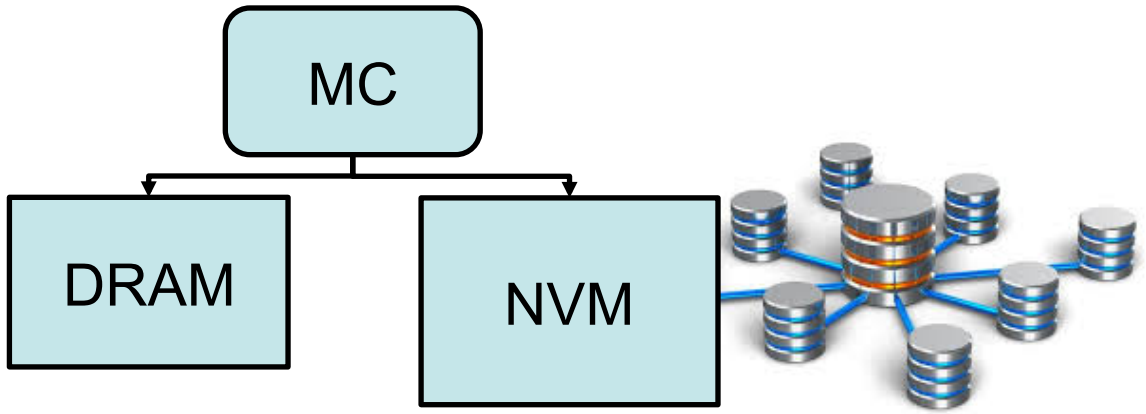
PageSeer: Using Page Walks to Trigger Page Swaps in Hybrid Memory Systems

Apostolos Kokolis, Dimitrios Skarlatos and Josep Torrellas
University of Illinois at Urbana-Champaign
http://iacoma.cs.uiuc.edu
HPCA 2019, Feb 2019

Introduction

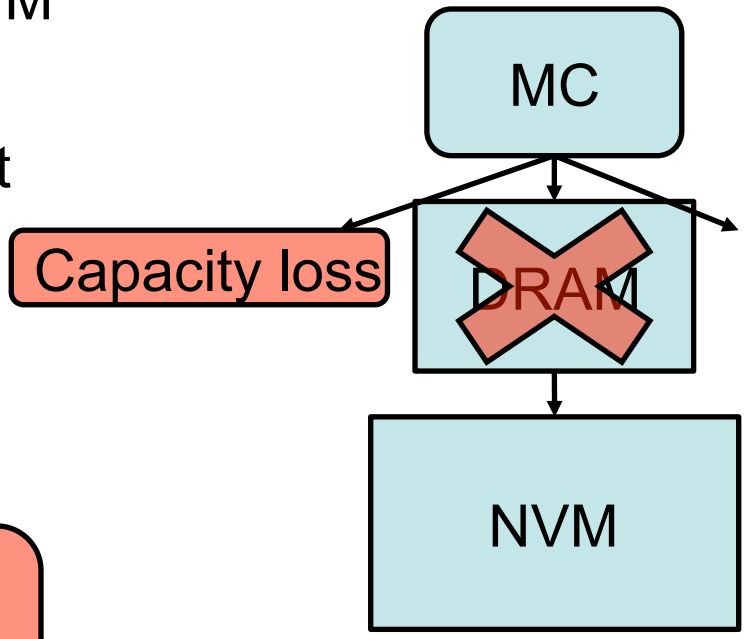
- Data intensive applications demand memory capacity
- DRAM can no longer provide the capacity needed
- Non-volatile memory (NVM) technologies
  - **+** higher density
  - **-** slower

  } Cannot replace DRAM entirely
- **Solution:** Hybrid Memory Systems



Managing a Hybrid Memory System

1. DRAM is a cache for the NVM
2. DRAM and NVM share a flat address space



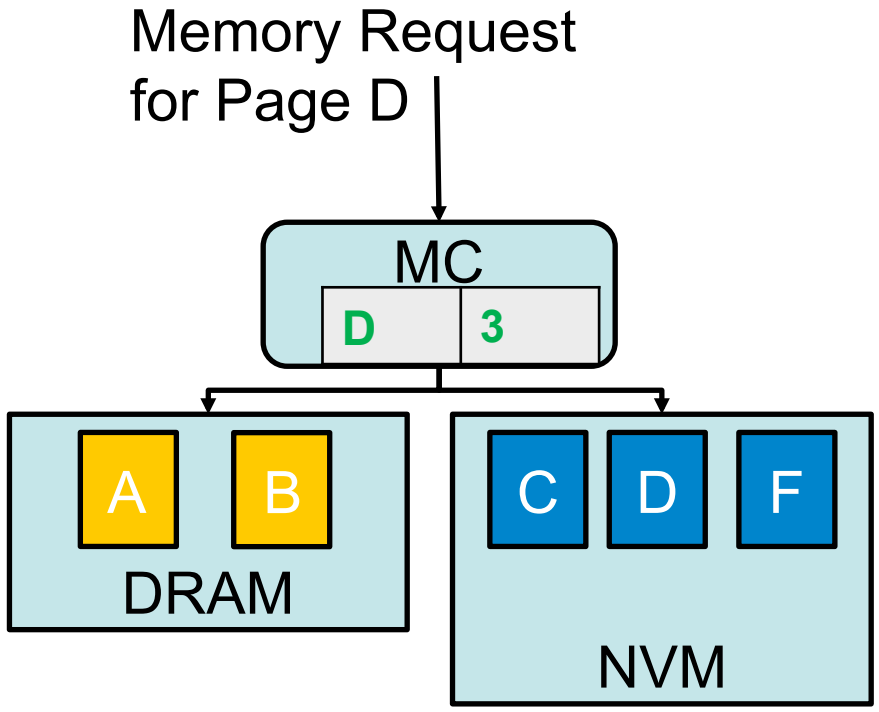
More capacity & BW

Need to swap memory between NVM and DRAM

Challenges of HW Managed Hybrid Schemes

HW needs to

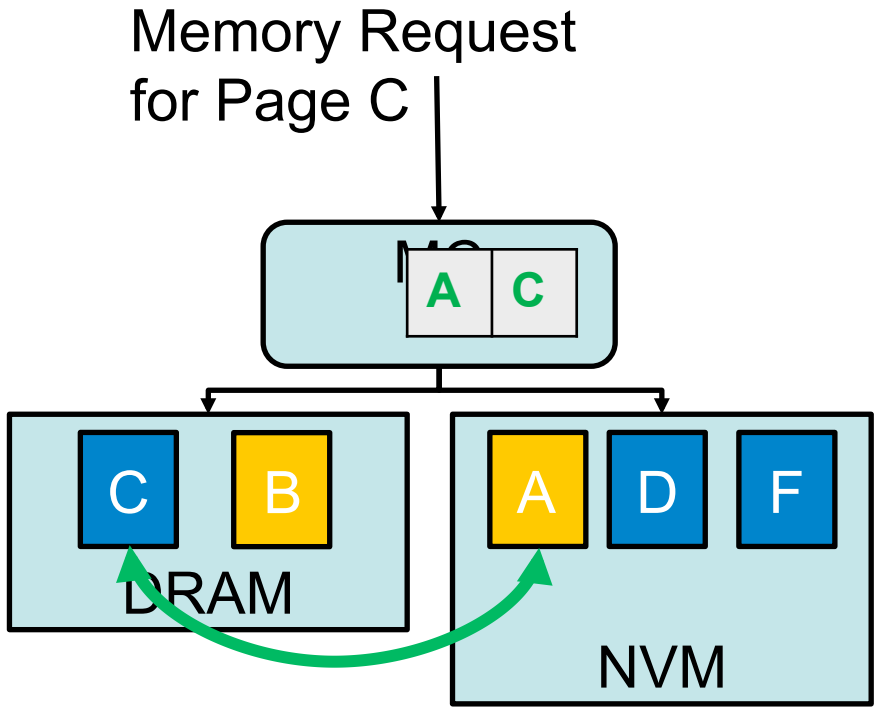
1. Swap pages
   - Decide which pages to swap
2. Track page activity
   - Accurately identify “hot” pages to swap
3. Record the page remappings



Challenges of HW Managed Hybrid Schemes

HW needs to

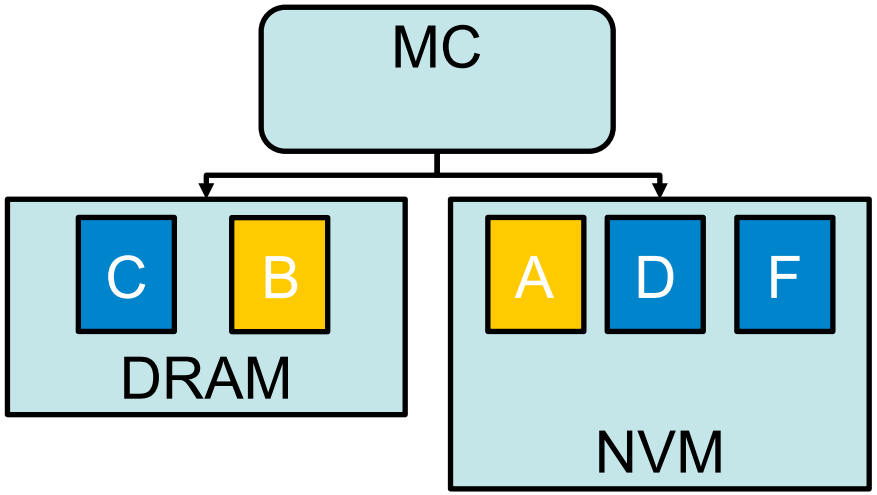
1. Swap pages
   - Decide which pages to swap
2. Track page activity
   - Accurately identify “hot” pages to swap
3. Record the page remappings



Challenges of HW Managed Hybrid Schemes

HW needs to

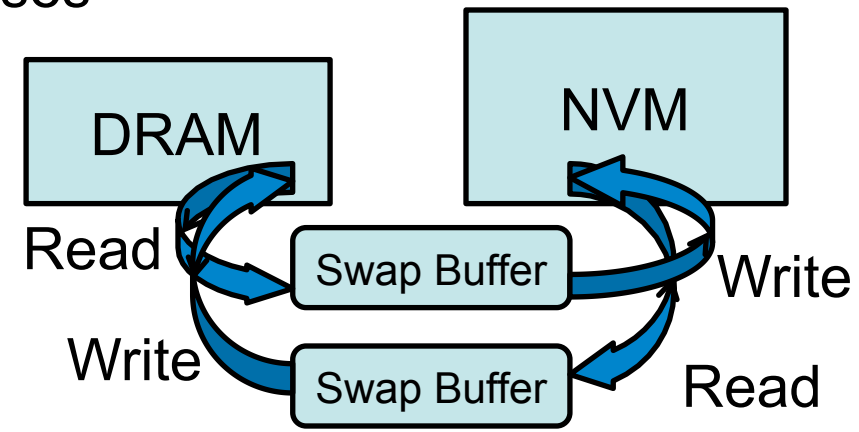
1. Swap pages
   - Decide which pages to swap
2. Track page activity
   - Accurately identify “hot” pages to swap
3. Record the page remappings
4. Store meta-data information
   - Remappings
   - Page activity



Prior Work

- Focused on identifying "hot" pages to swap:
  1. Use access counters per memory segment
     - ❖ Conservative swapping → Miss opportunities
  2. Start a swap on a first access to a memory segment
     - ❖ Aggressive swapping → Unnecessary traffic
  3. Predict future memory accesses
     - ❖ Hard to predict

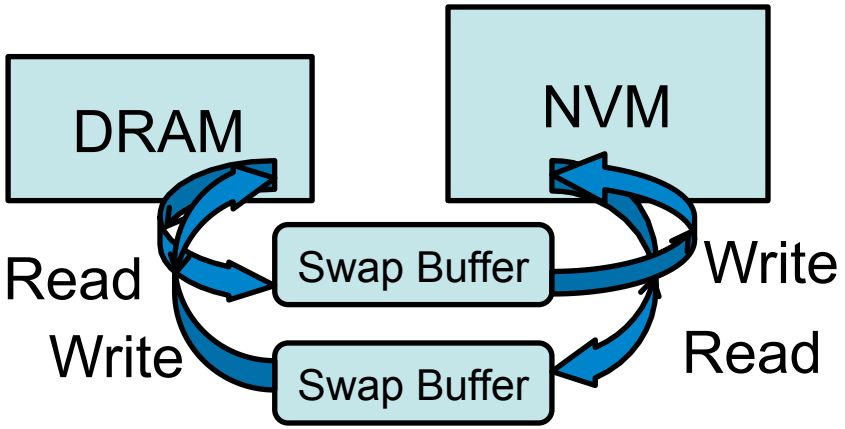
Segment ID	Counter
A	5 ≥ Threshold
B	4
C	2



