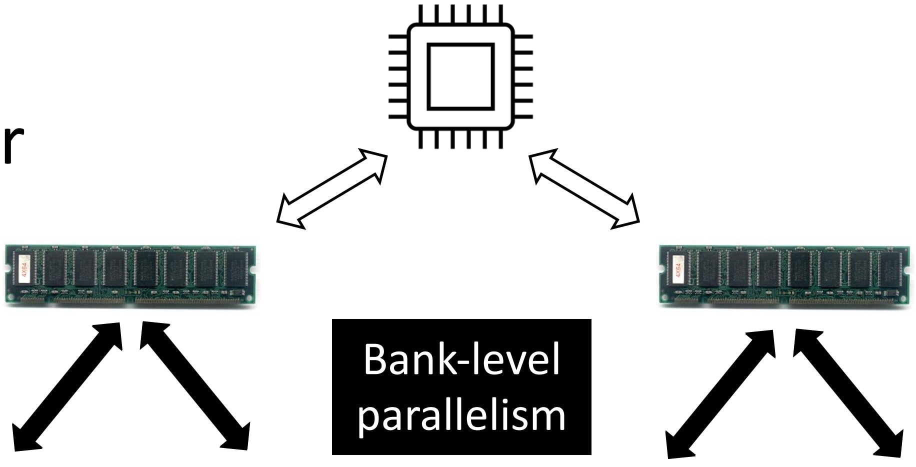
- **Background: What We Want, and Why We Don’t Have It**
- Subarray Group Primitive
- Siloz Hypervisor
- Evaluation

WHAT DO WE WANT?
HIGH-PERFORMANCE DRAM ISOLATION DOMAINS!
WHEN DO WE WANT THEM?
YESTERDAY!
imgflip.com

Problem: DRAM Performance Sacrifices Isolation

- DRAM architecture is a set of row-column banks
    - Different banks can be accessed in parallel
- Each page is interleaved across banks for performance of bank-level parallelism
- Downside: Rowhammer (RH) bit flips are possible between nearby rows in same bank

Page A
Page B



Activation rate above RH threshold → BIT FLIP(S) in nearby row(s) →

DRAM Bank 0

	C0	C1	C2	C3
R0	0	1	1	0
R1	1	1	0	1
R2	1	0	0	1
R3	0	1	0	0

DRAM Bank 1

	C0	C1	C2	C3
R0	0	1	1	0
R1	1	1	0	1
R2	1	1	1	1
R3	0	1	0	0

DRAM Bank 2

	C0	C1	C2	C3
R0	1	1	1	0
R1	1	0	1	1
R2	1	0	1	1
R3	0	0	0	1

DRAM Bank 3

	C0	C1	C2	C3
R0	0	1	1	1
R1	0	1	0	1
R2	0	0	1	0
R3	0	1	1	0

Our Primitive Must Have Two Properties

1. Allows page interleaving across banks **(performance)**

DRAM Bank 0

	C0	C1	C2	C3
R0	0	1	1	0
R1	1	1	0	1
R2	1	0	1	1
R3	0	1	0	0

DRAM Bank 1

	C0	C1	C2	C3
R0	0	1	1	0
R1	1	1	0	1
R2	1	0	1	1
R3	0	1	0	0

VM A’s Pages

VM B’s Pages

2. Isolates different VMs without wasting DRAM **(security)**

DRAM Bank 0

	C0	C1	C2	C3
R0	0	1	1	0
R1	1	1	0	1
???				
R2	1	0	1	1
R3	0	1	0	0

DRAM Bank 1

	C0	C1	C2	C3
R0	0	1	1	0
R1	1	1	0	1
???				
R2	1	0	1	1
R3	0	1	0	0

Siloz Outline

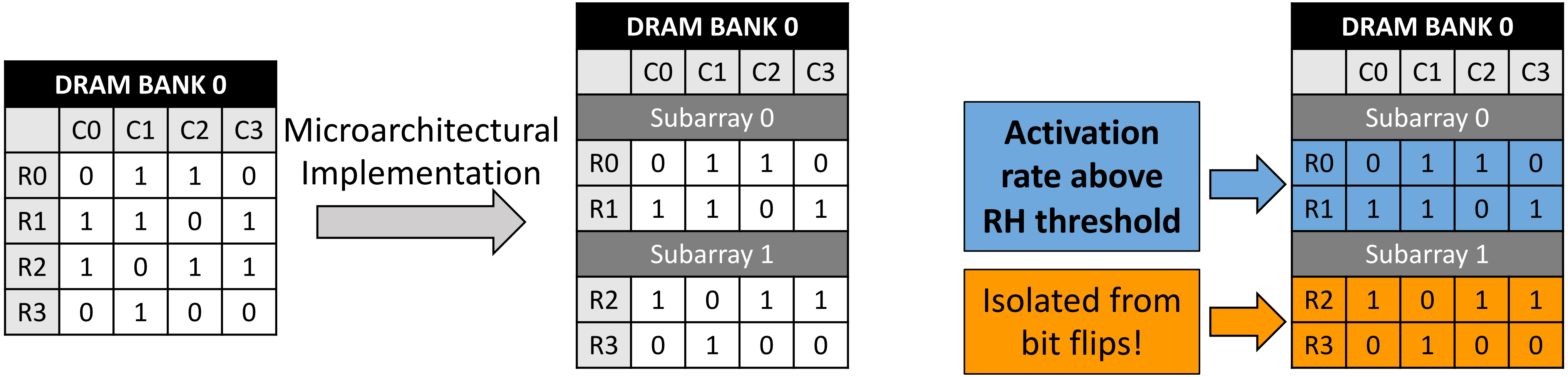
- Background: What We Want, and Why We Don’t Have It
- **Subarray Group Primitive**
- Siloz Hypervisor
- Evaluation

