18-742: Computer Architecture & Systems

Back to the Future: Leveraging Belady's Algorithm for Improved Cache Replacement

Akanksha Jain and Calvin Lin

Presented by Mitchell Fream

Spring 2025, Lecture 24

Cache Replacement

- The cache has a limited capacity

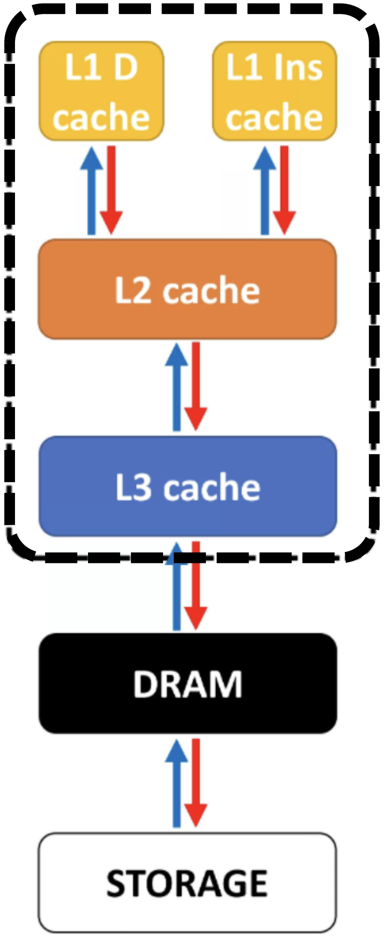
- We have to “evict” some items from the cache to make room for others

- Cache replacement: Which item to evict?

Cache Replacement in Computer Systems

- Whole throughout the hierarchy

- Hardware managed and software managed



What Is The Best Algorithm?

- **There is no one-fits-all algorithm**
  - Different cache layers, different applications, different configurations favor different algorithms

- **For hardware caches, LRU is the de facto algorithm**
  - Why it works?

Belady's Algorithm

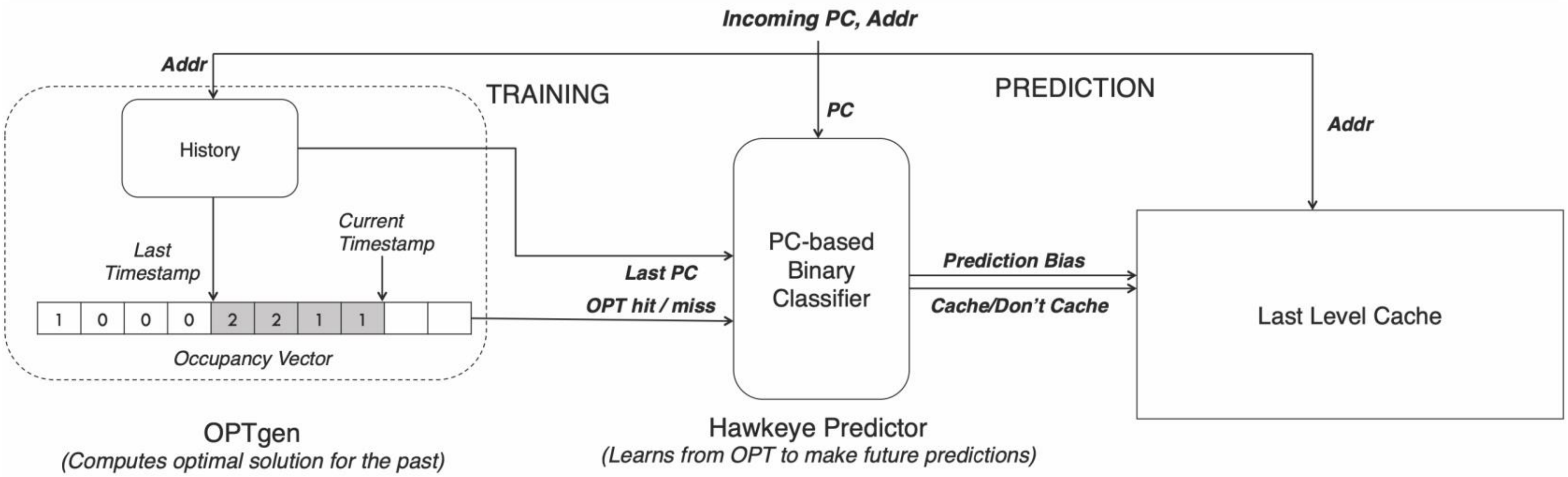
- **If the goal is maximizing hit ratio, Belady's algorithm is the optimal cache replacement**

- **The algorithm: evict the block which will be used later than the others in future**
  - Not practical ☹

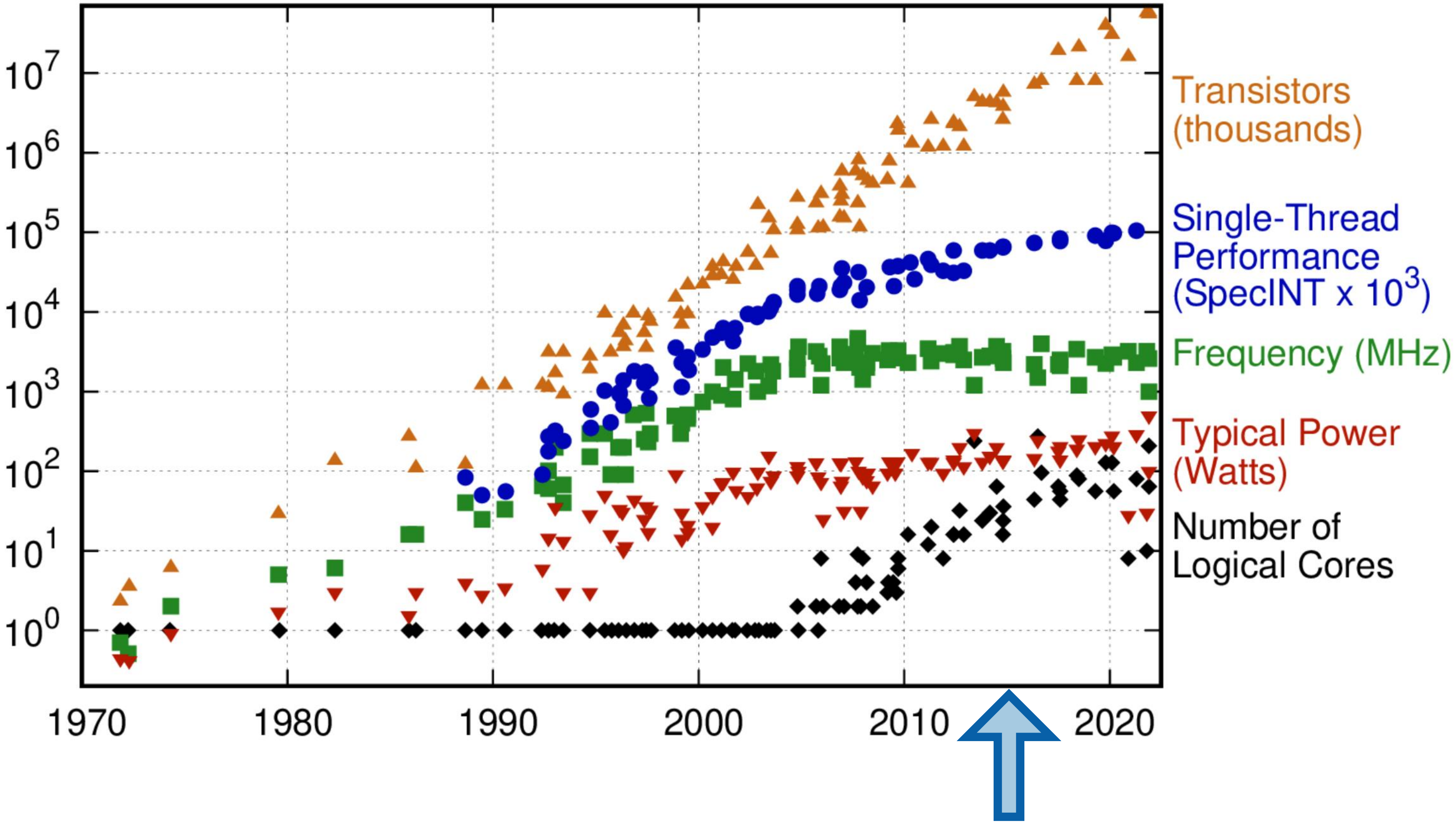
“Back to the Future: Leveraging Belady’s Algorithm for Improved Cache Replacement”

