Constructing Dynamic Policies for Paging Mode Selection

Jason Hiebel Laura E. Brown Zhenlin Wang

jshiebel@mtu.edu lebrown@mtu.edu zlwang@mtu.edu

Department of Computer Science
Michigan Technological University

International Conference on Parallel Processing
August 2018

Paging Mode Selection

Contextual Bandits

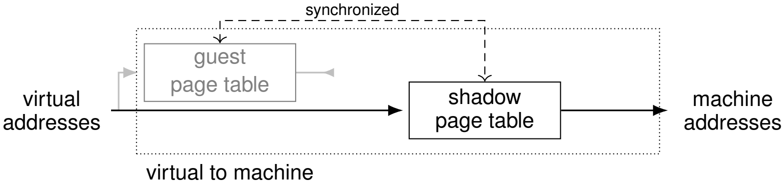
DSP-OFFSET

Evaluation

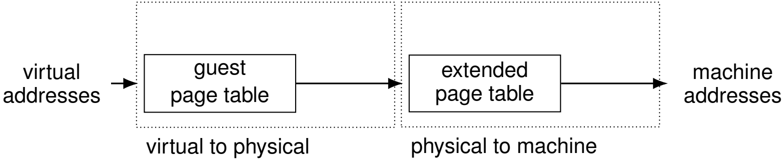
Conclusion

Virtual Address Translation

Shadow Paging (SP)



Hardware-Assisted Paging (HAP)



Workload Behavior Determines Performance

	Execution Time (s)		
Benchmark	**HAP**	**SP**	**SP / HAP**
gcc	413	632	+ 53%
tonto	950	1150	+ 21%
mcf	385	340	– 12%
cactusADM	1610	1309	– 19%

Shadow Paging

Page faults cause expensive context switches and VM exits

Hardware Assisted Paging

DTLB misses more expensive due to extended page table

Workload Behavior Determines Performance

	Execution Time (s)		
Benchmark	**HAP**	**SP**	**SP / HAP**
gcc	413	632	+ 53%
tonto	950	1150	+ 21%
mcf	385	340	– 12%
cactusADM	1610	1309	– 19%

Shadow Paging

Page faults cause expensive context switches and VM exits

Hardware Assisted Paging

DTLB misses more expensive due to extended page table

Workload Behavior Determines Performance

	Execution Time (s)		
Benchmark	**HAP**	**SP**	**SP / HAP**
gcc	413	632	+ 53%
tonto	950	1150	+ 21%
mcf	385	340	– 12%
cactusADM	1610	1309	– 19%

Shadow Paging

Page faults cause expensive context switches and VM exits

Hardware Assisted Paging

DTLB misses more expensive due to extended page table

Paging Mode Selection

Goal

- Utilize paging mode most suited to the current workload

Dynamic Selection

- Periodically select paging mode based on runtime behavior (page fault count, DTLB miss count)
- Paging mode performance depends on hardware, software
  - memory hierarchy
  - address space size

Existing Selection Methods

DSP-Manual (Wang et al.; VEE ‘11)

- Model constructed by domain experts
- Requires extensive manual profiling and analysis

ASP-SVM (Kuang et al.; ML ‘15)

- Model constructed using off-the-shelf machine learning tools (Support Vector Machines)
- Requires enumerative profiling method

DSP-OFFSET

Overview

- Paging mode selection as a **contextual bandit**
- Construct model using simple, uniformly random profiling

Advantages

- Equivalent performance to state-of-the-art (ASP-SVM)
- Significant (90%) reduction in profiling time

