Data-centric Resource Management for Complex Memory Fabrics

Ada Gavrilovska

Thaleia Dimitra Doudali, Pradeep Fernando

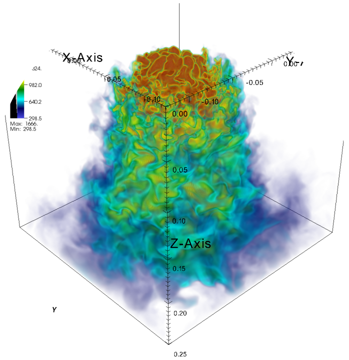
Tony Mason, Ranjan Venkatesh Sarpangala, Daniel Zahka

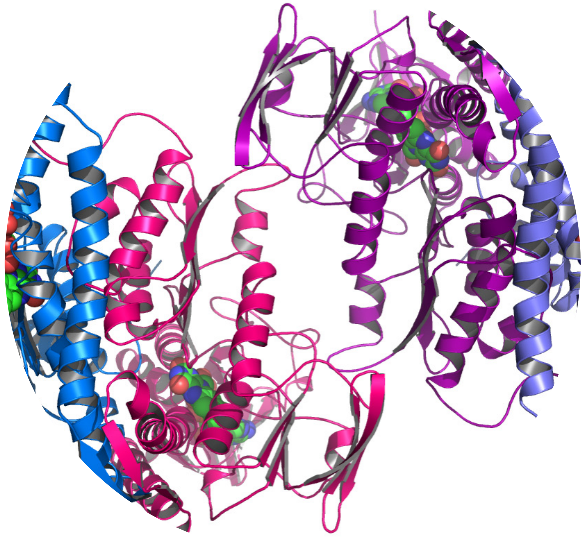
Georgia Tech

The Era of Data

The Era of Data

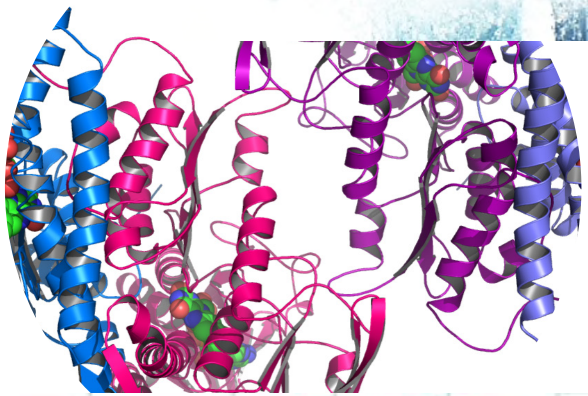
- BIG DATA
- *Fast data*
- HETEROGENEOUS DATA
- *Noisy data*
- metadata





The Era of Data and Extreme Heterogeneity

- BIG DATA
- *Fast data*
- HETEROGENEOUS DATA
- *Noisy data*
- metadata

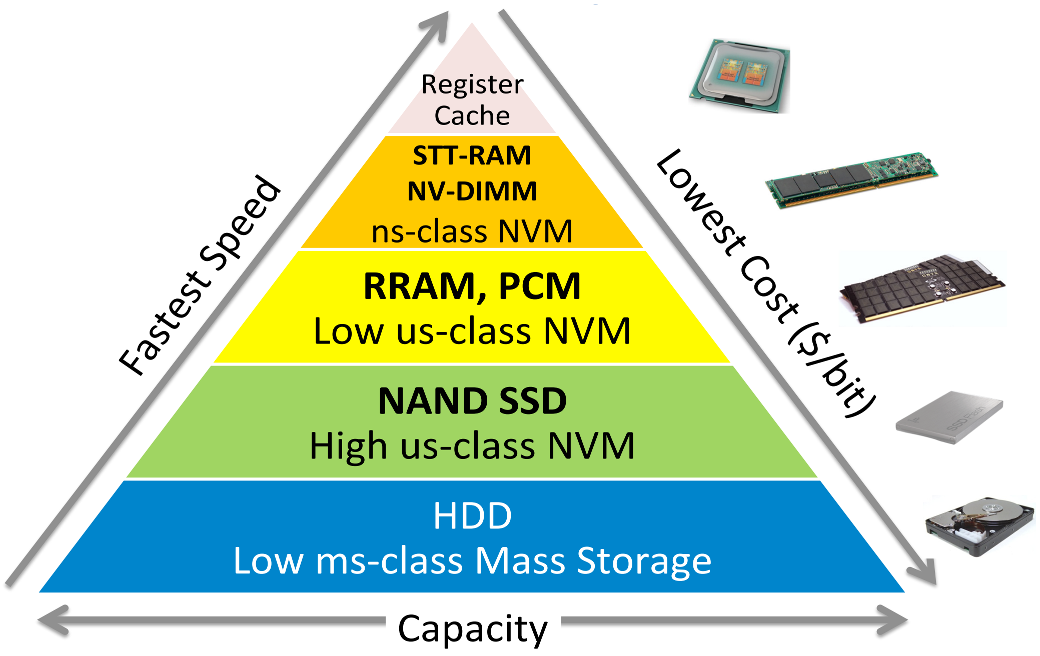


Heterogeneous Memories

- Much buzz over the last decades
- Potential for
  - ***memory capacity scaling***
  - ***memory access performance***
  - ***fast persistence***

- e.g., Intel Optane DC Persistent Memory

	HBM	DRAM	PCM/3D-XPoint DIMMs	3D-XPoint (NVMe)	Flash/NAND
capacity	0.1x	1x	4-10x	4-10x	10x
read latency	1x	1x	2-3x	10x	10,000x
write lantecy	1x	1x	5x	10x	10,000x
bandwidth	10x	1x	0.1x	PCIe 3.0 now	



Source: http://research.cs.wisc.edu/sonar/tutorial/01-technology.pdf

Our Contributions to Systems Software for Heterogeneous Memories

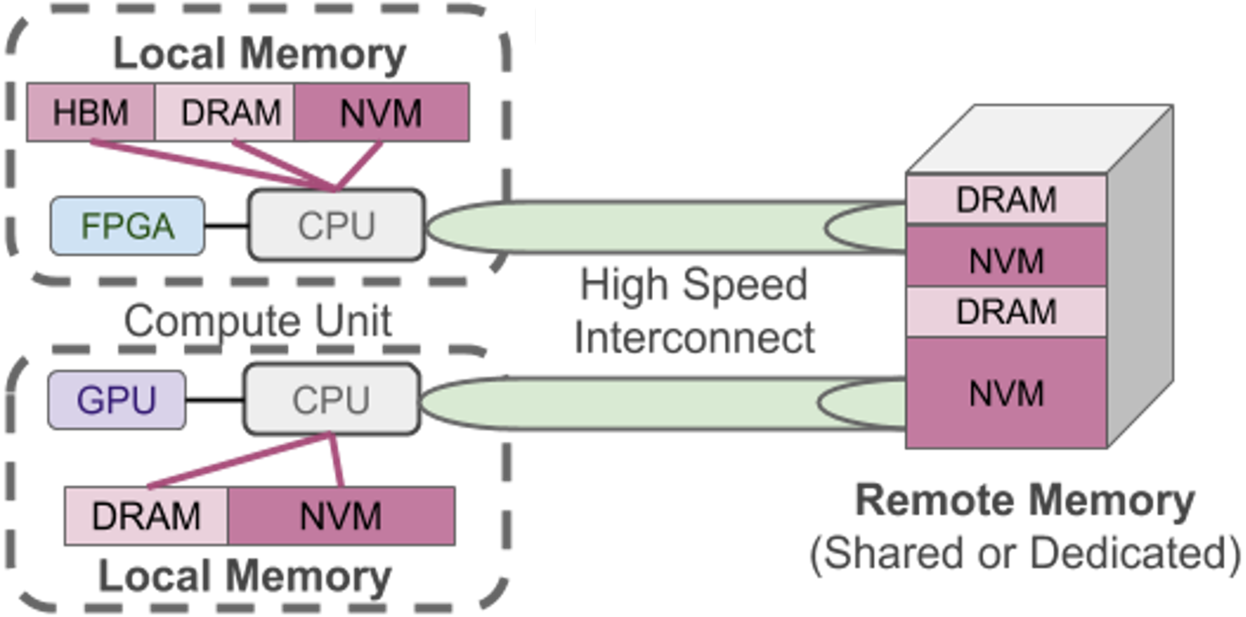
- pVM – OS support for persistent memory [EuroSys’16]
- HeteroOS – support for NVM in virtualized datacenters [ISCA’17]
- NoveLS
- Perform NVM [H
- Acceler
- Energy-
- NVM-specialized checkpoint/restart [IPDPS’13]
- NVM-specialized streaming I/O [HPDC’18]