Akanksha Jain, Calvin Lin 2016

- **A novel cache replacement policy**
- **A practical implementation inspired by the Belady’s algorithm**

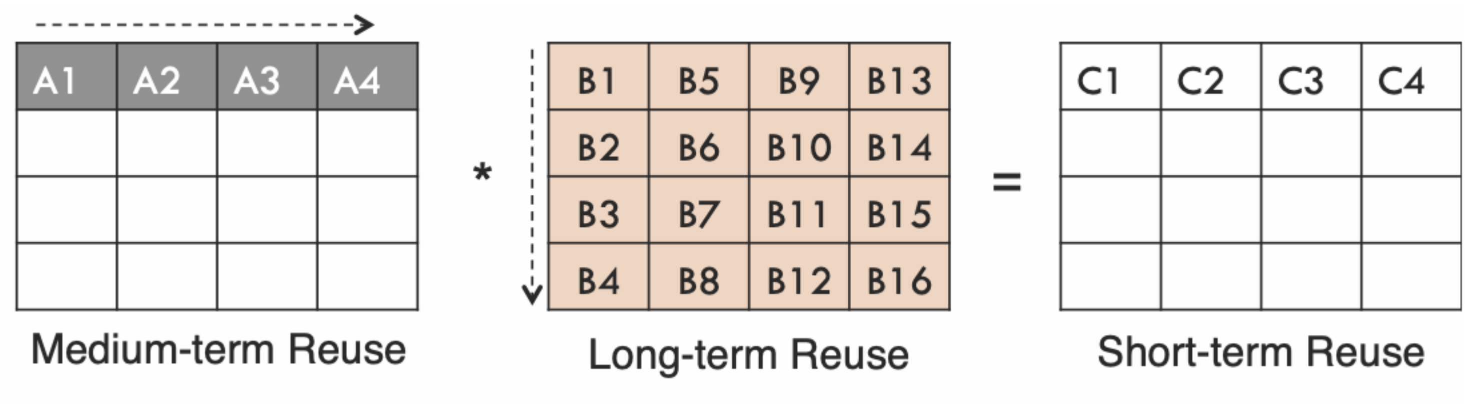


50 Years of Microprocessor Trend Data



We are here

Why LRU Is Not The Best?

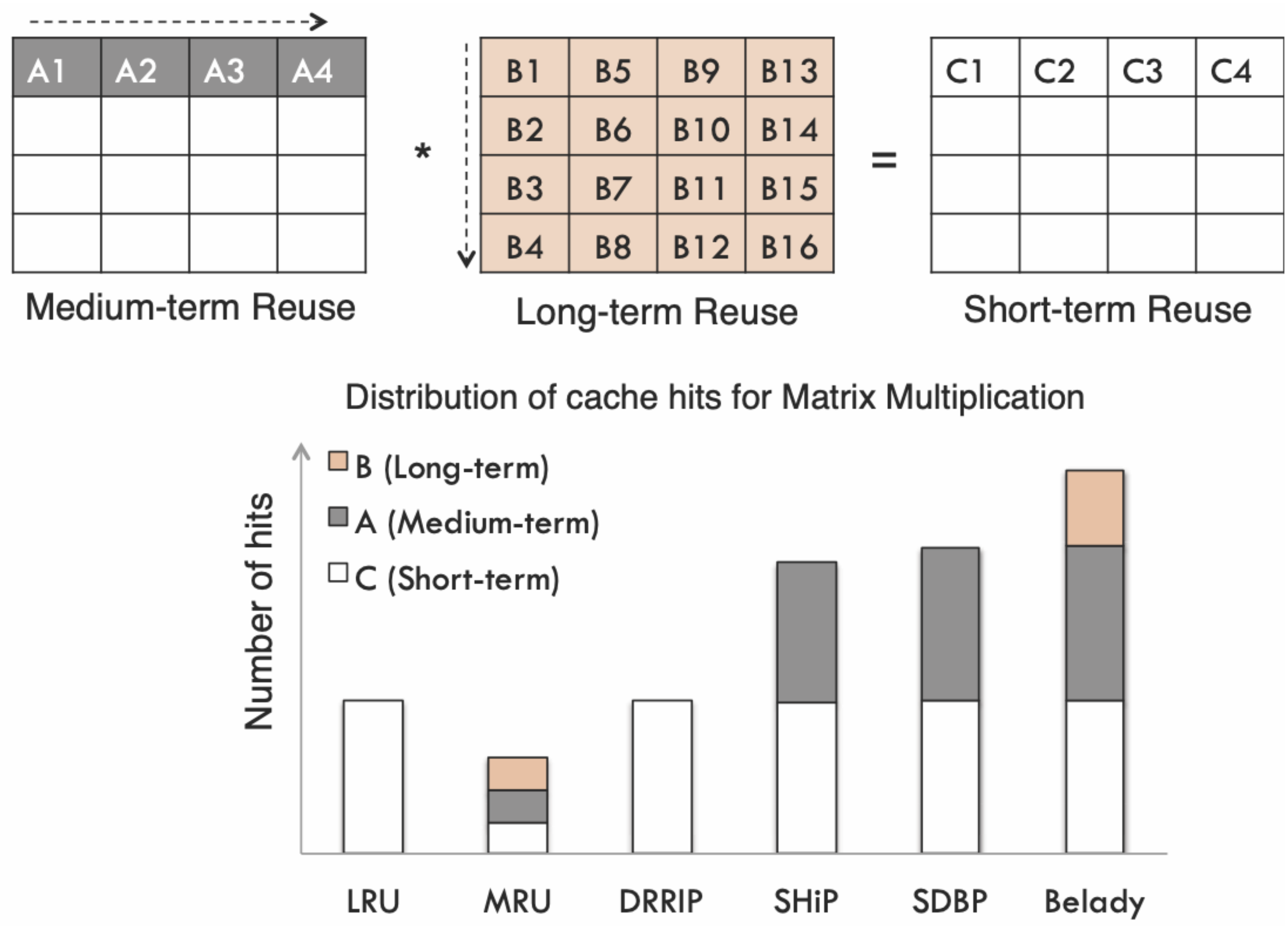


Medium-term Reuse

Long-term Reuse

Short-term Reuse

Why LRU Is Not The Best?



This Paper

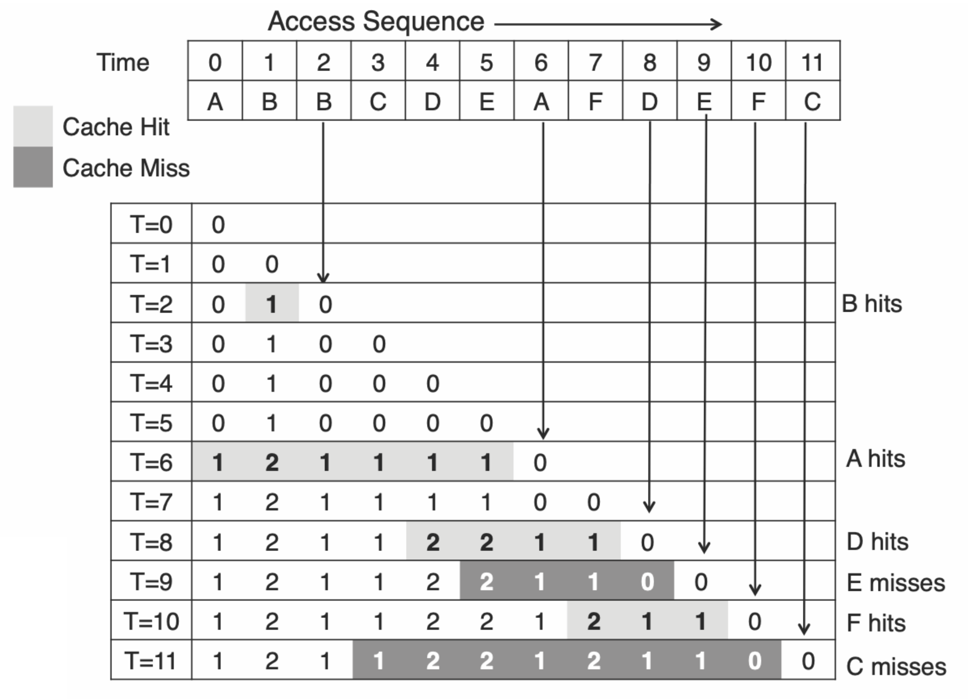
- **Try to learn Belady’s algorithm and mimic its behavior**
  - A practical approximation for an impractical algorithm