PageSeer Motivation

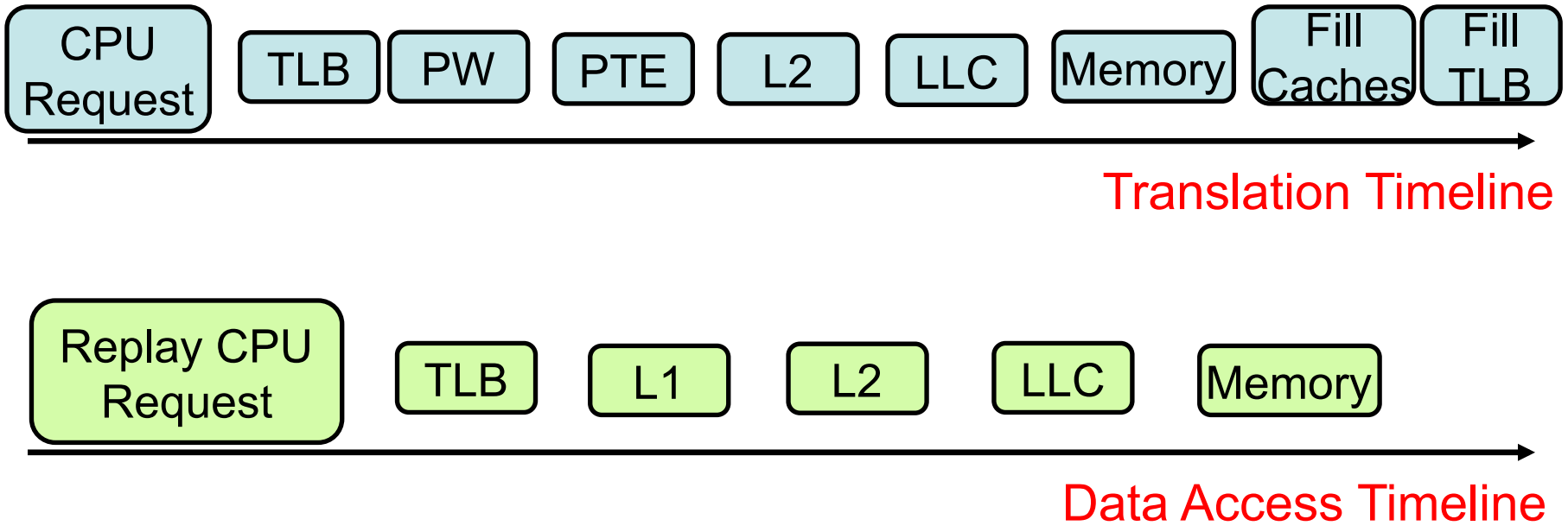
- Swapping is a costly operation and takes time
- Need accuracy in predicting future memory accesses and swapping
- Need to start the swaps as early as possible

❖ *How to predict future memory accesses?*



PageSeer Motivation

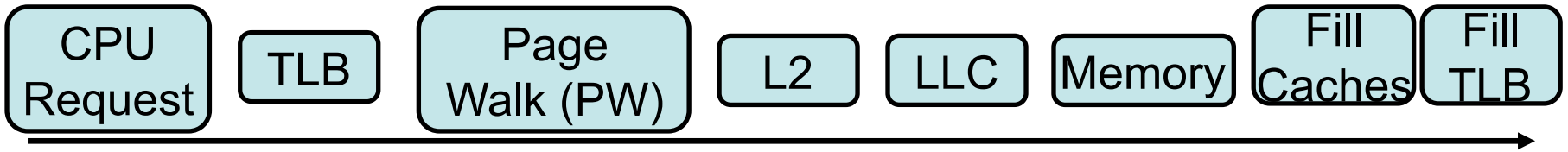
- Memory requests require two steps
  - Translation from VA to PA
  - Data access



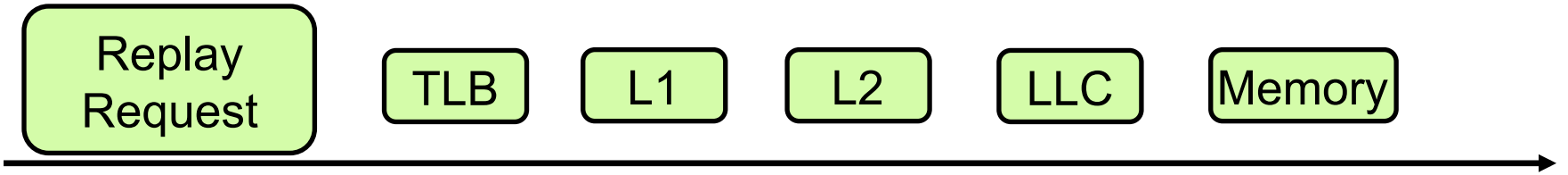
PageSeer Motivation

- Memory intensive applications cause TLB misses
- TLB miss results in a *page walk*
- If the page is cold
  - Page translation
  - Actual page data

Miss in caches



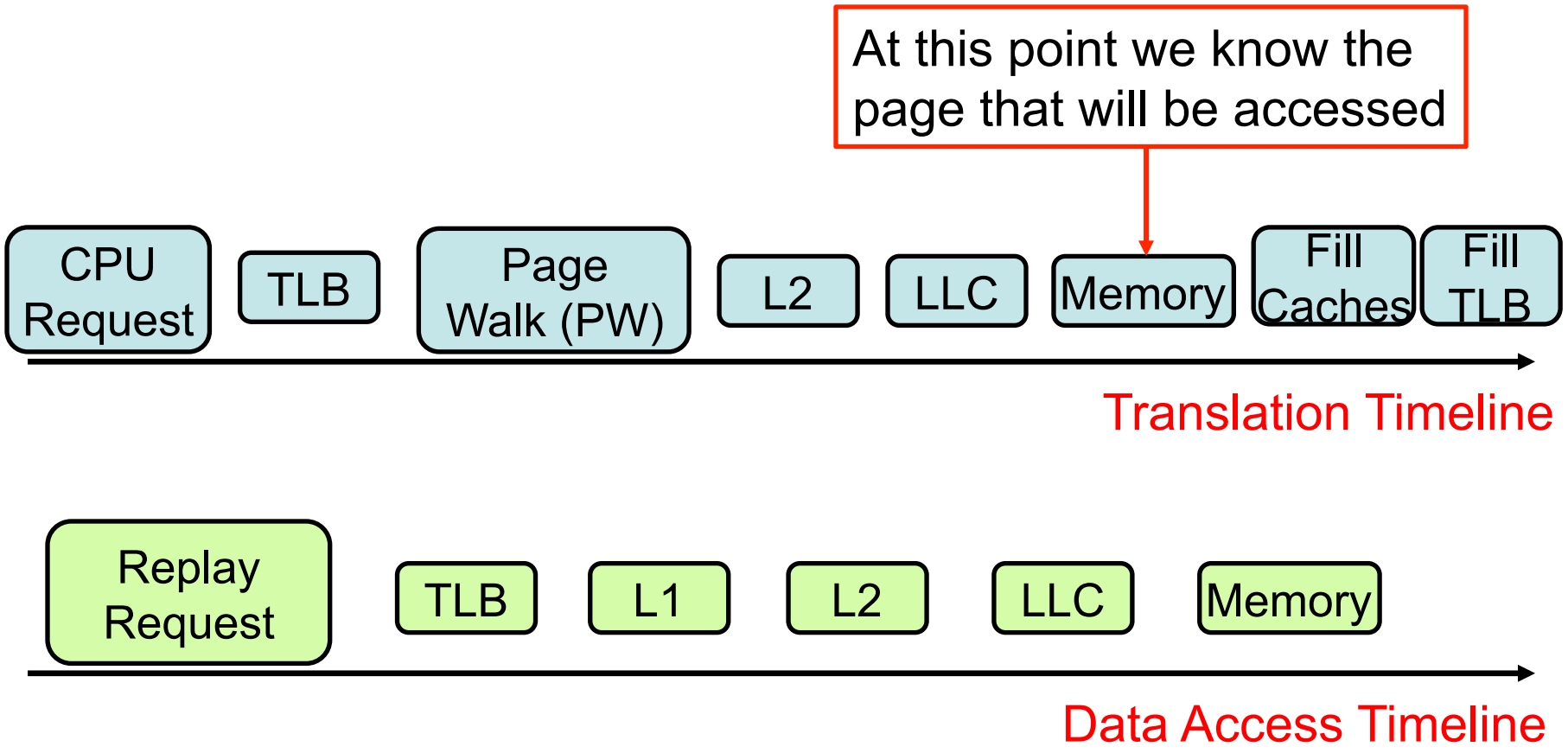
Translation Timeline



Data Access Timeline

PageSeer Motivation

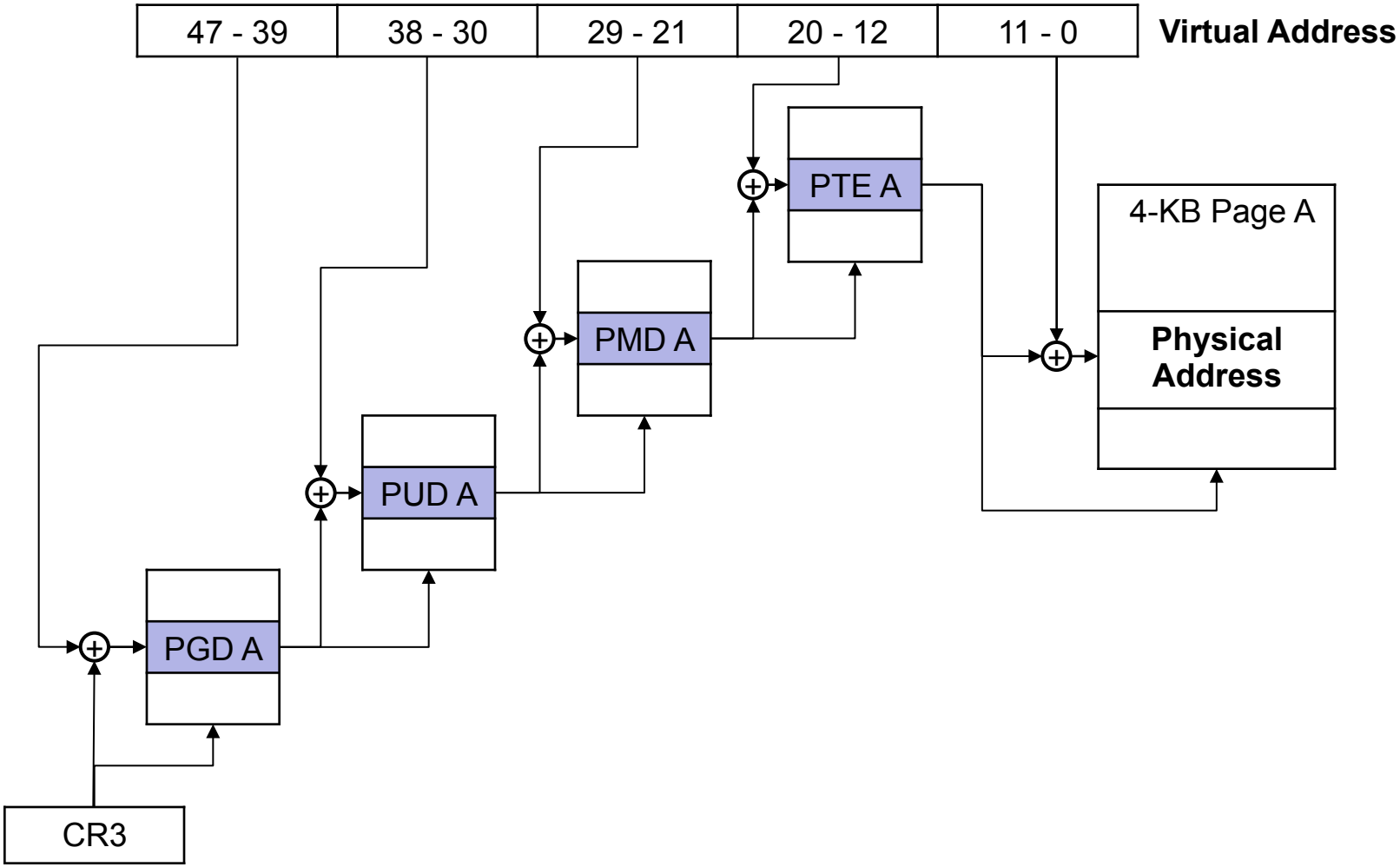
Insight: *Page Walks give information about future accesses to a page earlier*



