18-742 Fall 2012
Parallel Computer Architecture
Lecture 7: Emerging Memory Technologies

Prof. Onur Mutlu
Carnegie Mellon University

9/21/2012

Reminder: Review Assignments

- Due: Friday, September 21, 11:59pm.
- Smith, "Architecture and applications of the HEP multiprocessor computer system," SPIE 1981.
- Tullsen et al., "Exploiting Choice: Instruction Fetch and Issue on an Implementable Simultaneous Multithreading Processor," ISCA 1996.
- Chappell et al., "Simultaneous Subordinate Microthreading (SSMT)," ISCA 1999.
- Reinhardt and Mukherjee, "Transient Fault Detection via Simultaneous Multithreading," ISCA 2000.

Last Lecture

- More on Asymmetric Multi-Core

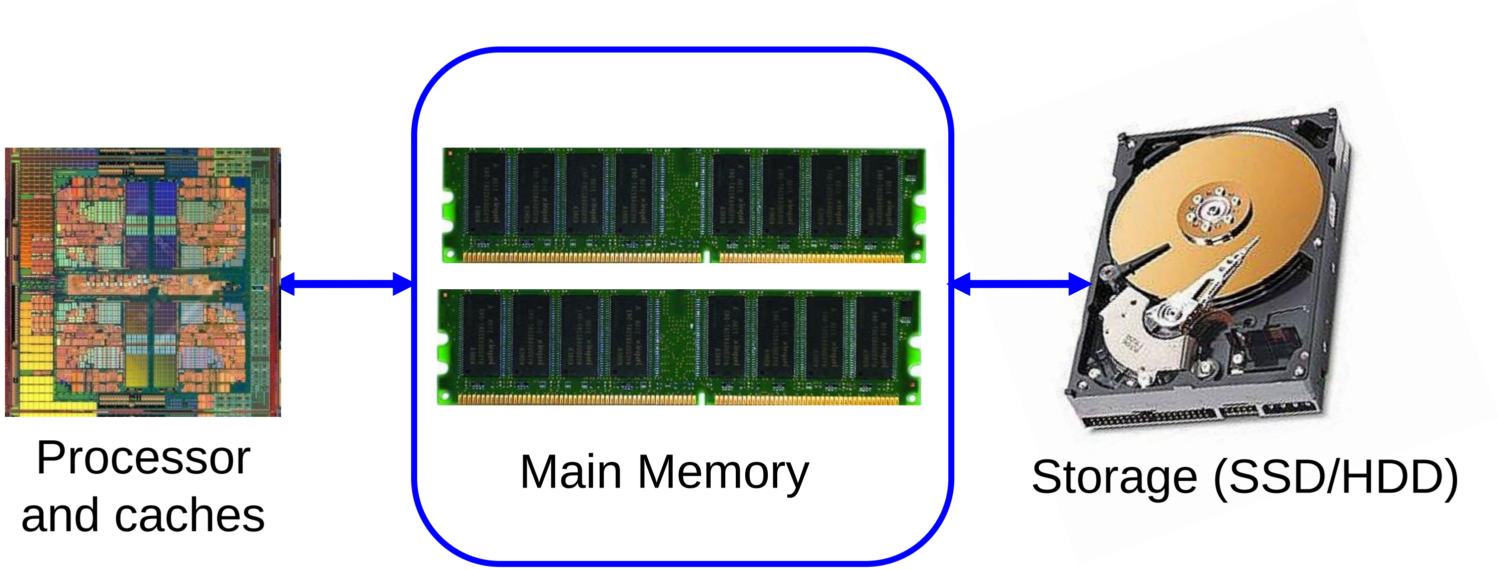
- And, Asymmetry in General

Today

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

Major Trends Affecting Main Memory

The Main Memory System



Processor and caches

Main Memory

Storage (SSD/HDD)

- **Main memory is a critical component of all computing systems**: server, mobile, embedded, desktop, sensor

- **Main memory system must scale** (in *size*, *technology*, *efficiency*, *cost*, and *management algorithms*) to maintain performance growth and technology scaling benefits

State of the Main Memory System

- Recent technology, architecture, and application trends
  - lead to new requirements from the memory system
  - exacerbate old requirements from the memory system

- DRAM alone is (will be) unlikely to satisfy all requirements

- Some emerging non-volatile memory technologies (e.g., PCM) appear promising to satisfy these requirements
  - and enable new opportunities

- We need to rethink the main memory system
  - to fix DRAM issues and enable emerging technologies
  - to satisfy all new and (exacerbated) old requirements

Major Trends Affecting Main Memory (I)

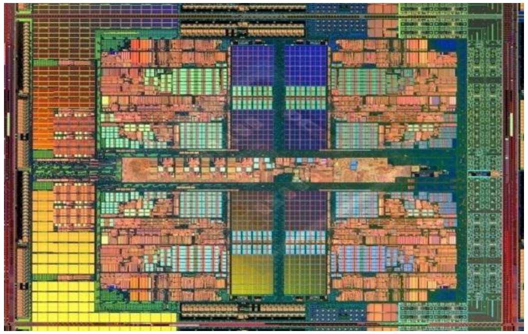
- Need for main memory capacity and bandwidth increasing

- Main memory energy/power is a key system design concern

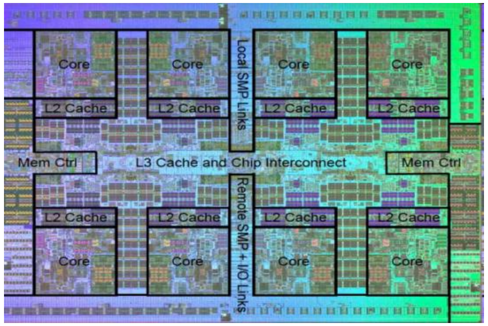
- DRAM technology scaling is ending

Demand for Memory Capacity

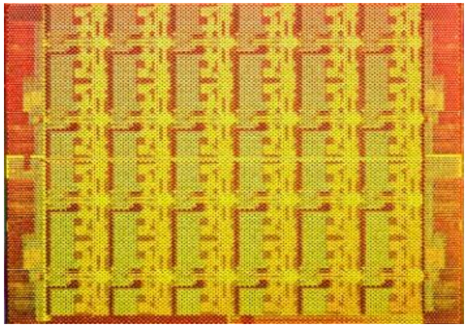
- More cores → More concurrency → Larger working set



AMD Barcelona: 4 cores



IBM Power7: 8 cores



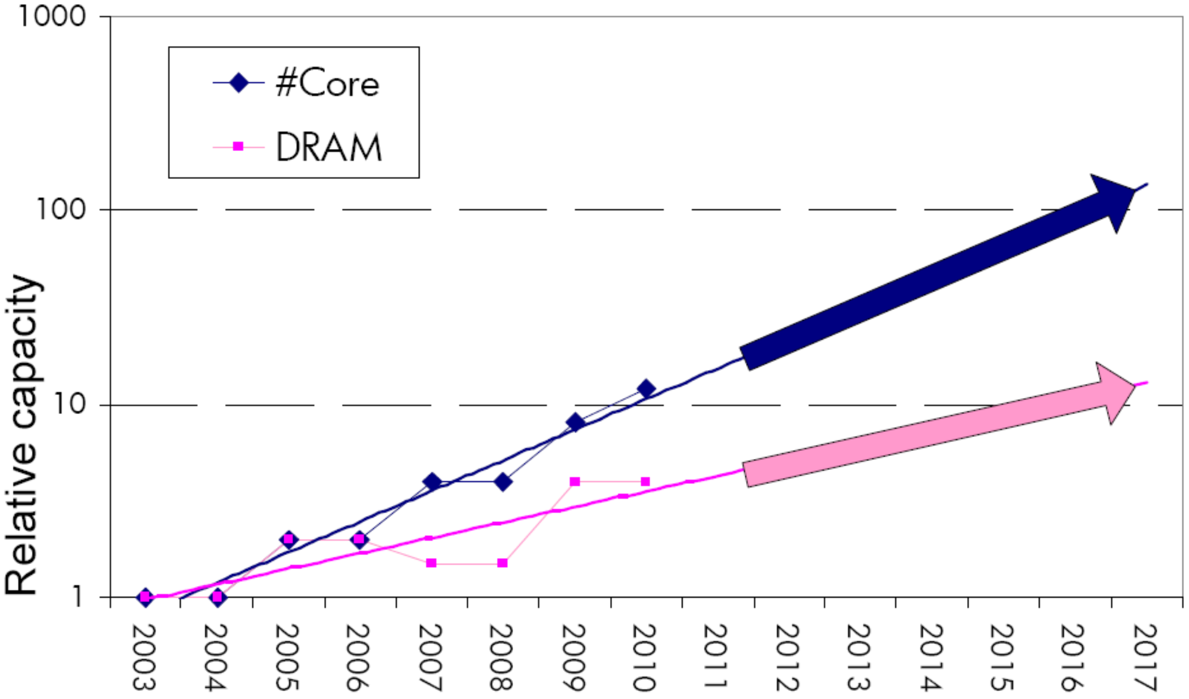
Intel SCC: 48 cores

- Emerging applications are data-intensive

- Many applications/virtual machines (will) share main memory
  - Cloud computing/servers: Consolidation to improve efficiency
  - GP-GPUs: Many threads from multiple parallel applications
  - Mobile: Interactive + non-interactive consolidation

The Memory Capacity Gap

Core count doubling ~ every 2 years
DRAM DIMM capacity doubling ~ every 3 years



Source: Lim et al., ISCA 2009.

- Memory capacity per core expected to drop by 30% every two years

Major Trends Affecting Main Memory (II)

- Need for main memory capacity and bandwidth increasing
  - Multi-core: increasing number of cores
  - Data-intensive applications: increasing demand/hunger for data
  - Consolidation: Cloud computing, GPUs, mobile

- Main memory energy/power is a key system design concern

- DRAM technology scaling is ending

Major Trends Affecting Main Memory (III)

- Need for main memory capacity and bandwidth increasing

- Main memory energy/power is a key system design concern
  - IBM servers: ~50% energy spent in off-chip memory hierarchy [Lefurgy, IEEE Computer 2003]
  - DRAM consumes power when idle and needs periodic refresh

- DRAM technology scaling is ending

Major Trends Affecting Main Memory (IV)

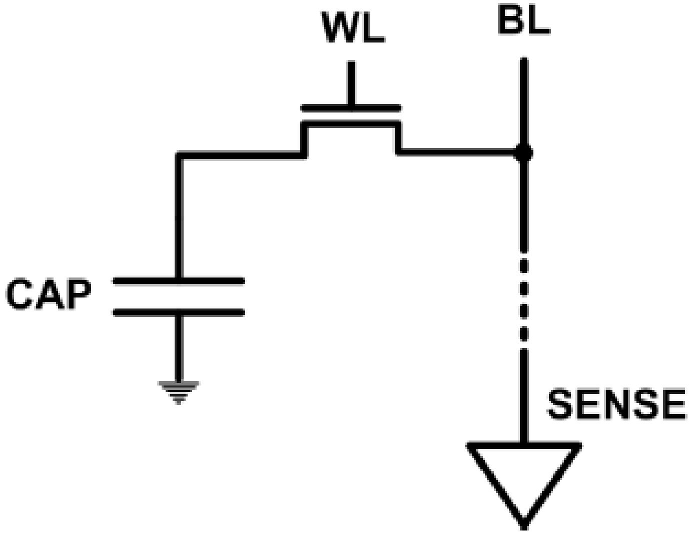
- Need for main memory capacity and bandwidth increasing

- Main memory energy/power is a key system design concern

- DRAM technology scaling is ending
  - ITRS projects DRAM will not scale easily below 40nm
  - Scaling has provided many benefits:
    - higher capacity, higher density, lower cost, lower energy

The DRAM Scaling Problem

- DRAM stores charge in a capacitor (charge-based memory)
  - Capacitor must be large enough for reliable sensing
  - Scaling beyond 40-35nm (2013) is challenging [ITRS, 2009]



- DRAM capacity, cost, and energy/power hard to scale

Trends: Problems with DRAM as Main Memory

- Need for main memory capacity and bandwidth increasing
  - DRAM capacity hard to scale

- Main memory energy/power is a key system design concern
  - DRAM consumes high power due to leakage and refresh

- DRAM technology scaling is ending
  - DRAM capacity, cost, and energy/power hard to scale

Requirements from an Ideal Main Memory System

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

Opportunity: Emerging Memory Technologies

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?

