A Row Buffer Locality-Aware Caching Policy for Hybrid Memories

HanBin Yoon
Justin Meza
Rachata Ausavarungnirun
Rachael Harding
Onur Mutlu

Carnegie Mellon University

Overview

- Emerging memories such as PCM offer higher density than DRAM, but have drawbacks
- Hybrid memories aim to achieve best of both
- We identify row buffer locality (RBL) as a key criterion for data placement
  - We develop a policy that caches to DRAM rows with low RBL and high reuse
- 50% perf. improvement over all-PCM memory
- Within 23% perf. of all-DRAM memory

Demand for Memory Capacity

- Increasing cores and thread contexts
  - – Intel Sandy Bridge: 8 cores (16 threads)
  - – AMD Abu Dhabi: 16 cores
  - – IBM POWER7: 8 cores (32 threads)
  - – Sun T4: 8 cores (64 threads)
- Modern data-intensive applications operate on huge datasets

SAP® Google facebook

Emerging High Density Memory

- DRAM density scaling becoming costly
- Phase change memory (PCM)
    - + Projected 3–12× denser than DRAM[1]
- However, cannot simply replace DRAM
    - – Higher access latency (4–12× DRAM[2])
    - – Higher access energy (2–40× DRAM[2])
    - – Limited write endurance (~10^8 writes[2])

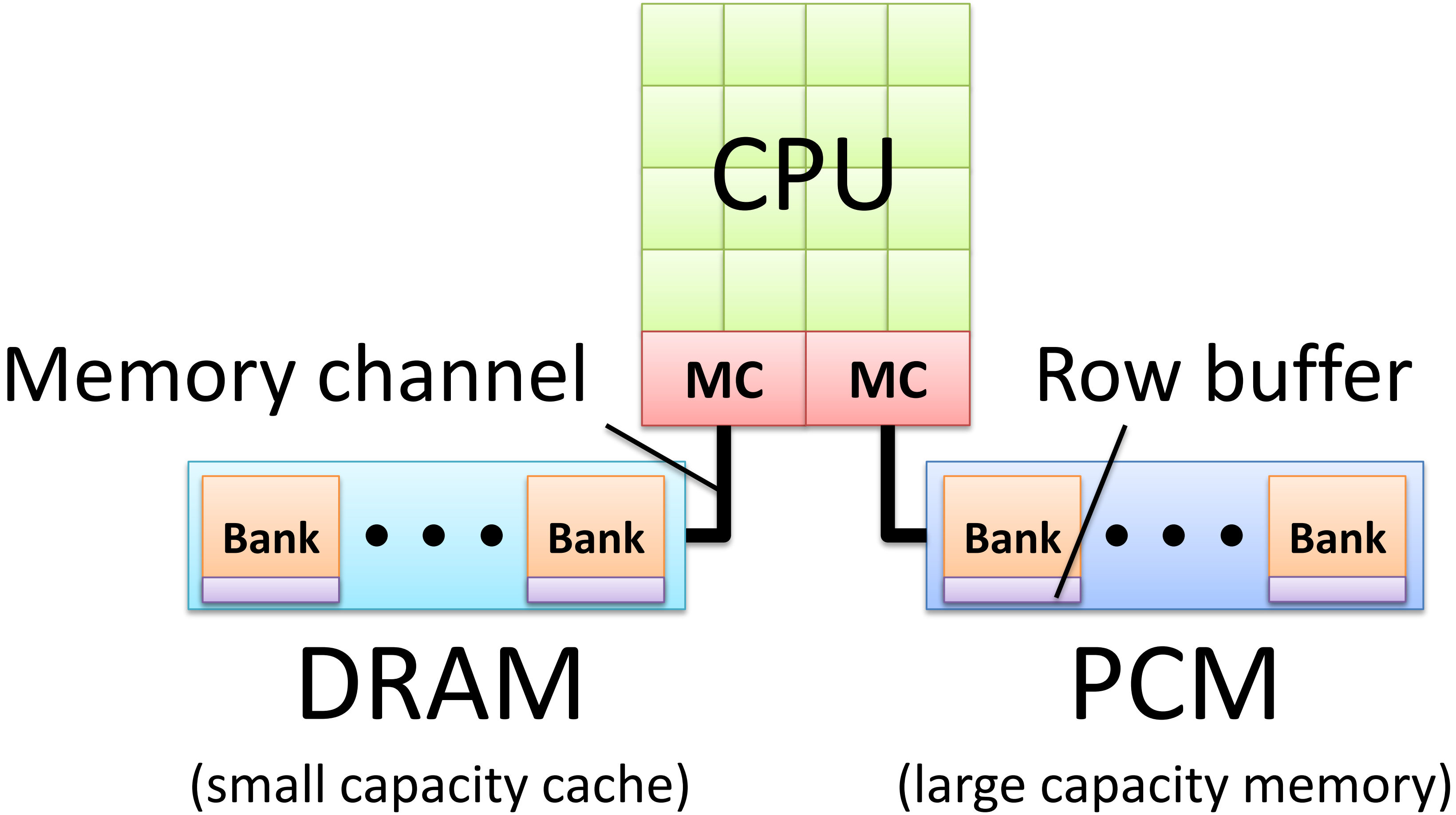
→Use DRAM as a cache to PCM memory[3]

