Kingsguard: Write-Rationing Garbage Collection for Hybrid Memories

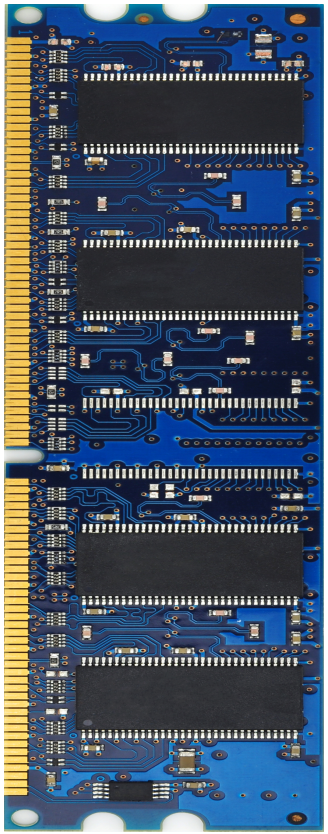
Shoaib Akram (Ghent), Jennifer B. Sartor (Ghent),
Kathryn S. Mckinley (Google), and Lieven Eeckhout (Ghent)
Shoaib.Akram@UGent.be

DRAM is facing challenges

Scalability

Reliability

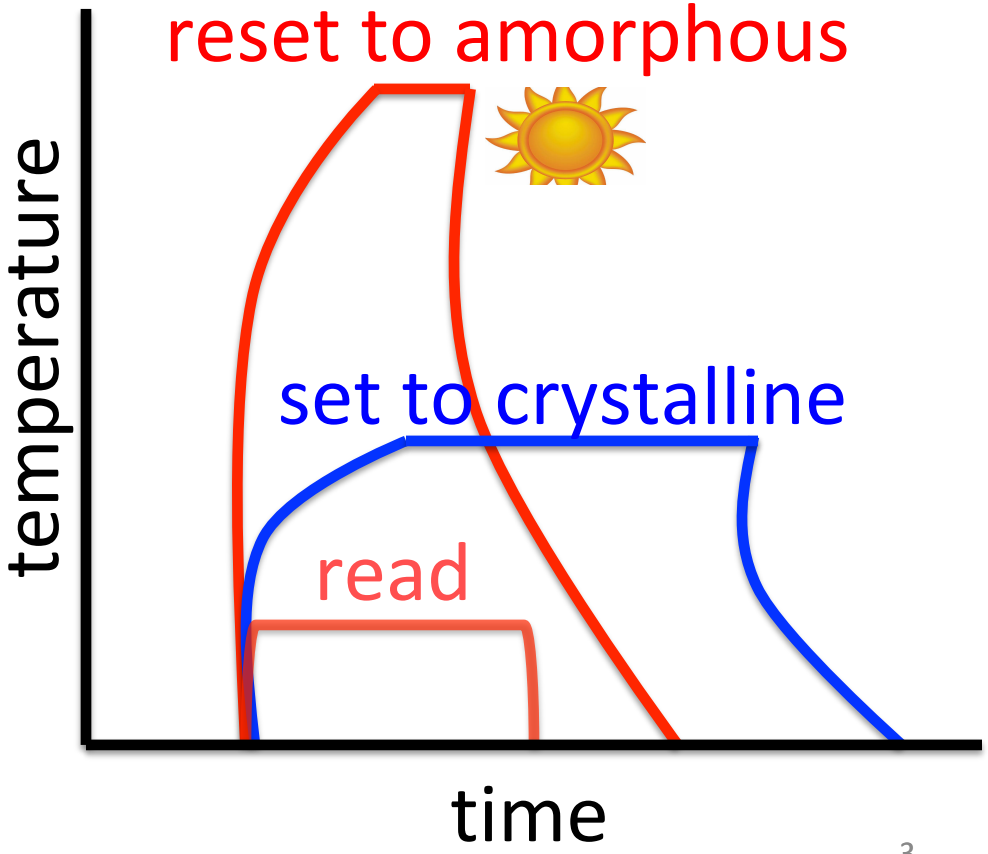
Energy



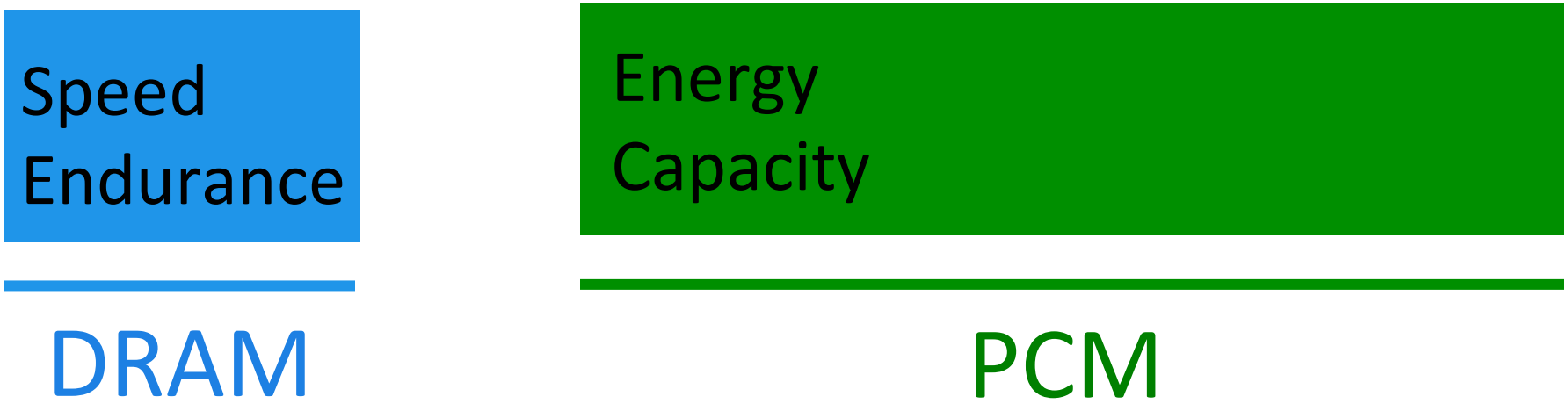
Phase change memory is promising

But …

- GB/$ ☺
- Latency ☹
- Endurance ☹



Hybrid DRAM-PCM memory

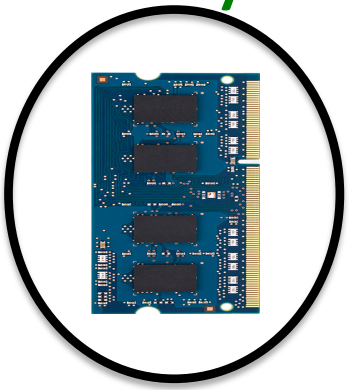


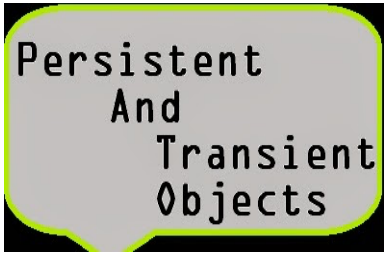
Challenge

Mitigate PCM wear-out and extend its lifetime

How to mitigate PCM wear-out?

