Machine Intelligent and Timely Data Management for Hybrid Memory Systems

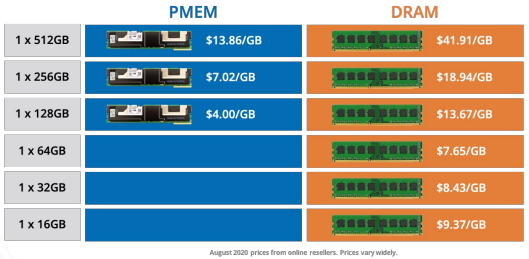
Thaleia Dimitra Doudali

Advisor: Ada Gavrilovska

@ SC 2020 Doctoral Showcase

The Era of Massive Hybrid Memory Systems

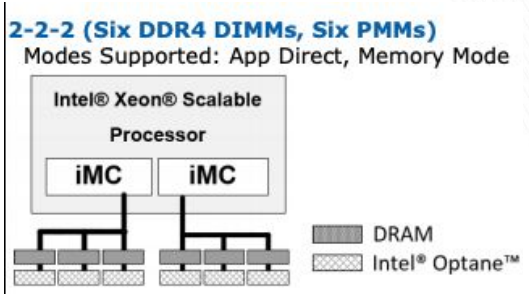
Data Explosion



Source:
https://www.memverge.com/more-memory-less-cost

TBs of Persistent Memory at ⅓ of the DRAM cost.

New Memory Technologies



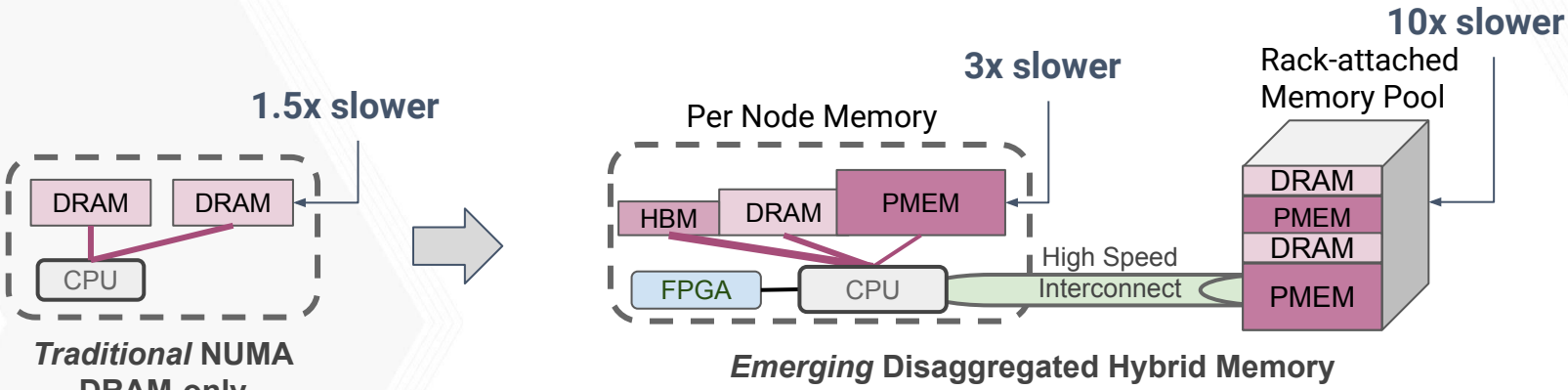
Source: Intel(R) Optane(TM) DC Persistent Memory Quick Start Guide

Use of PMEM alongside DRAM.

Hybrid Memory Systems

Georgia Tech
CREATING THE NEXT

Challenges in Hybrid Memory Systems



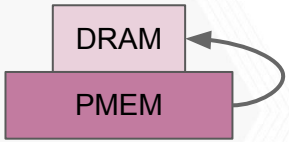
- Bigger difference in memory access speeds and capacities.
- Significant application performance slowdown from DRAM-only NUMA systems, due to the ineffectiveness of traditional data balancing solutions.
- Need to revisit data management approaches.

Data Management Approaches

How are hybrid memories configured?

Managed by the Hardware

Cache Organization.



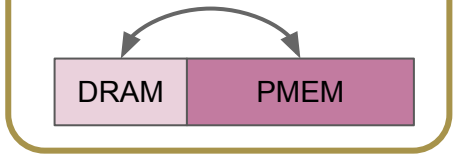
Pros:
No software overheads.

Cons:
No aggregate bandwidth.

Approach:
Data Prefetching from NVM to DRAM.

Managed by the Operating System

Flat Organization.



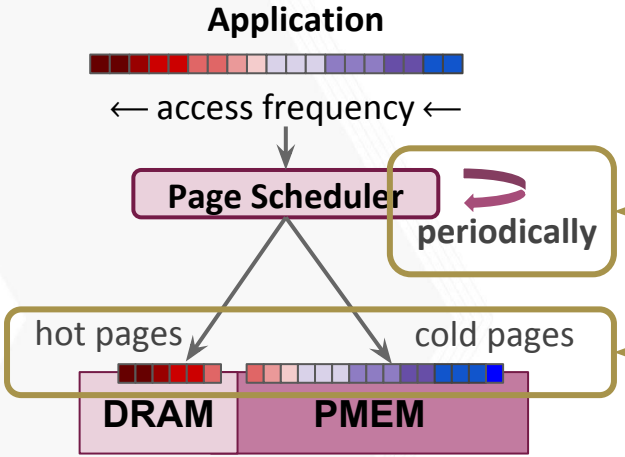
thesis target

Pros:
Explicit management. Bandwidth efficiency.

Cons:
Software overheads.

Approach:
Periodic data movements between NVM and DRAM.

Thesis Statement



Flat Hybrid Memory Organization

Existing approaches dynamically:

- Monitor data access behavior, keep a history.
- Identify frequently accessed pages.
- Page scheduler moves pages between DRAM and PMEM.

When to move the data?

Which data to move?