Opportunity: Emerging Memory Technologies

- Background
- PCM (or Technology X) as DRAM Replacement
- Hybrid Memory Systems

Charge vs. Resistive Memories

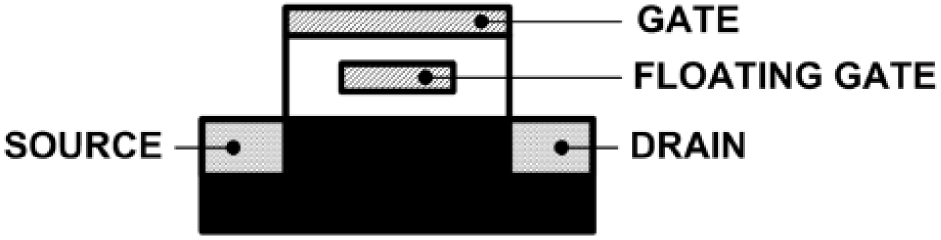
- Charge Memory (e.g., DRAM, Flash)
  - Write data by capturing charge Q
  - Read data by detecting voltage V

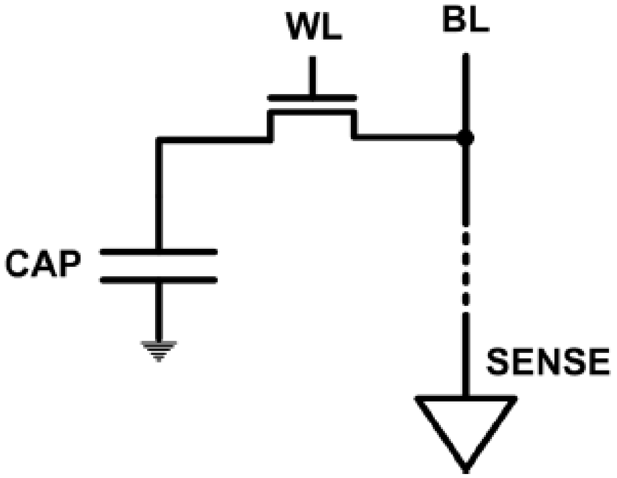
- Resistive Memory (e.g., PCM, STT-MRAM, memristors)
  - Write data by pulsing current dQ/dt
  - Read data by detecting resistance R

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

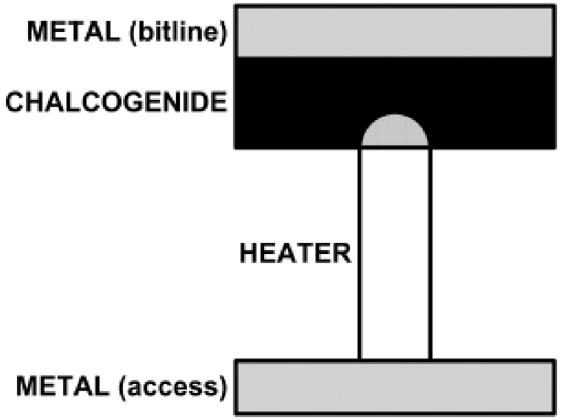
- PCM
  - Inject current to change material phase
  - Resistance determined by phase

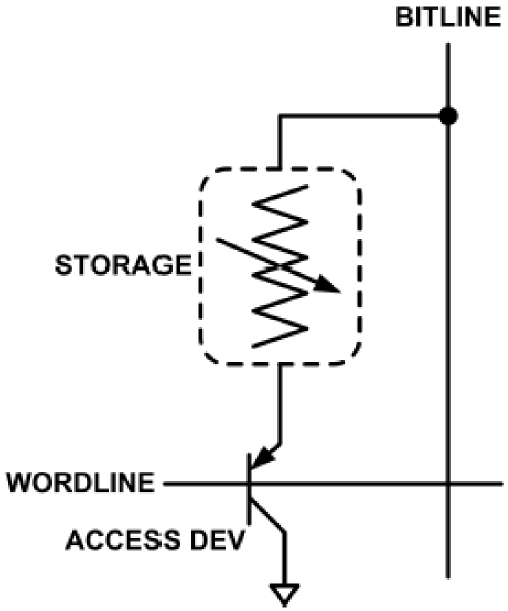
- STT-MRAM
  - Inject current to change magnet polarity
  - Resistance determined by polarity

- 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

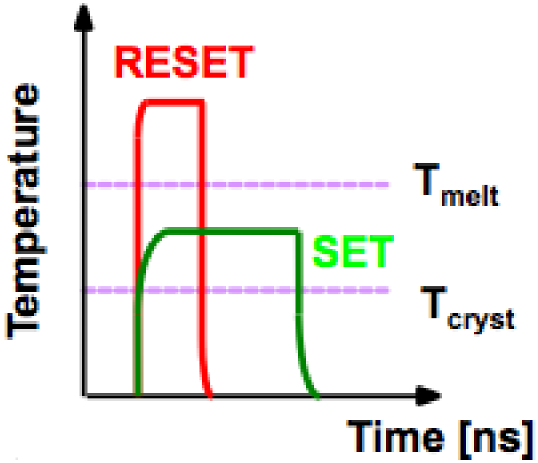


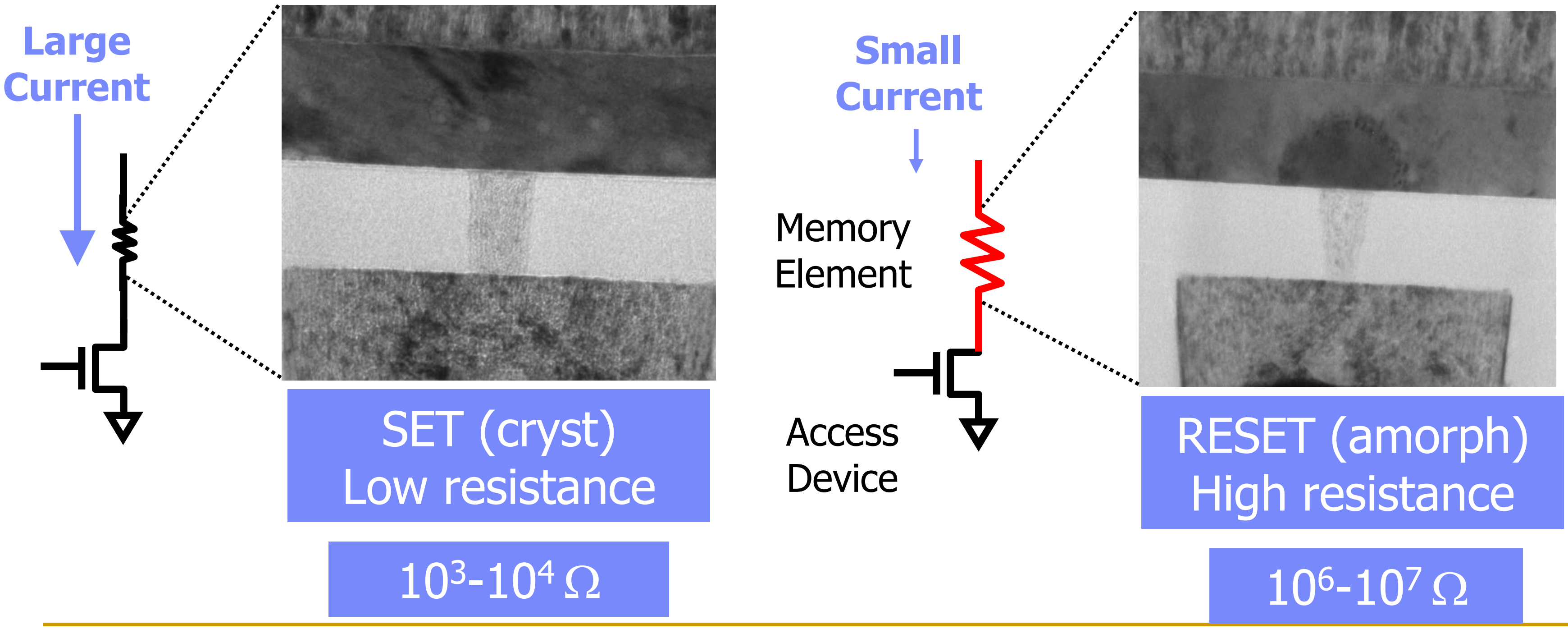


PCM is resistive memory: High resistance (0), Low resistance (1)
PCM cell can be switched between states reliably and quickly

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





SET (cryst)
Low resistance

10^3-10^4 Ω

RESET (amorph)
High resistance

10^6-10^7 Ω

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

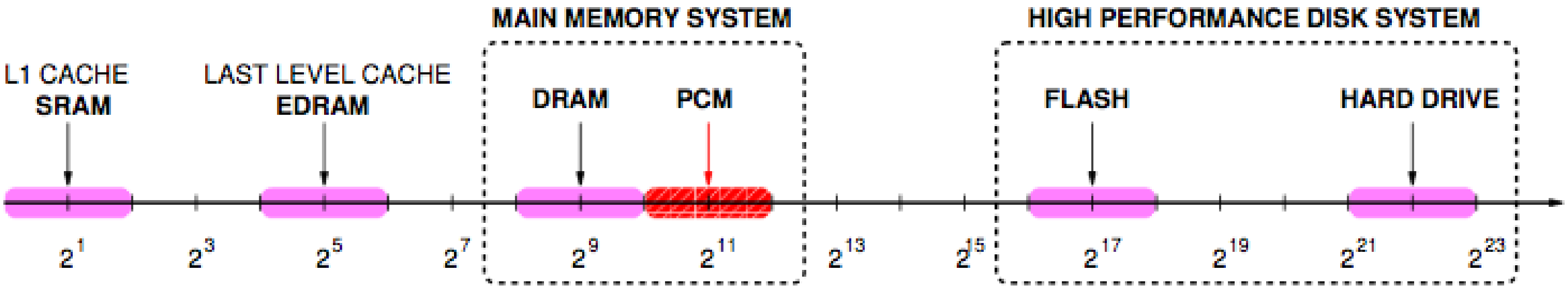
Phase Change Memory Properties

- 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.

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?

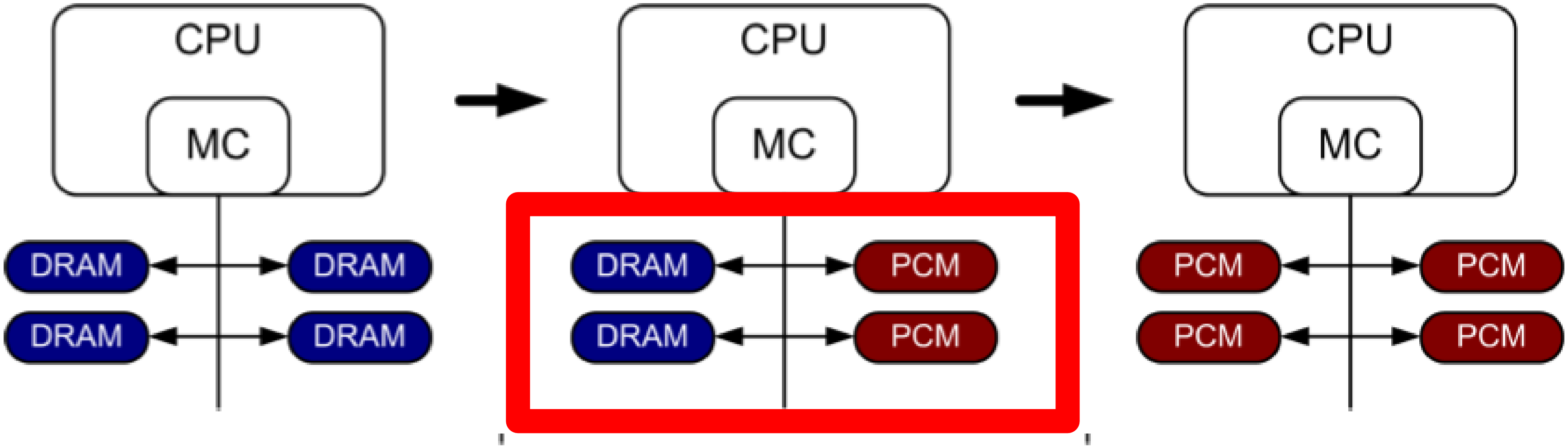
- How to minimize amount of DRAM in the system?

- 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: Research 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

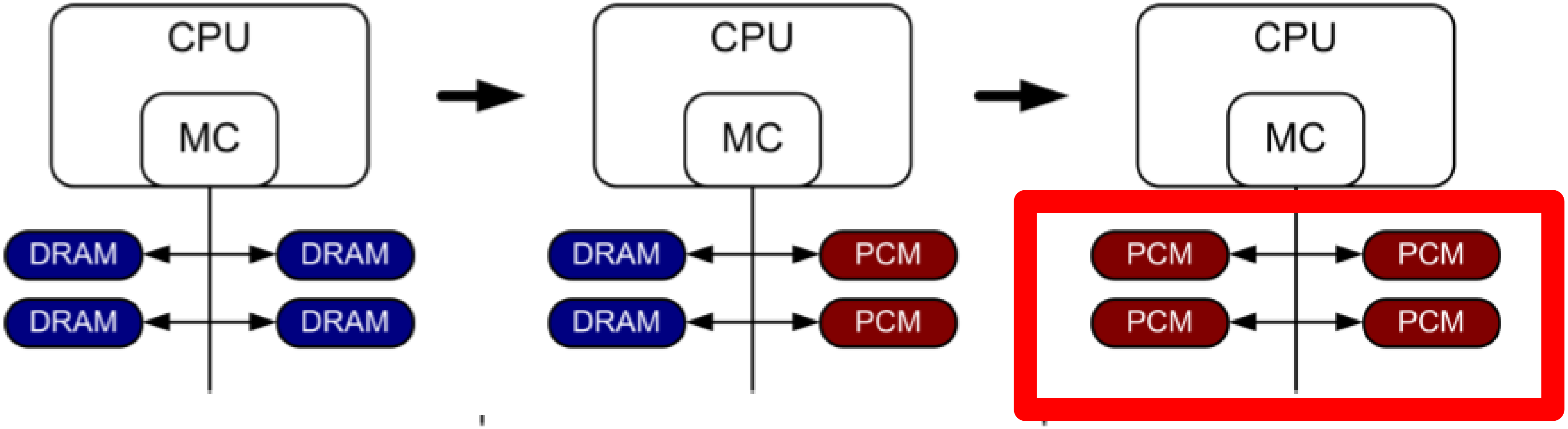
- Design of PCM/DRAM chips and modules
  - Rethink the design of PCM/DRAM with new requirements

Opportunity: Emerging Memory Technologies

- Background
- PCM (or Technology X) as DRAM Replacement
- Hybrid Memory Systems

PCM-based Main Memory (II)

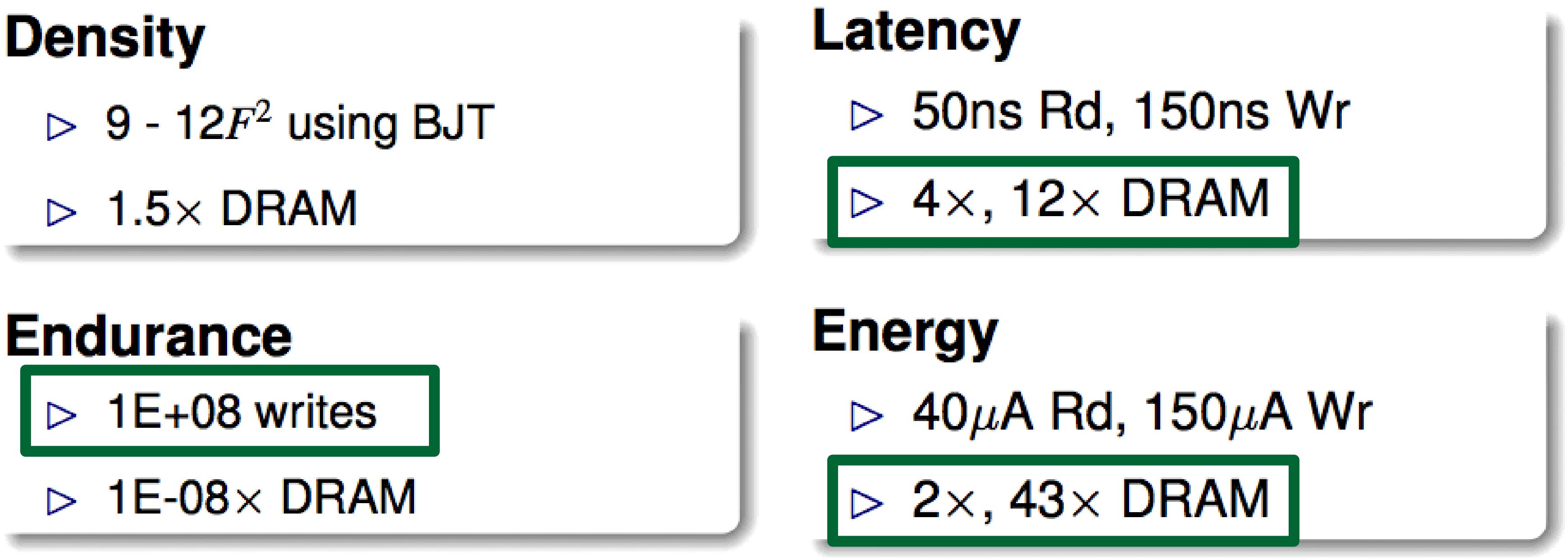
- How should PCM-based (main) memory be organized?