Phase change memory as …





Wear Level

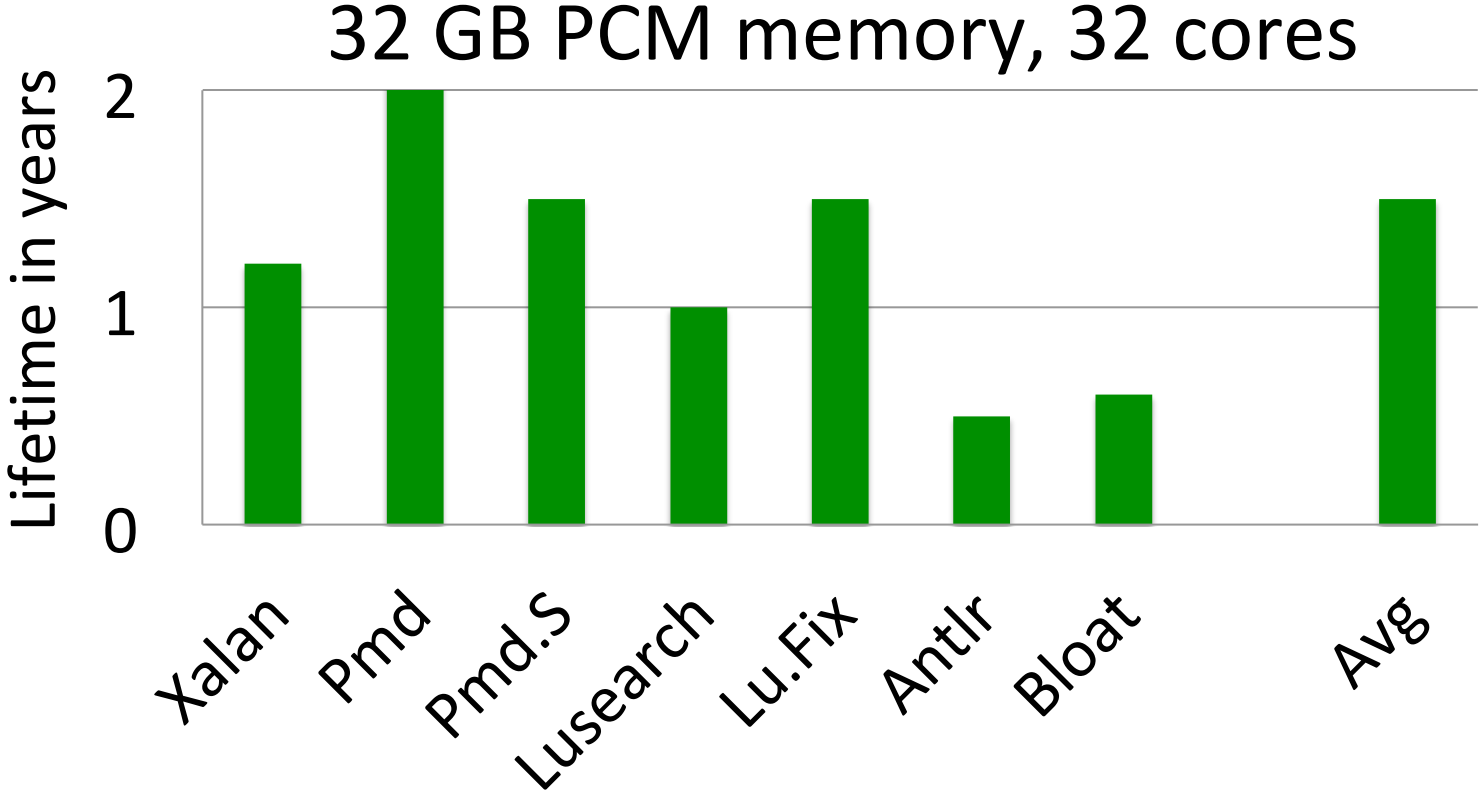
Wear Level

Wear Level

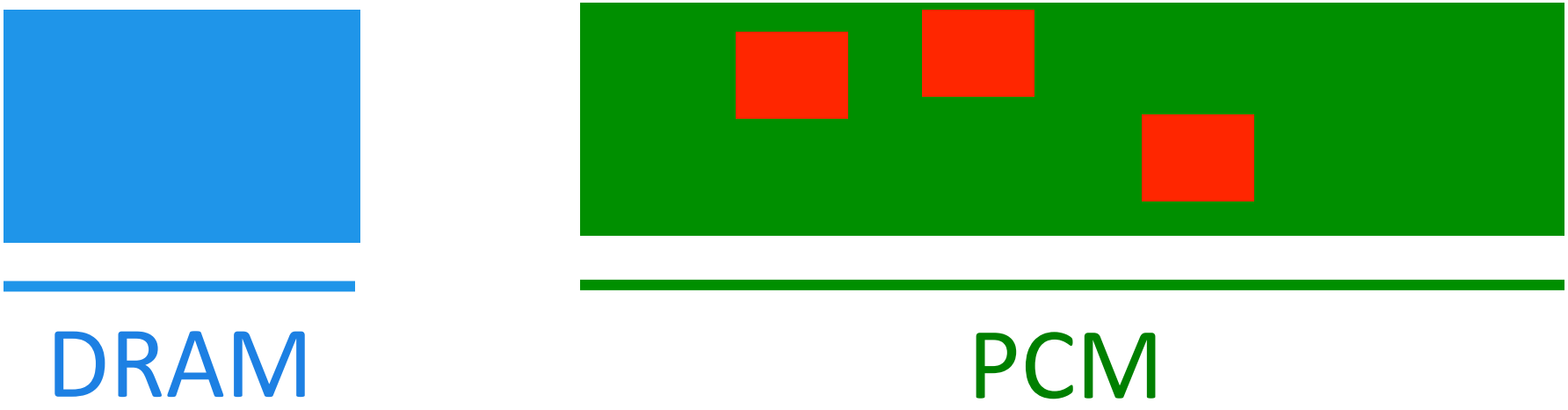
Operating System

Language runtime

PCM only with leveling is not practical



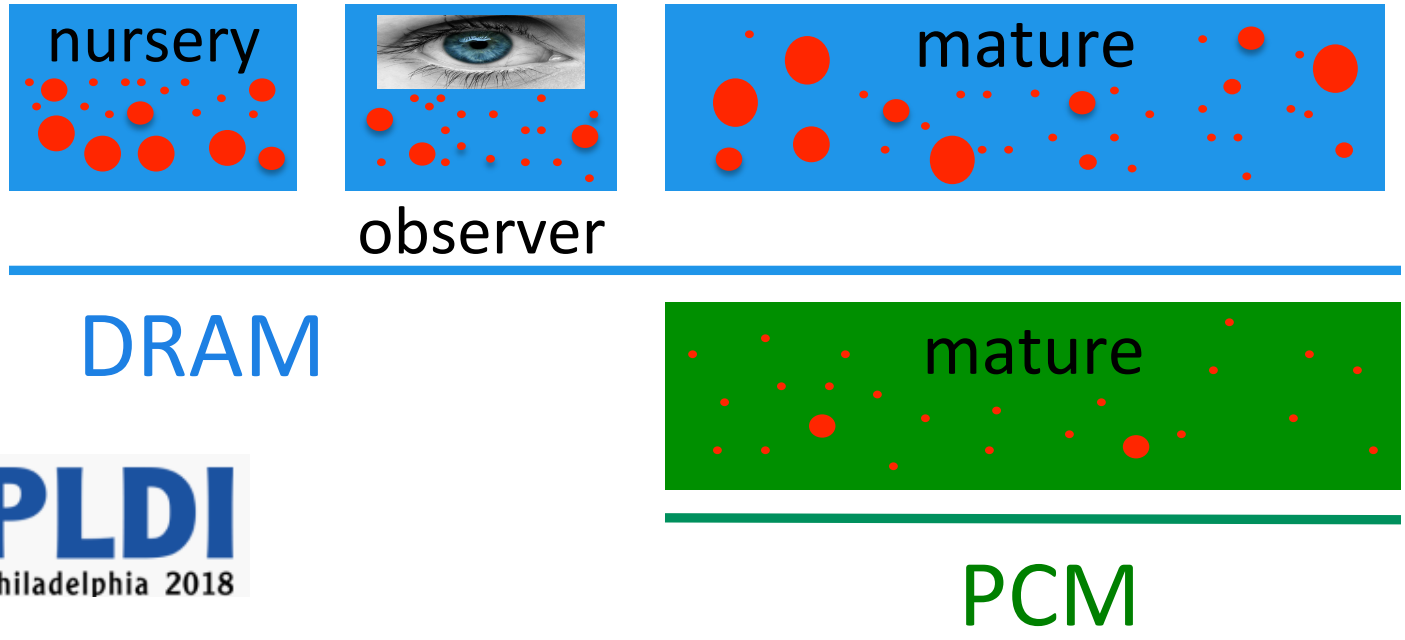