Bank Microarch is a **Set** of Row-Column **Subarrays**

- Subarrays are not directly-exposed, but visible with reverse engineering
- Subarrays provide Rowhammer isolation [mFIT 2021]
  - Each subarray is physically-separated by I/O circuitry

VM A's Pages

VM B's Pages



DRAM BANK 0

	C0	C1	C2	C3
R0	0	1	1	0
R1	1	1	0	1
R2	1	0	1	1
R3	0	1	0	0

Microarchitectural Implementation

DRAM BANK 0

	C0	C1	C2	C3
Subarray 0				
R0	0	1	1	0
R1	1	1	0	1
Subarray 1				
R2	1	0	1	1
R3	0	1	0	0

Activation rate above RH threshold

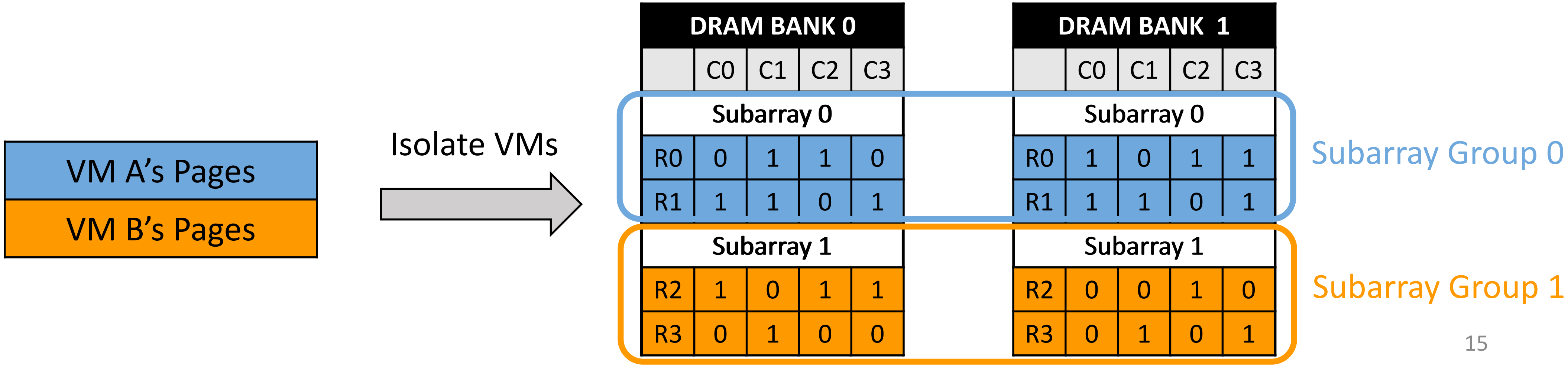
Isolated from bit flips!

DRAM BANK 0

	C0	C1	C2	C3
Subarray 0				
R0	0	1	1	0
R1	1	1	0	1
Subarray 1				
R2	1	0	1	1
R3	0	1	0	0

Siloz Insight: Provide Isolation via Subarray *Groups*

- A subarray group is comprised of a subarray from each bank
- Security benefit: subarray groups still provide subarray-level isolation
- Performance benefit: subarray groups preserve bank-level parallelism



DRAM BANK 0

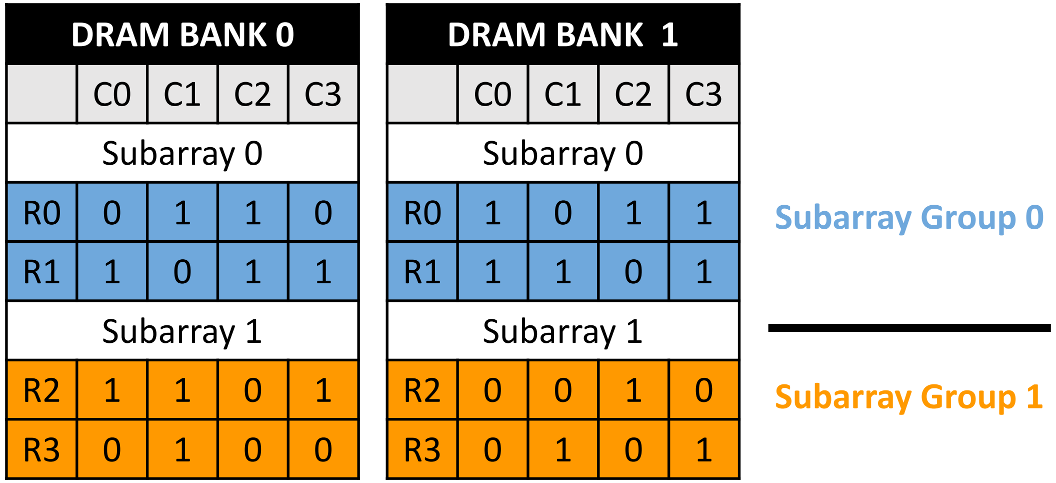
	C0	C1	C2	C3
Subarray 0				
R0	0	1	1	0
R1	1	1	0	1
Subarray 1				
R2	1	0	1	1
R3	0	1	0	0