- **Back to the future?**

This Paper

- **Try to learn Belady's algorithm and mimic its behavior**
  - A practical approximation for an impractical algorithm

- **Back to the future?**
  - Yes



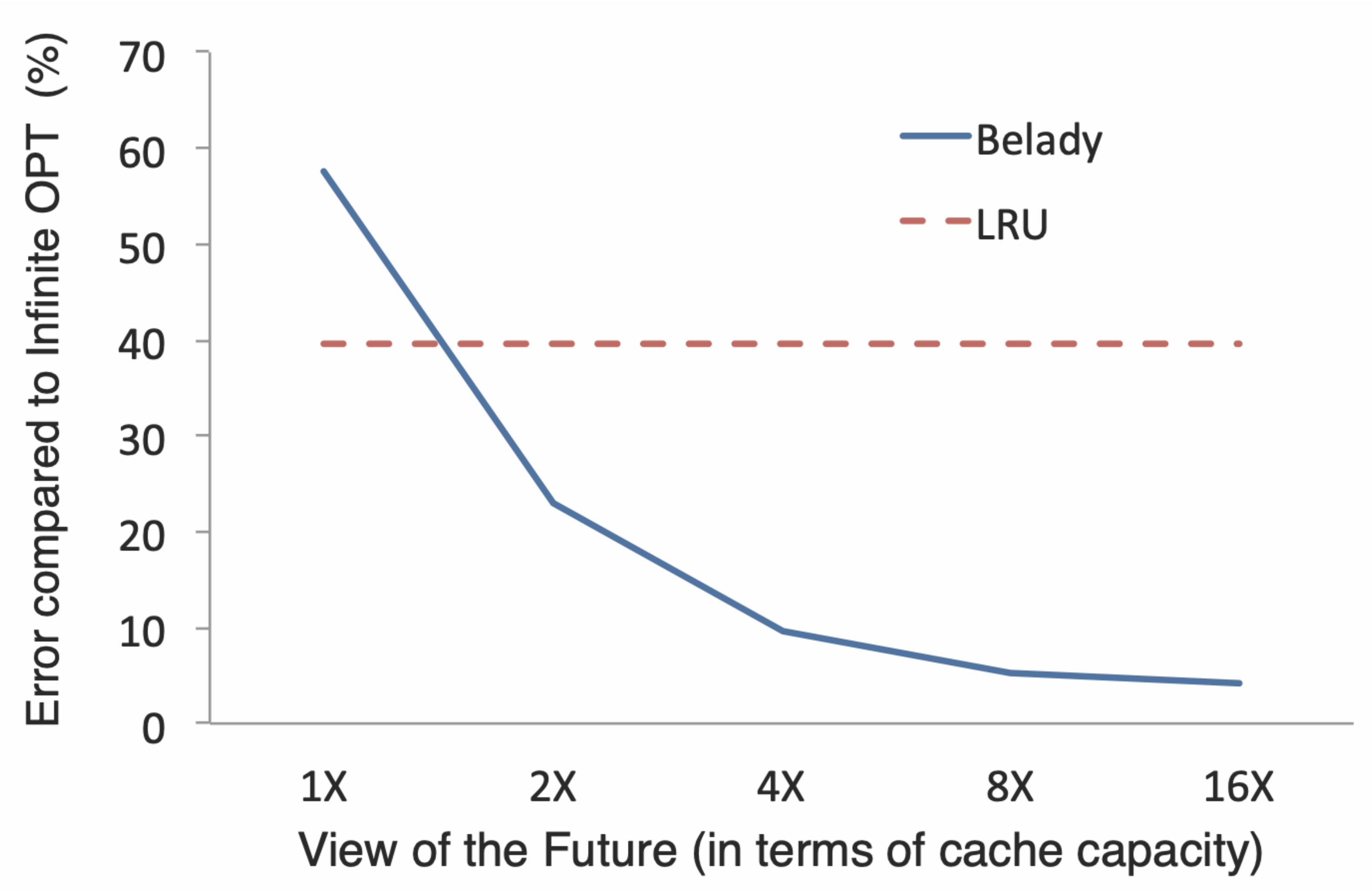
Access Sequence →

░ Cache Hit

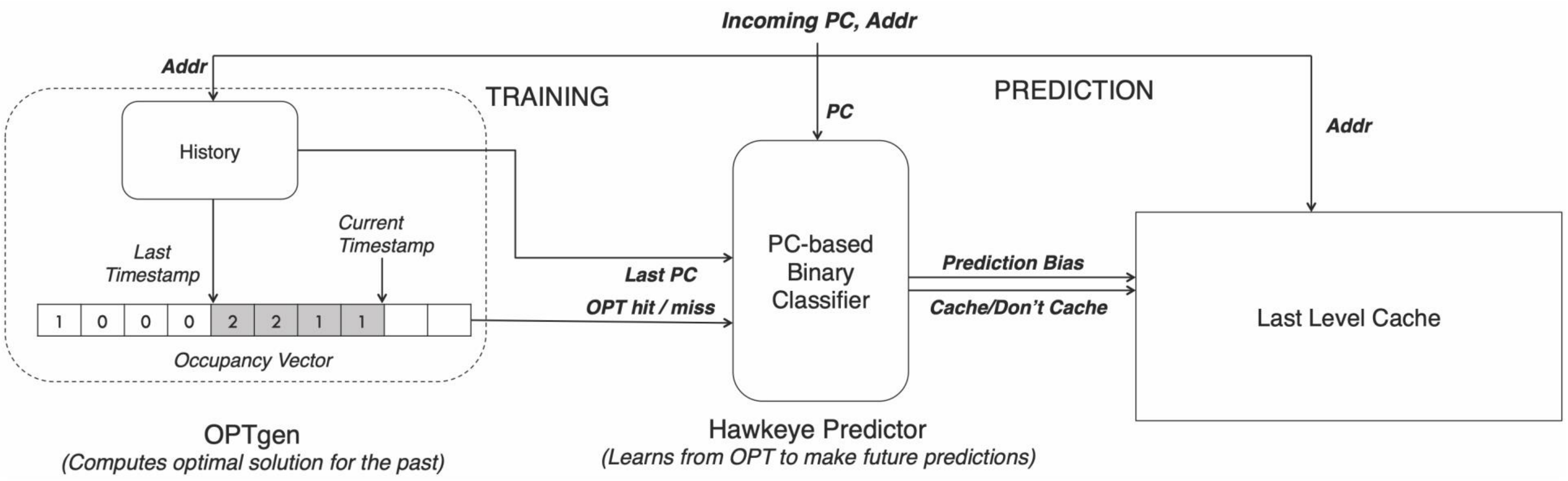
▓ Cache Miss

Time	0	1	2	3	4	5	6	7	8	9	10	11	
	A	B	B	C	D	E	A	F	D	E	F	C	
T=0	0												
T=1	0	0											
T=2	0	**1**	0										B hits
T=3	0	1	0	0									
T=4	0	1	0	0	0								
T=5	0	1	0	0	0	0							
T=6	**1**	**2**	**1**	**1**	**1**	**1**	0						A hits
T=7	1	2	1	1	1	1	0	0					
T=8	1	2	1	1	**2**	**2**	**1**	**1**	0				D hits
T=9	1	2	1	1	2	**2**	**1**	**1**	**0**	0			E misses
T=10	1	2	1	1	2	2	1	**2**	**1**	**1**	0		F hits
T=11	1	2	1	**1**	**2**	**2**	**1**	**2**	**1**	**1**	**0**	0	C misses