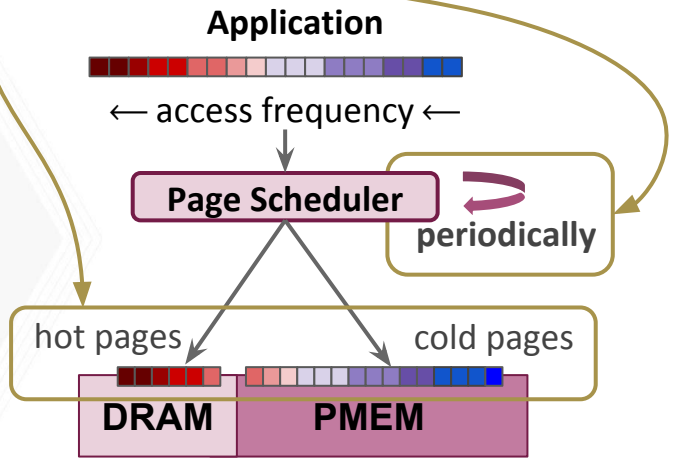
Challenges

The contributions of this thesis combine *machine intelligent* selection of data movements with *fine-tuned* data movement time intervals, to bridge the performance gap left by existing approaches, while allowing for practical system-level integration.

Thesis Highlights

Thesis: Machine Intelligent and Timely Data Management for Hybrid Memory Systems.

Kleio: a Hybrid Memory Page Scheduler with Machine Intelligence.
@ HPDC '19.
Best Paper Award Finalist.



Flat Hybrid Memory Organization

Cori: Dancing to the Right Beat of Periodic Data Movements over Hybrid Memory Systems.
[Ongoing Work]

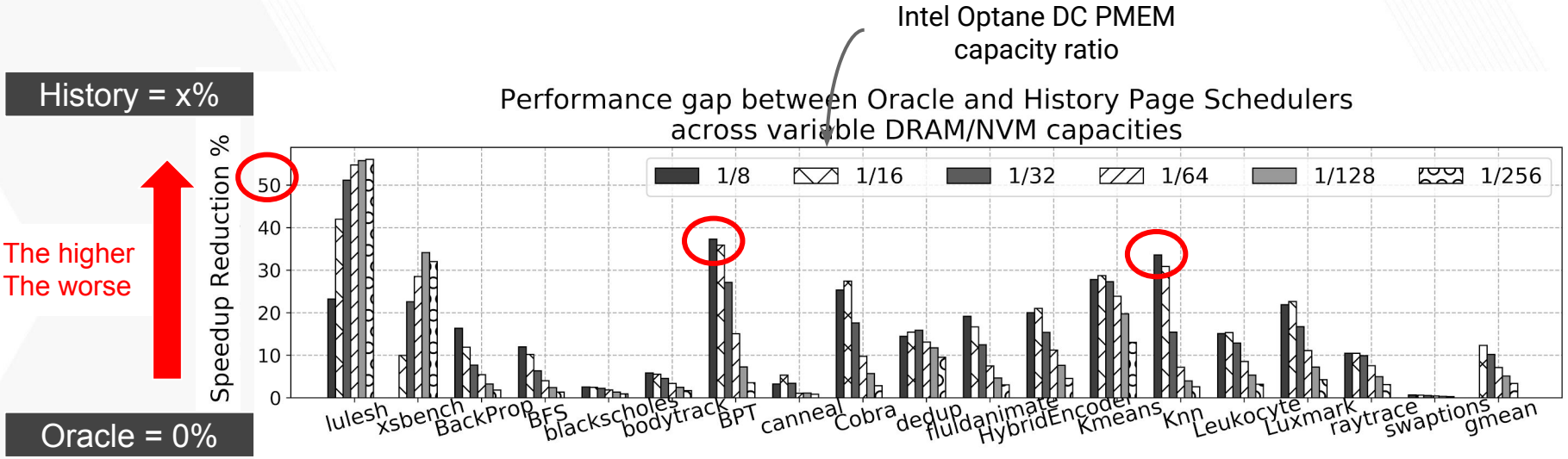
The Case for Optimizing the Frequency of Periodic Data Movements over Hybrid Memory Systems. @ MEMSYS '20

Prior Work on Optimizing Static Data Tiering across DRAM and PMEM:

CoMerge: *Toward Efficient Data Placement in Shared Heterogeneous Memory Systems.* @ MEMSYS '17

Mnemo: *Boosting Memory Cost Efficiency in Hybrid Memory Systems.* @HPBDC workshop of IPDPS '19

Thesis Highlight 1
Which data to move?



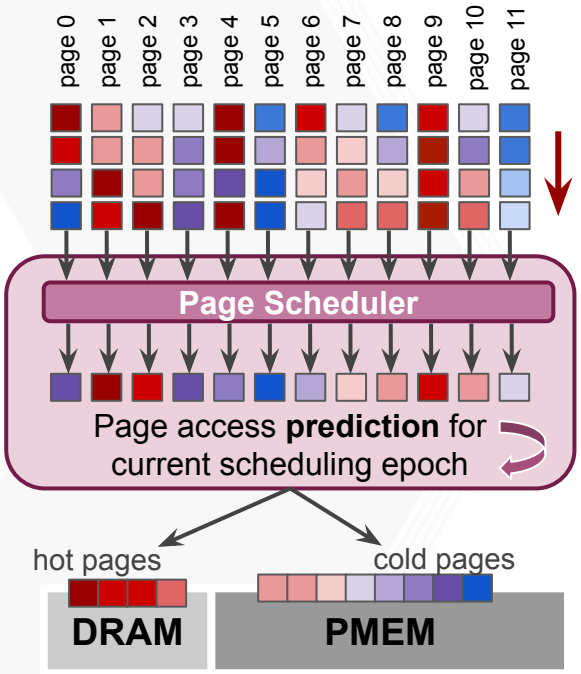
Added Performance Reduction due to Page Scheduling

Simple history-based *page scheduling methods end up causing significant* *additional* *performance degradation in applications executing over hybrid memories.*

We need something more clever to close the gap!

Georgia Tech
CREATING THE NEXT

Solution Design



Past Page Access Information

How can we use **Machine Intelligence** in order to combine *past* access information into an *accurate prediction* of *future* behavior?