DRAM BANK 1

	C0	C1	C2	C3
Subarray 0				
R0	1	0	1	1
R1	1	1	0	1
Subarray 1				
R2	0	0	1	0
R3	0	1	0	1

Factors Affecting Subarray Group Size

- Subarray group size is the product of 3 system factors
  - Number of interleaved banks (ex: **192**)
  - Rows per subarray (ex: **1024**)
  - Row size (ex: **8 KiB**)
  - **192 * 1024 * 8 KiB** = 1.5 GiB Subarray Group



DRAM BANK 0				
	C0	C1	C2	C3
Subarray 0				
R0	0	1	1	0
R1	1	0	1	1
Subarray 1				
R2	1	1	0	1
R3	0	1	0	0

DRAM BANK 1				
	C0	C1	C2	C3
Subarray 0				
R0	1	0	1	1
R1	1	1	0	1
Subarray 1				
R2	0	0	1	0
R3	0	1	0	1

Subarray Group 0

Subarray Group 1

- Finer-grained subarray group sizes are possible (see paper)

Siloz Outline

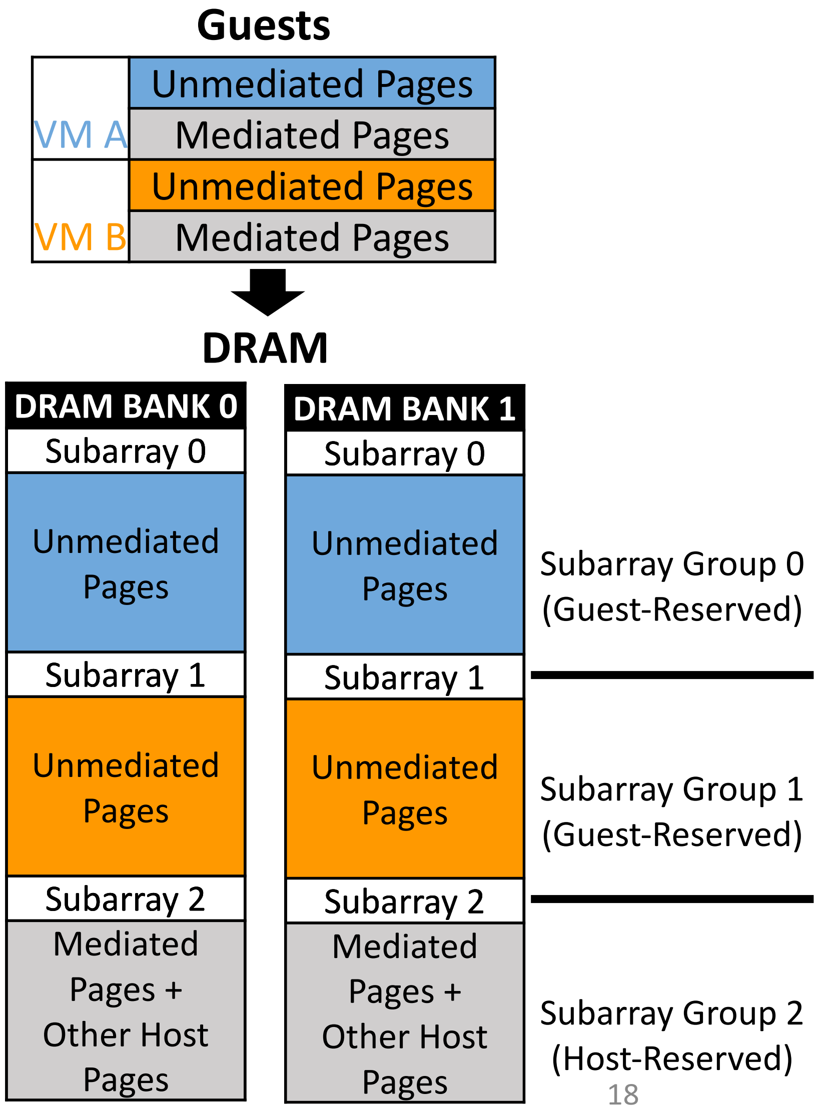
- Background: What We Want, and Why We Don’t Have It
- Subarray Group Primitive
- **Siloz Hypervisor Design**
- Evaluation