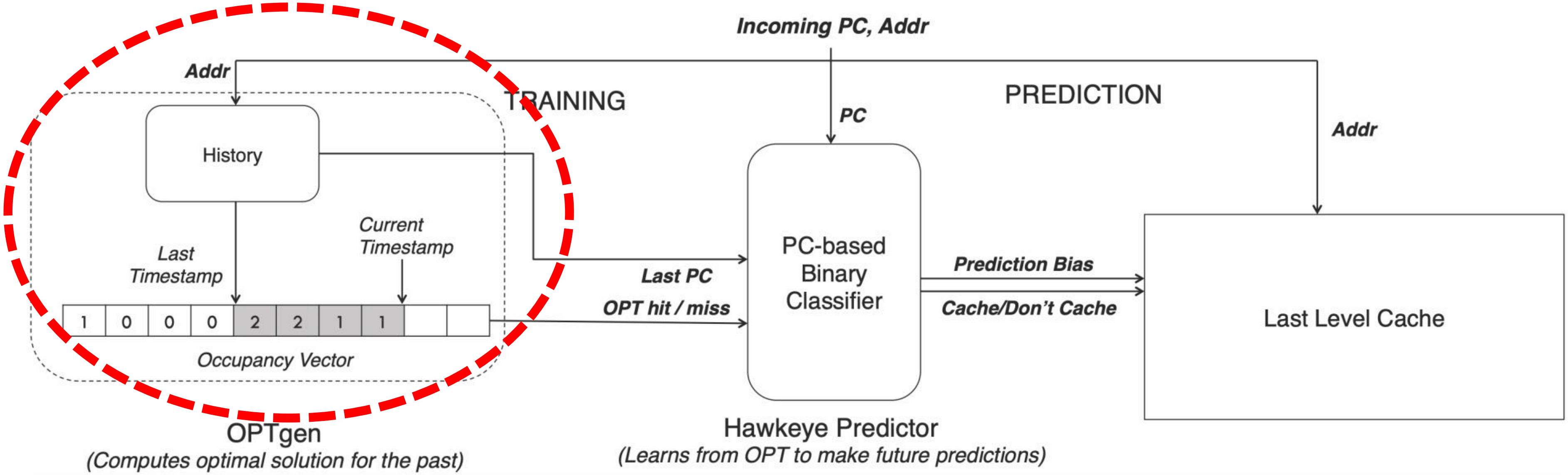
How Far Ahead Is Enough?



The Design


Incoming PC, Addr
Addr
TRAINING
PC
PREDICTION
Addr
History
Current Timestamp
Last Timestamp
1
0
0
0
2
2
1
1
Occupancy Vector
Last PC
OPT hit / miss
PC-based Binary Classifier
Prediction Bias
Cache/Don't Cache
Last Level Cache
OPTgen
(Computes optimal solution for the past)
Hawkeye Predictor
(Learns from OPT to make future predictions)

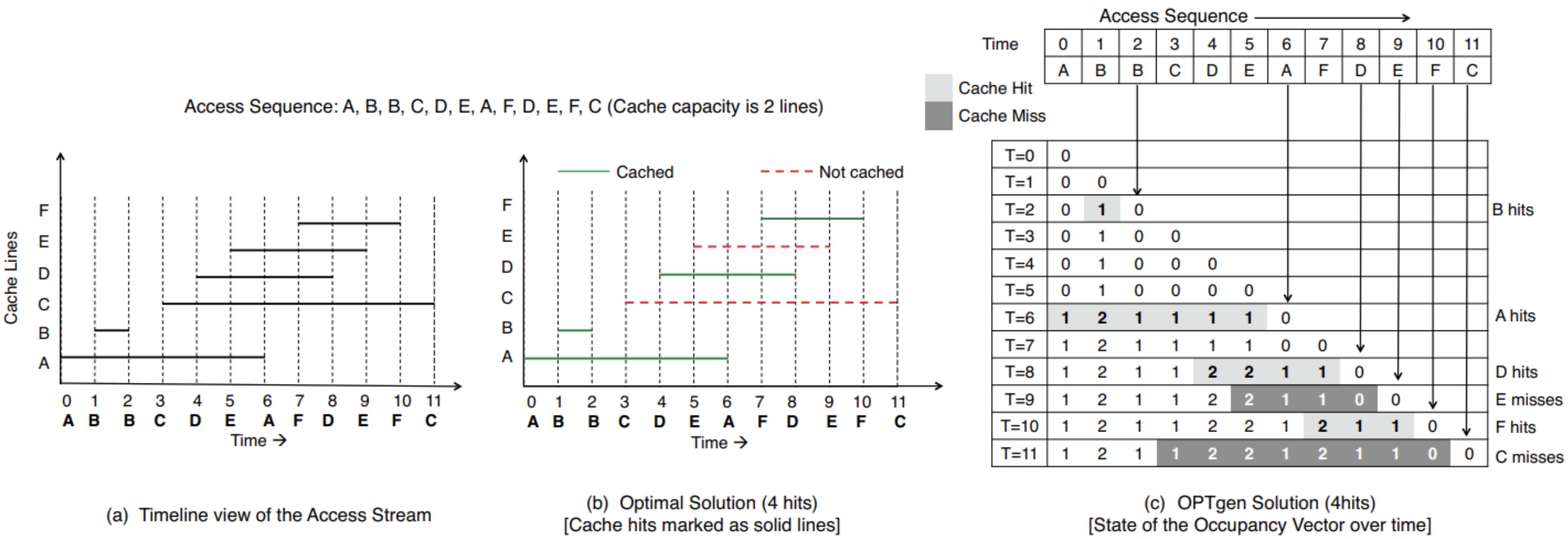
The Design


Incoming PC, Addr
Addr
TRAINING
PREDICTION
PC
Addr
History
Current Timestamp
Last Timestamp
1
0
0
0
2
2
1
1
Occupancy Vector
Last PC
OPT hit / miss
PC-based Binary Classifier
Prediction Bias
Cache/Don't Cache
Last Level Cache
OPTgen
(Computes optimal solution for the past)
Hawkeye Predictor
(Learns from OPT to make future predictions)

The Design

- OPTgen determines if the next access to the same address is likely to be a hit

- Look back at previous accesses to see reuse distance



Time	0	1	2	3	4	5	6	7	8	9	10	11	
	A	B	B	C	D	E	A	F	D	E	F	C	
T=0	0												
T=1	0	0											
T=2	0	1	0										B hits
T=3	0	1	0	0									
T=4	0	1	0	0	0								
T=5	0	1	0	0	0	0							
T=6	1	2	1	1	1	1	0						A hits
T=7	1	2	1	1	1	1	0	0					
T=8	1	2	1	1	2	2	1	1	0				D hits
T=9	1	2	1	1	2	2	1	1	0	0			E misses
T=10	1	2	1	1	2	2	1	2	1	1	0		F hits
T=11	1	2	1	1	2	2	1	2	1	1	0	0	C misses

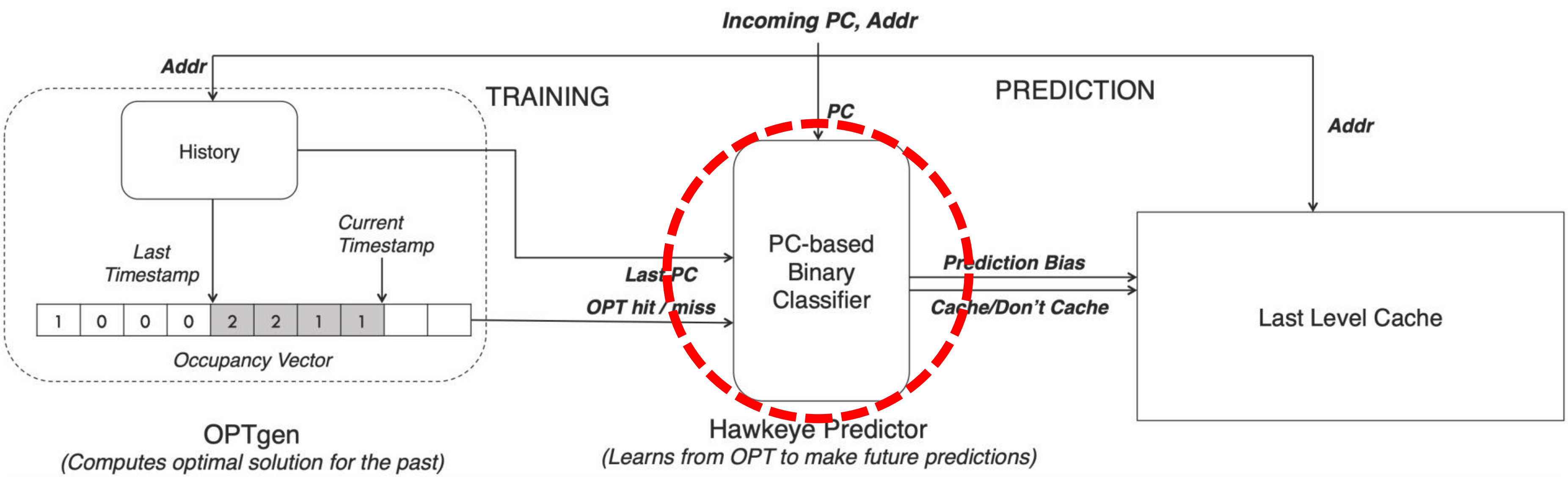
(a) Timeline view of the Access Stream

(b) Optimal Solution (4 hits)
[Cache hits marked as solid lines]

