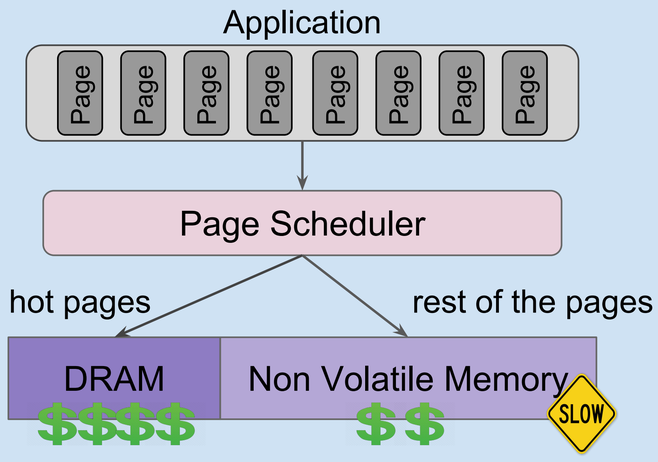
Page Scheduling on Hybrid Memory Systems with Machine Intelligence

Thaleia Dimitra Doudali, Ada Gavrilovska
Georgia Institute of Technology
Sergey Blagodurov, Abhinav Vishnu, Sudhanva Gurumurthi
Advanced Micro Devices, Inc.

1. Research Area



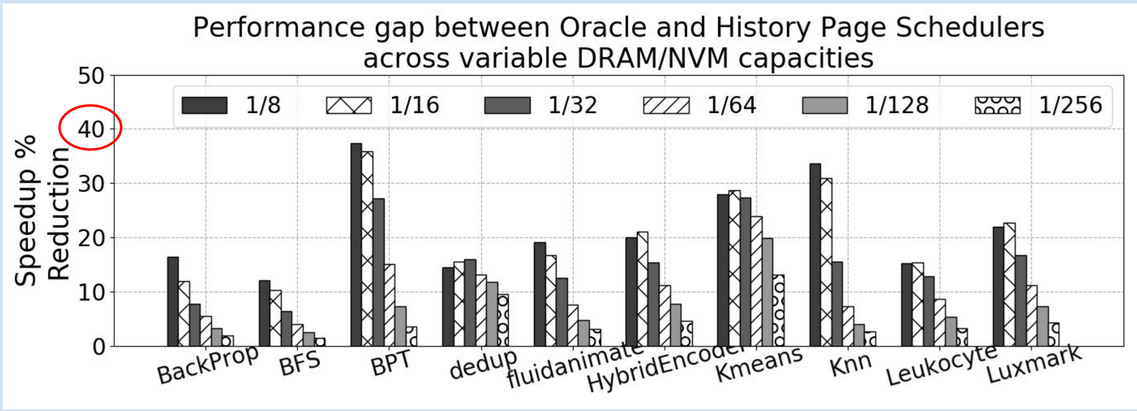
Hybrid Memory Systems: The high cost of DRAM in large capacities, limits its use in future systems. Denser technologies (e.g., NVM) are expected to extend the memory physical address space, thus the system's overall main memory capacity.

Page Scheduler: Periodically migrates application pages, allocating future hot pages in the fastest memory component (e.g., DRAM), until capacity is full. This methodology is proven to boost performance.

2. Motivation

Oracle Page Scheduler: Assumes oracular knowledge of the application's memory access pattern, setting the theoretical upper bound of performance that the application can achieve over the hybrid memory system.

History Page Scheduler: Current state-of-the-art design, which utilizes the immediately observed access pattern behavior to make future predictions. [1]

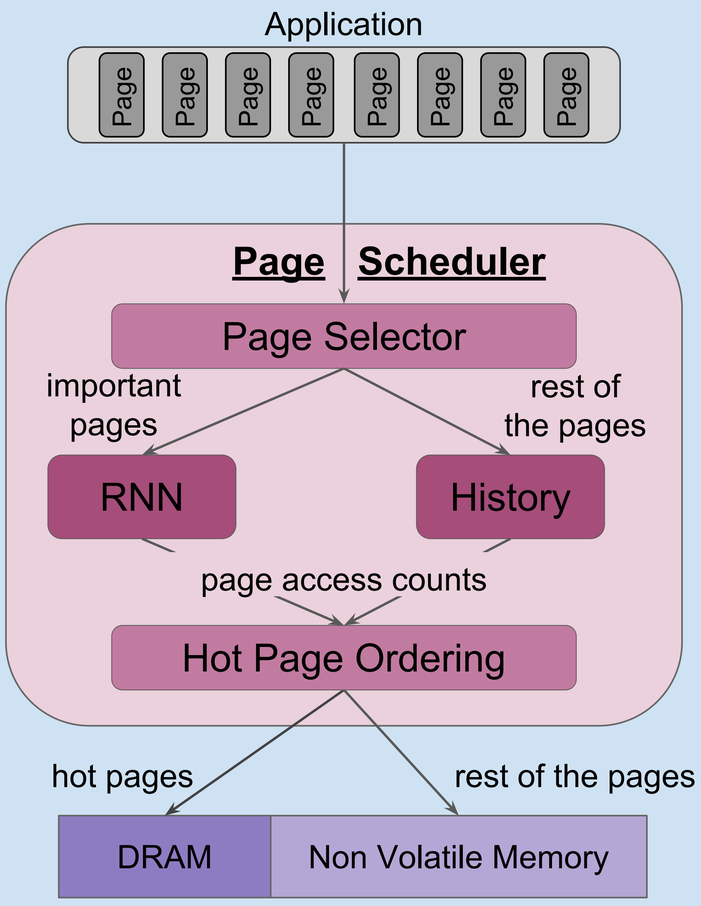


Performance Gap: The History Page Scheduler, results in up to **40%** reduction of the optimal application performance (i.e., speedup from the case where all data is allocated in NVM).