Siloz Places VMs in Private Subarray Groups

- Siloz places guest pages according to pre-existing **mediation status**

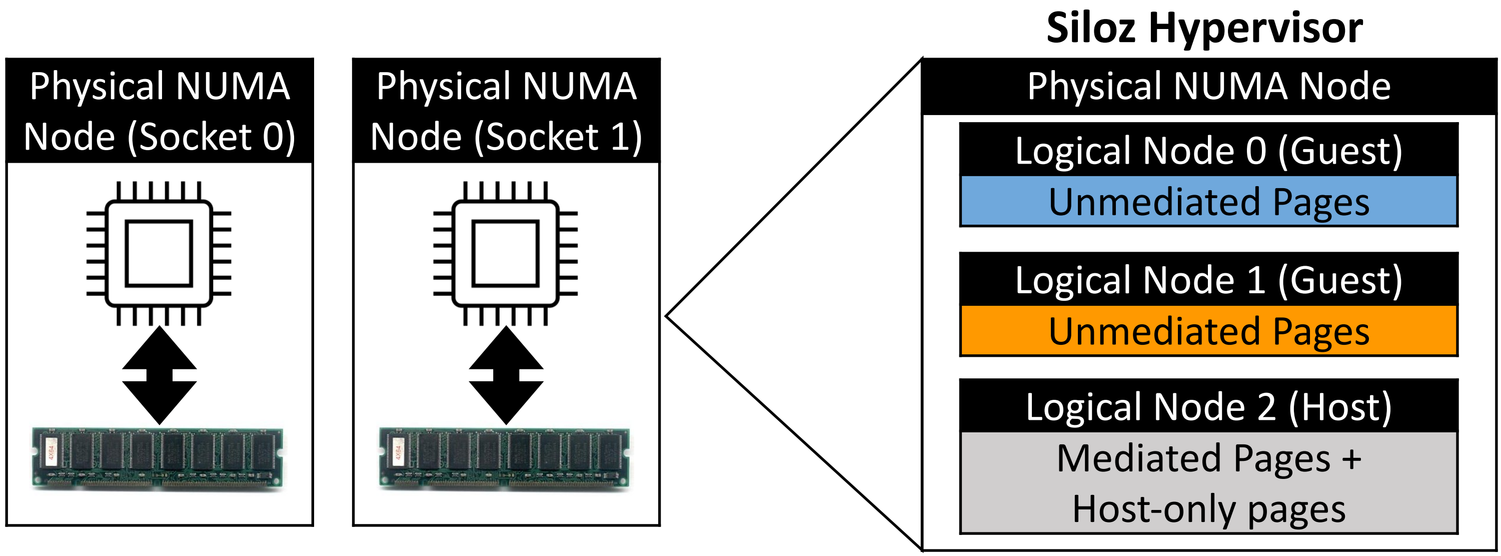
- Unmediated page: guest can access without host intervention
  - Guest can trivially-hammer unmediated pages

- Mediated page: traps to host on all accesses
  - Host can trivially rate-limit attempted hammering



Provisioning Private Subarray Groups via NUMA

- Requirement: Siloz must manage DRAM as subarray group partitions
- Existing NUMA support already provides DRAM partition management!
- Siloz extends **physical** NUMA node support (socket-level) to manage DRAM as **logical** NUMA nodes (subarray groups)



Physical NUMA Node (Socket 0)

Physical NUMA Node (Socket 1)

Siloz Hypervisor

Physical NUMA Node

Logical Node 0 (Guest)