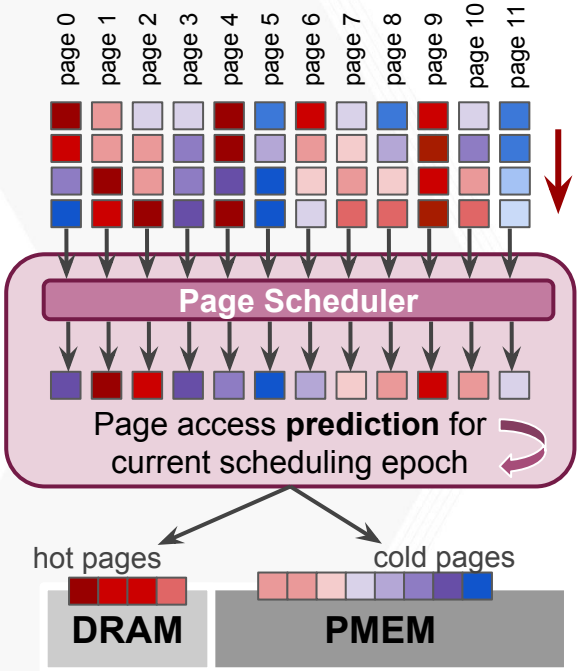
Design Questions:

1. Which Machine Intelligence (MI) method to use?
2. What input/output fits the page scheduling description?

Evaluation Questions:

1. How much can it reduce the performance gap?
   How accurate are the predictions?
2. Is it practical to integrate into future systems?

Solution Overview



Access counts across previous scheduling epochs

Kleio* is a machine intelligent page scheduler for hybrid memory systems.

*According to the ancient Greek mythology, Kleio was the muse of history, daughter of Mnemosyne, goddess of memory.

Design Answers:

1. Uses Recurrent Neural Networks (RNNs).

2. Predicts per page access counts.

Evaluation Answers:

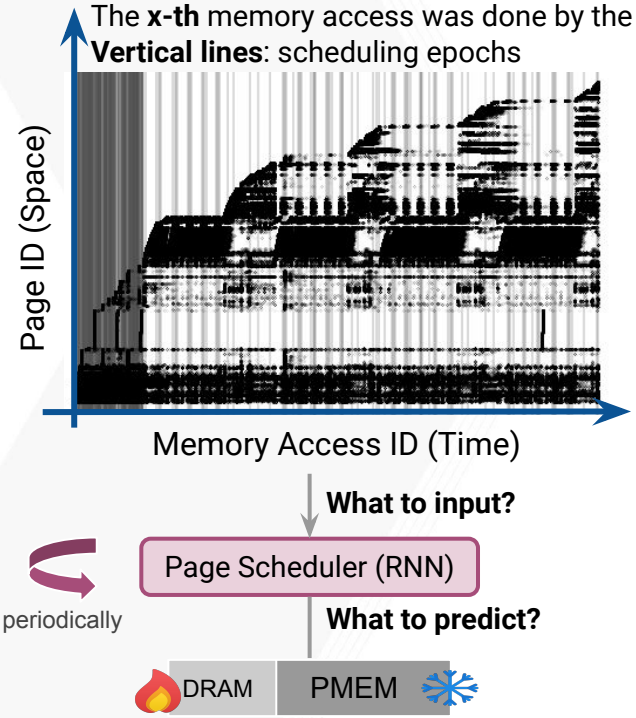
1. Closes the performance gap by **80%.**

2. It is practical since it identifies the page subset that needs ML-based management.

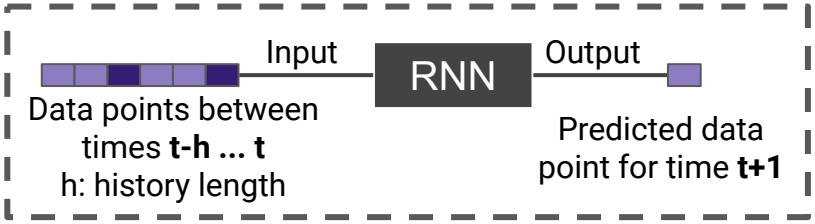
Georgia Tech
CREATING THE NEXT

Solution Design

Suitable RNN Input Format



We'll treat RNN as a black box throughout this presentation.



RNNs as used in **Prefetching:** Which page will be accessed next?

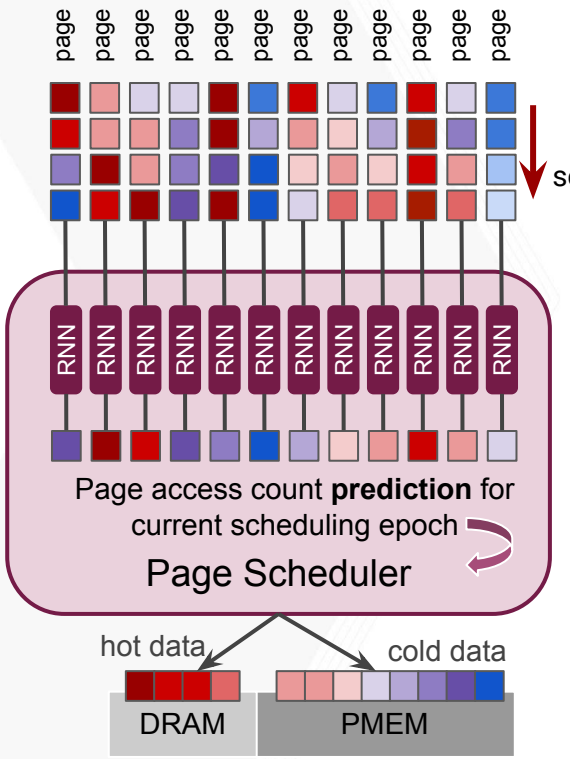
❌ Not suitable due to high training overheads and low accuracy levels in order to make a decision for all pages.

Per Page Prediction: How many times a page was accessed.

✔ Suitable to deliver low training times and adequate prediction accuracy.

Georgia Tech
CREATING THE NEXT

Solution Design
Per Page Prediction



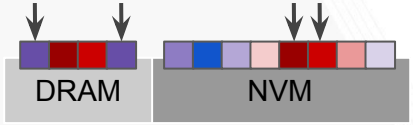
Not really scalable..
HPC and Big Data applications can have millions of pages!