Paging Mode Selection

Contextual Bandits

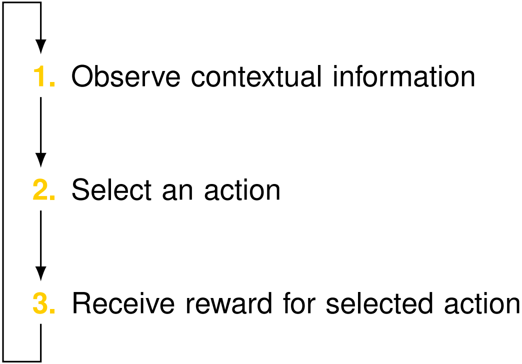
DSP-OFFSET

Evaluation

Conclusion

The Contextual Bandit

- Sequential decision making with limited feedback



1. Observe contextual information
2. Select an action
3. Receive reward for selected action

Action Selection

Online Selection

- Interactive — interleave exploration and exploitation
- Techniques not amenable to low-level implementation

Offline Selection

- Non-interactive — exploration before exploitation
- Learn from logged (random) choices

Contextual Bandit Formulation

Contextual Information

- Page Faults
- DTLB Misses

Action Space

- Hardware-Assisted Paging
- Shadow Paging

Reward Function

- Throughput (Instructions Per Cycle)