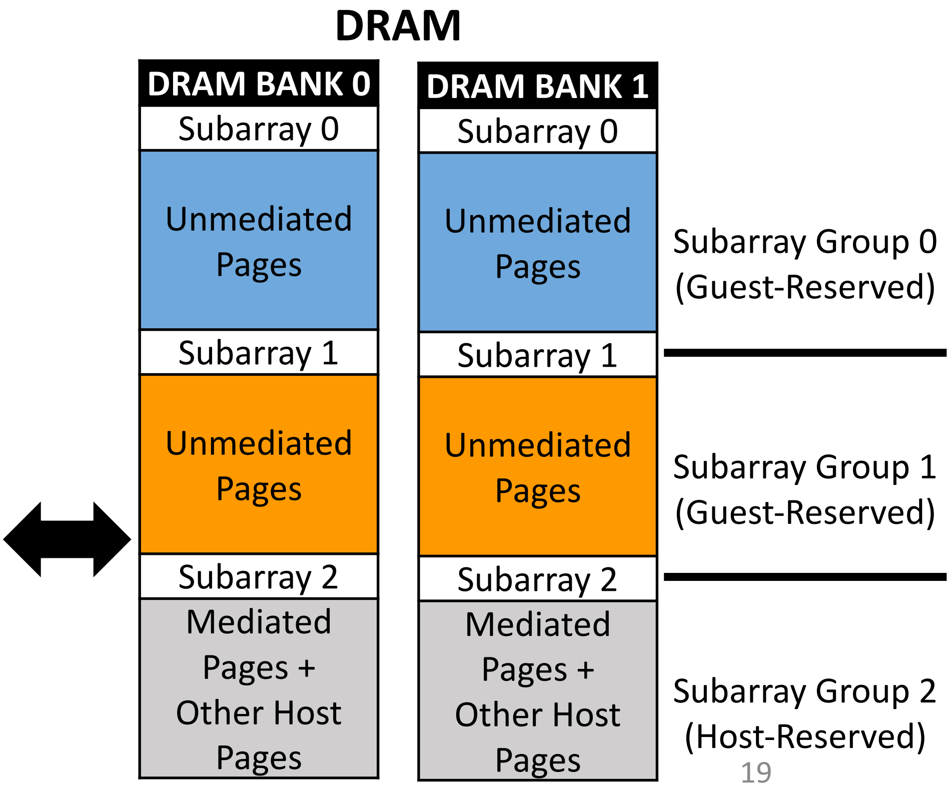
Unmediated Pages

Logical Node 1 (Guest)

Unmediated Pages

Logical Node 2 (Host)

Mediated Pages + Host-only pages



DRAM

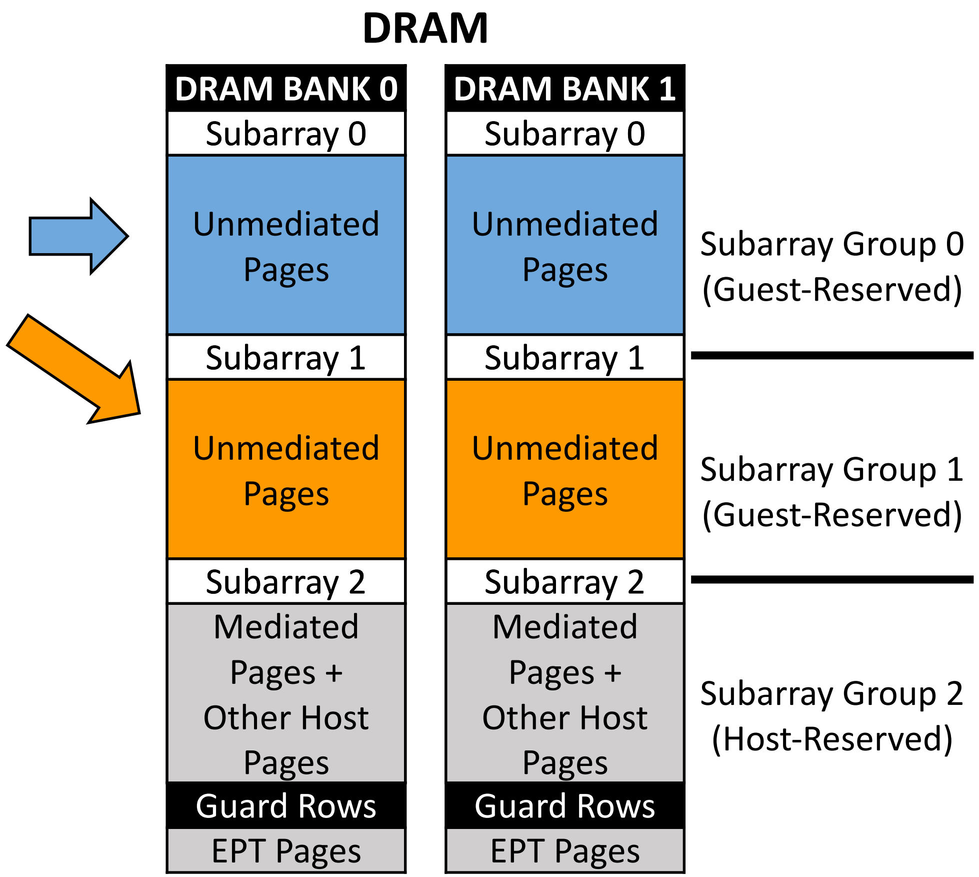
DRAM BANK 0	DRAM BANK 1	
Subarray 0	Subarray 0	
Unmediated Pages	Unmediated Pages	Subarray Group 0 (Guest-Reserved)
Subarray 1	Subarray 1	
Unmediated Pages	Unmediated Pages	Subarray Group 1 (Guest-Reserved)
Subarray 2	Subarray 2	
Mediated Pages + Other Host Pages	Mediated Pages + Other Host Pages	Subarray Group 2 (Host-Reserved)

Is Isolation Enough for Extended Page Tables (EPTs)?