(c) OPTgen Solution (4hits)
[State of the Occupancy Vector over time]

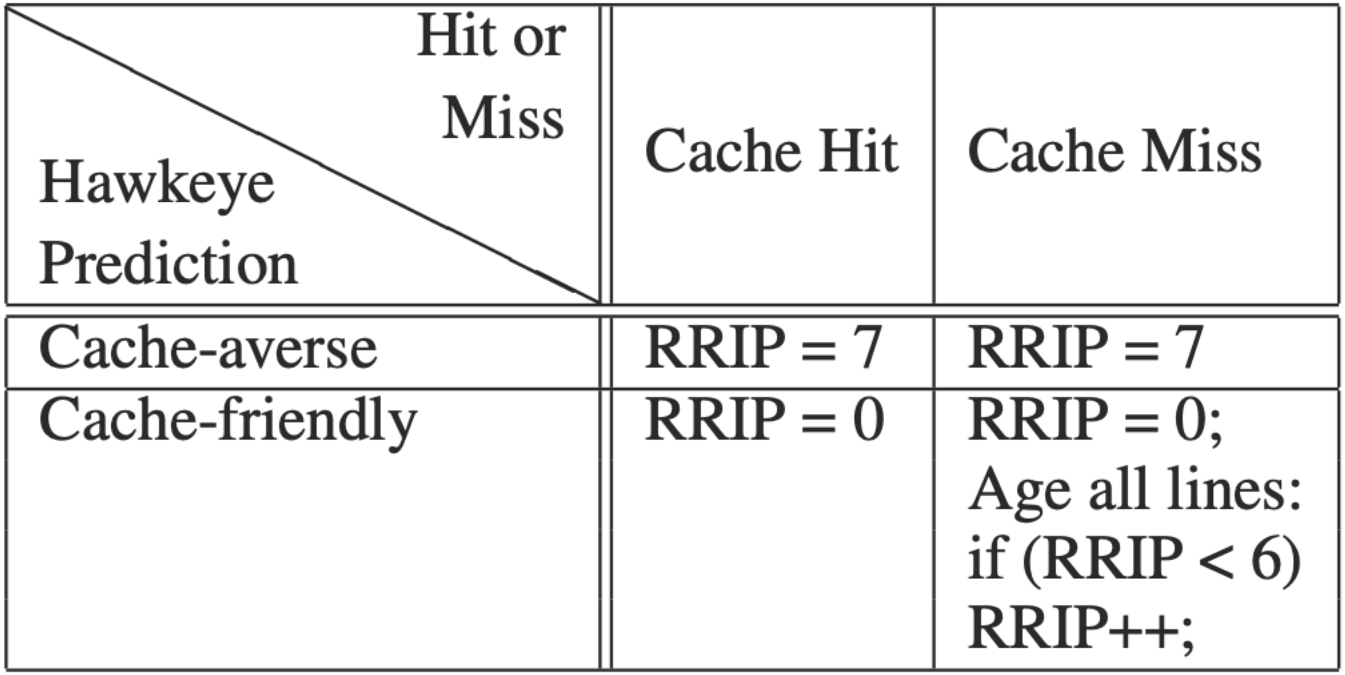
Figure 6: Example to illustrate OPTgen.

The Design



Incoming PC, Addr
Addr
TRAINING
PC
PREDICTION
Addr
History
Current Timestamp
Last Timestamp
Last PC
OPT hit / miss
PC-based Binary Classifier
Prediction Bias
Cache/Don't Cache
Last Level Cache
1 0 0 0 2 2 1 1
Occupancy Vector
OPTgen
(Computes optimal solution for the past)
Hawkeye Predictor
(Learns from OPT to make future predictions)

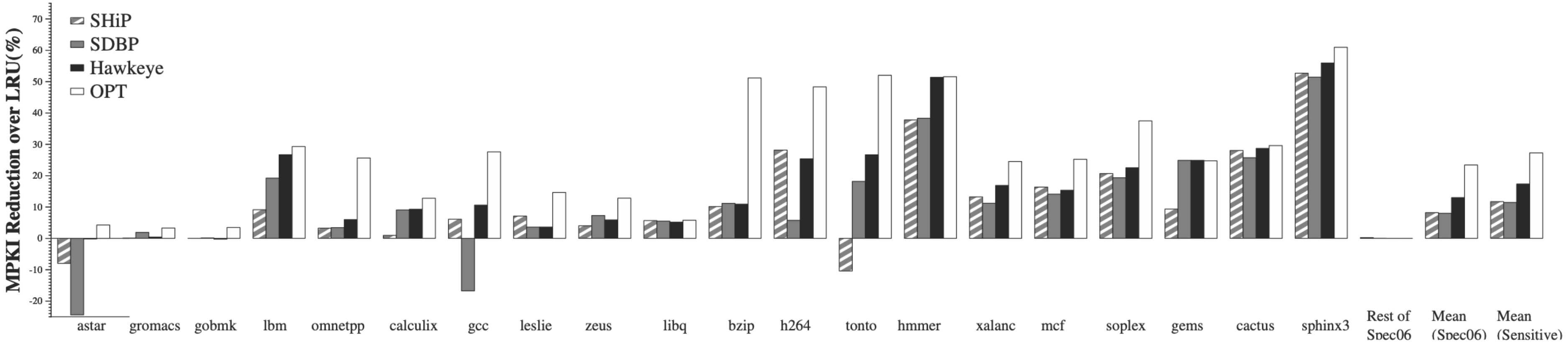
The Design



Hit or Miss Hawkeye Prediction	Cache Hit	Cache Miss
Cache-averse	RRIP = 7	RRIP = 7

Cache-friendly	RRIP = 0	RRIP = 0;
		Age all lines:
		if (RRIP < 6)
		RRIP++;

Miss Reduction



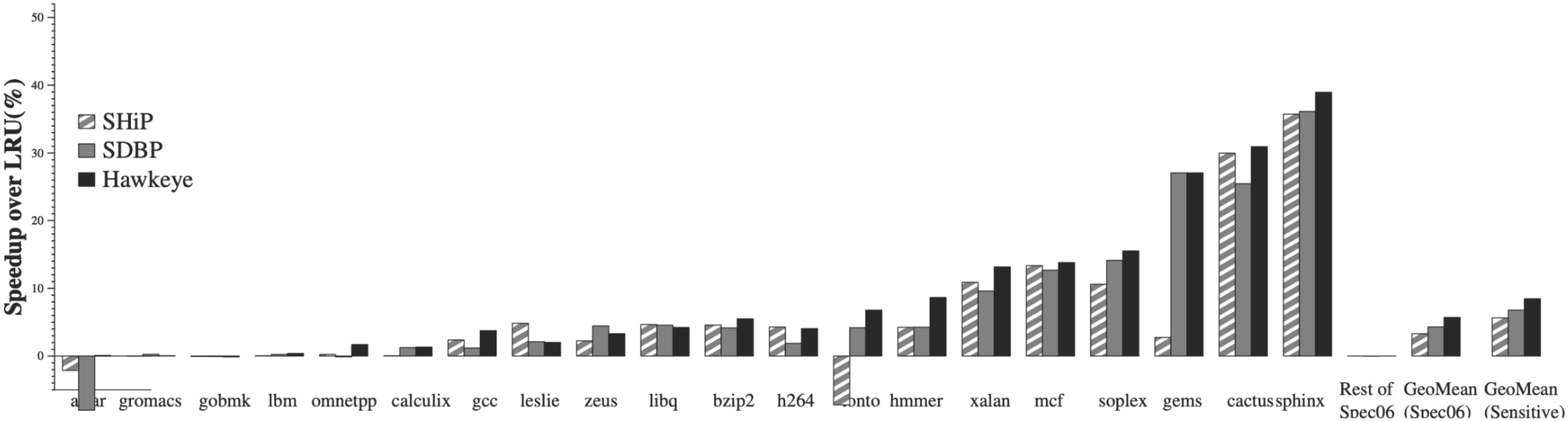
Discussion: Summary Question #1

What Did the Paper Get Right?