Do we still need to work on heterogeneous memories?

ith Sudarsun Kannan,
ow at Rutgers University

Systems with Heterogeneous Memories

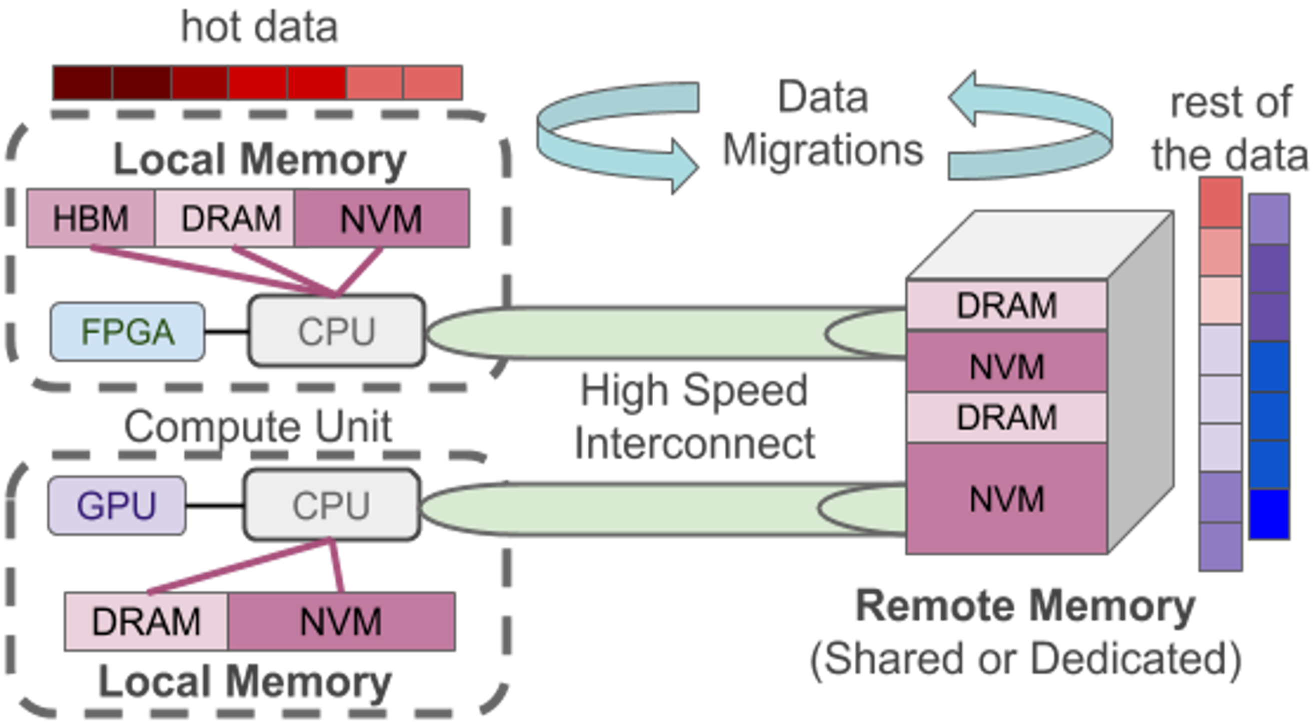
- Number and type of devices
- Performance, reliability, persistence, …
- Direct access mode or strict hierarchy
- Locality, on-node, to remote nodes…
- Sharing
- Coherence
- Affinity to accelerators
- In-memory/controller/data-path accelerators



Emerging Hybrid Memory Systems.

Systems with Heterogeneous Memories

- Number and type of devices
- Performance, reliability, persistence, …
- Direct access mode or strict hierarchy
- Locality, on-node, to remote nodes…
- Sharing
- Affinity to accelerators
- Coherence
- In-memory/controller/data-path accelerators
- Software stack: file system, OS version, memory mapped, …
- Page size
- Allocation policy, interleave, membind, localalloc,…
- Migration policy
- Management frequency
- CPU and cluster scheduler
- …