- Subarray groups contain, ***but do not prevent,*** bit flips
- EPTs store subarray group mapping
  - **Bit flips in EPTs could compromise subarray group isolation!!!**
- Siloz prevents EPT bit flips via **guard rows**
  - Guard rows work for small quantities of data
  - EPTs + guard rows: 0.024% of DRAM

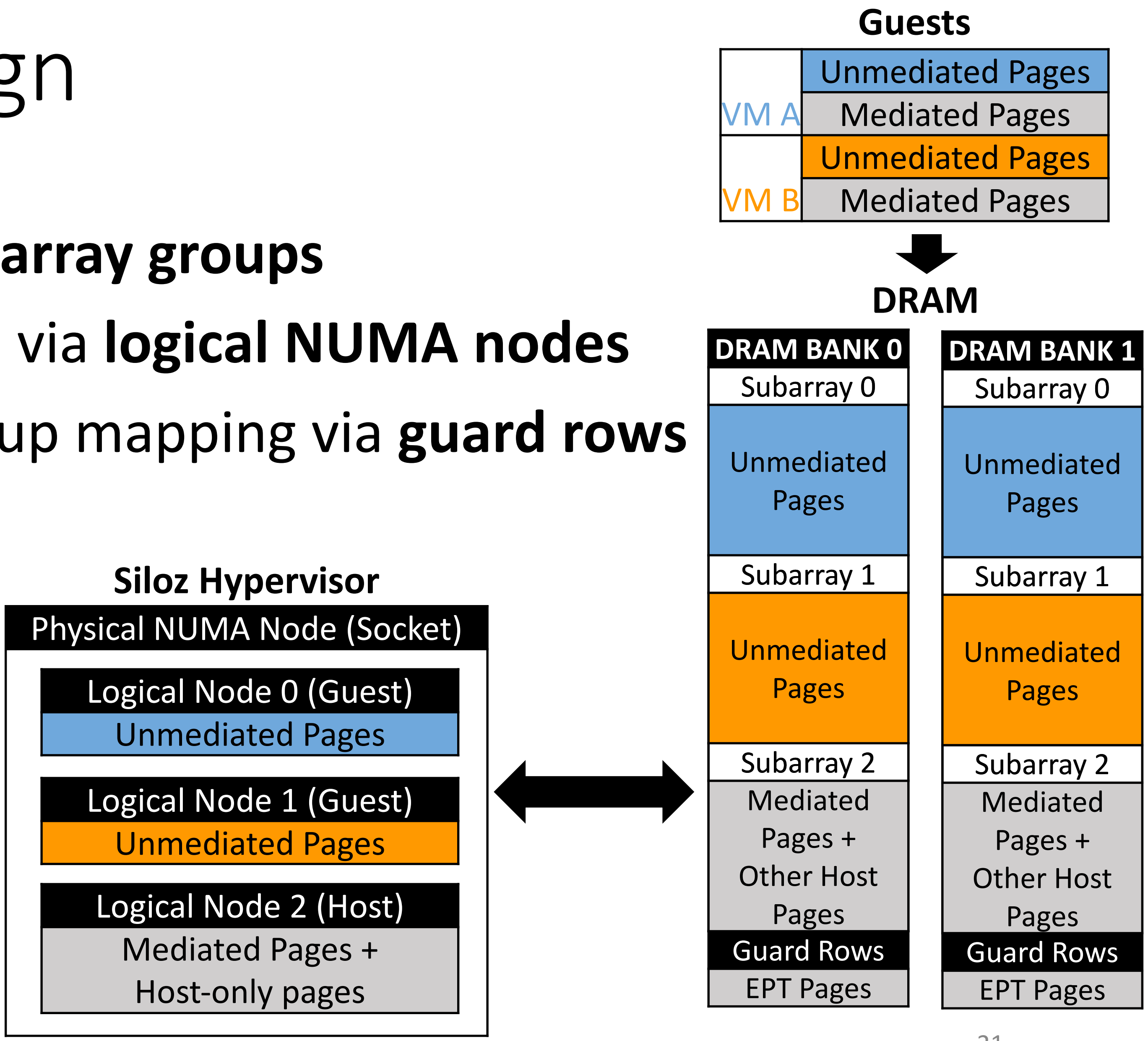
Extended Page Table Entry (Corrupted)

Guest Address → Host Address ~~X~~ Y



Recapping Siloz Design

- Siloz places VMs in private **subarray groups**
- Siloz manages subarray groups via **logical NUMA nodes**
- Siloz protects the subarray group mapping via **guard rows**



Guests

VM A: Unmediated Pages, Mediated Pages

VM B: Unmediated Pages, Mediated Pages

DRAM

DRAM BANK 0: Subarray 0 – Unmediated Pages; Subarray 1 – Unmediated Pages; Subarray 2 – Mediated Pages + Other Host Pages; Guard Rows; EPT Pages

DRAM BANK 1: Subarray 0 – Unmediated Pages; Subarray 1 – Unmediated Pages; Subarray 2 – Mediated Pages + Other Host Pages; Guard Rows; EPT Pages

Siloz Hypervisor

Physical NUMA Node (Socket)