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

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

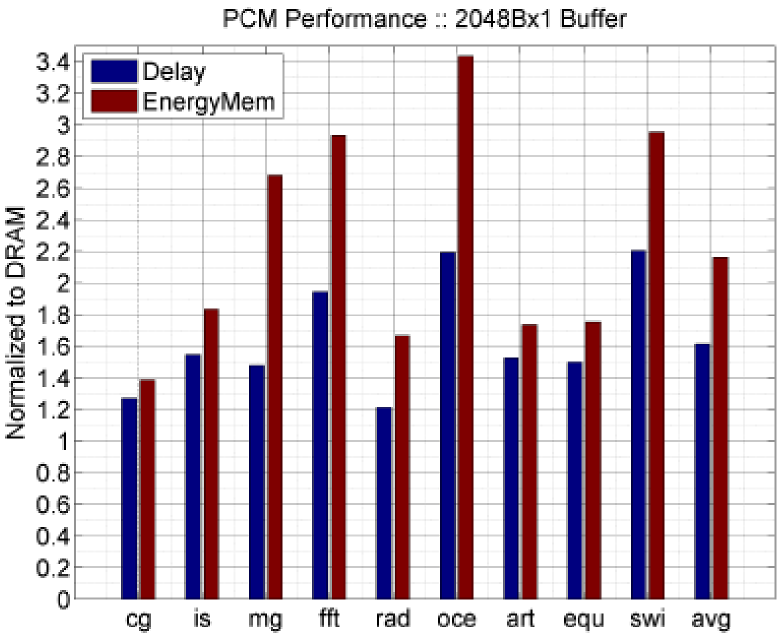
- ▷ 1E+08 writes
- ▷ 1E-08× DRAM

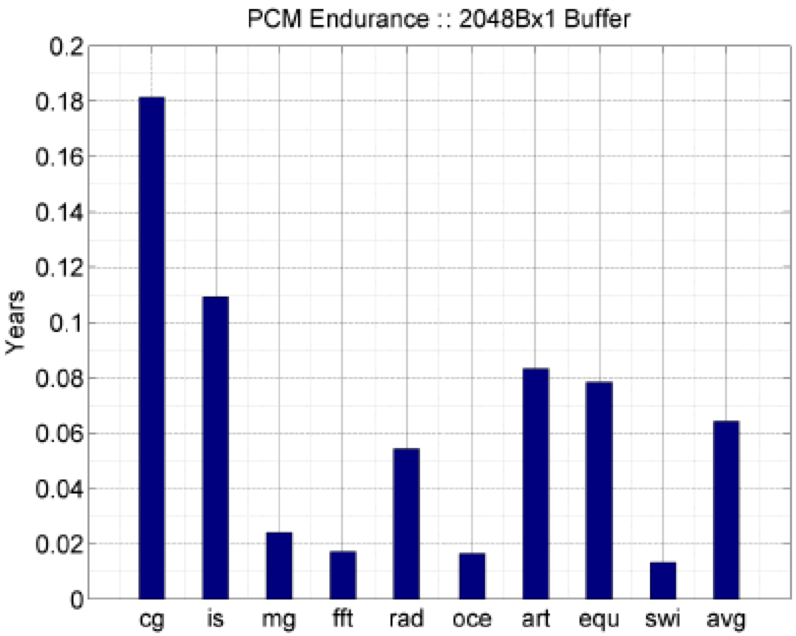
Energy

- ▷ 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



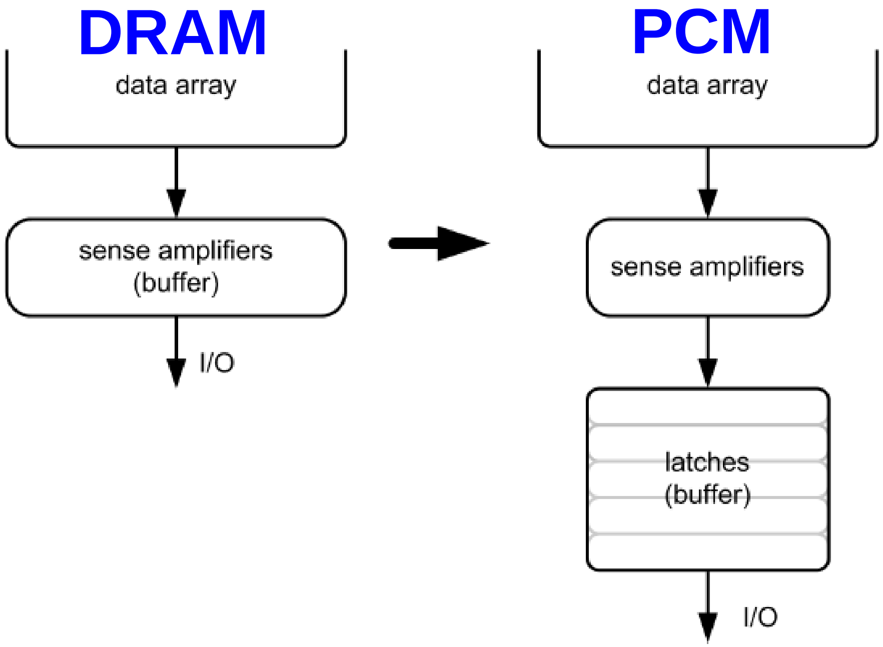


- Lee, Ipek, Mutlu, Burger, "Architecting Phase Change Memory as a Scalable DRAM Alternative," ISCA 2009.

Architecting PCM to Mitigate Shortcomings

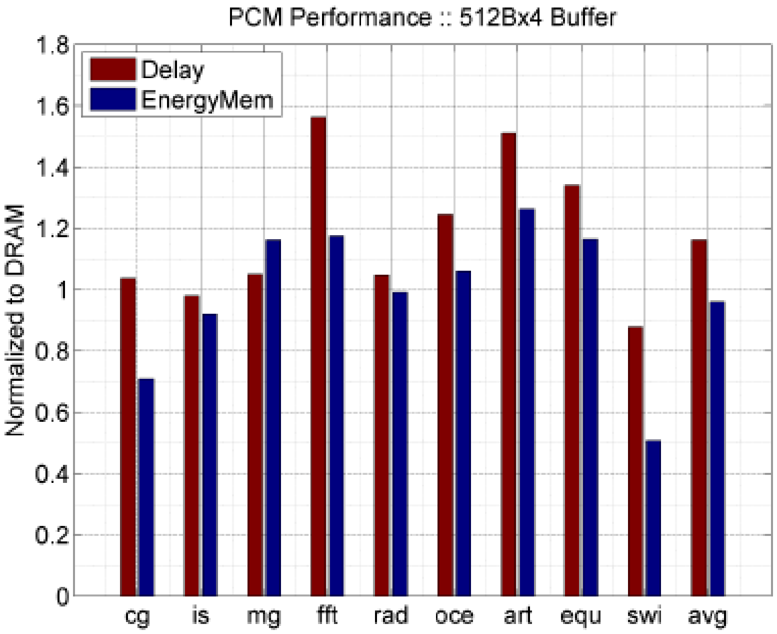
- Idea 1: Use multiple narrow row buffers in each PCM chip
  → Reduces array reads/writes → better endurance, latency, energy

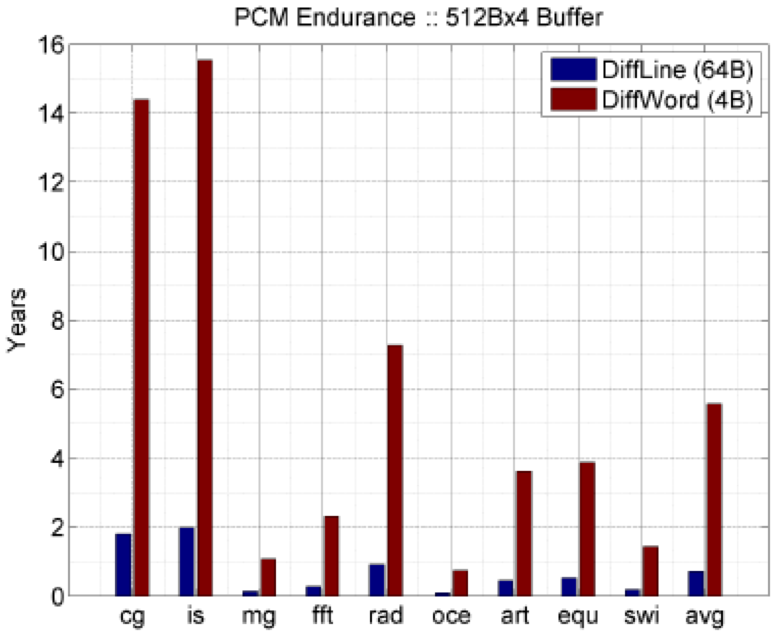
- Idea 2: Write into array at cache block or word granularity
  → Reduces unnecessary wear



Results: Architected PCM as Main Memory

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



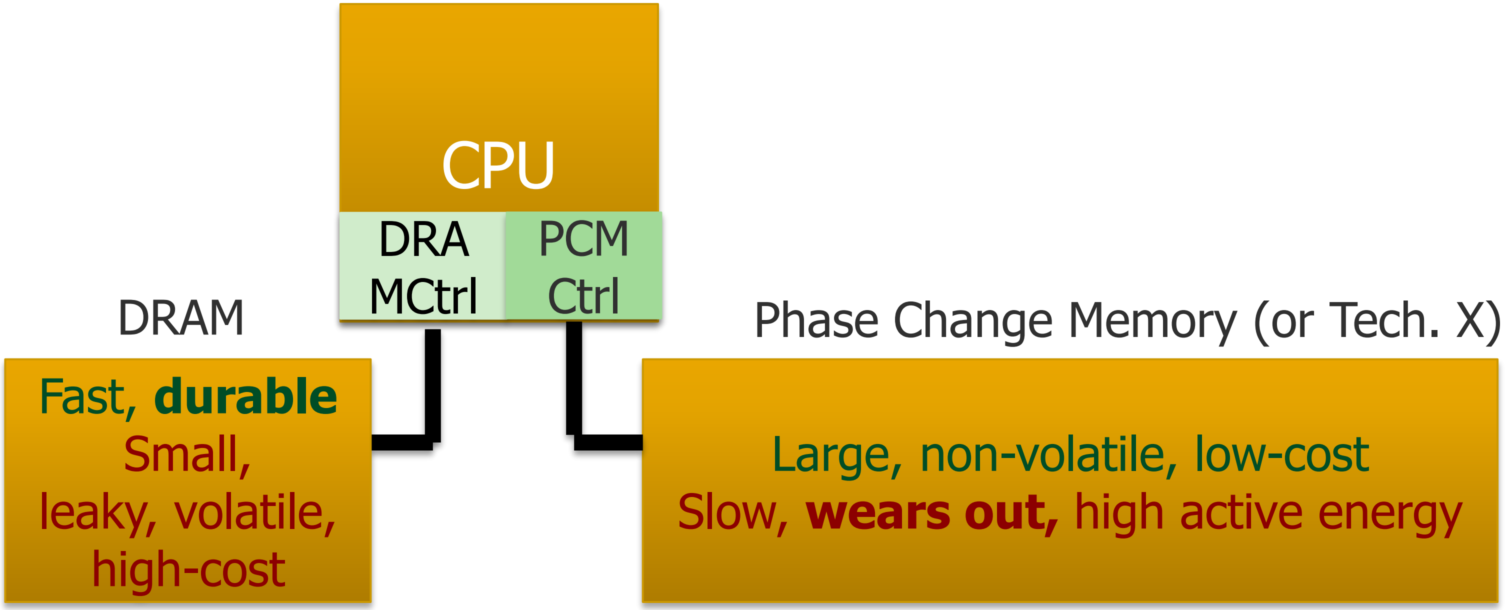


- Caveat 1: Worst-case lifetime is much shorter (no guarantees)
- Caveat 2: Intensive applications see large performance and energy hits
- Caveat 3: Optimistic PCM parameters?

Opportunity: Emerging Memory Technologies

- Background
- PCM (or Technology X) as DRAM Replacement
- Hybrid Memory Systems

Hybrid Memory Systems

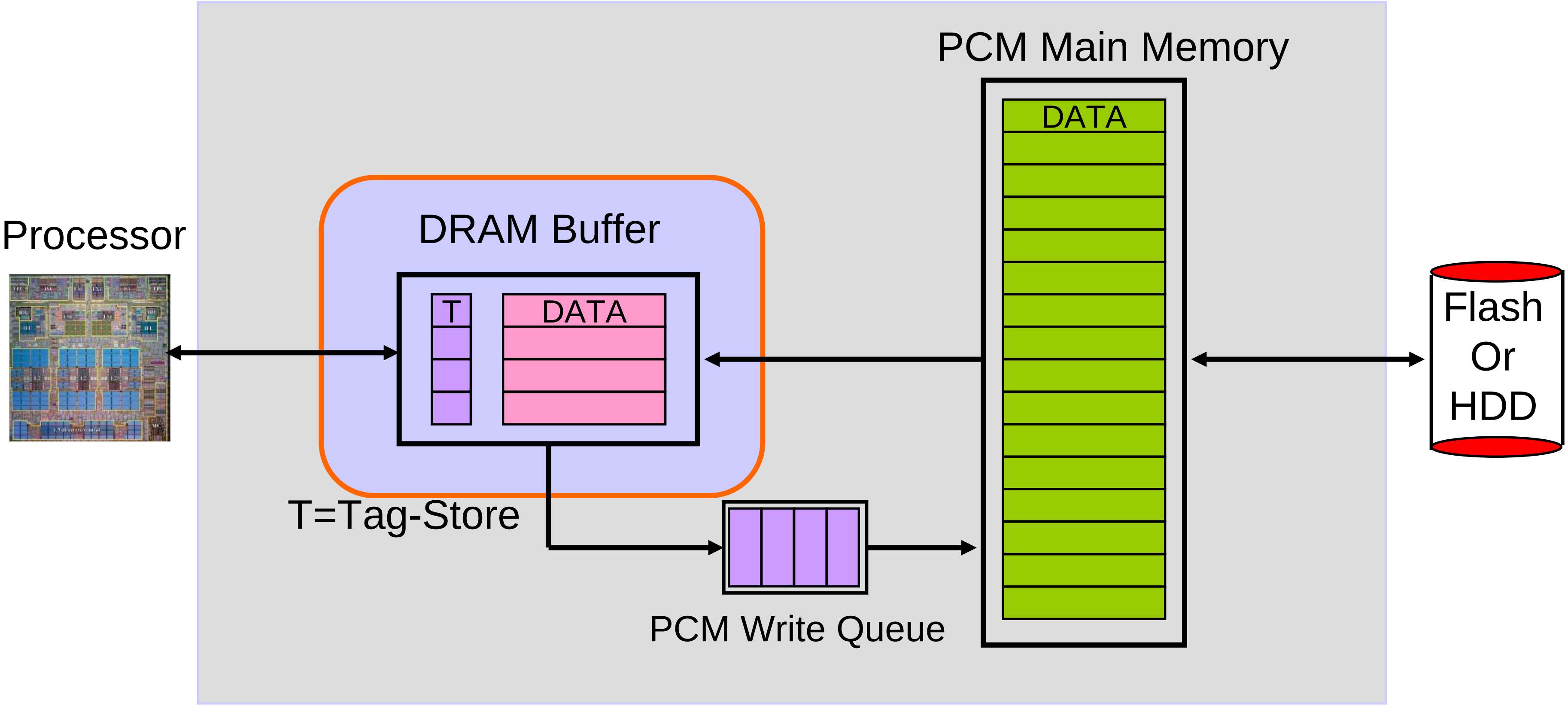

CPU
DRA MCtrl
PCM Ctrl
DRAM
Phase Change Memory (or Tech. X)
Fast, **durable**
Small,
leaky, volatile,
high-cost
Large, non-volatile, low-cost
Slow, **wears out,** high active energy

Hardware/software manage data allocation and movement
to achieve the best of multiple technologies
(5-9 years of average lifetime)

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

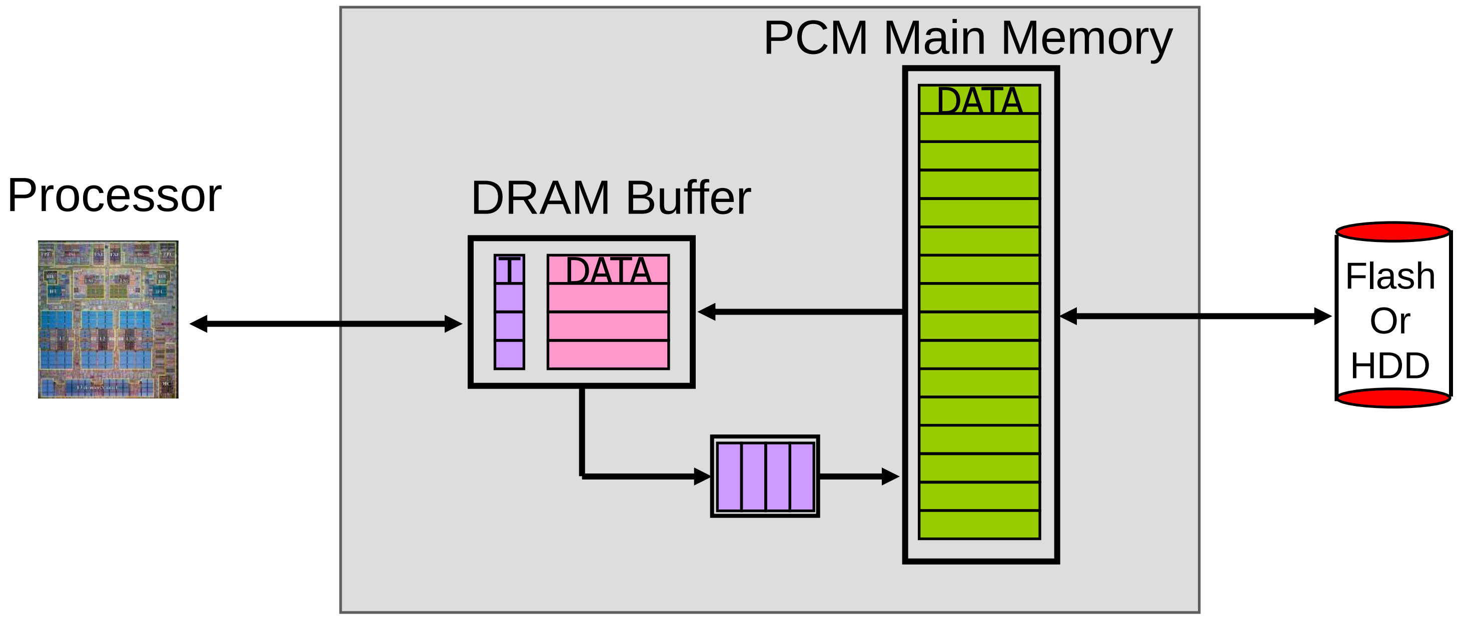
DRAM as PCM Cache

- Goal: Achieve the best of both DRAM and PCM/NVM
  - Minimize amount of DRAM w/o sacrificing performance, endurance
  - DRAM as cache to tolerate PCM latency and write bandwidth
  - PCM as main memory to provide large capacity at good cost and power



Write Filtering Techniques

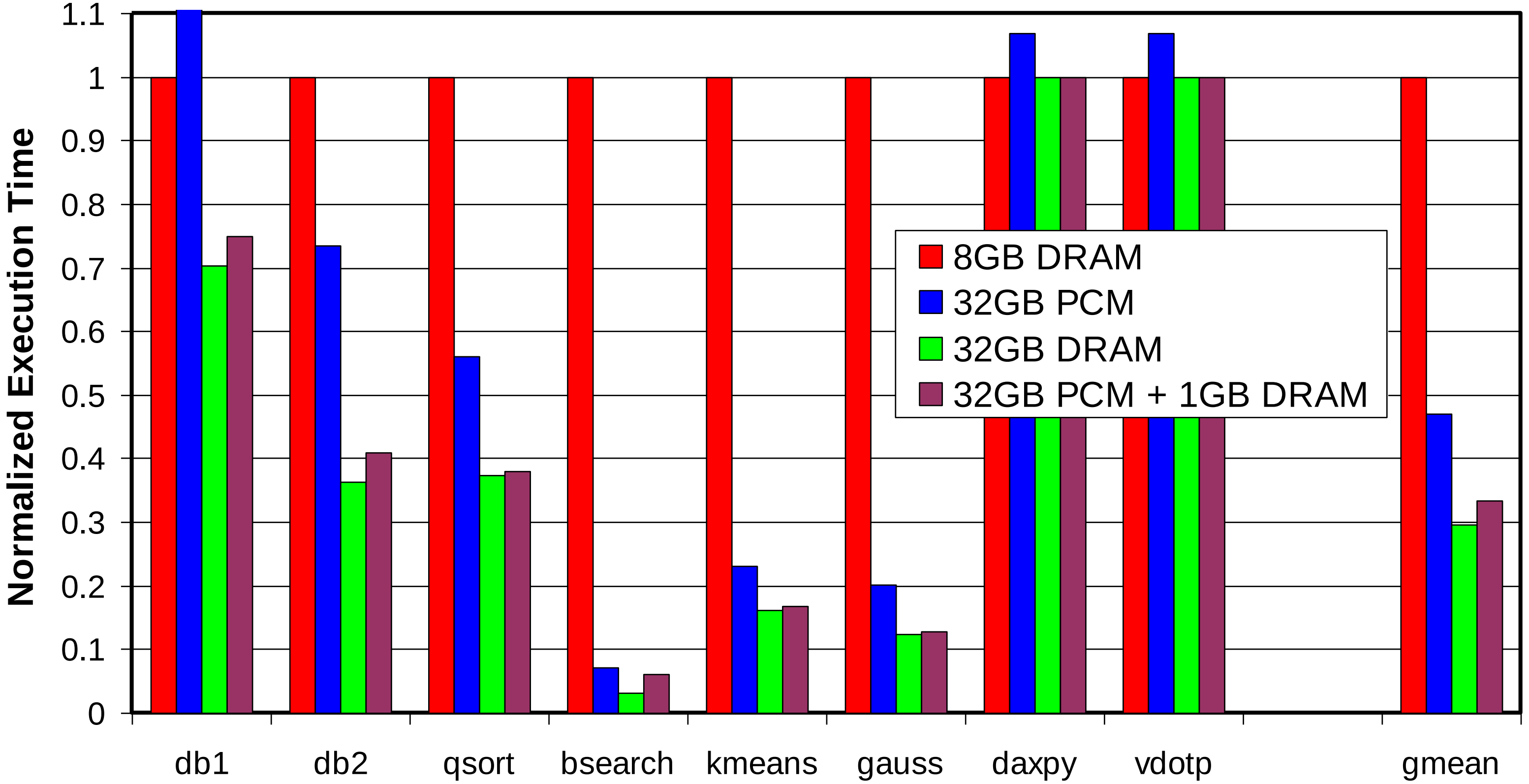
- Lazy Write: Pages from disk installed only in DRAM, not PCM
- Partial Writes: Only dirty lines from DRAM page written back
- Page Bypass: Discard pages with poor reuse on DRAM eviction



- Qureshi et al., "Scalable high performance main memory system using phase-change memory technology," ISCA 2009.

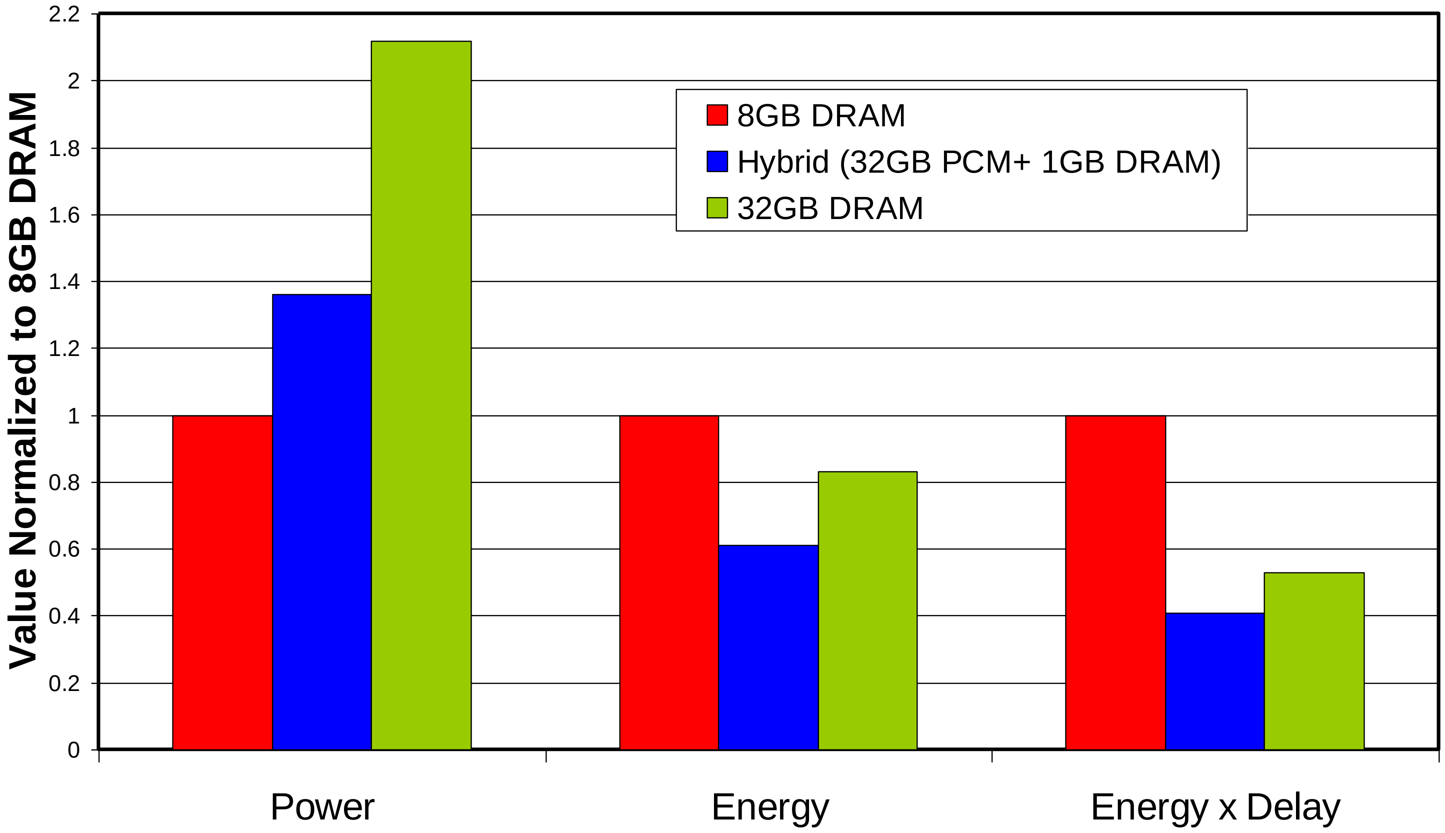
Results: DRAM as PCM Cache (I)

- Simulation of 16-core system, 8GB DRAM main-memory at 320 cycles, HDD (2 ms) with Flash (32 us) with Flash hit-rate of 99%
- Assumption: PCM 4x denser, 4x slower than DRAM
- DRAM block size = PCM page size (4kB)



Results: DRAM as PCM Cache (II)

- PCM-DRAM Hybrid performs similarly to similar-size DRAM
- Significant power and energy savings with PCM-DRAM Hybrid
- Average lifetime: 9.7 years (no guarantees)



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?

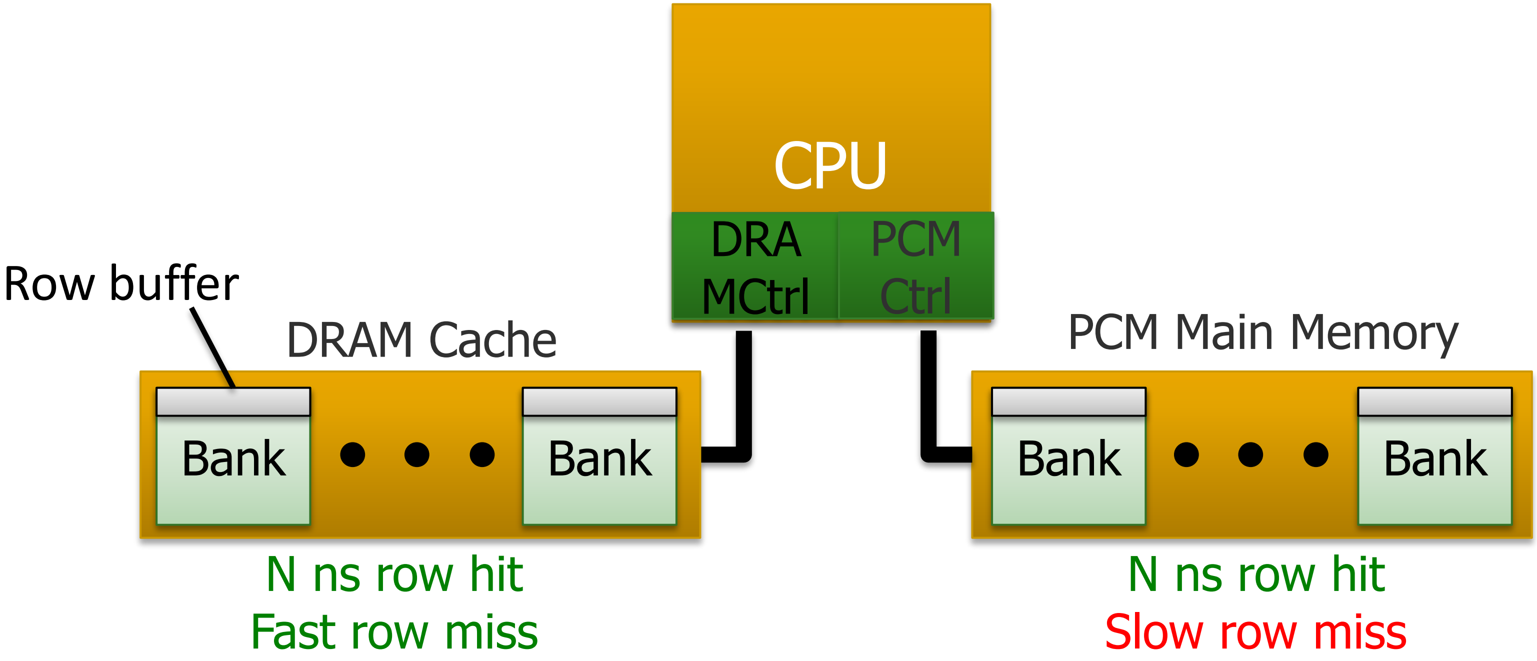
- Two idea directions:
  - Locality-aware data placement **[Yoon+ , ICCD 2012]**
  - Cheap tag stores and dynamic granularity **[Meza+, IEEE CAL 2012]**

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

DRAM vs. PCM: An Observation

- DRAM and PCM both have row buffers
- Row buffer hit latency **similar** in DRAM and PCM
- Row buffer miss latency **small** in DRAM, **large** in PCM



- Accessing the row buffer in PCM is fast
- What incurs high latency is the PCM array access → avoid this

Row-Locality-Aware Data Placement

- Idea: Cache in DRAM only those rows that
  - Frequently cause row buffer conflicts → because row-conflict latency is smaller in DRAM
  - Are reused many times → to reduce cache pollution and bandwidth waste

- Simplified rule of thumb:
  - Streaming accesses: Better to place in PCM
  - Other accesses (with some reuse): Better to place in DRAM

- Bridges half of the performance gap between all-DRAM and all-PCM memory on memory-intensive workloads

- Yoon et al., "Row Buffer Locality Aware Caching Policies for Hybrid Memories," ICCD 2012.

Row-Locality-Aware Data Placement: Mechanism

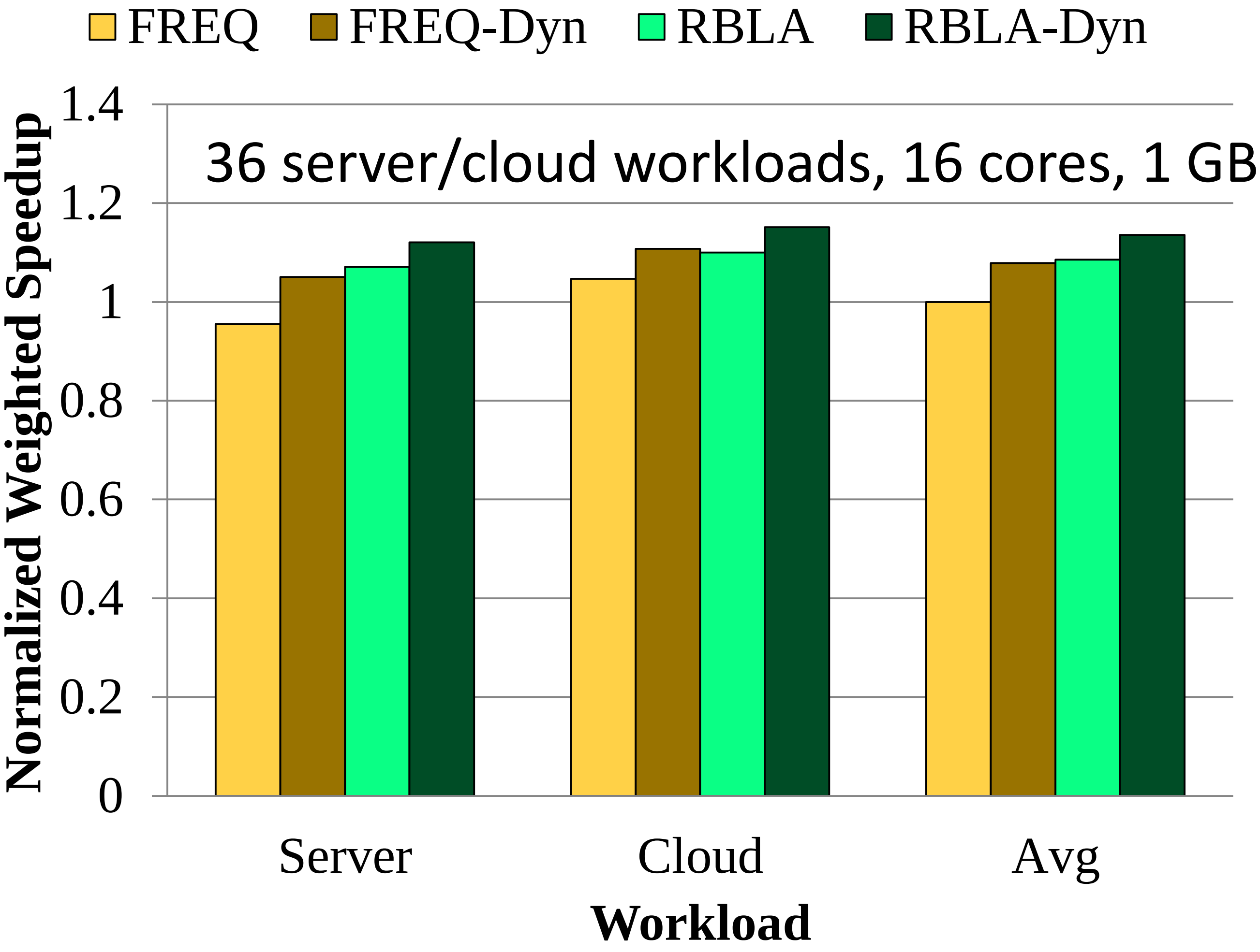
- For a subset of rows in PCM, memory controller:
  - Tracks **row conflicts** as a predictor of future locality
  - Tracks **accesses** as a predictor of future reuse
- Cache a row in DRAM if its row conflict and access counts are greater than certain thresholds
- Determine thresholds dynamically to adjust to application/workload characteristics
  - Simple cost/benefit analysis every fixed interval

Evaluation Methodology

- Core model
  - 3-wide issue with 128-entry instruction window
  - 32 KB L1 D-cache per core
  - 512 KB L2 cache per core

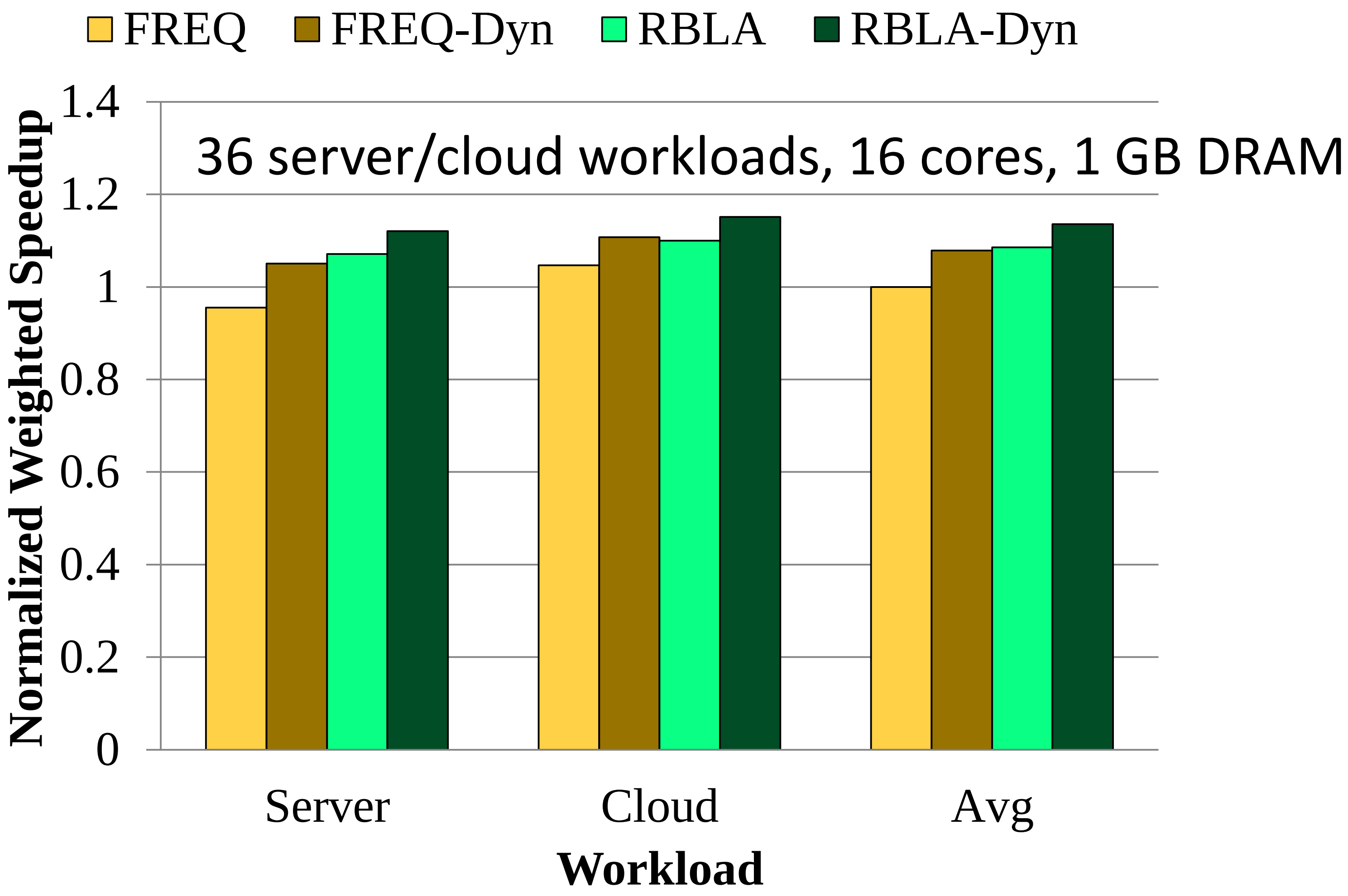
- Memory model
  - 1 GB DRAM Cache / 16 GB PCM
  - Separate memory controllers, 8 banks per device
  - Row buffer hit: 40 ns
  - Row buffer miss: 80 ns (DRAM); 128, 368 ns (PCM)
  - Cache data at 4 KB row granularity

Performance



36 server/cloud workloads, 16 cores, 1 GB DRAM

Performance



Benefits come from: (1) better row buffer locality in PCM (2) reduced bandwidth consumption and reduced pollution

Row-Locality-Aware Data Placement: Results

- Heterogeneous DRAM cache + PCM memory with locality-aware data placement on a 16-core system

- Compared to **all PCM** main memory
  - 31% performance improvement

- Compared to an **all DRAM** main memory
  - Within 29% of performance

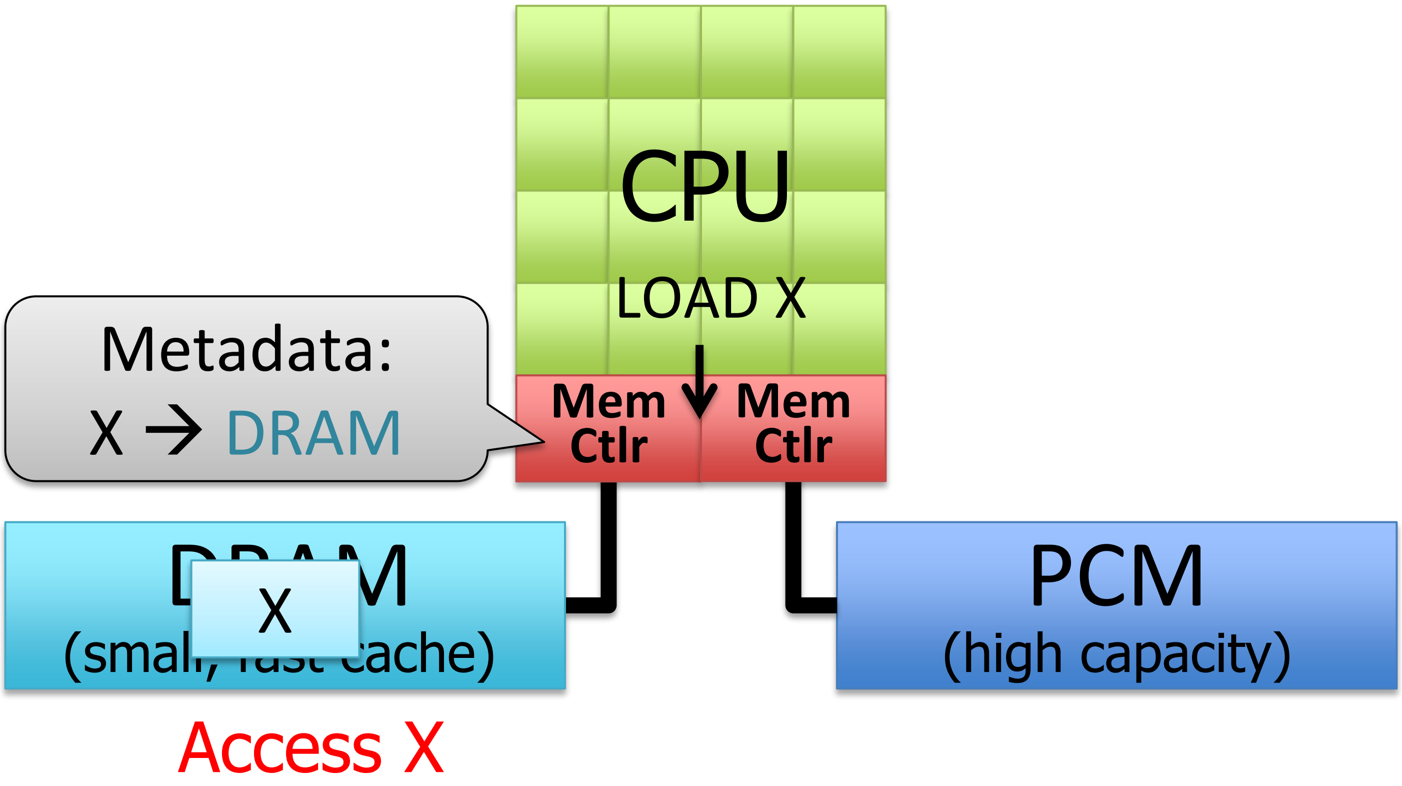
- Power, energy, endurance evaluations in paper
  - Yoon et al., "Row Buffer Locality Aware Caching Policies for Hybrid Memories," ICCD 2012.