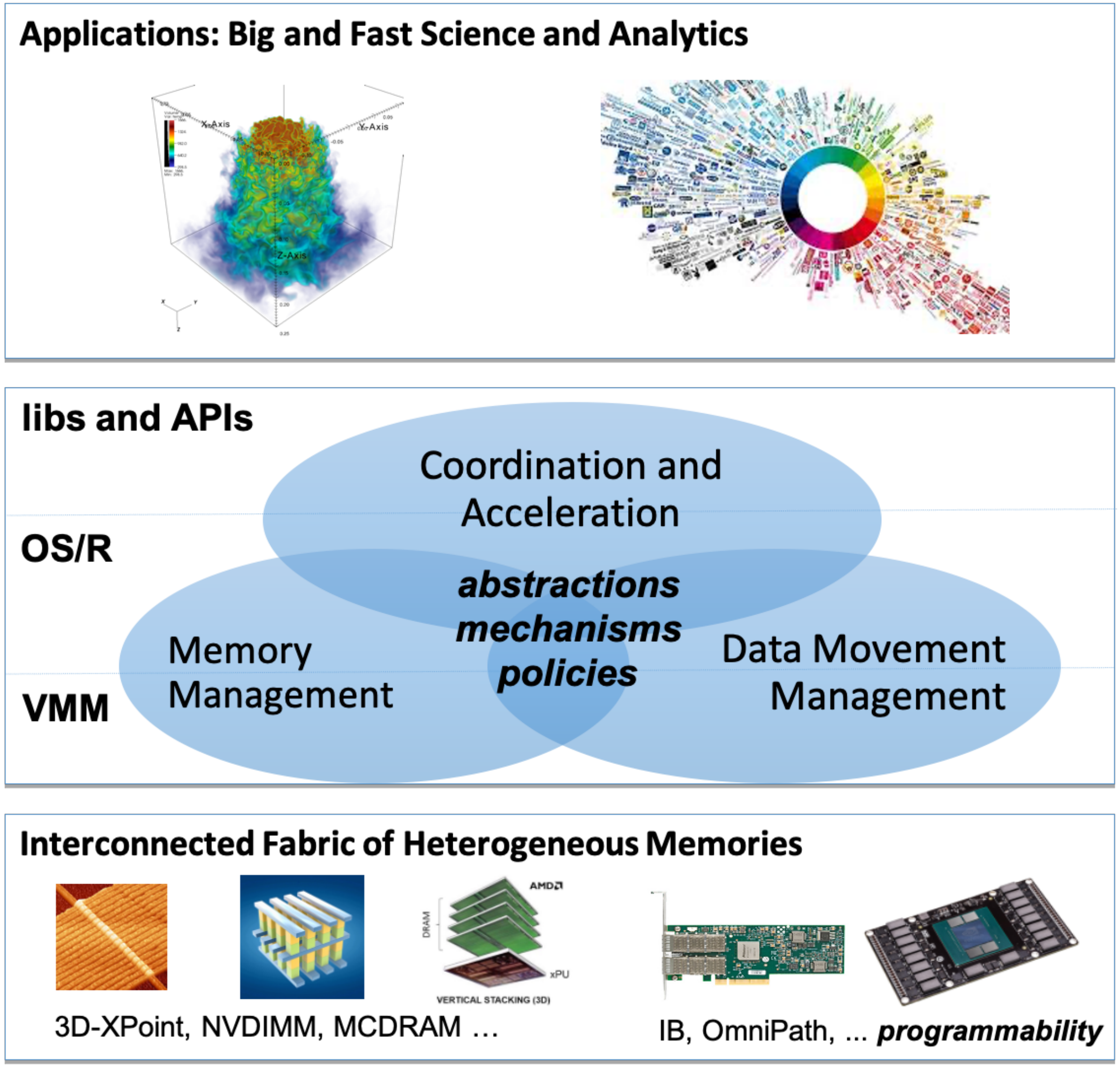


Emerging Hybrid Memory Systems.

Complex Systems with Heterogeneous Memory Fabrics

- Scale and heterogeneity across the software/hardware boundary of the memory subsystem
- => **Complex Memory Fabrics**

- Existing policies and heuristics not built for this
- => **Complex access and management policies and controls**



Applications: Big and Fast Science and Analytics

libs and APIs

OS/R

VMM

Coordination and Acceleration

Memory Management

Data Movement Management

abstractions
mechanisms
policies

Interconnected Fabric of Heterogeneous Memories

3D-XPoint, NVDIMM, MCDRAM …

IB, OmniPath, ... ***programmability***

Complexity == New Tradeoffs and Opportunities

Replace *heuristics* with

data driven models, tools and techniques

⇒ enable new intelligent, efficient and effective management of complex memory fabrics

⇒ maximize technology benefits

Complexity == New Tradeoffs and Opportunities

When does complex management pay off?

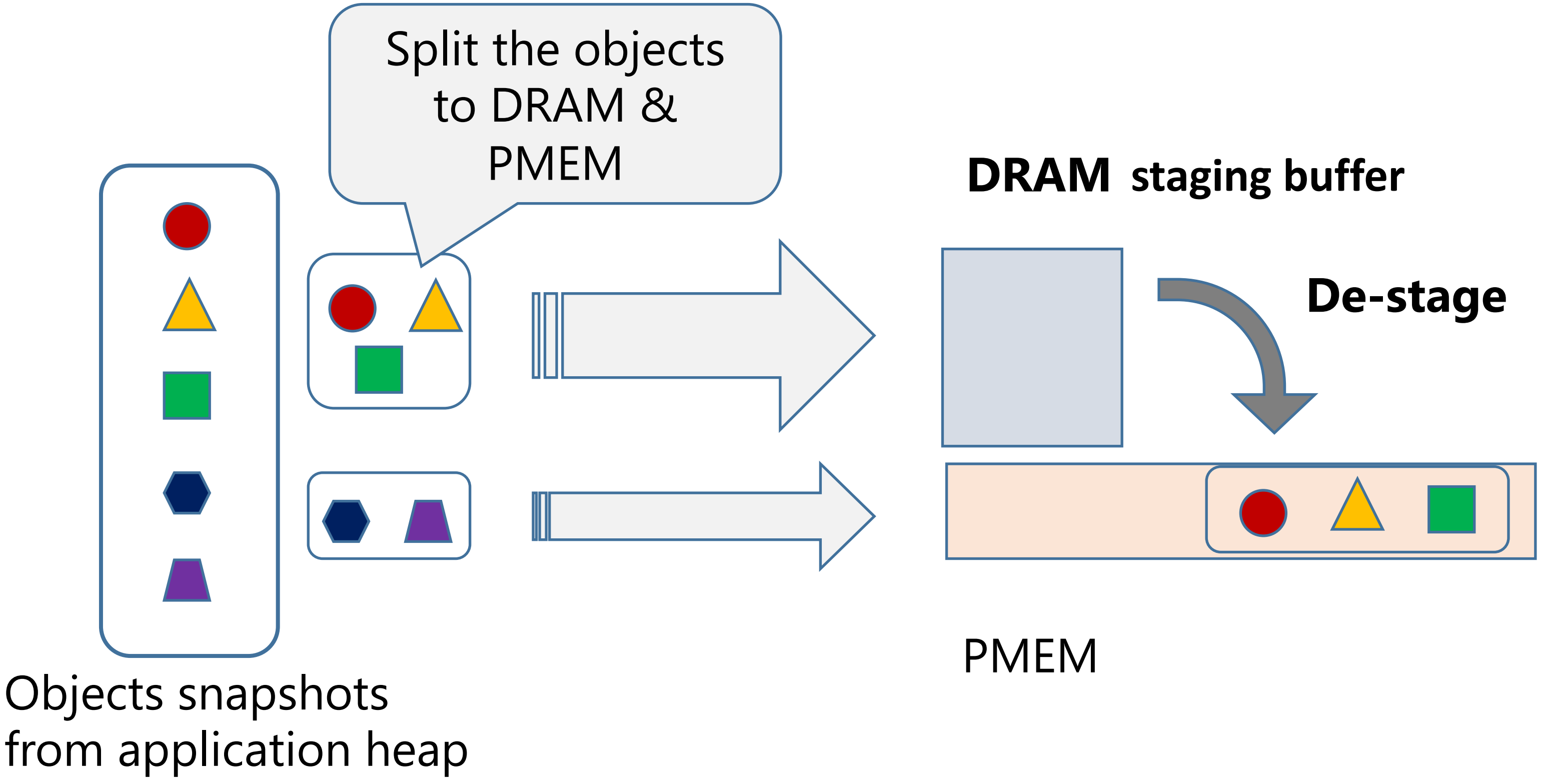
How to maximize the opportunity?