OS to limit PCM writes



Drawbacks

- Coarse grained
- Page migrations can be costly

Managed runtime to limit PCM writes



PLDI Philadelphia 2018

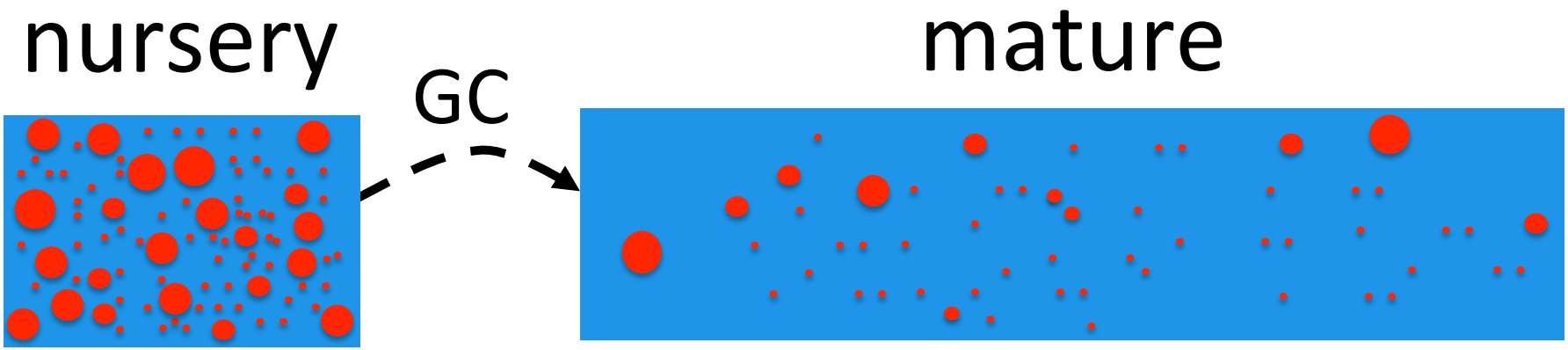
Our work uses garbage collection to keep highly written objects in DRAM