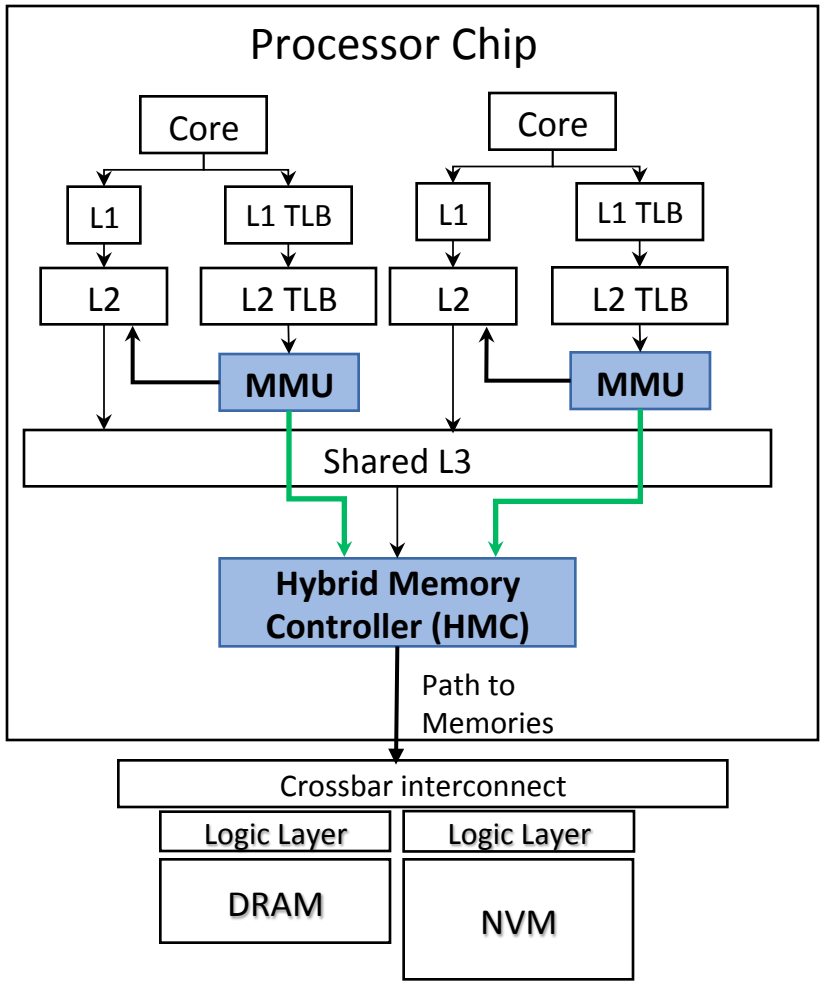
Contribution: **PageSeer**

- Uses pages walks to trigger page swaps in HW-only hybrid memory systems
- Uses a page-correlation mechanism to prefetch page swaps
- Includes HW structures to recognize and move “hot” pages to DRAM
- Improves performance by 19% and reduces Average Main Memory Access Time (AMMAT) by 29% over state-of-the-art

Background – Page Walk



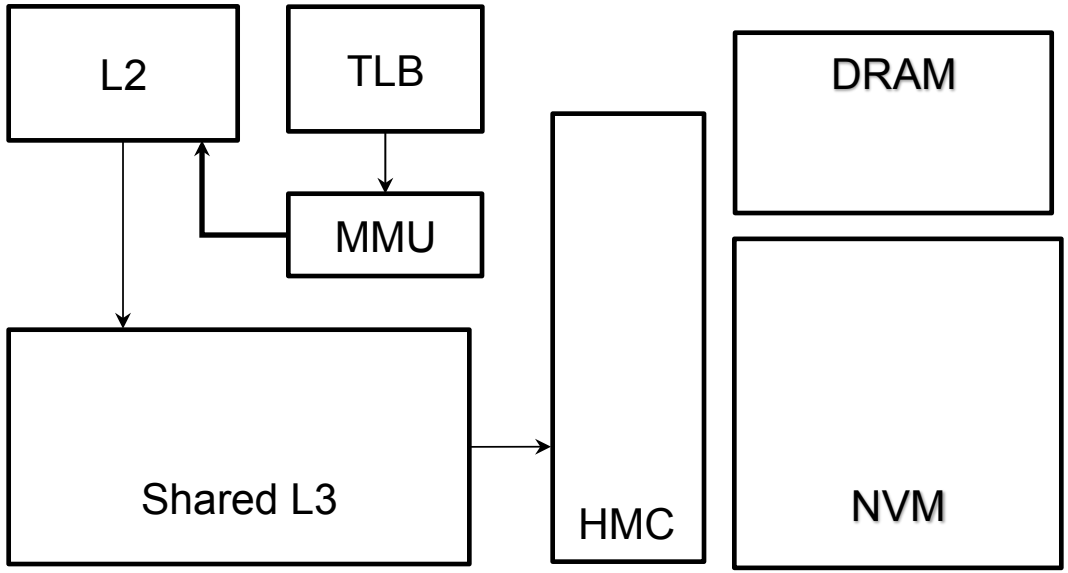
PageSeer Design



- Communication link between the MMU and the Hybrid Memory Controller
- When a Page Walk reaches the PTE level
  - Conventional: MMU sends a memory request to the caches
  - PageSeer: in addition MMU sends it to the HMC

❖ *Inform the HMC about forthcoming memory accesses*

PageSeer Overview

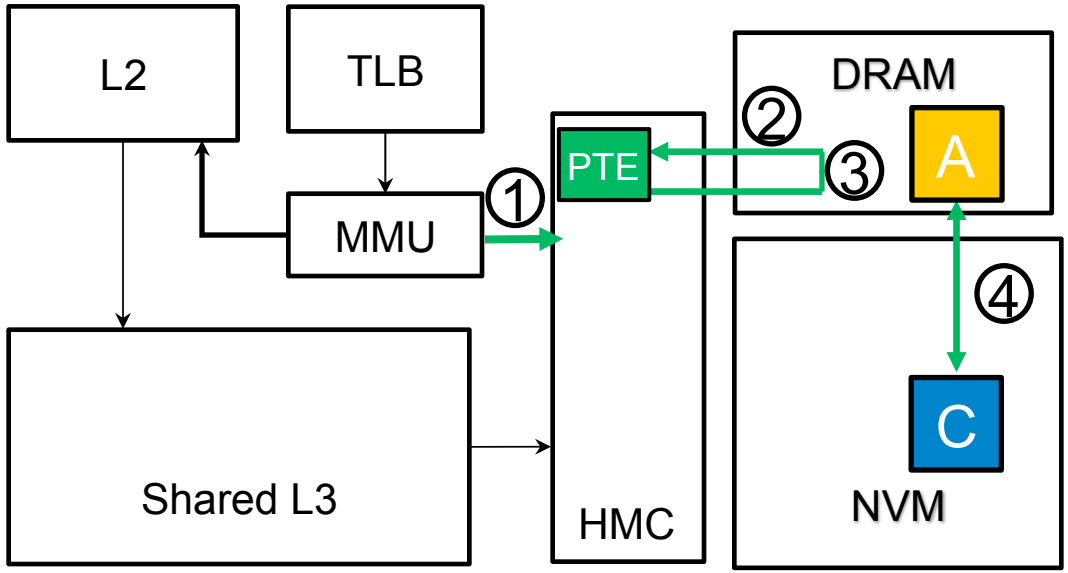


Goal: give time to the HMC for swaps and to prepare its HW structs

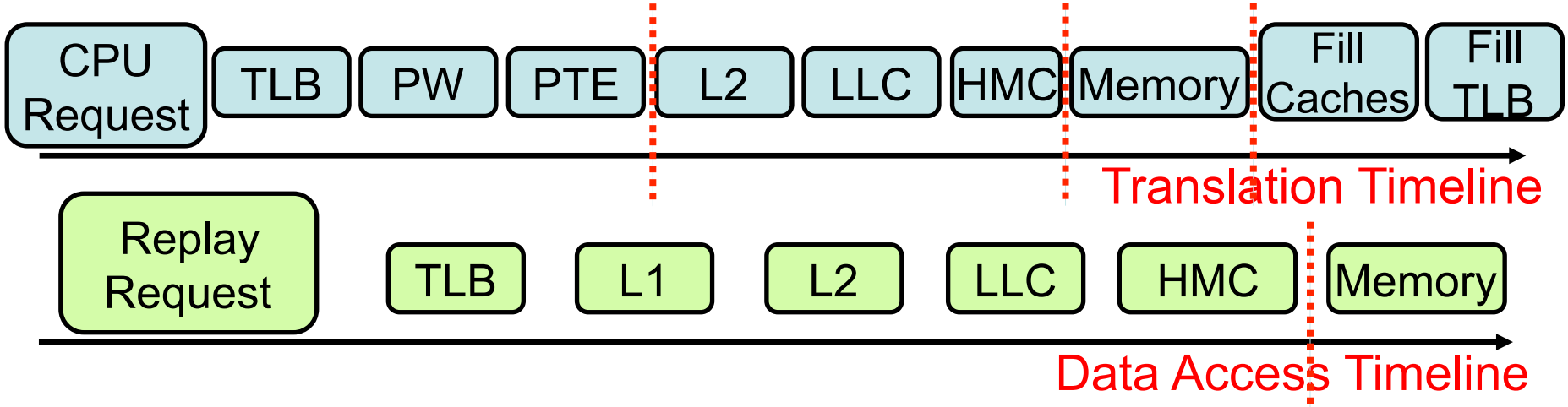
Translation Timeline

Data Access Timeline

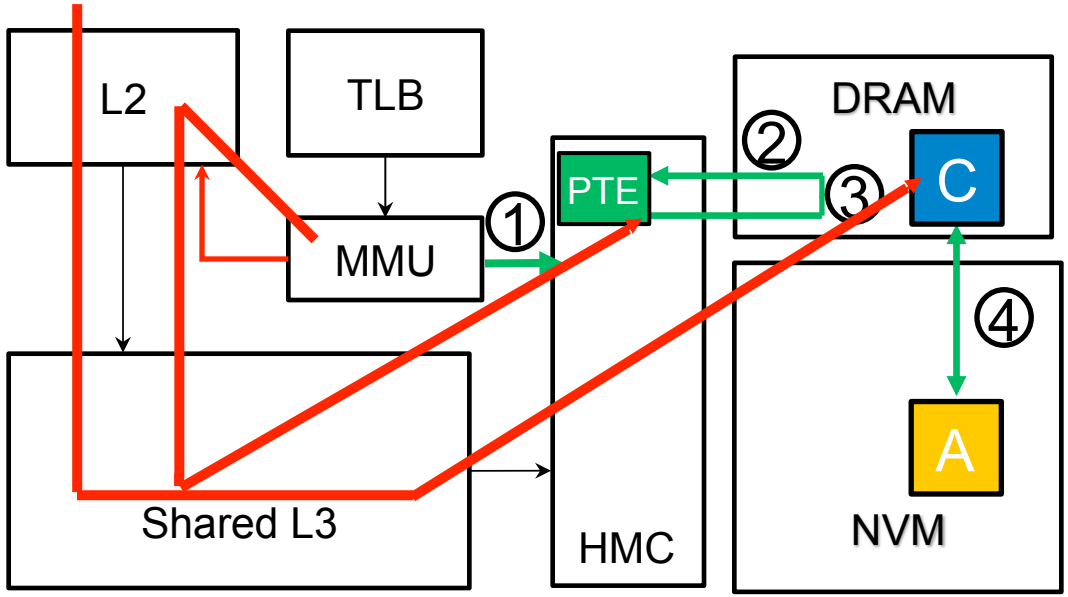
PageSeer Overview



1. MMU signals the HMC
2. HMC finds physical page number
3. HMC prepares its HW structures
4. Starts swapping



PageSeer Overview



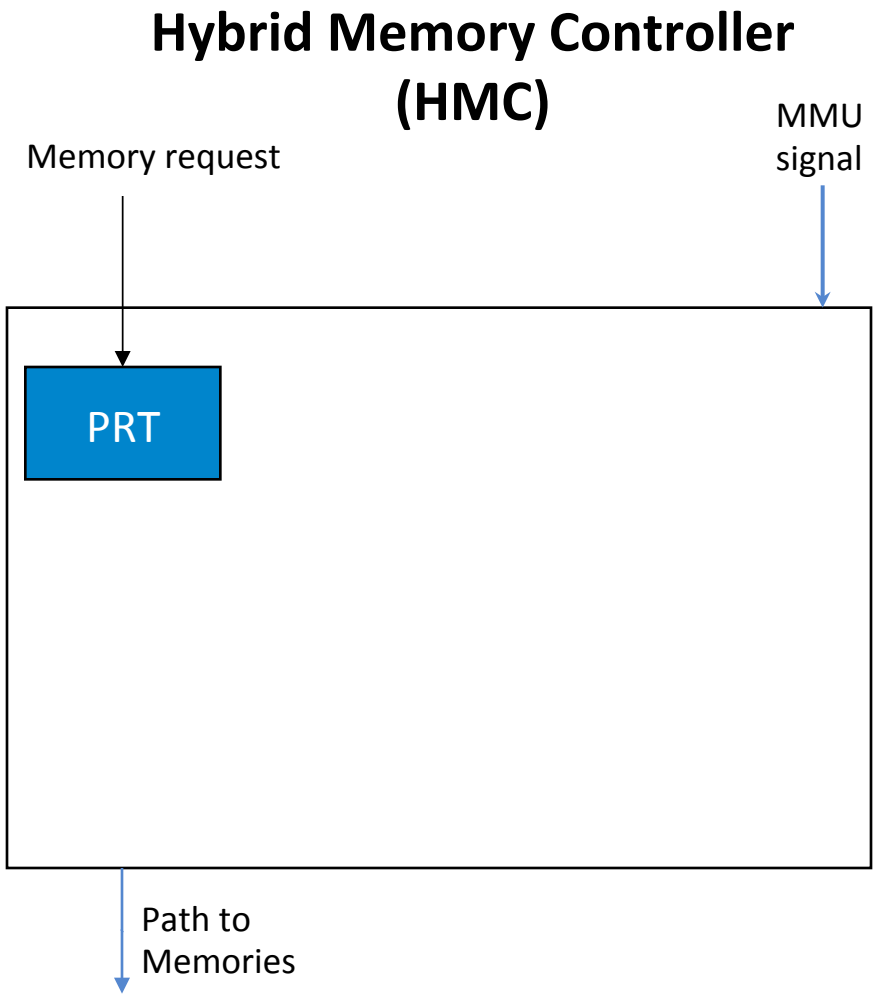
1. MMU signals the HMC
2. HMC finds physical page number
3. HMC prepares its HW structures
4. Starts swapping