3. Problem Statement

Existing Page Scheduler designs fail to accurately predict future highly accessed pages. Thus, they do not migrate them to DRAM in a timely fashion, leading in significant performance reduction. *Can **Machine Intelligence** provide us with more **accurate predictions**, so as to deliver higher application performance?*

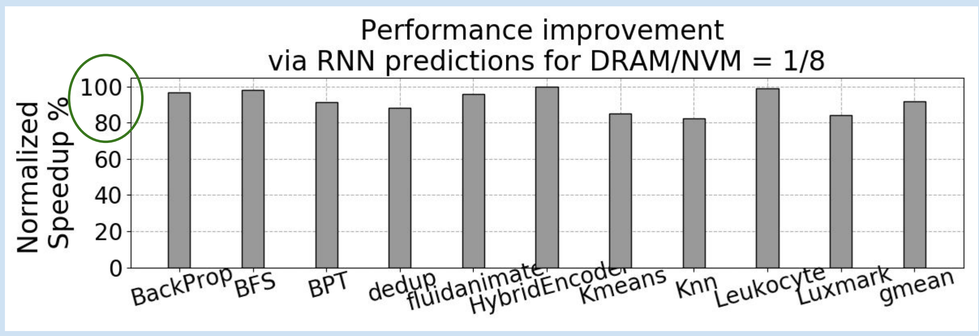
4. Solution



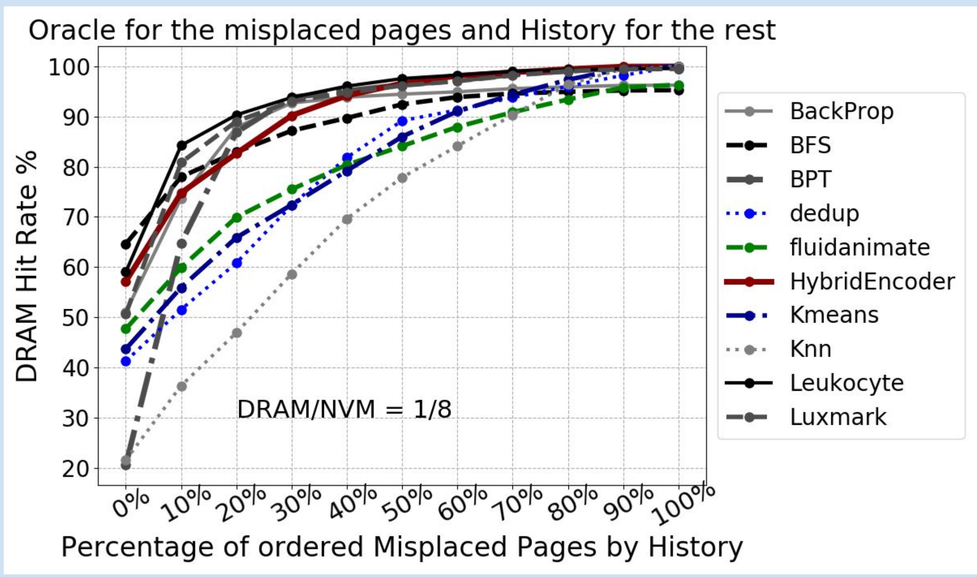
Page Scheduler Overview:

- Identifies the set of application pages whose timely placement in DRAM will bring most of the performance improvement (i.e., important pages).
- Trains separate Recurrent Neural Networks for each of these pages, learning the aggregate number of per page access counts across the scheduling intervals.
- Leverages the lightweight methodology followed by the current state-of-the-art History Page Scheduler for the rest of the pages.
- During a scheduling interval, the scheduler collects the predicted access counts and orders pages in descending access frequency, prioritizing DRAM allocations for the hot pages, until capacity is full.

5. Results



Achieving 1st Goal: We bridge **95%** of the performance gap between the Oracle and History Page Schedulers when applying RNN access count predictions for the first 100 most important to performance pages.



Achieving 2nd Goal: We deliver low training and inference times by:

- Reducing the number of application pages whose timely allocation in DRAM through the RNN predicted access counts significantly boosts performance.
- Leveraging the current state-of-the-art methodology for the pages whose access pattern can be predicted without machine intelligence, or whose access frequency does not drastically impact performance.

[1] M. R. Meswani, S. Blagodurov, D. Roberts, J. Slice, M. Ignatowski and G. H. Loh, "Heterogeneous memory architectures: A HW/SW approach for mixing die-stacked and off-package memories," *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*, Burlingame, CA, USA, 2015, pp. 126-136. doi:10.1109/HPCA.2015.7056027