Distribution of writes in GC runtime

70%

of writes

Distribution of writes in GC runtime



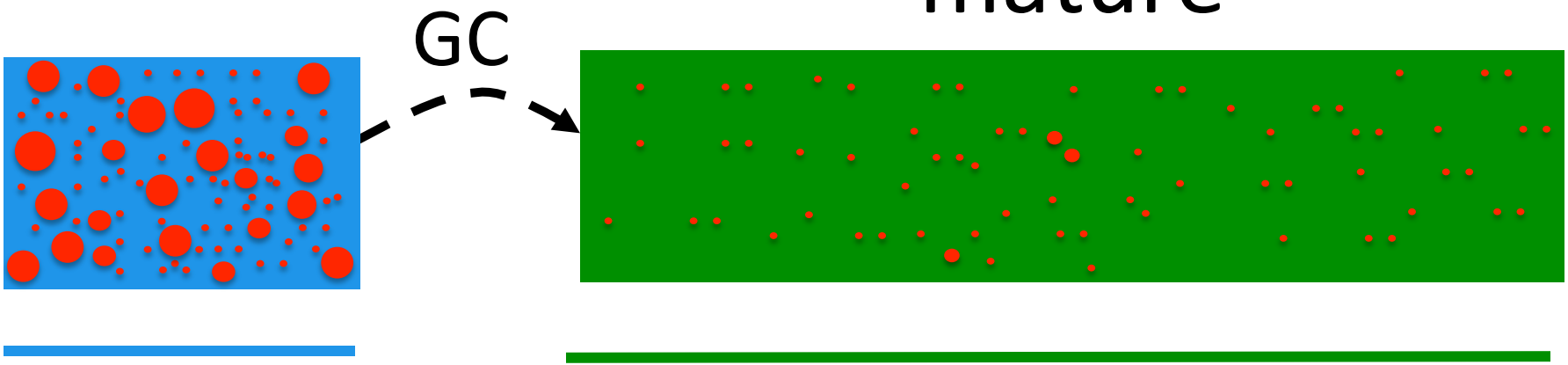
70%

of writes

22%

to 2% of objects

Contribution
Write-Rationing Garbage Collectors

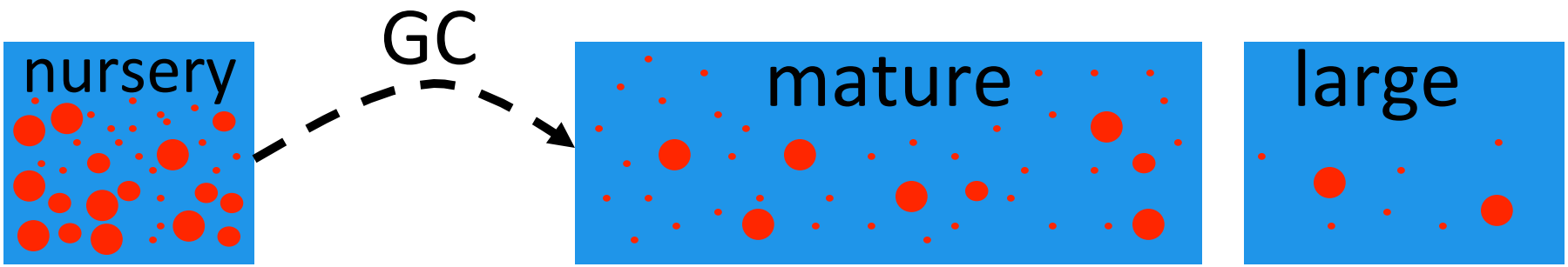


Two write-rationing garbage collectors

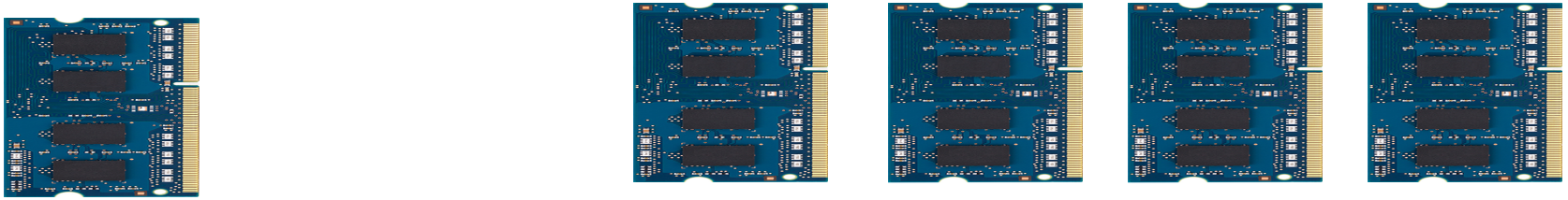
Kingsguard-Nursery

Kingsguard-Writers

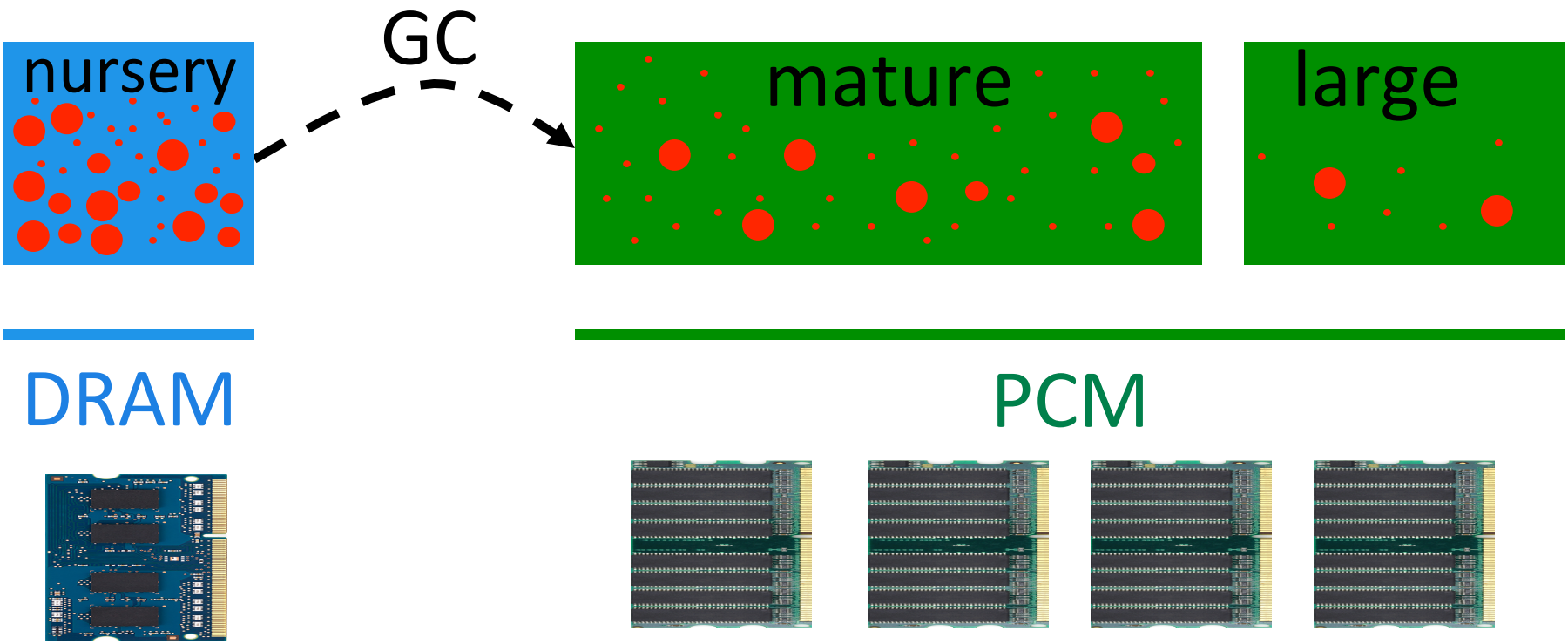
Heap organization in DRAM



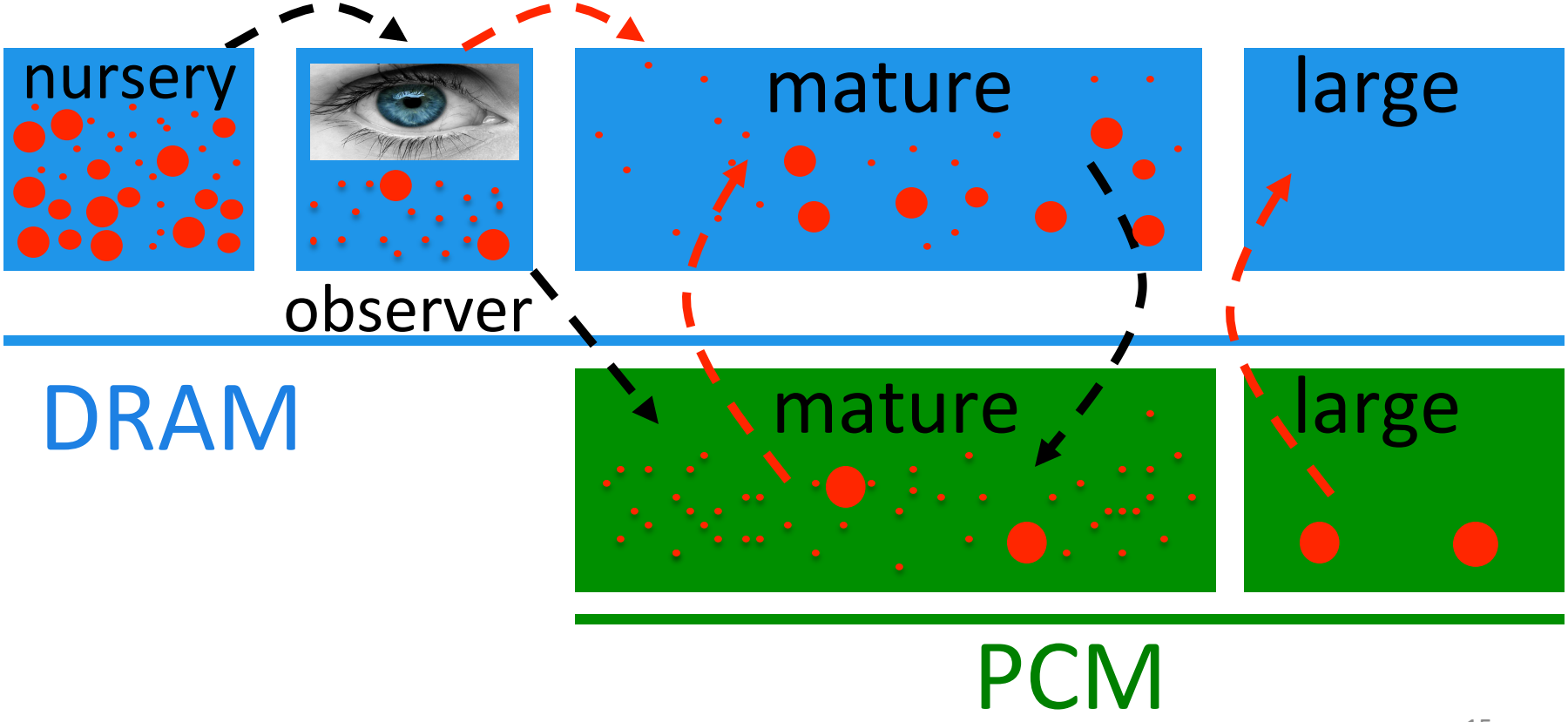
DRAM



KG-N Kingsguard-Nursery



KG-W Kingsguard-Writers