Paging Mode Selection

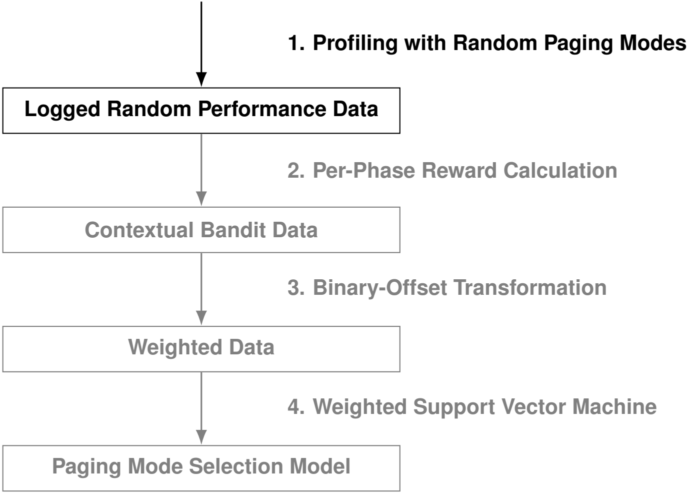
Contextual Bandits

DSP-OFFSET

Evaluation

Conclusion

DSP-OFFSET



1. Profiling with Random Paging Modes

Logged Random Performance Data

2. Per-Phase Reward Calculation

Contextual Bandit Data

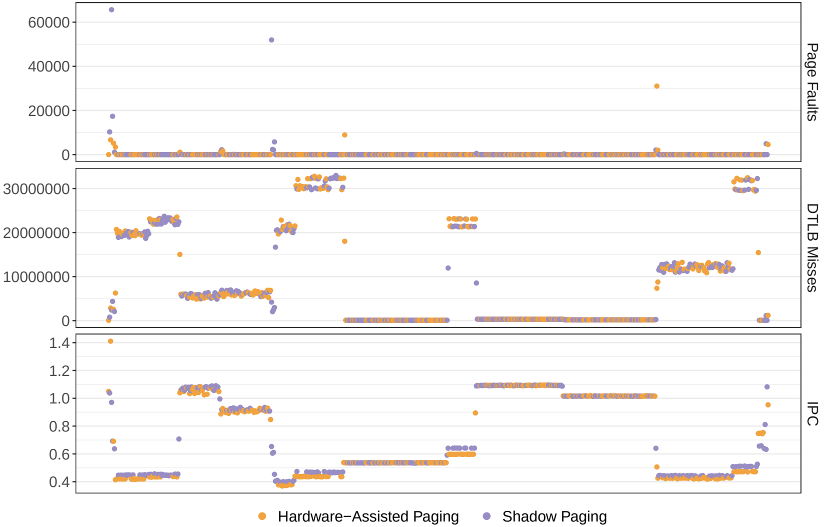
3. Binary-Offset Transformation

Weighted Data

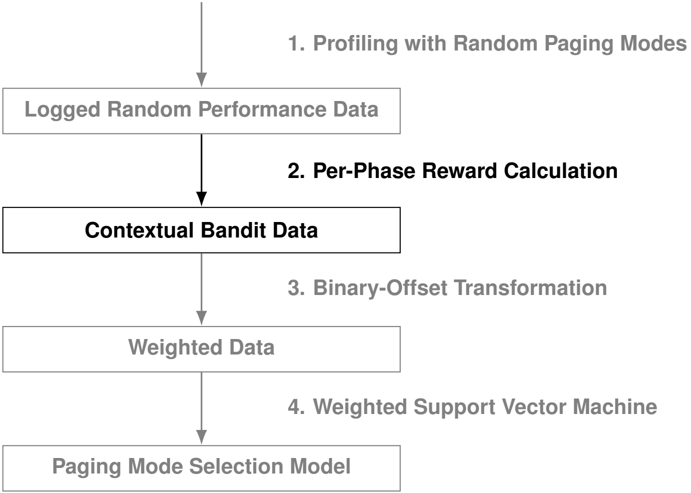
4. Weighted Support Vector Machine

Paging Mode Selection Model

1. Profiling with Random Paging Modes



DSP-OFFSET



1. Profiling with Random Paging Modes

Logged Random Performance Data

2. Per-Phase Reward Calculation

Contextual Bandit Data

3. Binary-Offset Transformation

Weighted Data

4. Weighted Support Vector Machine

Paging Mode Selection Model

2. Per-Phase Reward Calculation

Phase Detection

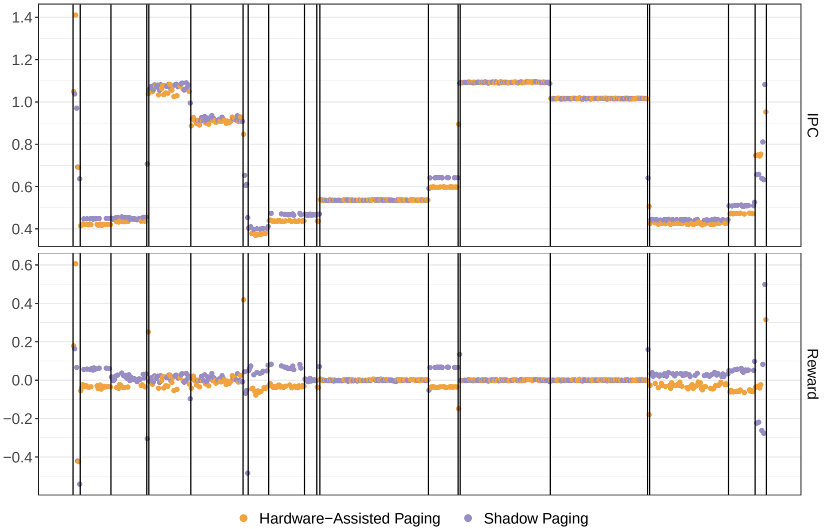
- Determine program phasing from random profiling data
- Segment data into phases using IPC change-points
- PELT (Pruned Exact Linear Time) change-point detection