Approach:

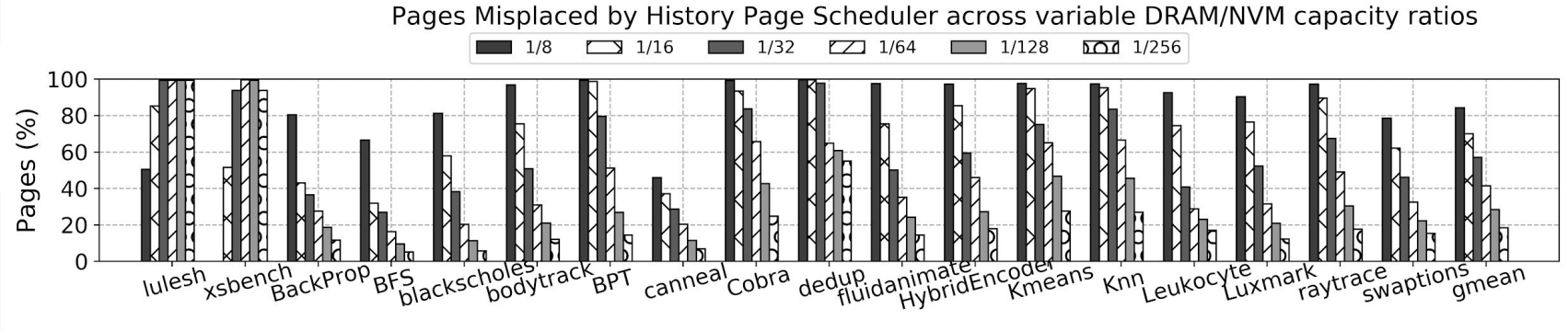
- **Focus learning** on the **subset of pages** whose timely DRAM allocation brings significant performance improvement.
- Incorporate **lightweight current state-of-the art** solutions without machine intelligence for the remaining pages.

Solution Design

Not all pages need Machine Intelligence



A page is **misplaced** when at the start of a scheduling epoch it is not allocated in DRAM, even though it was hot, because the scheduler mispredicted its high access frequency.

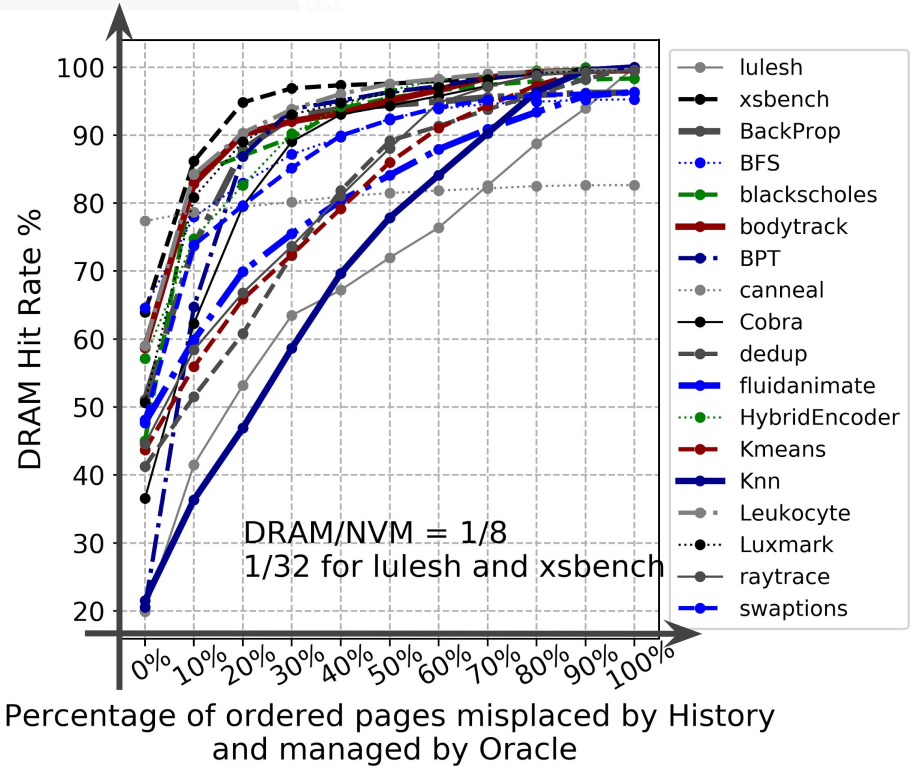


The History page scheduler reduces the number of pages we need to manage more cleverly.
Still, the number be significant especially for the Intel Optane DC PMEM 1/16 capacity ratio.

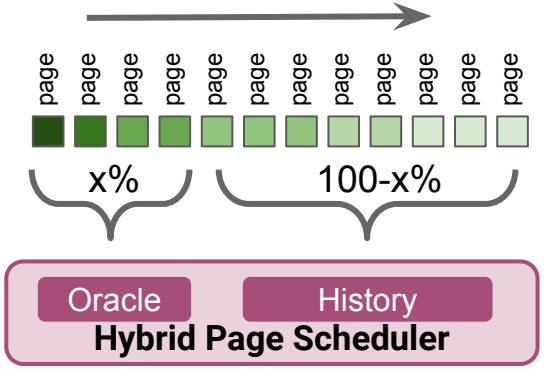
Can we further reduce the number of pages that need more intelligent management?

Solution Design

Prioritize for Machine Learning highly accessed and misplaced pages



Pages misplaced by History in descending order of:
benefit = # accesses x # misplacements