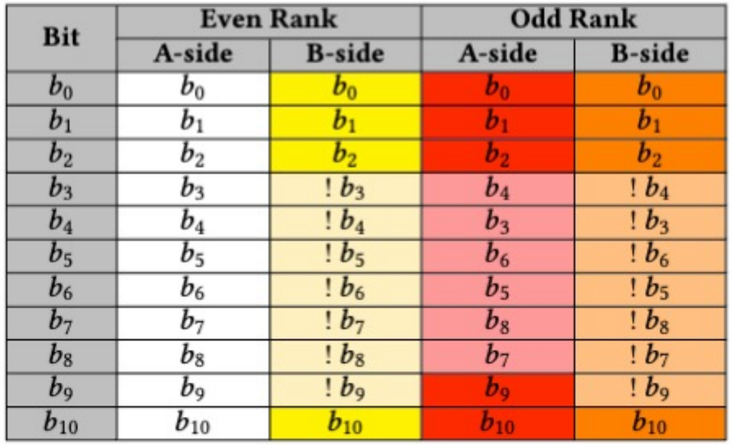
Logical Node 0 (Guest): Unmediated Pages

Logical Node 1 (Guest): Unmediated Pages

Logical Node 2 (Host): Mediated Pages + Host-only pages

Accounting for DRAM-Internal Remaps

- DIMMs can internally remap rows
  - Risks violating subarray group isolation!
  - See paper for how we handle remaps



Bit	Even Rank		Odd Rank	
	A-side	B-side	A-side	B-side
b_0	b_0	b_0	b_0	b_0
b_1	b_1	b_1	b_1	b_1
b_2	b_2	b_2	b_2	b_2
b_3	b_3	$!\,b_3$	b_4	$!\,b_4$
b_4	b_4	$!\,b_4$	b_3	$!\,b_3$
b_5	b_5	$!\,b_5$	b_6	$!\,b_6$
b_6	b_6	$!\,b_6$	b_5	$!\,b_5$
b_7	b_7	$!\,b_7$	b_8	$!\,b_8$
b_8	b_8	$!\,b_8$	b_7	$!\,b_7$
b_9	b_9	$!\,b_9$	b_9	$!\,b_9$
b_{10}	b_{10}	b_{10}	b_{10}	b_{10}

Siloz Outline

- Background: What We Want, and Why We Don’t Have It
- Subarray Group Primitive
- Siloz Hypervisor
- **Evaluation**

Siloz Evaluation Methodology

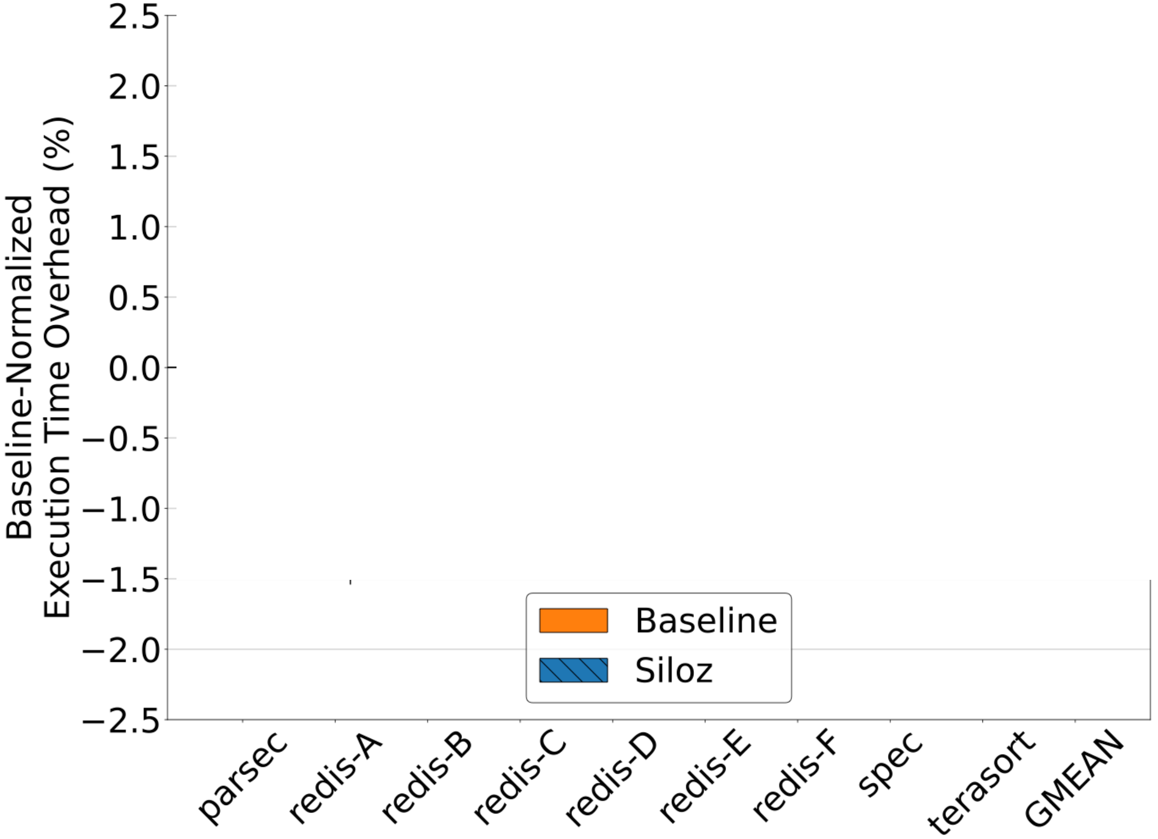
- Siloz is evaluated against Ubuntu 22.04 LTS baseline