Reward

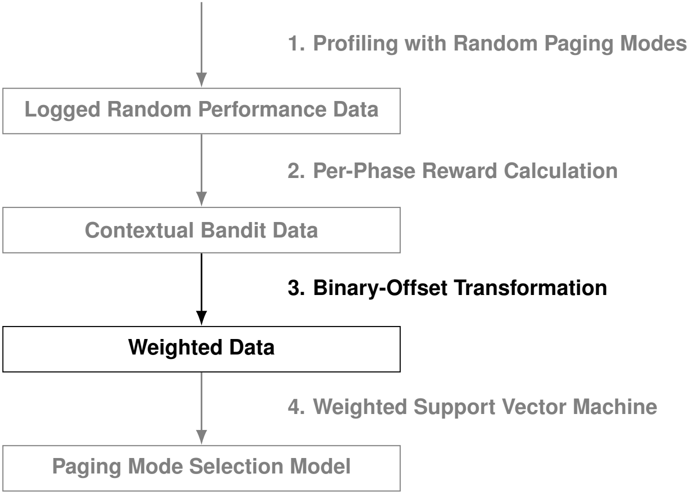
- Normalized *IPC* per-phase

$$reward_i = \log \frac{IPC_i}{\mathrm{mean}(IPC_{phase_i})}$$

2. Per-Phase Reward Calculation



DSP-OFFSET



1. Profiling with Random Paging Modes

Logged Random Performance Data

2. Per-Phase Reward Calculation

Contextual Bandit Data

3. Binary-Offset Transformation

Weighted Data

4. Weighted Support Vector Machine

Paging Mode Selection Model

3. Binary-Offset Transformation

(Beygelzimer and Langford; SIGKDD '09)

$$\begin{pmatrix} \textit{context} \\ \textit{action} \\ \textit{reward} \end{pmatrix} \Rightarrow \begin{pmatrix} \textit{context} \\ \textit{label} \\ \textit{weight} \end{pmatrix}$$

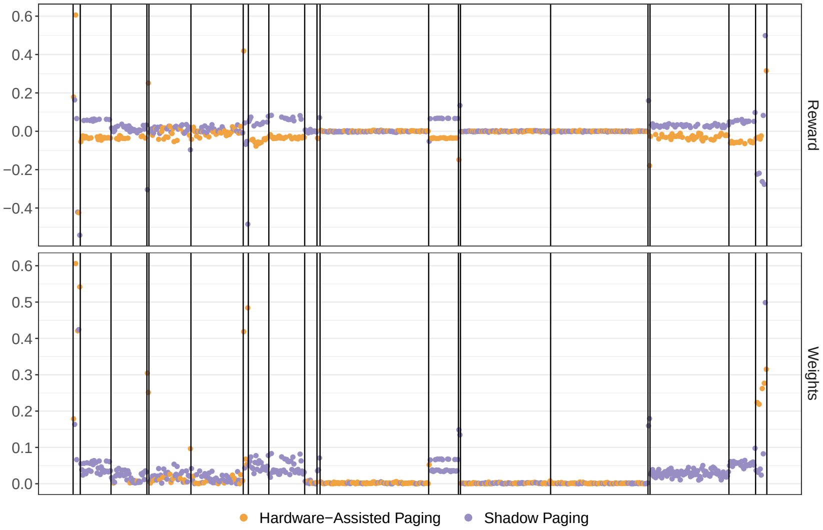
- Does the selected action perform **better** than average?

  $\textit{reward} > 0 : (\textit{label}, \textit{weight}) \leftarrow (\textit{action}, \textit{reward})$

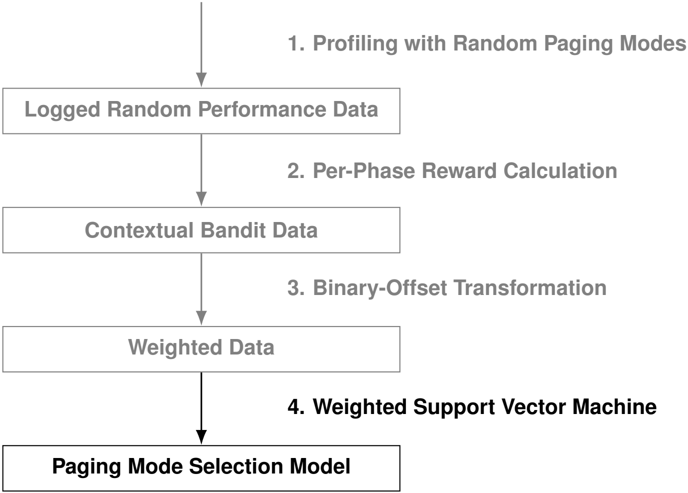
- Does the selected action perform **worse** than average?