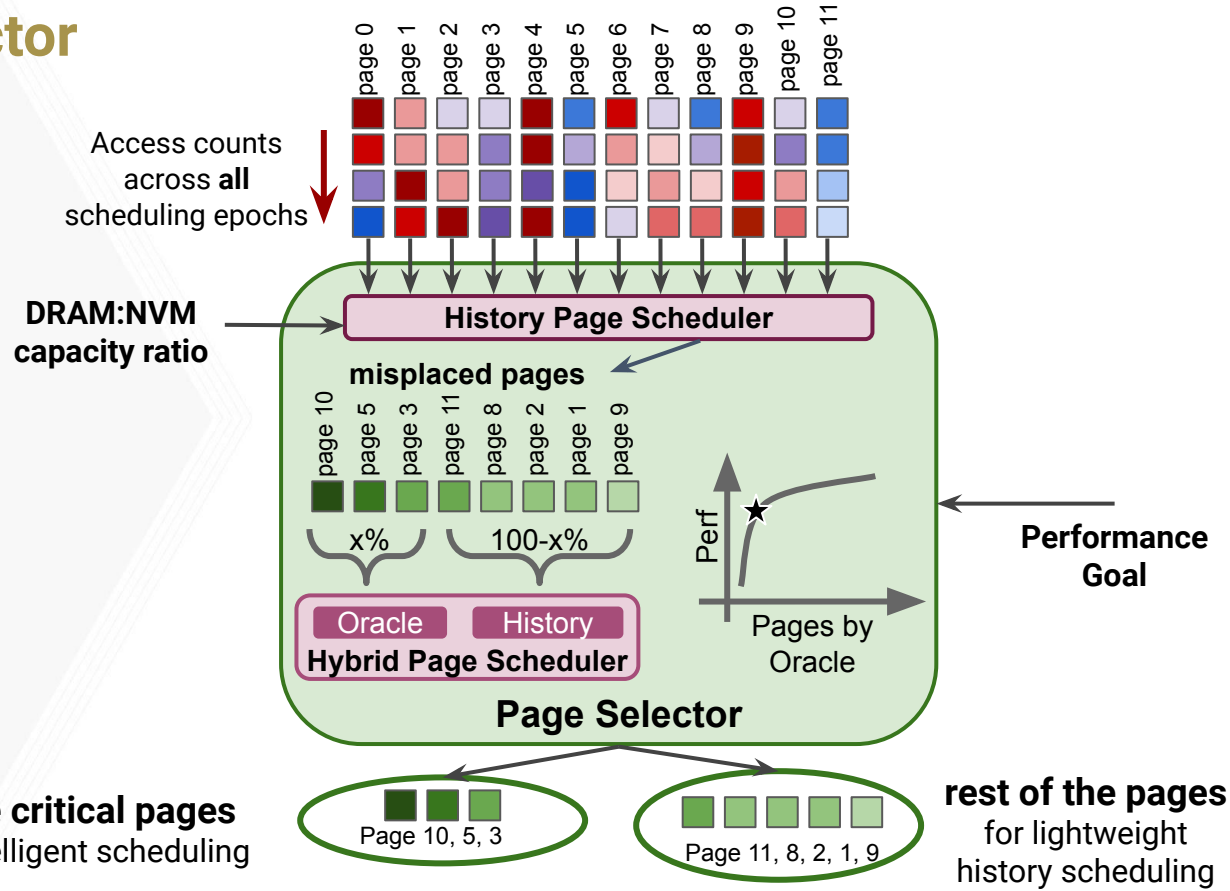
Question: How does performance increase, the more pages we manage intelligently via Oracle?

Answer: Non linearly. Only a small page subset with high benefit needs intelligent management.

Georgia Tech
CREATING THE NEXT

Solution Design

Page Selector



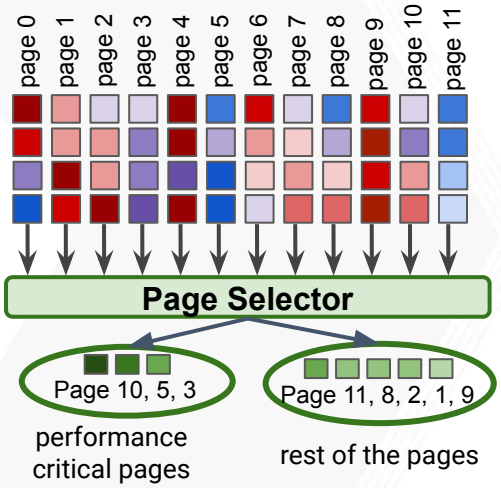
performance critical pages
for machine intelligent scheduling

rest of the pages
for lightweight history scheduling

Georgia Tech
CREATING THE NEXT

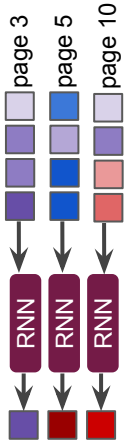
Solution Overview

Step 1: Page Selection



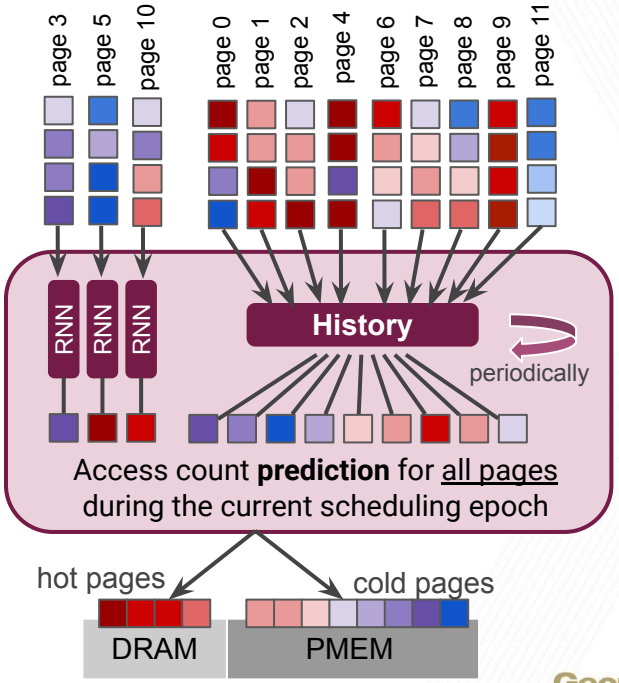
The Page Selector is run only **once**, to find the pages that require machine learning.

Step 2: RNN training



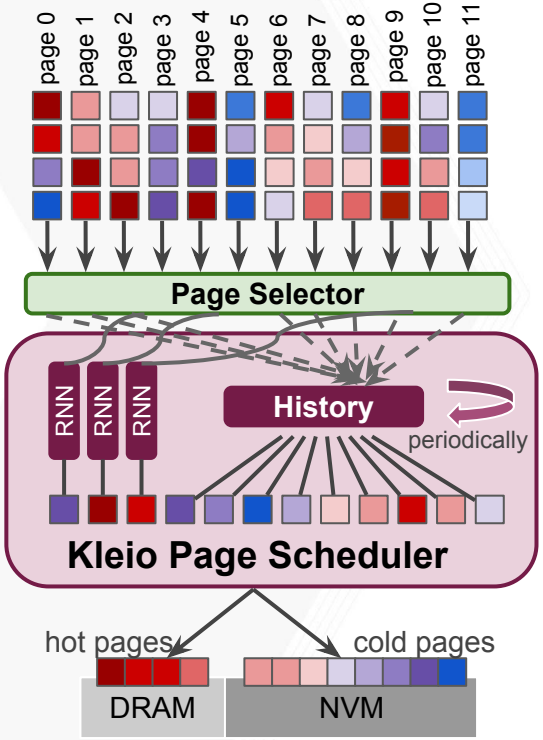
Trained models are saved.