[[1]Mohan HPTS’09; [2]Lee+ ISCA’09; [3]Qureshi+ ISCA’09]

Hybrid Memory

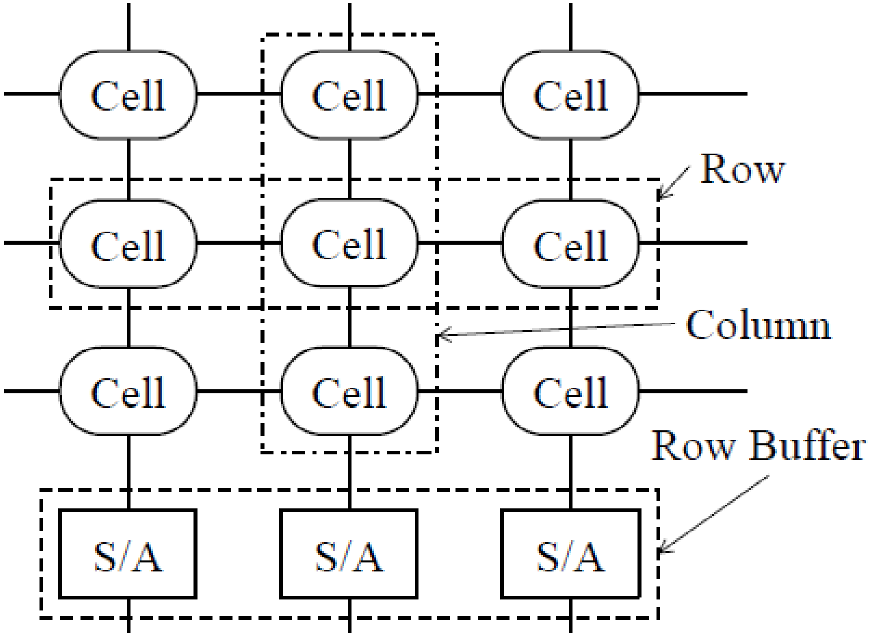
- Benefits from both DRAM and PCM
  - DRAM: Low latency, high endurance
  - PCM: High capacity
- Key question: Where to place data between these heterogeneous devices?
- To help answer this question, let's take a closer look at these technologies

Hybrid Memory: A Closer Look



Row Buffers and Latency

- Memory cells organized in columns and rows



- Row buffers store last accessed row
  - Hit: Access data from row buffer → fast
  - Miss: Access data from cell array → slow

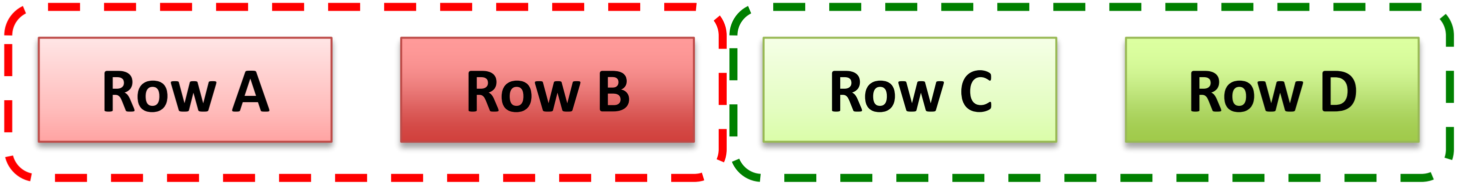
Key Observation

- Row buffers exist in both DRAM and PCM
  - Row buffer hit latency **similar** in DRAM & PCM[2]
  - Row buffer miss latency **small** in DRAM
  - Row buffer miss latency **large** in PCM
- Place data in DRAM which
  - Frequently miss in row buffer (low row buffer locality)→ miss penalty is smaller in DRAM
  - Are reused many times → data worth the caching effort (contention in mem. channel and DRAM)

