MOESI-prime: Preventing *Coherence-Induced Hammering* in Commodity Workloads

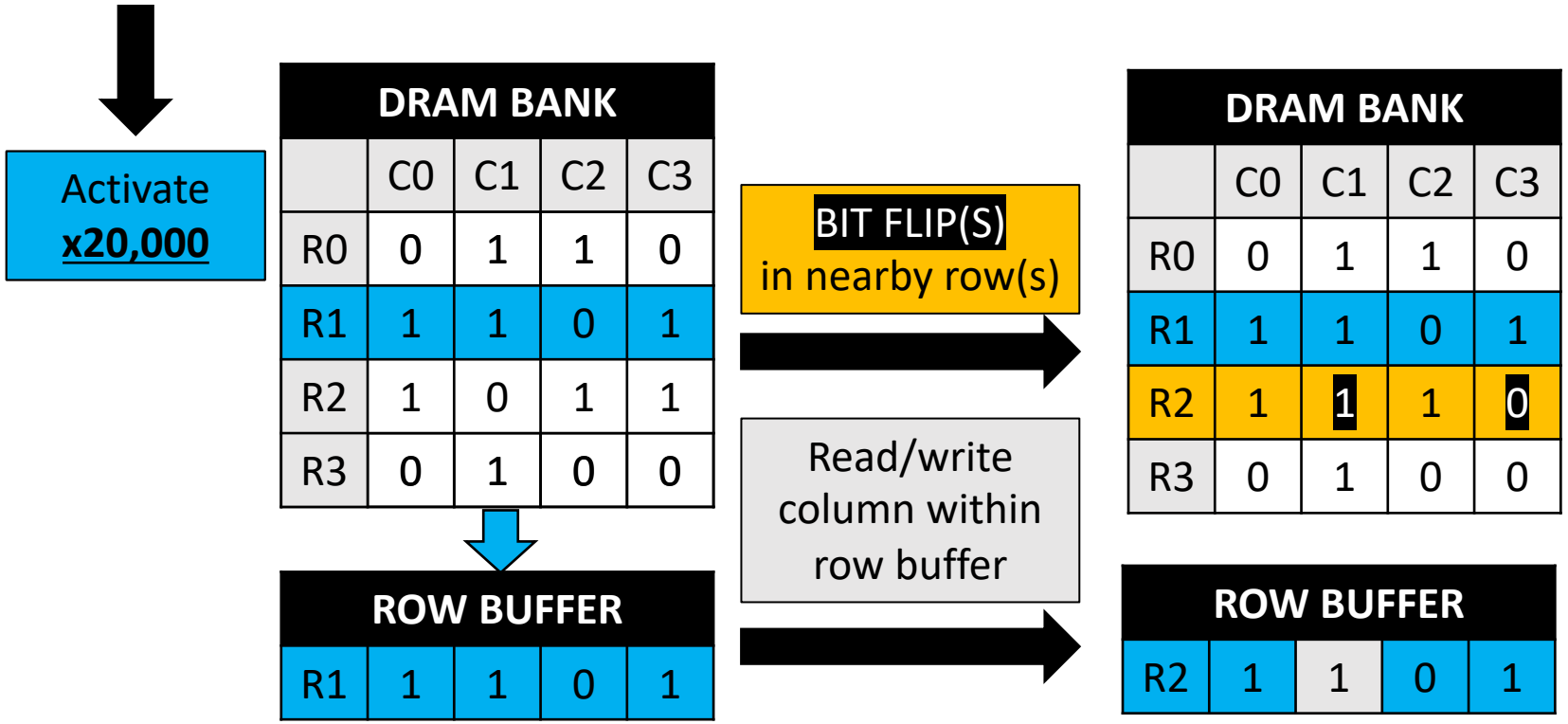
Kevin Loughlin
Stefan Saroiu, Alec Wolman, Yatin A. Manerkar, Baris Kasikci

COMPUTER SCIENCE AND ENGINEERING
UNIVERSITY of MICHIGAN

Microsoft

What is Rowhammer (RH)?

- Frequent ACTs of same DRAM row(s) can corrupt data in nearby rows
- ACT rate above **RH threshold** (ex: 20,000 ACTs/64 ms) can flip bits



Commodity Workloads: Dangerous ACT Rates

- Motivation Decreasing RH thresholds (fewer ACTs needed to flip bits)
  - Carefully-crafted, ***malicious*** code known to pose increasing danger
- Key Contribution #1 Coherence-induced hammering
  - Common, ***non-malicious*** code can also yield dangerous ACT rates
- Key Contribution #2 MOESI-prime coherence protocol
  - Mitigates coherence-induced hammering

Outline

- Background: Rowhammer, ccNUMA
- Problem: Coherence-Induced Hammering
- Mitigation: MOESI-prime
- Evaluation and Takeaways

Malicious Hammering

- Ex: repeatedly flush cache line in aggressor row to force DRAM accesses

```
1  while(true)
2      flush(row1_addr)
3      read(row1_addr)
```

Aggressor Row

Victim Row

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

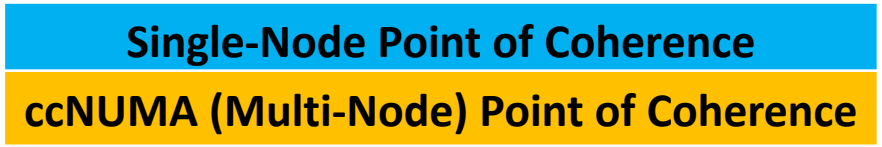
BIT FLIP(S) in victim row

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

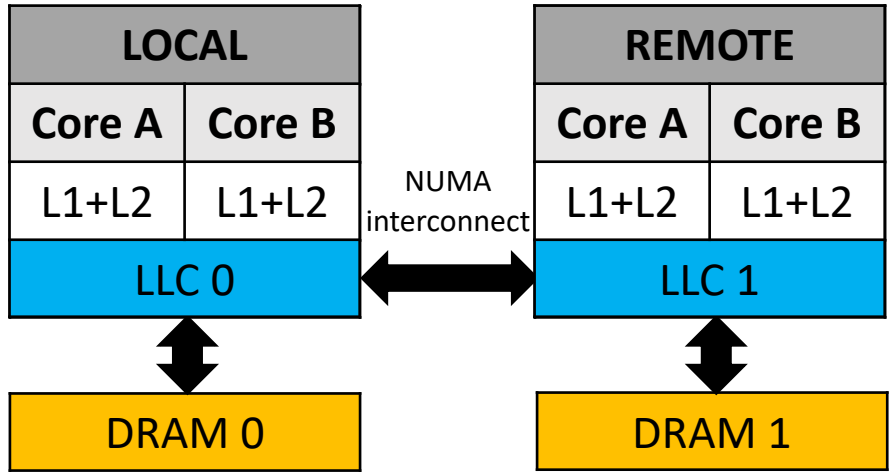
- Worst-case: every DRAM access requires row ACT
  - Additional techniques/conditions increase likelihood of row ACT

ccNUMA Can Change DRAM Access Frequency

- ccNUMA: cache coherency across multiple nodes (ex: sockets)
  - Each cache line has a single “local” node

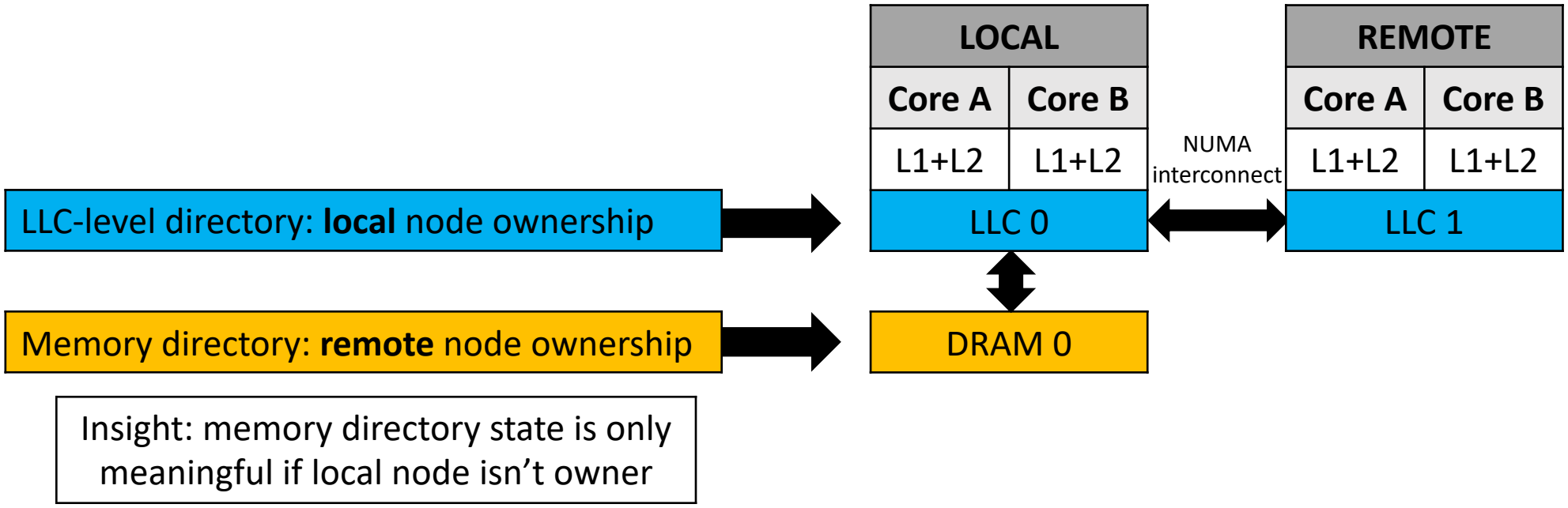


- **Remote** LLC miss: go to local node
- **Local** LLC miss: check ***memory directory***…



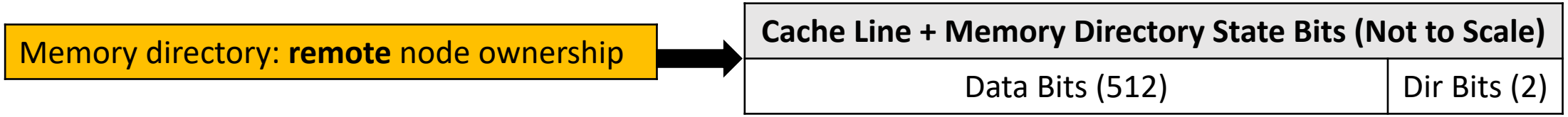
State-of-the-Art ccNUMA: Multiple Directories

- Directories track cache line ownership across cores
  - Ex: is a core's copy of a cache line **M**odified, **I**nvalid, etc.?
- Separate directories track local/remote ownership



Memory Directory Implementation

- Each cache line's remote state co-located with line in DRAM



- For today, two important memory directory states…
  - **A:** snoop-**A**ll: line *might* be owned (dirty) on a remote node
  - **I:** remote-**I**nvalid: line not valid on any remote node

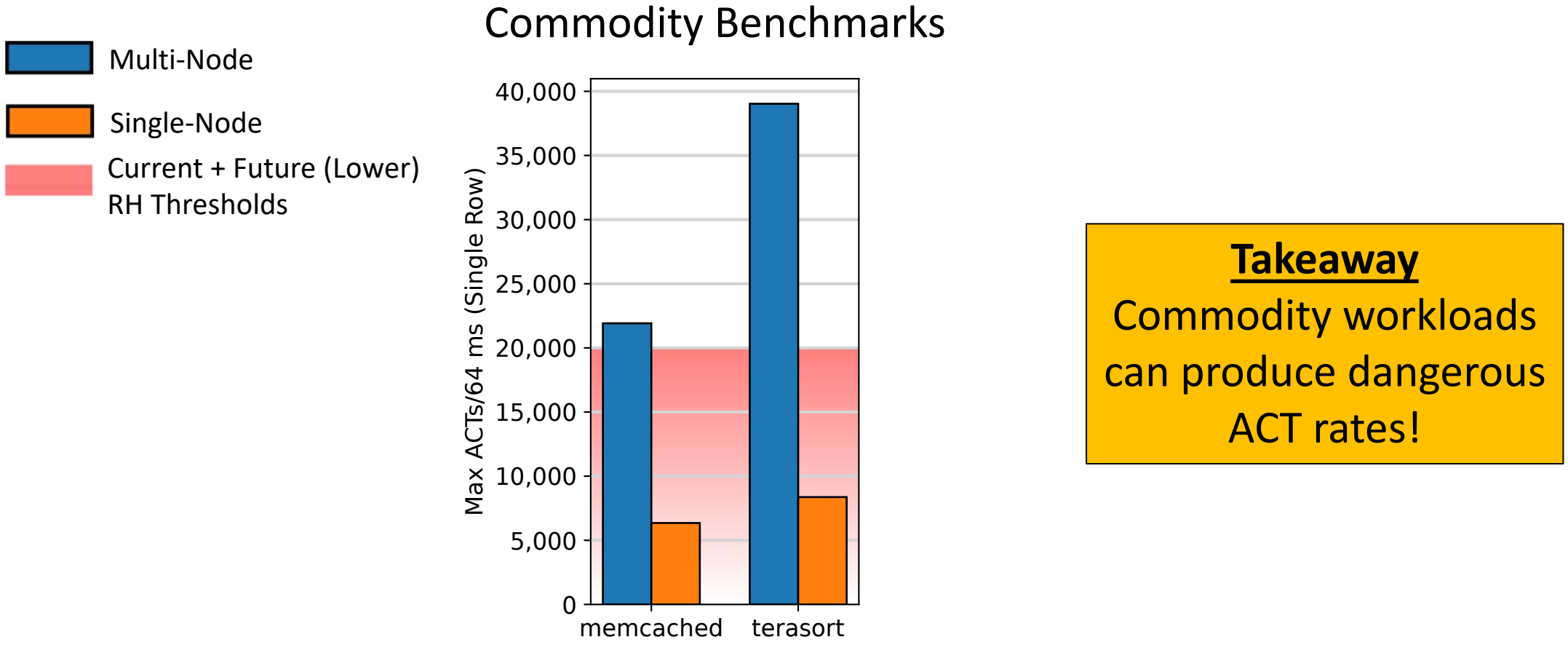
Outline

- Background: Rowhammer, ccNUMA
- **Problem: Coherence-Induced Hammering**
- Mitigation: MOESI-prime
- Evaluation and Takeaways

Identifying Coherence-Induced Hammering

- Platform: Intel dual-socket Skylake server (ccNUMA)
  - Used DDR4 bus analyzer to record memory traces
- Ran commodity workloads on single node and two nodes
  - Measured highest ACT rate observed for single row within 64 ms (DDR4)
  - Compared to RH threshold of 20,000 ACTs
- Ran additional micro-benchmarks to isolate hammering sources
  - See paper

ccNUMA Increases Highest Row ACT Rates



Takeaway

Commodity workloads can produce dangerous ACT rates!

Common Across Benchmarks: Dirty Sharing

- Dirty sharing: cache line sharing with at least 1 writer
- Consider migratory sharing of lock-protected data

Migratory sharing occurs in commodity code!

Thread A	
1	while(true)
2	acquire_lock()
3	write(shared_var)
4	release_lock()

Thread B	
1	while(true)
2	acquire_lock()
3	write(shared_var)
4	release_lock()

Sources of **Coherence-Induced Hammering**

- Problem #1: Redundant Memory Directory Writes TODAY!
- Problem #2: Mis-Speculated DRAM Reads
- Problem #3: Downgrade Writebacks

Hammering Writes: Migratory Sharing

Nodes	Local Writer	Remote Writer
LLC Dir State	Invalid	Invalid
Mem Dir State	remote-Invalid	

Remote write

Nodes	Local Writer	Remote Writer
LLC Dir State	Invalid	Modified
Mem Dir State	snoop-All	

Local write

Local Writer	Remote Writer
Modified	Invalid
snoop-All (local owner, N/A)	

Local write

Remote write

Local Writer	Remote Writer
Invalid	Modified
snoop-All (***redundant*** write)	

WHY REDUNDANT?
No processor state indicates memory directory is already in snoop-All

Outline

- Background: Rowhammer, ccNUMA
- Problem: Coherence-Induced Hammering
- **Mitigation: MOESI-prime**
- Evaluation and Takeaways

“Prime” States to Avoid Redundant Writes

- Problem: processor can’t recognize memory directory is already **snoop-All**
  - snoop-All: cache line might be dirty on a remote node
- Fix: for “conventional” dirty processor coherence states, add “prime” states
  - Prime means memory directory in **snoop-All**, otherwise equivalent to conventional

Modified - dirty + read-write - Mem dir state unknown	+	**M**odified-**prime** - dirty + read-write - Mem dir state = **snoop-All**

- Similarity of conventional and prime states helps preserve correctness

MOESI-prime in Action

Modified-**prime**
- dirty + read-write
- Mem dir state = **snoop-All**

Nodes	Local Writer	Remote Writer
LLC Dir State	Invalid	Invalid
Mem Dir State	remote-Invalid	

Remote write

Nodes	Local Writer	Remote Writer
LLC Dir State	Invalid	Modified-prime
Mem Dir State	snoop-All	

Local write

Local Writer	Remote Writer
Modified-prime	Invalid
snoop-All (local owner, N/A)	

Local write

Remote write

Local Writer	Remote Writer
Invalid	Modified-prime
snoop-All **(write omitted)**	

Outline

- Background: Rowhammer, ccNUMA
- Problem: Coherence-Induced Hammering
- Mitigation: MOESI-prime
- **Evaluation and Takeaways**

Evaluation

- Gem5 configurations modelled after major cloud provider's settings
  - Compared MOESI-prime to MESI and MOESI baseline protocols
- Micro-benchmarks: MOESI-prime prevents coherence-induced hammering
- Commodity benchmarks: PARSEC-3.0 and SPLASH-2x
  - Many workloads exhibit >20,000 ACTs/64 ms to single row in baseline protocols

(2-nodes) Average Metrics Normalized to MESI Baseline (Higher is Better)		
Metric	MOESI	**MOESI-prime**
Decrease in Max ACTs	+5.58%	**+77.38%**
Exec Time	+0.61%	**+0.48%**
DRAM Power	0.00%	**+0.22%**

Takeaway

MOESI-prime mitigates coherence-induced hammering, and can even slightly improve performance and power!

Recap

- Key Contribution #1: Coherence-induced hammering
  - Commodity workloads can yield dangerous ACT rates
- Key Contribution #2: MOESI-prime coherence protocol
  - Mitigates coherence-induced hammering
- Check out the paper for much more!
  - Ex: other sources of coherence-induced hammering, proof of correctness

Thanks to my awesome collaborators!

Stefan Saroiu

Alec Wolman

Yatin A. Manerkar

Baris Kasikci

Thanks for listening! Questions?

Scan for Full Paper