FIU
Computing &
Information Sciences

LEARNING CACHE REPLACEMENT WITH CACHEUS

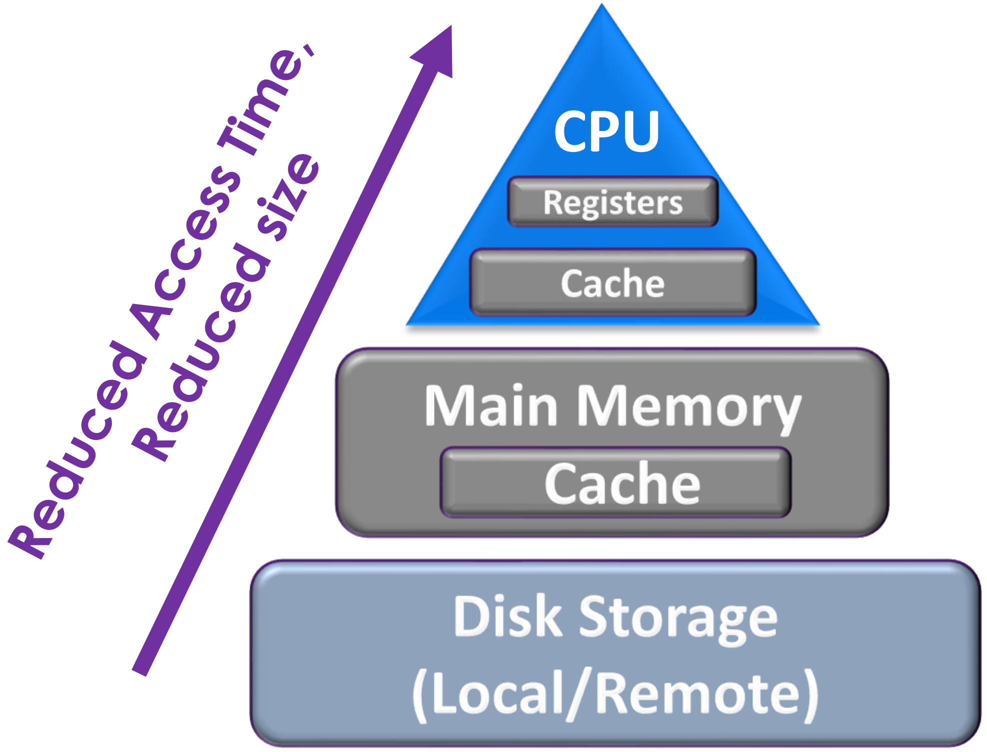
Farzana Beente Yusuf

Liana Rodriguez, Steven Lyons, Eysler Paz, Raju Rangaswami, Jason Liu, Giri Narasimhan, Florida International University

Ming Zhao, Arizona State University

Introduction

- Cache
  - Fast but relatively small in capacity



- Machine Learning
  - Improves decision processes

Cache management + ML

↓

Improved performance

Cache Replacement Algorithms

- Non-adaptive
  - Least Recently Used (LRU)
  - Least Frequently Used (LFU)
  - Low Inter-reference Recency Set (LIRS) [Jiang et al., ‘02]
- Adaptive
  - Adaptive Replacement Cache (ARC) [Modha, Megiddo, ‘02]
  - Dynamic LIRS (DLIRS) [Li, ‘18]
- ML-based Adaptive
  - Adaptive Caching Using Multiple Experts (ACME) [Ari et al., ‘02]
  - Reinforcement **Le**arning On **Ca**che **R**eplacement (LeCaR) [Vietri et al., ‘18]
  - Reinforcement **Le**arning (Cacheus) [This Work]