- Host is a major cloud provider Intel Skylake server configuration

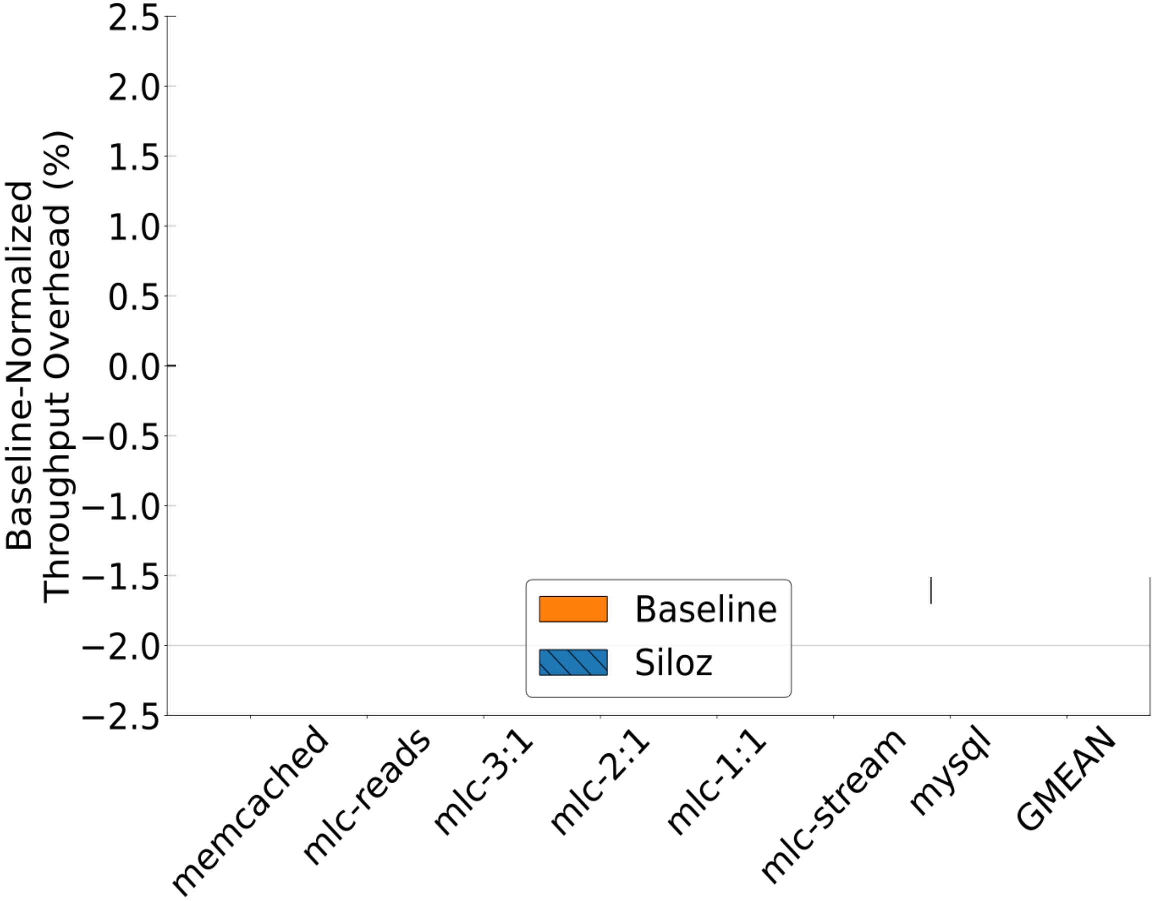
- Security verified via Rowhammer fuzzer [Blacksmith, 2022]
    - Siloz contains bit flips to subarray groups + prevents EPT bit flips

- Performance effects measured across variety of benchmarks
    - Cloud workloads (ex: memcached)
    - Intel Memory Latency Checker (MLC)
    - SPEC CPU 2017 + PARSEC 3.0

Siloz's Effect on Execution Time is Negligible



Siloz's Effect on Throughput is Also Negligible



Siloz Recap

- **Objective:** prevent inter-VM hammering with negligible effect on performance
- **Approach:** isolate VMs to private subarray groups
- **Deliverable:** Linux/KVM implementation provides comprehensive protection within ±0.5% of baseline average performance
- **Broader Impact:** 1st step toward managing DRAM as set of isolated domains

Thank You!

Jonah Rosenblum

Stefan Saroiu

Alec Wolman

Dimitrios Skarlatos

Baris Kasikci

QUESTIONS?

Siloz: Leveraging DRAM Isolation Domains to Prevent Inter-VM Rowhammer

Kevin Loughlin

Jonah Rosenblum, Stefan Saroiu, Alec Wolman, Dimitrios Skarlatos, Baris Kasikci

COMPUTER SCIENCE AND ENGINEERING
UNIVERSITY of MICHIGAN

Microsoft

Carnegie Mellon University
Computer Science Department

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING