Scalable Many-Core Memory Systems Topic 2: Emerging Technologies and Hybrid Memories

Prof. Onur Mutlu
http://www.ece.cmu.edu/~omutlu
onur@cmu.edu

HiPEAC ACACES Summer School 2013
July 15-19, 2013

Carnegie Mellon

Requirements from an Ideal Memory System

- Traditional
  - Enough capacity
  - Low cost
  - High system performance (high bandwidth, low latency)

- New
  - Technology scalability: lower cost, higher capacity, lower energy
  - Energy (and power) efficiency
  - QoS support and configurability (for consolidation)

Requirements from an Ideal Memory System

- Traditional
  - Higher capacity
  - Continuous low cost
  - High system performance (higher bandwidth, low latency)

- New
  - Technology scalability: lower cost, higher capacity, lower energy
  - Energy (and power) efficiency
  - QoS support and configurability (for consolidation)

Emerging, resistive memory technologies (NVM) can help

Agenda

- Major Trends Affecting Main Memory
- Requirements from an Ideal Main Memory System
- Opportunity: Emerging Memory Technologies
- Conclusions
- Discussion

The Promise of Emerging Technologies

- Likely need to replace/augment DRAM with a technology that is
  - Technology scalable
  - And at least similarly efficient, high performance, and fault-tolerant
    - or can be architected to be so

- Some emerging resistive memory technologies appear promising
  - Phase Change Memory (PCM)?
  - Spin Torque Transfer Magnetic Memory (STT-MRAM)?
  - Memristors?
  - And, maybe there are other ones
  - Can they be enabled to replace/augment/surpass DRAM?

Agenda

- Major Trends Affecting Main Memory
- Requirements from an Ideal Main Memory System
- Opportunity: Emerging Memory Technologies
  - Background
  - PCM (or Technology X) as DRAM Replacement
  - Hybrid Memory Systems
- Conclusions
- Discussion

Charge vs. Resistive Memories

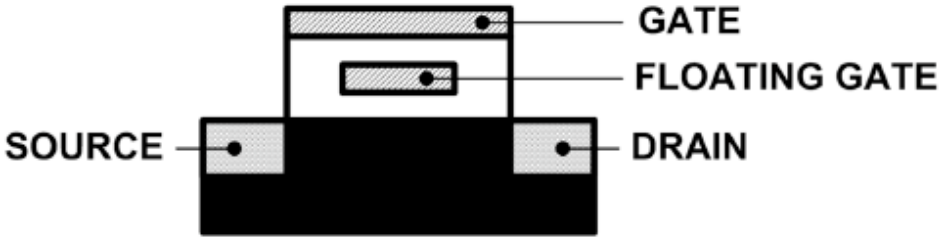
- Charge Memory (e.g., DRAM, Flash)
  - Write data by capturing charge Q
  - Read data by detecting voltage V

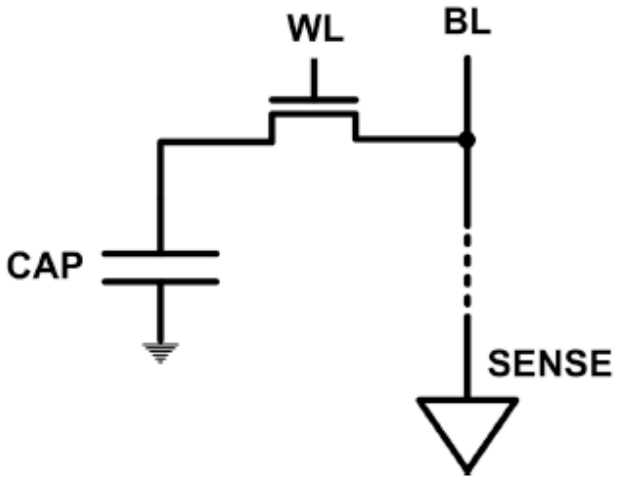
- Resistive Memory (e.g., PCM, STT-MRAM, memristors)
  - Write data by pulsing current dQ/dt
  - Read data by detecting resistance R

Limits of Charge Memory

- Difficult charge placement and control
  - Flash: floating gate charge
  - DRAM: capacitor charge, transistor leakage

- Reliable sensing becomes difficult as charge storage unit size reduces



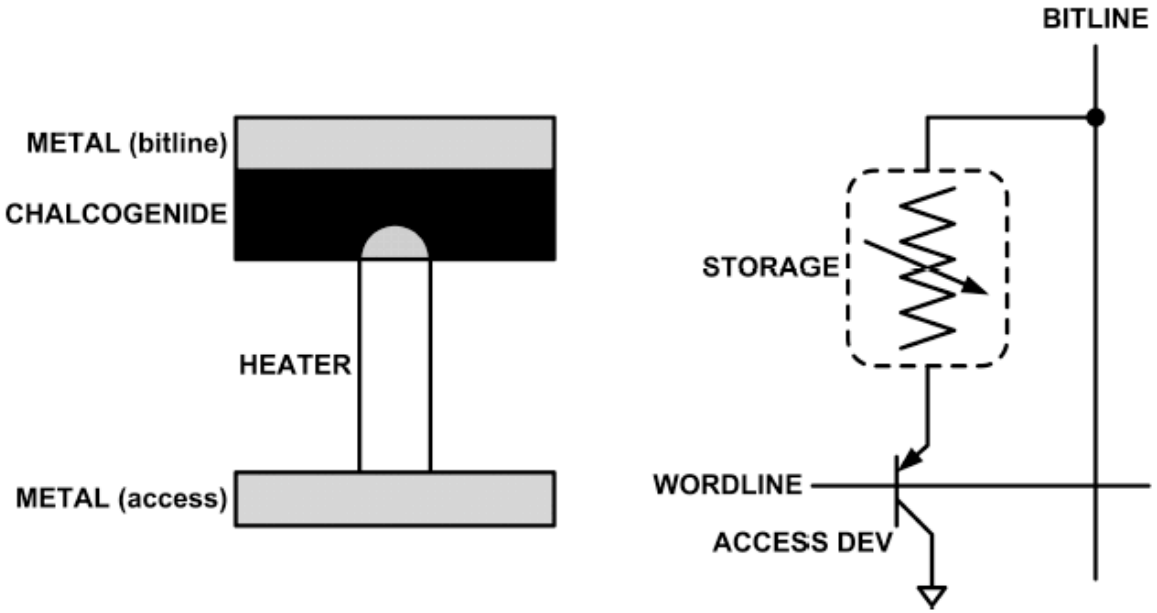


Emerging Resistive Memory Technologies

- PCM
  - Inject current to change material phase
  - Resistance determined by phase
- STT-MRAM
  - Inject current to change magnet polarity
  - Resistance determined by polarity
- Memristors
  - Inject current to change atomic structure
  - Resistance determined by atom distance

What is Phase Change Memory?

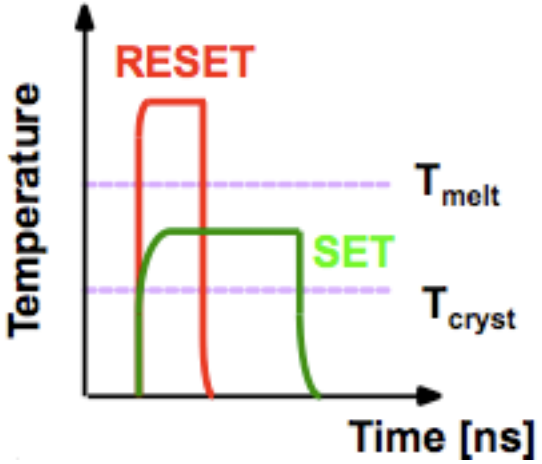
- Phase change material (chalcogenide glass) exists in two states:
  - Amorphous: Low optical reflexivity and high electrical resistivity
  - Crystalline: High optical reflexivity and low electrical resistivity

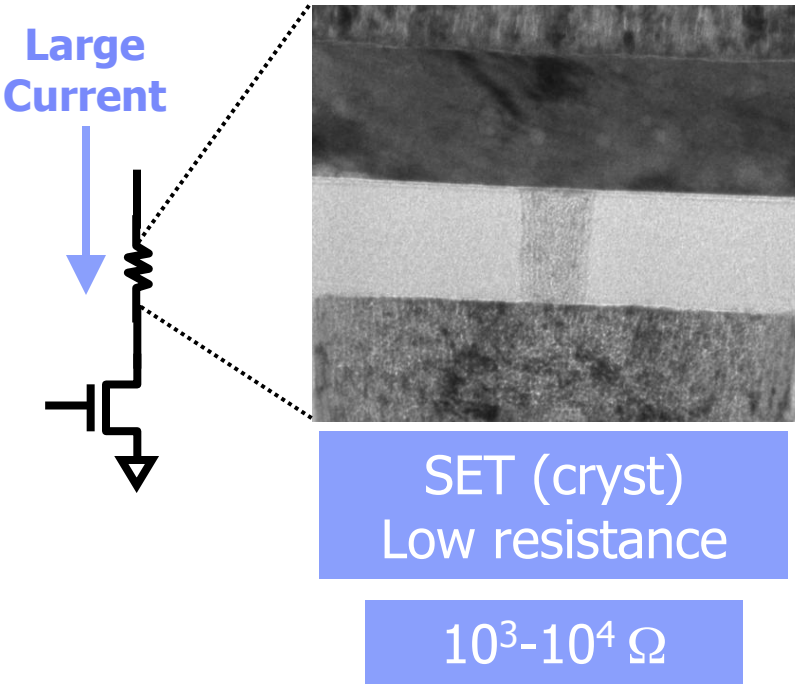


PCM is resistive memory: High resistance (0), Low resistance (1)
PCM cell can be switched between states reliably and quickly

How Does PCM Work?

- Write: change phase via current injection
  - SET: sustained current to heat cell above T*cryst*
  - RESET: cell heated above T*melt* and quenched
- Read: detect phase via material resistance
  - amorphous/crystalline





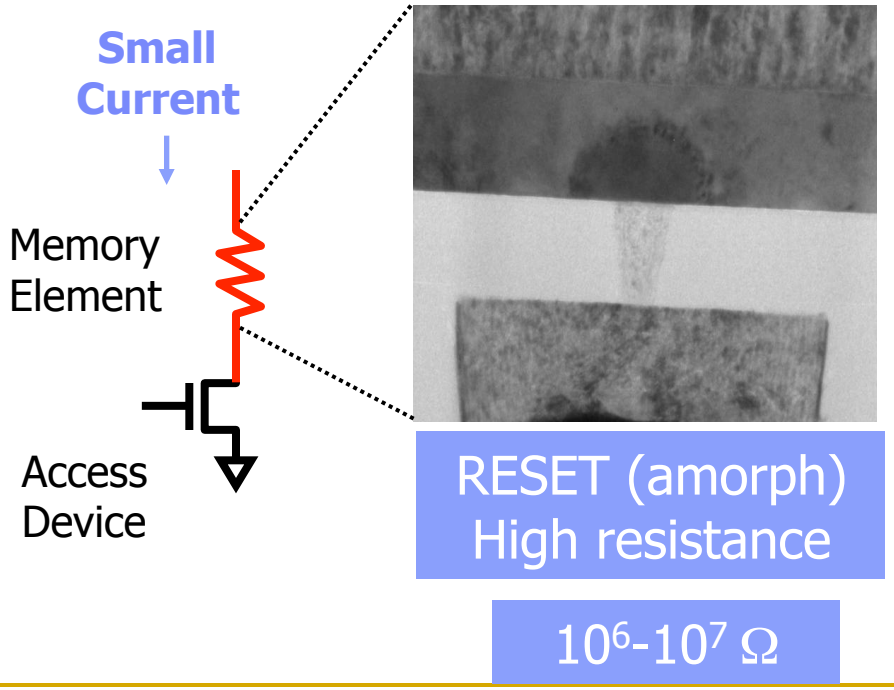


Photo Courtesy: Bipin Rajendran, IBM Slide Courtesy: Moinuddin Qureshi, IBM

Opportunity: PCM Advantages

- Scales better than DRAM, Flash
  - Requires current pulses, which scale linearly with feature size
  - Expected to scale to 9nm (2022 [ITRS])
  - Prototyped at 20nm (Raoux+, IBM JRD 2008)
- Can be denser than DRAM
  - Can store multiple bits per cell due to large resistance range
  - Prototypes with 2 bits/cell in ISSCC'08, 4 bits/cell by 2012
- Non-volatile
  - Retain data for >10 years at 85C
- No refresh needed, low idle power

Phase Change Memory Properties

- Surveyed prototypes from 2003-2008 (ITRS, IEDM, VLSI, ISSCC)
- Derived PCM parameters for F=90nm

- Lee, Ipek, Mutlu, Burger, "Architecting Phase Change Memory as a Scalable DRAM Alternative," ISCA 2009.

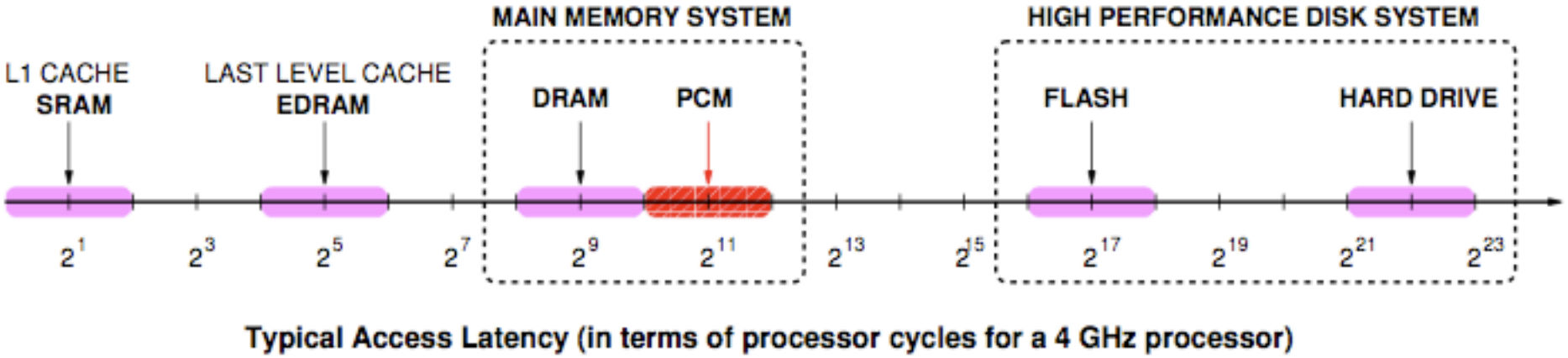
Table 1. Technology survey.

	Published prototype									
Parameter*	**Horri[6]**	**Ahn[12]**	**Bedeschi[13]**	**Oh[14]**	**Pellizer[15]**	**Chen[5]**	**Kang[16]**	**Bedeschi[9]**	**Lee[10]**	**Lee[2]**
Year	2003	2004	2004	2005	2006	2006	2006	2008	2008	**
Process, F (nm)	**	120	180	120	90	**	100	90	90	90
Array size (Mbytes)	**	64	8	64	**	**	256	256	512	**
Material	GST, N-d	GST, N-d	GST	GST	GST	GS, N-d	GST	GST	GST	GST, N-d
Cell size (μm^2)	**	0.290	0.290	**	0.097	60 nm^2	0.166	0.097	0.047	0.065 to 0.097
Cell size, F^2	**	20.1	9.0	**	12.0	**	16.6	12.0	5.8	9.0 to 12.0
Access device	**	**	BJT	FET	BJT	**	FET	BJT	Diode	BJT
Read time (ns)	**	70	48	68	**	**	62	**	55	48
Read current (μA)	**	**	40	**	**	**	**	**	**	40
Read voltage (V)	**	3.0	1.0	1.8	1.6	**	1.8	**	1.8	1.0
Read power (μW)	**	**	40	**	**	**	**	**	**	40
Read energy (pJ)	**	**	2.0	**	**	**	**	**	**	2.0
Set time (ns)	100	150	150	180	**	80	300	**	400	150
Set current (μA)	200	**	300	200	**	55	**	**	**	150
Set voltage (V)	**	**	2.0	**	**	1.25	**	**	**	1.2
Set power (μW)	**	**	300	**	**	34.4	**	**	**	90
Set energy (pJ)	**	**	45	**	**	2.8	**	**	**	13.5
Reset time (ns)	50	10	40	10	**	60	50	**	50	40
Reset current (μA)	600	600	600	600	400	90	600	300	600	300
Reset voltage (V)	**	**	2.7	**	1.8	1.6	**	1.6	**	1.6
Reset power (μW)	**	**	1620	**	**	80.4	**	**	**	480
Reset energy (pJ)	**	**	64.8	**	**	4.8	**	**	**	19.2
Write endurance (MLC)	10^7	10^9	10^6	**	10^8	10^4	**	10^5	10^5	10^8

* BJT: bipolar junction transistor; FET: field-effect transistor; GST: $Ge_2Sb_2Te_5$; MLC: multilevel cells; N-d: nitrogen doped.
** This information is not available in the publication cited.

Phase Change Memory Properties: Latency

- Latency comparable to, but slower than DRAM



Typical Access Latency (in terms of processor cycles for a 4 GHz processor)

- Read Latency
  - 50ns: 4x DRAM, 10^{-3}x NAND Flash
- Write Latency
  - 150ns: 12x DRAM
- Write Bandwidth
  - 5-10 MB/s: 0.1x DRAM, 1x NAND Flash

Phase Change Memory Properties

- Dynamic Energy
  - 40 uA Rd, 150 uA Wr
  - 2-43x DRAM, 1x NAND Flash
- Endurance
  - Writes induce phase change at 650C
  - Contacts degrade from thermal expansion/contraction
  - 10^8 writes per cell
  - 10^{-8}x DRAM, 10^3x NAND Flash
- Cell Size
  - 9-12F^2 using BJT, single-level cells
  - 1.5x DRAM, 2-3x NAND (will scale with feature size, MLC)

Phase Change Memory: Pros and Cons

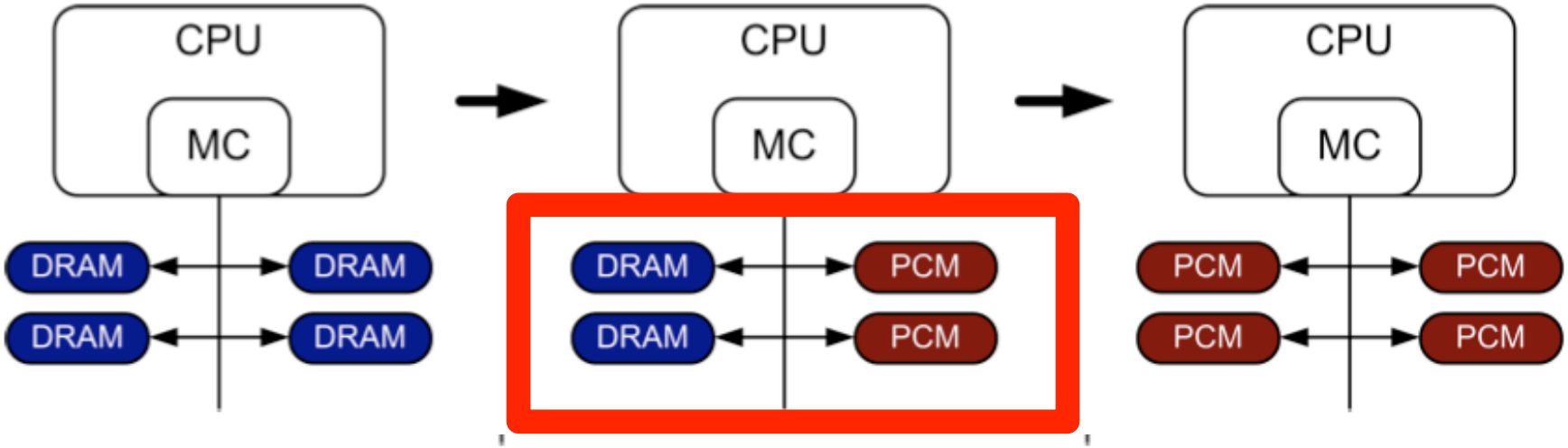
- Pros over DRAM
  - Better technology scaling
  - Non volatility
  - Low idle power (no refresh)
- Cons
  - Higher latencies: ~4-15x DRAM (especially write)
  - Higher active energy: ~2-50x DRAM (especially write)
  - Lower endurance (a cell dies after ~10^8 writes)
- Challenges in enabling PCM as DRAM replacement/helper:
  - Mitigate PCM shortcomings
  - Find the right way to place PCM in the system
  - Ensure secure and fault-tolerant PCM operation

PCM-based Main Memory: Research Challenges

- Where to place PCM in the memory hierarchy?
  - Hybrid OS controlled PCM-DRAM
  - Hybrid OS controlled PCM and hardware-controlled DRAM
  - Pure PCM main memory
- How to mitigate shortcomings of PCM?
- How to minimize amount of DRAM in the system?
- How to take advantage of (byte-addressable and fast) non-volatile main memory?
- Can we design specific-NVM-technology-agnostic techniques?

PCM-based Main Memory (I)

- How should PCM-based (main) memory be organized?



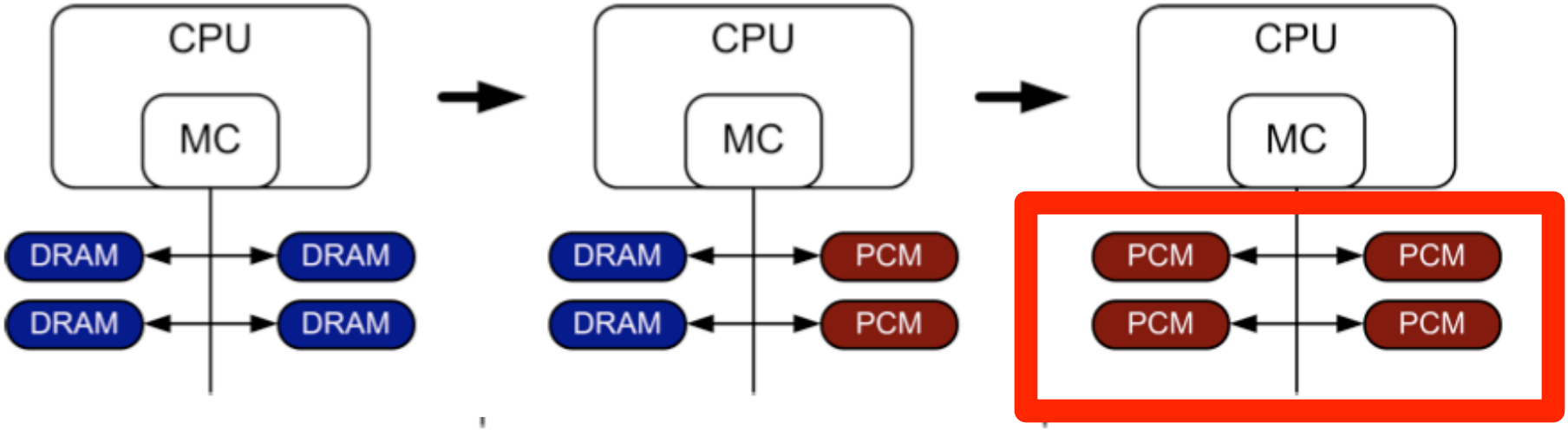
- Hybrid PCM+DRAM [Qureshi+ ISCA'09, Dhiman+ DAC'09, Meza+ IEEE CAL'12]:
  - How to partition/migrate data between PCM and DRAM

Hybrid Memory Systems: Challenges

- Partitioning
  - Should DRAM be a cache or main memory, or configurable?
  - What fraction? How many controllers?
- Data allocation/movement (energy, performance, lifetime)
  - Who manages allocation/movement?
  - What are good control algorithms?
  - How do we prevent degradation of service due to wearout?
- Design of cache hierarchy, memory controllers, OS
  - Mitigate PCM shortcomings, exploit PCM advantages
- Design of PCM/DRAM chips and modules
  - Rethink the design of PCM/DRAM with new requirements

PCM-based Main Memory (II)

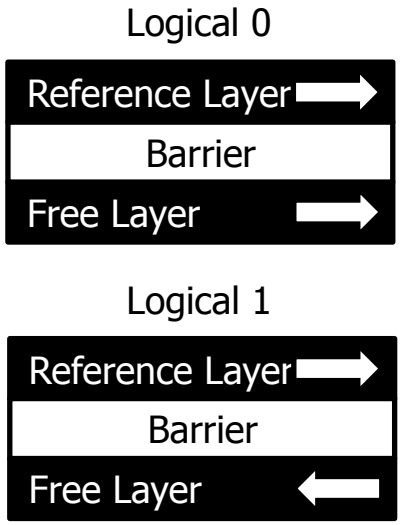
- How should PCM-based (main) memory be organized?

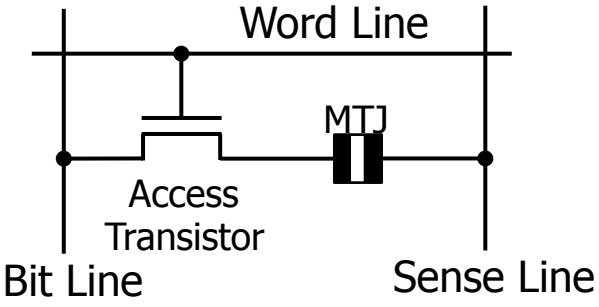


- Pure PCM main memory [Lee et al., ISCA'09, Top Picks'10]:
  - How to redesign entire hierarchy (and cores) to overcome PCM shortcomings

Aside: STT-RAM Basics

- Magnetic Tunnel Junction (MTJ)
  - Reference layer: Fixed
  - Free layer: Parallel or anti-parallel
- Cell
  - Access transistor, bit/sense lines
- Read and Write
  - Read: Apply a small voltage across bitline and senseline; read the current.
  - Write: Push large current through MTJ. Direction of current determines new orientation of the free layer.





- Kultursay et al., "Evaluating STT-RAM as an Energy-Efficient Main Memory Alternative," ISPASS 2013

Aside: STT MRAM: Pros and Cons

- Pros over DRAM
  - Better technology scaling
  - Non volatility
  - Low idle power (no refresh)
- Cons
  - Higher write latency
  - Higher write energy
  - Reliability?
- Another level of freedom
  - Can trade off non-volatility for lower write latency/energy (by reducing the size of the MTJ)

Agenda

- Major Trends Affecting Main Memory
- Requirements from an Ideal Main Memory System
- Opportunity: Emerging Memory Technologies
  - Background
  - PCM (or Technology X) as DRAM Replacement
  - Hybrid Memory Systems
- Conclusions
- Discussion

An Initial Study: Replace DRAM with PCM

- Lee, Ipek, Mutlu, Burger, "Architecting Phase Change Memory as a Scalable DRAM Alternative," ISCA 2009.
  - Surveyed prototypes from 2003-2008 (e.g. IEDM, VLSI, ISSCC)
  - Derived "average" PCM parameters for F=90nm

Density

- 9 - 12F^2 using BJT
- 1.5× DRAM

Latency

- 50ns Rd, 150ns Wr
- 4×, 12× DRAM

Endurance

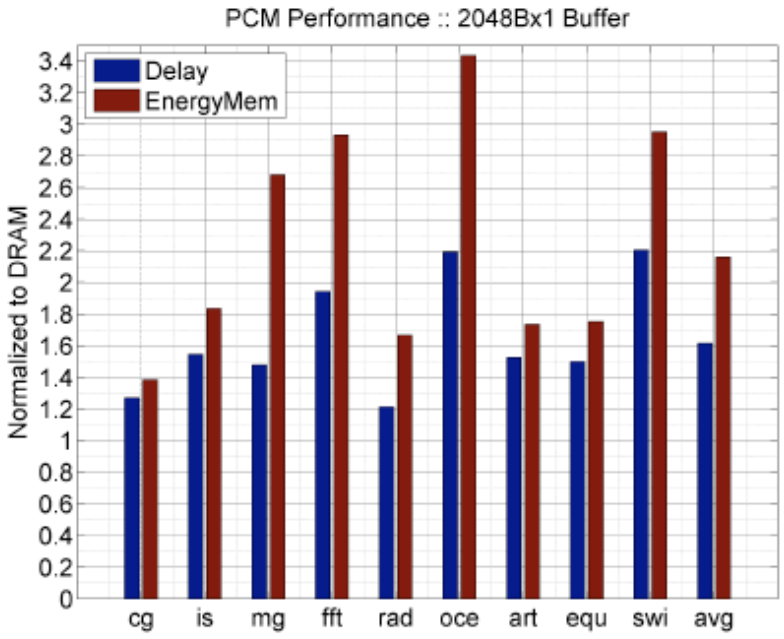
- 1E+08 writes
- 1E-08× DRAM

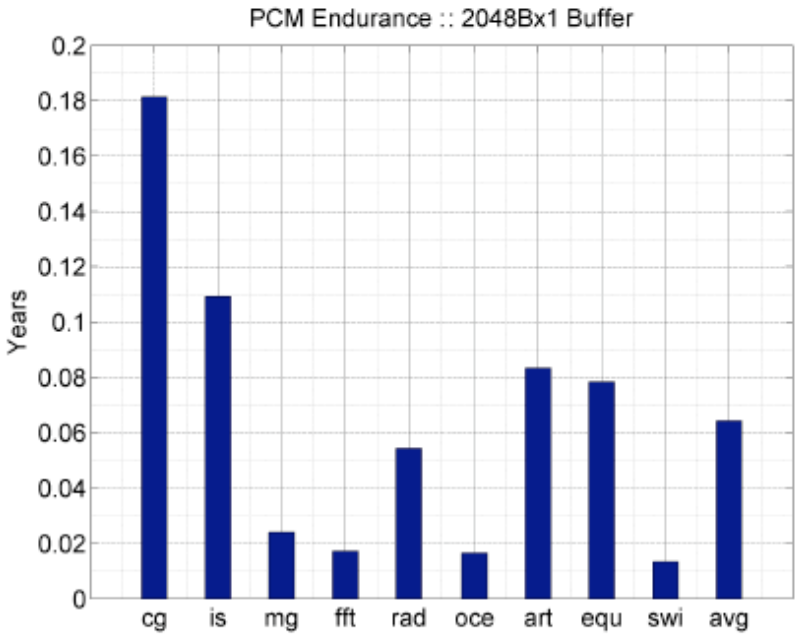
Energy

- 40μA Rd, 150μA Wr
- 2×, 43× DRAM

Results: Naïve Replacement of DRAM with PCM

- Replace DRAM with PCM in a 4-core, 4MB L2 system
- PCM organized the same as DRAM: row buffers, banks, peripherals
- 1.6x delay, 2.2x energy, 500-hour average lifetime



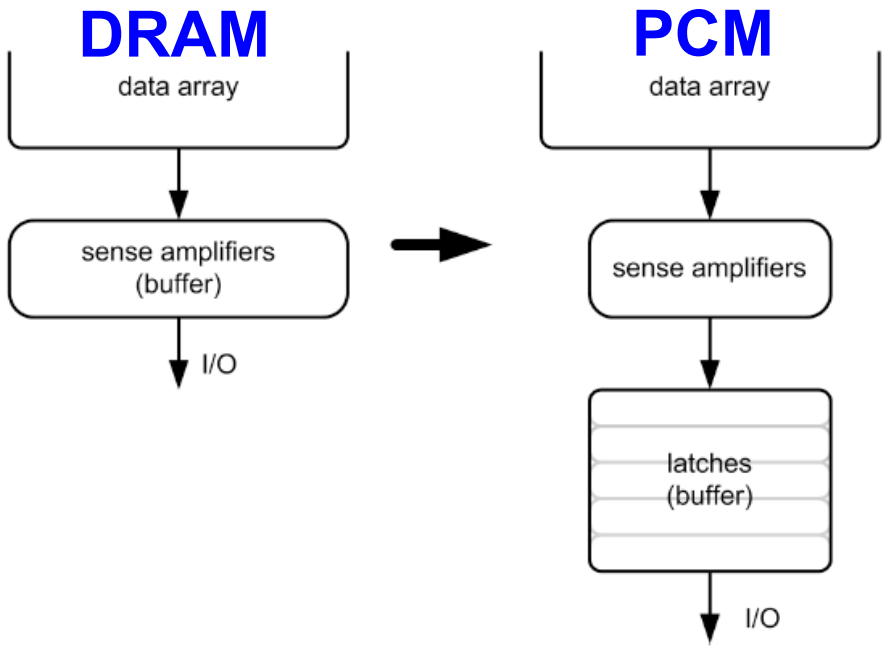


- Lee, Ipek, Mutlu, Burger, "Architecting Phase Change Memory as a Scalable DRAM Alternative," ISCA 2009.

Architecting PCM to Mitigate Shortcomings

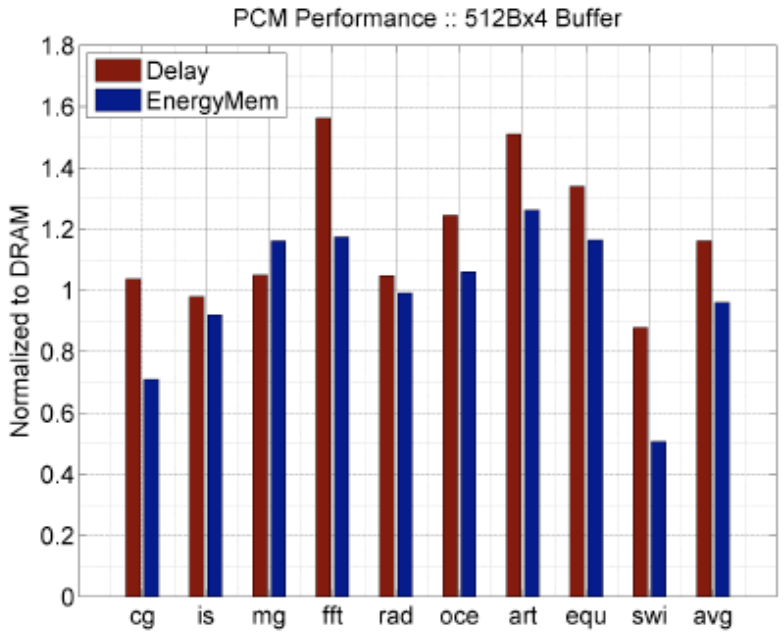
- Idea 1: Use multiple narrow row buffers in each PCM chip
  → Reduces array reads/writes → better endurance, latency, energy

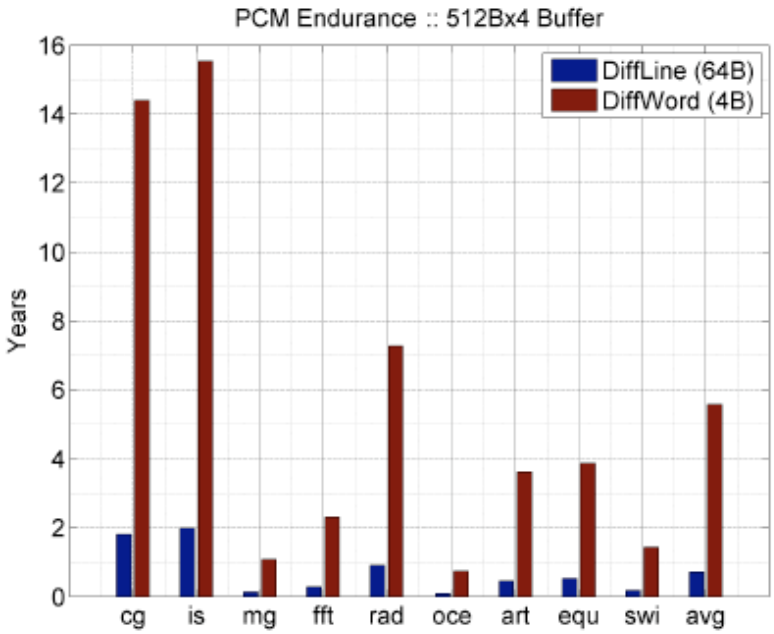
- Idea 2: Write into array at cache block or word granularity
  → Reduces unnecessary wear



Results: Architected PCM as Main Memory

- 1.2x delay, 1.0x energy, 5.6-year average lifetime
- Scaling improves energy, endurance, density



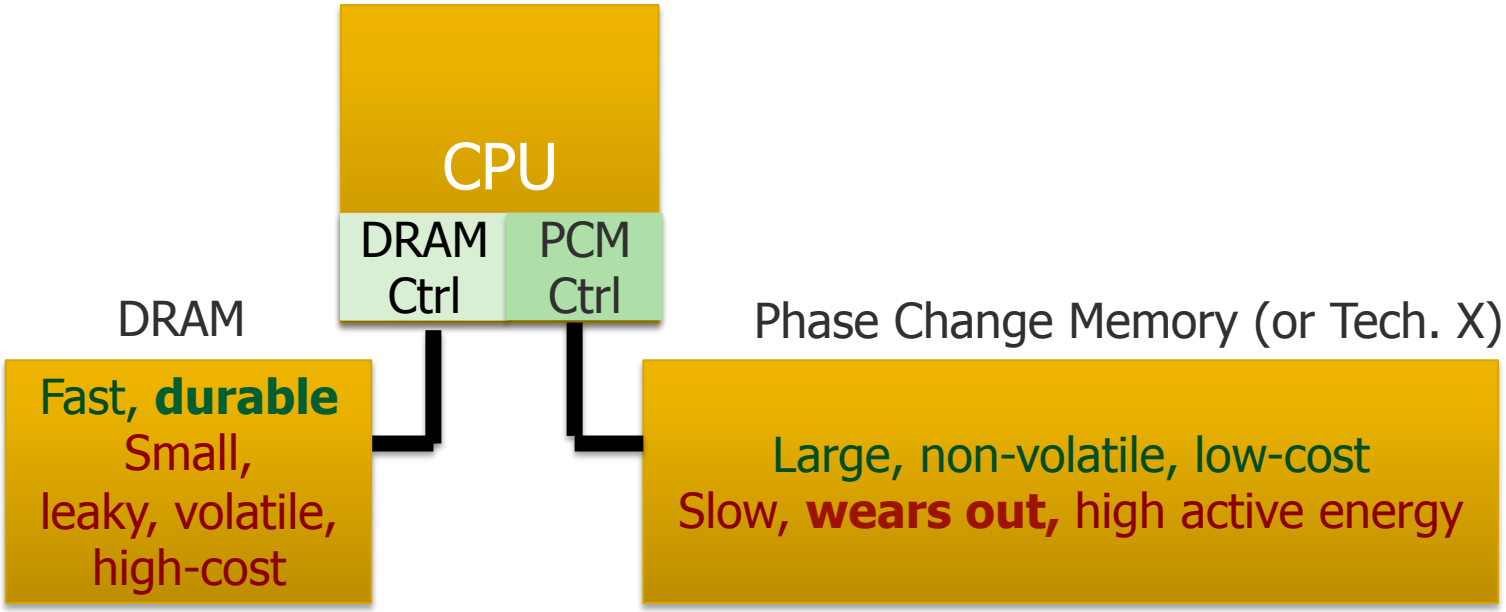


- Caveat 1: Worst-case lifetime is much shorter (no guarantees)
- Caveat 2: Intensive applications see large performance and energy hits
- Caveat 3: Optimistic PCM parameters?

Agenda

- Major Trends Affecting Main Memory
- Requirements from an Ideal Main Memory System
- Opportunity: Emerging Memory Technologies
  - Background
  - PCM (or Technology X) as DRAM Replacement
  - Hybrid Memory Systems
- Conclusions
- Discussion

Hybrid Memory Systems



Hardware/software manage data allocation and movement
to achieve the best of multiple technologies

Meza, Chang, Yoon, Mutlu, Ranganathan, "Enabling Efficient and Scalable Hybrid Memories," IEEE Comp. Arch. Letters, 2012.

One Option: DRAM as a Cache for PCM

- PCM is main memory; DRAM caches memory rows/blocks
  - Benefits: Reduced latency on DRAM cache hit; write filtering
- Memory controller hardware manages the DRAM cache
  - Benefit: Eliminates system software overhead

- Three issues:
  - What data should be placed in DRAM versus kept in PCM?
  - What is the granularity of data movement?
  - How to design a low-cost hardware-managed DRAM cache?

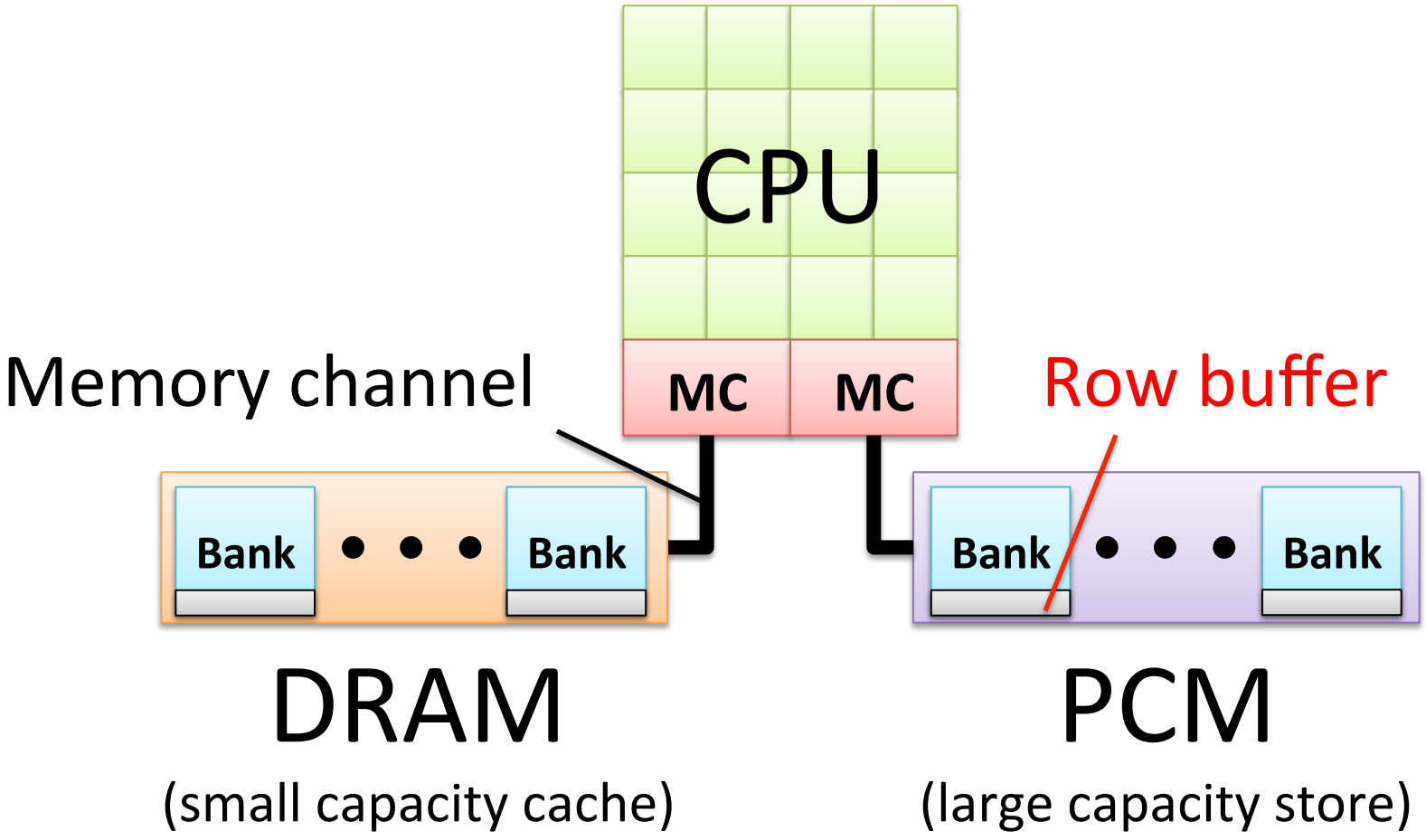
- Two idea directions:
  - Locality-aware data placement **[Yoon+ , ICCD 2012]**
  - Cheap tag stores and dynamic granularity **[Meza+, IEEE CAL 2012]**

Row Buffer Locality Aware Caching Policies for Hybrid Memories

HanBin Yoon
Justin Meza
Rachata Ausavarungnirun
Rachael Harding
Onur Mutlu

Carnegie Mellon University

Hybrid Memory: A Closer Look



Key Observation

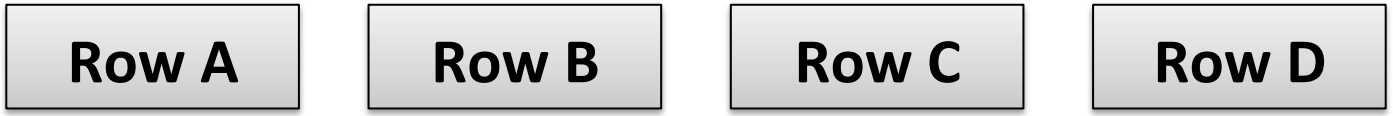
- Row buffers exist in both DRAM and PCM
  - Row hit latency **similar** in DRAM & PCM [Lee+ ISCA'09]
  - Row miss latency **small** in DRAM, **large** in PCM

- Place data in DRAM which
  - is likely to miss in the row buffer (low row buffer locality)→ miss penalty is smaller in DRAM

    AND
  - is reused many times → cache only the data worth the movement cost and DRAM space

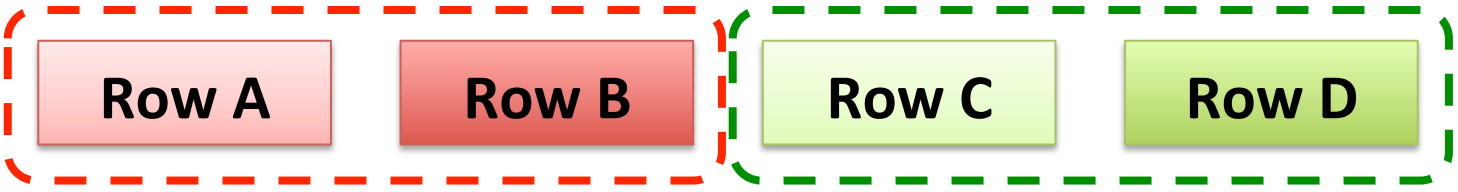
RBL-Awareness: An Example

Let's say a processor accesses four rows



RBL-Awareness: An Example

Let's say a processor accesses four rows with different row buffer localities (RBL)



Low RBL
(Frequently miss in row buffer)

High RBL
(Frequently hit in row buffer)

Case 1: RBL-*Unaware* Policy (state-of-the-art)

Case 2: RBL-Aware Policy (RBLA)

Case 1: RBL-*Unaware* Policy

A **row buffer locality-*unaware*** policy could place these rows in the following manner

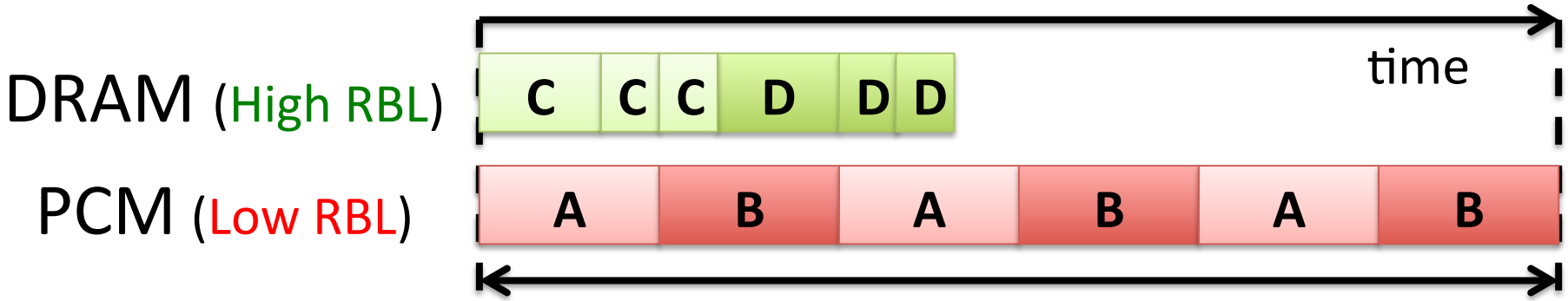
DRAM (High RBL)

PCM (Low RBL)

Case 1: RBL-*Unaware* Policy

Access pattern to main memory:

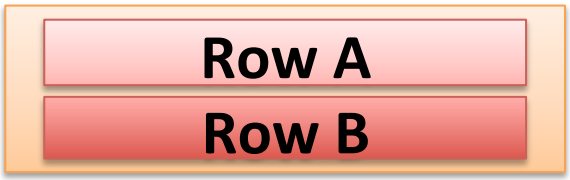
A (oldest), B, C, C, C, A, B, D, D, D, A, B (youngest)



RBL-*Unaware*: Stall time is 6 PCM device accesses

Case 2: RBL-Aware Policy (RBLA)

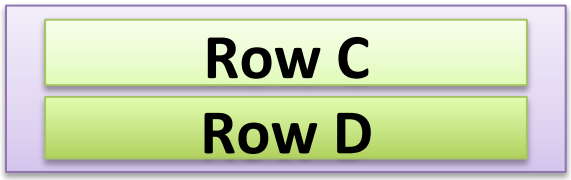
A **row buffer locality-aware** policy would place these rows in the **opposite** manner



DRAM

(Low RBL)

→ Access data at lower row buffer miss latency of DRAM



PCM

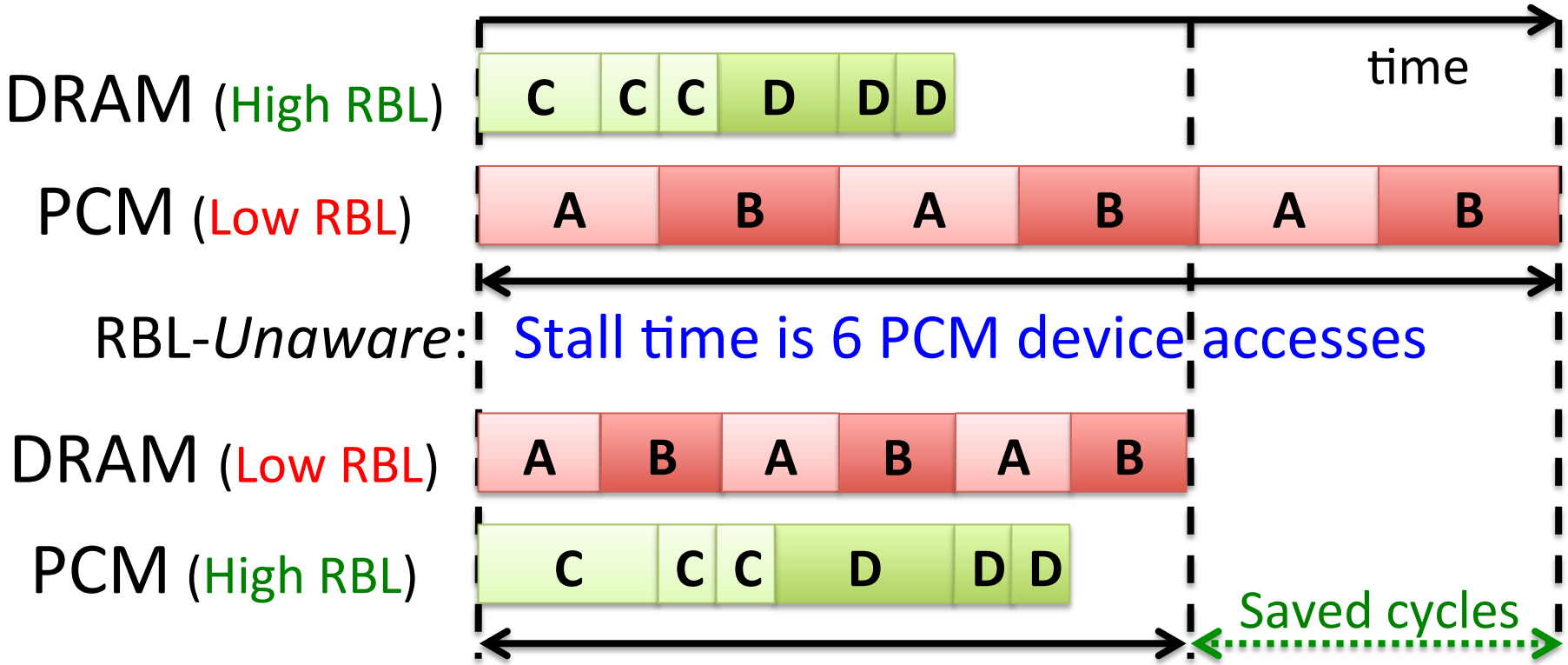
(High RBL)

→ Access data at low row buffer hit latency of PCM

Case 2: RBL-Aware Policy (RBLA)

Access pattern to main memory:

A (oldest), B, C, C, C, A, B, D, D, D, A, B (youngest)



RBL-Aware: Stall time is 6 **DRAM** device accesses

Our Mechanism: RBLA

1. For recently used rows in PCM:
   – Count row buffer misses as indicator of row buffer locality (RBL)

2. Cache to DRAM rows with misses ≥ threshold
   – Row buffer miss counts are periodically reset (only cache rows with high reuse)

Our Mechanism: RBLA-Dyn

1. For recently used rows in PCM:
   – Count row buffer misses as indicator of row buffer locality (RBL)

2. Cache to DRAM rows with misses ≥ threshold
   – Row buffer miss counts are periodically reset (only cache rows with high reuse)

3. Dynamically adjust threshold to adapt to workload/system characteristics
   – Interval-based cost-benefit analysis

Implementation: "Statistics Store"

- Goal: To keep count of row buffer misses to recently used rows in PCM

- Hardware structure in memory controller
  - Operation is similar to a cache
    - Input: row address
    - Output: row buffer miss count
  - 128-set 16-way statistics store (9.25KB) achieves system performance within 0.3% of an unlimited-sized statistics store

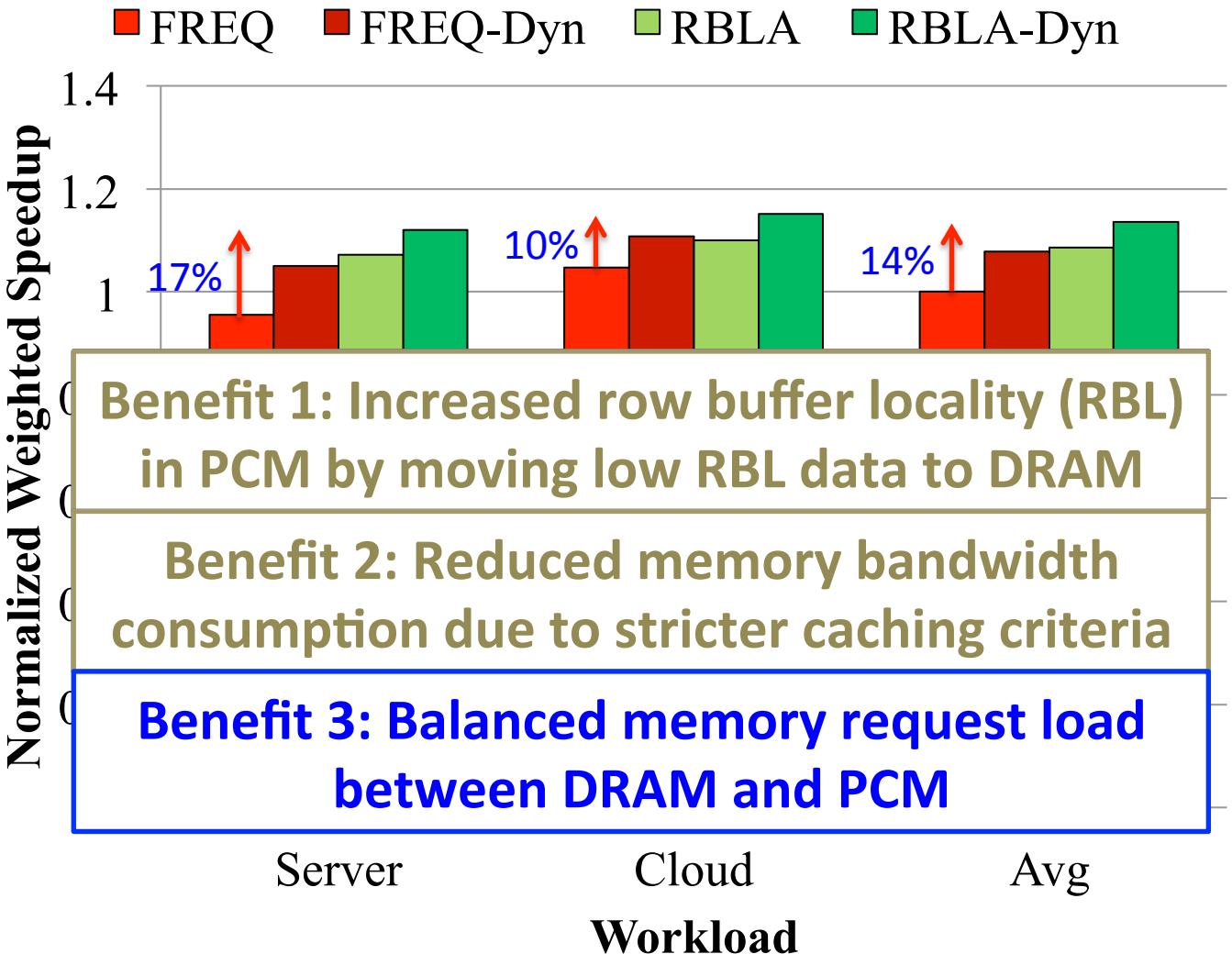
Evaluation Methodology

- Cycle-level x86 CPU-memory simulator
  - **CPU**: 16 out-of-order cores, 32KB private L1 per core, 512KB shared L2 per core
  - **Memory**: 1GB DRAM (8 banks), 16GB PCM (8 banks), 4KB migration granularity
- 36 multi-programmed server, cloud workloads
  - Server: TPC-C (OLTP), TPC-H (Decision Support)
  - Cloud: Apache (Webserv.), H.264 (Video), TPC-C/H
- Metrics: Weighted speedup (perf.), perf./Watt (energy eff.), Maximum slowdown (fairness)

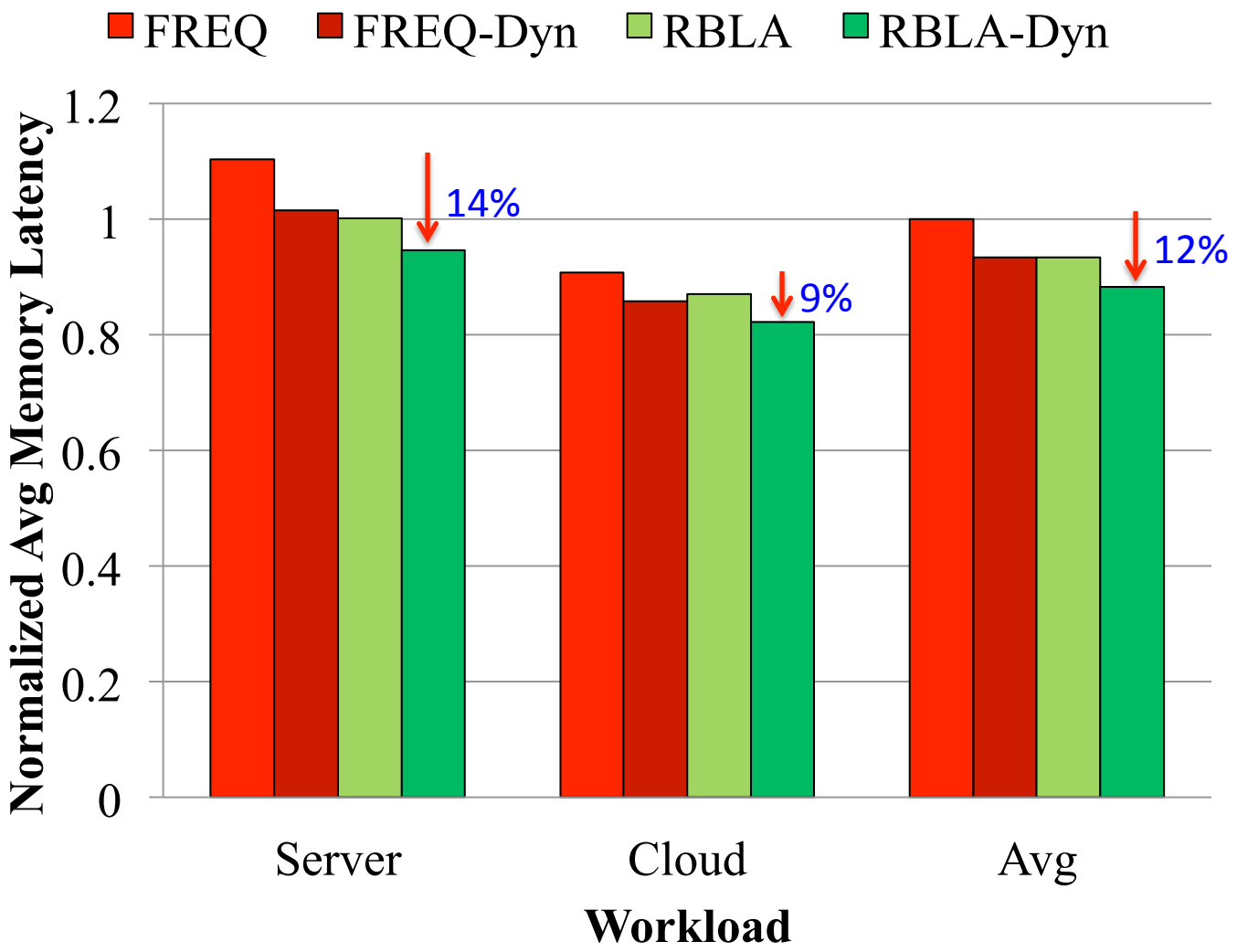
Comparison Points

- **Conventional LRU Caching**
- **FREQ**: Access-frequency-based caching
  - Places "hot data" in cache [Jiang+ HPCA'10]
  - Cache to DRAM rows with accesses ≥ threshold
  - Row buffer locality-*unaware*
- **FREQ-Dyn**: Adaptive Freq.-based caching
  - **FREQ** + our dynamic threshold adjustment
  - Row buffer locality-*unaware*
- **RBLA**: Row buffer locality-aware caching
- **RBLA-Dyn**: Adaptive RBL-aware caching

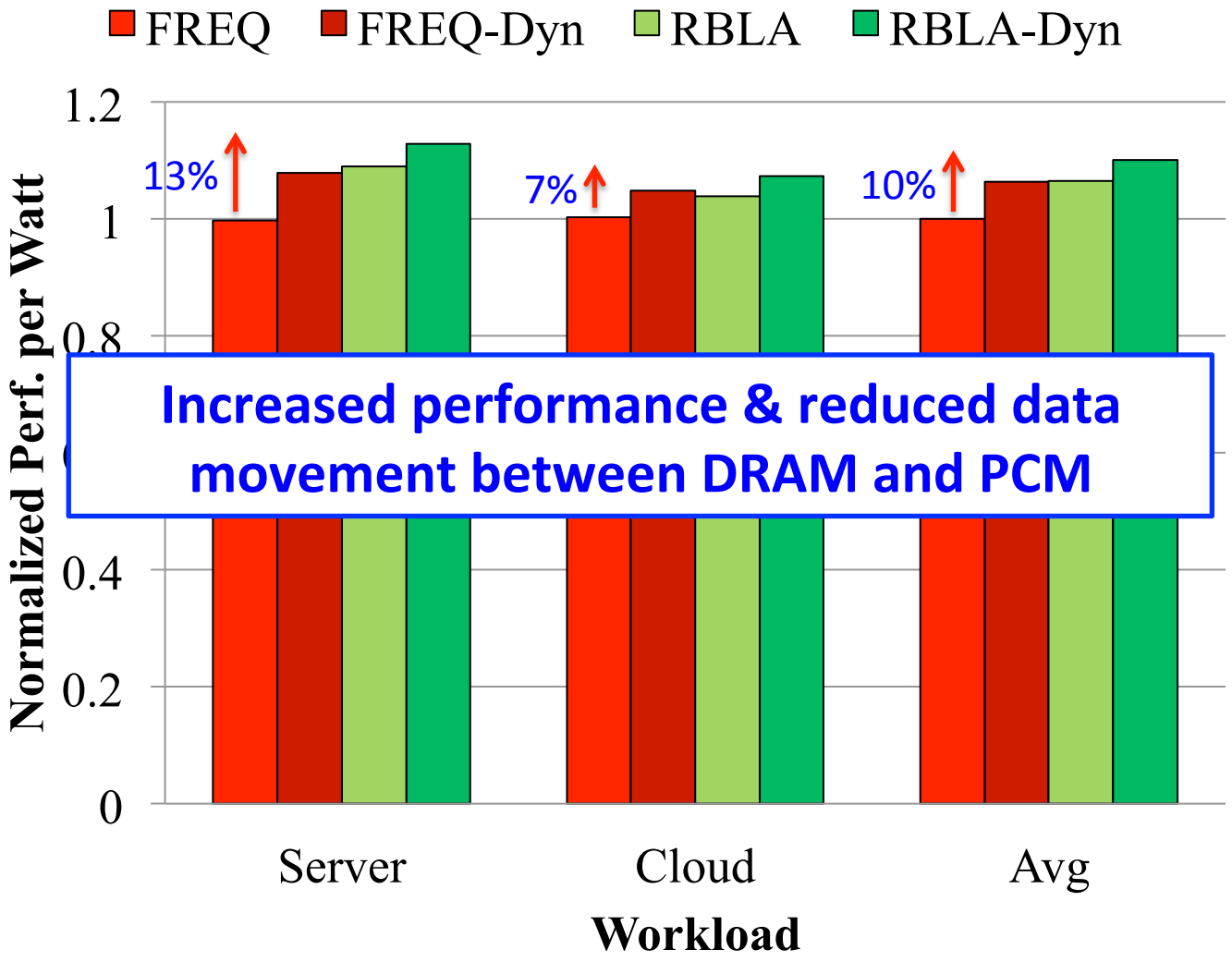
System Performance



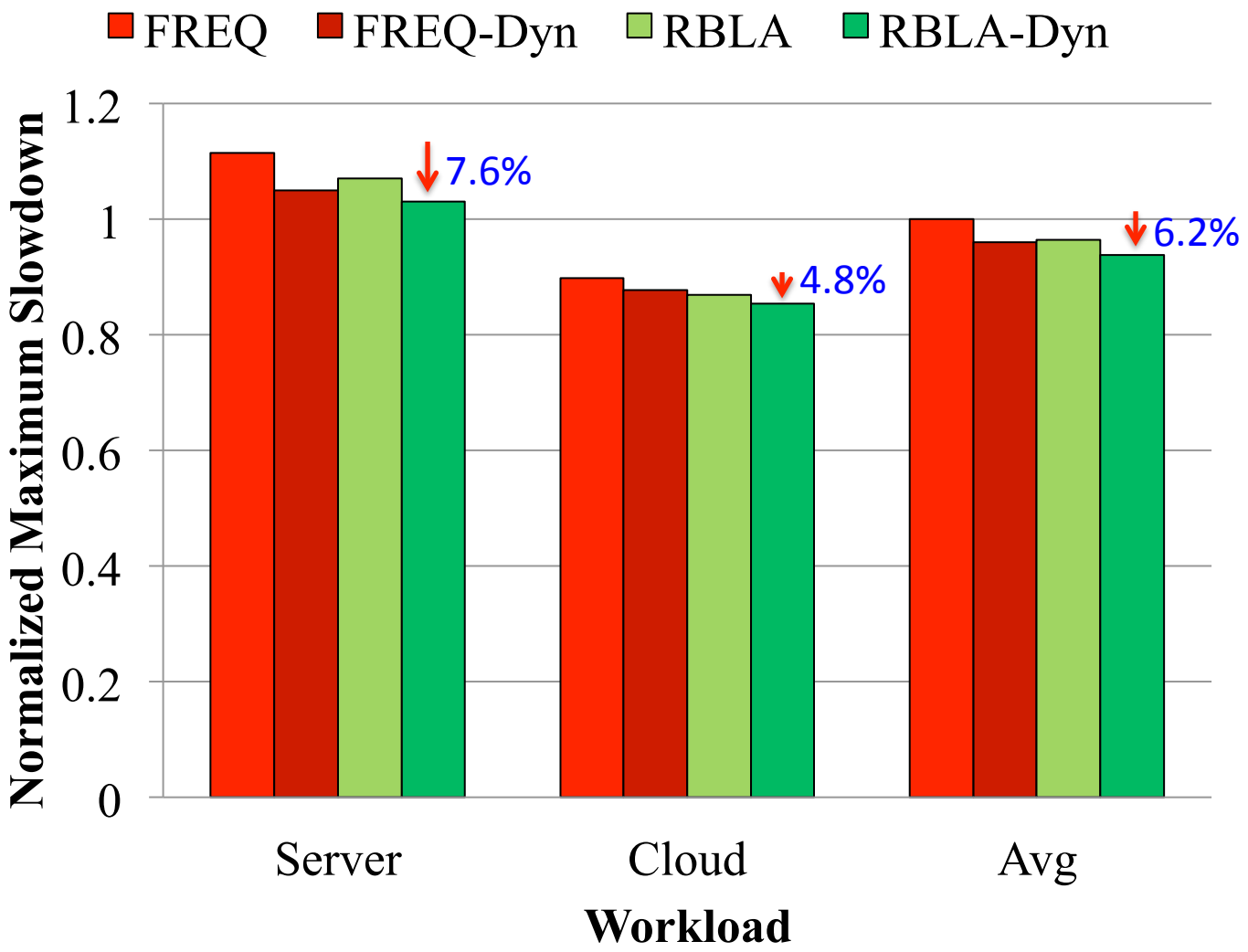
Average Memory Latency



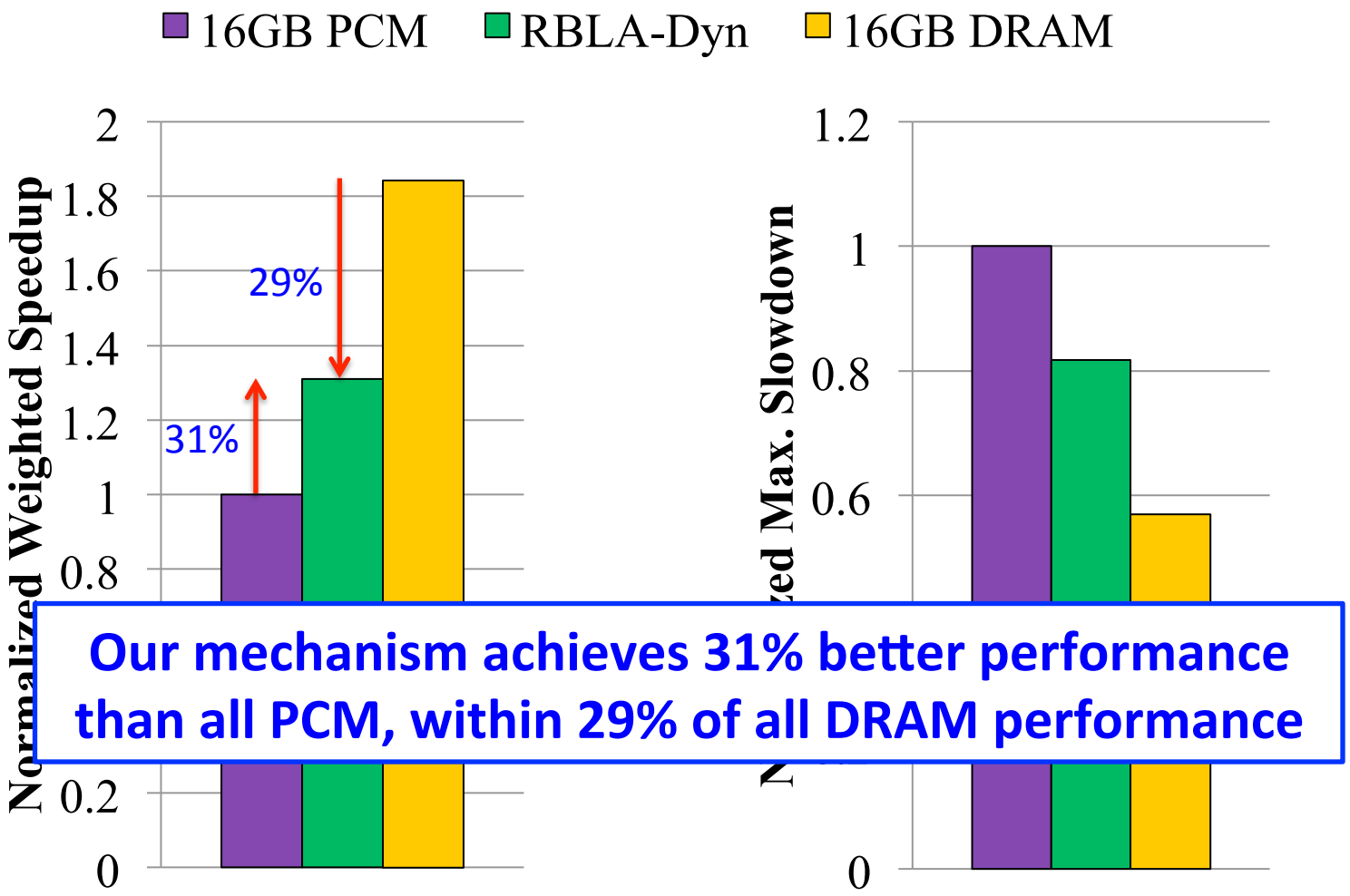
Memory Energy Efficiency



Thread Fairness



Compared to All-PCM/DRAM



Our mechanism achieves 31% better performance than all PCM, within 29% of all DRAM performance

Summary

- Different memory technologies have different strengths
- A hybrid memory system (DRAM-PCM) aims for best of both
- **Problem:** How to place data between these heterogeneous memory devices?
- **Observation:** PCM array access latency is higher than DRAM's – But peripheral circuit (row buffer) access latencies are similar
- **Key Idea:** Use row buffer locality (RBL) as a key criterion for data placement
- **Solution:** Cache to DRAM rows with low RBL and high reuse
- Improves both performance and energy efficiency over state-of-the-art caching policies

Row Buffer Locality Aware Caching Policies for Hybrid Memories

HanBin Yoon
Justin Meza
Rachata Ausavarungnirun
Rachael Harding
Onur Mutlu

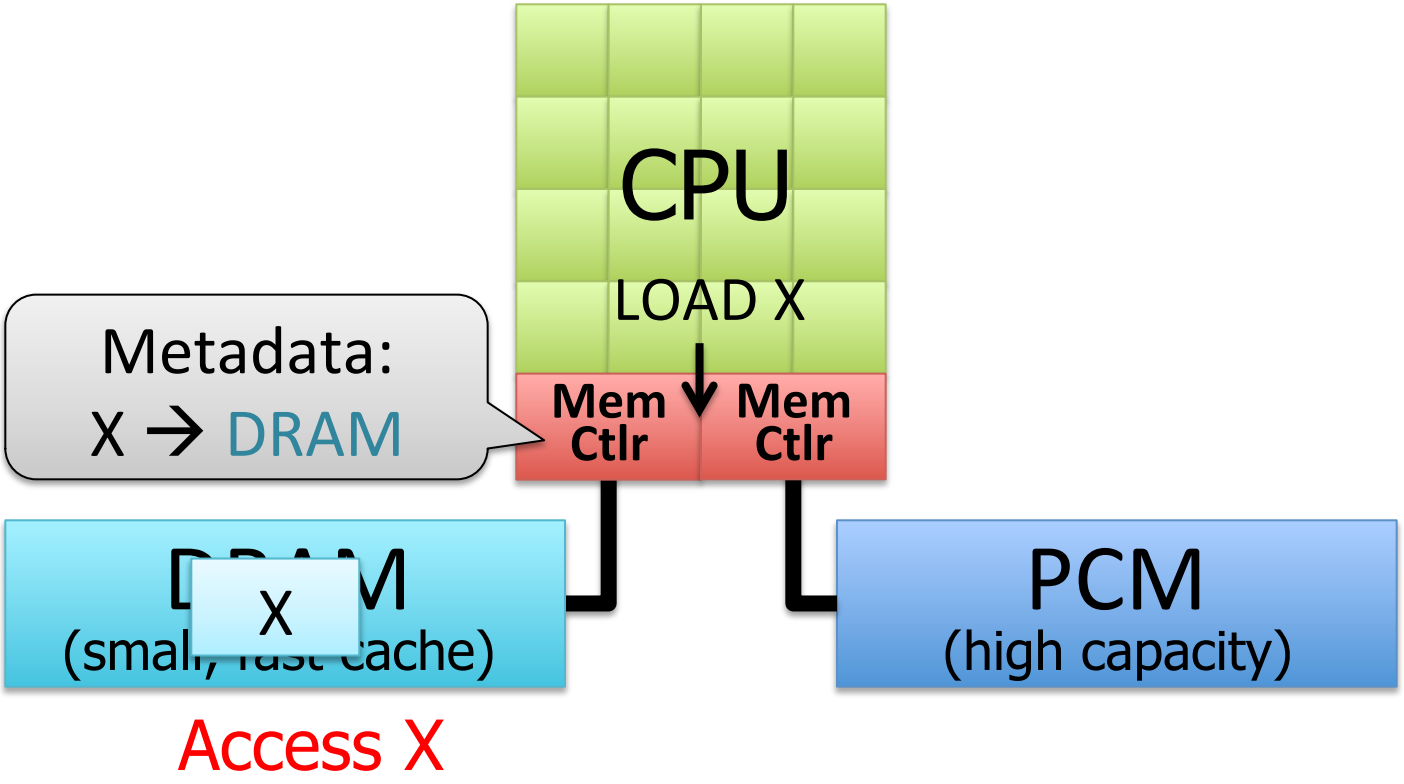
Carnegie Mellon University

Agenda

- Major Trends Affecting Main Memory
- Requirements from an Ideal Main Memory System
- Opportunity: Emerging Memory Technologies
  - Background
  - PCM (or Technology X) as DRAM Replacement
  - Hybrid Memory Systems
    - Row-Locality Aware Data Placement
    - Efficient DRAM (or Technology X) Caches
- Conclusions
- Discussion

The Problem with Large DRAM Caches

- A large DRAM cache requires a large metadata (tag + block-based information) store
- How do we design an efficient DRAM cache?



Idea 1: Tags in Memory

- Store tags in the same row as data in DRAM
  - Store metadata in same row as their data
  - Data and metadata can be accessed together

- Benefit: No on-chip tag storage overhead
- Downsides:
  - Cache hit determined only after a DRAM access
  - Cache hit requires two DRAM accesses

Idea 2: Cache Tags in SRAM

- Recall Idea 1: Store all metadata in DRAM
  - To reduce metadata storage overhead

- Idea 2: Cache in on-chip SRAM frequently-accessed metadata
  - Cache only a small amount to keep SRAM size small

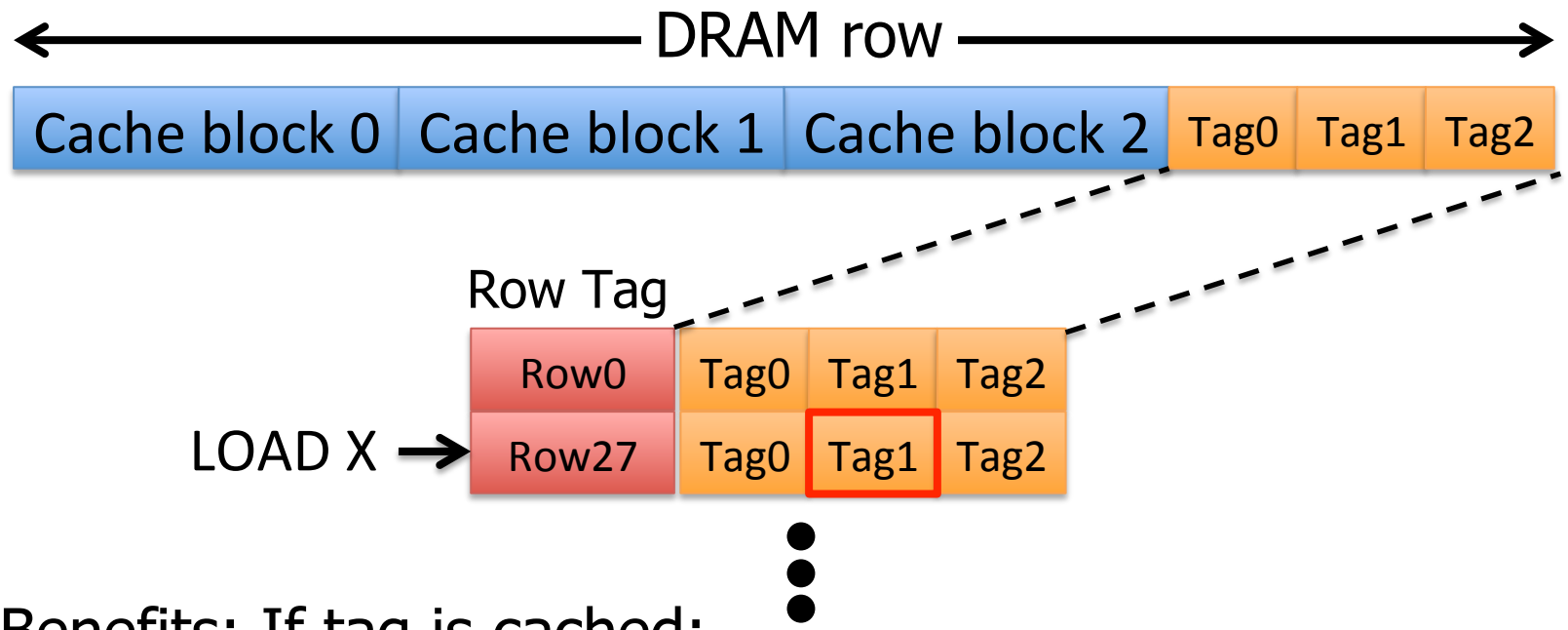
Idea 3: Dynamic Data Transfer Granularity

- Some applications benefit from caching more data
  - They have good spatial locality
- Others do not
  - Large granularity wastes bandwidth and reduces cache utilization

- Idea 3: Simple dynamic caching granularity policy
  - Cost-benefit analysis to determine best DRAM cache block size
  - Group main memory into sets of rows
  - Some row sets follow a fixed caching granularity
  - The rest of main memory follows the best granularity
    - Cost–benefit analysis: access latency versus number of cachings
    - Performed every quantum

TIMBER Tag Management

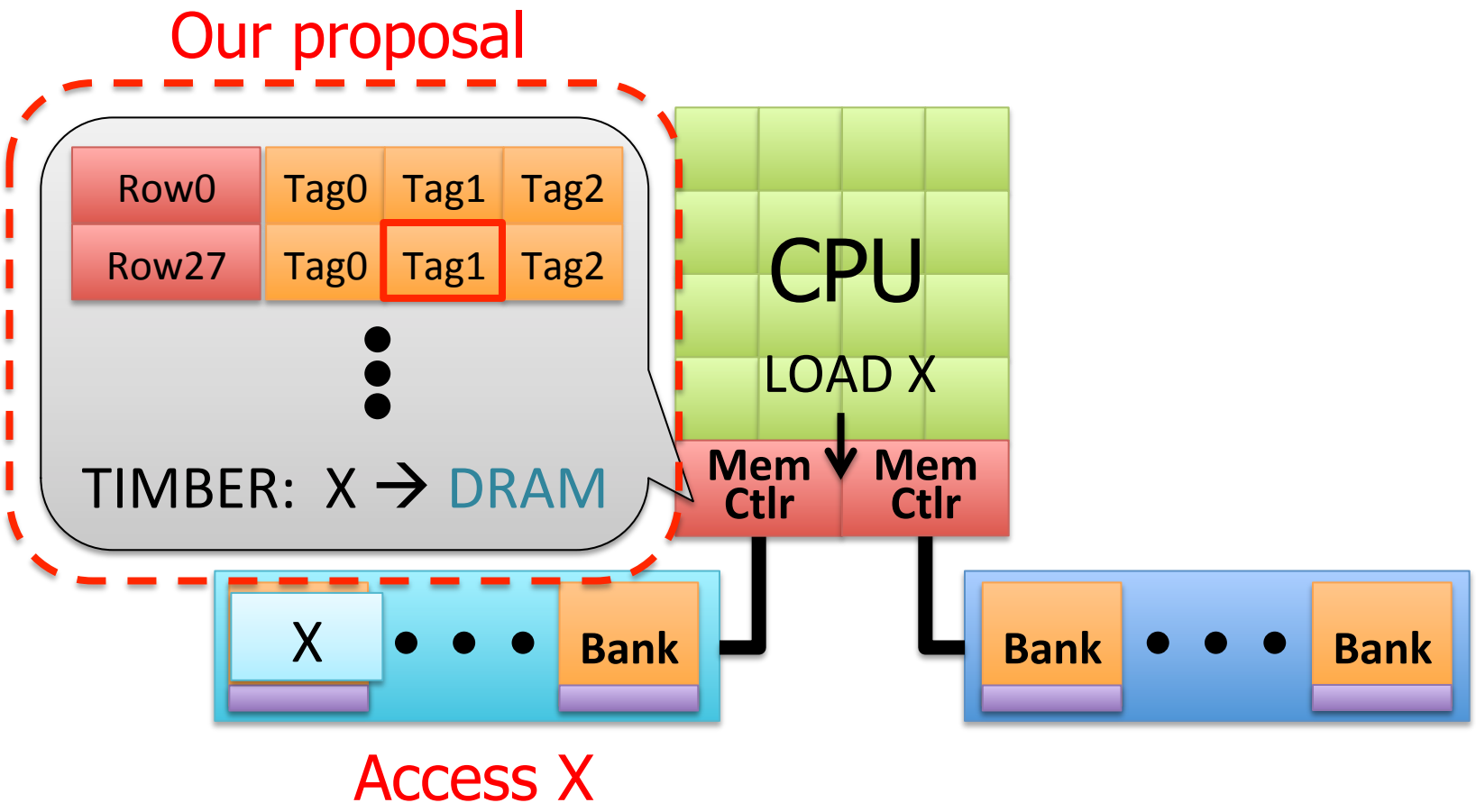
- A Tag-In-Memory BuffER (TIMBER)
  - Stores recently-used tags in a small amount of SRAM



- Benefits: If tag is cached:
  - no need to access DRAM twice
  - cache hit determined quickly

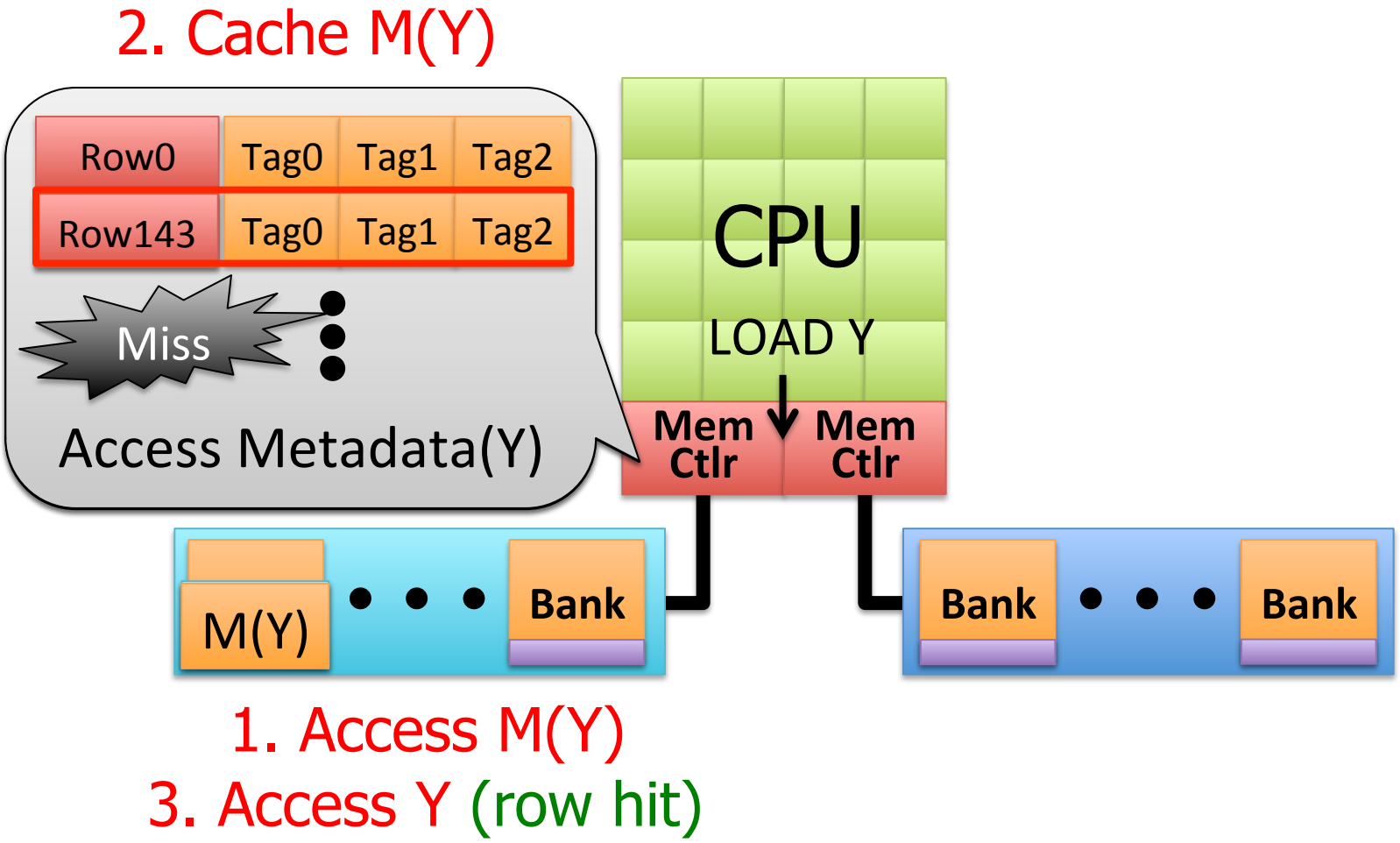
TIMBER Tag Management Example (I)

- Case 1: TIMBER hit



TIMBER Tag Management Example (II)

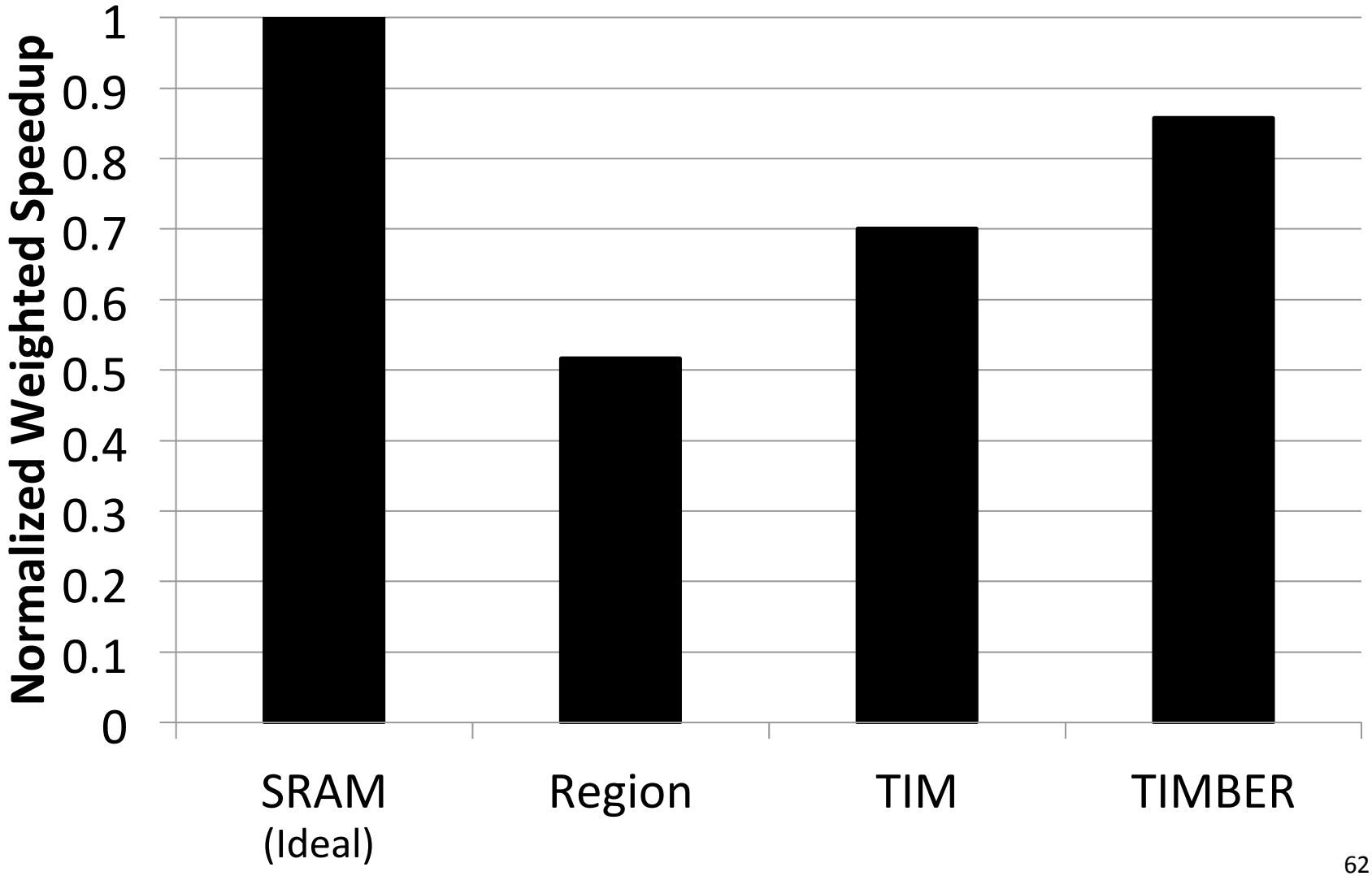
- Case 2: TIMBER miss



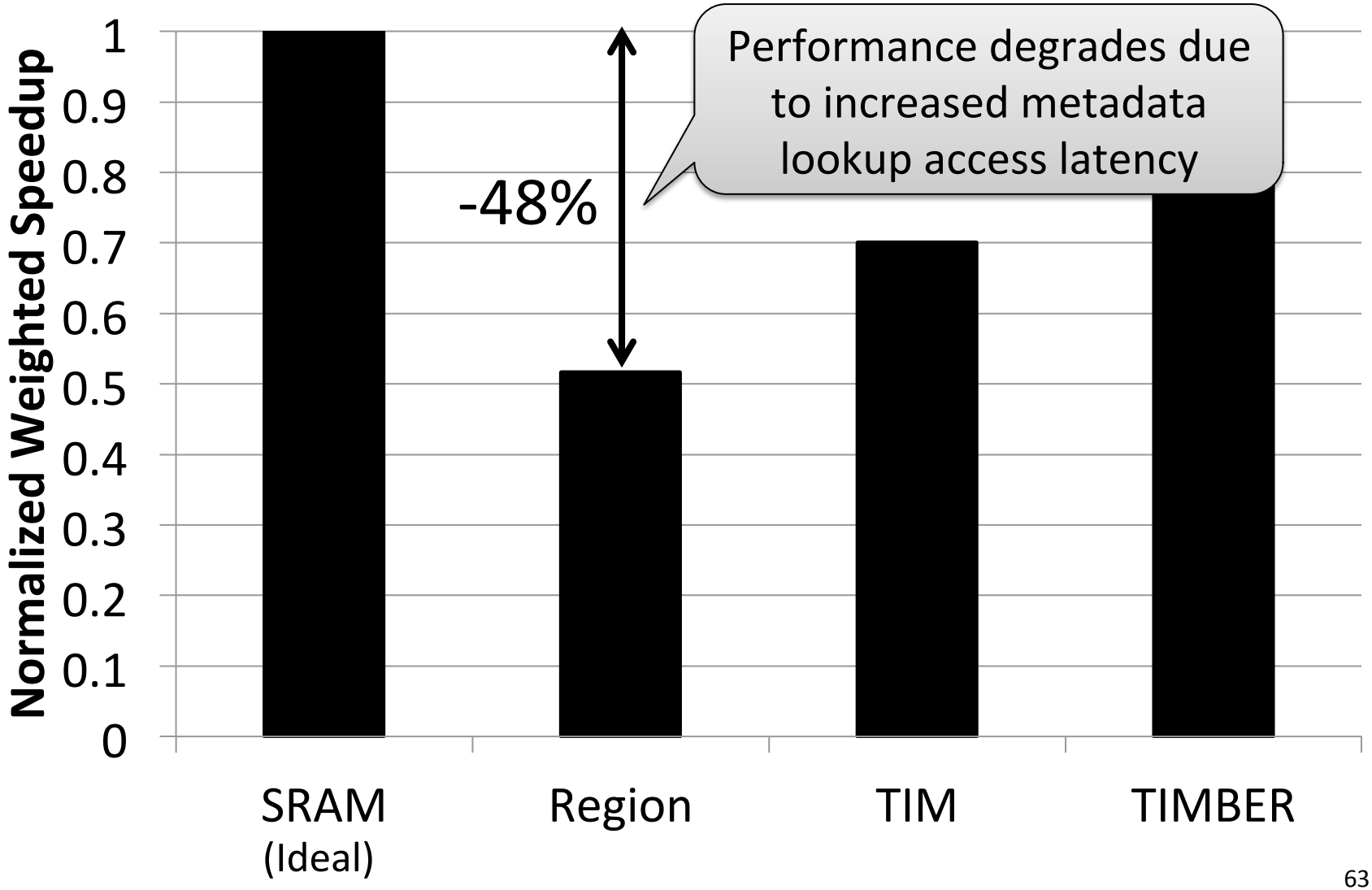
Methodology

- System: 8 out-of-order cores at 4 GHz
- Memory: 512 MB direct-mapped DRAM, 8 GB PCM
  - 128B caching granularity
  - DRAM row hit (miss): 200 cycles (400 cycles)
  - PCM row hit (clean / dirty miss): 200 cycles (640 / 1840 cycles)
- Evaluated metadata storage techniques
  - All SRAM system (8MB of SRAM)
  - Region metadata storage
  - TIM metadata storage (same row as data)
  - TIMBER, 64-entry direct-mapped (8KB of SRAM)

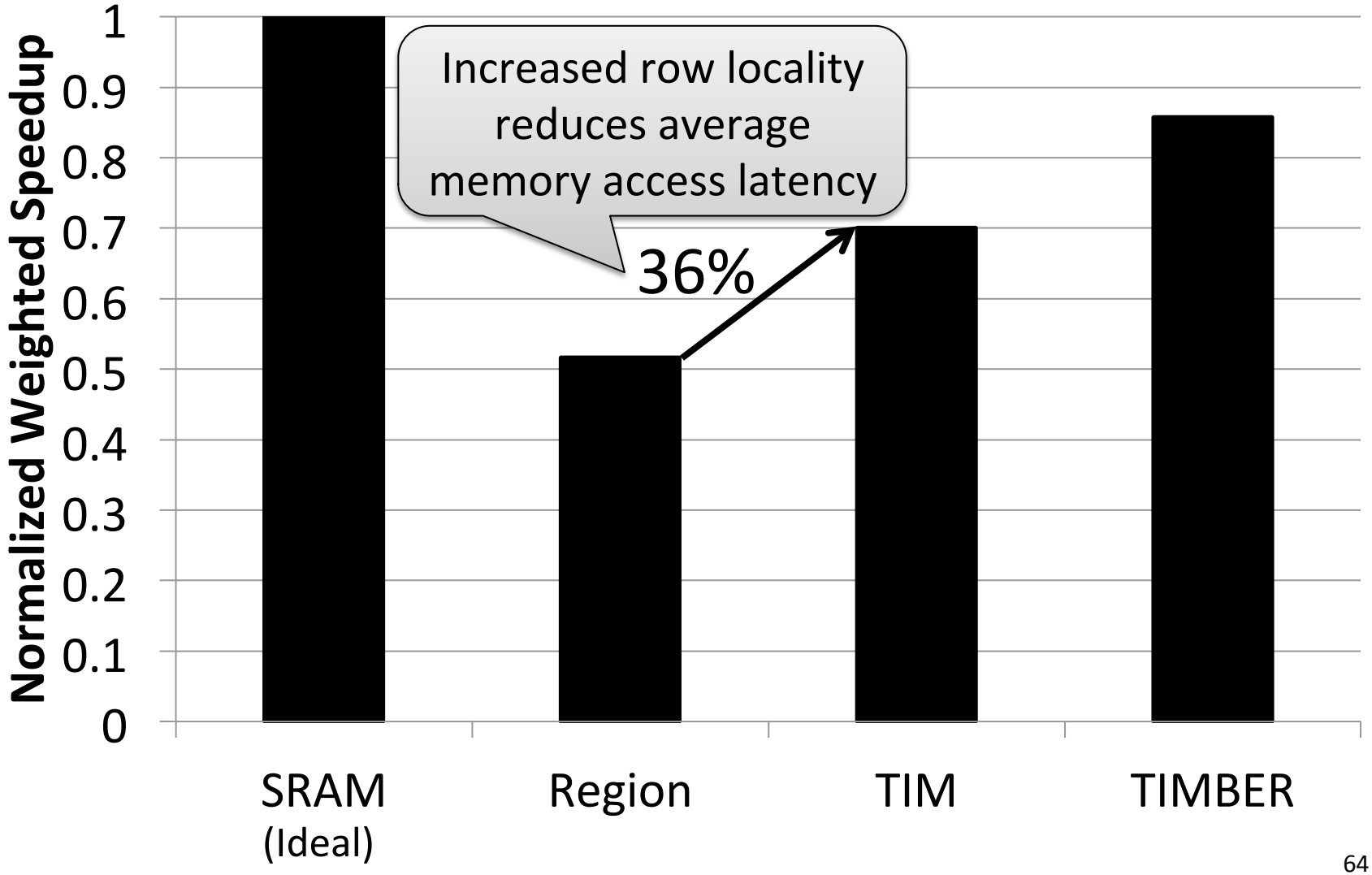
Metadata Storage Performance



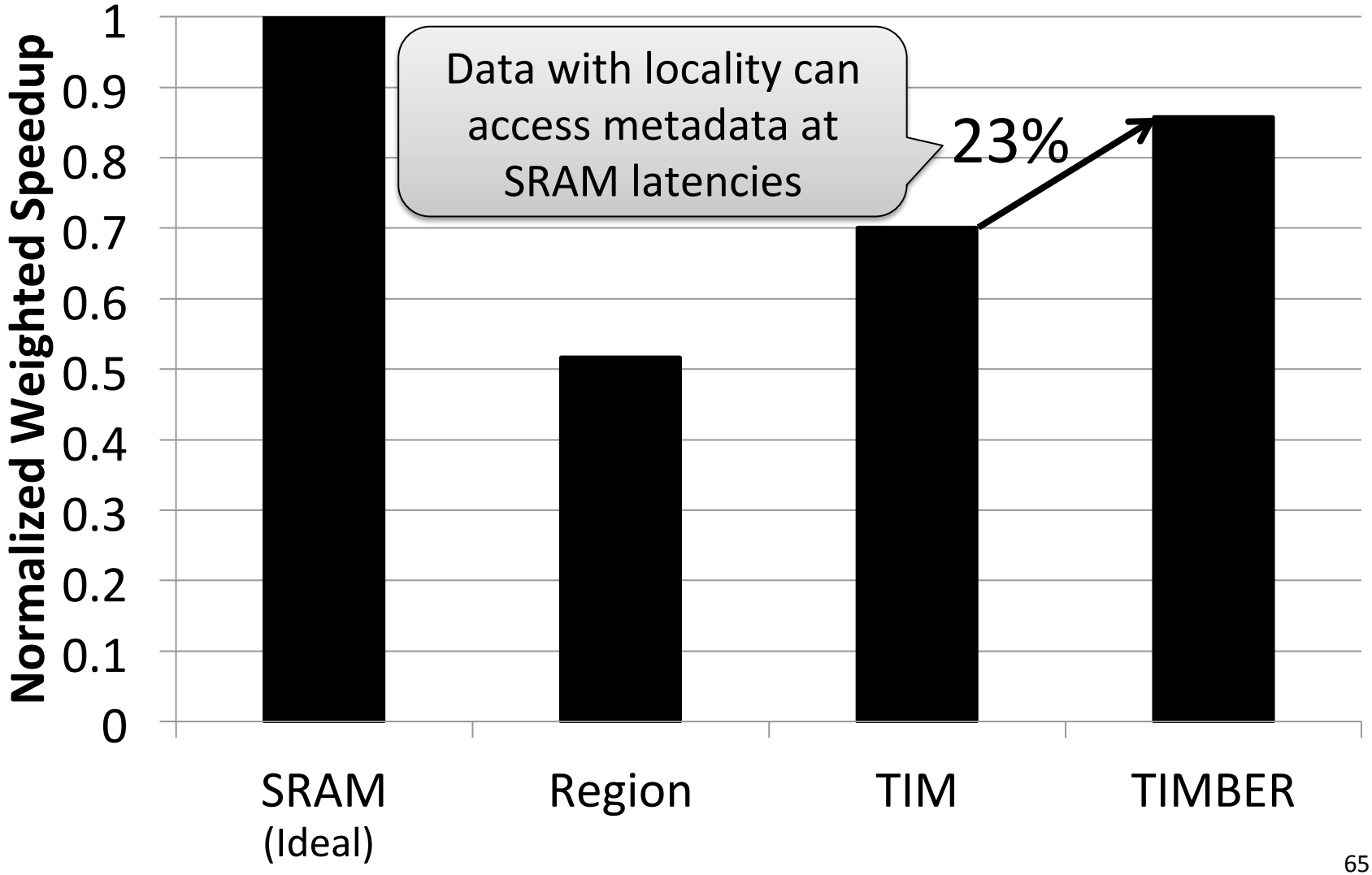
Metadata Storage Performance



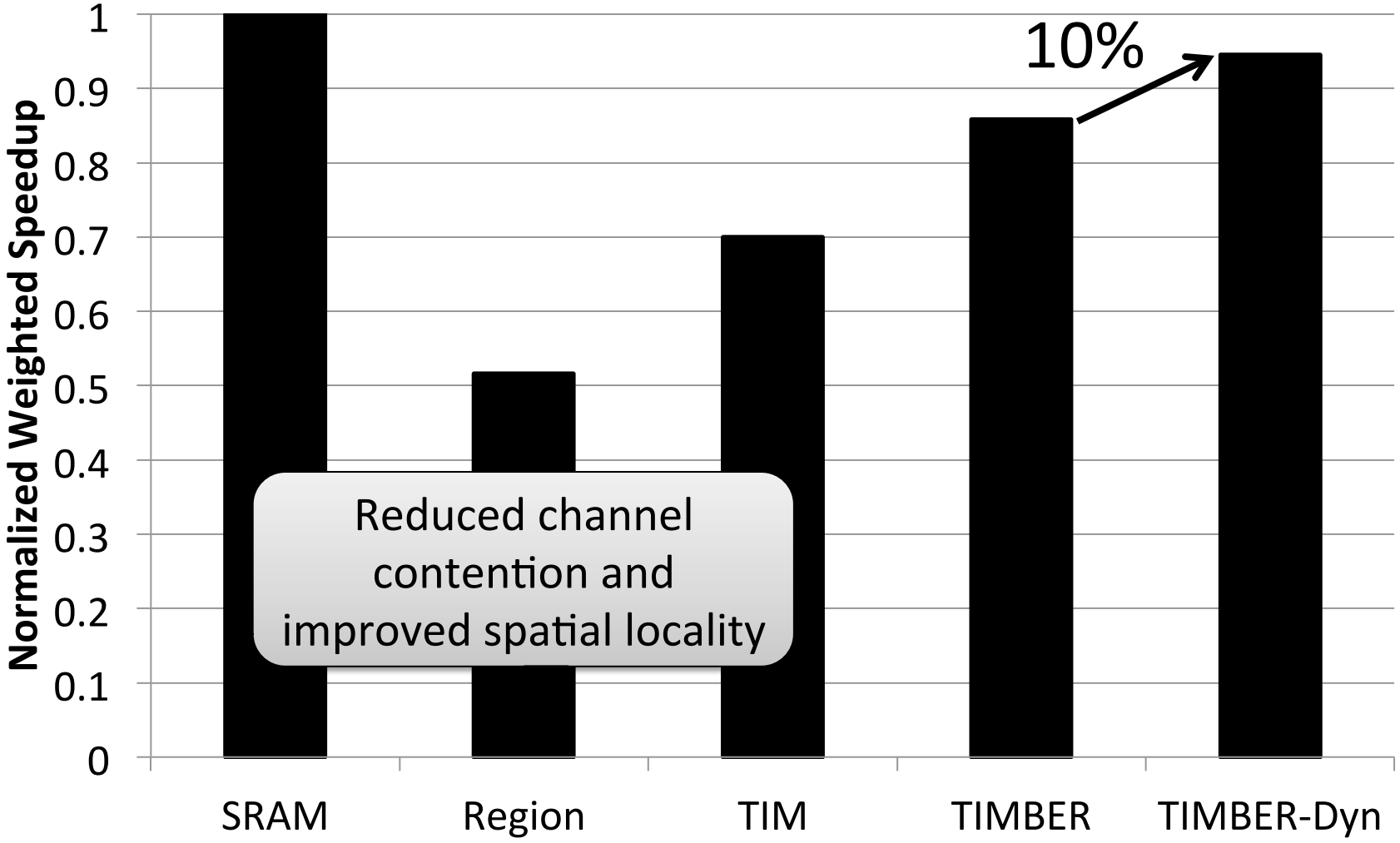
Metadata Storage Performance



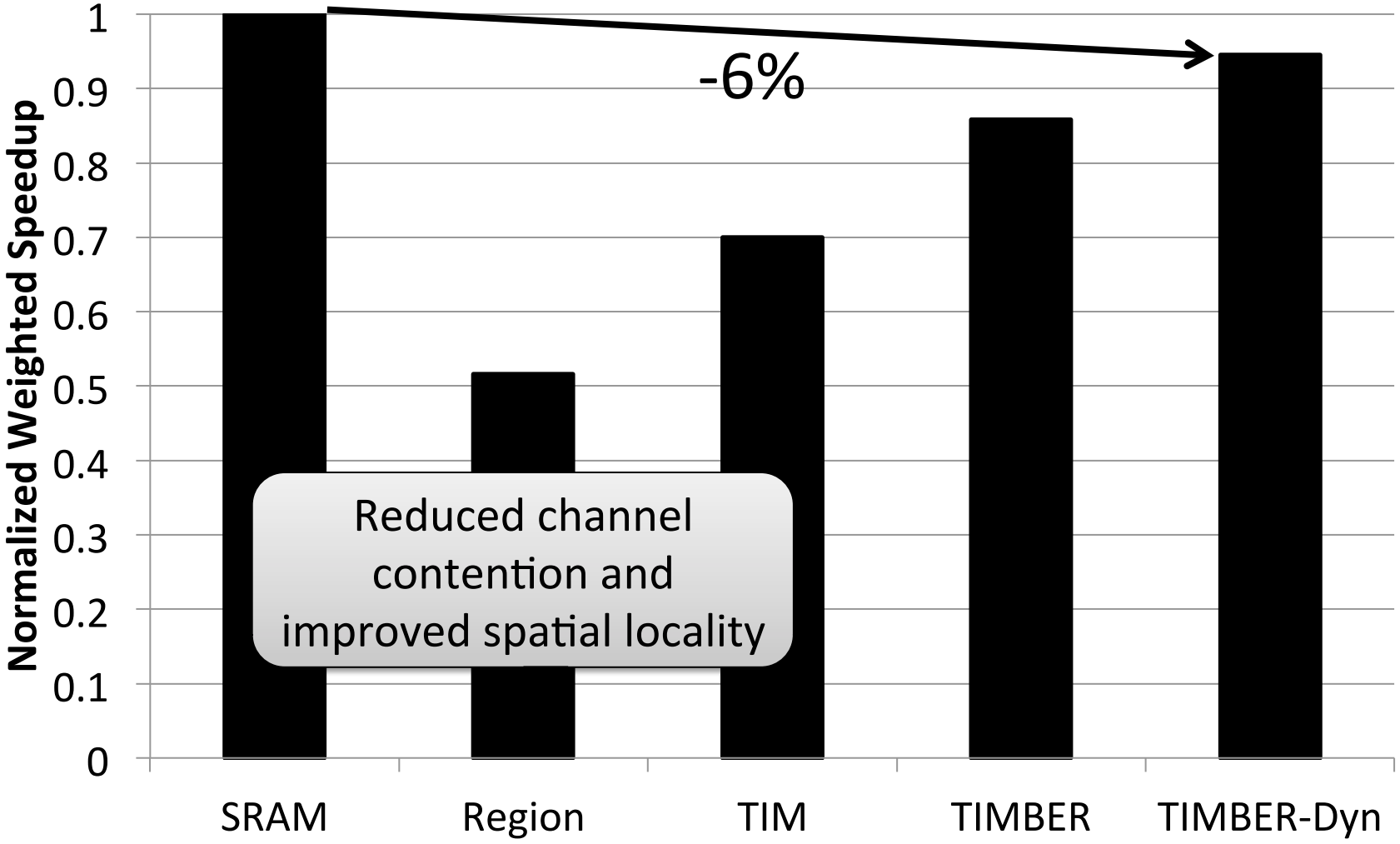
Metadata Storage Performance



Dynamic Granularity Performance

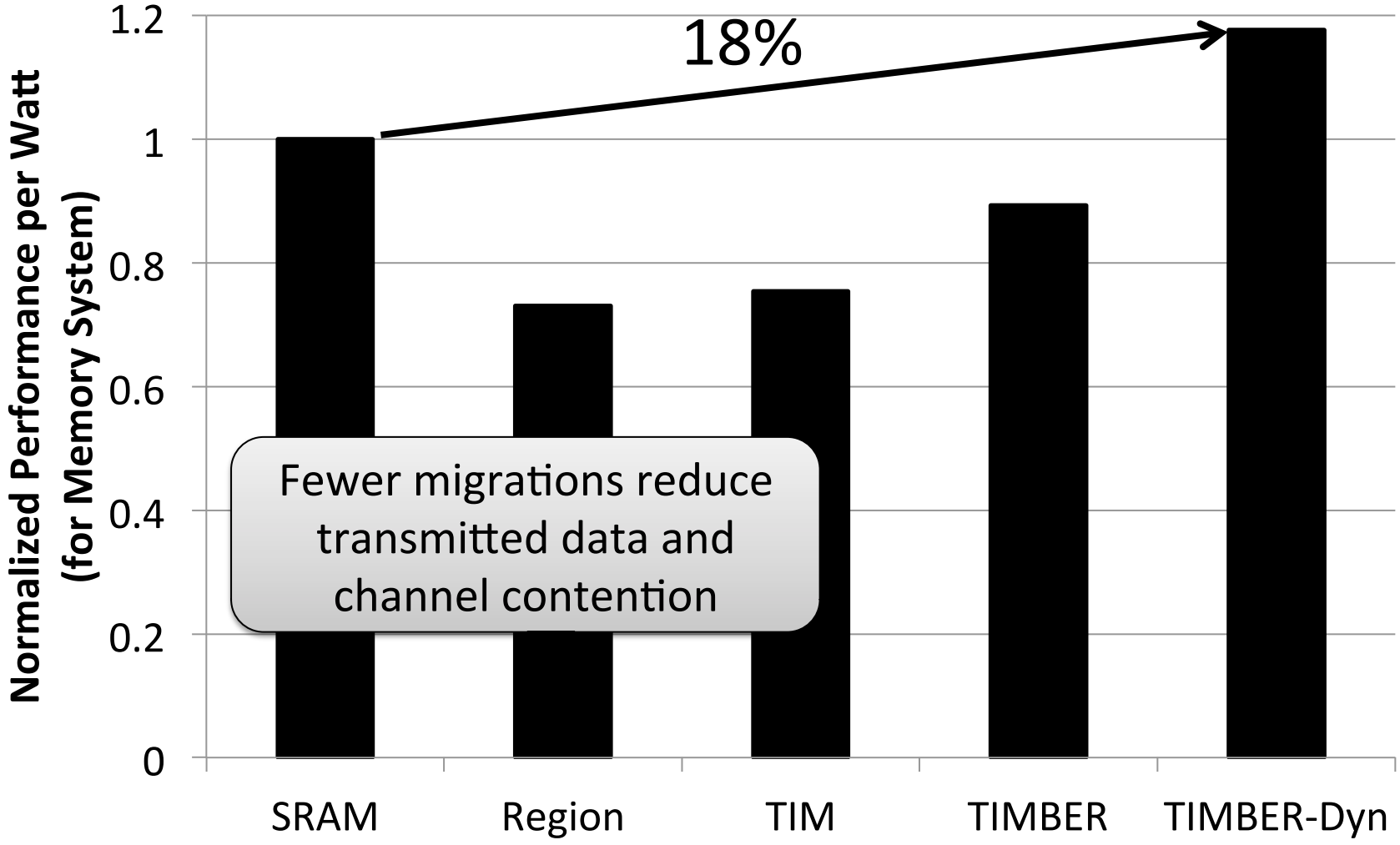


TIMBER Performance



Meza, Chang, Yoon, Mutlu, Ranganathan, "Enabling Efficient and Scalable Hybrid Memories," IEEE Comp. Arch. Letters, 2012.

TIMBER Energy Efficiency



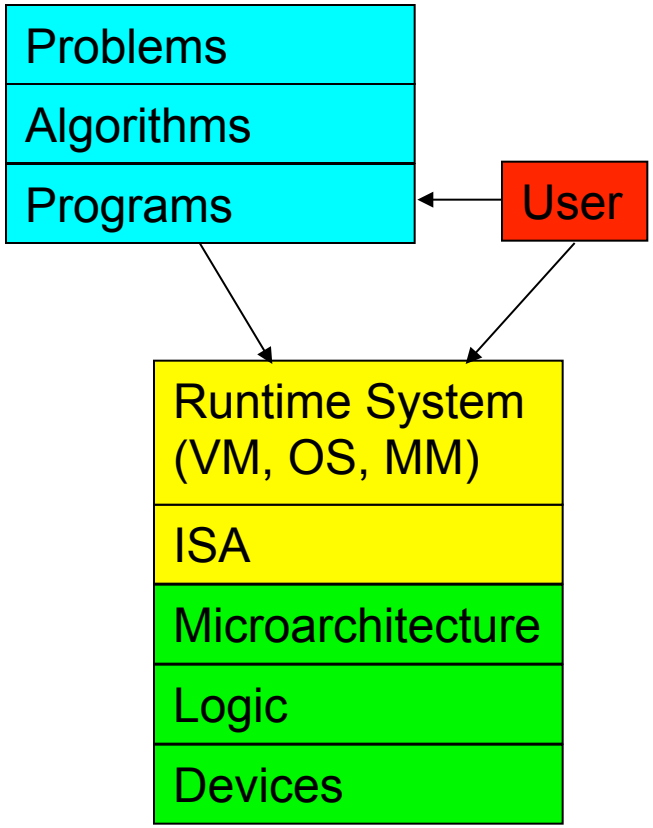
Meza, Chang, Yoon, Mutlu, Ranganathan, "Enabling Efficient and Scalable Hybrid Memories," IEEE Comp. Arch. Letters, 2012.

Enabling and Exploiting NVM: Issues

- Many issues and ideas from technology layer to algorithms layer

- Enabling NVM and hybrid memory
  - How to tolerate errors?
  - How to enable secure operation?
  - How to tolerate performance and power shortcomings?
  - How to minimize cost?

- Exploiting emerging technologies
  - How to exploit non-volatility?
  - How to minimize energy consumption?
  - How to exploit NVM on chip?



Security Challenges of Emerging Technologies

1. Limited endurance → Wearout attacks

2. Non-volatility → Data persists in memory after powerdown
   → Easy retrieval of privileged or private information

3. Multiple bits per cell → Information leakage (via side channel)

Securing Emerging Memory Technologies

1. Limited endurance → Wearout attacks
   - Better architecting of memory chips to absorb writes
   - Hybrid memory system management
   - Online wearout attack detection

2. Non-volatility → Data persists in memory after powerdown
   → Easy retrieval of privileged or private information
   - Efficient encryption/decryption of whole main memory
   - Hybrid memory system management

3. Multiple bits per cell → Information leakage (via side channel)
   - System design to hide side channel information

Agenda

- Major Trends Affecting Main Memory
- Requirements from an Ideal Main Memory System
- Opportunity: Emerging Memory Technologies
  - Background
  - PCM (or Technology X) as DRAM Replacement
  - Hybrid Memory Systems
- Conclusions
- Discussion

Summary: Memory Scaling (with NVM)

- Main memory scaling problems are a critical bottleneck for system performance, efficiency, and usability

- Solution 1: Tolerate DRAM (yesterday)
- Solution 2: Enable emerging memory technologies
  - Replace DRAM with NVM by architecting NVM chips well
  - Hybrid memory systems with automatic data management

- An exciting topic with many other solution directions & ideas
  - Hardware/software/device cooperation essential
  - Memory, storage, controller, software/app co-design needed
  - Coordinated management of persistent memory and storage
  - Application and hardware cooperative management of NVM

Scalable Many-Core Memory Systems Topic 2: Emerging Technologies and Hybrid Memories

Prof. Onur Mutlu
http://www.ece.cmu.edu/~omutlu
onur@cmu.edu

HiPEAC ACACES Summer School 2013
July 15-19, 2013

Carnegie Mellon

Additional Material

Overview Papers on Two Topics

- Merging of Memory and Storage
  - Justin Meza, Yixin Luo, Samira Khan, Jishen Zhao, Yuan Xie, and Onur Mutlu,
    "A Case for Efficient Hardware-Software Cooperative Management of Storage and Memory"
    *Proceedings of the 5th Workshop on Energy-Efficient Design (**WEED**)*, Tel-Aviv, Israel, June 2013. Slides (pptx) Slides (pdf)

- Flash Memory Scaling
  - Yu Cai, Gulay Yalcin, Onur Mutlu, Erich F. Haratsch, Adrian Cristal, Osman Unsal, and Ken Mai,
    "Error Analysis and Retention-Aware Error Management for NAND Flash Memory"
    *Intel Technology Journal (**ITJ**) Special Issue on Memory Resiliency*, Vol. 17, No. 1, May 2013.