WORKLOAD PRIMITIVES

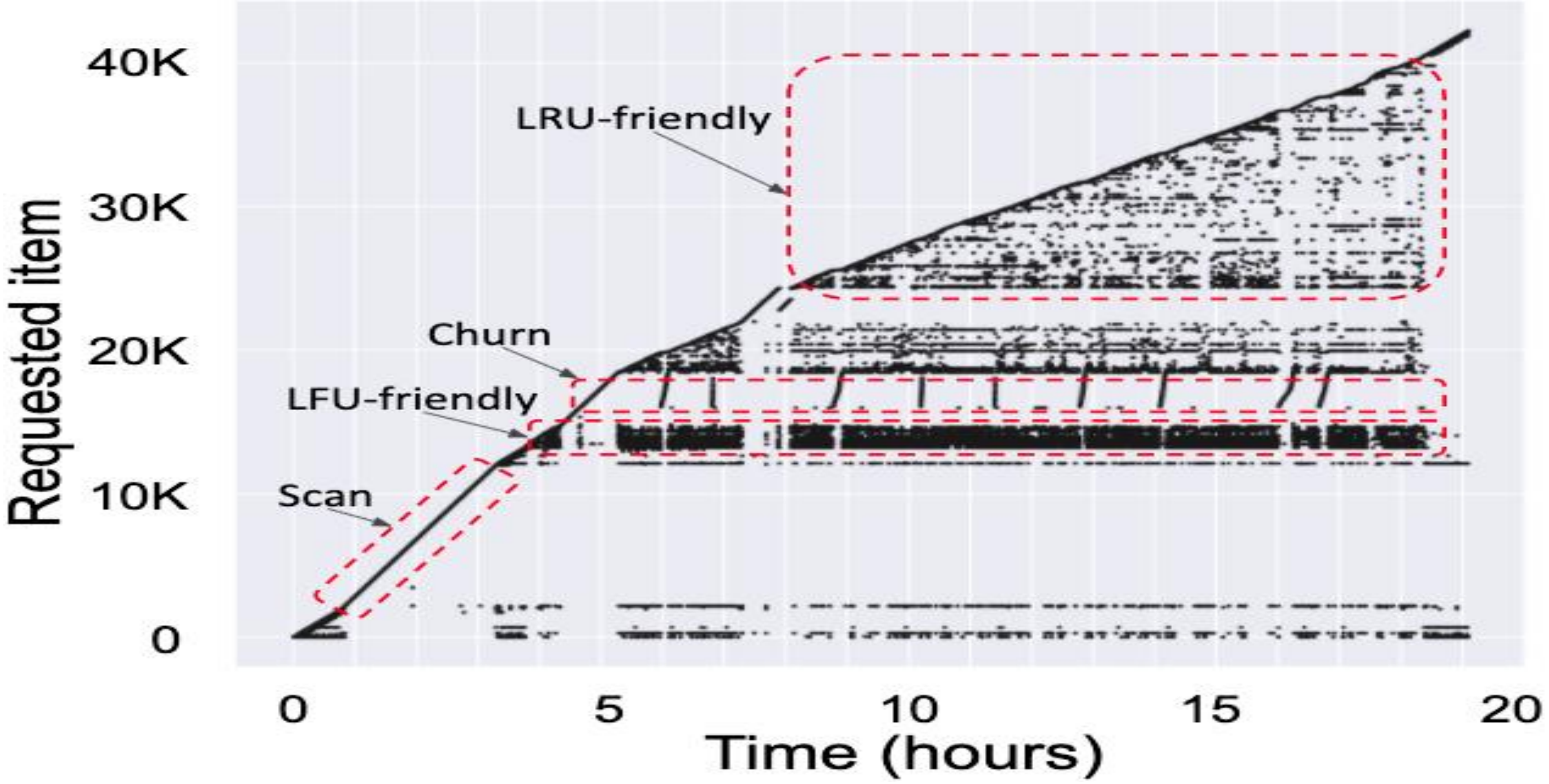
LRU-Friendly

LFU-Friendly

Churn

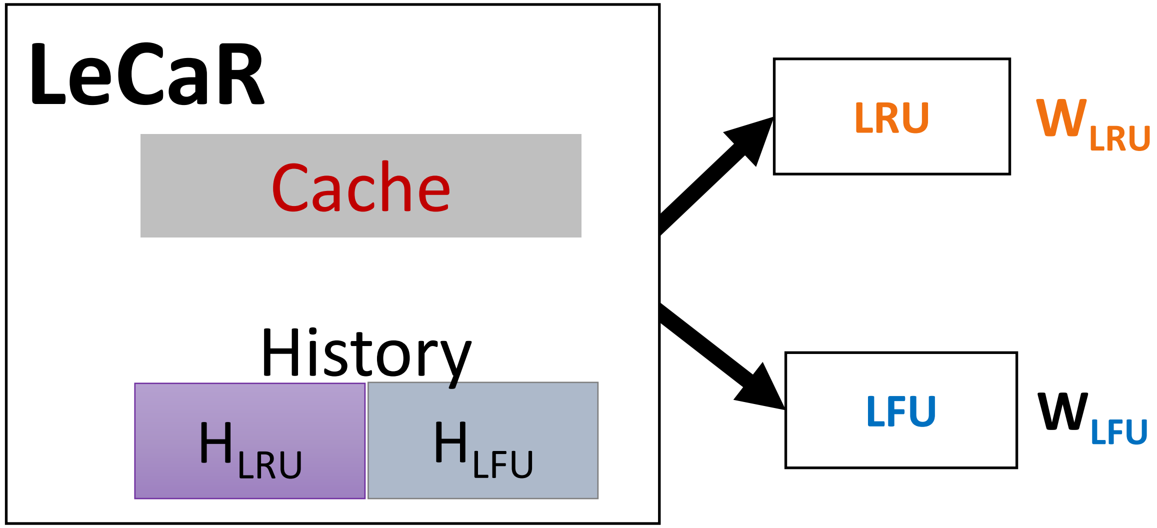
Scan

Algorithm	LRU	LFU	Churn	Scan
ARC	✓	✗	✗	✓
LIRS	✗	✗	✗	✓
LeCaR	✓	✓	✓	✗
DLIRS	✓	✗	✗	✓