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

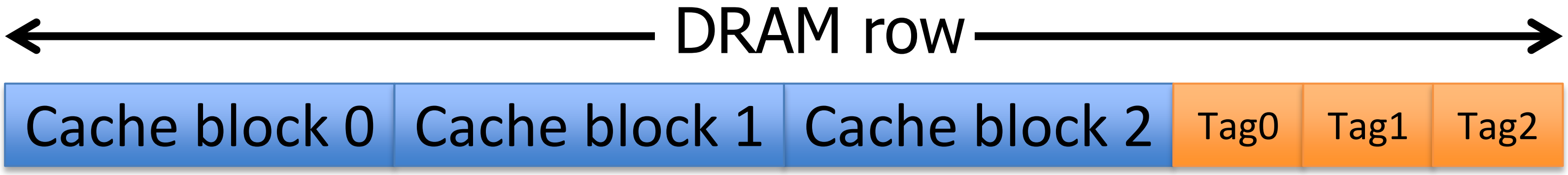
The Problem with Large DRAM Caches

- 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

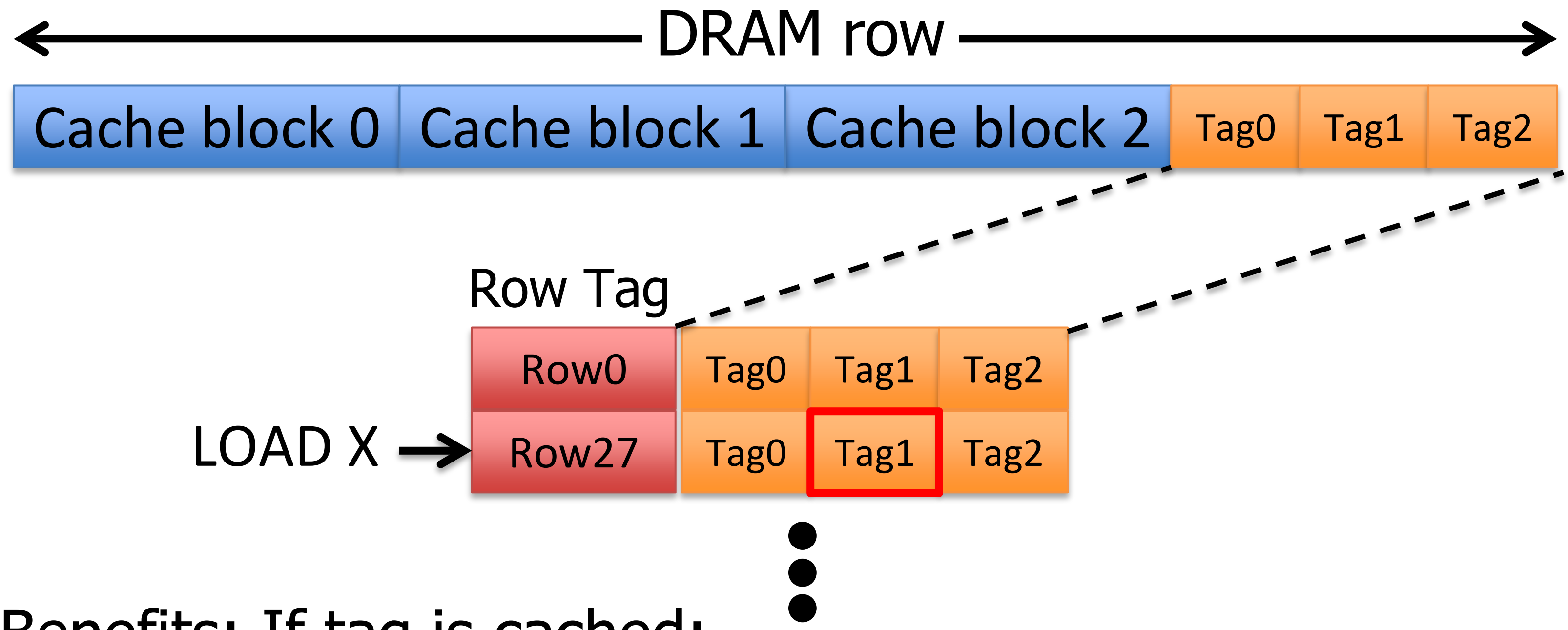
Idea 3: Dynamic Data Transfer Granularity

- 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

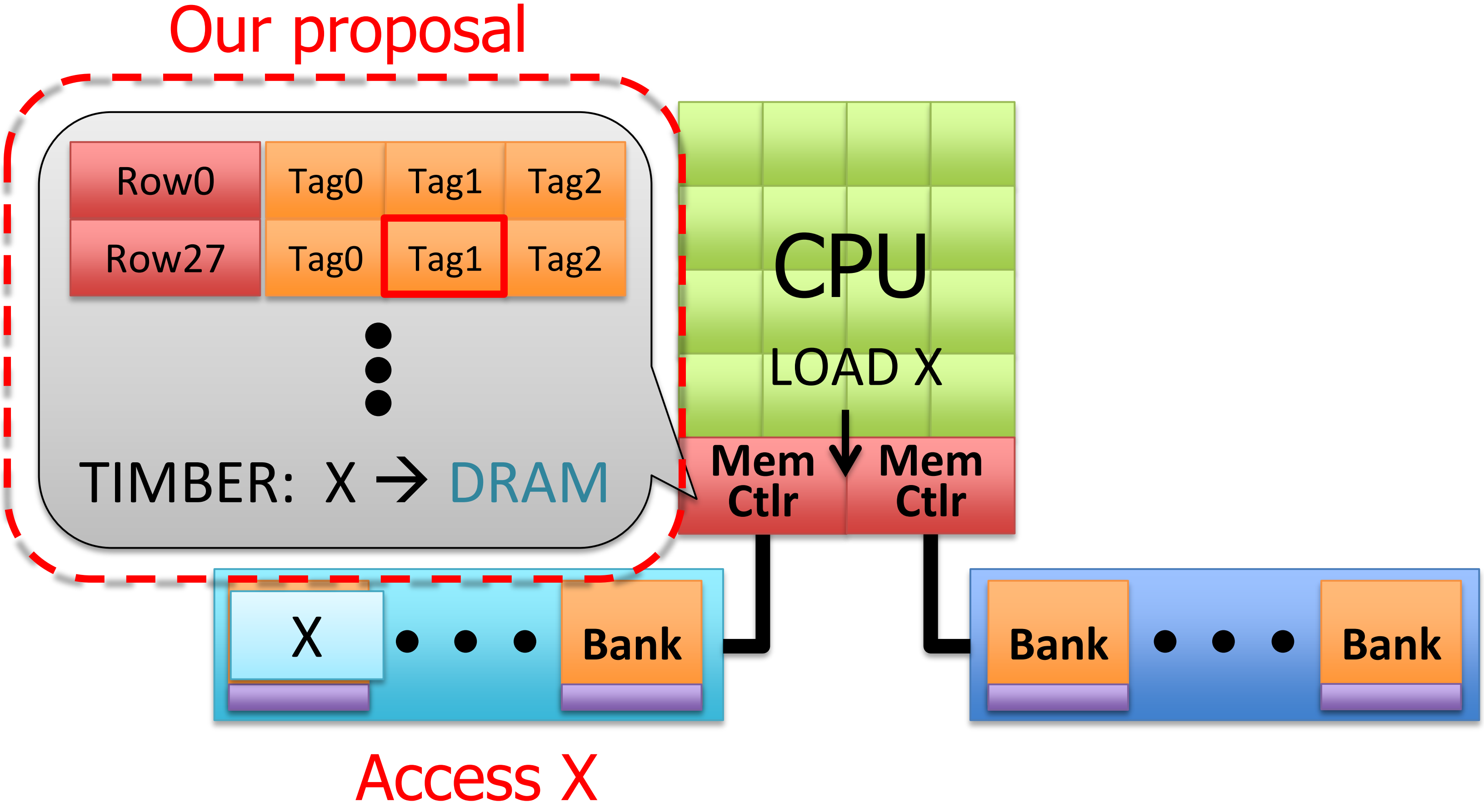
- A Tag-In-Memory BuffER (TIMBER)
  - Stores recently-used tags in a small amount of SRAM



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

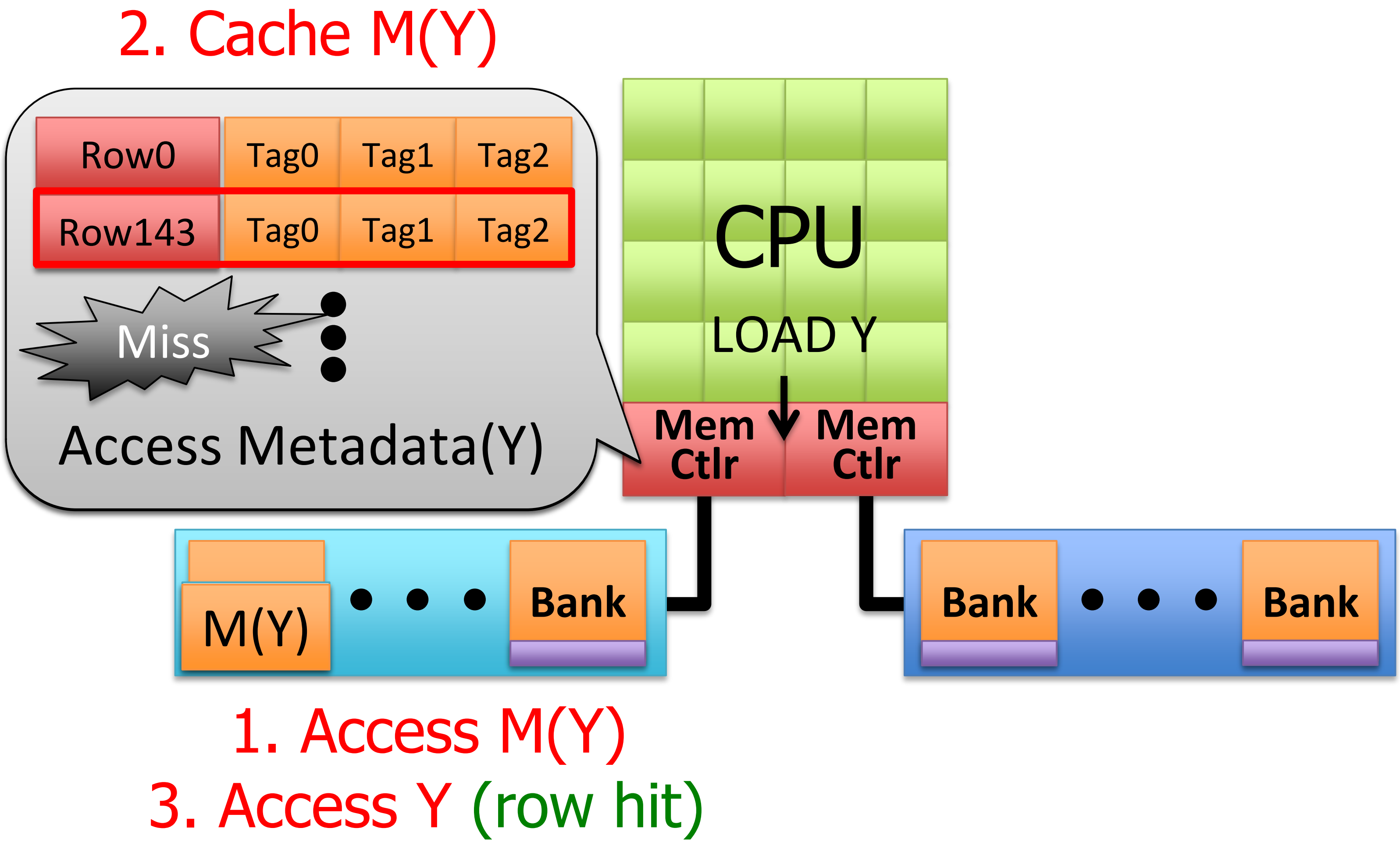
TIMBER Tag Management Example (I)

- Case 1: TIMBER hit



TIMBER Tag Management Example (II)