Observing writes

Object format

Write barrier sets a header bit on object writes

Write barrier configurations

- Observe references
- Observe references and primitives

Additional optimizations in KG-W

Large object optimization

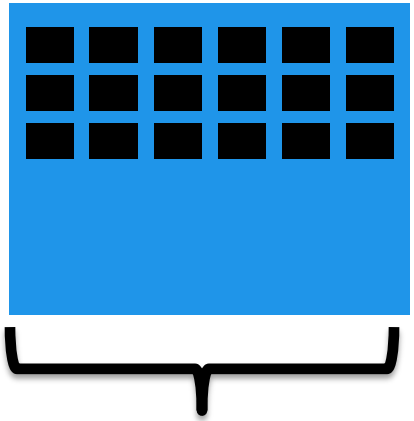
- Allocate selected large objects in DRAM

Metadata optimization

- Allocate PCM metadata in DRAM

Large object optimization

nursery



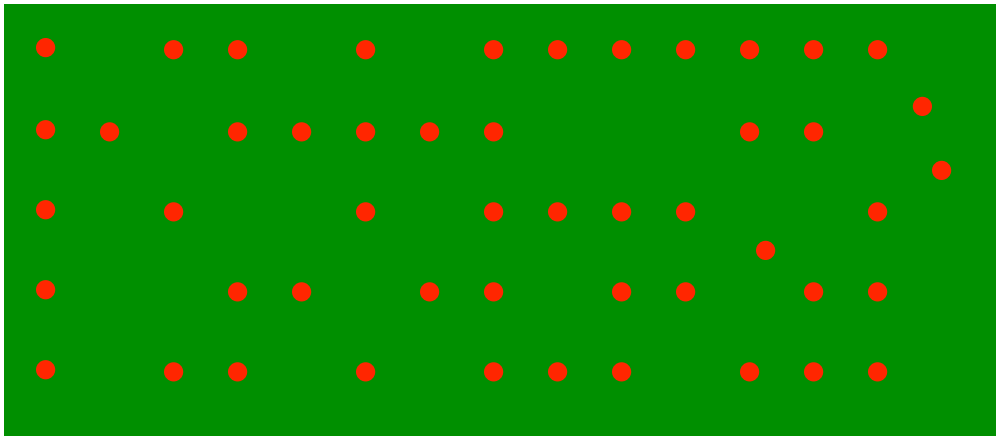
½ of remaining nursery

large

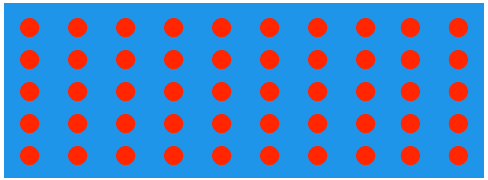
Monitor PCM write rate to tu[redacted]