Benefits:

- If the MMU request reaches HMC → PTE is already prefetched
- When the request is replayed → page swap to DRAM has already started

Data Access Timeline

Hybrid Memory Controller

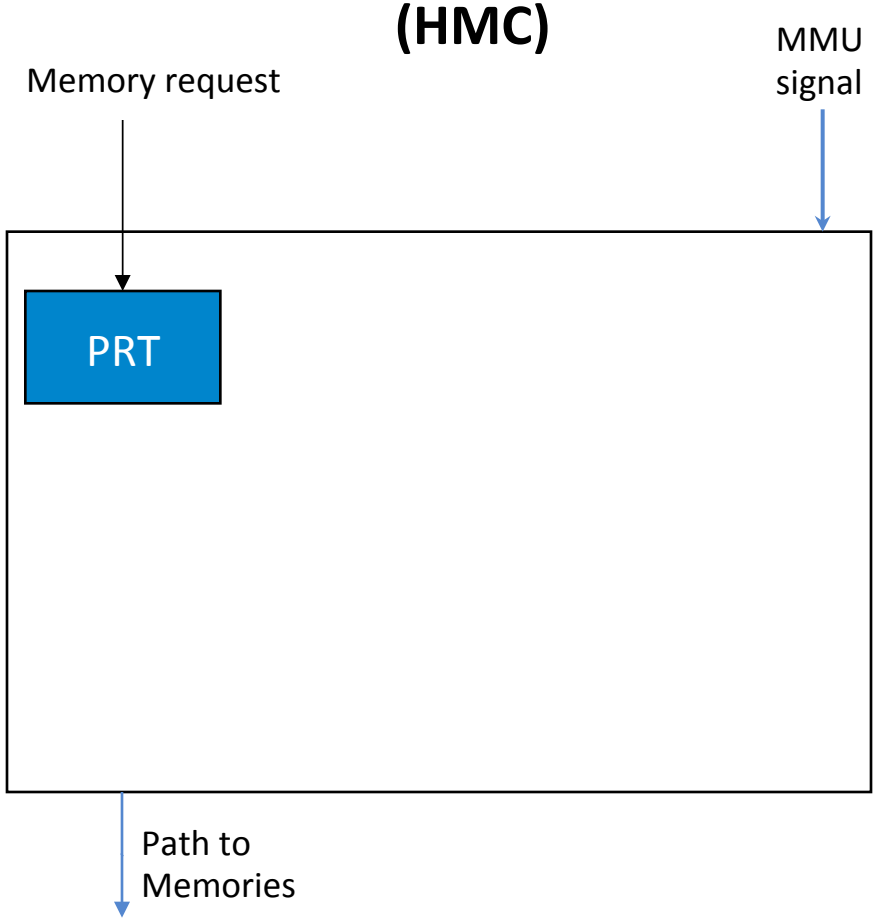


Page Remapping Table (PRT)

- Records remappings between DRAM-NVM pages
- On the critical path
- Every memory request checks the PRT

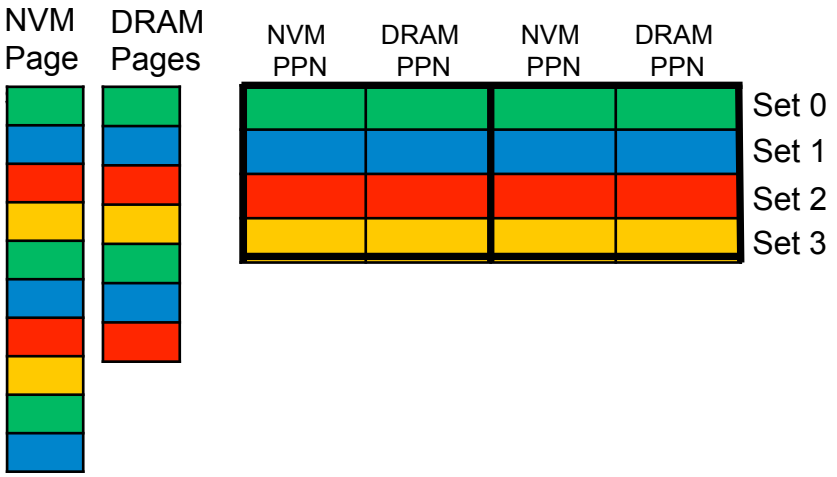
Hybrid Memory Controller

Hybrid Memory Controller (HMC)

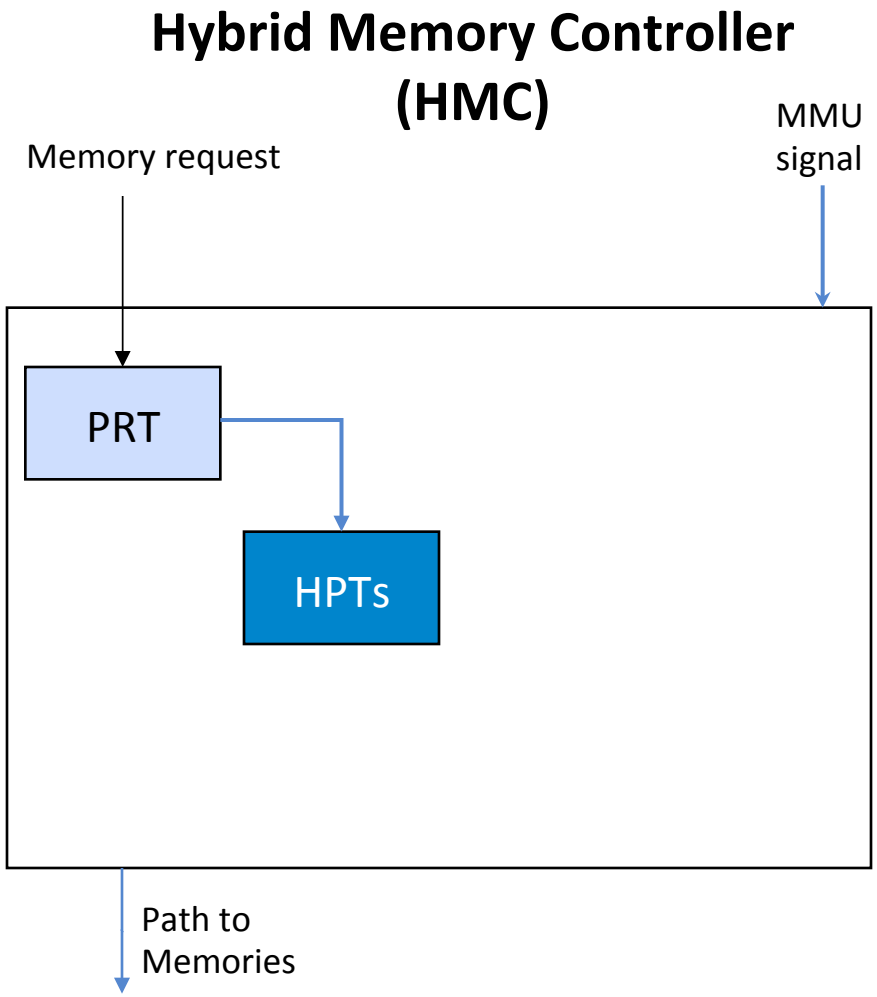


Page Remapping Table (PRT)

- Hit rate and lookup time is important => Cache some entries
- Swap at page granularity



Hybrid Memory Controller

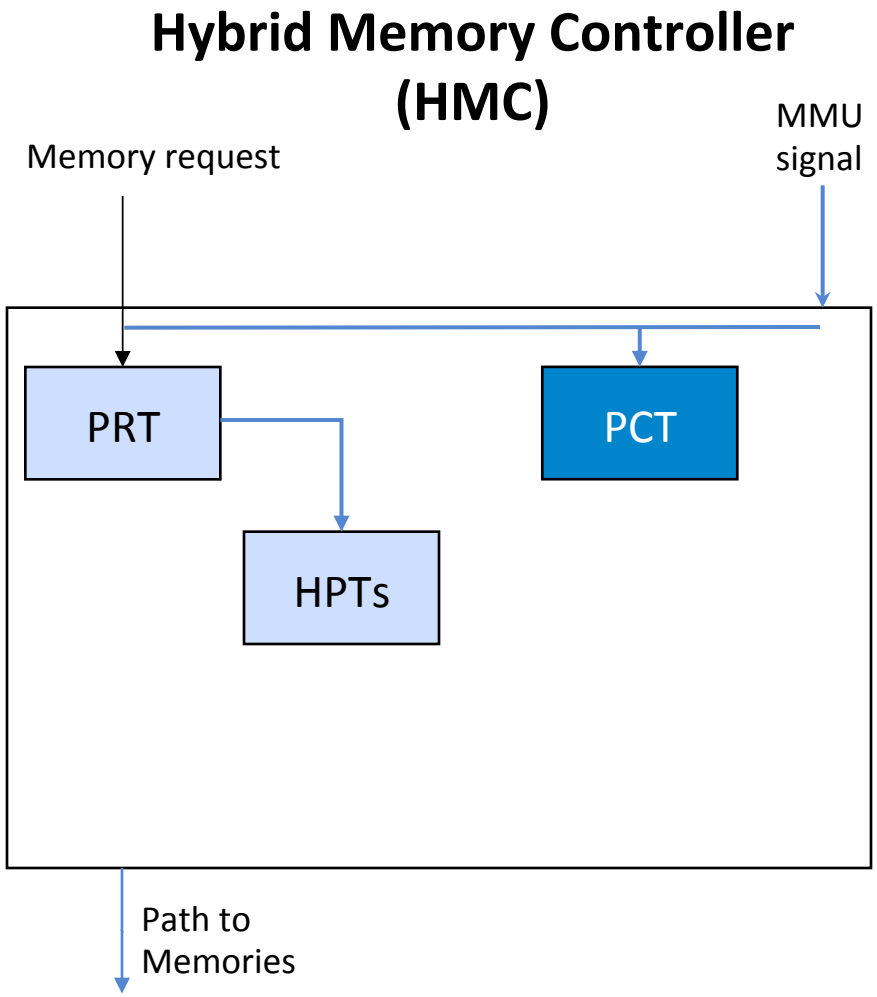


Hot Page Tables (HPTs)

- 1 for DRAM – 1 for NVM
- Track hot pages
- **DRAM HPT**
  - Pages that should remain in DRAM
- **NVM HPT**
  - Candidate pages to swap to DRAM

Page Number	Counter

Hybrid Memory Controller



Page Correlation Table (PCT)

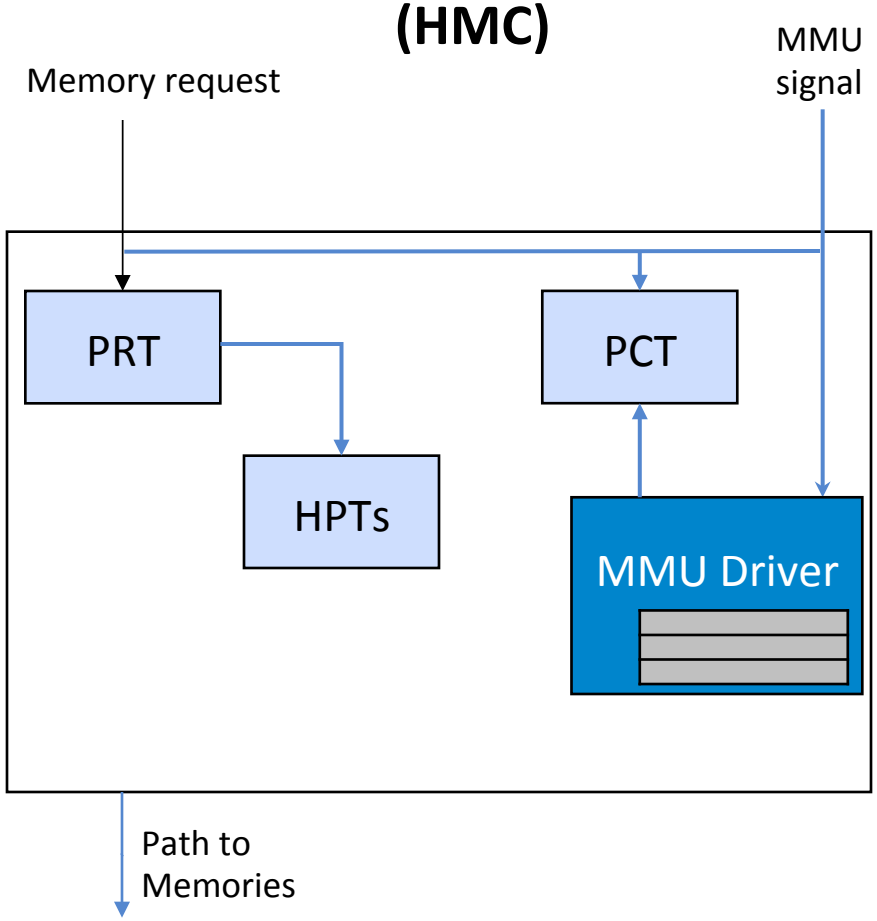
- Keeps historic data for page accesses
- Captures the correlation between pages → prefetch