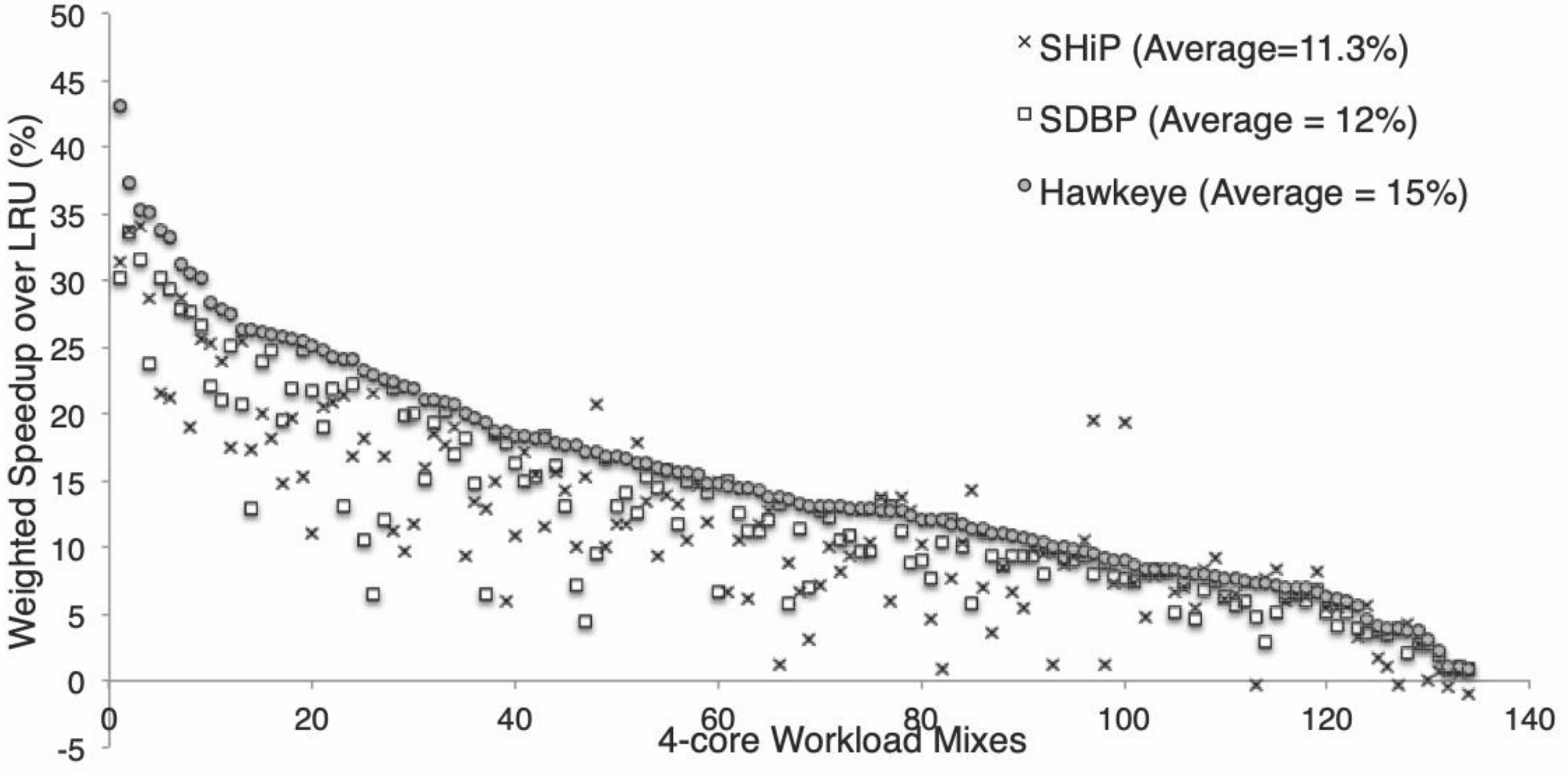
State the 3 most important things the paper says.

These could be some combination of the motivations, observations, interesting parts of the design, or clever parts of the implementation.

Performance Improvement



Performance Improvement



Discussion: Summary Question #2

What Did the Paper Get Wrong?

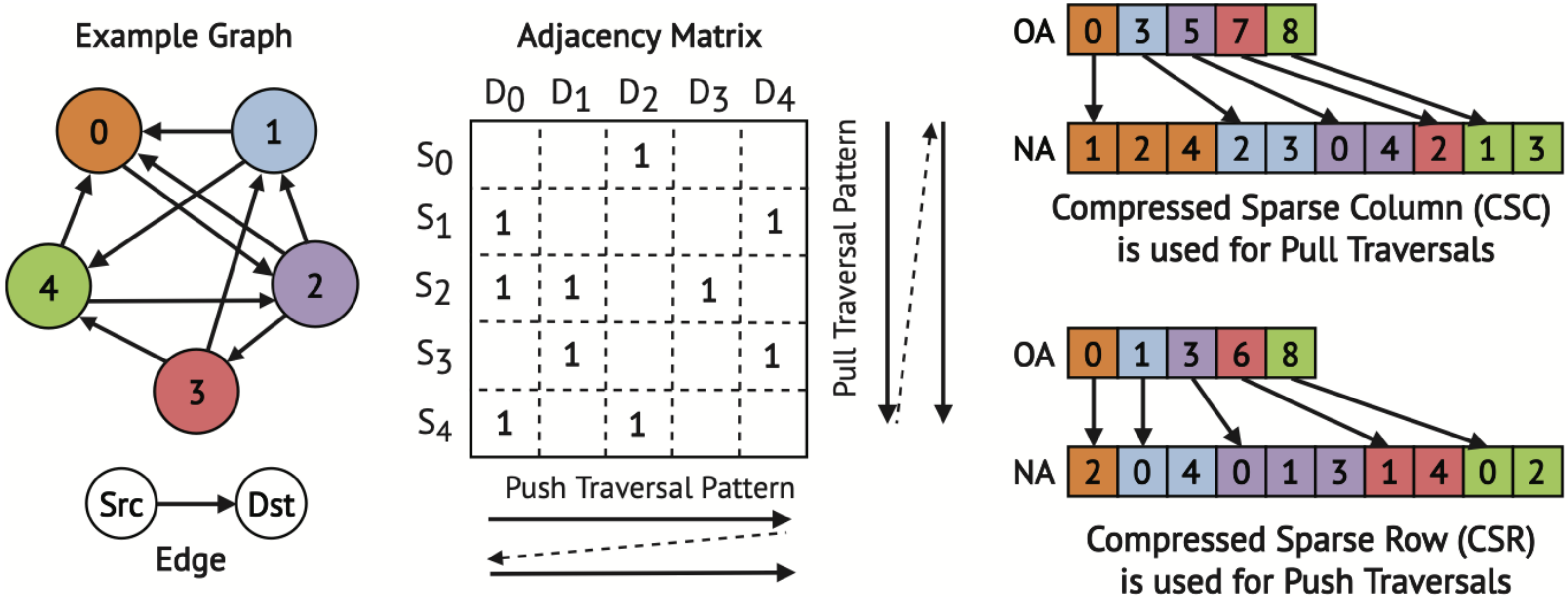
Describe the paper's single most glaring deficiency.

Every paper has some fault. Perhaps an experiment was poorly designed or the main idea had a narrow scope or applicability.

"P-OPT: Practical Optimal Cache Replacement for Graph Analytics"

Vignesh Balaji, Neal Crago, Aamer Jaleel, Brandon Lucia 2021

- **A replacement policy for graph analytics**



Adjacency Matrix

	D_0	D_1	D_2	D_3	D_4
S_0			1		
S_1	1				1
S_2	1	1		1	
S_3		1			1
S_4	1		1		

OA	0	3	5	7	8					
NA	1	2	4	2	3	0	4	2	1	3

Compressed Sparse Column (CSC) is used for Pull Traversals

OA	0	1	3	6	8					
NA	2	0	4	0	1	3	1	4	0	2

Compressed Sparse Row (CSR) is used for Push Traversals

Cache Replacement Policy Now

- **Still an active area**
  - Especially in industry!