Metadata optimization

Mature



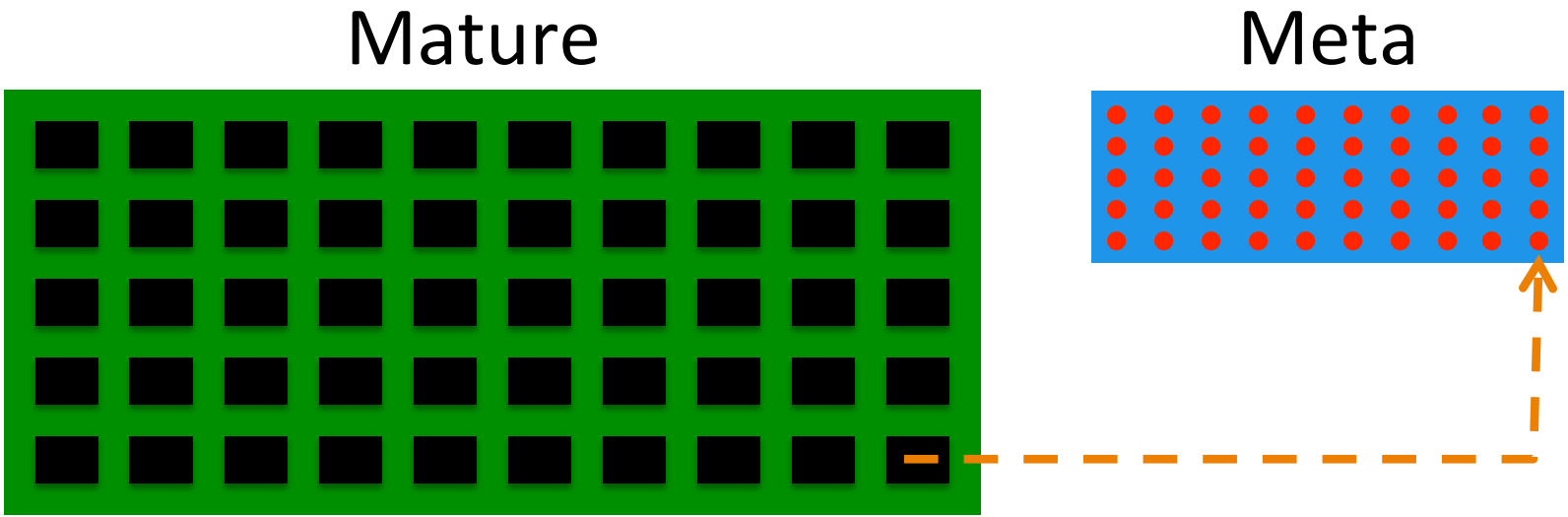
Meta



Full-heap GC: Mark live PCM objects

KG-W: Keep mark bytes of PCM objects in DRAM

Metadata optimization



Full-heap GC: Mark live PCM objects

KG-W: Keep mark bytes of PCM objects in DRAM

address_mark_bit = start_meta + idx_pcm_obj

Evaluation Methodology

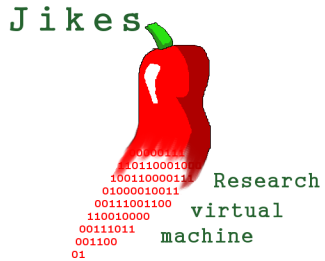
Hardware

(1) Simulator

(2) Real

Software

Jikes research virtual machine



Java applications

Simulation with Sniper

7 DaCapo applications