PCT entry

PPN	Counter	Next PPN	Next counter

Hybrid Memory Controller

Hybrid Memory Controller (HMC)

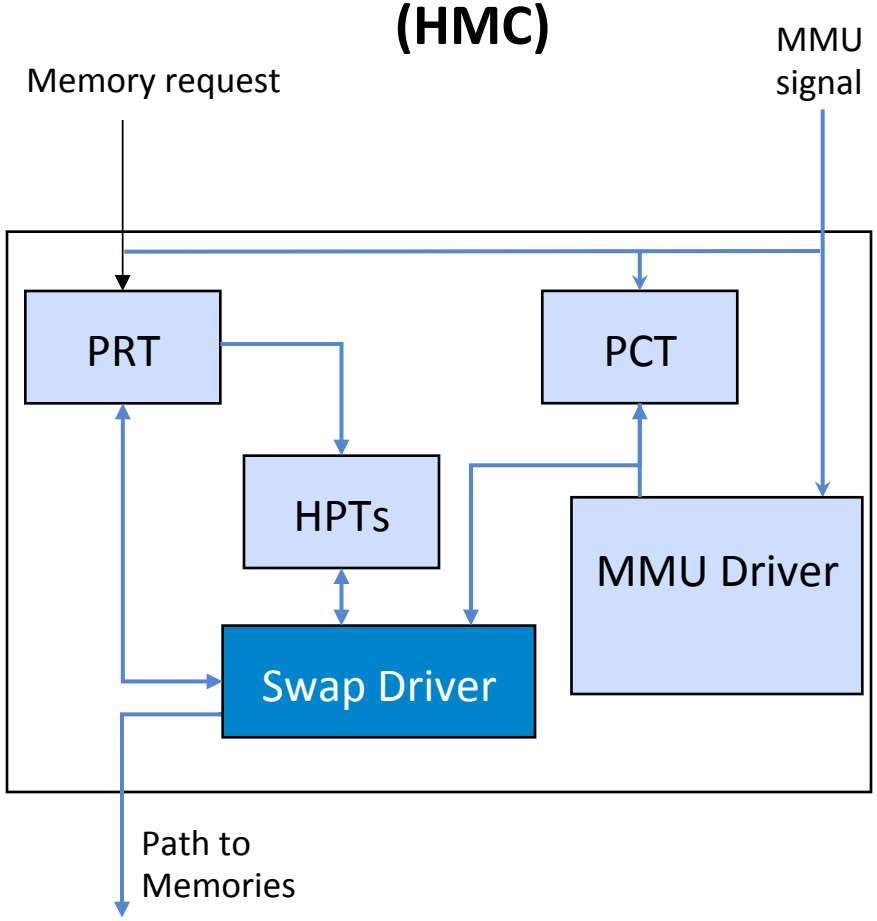


MMU Driver

- Receives the MMU signal and checks for prefetch swaps
- Receives requests for PTEs
- Caches recently fetched PTEs

Hybrid Memory Controller

Hybrid Memory Controller (HMC)



Swap Driver

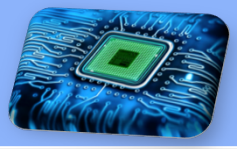
- Initiates page swaps triggered from
  - NVM HPT
  - PCT
- Checks if an access is for a page that is being swapped

Evaluation Methodology

Simulator	• Simics + SST + DRAMSim2 • Modified DRAMSim2 for NVM
Architectures Compared	• MemPod [Prodromou '18] • PoM [Sim '14] • **PageSeer**
Workloads	• 20 benchmarks (SPEC CPU2006, Splash-3 and CORAL)+ 6 mixes of the benchmarks