  $\textit{reward} < 0 : (\textit{label}, \textit{weight}) \leftarrow (\textit{opposite action}, |\textit{reward}|)$

3. Binary-Offset Transformation



DSP-OFFSET



1. Profiling with Random Paging Modes

Logged Random Performance Data

2. Per-Phase Reward Calculation

Contextual Bandit Data

3. Binary-Offset Transformation

Weighted Data

4. **Weighted Support Vector Machine**

Paging Mode Selection Model

4. Weighted Support Vector Machine

- Construct linear classifier for $\vec{x} = \{\text{Page Faults}, \text{DTLB Misses}\}$

$$f(\vec{x}) = \text{sign}(\vec{\beta} \cdot \vec{x} + \beta_0)$$

  using the weighted support vector machine

- Prevent rapid switching using a margin

$$\begin{array}{ll} \vec{\beta} \cdot \vec{x} + \beta_0 > +0.25 & \text{switch to Shadow Paging} \\ \vec{\beta} \cdot \vec{x} + \beta_0 < -0.25 & \text{switch to Hardware-Assisted Paging} \end{array}$$

- Low overhead Xen VM implementation

Paging Mode Selection

Contextual Bandits

DSP-OFFSET

Evaluation

Conclusion

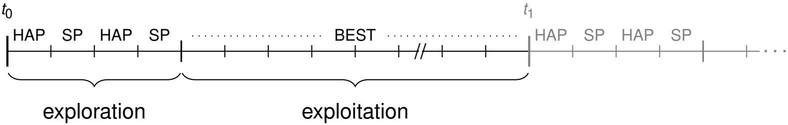
Evaluation Environment

CPU (GHz)	**Cache** (KB)			**DTLB** (entries)	
	L1	**L2**	**L3**	**L1**	**L2**
2.8	64 4-way	512 8-way	8192 16-way	64 4-way	512 4-way

- 1st generation Intel Core i5 processor (Nehalem)
- Xen 3.3.1 with paging mode selection patch
- 32-bit guest OS with 1 dedicated core and 3 GB memory

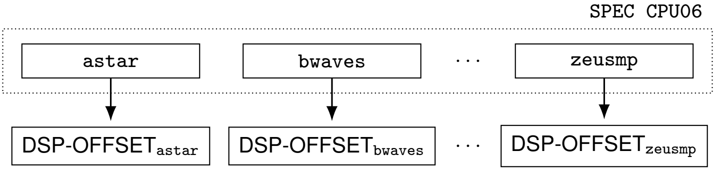
DSP-SAMPLE

- Simple online direct sampling method
- Alternate between exploration and exploitation
  ("context-less" bandit)



DSP-OFFSET

Benchmark-Specific



SPEC CPU06

astar

bwaves

...

zeusmp

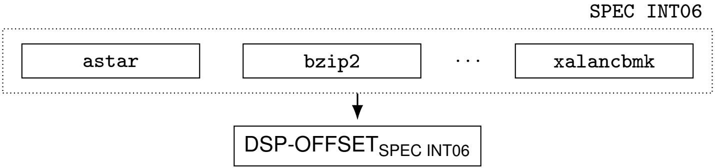
$\text{DSP-OFFSET}_{\texttt{astar}}$

$\text{DSP-OFFSET}_{\texttt{bwaves}}$

...

$\text{DSP-OFFSET}_{\texttt{zeusmp}}$

Benchmark-Agnostic



SPEC INT06

astar

bzip2

...