- Case 2: TIMBER miss



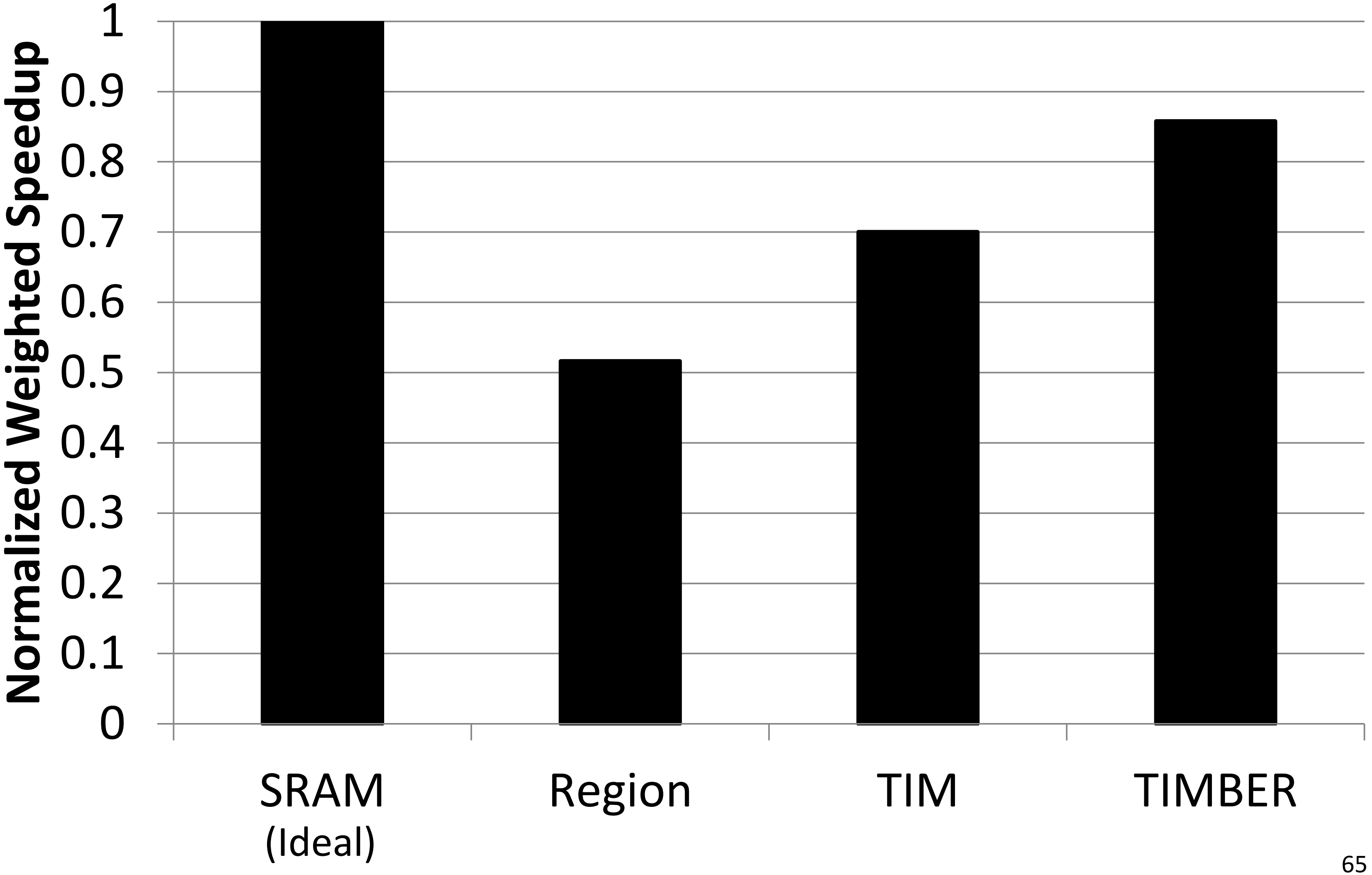
Methodology

- System: 8 out-of-order cores at 4 GHz

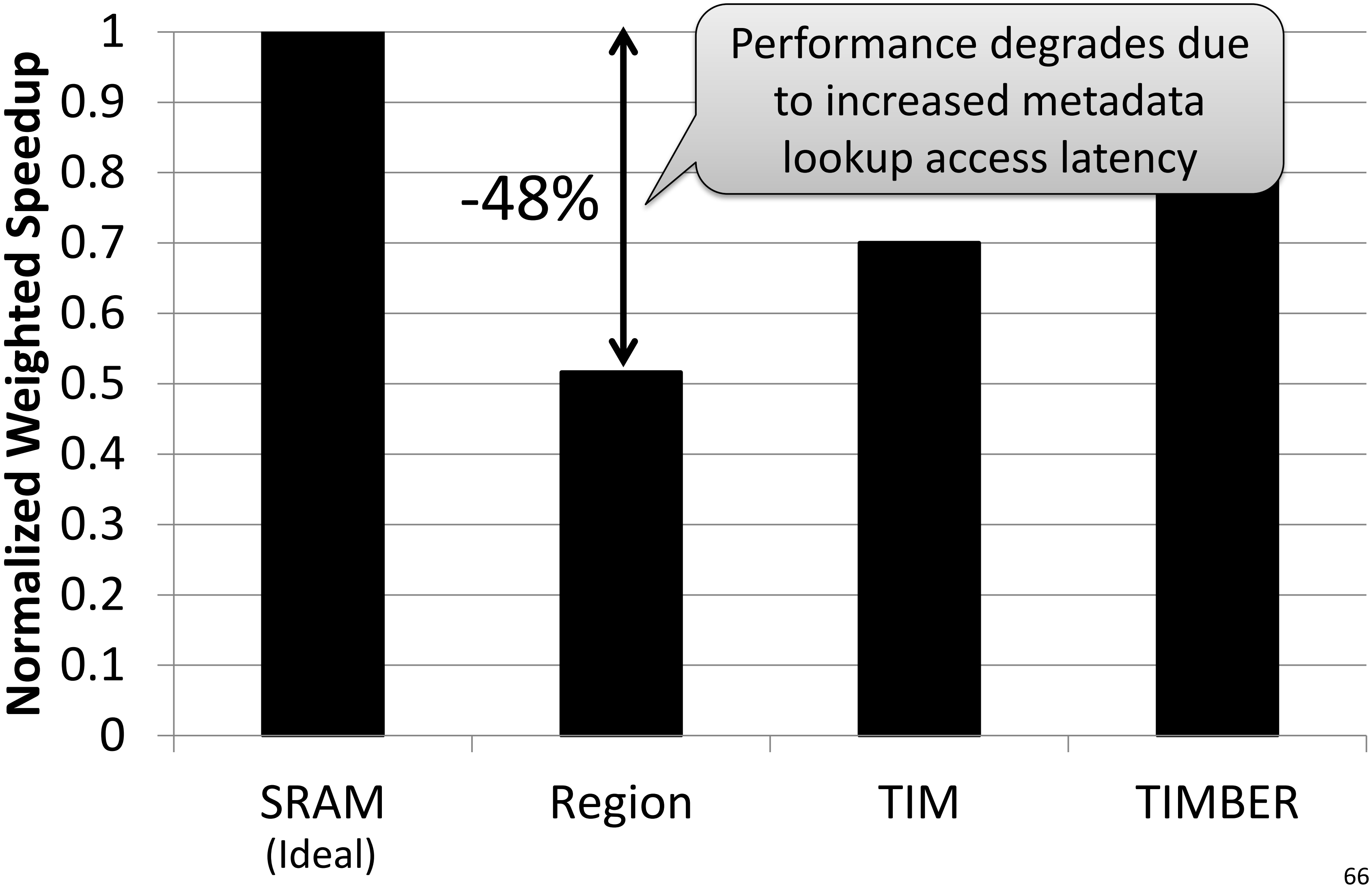
- Memory: 512 MB direct-mapped DRAM, 8 GB PCM
  - 128B caching granularity
  - DRAM row hit (miss): 200 cycles (400 cycles)
  - PCM row hit (clean / dirty miss): 200 cycles (640 / 1840 cycles)

- 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

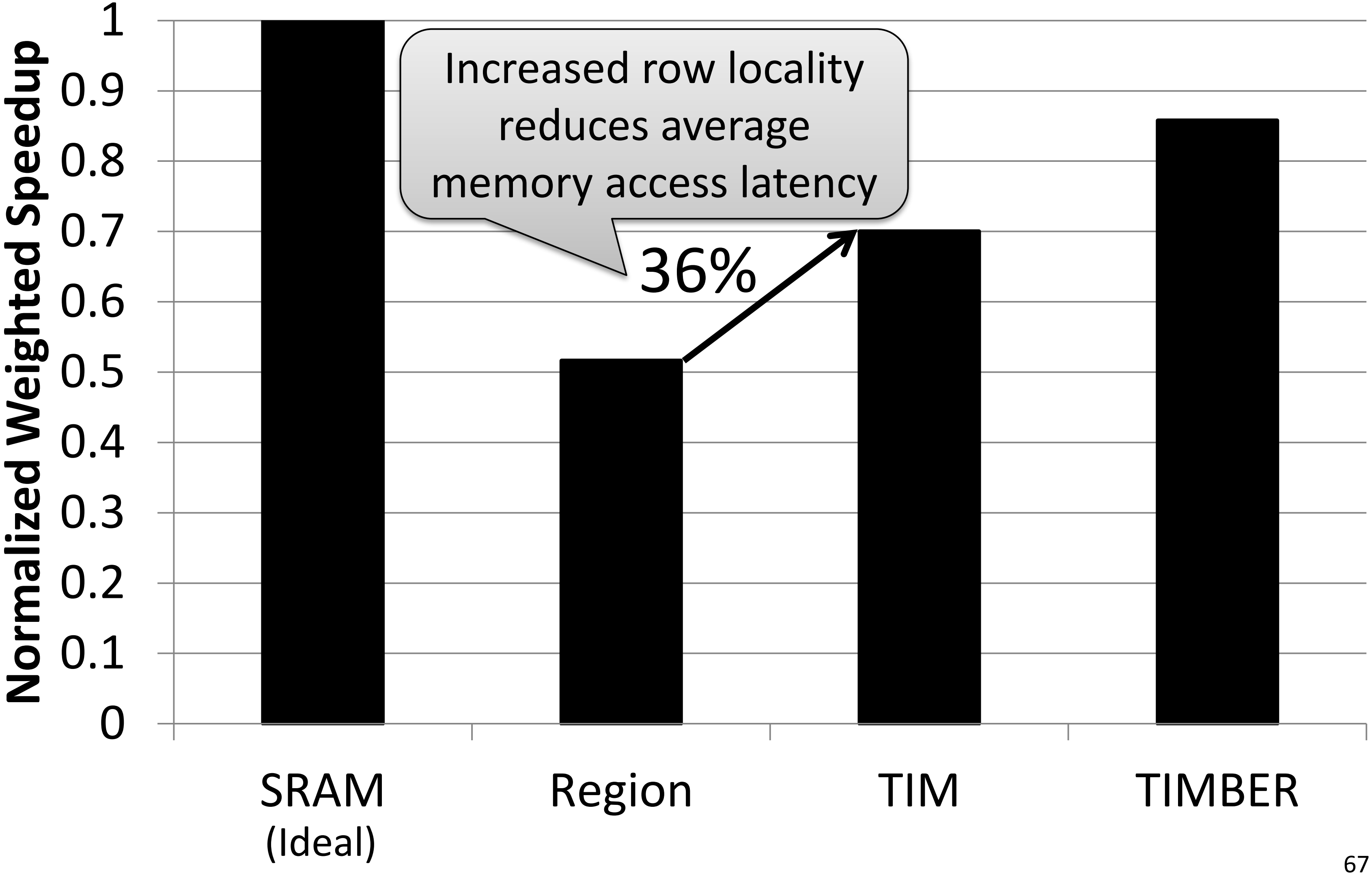


Metadata Storage Performance



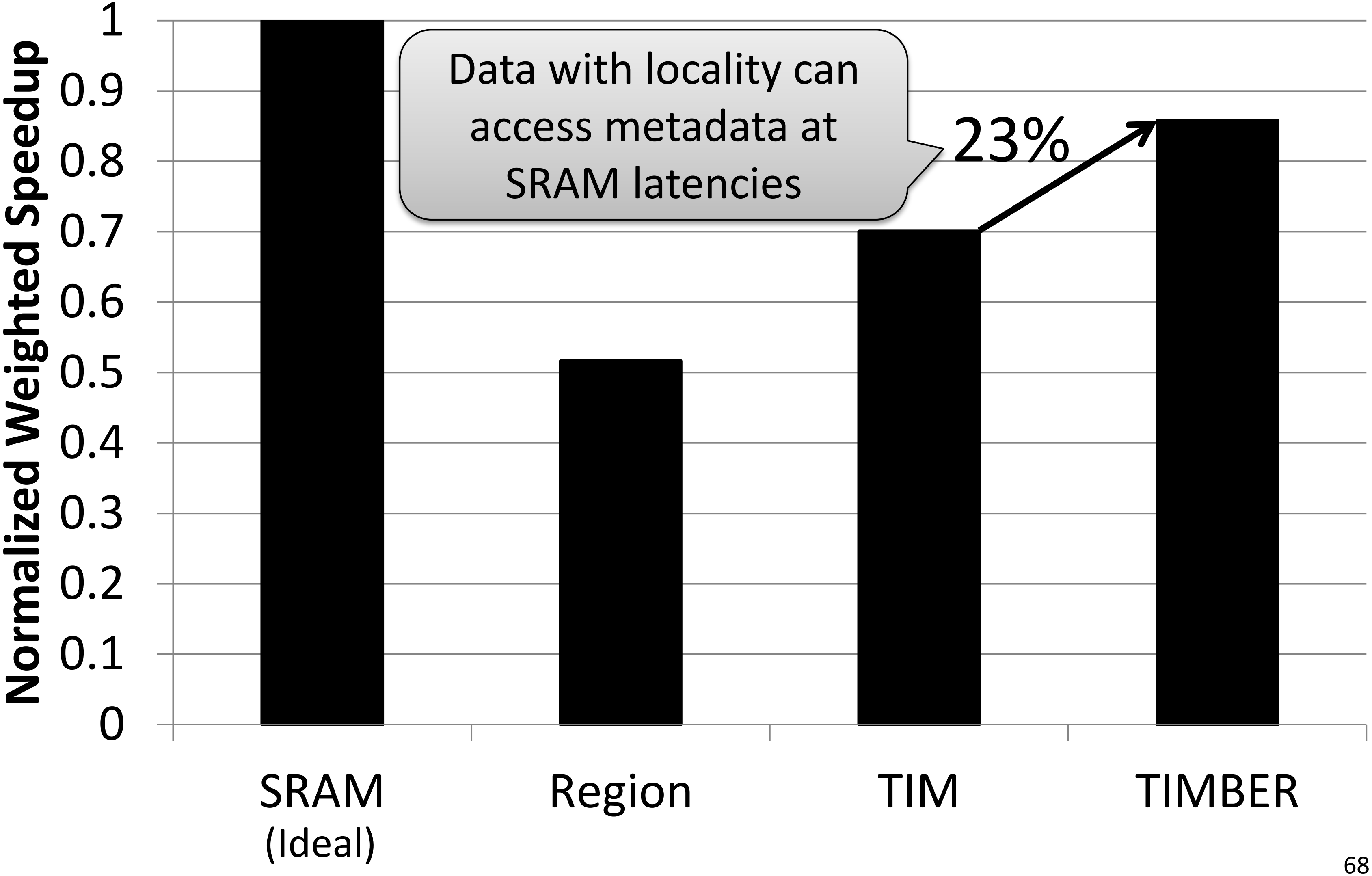
Performance degrades due to increased metadata lookup access latency