Evaluation Parameters

Processor



- 4 cores @ 2GHz
- 8MB shared L3

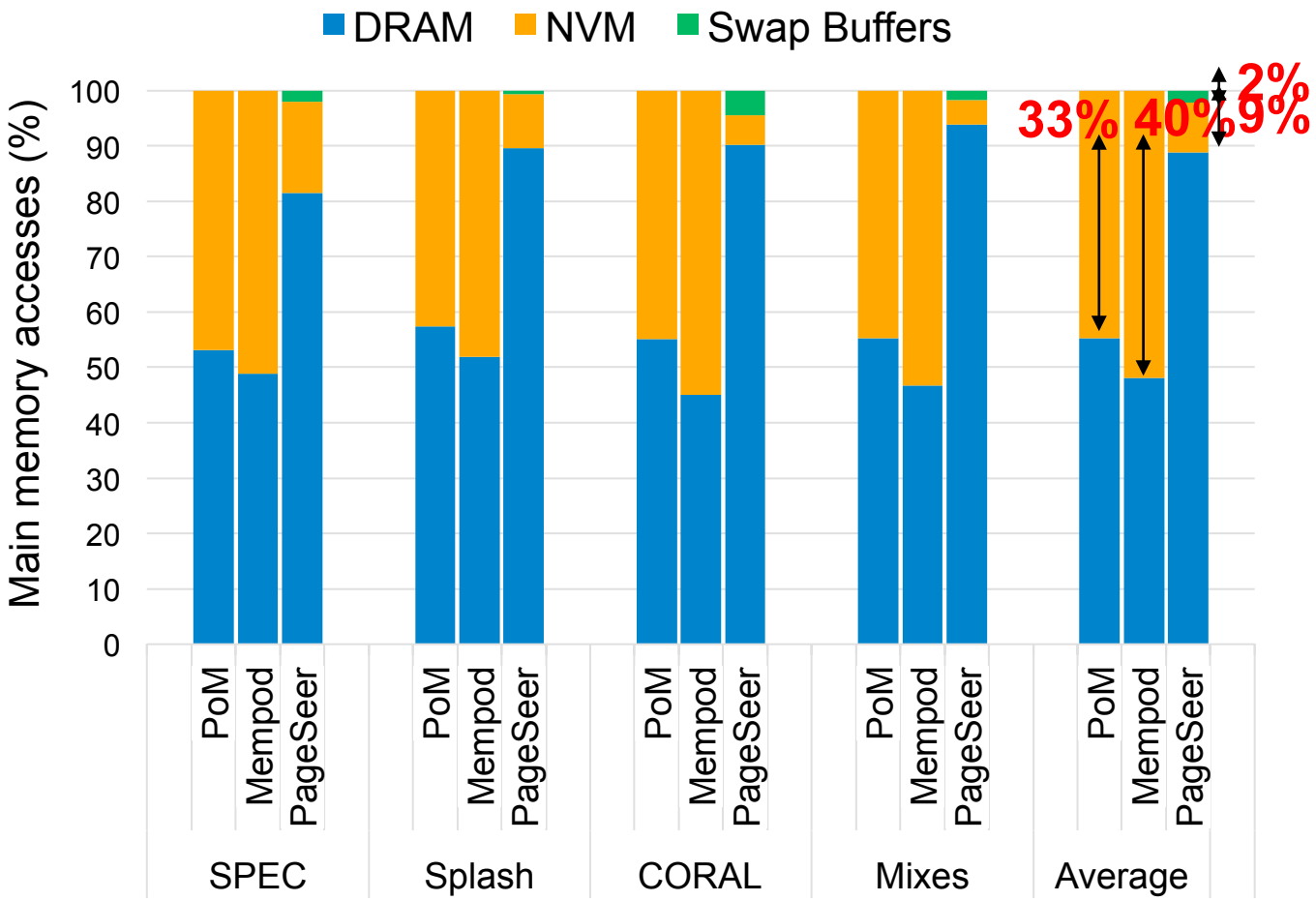
Memory

- 1 GHz DDR
- **DRAM:** 512MB, 4 channels
- **NVM:** 4GB, 2 channels

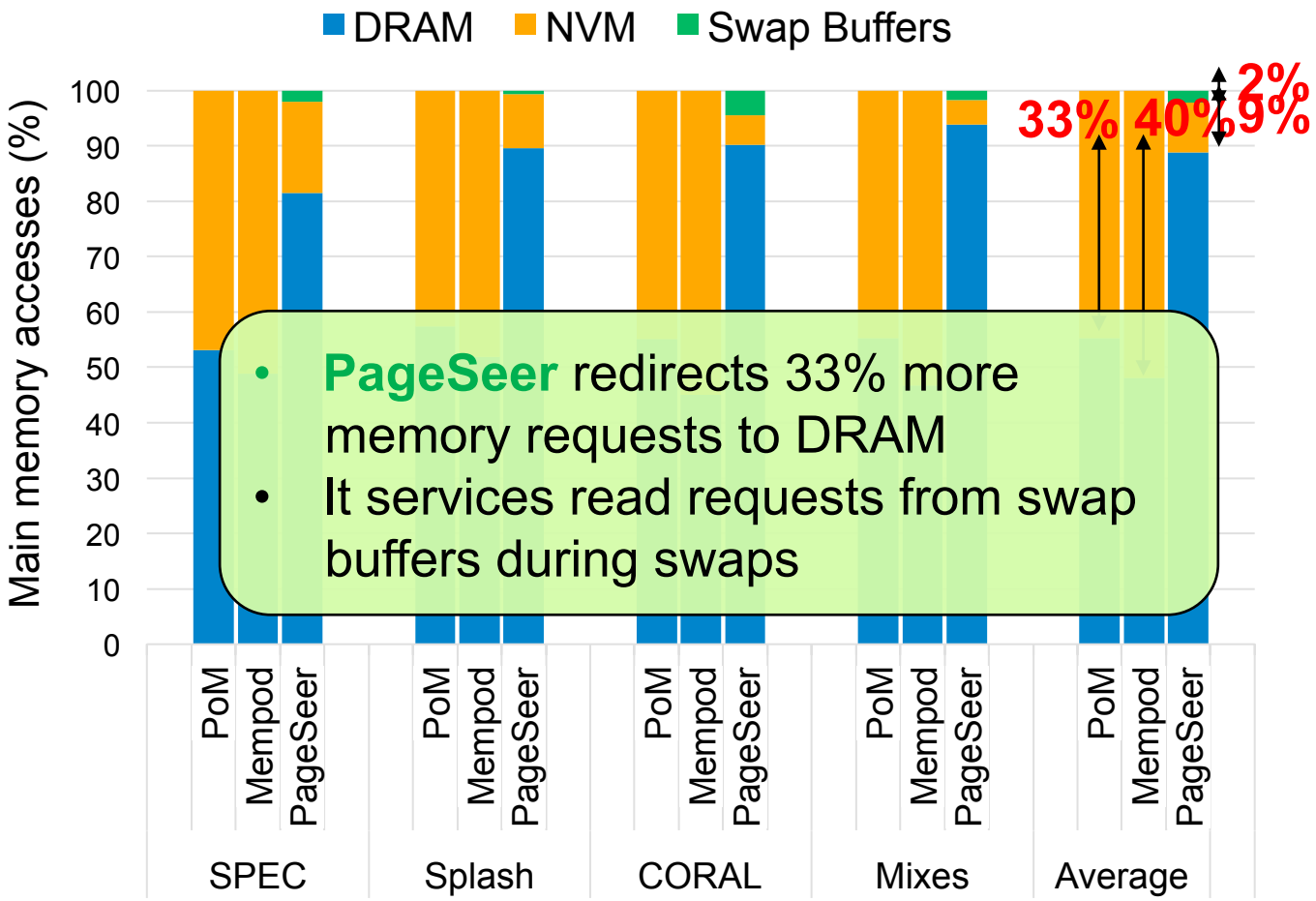
HMC

- Swap granularity 4KB
- PRT and PCT: 32KB, 4-way set associative

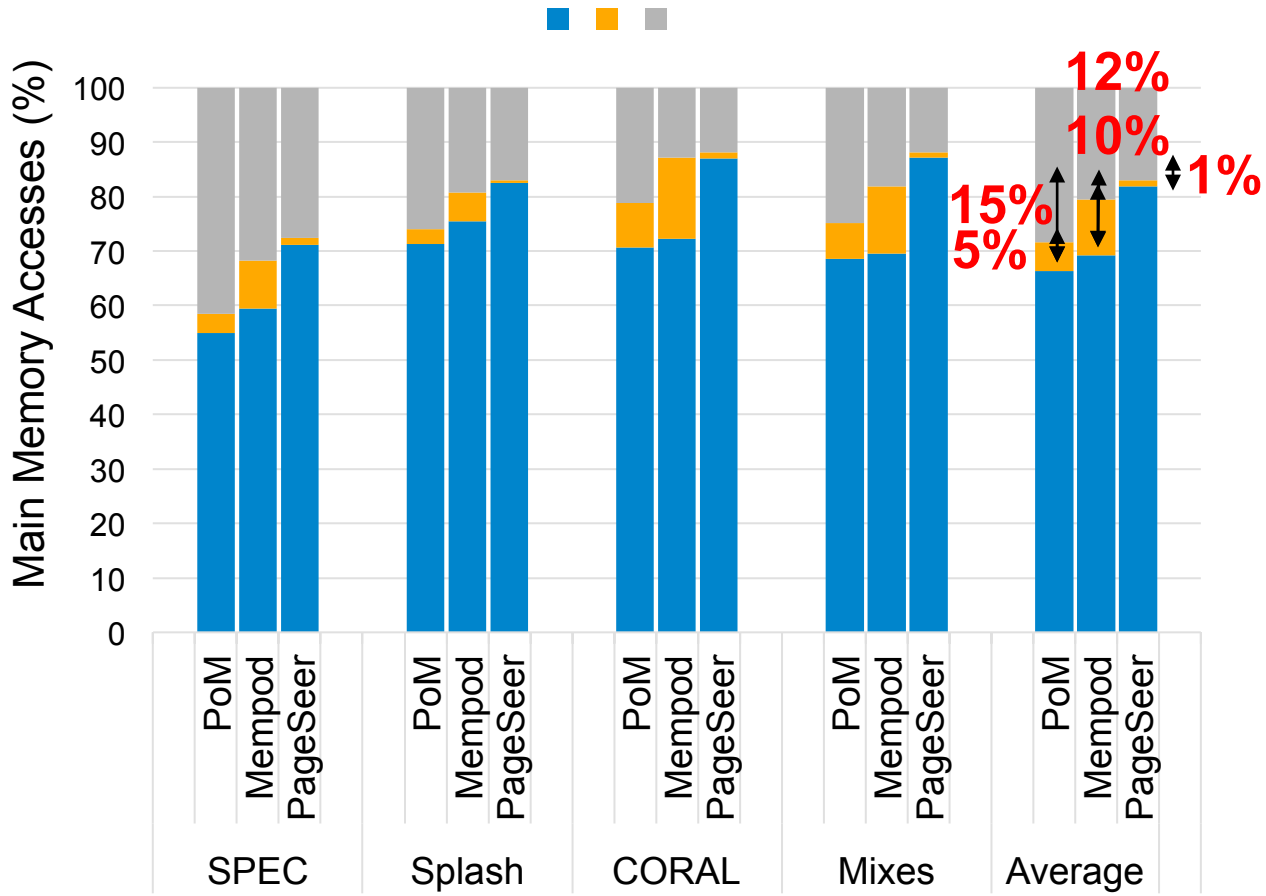
Main Memory Accesses



Main Memory Accesses



Swap Effectiveness



Positive

Access DRAM thanks to a swap

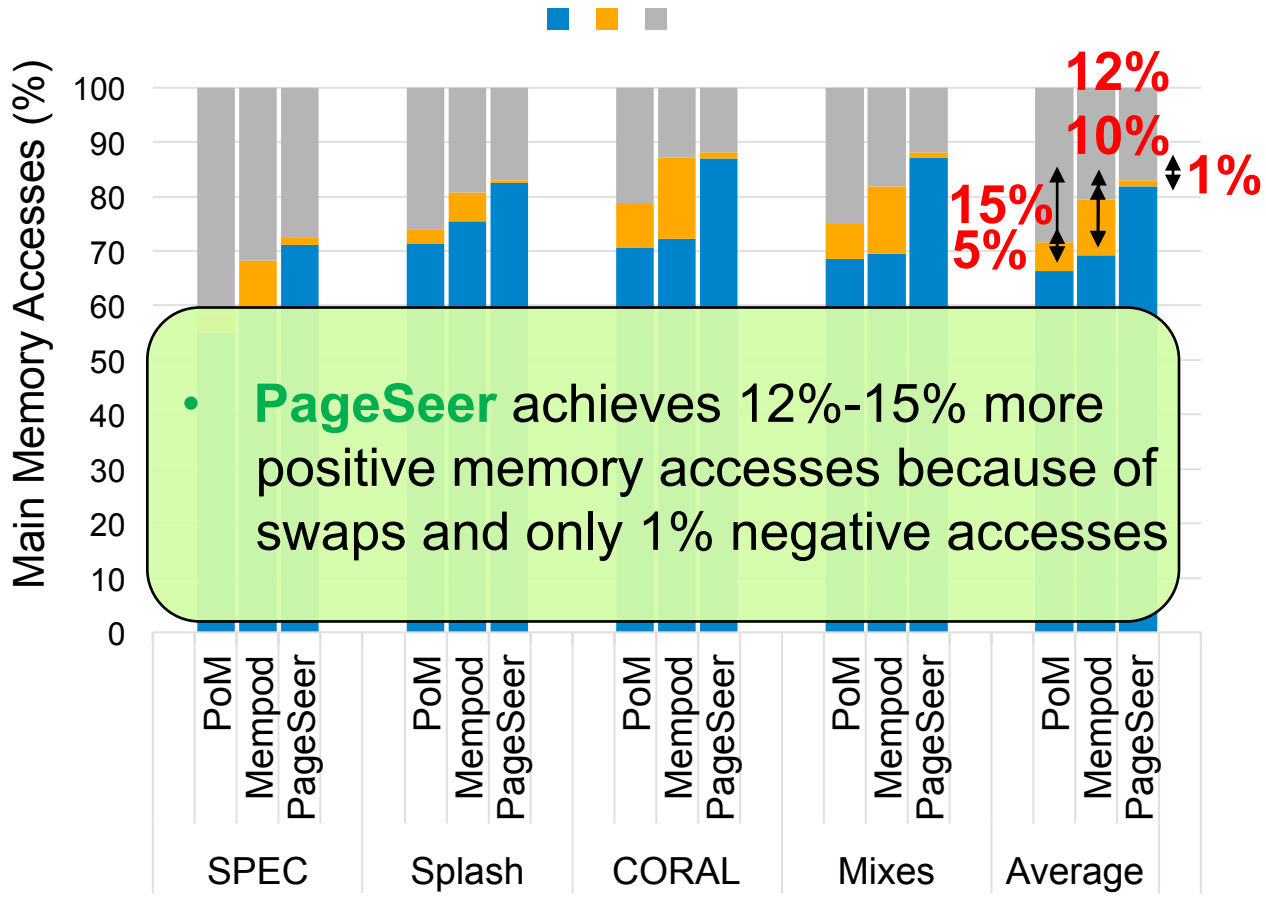
Negative

Access NVM due to a swap

Neutral

Access DRAM or NVM as if no swap happened

Swap Effectiveness



Positive

Access DRAM thanks to a swap

Negative

Access NVM due to a swap

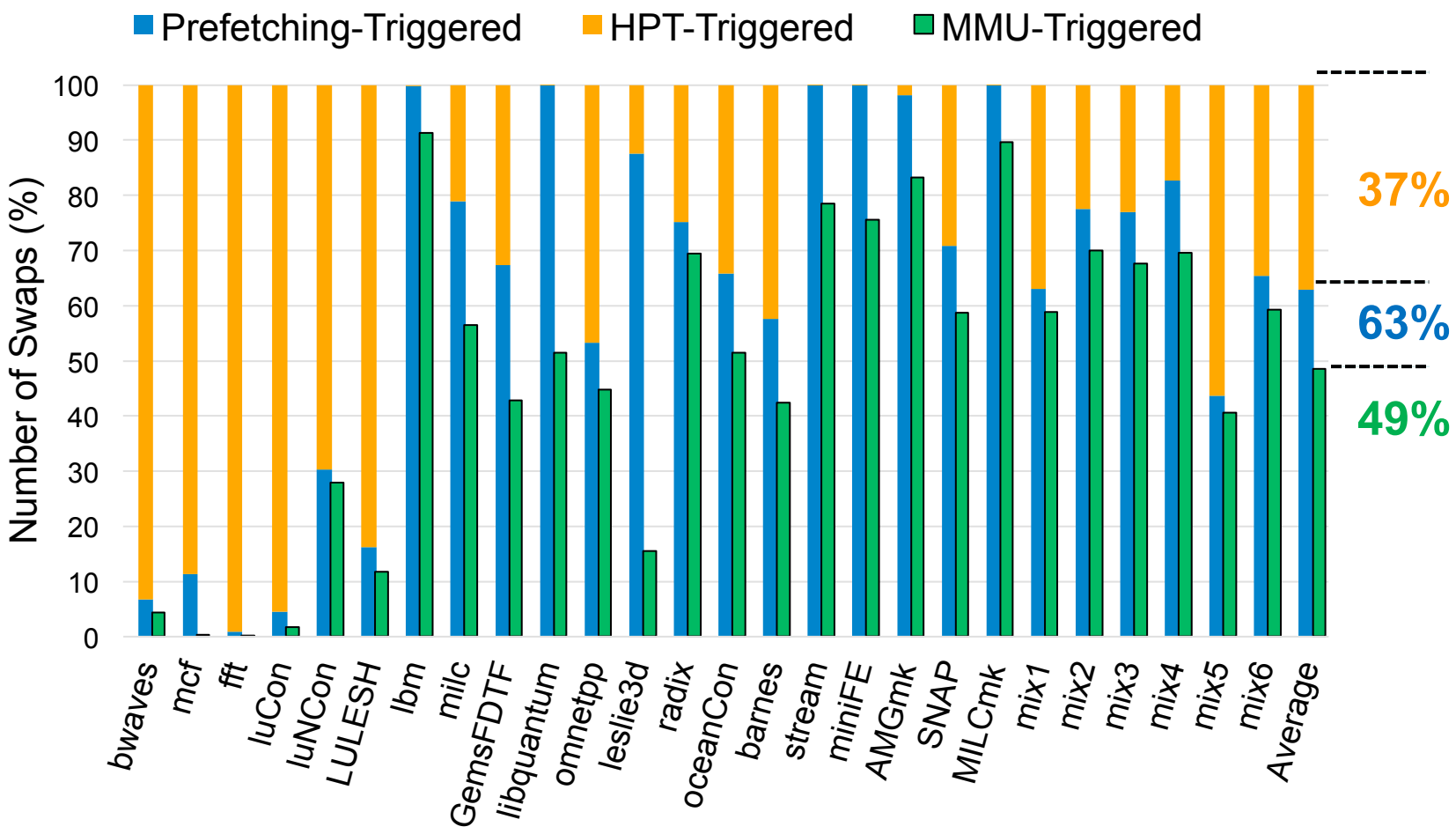
Neutral

Access DRAM or NVM as if no swap happened

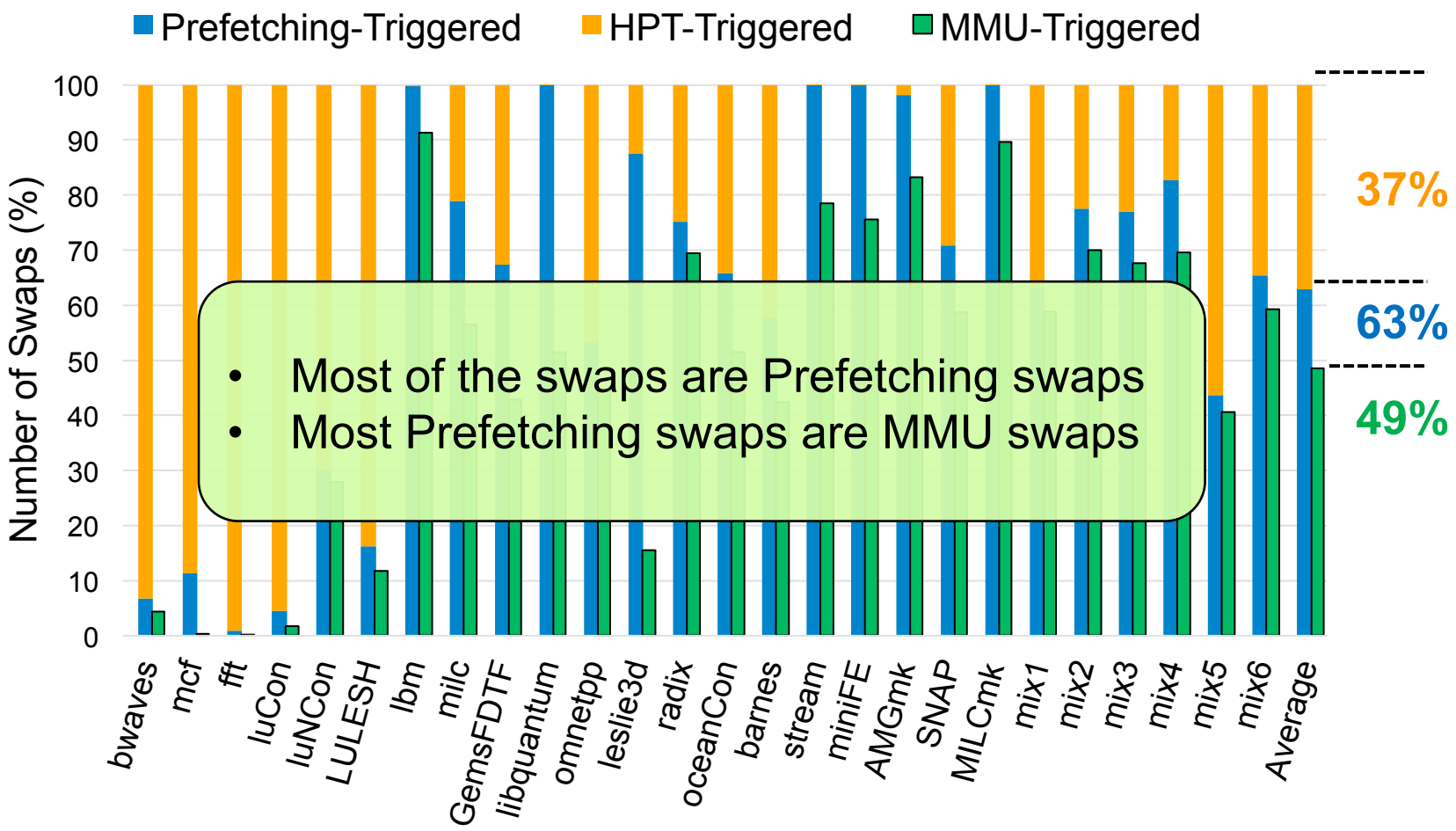
Types of swaps

- Prefetching-Triggered: Aggressive swaps triggered by the Page Correlation Table based on historic access patterns
  - MMU-Triggered: Subset of Prefetching Triggered directly initiated by the MMU
- HPT-Triggered: Conservative swaps triggered by the NVM Hot Page Table when accesses to a page reach a threshold