[[2]Lee+ ISCA'09]

Data Placement Implications

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



Row A

Row B

Row C

Row D

Low RBL

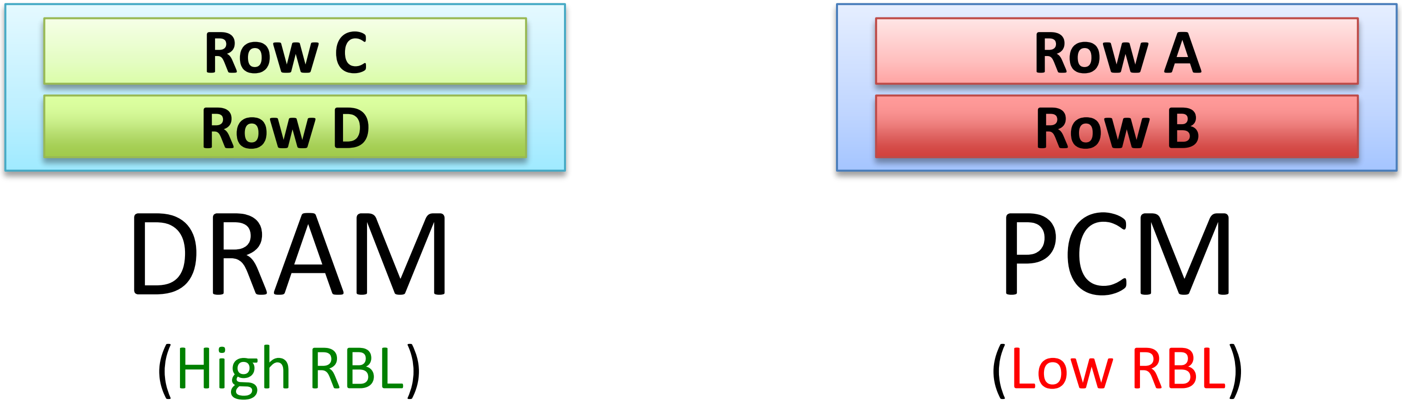
(Frequently miss in row buffer)

High RBL

(Frequently hit in row buffer)

RBL-Unaware Policy

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



Row C

Row D

DRAM

(High RBL)