Step 3: RNN inference during page scheduling



Solution Overview

With some of the Implementation Details



Applications: CORAL, PARSEC, Rodinia
Number of pages: 8K - 800K
Number of Scheduling Epochs: up to 856 (x 1 sec)

Memory Access Trace Collection:
IBS sampling and unsampled traces of Last Level Cache Misses
(time, virtual address, physical address, cpu core, thread id, load/store, hit/miss)

RNN Implementation:
Long Short Term Memory (LSTM) Networks, Keras API, Tensorflow Backend
(more on the paper!)

Hybrid Memory System:
Trace-based analysis for DRAM hit rates.
Analytical model to extrapolate runtime based on access distribution across DRAM and NVM assuming zero cost migrations.

Evaluation

Kleio closes on average 80% of the performance gap

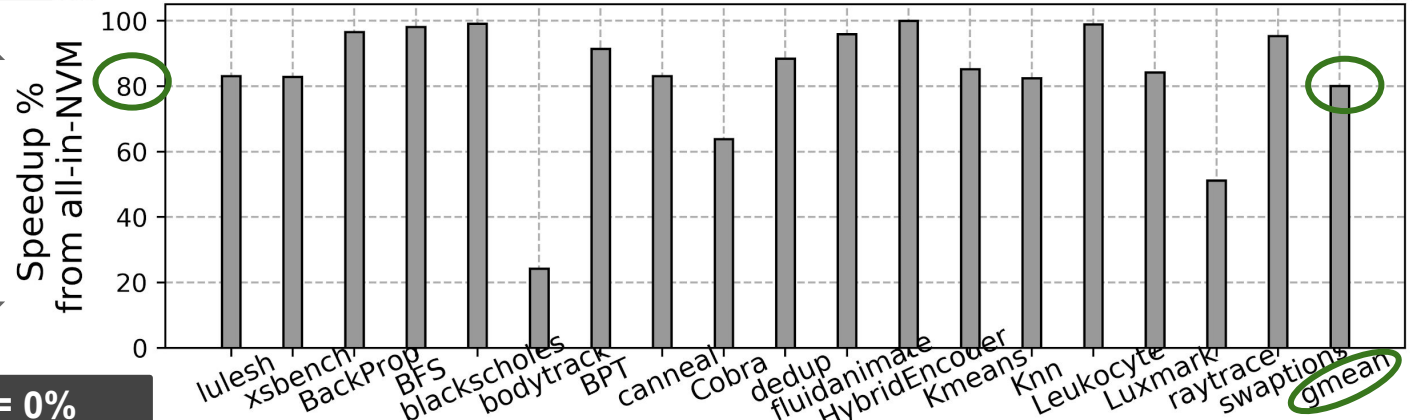
Oracle selected pages
History rest of the pages
Optimal = 100%

RNN selected pages
History rest of the pages
Kleio

Baseline = 0%
History all pages

The higher
The better

More than **95%** for **half** of the applications!



For fixed DRAM:NVM capacity.
For 100 selected pages.

Georgia Tech
CREATING THE NEXT

Evaluation
Practical Considerations

Resource Utilization **per RNN model** on general purpose CPU:

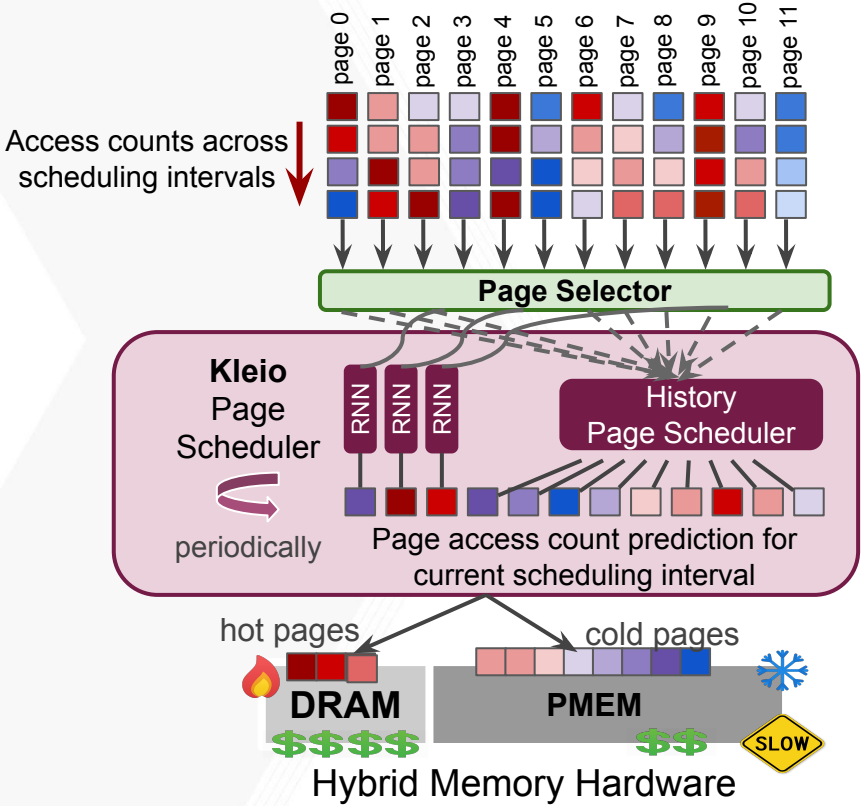
Training 2 hours Tens GBs of Memory | **Inference** 3-4 sec 0.5 MB of Storage

- ✔️ **Duration** can be further reduced by multiple orders of magnitude with anticipated ML accelerators.
- ✔️ Large **memory** footprint can be accommodated by hybrid memory systems!
- ✔️ Kleio's Page Selector already drastically reduces the **problem space**.
- ✔️ RNNs can also be trained in an **online** manner.