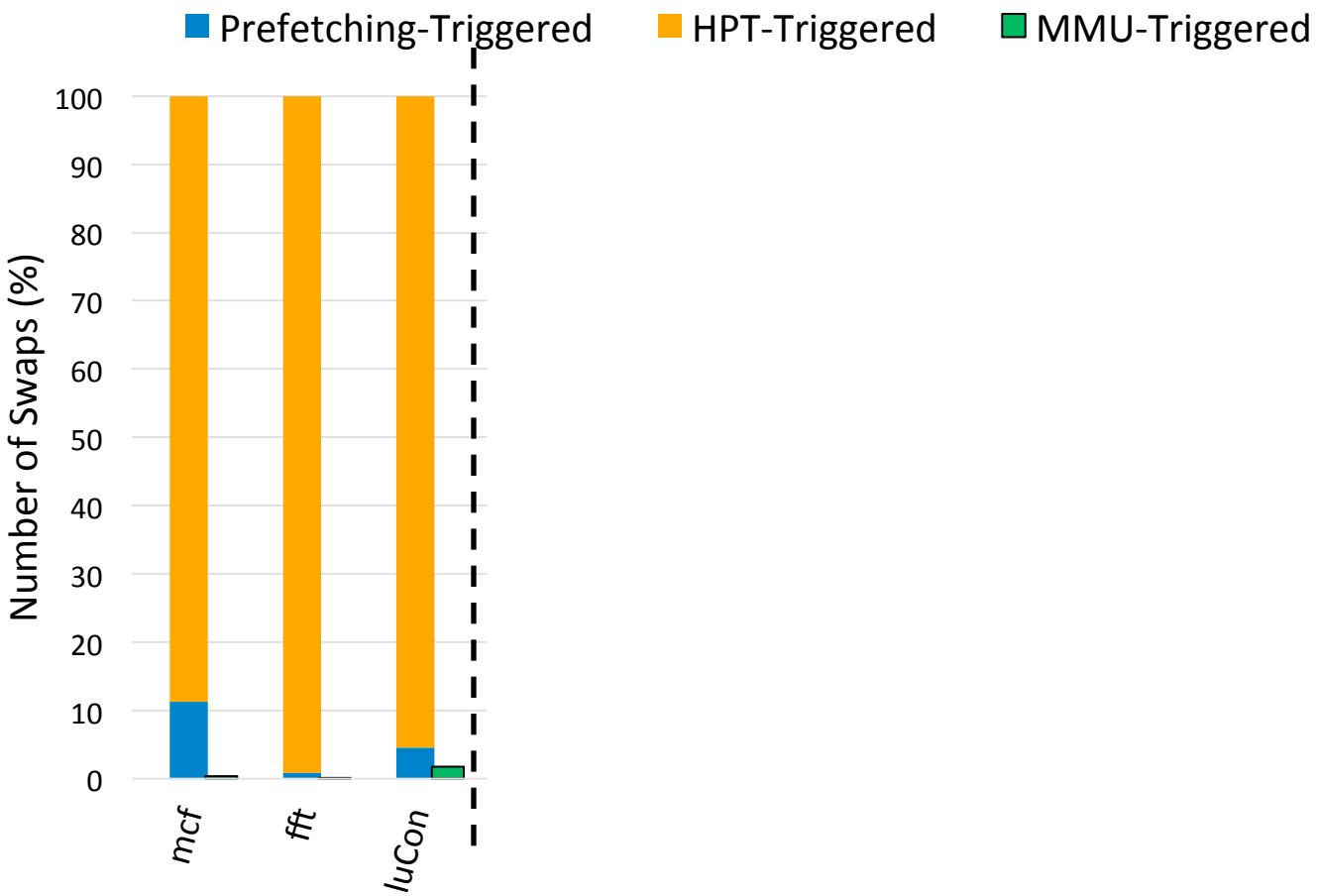
Swap Characterization



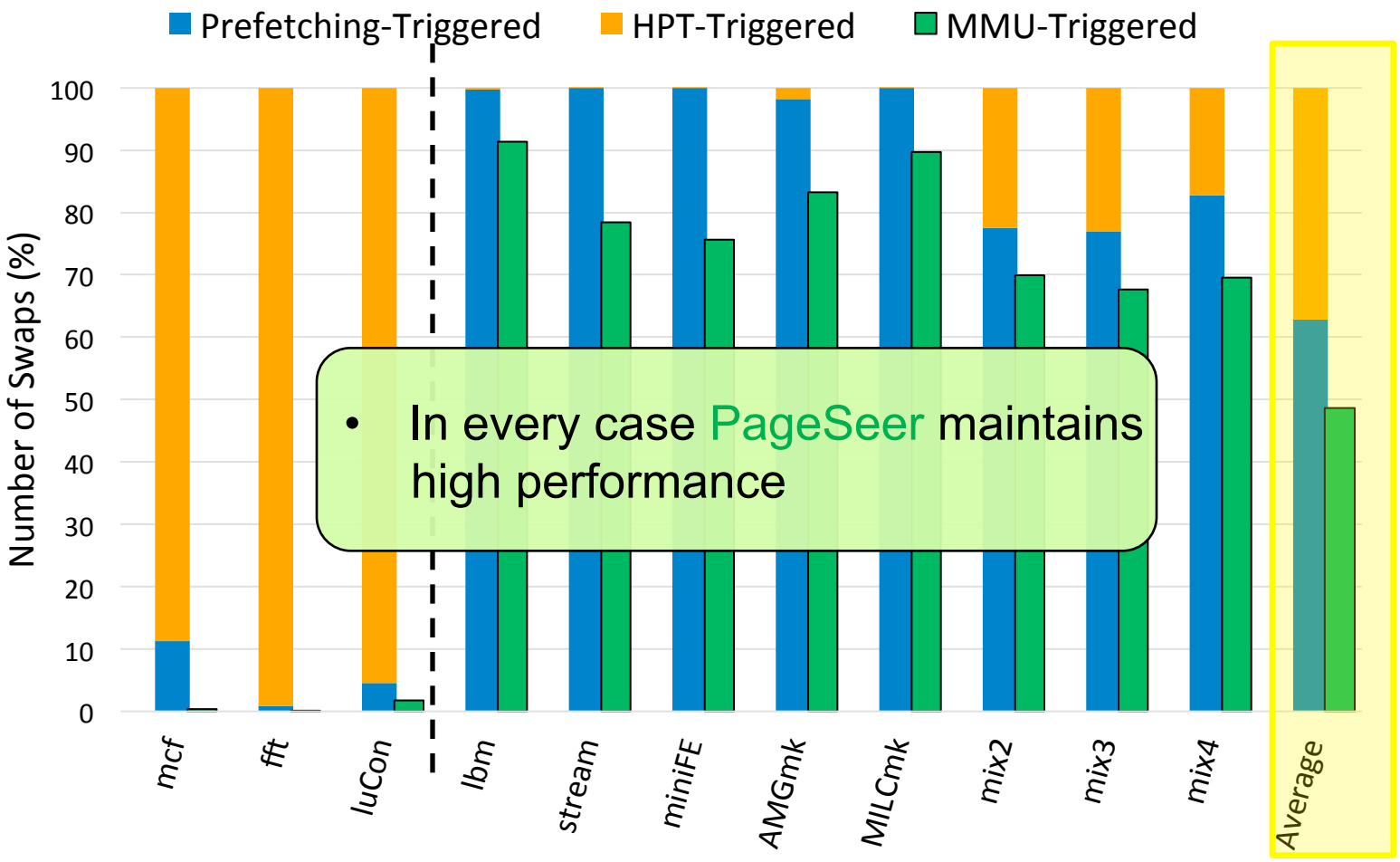
Swap Characterization



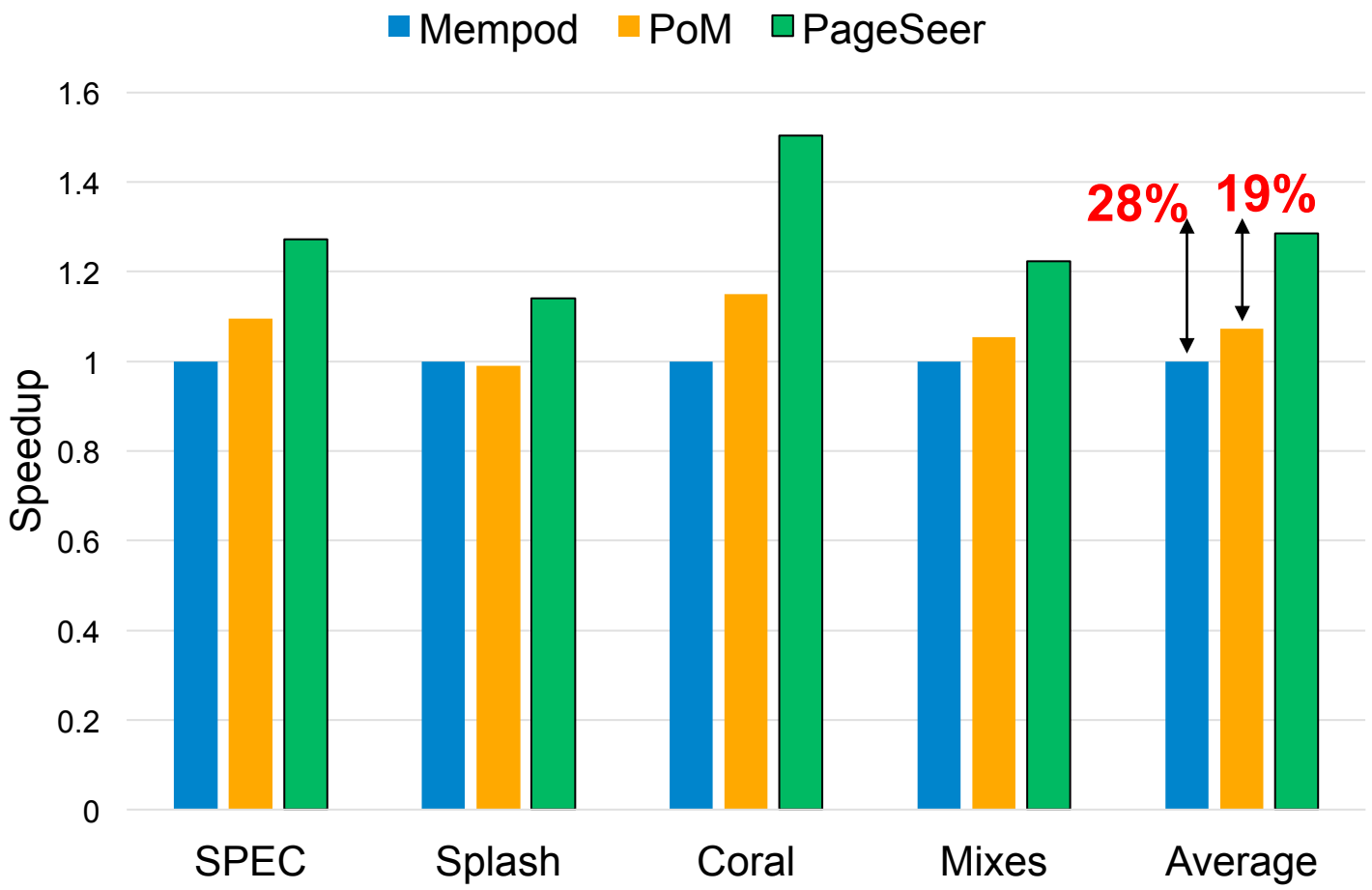
Swap Characterization



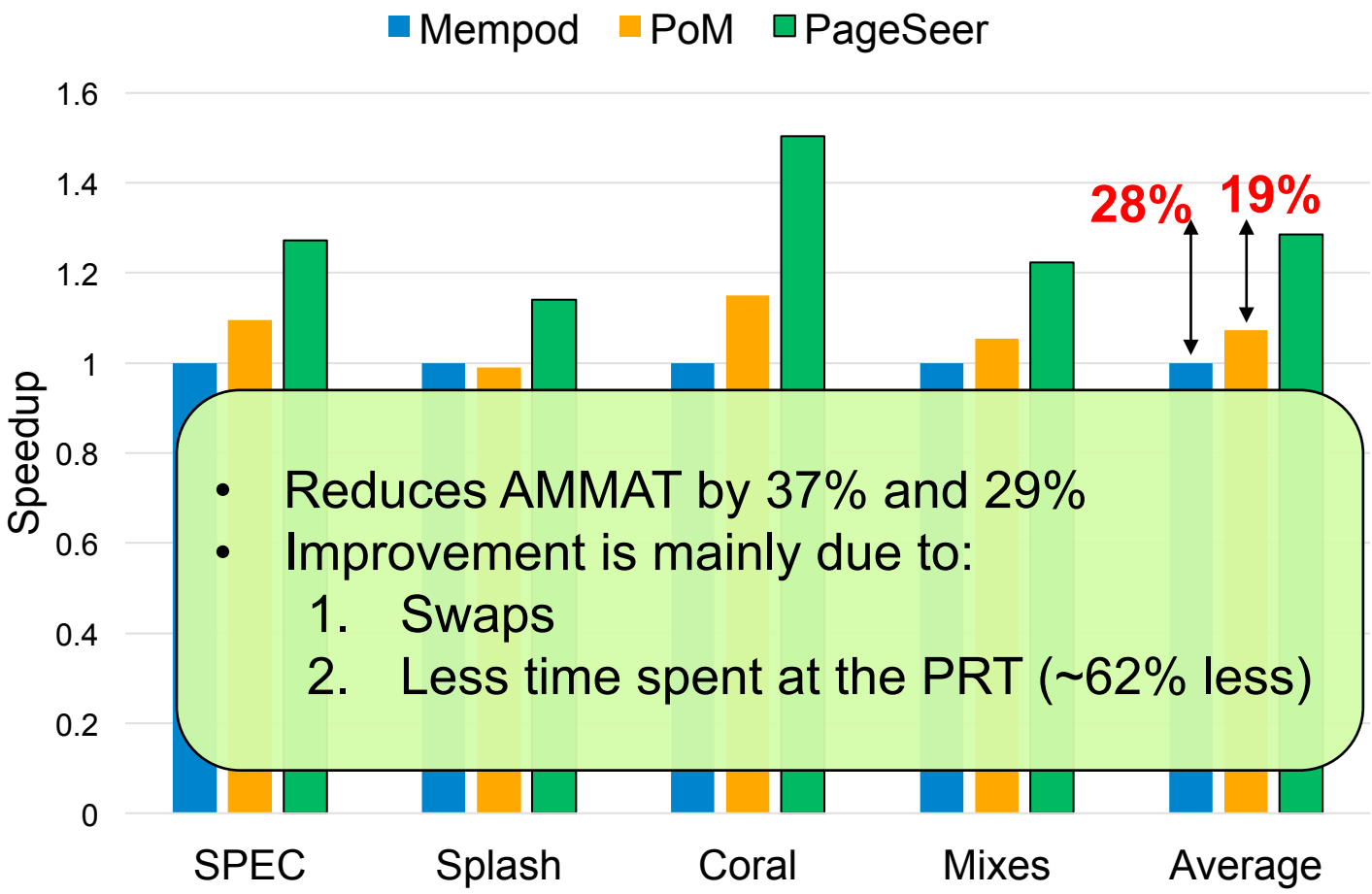
Swap Characterization



Performance Speedup



Performance Speedup



- Reduces AMMAT by 37% and 29%
- Improvement is mainly due to:
  1. Swaps
  2. Less time spent at the PRT (~62% less)

Also in the paper . . .

- More details on **PageSeer**
  - Operation
  - Power and area numbers for the HMC hardware structures
- Swap accuracy, page walk requests and swap characterization
- How to utilize the swap buffers during the swap process

Conclusion

- **PageSeer** uses page walks and correlation prefetching to trigger page swaps accurately and promptly