Row A

Row B

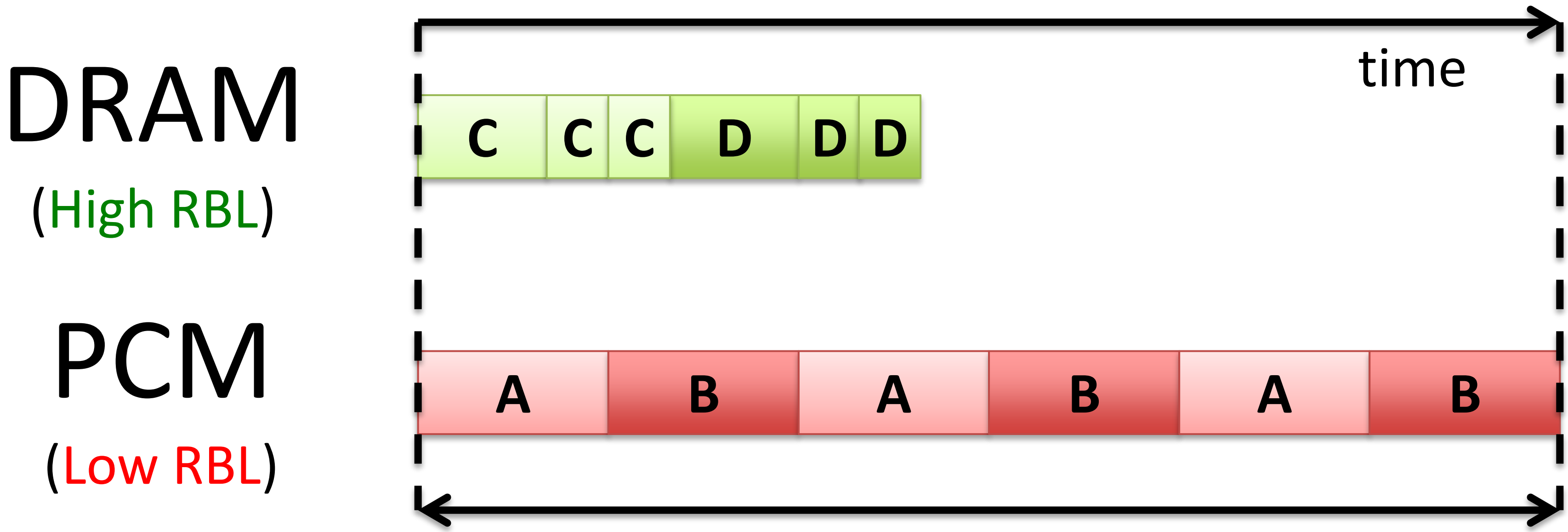
PCM

(Low RBL)

RBL-Unaware Policy

Accesses pattern to main memory:

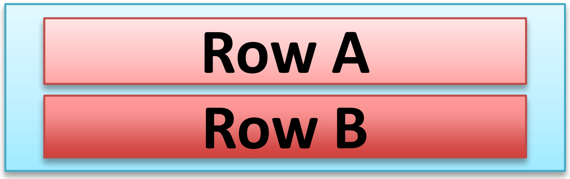
A (oldest), B, C, C, C, A, B, D, D, D, A, B (youngest)



Stall time: 6 PCM device accesses

RBL-Aware Policy

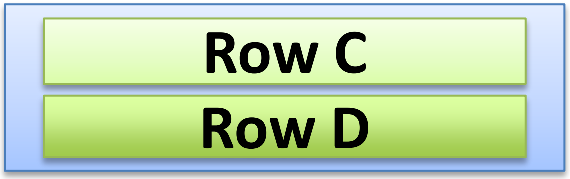
A **row buffer locality-aware** policy would place these rows in the following manner



DRAM

(Low RBL)

→ Access data at lower row buffer miss latency of DRAM



PCM

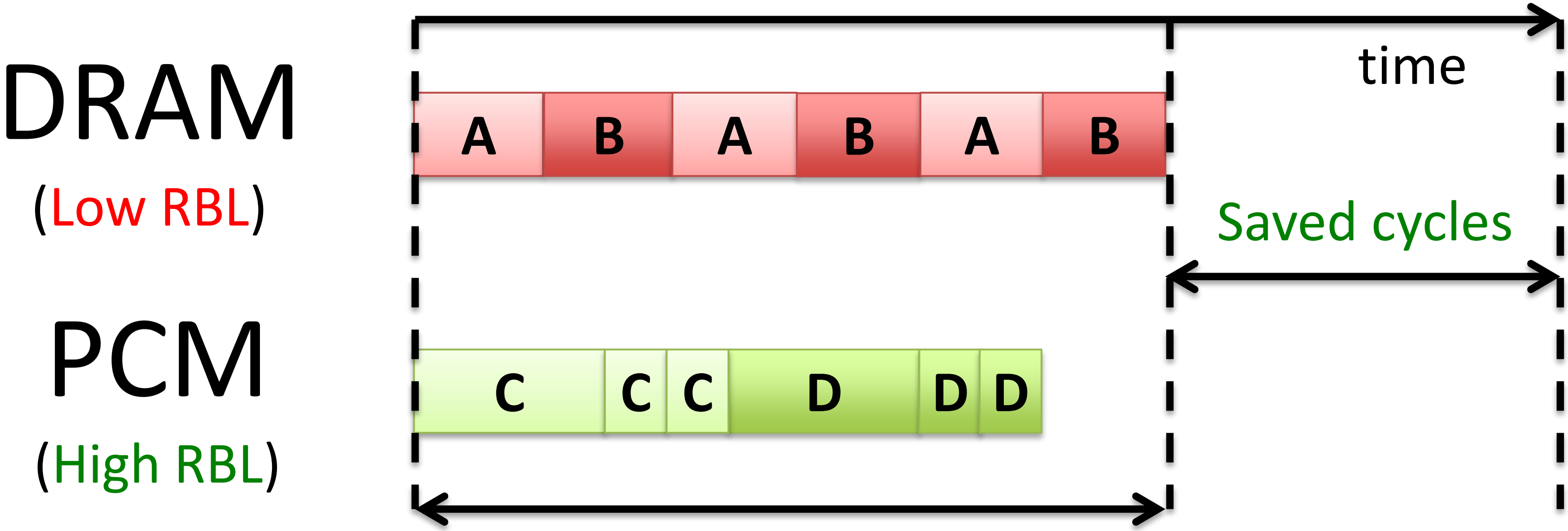
(High RBL)