xalancbmk

$\text{DSP-OFFSET}_{\text{SPEC INT06}}$

Paging Mode Selection Models

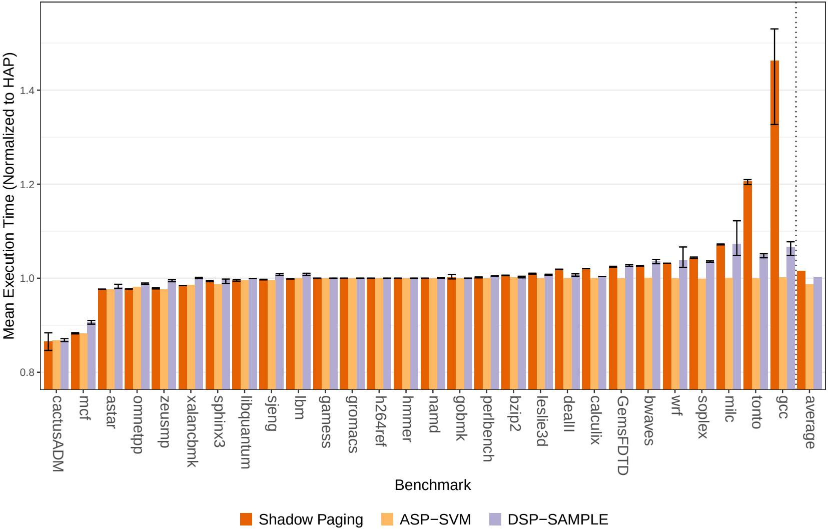
Static Models

- Hardware-Assisted Paging (baseline)
- Shadow Paging

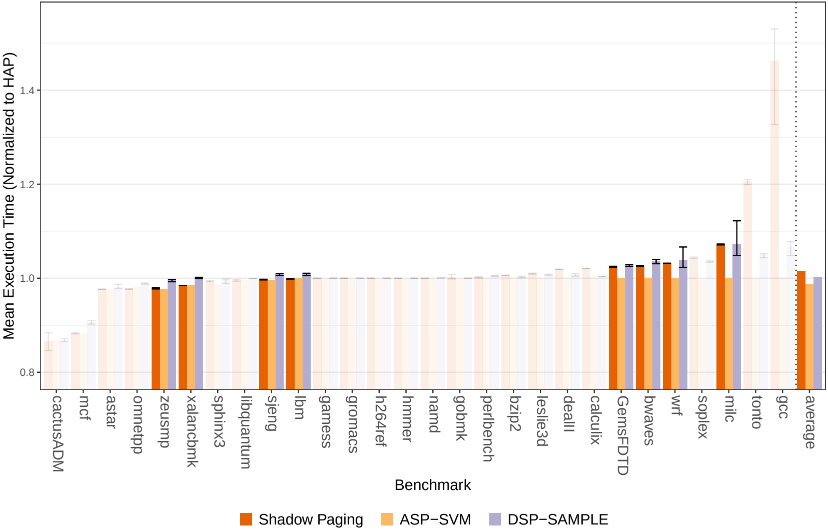
Dynamic Models

- ASP-SVM (state-of-the-art)
- DSP-SAMPLE
- DSP-OFFSET (Benchmark-Agnostic, Benchmark-Specific)