- MMU hints help to identify future memory accesses; start swaps and prepare the HMC hardware structures
- **PageSeer** improves IPC by 19% and reduces AMMAT by 29% over prior state-of-the-art schemes

PageSeer: Using Page Walks to Trigger Page Swaps in Hybrid Memory Systems

Apostolos Kokolis, Dimitrios Skarlatos and Josep Torrellas
University of Illinois at Urbana-Champaign
http://iacoma.cs.uiuc.edu

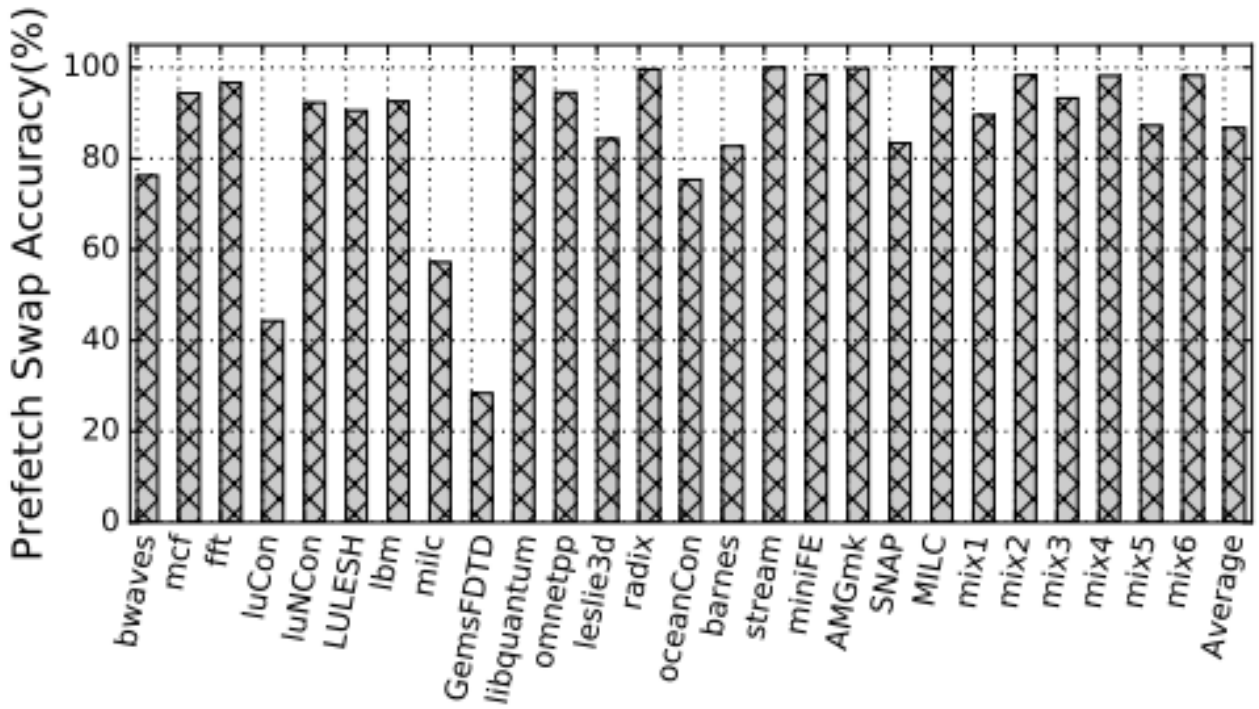
THANK YOU!!

HPCA 2019
Session 8A: MEMORY

i-acoma group

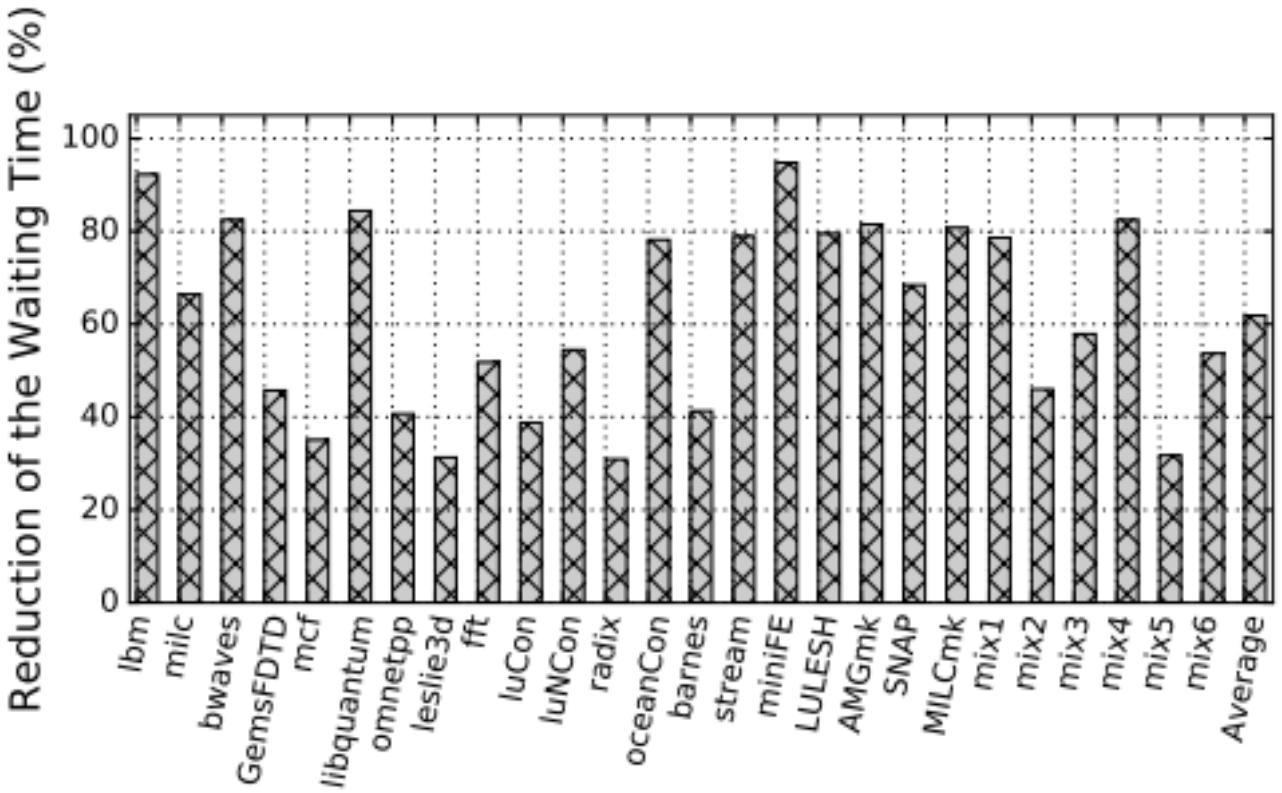
ILLINOIS

BACKUP SLIDES



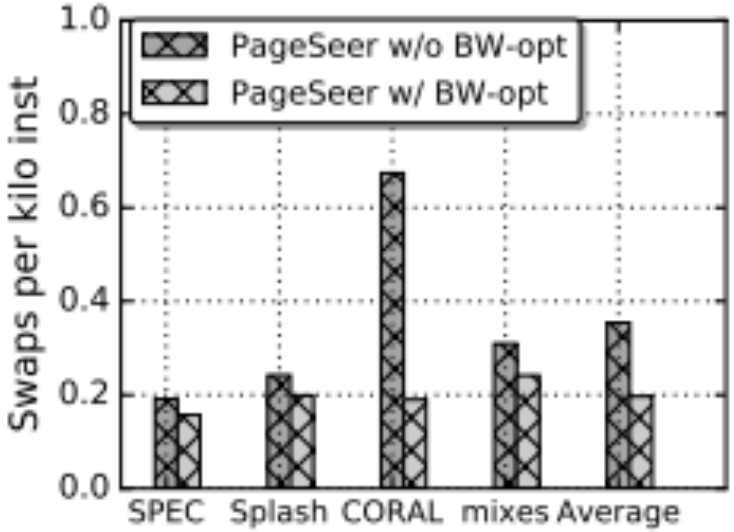
Average: 87% accuracy

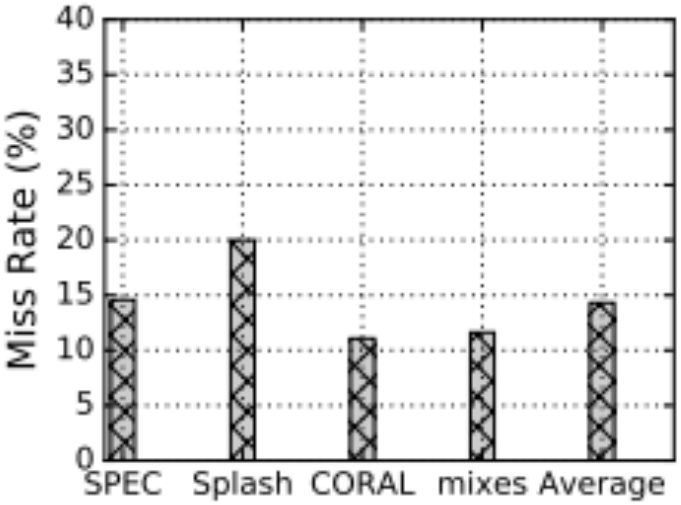
BACKUP SLIDES



Average: 62%
PRT reduction

BACKUP SLIDES





0.35 to 0.19 swaps per KInst

15% PTE cache misses