➡️ **There is great potential for Kleio to be adapted in an online practical system-level solution.**

Summary

Paper Contributions



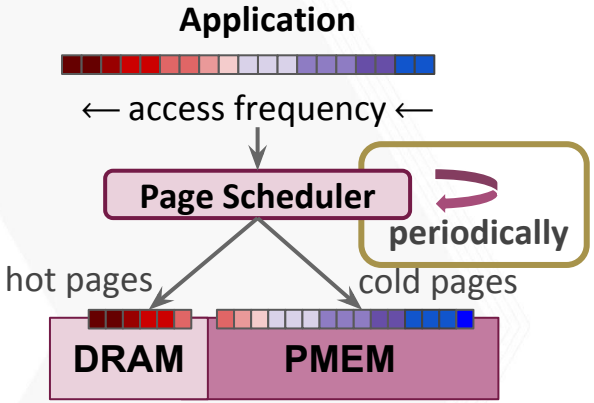
Kleio *is a machine intelligent page scheduler for hybrid memory systems.*

✔ Bridges the existing **performance gap by 80%**.

✔ Cleverly identifies the **page subset** whose timely allocation in DRAM will boost performance via machine intelligent placement.

✔ Lays the ground for **practical integration** of machine intelligent memory management solutions in future systems.

Thesis Highlight 2
When to move the data?



Flat Hybrid Memory Organization

Existing Approaches:

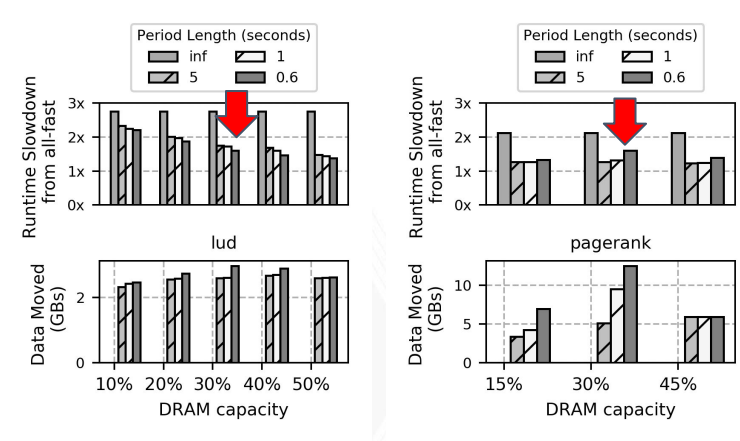
- According to execution phases.
  For example, MPI phases or task-based execution.

Focus of this work.

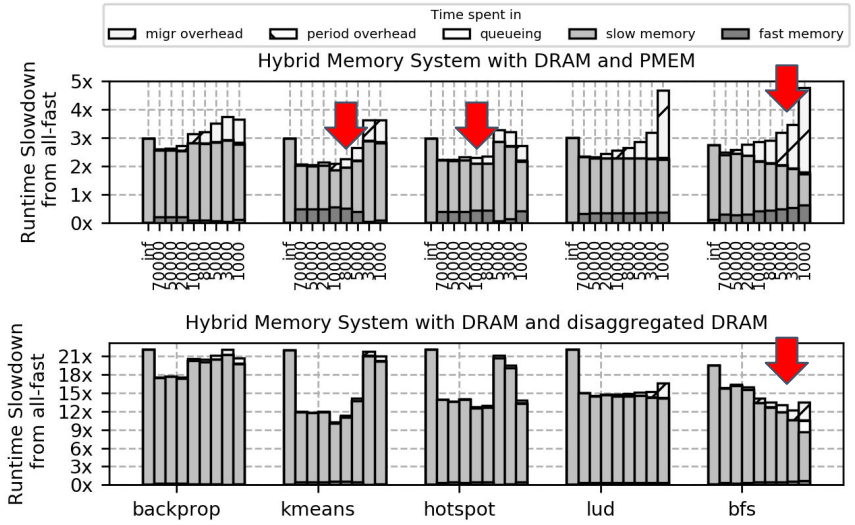
- In fixed time intervals, i.e., *periods*.
  Empirically set across majority of application-agnostic system-level solutions.

Problem Statement: What should be the frequency of data movements, so as to boost application performance in return for minimal monitoring and migration costs? What application-level properties hint towards a more sophisticated, rather than empirical, choice?

Why is it important?



Optane DC PMEM experiments



Simulation experiments

Observations:

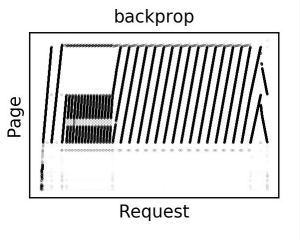
- Tuning the frequency leads to performance improvements (70% for PMEM, 5x for disaggregated on average)
- A certain frequency may not work across different applications or different platforms.
  - Same platform: different applications may benefit most from completely different frequencies.
  - Same application: different platforms may better offset any data movement costs.

It is important to choose the right frequency, but it is not intuitive how to choose one.