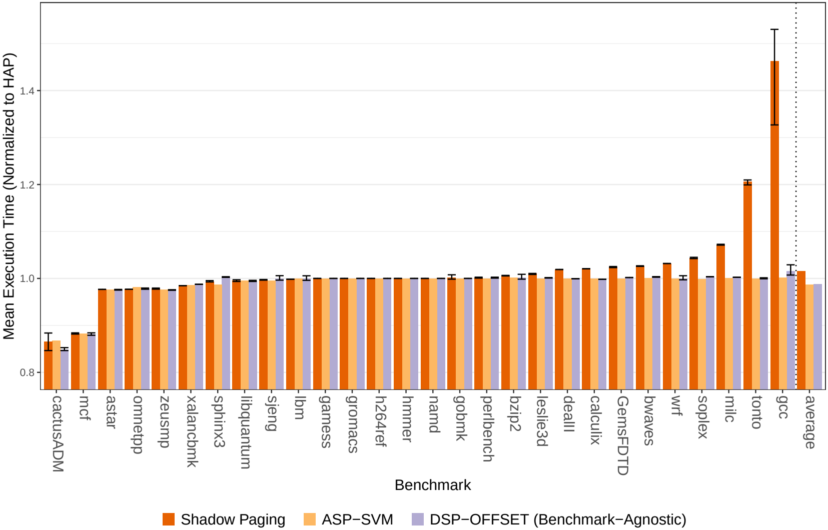
Direct Sampling is Insufficient



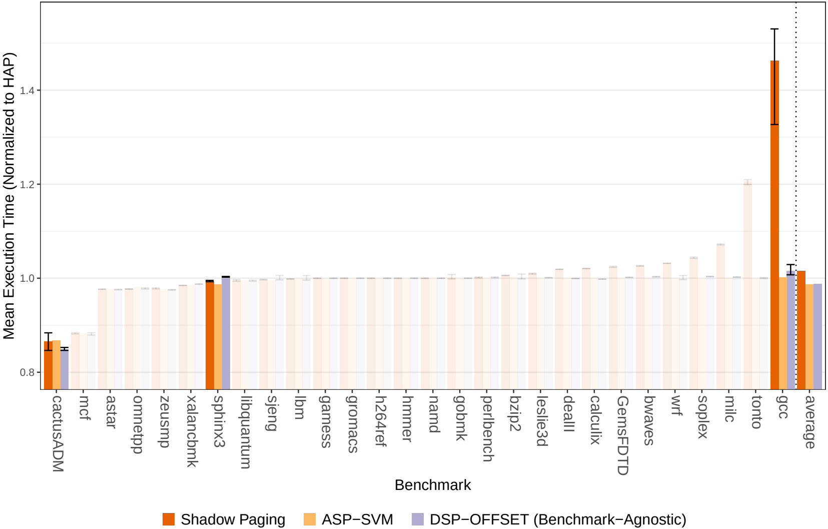
Direct Sampling is Insufficient



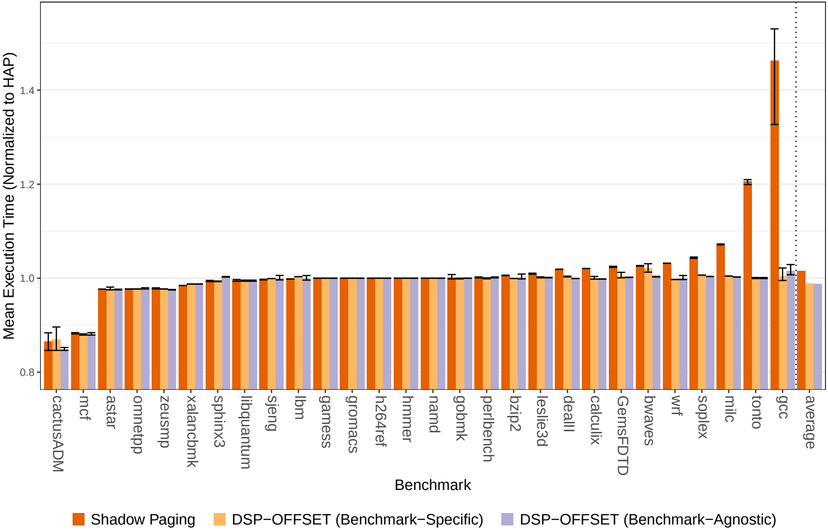
DSP-OFFSET Matches State-Of-The-Art



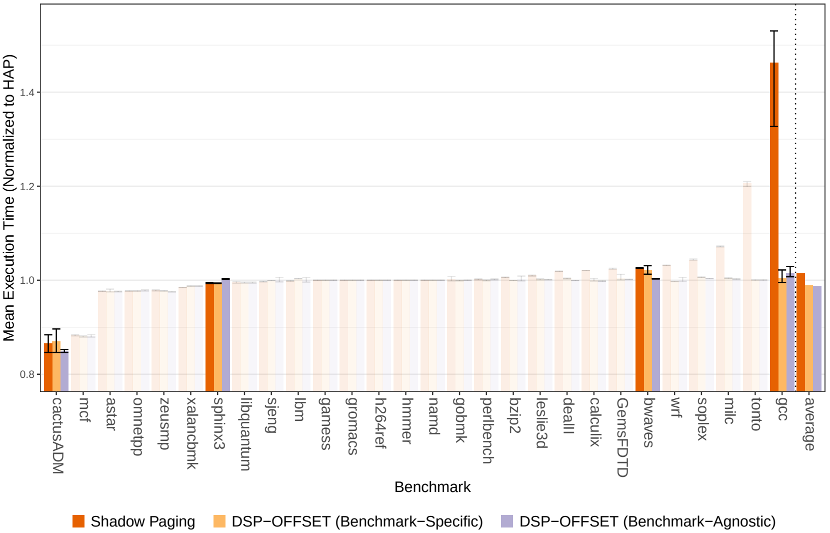
DSP-OFFSET Matches State-Of-The-Art



DSP-OFFSET Generalizes Well



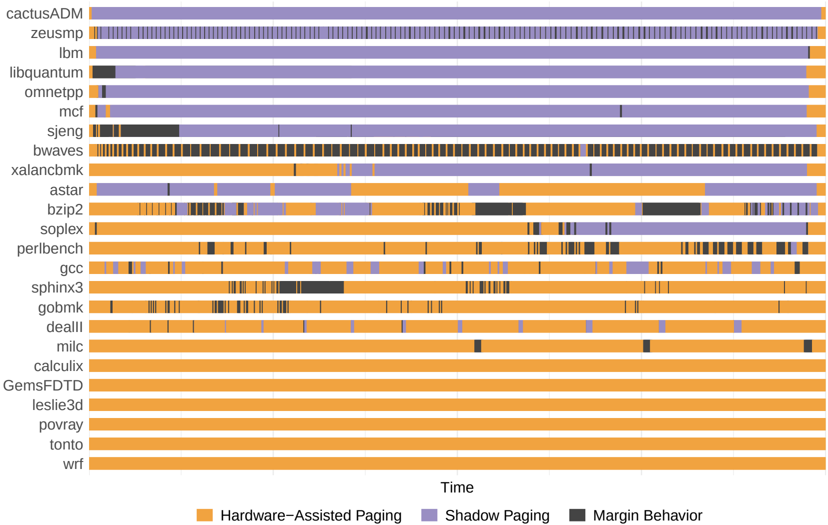
DSP-OFFSET Generalizes Well



DSP-OFFSET Reduces Profiling Time

	SPEC INT06		
Benchmark	**Executions**	**Profiling (h)**	**Samples**
ASP-SVM	~ 5	> 24.0	60
DSP-OFFSET	1	2.5	25000

Action Selection Indicates Program Behavior



Paging Mode Selection

Contextual Bandits

DSP-OFFSET

Evaluation

Conclusion

Conclusion

- Comparable performance vs state-of-the-art
- Over 90% reduction in profiling time vs state-of-the-art
- The contextual bandit is a viable selection model

Constructing Dynamic Policies for Paging Mode Selection

Jason Hiebel Laura E. Brown Zhenlin Wang

jshiebel@mtu.edu lebrown@mtu.edu zlwang@mtu.edu

Department of Computer Science
Michigan Technological University

International Conference on Parallel Processing
August 2018

References

[1] Alina Beygelzimer and John Langford.
The Offset Tree for Learning with Partial Labels.
KDD ’09, 2009.

[2] Wei Kuang, Laura E. Brown, and Zhenlin Wang.
Selective Switching Mechanism in Virtual Machines via Support Vector Machines and Transfer Learning.
Machine Learning, 101(1), 2015.

[3] Xiaolin Wang, Jiarui Zang, Zhenlin Wang, Yingwei Luo, and Xiaoming Li.
Selective Hardware/Software Memory Virtualization.
VEE ’11, 2011.

Future Directions

Extend the approach to similar problems (e.g. hardware prefetchers)

- Selecting contextual information
- Larger, combinatorial action spaces
- Multi-core, co-tenant workloads

Evaluation using Random Profiling

- DSP-OFFSET still requires **in situ** evaluation
- Alleviate cost of model validation