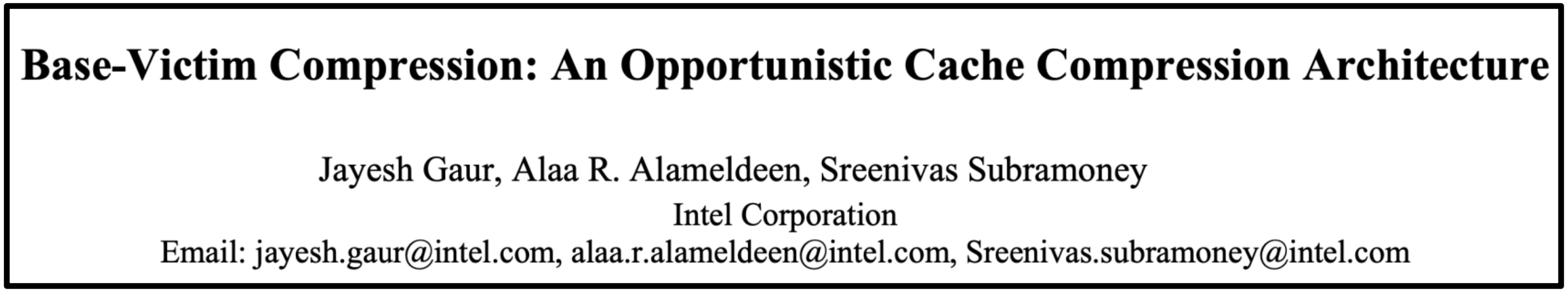
- **Modern processors typically use sophisticated replacement policies beyond LRU for their last-level caches**
  - Mostly undocumented

Is Belady's Algorithm Always The Best?

- What if the cache is compressed?

Is Belady's Algorithm Always The Best?

- What if the cache is compressed?

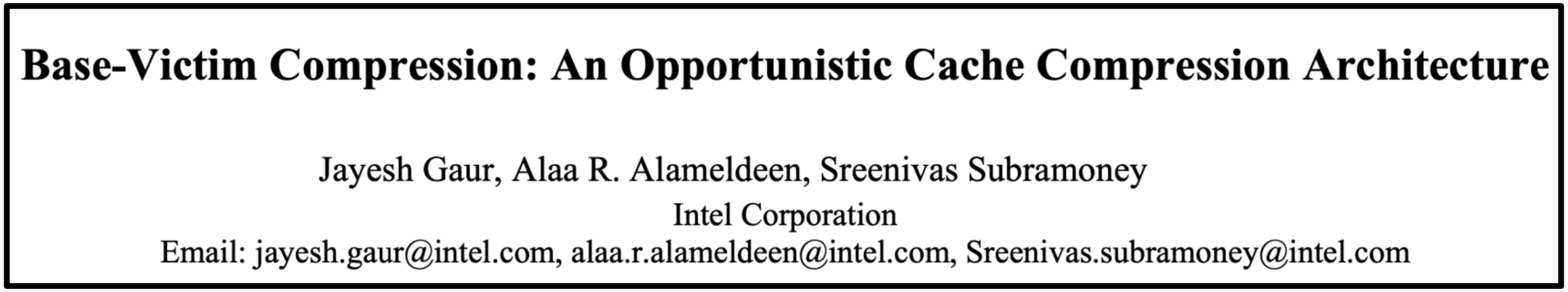


Base-Victim Compression: An Opportunistic Cache Compression Architecture

Jayesh Gaur, Alaa R. Alameldeen, Sreenivas Subramoney
Intel Corporation
Email: jayesh.gaur@intel.com, alaa.r.alameldeen@intel.com, Sreenivas.subramoney@intel.com

Is Belady's Algorithm Always The Best?

- **What if the cache is compressed?**



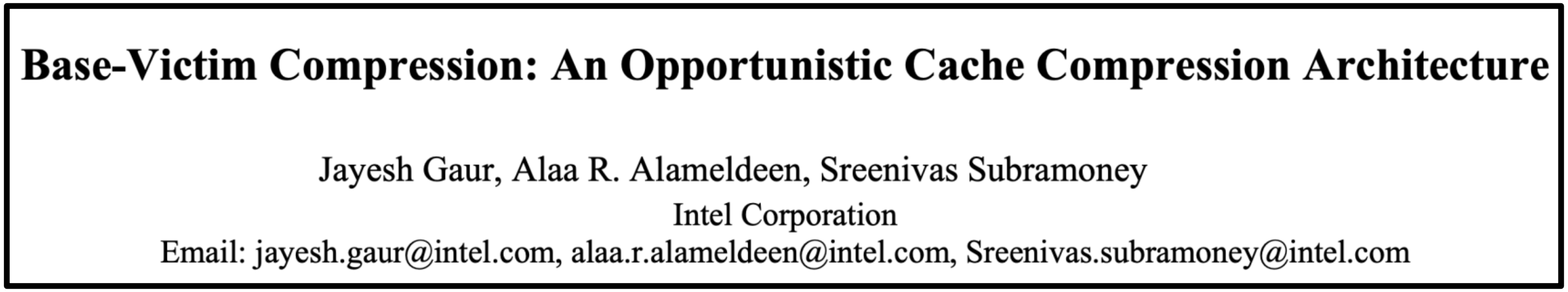
Base-Victim Compression: An Opportunistic Cache Compression Architecture

Jayesh Gaur, Alaa R. Alameldeen, Sreenivas Subramoney
Intel Corporation
Email: jayesh.gaur@intel.com, alaa.r.alameldeen@intel.com, Sreenivas.subramoney@intel.com

- **Are all cache blocks equally important for performance?**

Is Belady's Algorithm Always The Best?

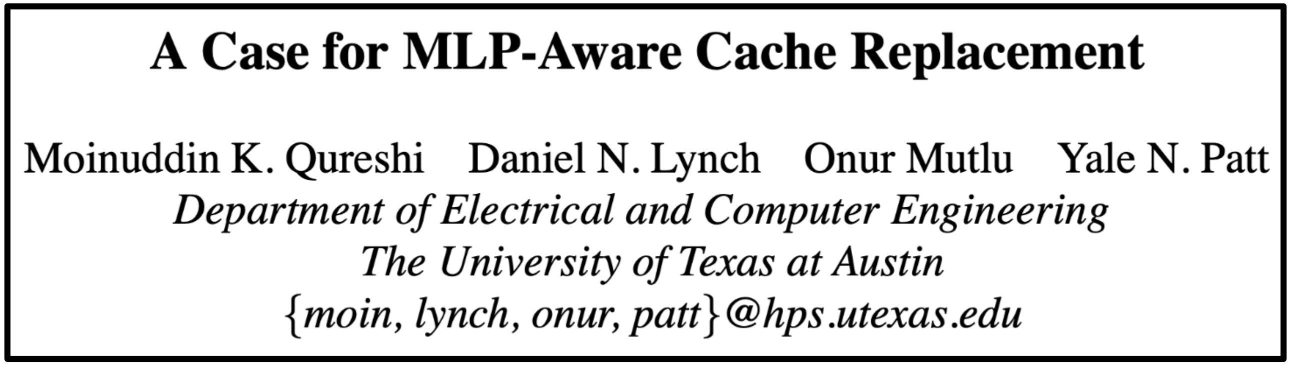
- What if the cache is compressed?



Base-Victim Compression: An Opportunistic Cache Compression Architecture

Jayesh Gaur, Alaa R. Alameldeen, Sreenivas Subramoney
Intel Corporation
Email: jayesh.gaur@intel.com, alaa.r.alameldeen@intel.com, Sreenivas.subramoney@intel.com

- Are all cache blocks equally important for performance?