- Scheduling **data movement paths** across memory fabric [HiPC’16]
- Selectively use **DL for page placement** [HPDC’19]
- Configure **page management frequency** [MEMSYS’20, ….]
- **Page size** selection [CAL’20]
- **Capacity allocation** in workflows and multi-tenant workloads [MEMSYS’17]
- **Workflow placement** on cluster servers […]

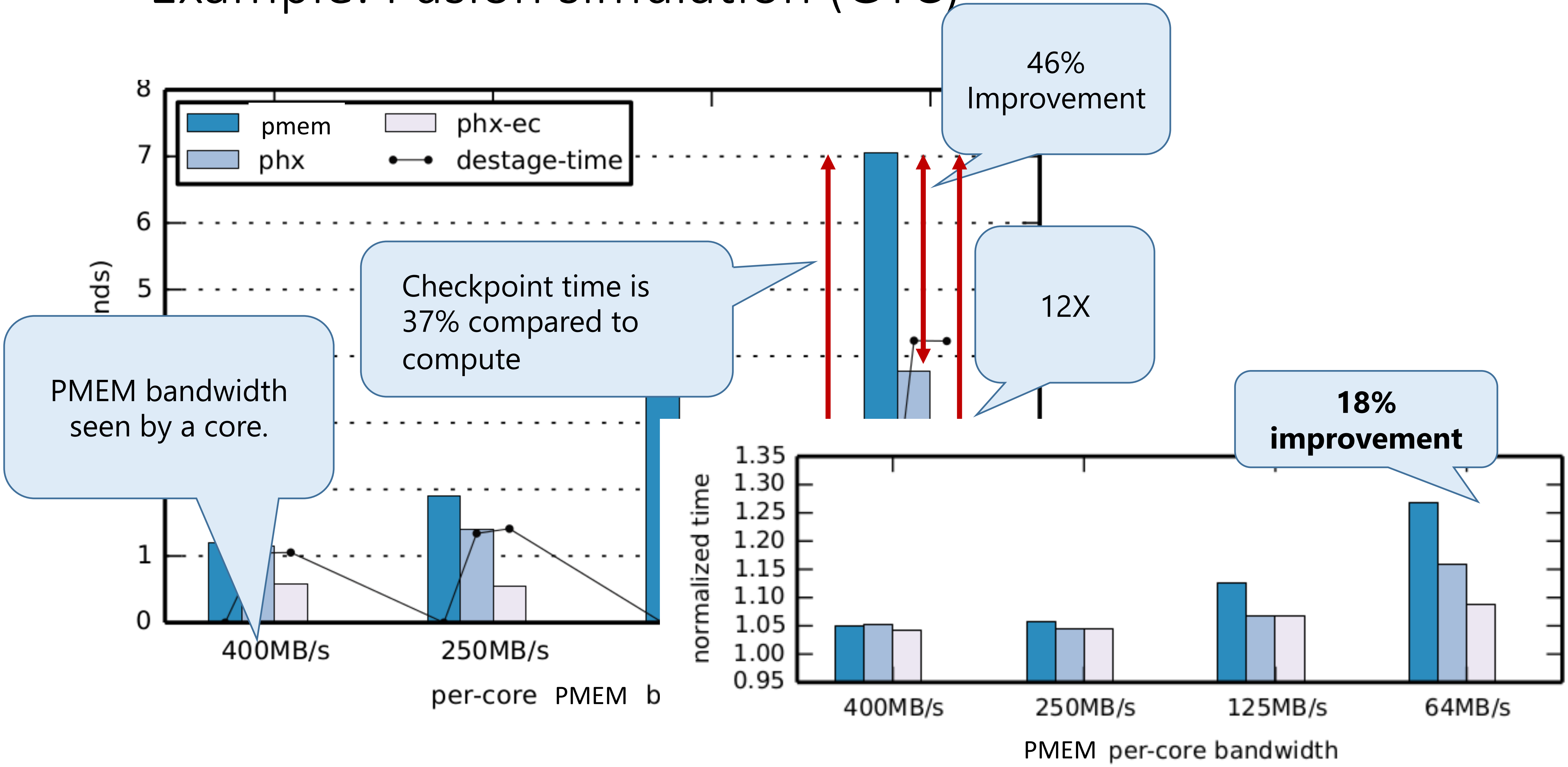
Phoenix: Data movement for PMEM-based Checkpoint I/O

Problem: Limited PMEM Bandwidth

- Support for highly concurrent PMEM access patterns: I/O from parallel computations (e.g., checkpoint, analytics pipelines…)
- Simultaneous bandwidth usage of both PMEM and DRAM
- Leverage **fast interconnect bandwidth to remote DRAM**


Split the objects to DRAM & PMEM
DRAM staging buffer
De-stage
PMEM
Objects snapshots from application heap

Example: Fusion simulation (GTC)



46% Improvement

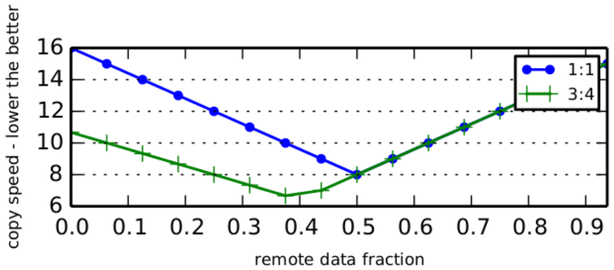
12X

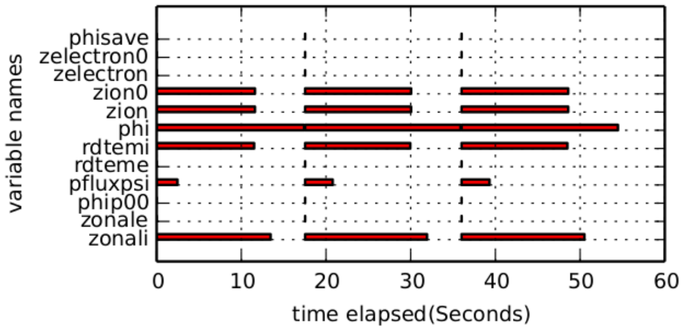
Checkpoint time is 37% compared to compute

PMEM bandwidth seen by a core.

18% improvement

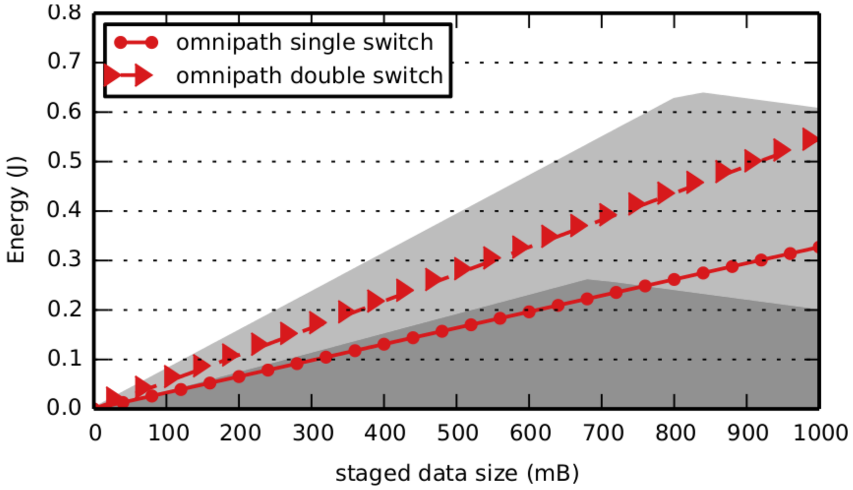
Problem: How to use different paths to memory?





http://www.computerhope.com

- How to split data ratio given the *available* bandwidths?
- When and which data to prioritize for staging?
  - Early access variables allow for optimizations such as ***pre-copy***
  - e.g., based on runtime profiling
- What do we need to optimize for?
  - memory budget allocated, performance, energy

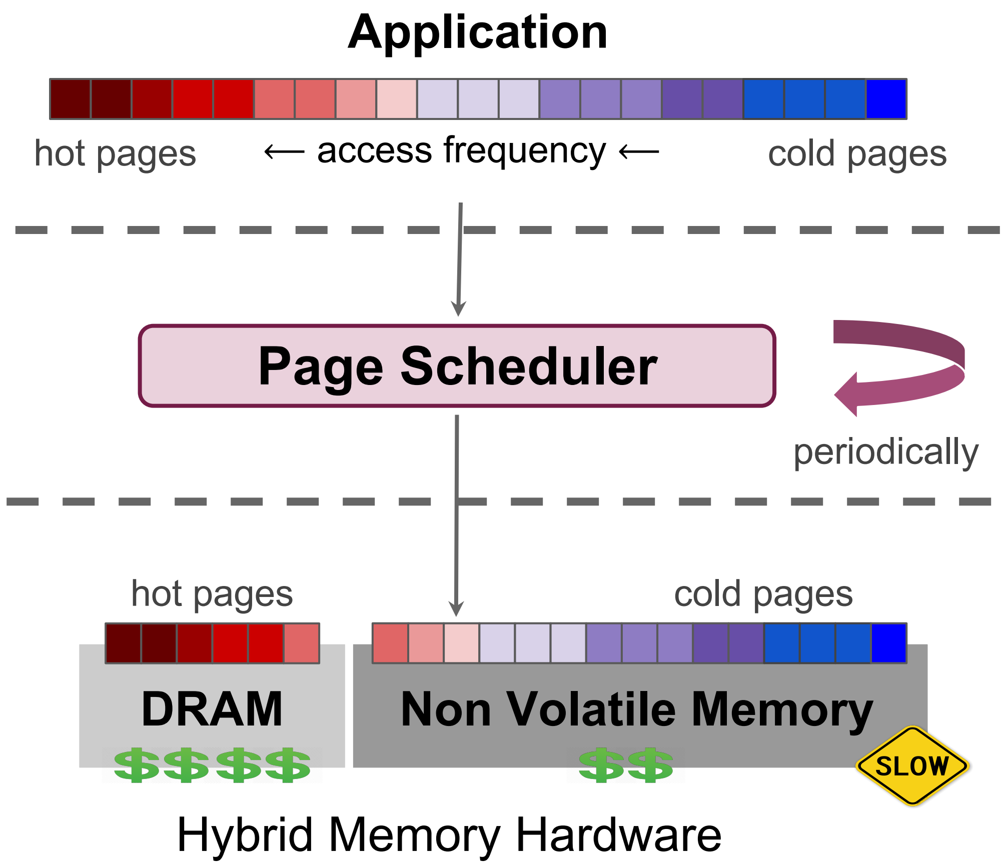


Problem: Which Pages to Move?

Dynamic Data Management in Hybrid Memory Systems

Problem: Which Pages to Move?

Dynamic Data Management in Hybrid Memory Systems



3. Problem
How to predict which data is hot so as to timely migrate it in DRAM.

2. Approach
Timely allocation in DRAM of frequently accessed (hot) data through periodic data migrations can boost application performance.

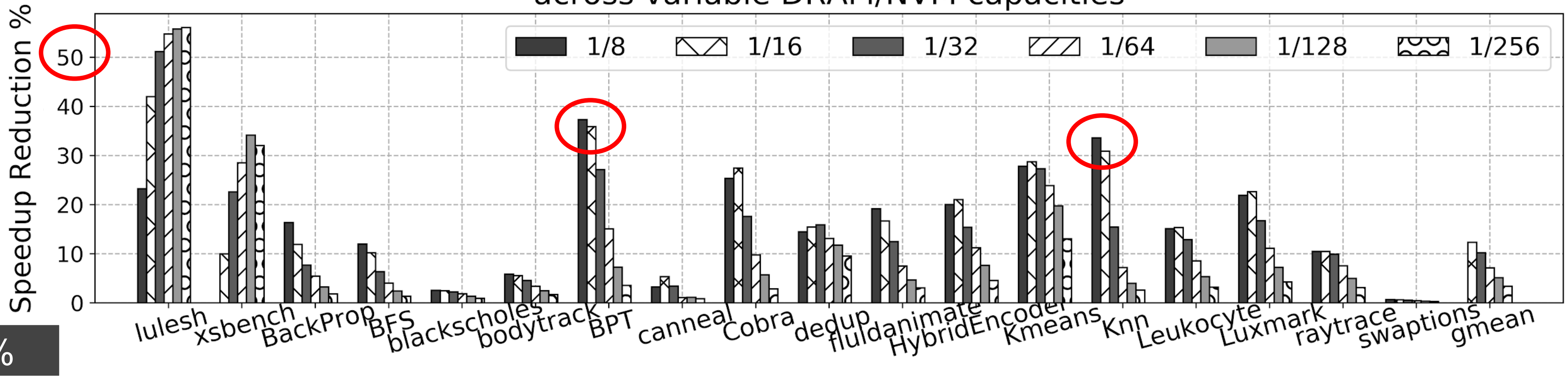
1. Challenge
Use of Non Volatile Memory (NVM) to extend main memory capacity reduces the system cost in return for application performance degradation.

Existing Solutions

Leave a significant gap for possible performance improvements

History = x%
The higher
The worse
Oracle = 0%

Performance gap between Oracle and History Page Schedulers across variable DRAM/NVM capacities