Metadata Storage Performance



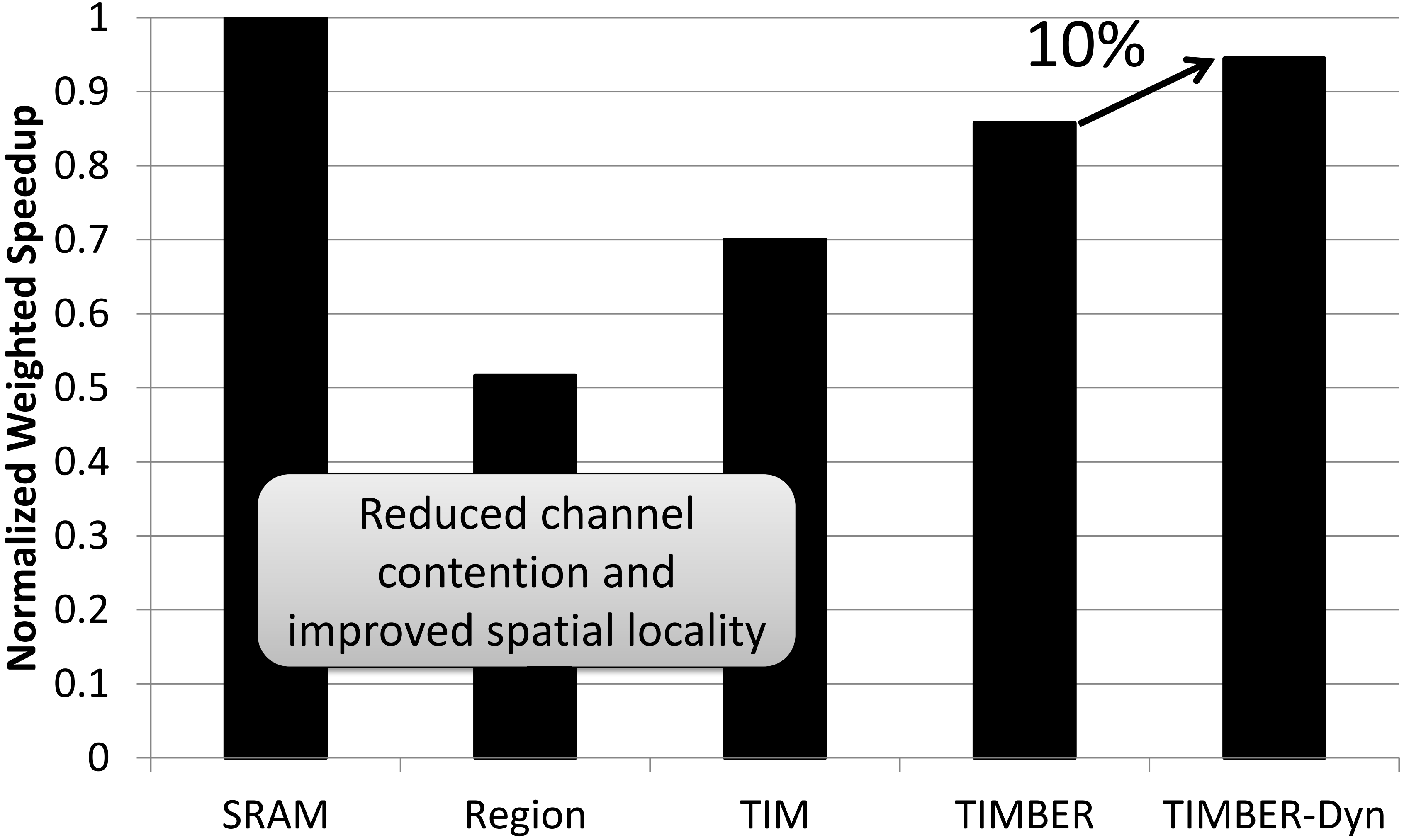
Increased row locality reduces average memory access latency

Metadata Storage Performance



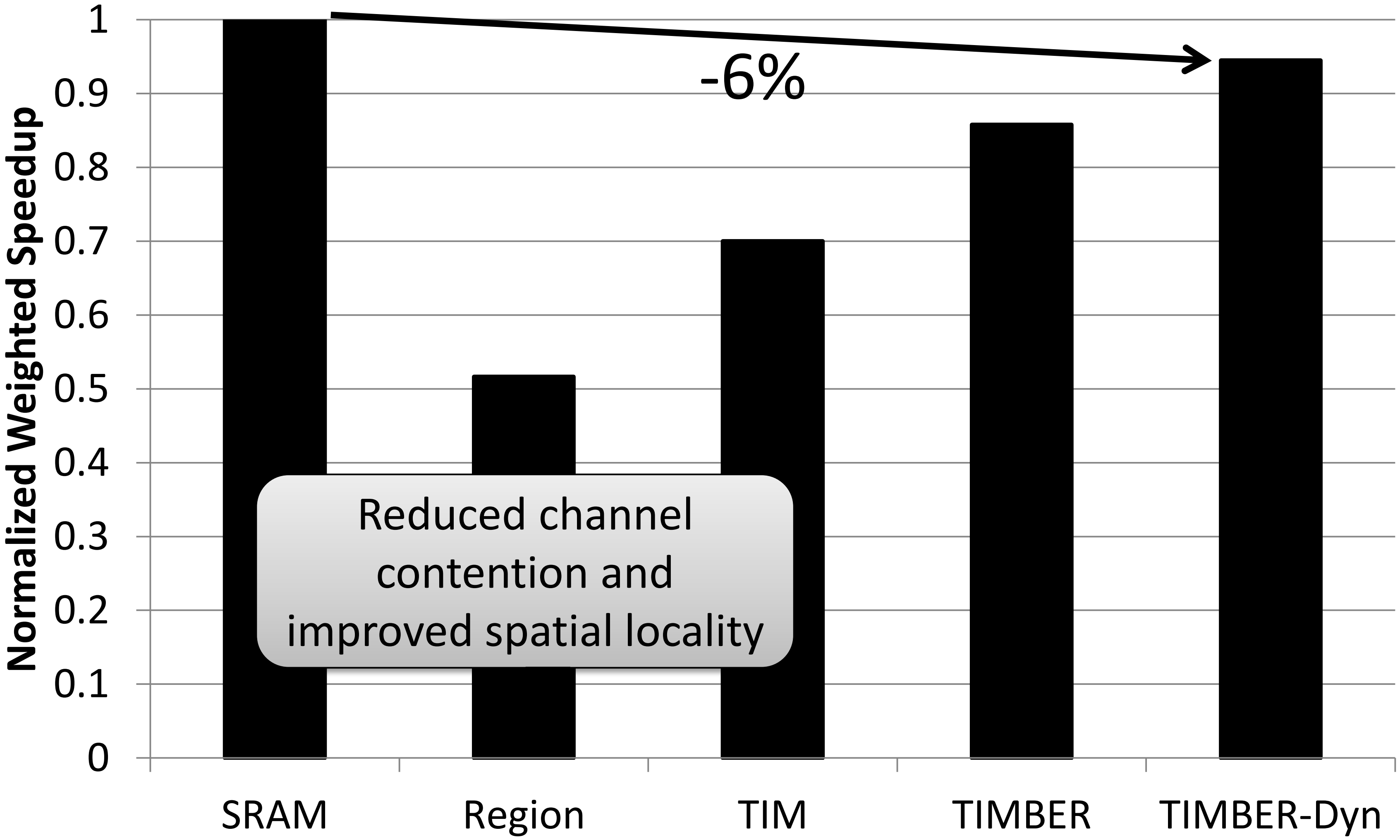
Data with locality can access metadata at SRAM latencies

Dynamic Granularity Performance



Reduced channel contention and improved spatial locality

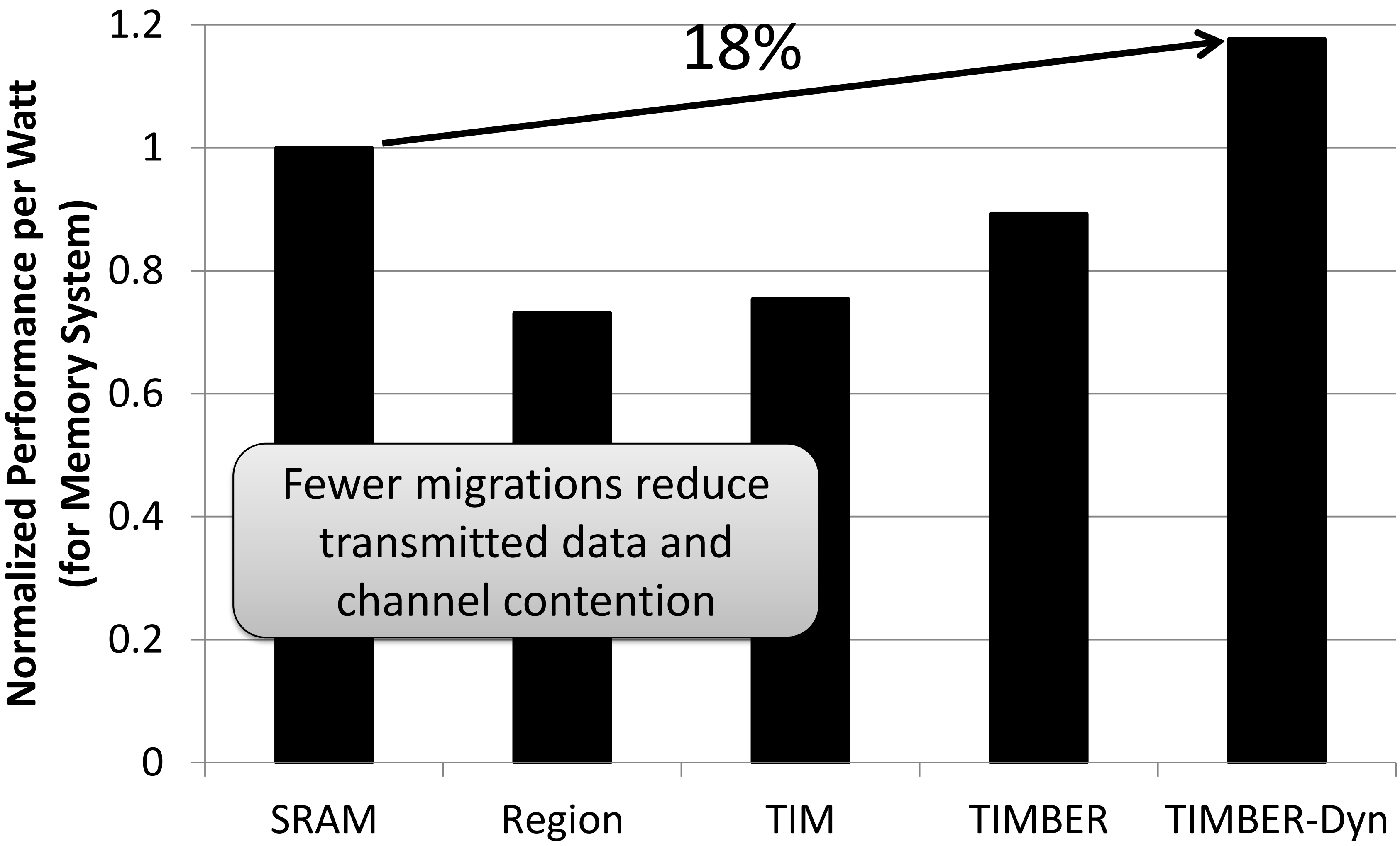
TIMBER Performance



Reduced channel contention and improved spatial locality

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

TIMBER Energy Efficiency



Fewer migrations reduce transmitted data and channel contention

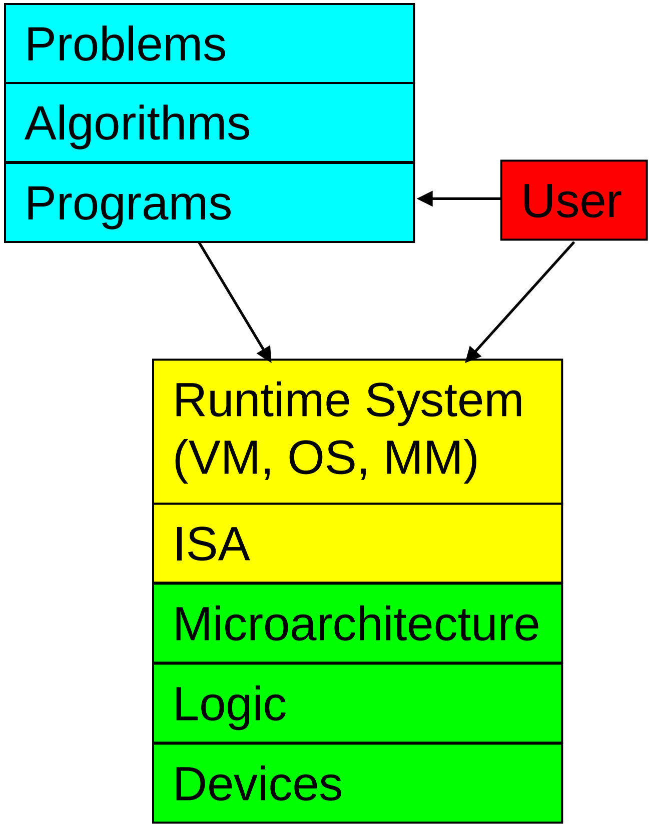
Meza, Chang, Yoon, Mutlu, Ranganathan, "Enabling Efficient and Scalable Hybrid Memories," IEEE Comp. Arch. Letters, 2012.

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 tecnologies
  - 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

Reminder: Project Proposals

- Due: Tuesday, September 25, 11:59pm.

- Extended office hours: Saturday, September 22, 11am-1pm.