→ Access data at low row buffer hit latency of PCM

RBL-Aware Policy

Accesses pattern to main memory:

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



Stall time: 6 **DRAM** device accesses

Our Mechanism: DynRBLA

1. For a subset of recently used rows in PCM:
   - Count row buffer **misses** as indicator of row buffer locality (RBL)
2. Cache to DRAM rows with **misses** $\geq$ 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

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**: DDR3 1066 MT/s, 1GB DRAM, 16GB PCM, 1KB migration granularity
- Multi-programmed server & cloud workloads
  - Server (18): TPC-C, TPC-H
  - Cloud (18): Webserver, Video, TPC-C/H

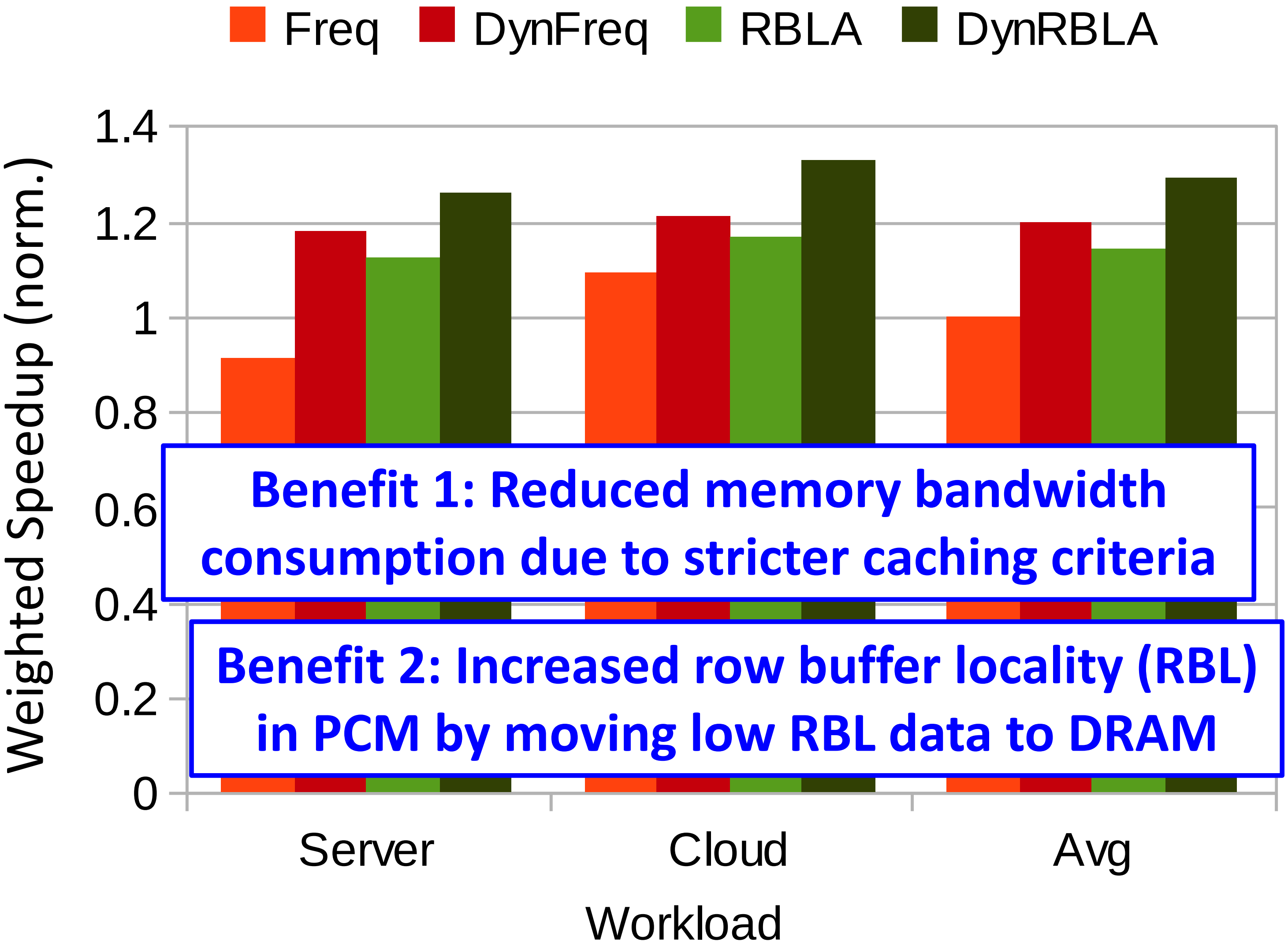
Comparison Points and Metrics

- **DynRBLA**: Adaptive RBL-aware caching
- **RBLA**: Row buffer locality-aware caching
- **Freq**[4]: Frequency-based caching
- **DynFreq**: Adaptive Freq.-based caching