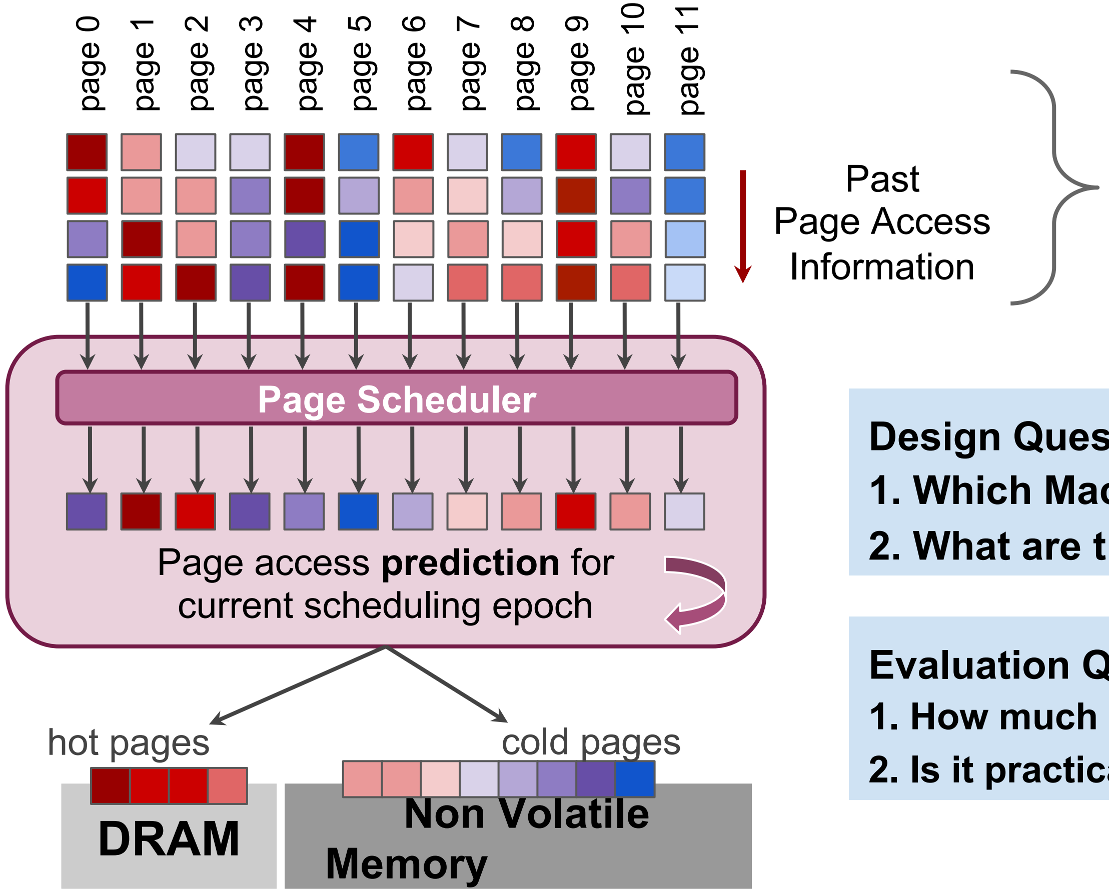


Added Performance Reduction due to Page Scheduling

Simple history-based page scheduling methods may end up causing significant <u>additional</u> performance degradation in applications executing over hybrid memory systems. We need something more clever to close the gap!

Solution Design

Questions that need to be answered



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 are the insights that MI can provide for page scheduling?

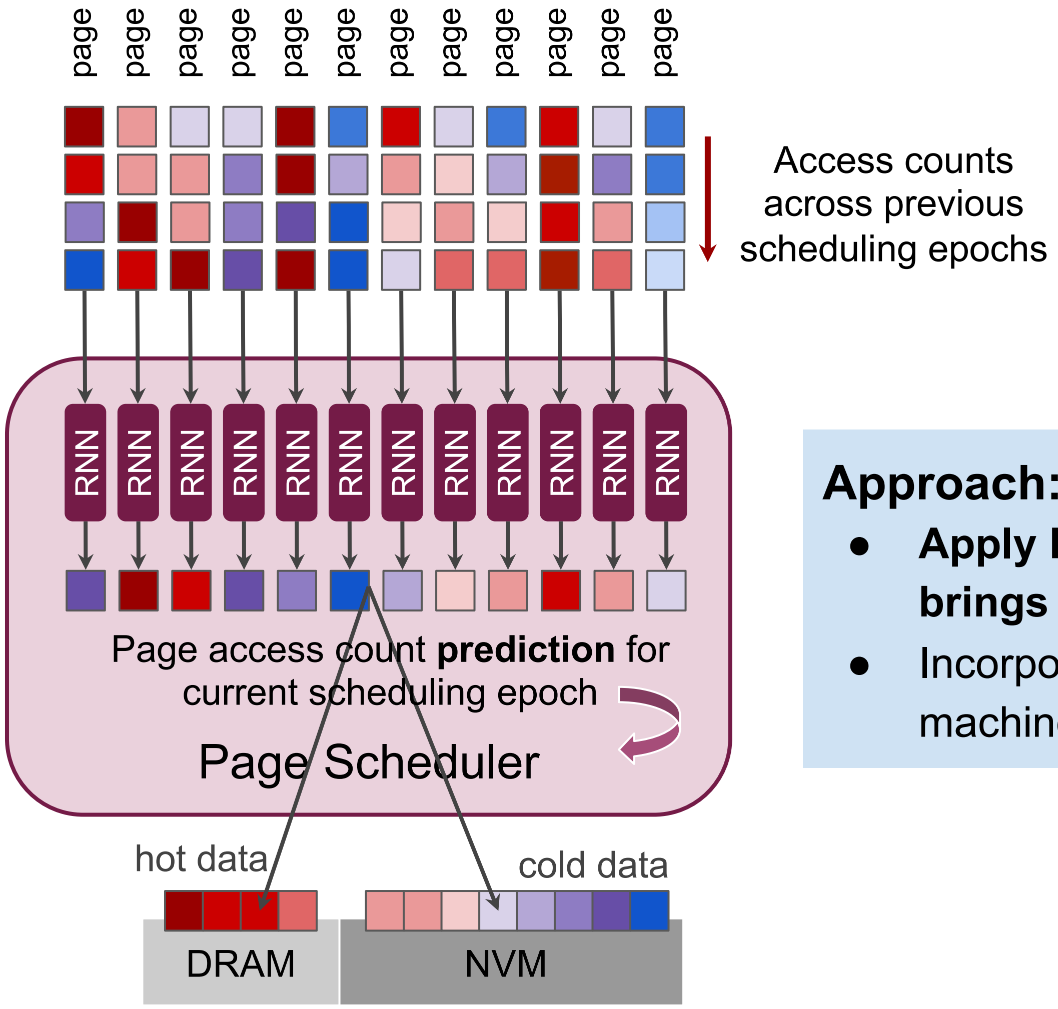
Evaluation Questions:

1. How much can it reduce the performance gap? How accurate are the predictions?

2. Is it practical to integrate into future systems?

Solution Design

Per Page Prediction of number of accesses


page
page
page
page
page
page
page
page
page
page
page
page
Access counts across previous scheduling epochs
RNN
Page access count **prediction** for current scheduling epoch
Page Scheduler
hot data
cold data
DRAM
NVM

Not really scalable..

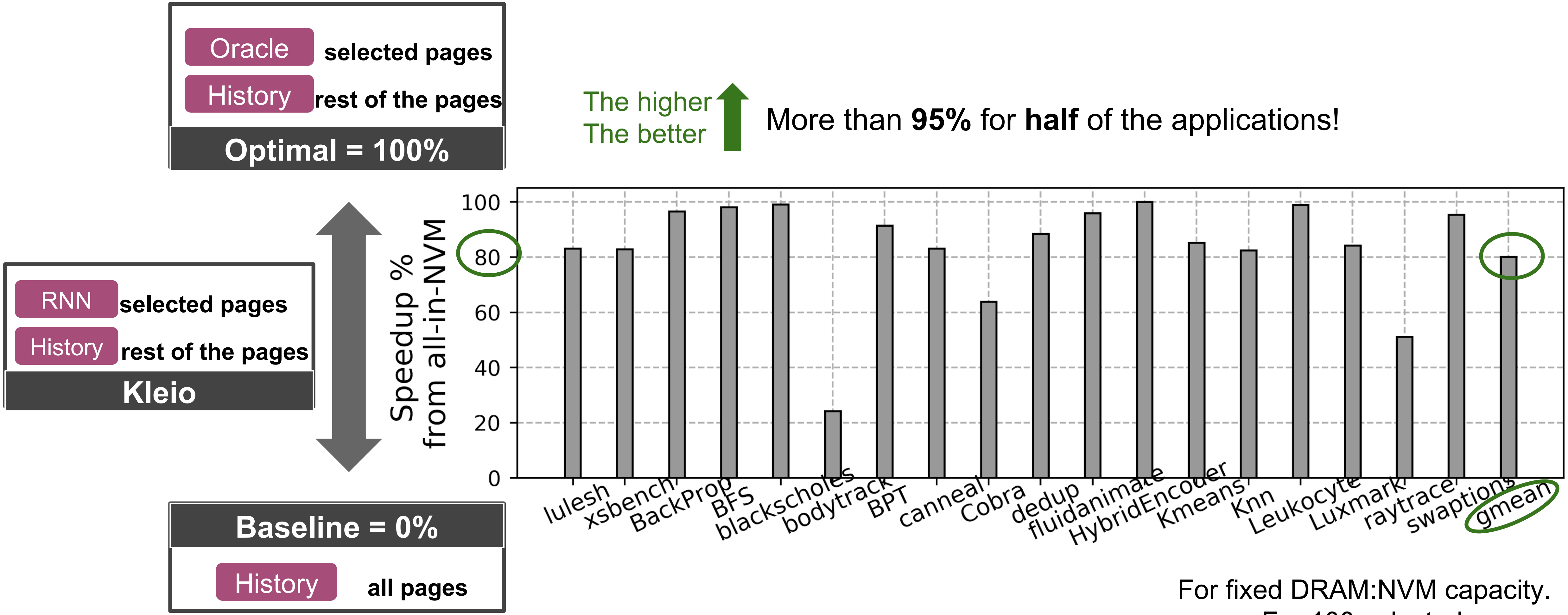
HPC and Big Data applications can have millions of pages!