4 cores, 1 MB per core LLC

Scale simulated rates to a 32 core machine using trends from real hw

Memory systems

Homogeneous

- 32 GB DRAM
- 32 GB PCM

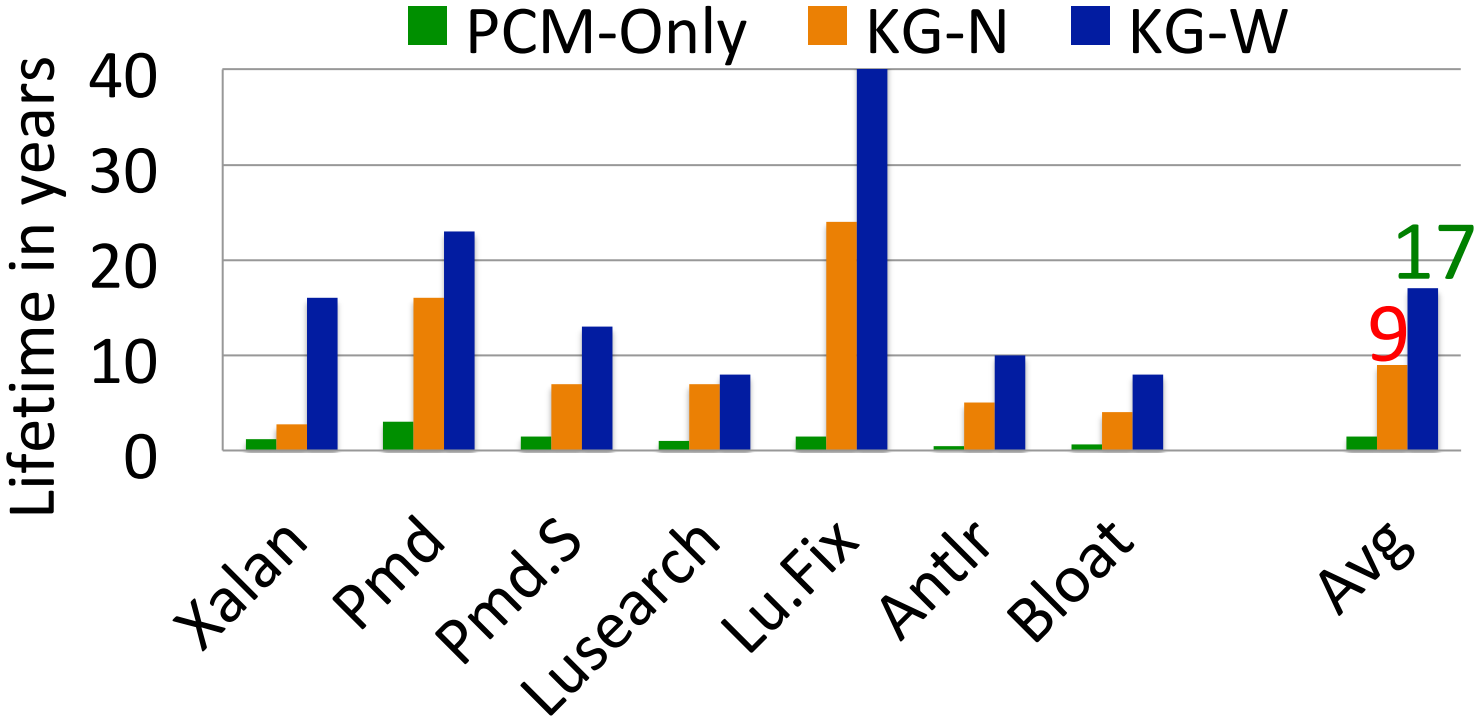
Hybrid

- 1 GB DRAM
- 32 GB PCM

PCM parameters

- 4X read latency
- 4X write energy
- 10 M writes/cell

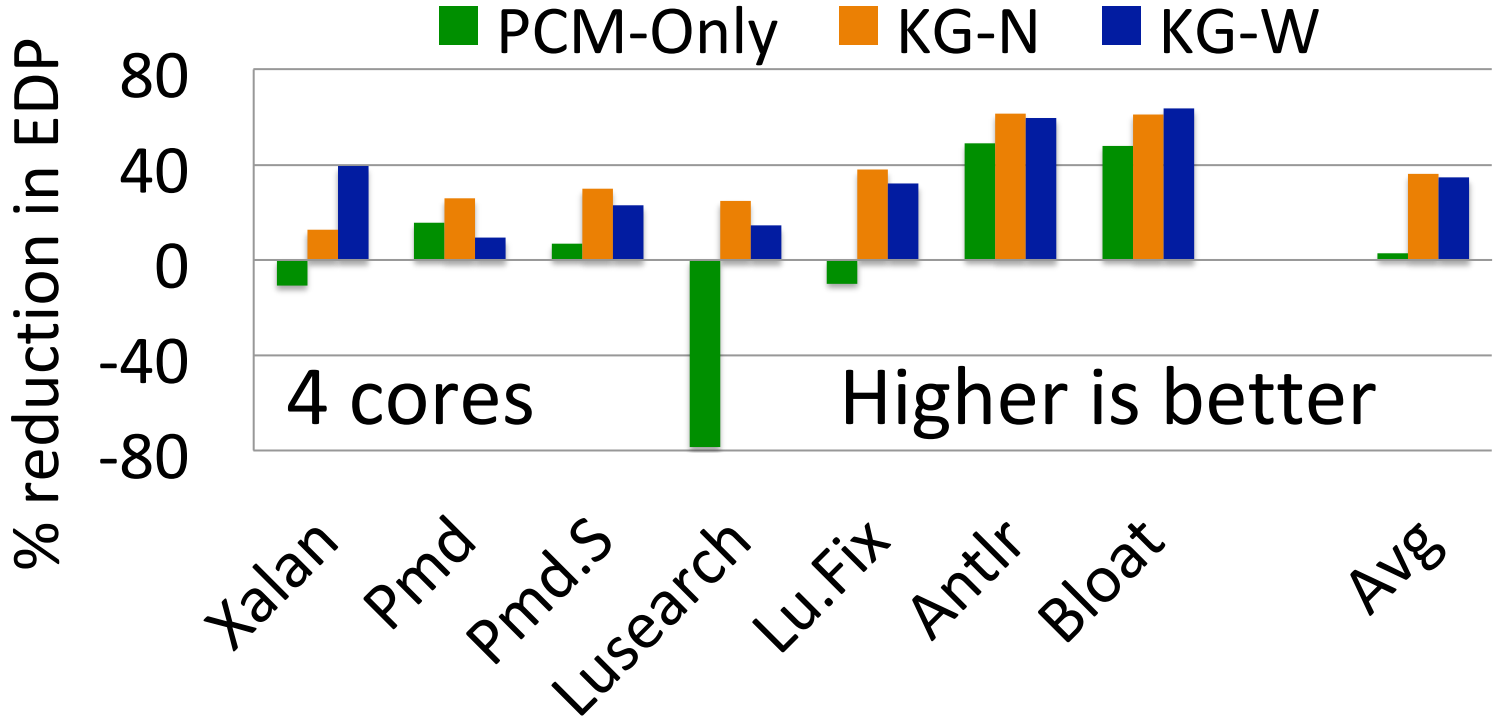
PCM lifetimes



PCM alone is not practical

PCM lasts more than 10 years with KG-W

EDP reduction compared to DRAM



EDP : Energy Delay Product

KG-W has 35% better EDP than DRAM-Only

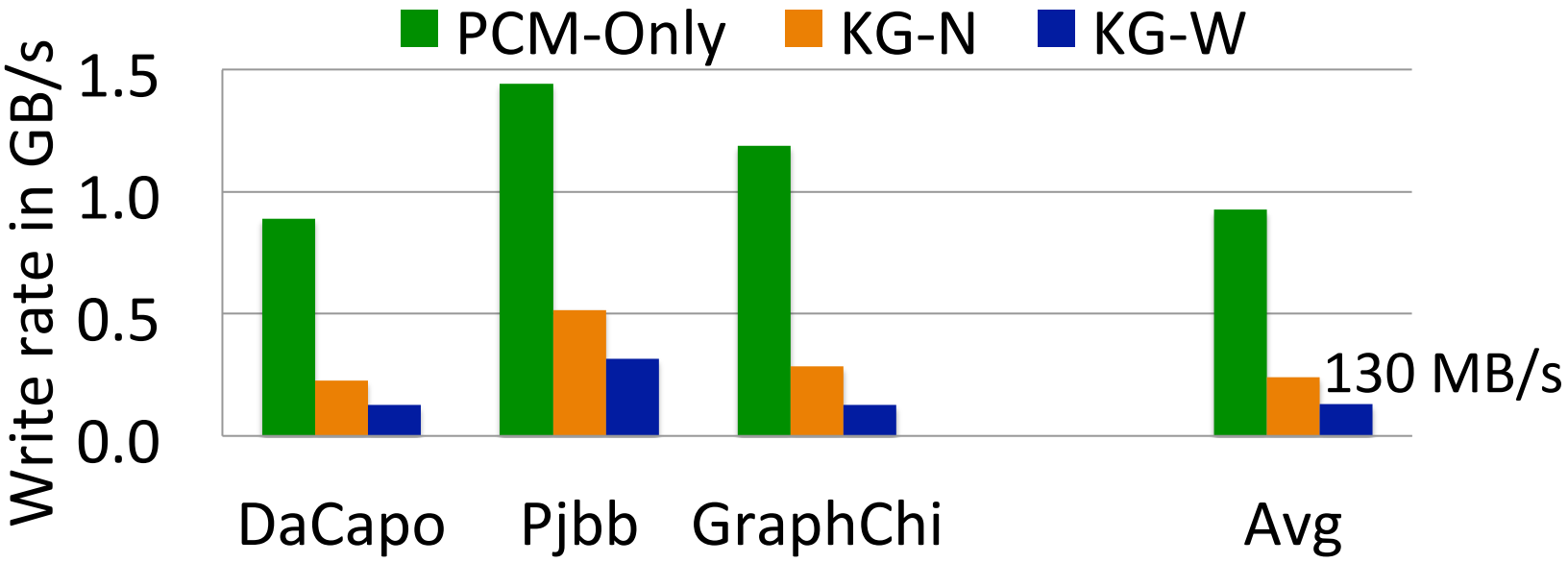
Emulation on NUMA hardware

DRAM: Socket 0 PCM: Socket 1

Modi

Use Intel perf monitor to measure writes

PCM write rates on NUMA hardware



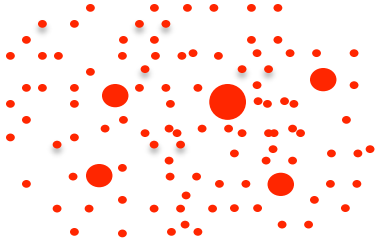
KG-N reduces write rate by 3.8X over PCM-Only

KG-W reduces write rate by 1.9X over KG-N

Crystal Gazer: Profile-Driven Write-Rationing Garbage Collection for Hybrid Memories

Takeaways

Promising to monitor heaps at a fine granularity



Write-rationing GC makes PCM practical as main memory

Similar conclusion with different evaluation methods