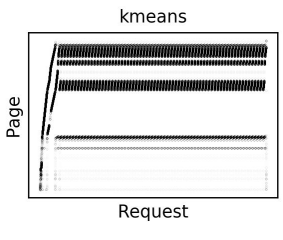
Georgia Tech
CREATING THE NEXT

Data Access Behavior

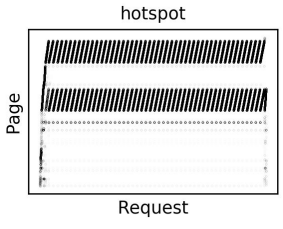
Memory Access Patterns



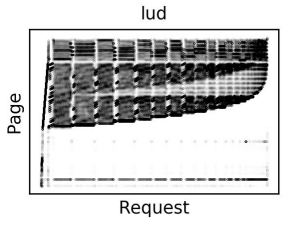
sequential strides



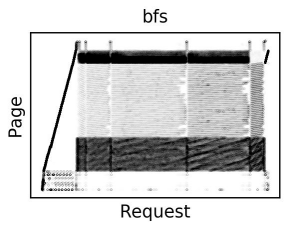
sequential strides



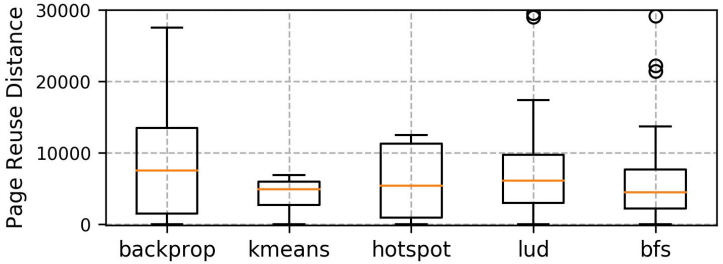
sequential strides



sequential, irregular shrinking work set



irregular breadth first graph traversal



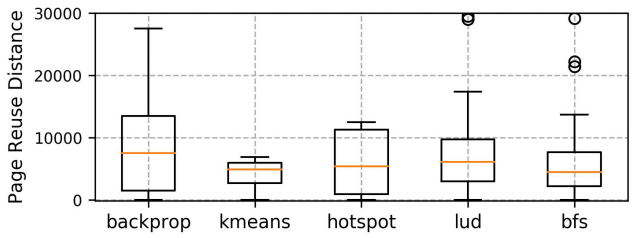
Page Reuse Distance = Number of memory accesses to other pages, between two consecutive accesses to that particular page.

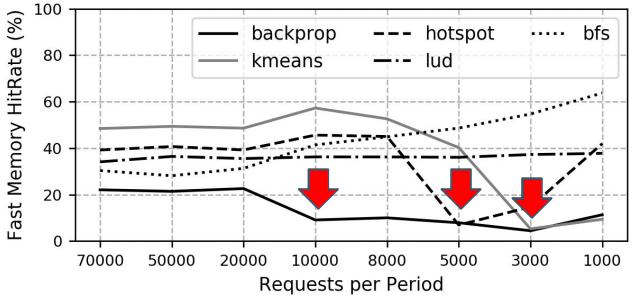
Irregular access behavior: outliers.
Sequential strides: no outliers.

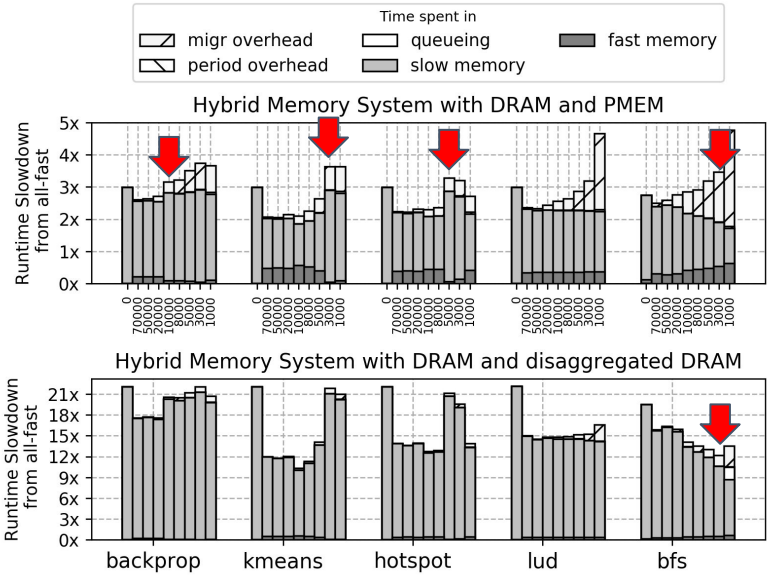
Data reuse indirectly shows data access behavior, with no prior information.

Georgia Tech
CREATING THE NEXT

Data Reuse Distance vs. Application Performance







Observation: When period length < median page reuse distance, the effectiveness of the page scheduler suffers.

Takeaway: Insights regarding data reuse lead to an informed selection of data movement frequencies.
The final choice of frequency highly depends on the effectiveness of the page scheduler and the way the platform can hide the data movement costs under a higher fast memory hit rate, depending on its memory access speeds.

Georgia Tech
CREATING THE NEXT

Ongoing Work

After establishing the importance of data reuse insights and the parameters which affect application performance on hybrid memory systems, we are building..

Cori, a profiler that synthesizes:

- data reuse distance
- data access patterns
- page scheduler efficiency
- platform configuration

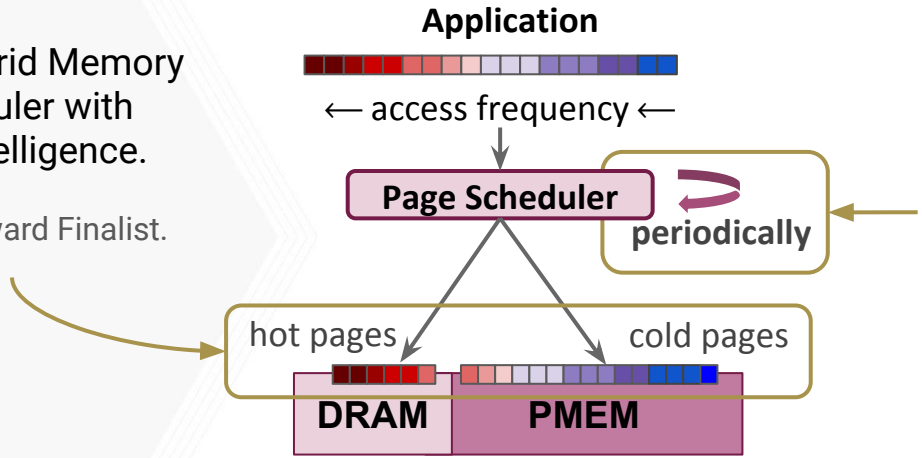
into the selection of the data movement frequency, that will further boost application performance, irrespective of the effectiveness of the data movement selection itself.

… Stay tuned!

Thesis Highlights Summary

Thesis: Machine Intelligent and Timely Data Management for Hybrid Memory Systems.

Kleio: a Hybrid Memory Page Scheduler with Machine Intelligence.
@ HPDC '19.
Best Paper Award Finalist.



Flat Hybrid Memory Organization

Cori: Dancing to the Right Beat of Periodic Data Movements over Hybrid Memory Systems.
[Ongoing Work]

The Case for Optimizing the Frequency of Periodic Data Movements over Hybrid Memory Systems. @ MEMSYS '20

Prior Work on Optimizing Static Data Tiering across DRAM and PMEM:

CoMerge: *Toward Efficient Data Placement in Shared Heterogeneous Memory Systems.* @ MEMSYS '17

Mnemo: *Boosting Memory Cost Efficiency in Hybrid Memory Systems.* @HPBDC workshop of IPDPS '19