- **Weighted speedup (performance)** = sum of speedups versus when run alone
- **Max slowdown (fairness)** = highest slowdown experienced by any thread

[4Jiang+ HPCA'10]

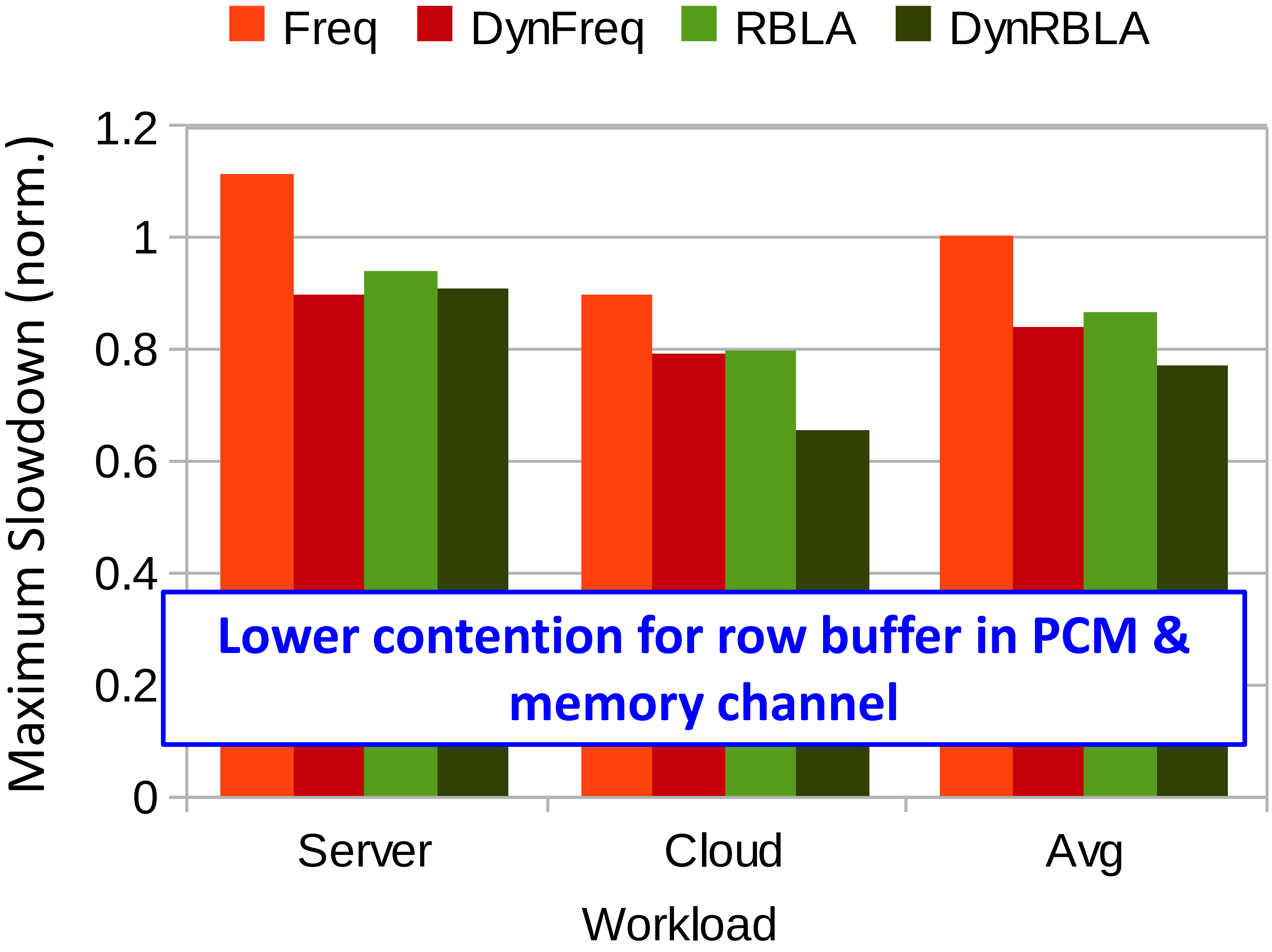
Performance



Benefit 1: Reduced memory bandwidth consumption due to stricter caching criteria

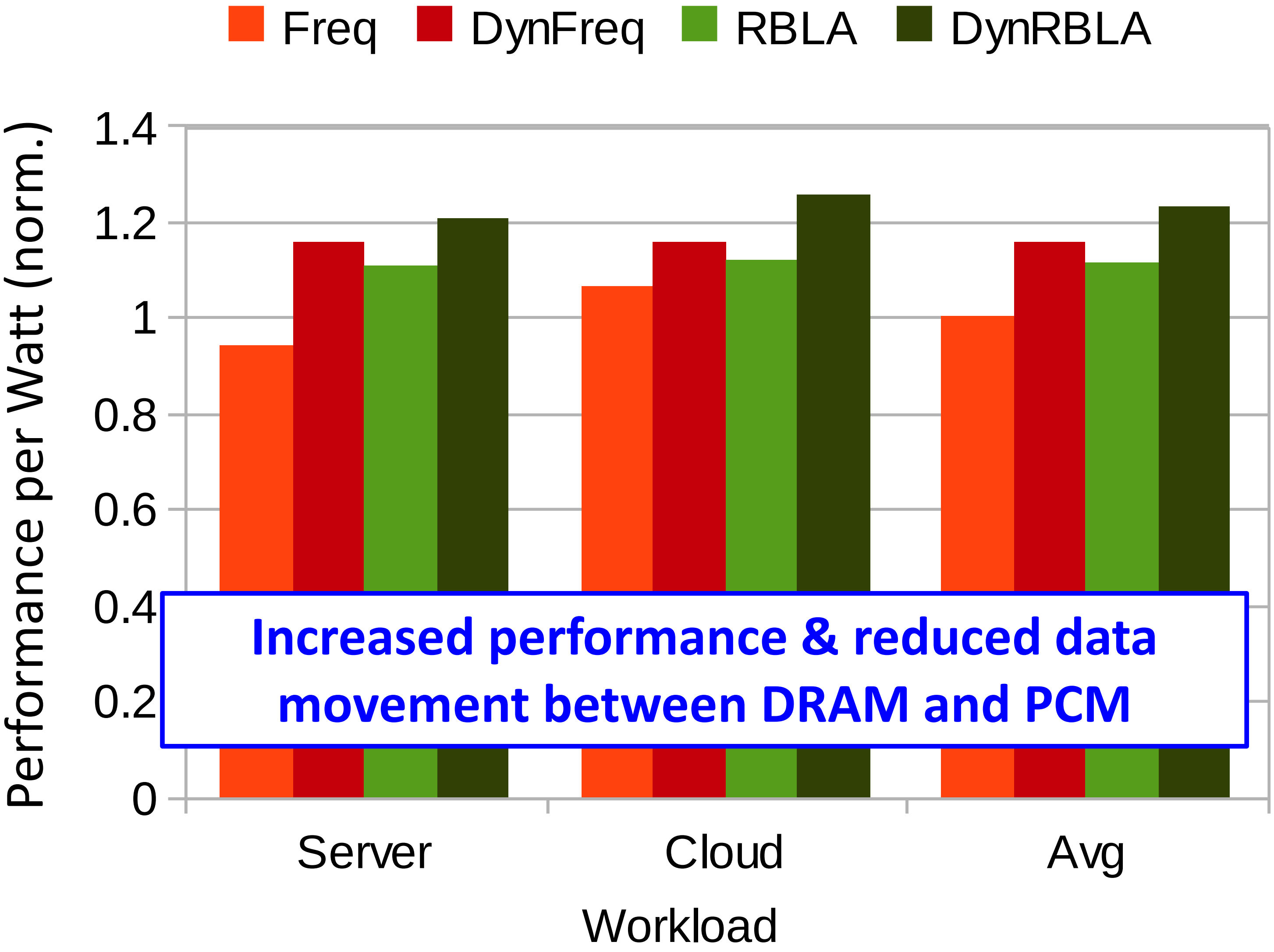
Benefit 2: Increased row buffer locality (RBL) in PCM by moving low RBL data to DRAM