A Case for MLP-Aware Cache Replacement

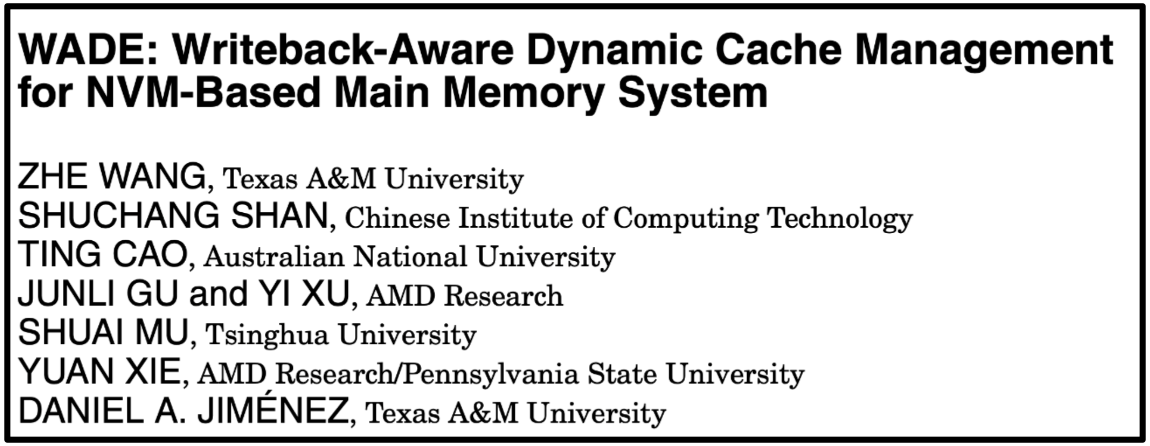
Moinuddin K. Qureshi   Daniel N. Lynch   Onur Mutlu   Yale N. Patt
Department of Electrical and Computer Engineering
The University of Texas at Austin
{*moin, lynch, onur, patt*}@*hps.utexas.edu*

Is Belady's Algorithm Always The Best?

- What if the backing memory is non-volatile?

Is Belady’s Algorithm Always The Best?

- **What if the backing memory is non-volatile?**

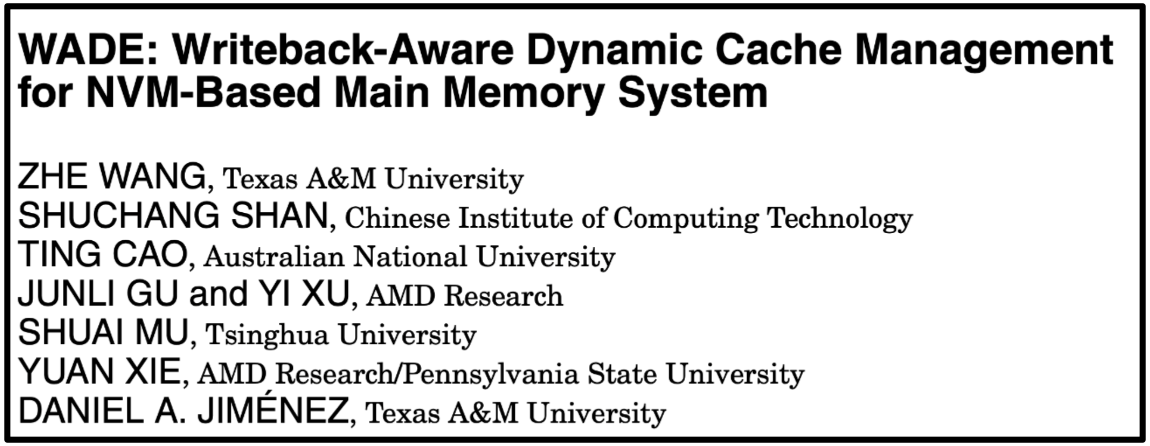


WADE: Writeback-Aware Dynamic Cache Management for NVM-Based Main Memory System

ZHE WANG, Texas A&M University
SHUCHANG SHAN, Chinese Institute of Computing Technology
TING CAO, Australian National University
JUNLI GU and YI XU, AMD Research
SHUAI MU, Tsinghua University
YUAN XIE, AMD Research/Pennsylvania State University
DANIEL A. JIMÉNEZ, Texas A&M University

Is Belady's Algorithm Always The Best?

- What if the backing memory is non-volatile?



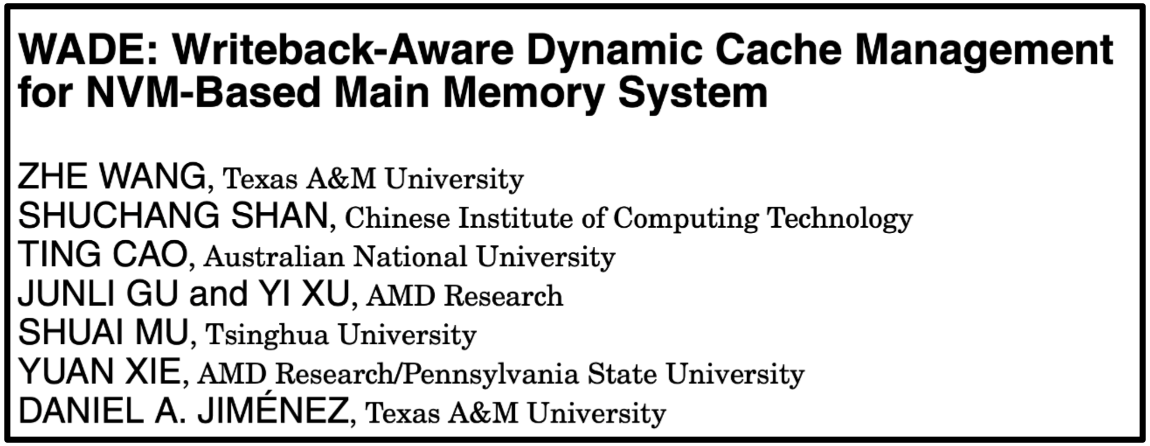
WADE: Writeback-Aware Dynamic Cache Management for NVM-Based Main Memory System

ZHE WANG, Texas A&M University
SHUCHANG SHAN, Chinese Institute of Computing Technology
TING CAO, Australian National University
JUNLI GU and YI XU, AMD Research
SHUAI MU, Tsinghua University
YUAN XIE, AMD Research/Pennsylvania State University
DANIEL A. JIMÉNEZ, Texas A&M University

- Is performance the only metric?

Is Belady's Algorithm Always The Best?

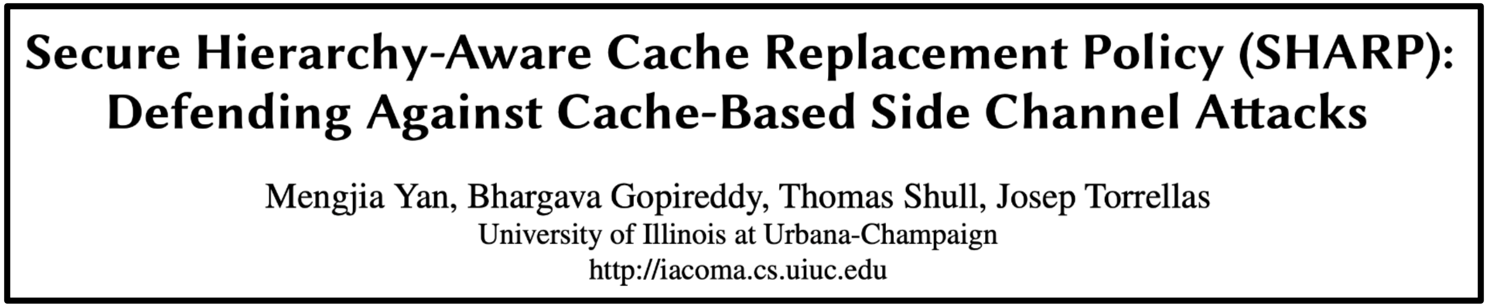
- What if the backing memory is non-volatile?



WADE: Writeback-Aware Dynamic Cache Management for NVM-Based Main Memory System

ZHE WANG, Texas A&M University
SHUCHANG SHAN, Chinese Institute of Computing Technology
TING CAO, Australian National University
JUNLI GU and YI XU, AMD Research
SHUAI MU, Tsinghua University
YUAN XIE, AMD Research/Pennsylvania State University
DANIEL A. JIMÉNEZ, Texas A&M University

- Is performance the only metric?



Secure Hierarchy-Aware Cache Replacement Policy (SHARP): Defending Against Cache-Based Side Channel Attacks

Mengjia Yan, Bhargava Gopireddy, Thomas Shull, Josep Torrellas
University of Illinois at Urbana-Champaign
http://iacoma.cs.uiuc.edu

To Read for Wednesday

“A New Case for the TAGE Branch Predictor”

Andre Seznec 2011

Optional Further Reading:

“BranchNet: A Convolutional Neural Network to Predict Hard-to-Predict Branches”

Siavash Zangeneh, Stephen Pruett, Sangkug Lym, Yale Patt 2020