Approach:

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

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.

Problem: When to move pages?

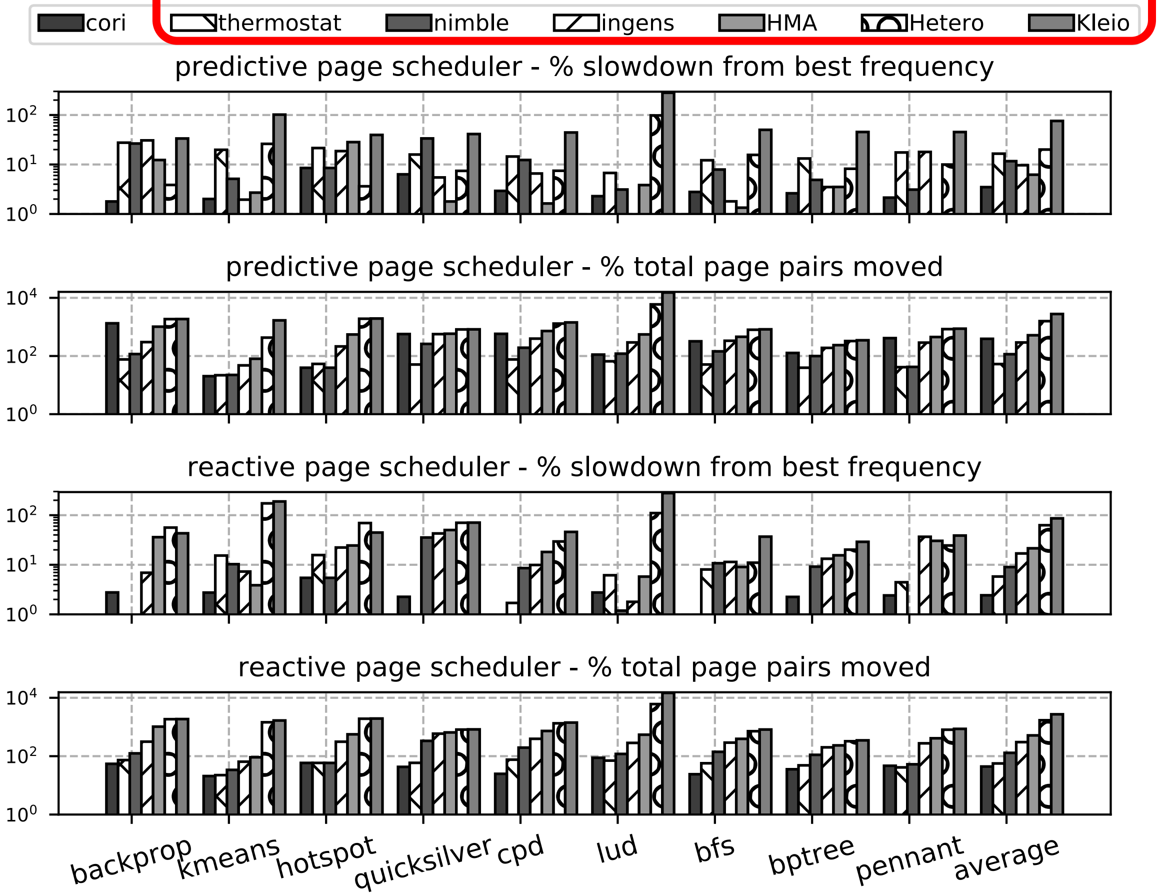
- Does it matter?

Solution	Period Duration
Thermostat [5]	10 sec
Nimble [38]	5 sec
Ingens [23]	2 sec
HMA [30]	1 sec
Hetero-OS [21], -Visor [17]	0.1 sec
Kleio [11]	0.01 sec
Unimem [36]	MPI phase

TABLE I: Frequency of data monitoring and movement across existing solutions mapped to our simulation-based analogy.

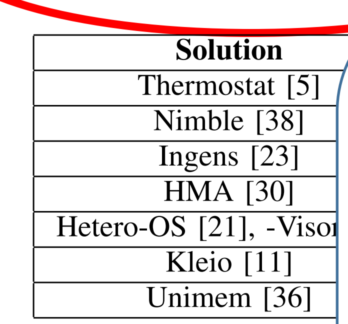
- **Cori:** data-driven and system-level tool for configuring memory management periodicity

No single answer



Problem: When to move pages?

- Does it matter?



Solution
Thermostat [5]
Nimble [38]
Ingens [23]
HMA [30]
Hetero-OS [21], -Visor
Kleio [11]
Unimem [36]

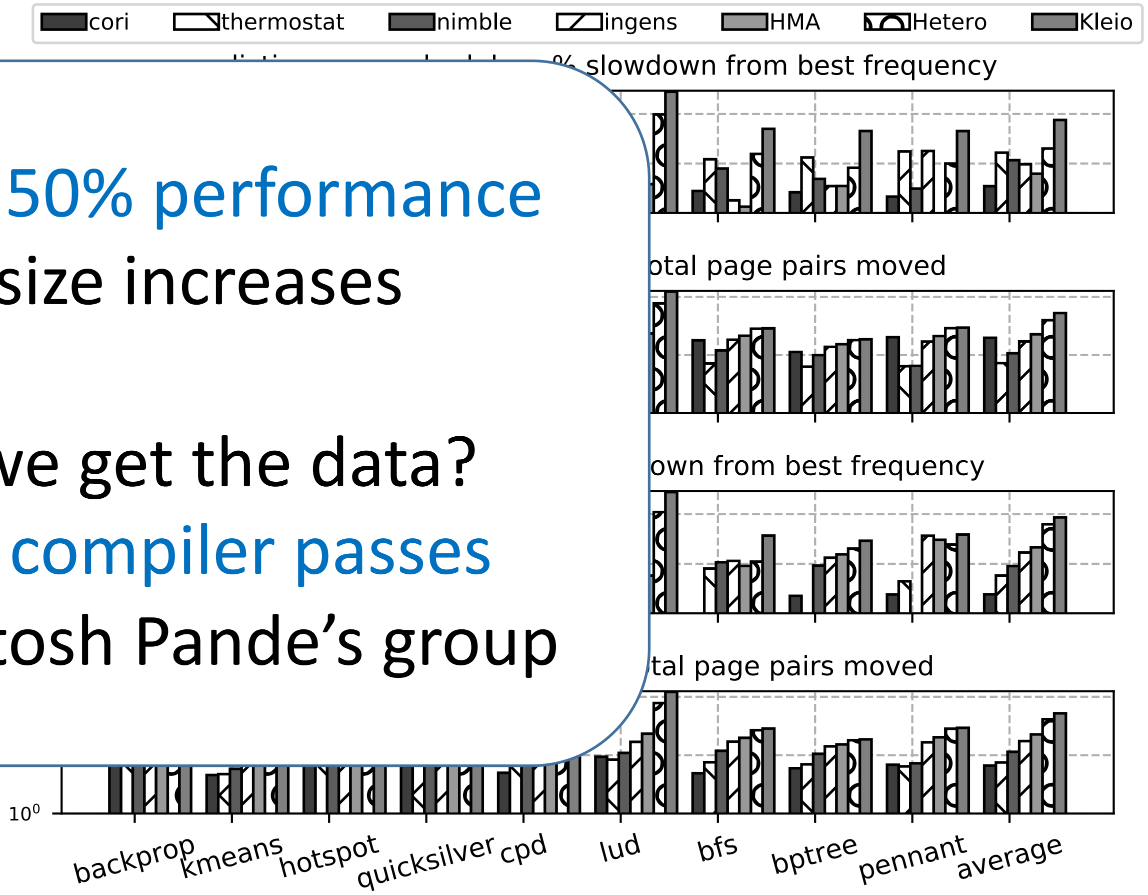
TABLE I: Frequenc
existing solutions n

- **Cori:** data-
  level tool f
  memory management
  periodicity

- Yes!
- On a real system -> 50% performance loss, worse as data size increases

Is this practical? Can we get the data?

- Yes! Hardware bits, compiler passes
  - Beacons w/ Santosh Pande’s group



Problem: When, which, where to move pages?

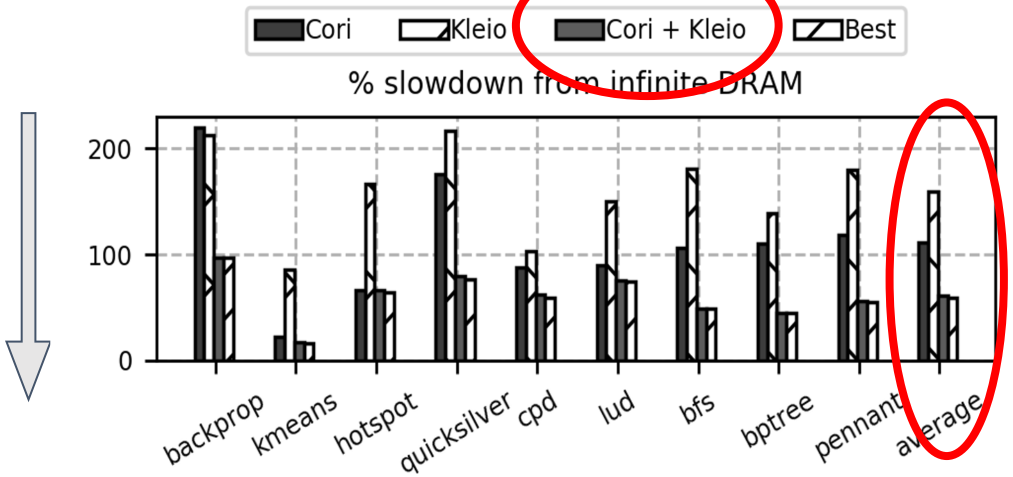
- Does it matter?

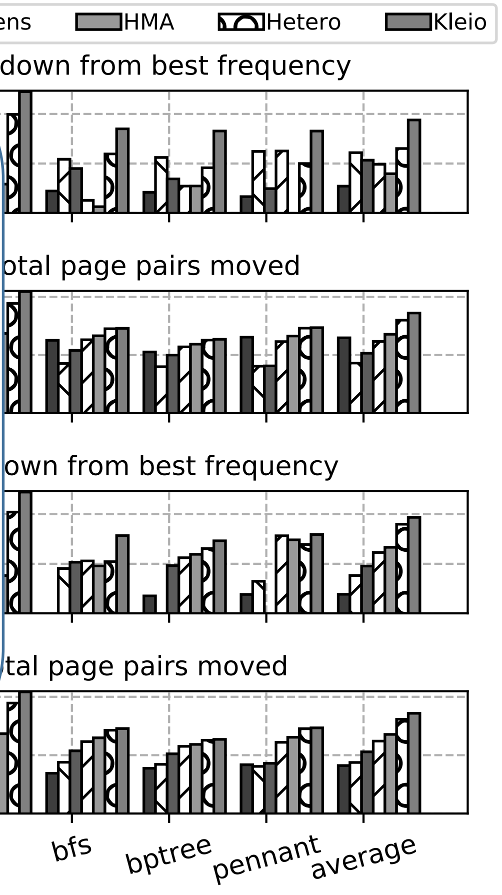
Solution
Thermostat [5]
Nimble [38]
Ingens [23]
HMA [30]
Hetero-OS [21], -Visor
Kleio [11]
Unimem [36]

TABLE I: Frequenc
existing solutions n

- **Cori:** data-
level tool f
memory management
periodicity

The lower the better





Takeaways

- Memory fabric heterogeneity introduces new tradeoffs & opportunities
  - From a small set of best practices
    - acceptable due to lower complexity, trivial decision, smaller scope
  - To an explosion of choices with major impact on performance and efficiency
  - Many more examples with similar observations

- Data-driven decisions on how to use new technologies as a path forward
- Rethink cross-stack techniques for making it possible and practical

KERNEL Group

Ada Gavrilovska

Greg Eisenhauer

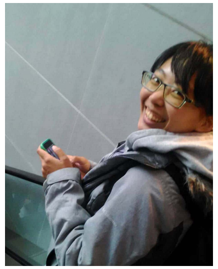
Ketan Bhardwaj

Pradeep Fernando

Thaleia Doudali

Ranjan S. Venkatesh

Harshit Daga


Carol Hsu

Rafael Oliveira

Tony Mason

Jim Choncholas

Jin Heo

Misun Park

Daniel Zahka

Vaibhav Bhasole

SRC® ada
Applications Driving Architectures

NSF

DEPARTMENT OF ENERGY UNITED STATES OF AMERICA

vmware®

intel®

facebook

SAMSUNG

Hewlett Packard
Enterprise

AT&T

GTRI

CISCO

Takeaways

- Memory fabric heterogeneity introduces new tradeoffs & opportunities
  - From a small set of best practices
    - acceptable due to lower complexity, trivial decision, smaller scope
  - To an explosion of choices with major impact on performance and efficiency
  - Many more examples with similar observations

- Data-driven decisions on how to use new technologies as a path forward
- Rethink cross-stack techniques for making it possible and practical