Fairness



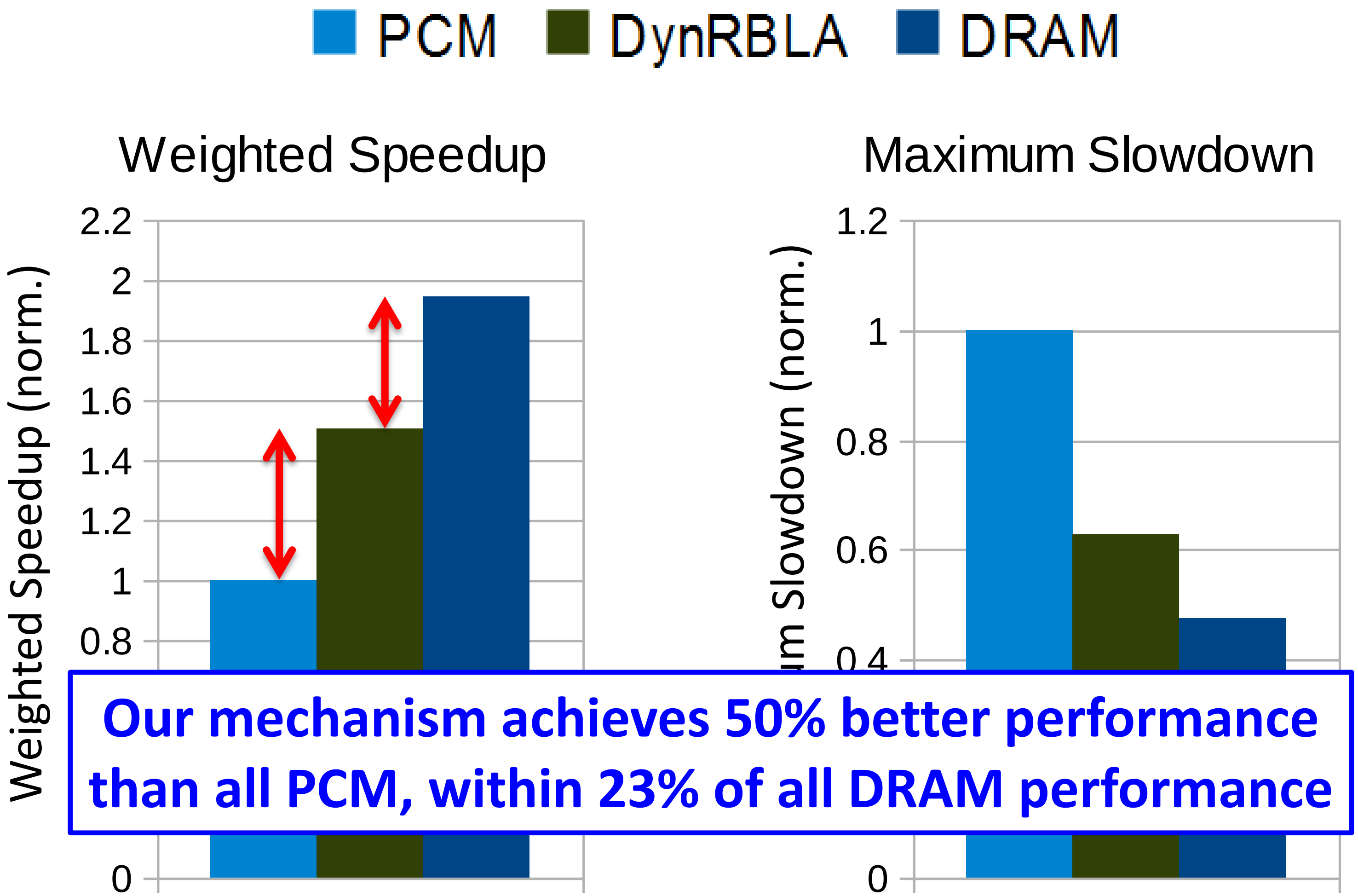
Lower contention for row buffer in PCM & memory channel

Energy Efficiency



Increased performance & reduced data movement between DRAM and PCM

Compared to All-PCM/DRAM



Our mechanism achieves 50% better performance than all PCM, within 23% of all DRAM performance

Conclusion

- Demand for huge main memory capacity
  - PCM offers greater density than DRAM
  - Hybrid memories achieve the best of both
- We identify row buffer locality (RBL) as a key metric for caching decisions
- We develop a policy that caches to DRAM rows with low RBL and high reuse
- Enables high-performance energy-efficient hybrid main memories

Thank you! Questions?