Cori: Dancing to the Right Beat of Periodic Data Movements over Hybrid Memory Systems

Thaleia Dimitra Doudali, Daniel Zahka, Ada Gavrilovska

@ IPDPS ’21
Heterogeneous (Hybrid) Memory Hardware

Application Classes

Exploded Data Sizes

Need for more and faster memory.

Emerging Memory Hardware

Scientific Simulations

Big Data

Artificial Intelligence

Video Analytics

Exploded Data Sizes

Low Latency

High Bandwidth

Data Persistence
Hybrid Memory Management Systems

Applications

- Video Analytics
- Machine Learning
- Science simulations

Data access patterns

System-level Memory Manager (Page Scheduler)

Data Access Monitoring

- Page access counts

Data Access Characterization

- Data tiering decision

Policy

- Reactive
  - Based only on past access history.
  - "Reacting" to behaviors.
- Predictive
  - Using robust models (e.g., Machine Learning).
  - "Predicting" behaviors.

Repeat periodically

Data Management

- Move pages across memories

Hybrid Memory

- MRAM
- HBM
- DRAM
- PMEM

Georgia Tech
CREATING THE NEXT
Plethora of Existing Solutions

Coordinated and Efficient Huge Page Management with Ingens

Youngjin Kwon, Hangchen Yu, Simon Peter, Christopher J. Rossbach¹, Emmett Witchel
The University of Texas at Austin
OSDI '16

HeteroOS - OS Design for Heterogeneous Memory Management in Datacenter

Sudarshan Kampani¹ Ada Gavriloška² Vishal Gupta³ KarstenSchwan³
¹Department of Computer Sciences, University of Wisconsin-Madison
³School of Computer Science, Georgia Tech,
²VMware
ISCA '17

Thermostat: Application-transparent Page Management for Two-tiered Main Memory

Neha Agarwal Thomas F. Wenisch
University of Michigan
nhaag@umich.edu, twenisch@umich.edu
ASPLOS '17

Nimble Page Management for Tiered Memory Systems

Ziyi Yan David Nellans
Rutgers University & NVIDIA
dzell@nvidia.com d nellans@nvidia.com
ASPLOS '19

High Performance Distributed Systems (Best Paper Nominees)

Kleio: A Hybrid Memory Page Scheduler with Machine Intelligence

Thaleia Dimitra Doudalii
Georgia Institute of Technology
thdoudali@gatech.edu
HPDC '19

Sergey Blagodurov
Advanced Micro Devices, Inc.
Sergey.Blagodurov@amd.com
Abhinav Vishnu
Advanced Micro Devices, Inc.
Abhinav.Vishnu@amd.com

Sudhanvi Gurumurthi
Advanced Micro Devices, Inc.
Sudhanvi.Gurumurthi@amd.com
Ada Gavriloška
Georgia Institute of Technology
ada@cc.gatech.edu

All these systems make periodic memory management decisions, based on reactive or predictive policies.
Lost Opportunity for Performance Due to empirical configuration.

Systems are **empirically** tuned.

Periodicity differs by **orders of magnitude**!

Which period duration to use? Which one maximizes performance?

<table>
<thead>
<tr>
<th>System</th>
<th>Periodicity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thermostat</td>
<td>10 sec</td>
</tr>
<tr>
<td>Nimble</td>
<td>5 sec</td>
</tr>
<tr>
<td>Ingens</td>
<td>2 sec</td>
</tr>
<tr>
<td>HMA</td>
<td>1 sec</td>
</tr>
<tr>
<td>Hetero-OS</td>
<td>0.1 sec</td>
</tr>
<tr>
<td>Kleio</td>
<td>0.01 sec</td>
</tr>
</tbody>
</table>

No single proposed period value maximizes performance across applications and schedulers. **10% - 100% performance slowdown.**
Empirical Configuration
Execution-based tuning of the periodicity.

1. Set the page scheduler's frequency
2. Execute Application
3. If low performance, try next frequency

List of Candidate Periods
E.g., [10 sec, 1 sec, 0.01 sec]

Empirical Selection
Replacing Empirical with Insight-based Configuration

Execution-based tuning of the periodicity.

Solution Goals:
- Minimize Overheads (Tuning Trials).
- Maximize Performance.

Need for insight!
Page Reuse Distance = The time gap between two accesses to the same page.

Period Length

- Very short = Overheads
- ≈ Sweet spot
- Very long = Insufficient

Insight: Periods that align with the data reuse distance, maximize performance.
System Design of “Cori”

Cori is an insight-based system-level solution for tuning the frequency of periodic page schedulers.

1. Reuse Collector
   - Memory Access Trace
   - Order of tuning trials

2. Frequency Generator
   - Calculate “Dominant Reuse (DR) Time”
   - $DR = \frac{\sum_{i=1}^{N} (N - i) \times repeat_i \times reuse_i}{\sum_{i=1}^{N} (N - i) \times repeat_i}$

3. Tuner
   - CandidatePeriods $= [DR, 2 \times DR, \ldots, \frac{Runtime}{2}]$

Input

Order of tuning trials

Application Execution

Cori

Page Scheduler

tuning trials
Evaluation Methodology

Metrics
- Application performance.
  Slowdown from optimally selected frequency (identified via extensive experimentation).
- Tuning Overheads.
  Number of trials to find the frequency that delivers best performance.

Comparison
- Proposed values from existing solutions.
  HMA [HPCA ’15], Ingens [OSDI ’16], Hetero-OS [ISCA ’17], Thermostat [ASPLOS ’17], Nimble [ASPLOS ’19], Kleio [HPDC ’19].
- Cori’s selection of period values that differ by the dominant reuse time.
  Tuning trials in increasing order of values.
- “Baseline” selection of period values that differ by a constant time step.
  Tuning trials in increasing, decreasing and random order of values.

Methodology
- Python-based simulation of hybrid memory system and page scheduler.
  https://github.com/GTkernel/cori-sim
- Validation using a hardware testbed with DRAM and Intel’s Optane persistent memory.
  https://github.com/GTkernel/x86-Linux-Page-Scheduler
Application performance.

Cori reduces the performance slowdown down to only 3% across applications and page schedulers, closing the 10% -100% gap.
Evaluation (2)

Number of tuning trials needed to find best performance.

The lower the better

Cori only requires 2 trials on average for predictive ML-based schedulers.

Insight-less baselines are sensitive to the trial ordering (left, right, random).

5x reduction in tuning trials
Evaluation (3)
Validation on Optane persistent memory.

Even a difference of 1-2 seconds in period duration can reduce performance by 30%-50%.
**Summary of Cori**

Greek Trivia: According to the ancient Greek mythology, Cori (short for Terpsichore) was the muse of dance, sister of Kleio, daughter of Mnemosyne, goddess of memory.

Cori is open source.

Cori delivers **maximum** performance improvements for **minimal** tuning overheads.