Prior Work: LeCaR

- ML-Based: Reinforcement **Le**arning On **Ca**che **R**eplacement
  - Simple
    - LRU, LFU as experts
  - Adaptive
    - Update weights
  - Outperforms state-of-the-art
    - Small cache sizes

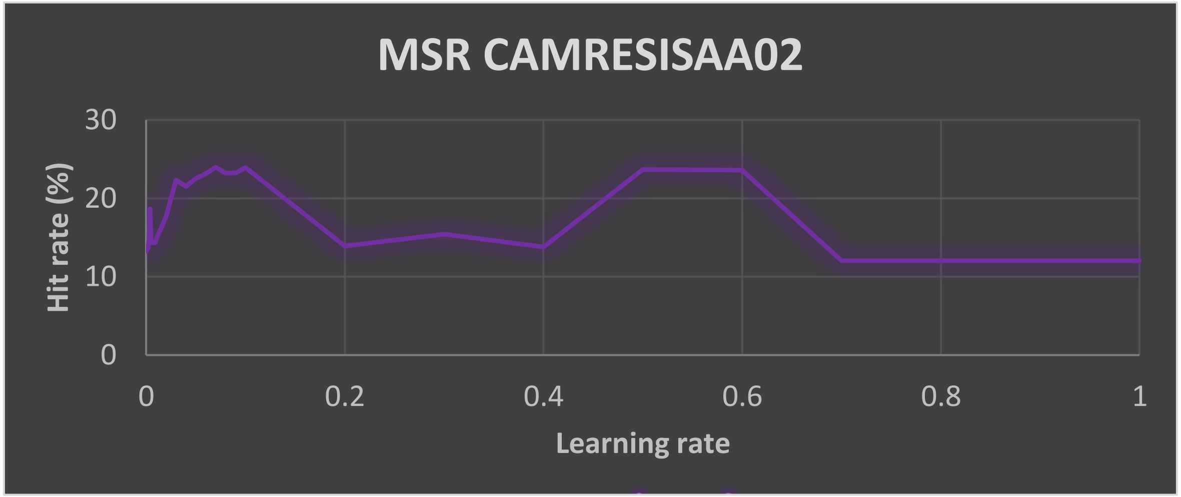


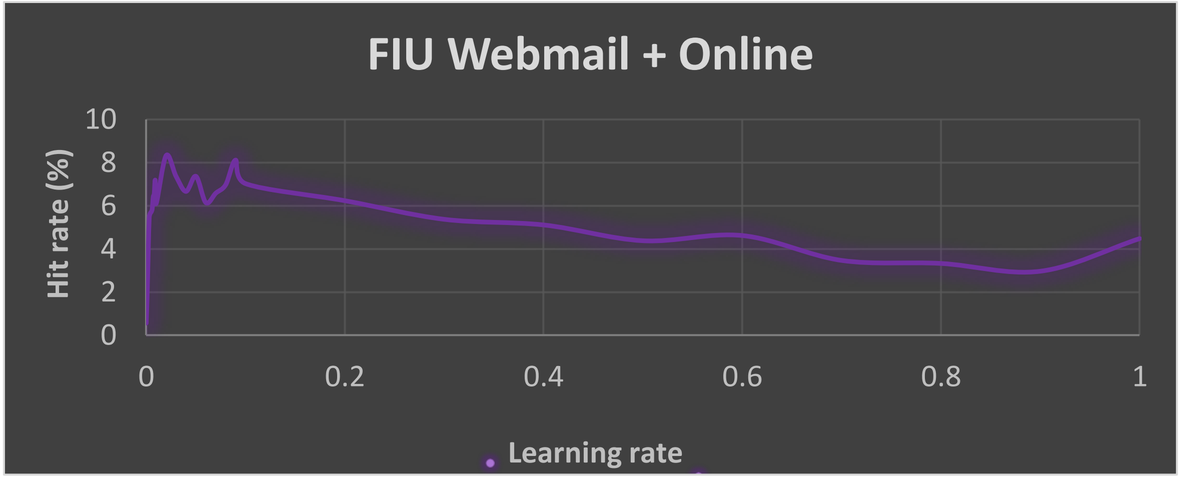
Vietri, et al., “Driving cache replacement with ML-based LeCaR.” HotStorage ‘18.

Limitations of LeCaR

- Fixed Learning rate
  - 0.45 (empirically chosen)
- Can’t handle Scan

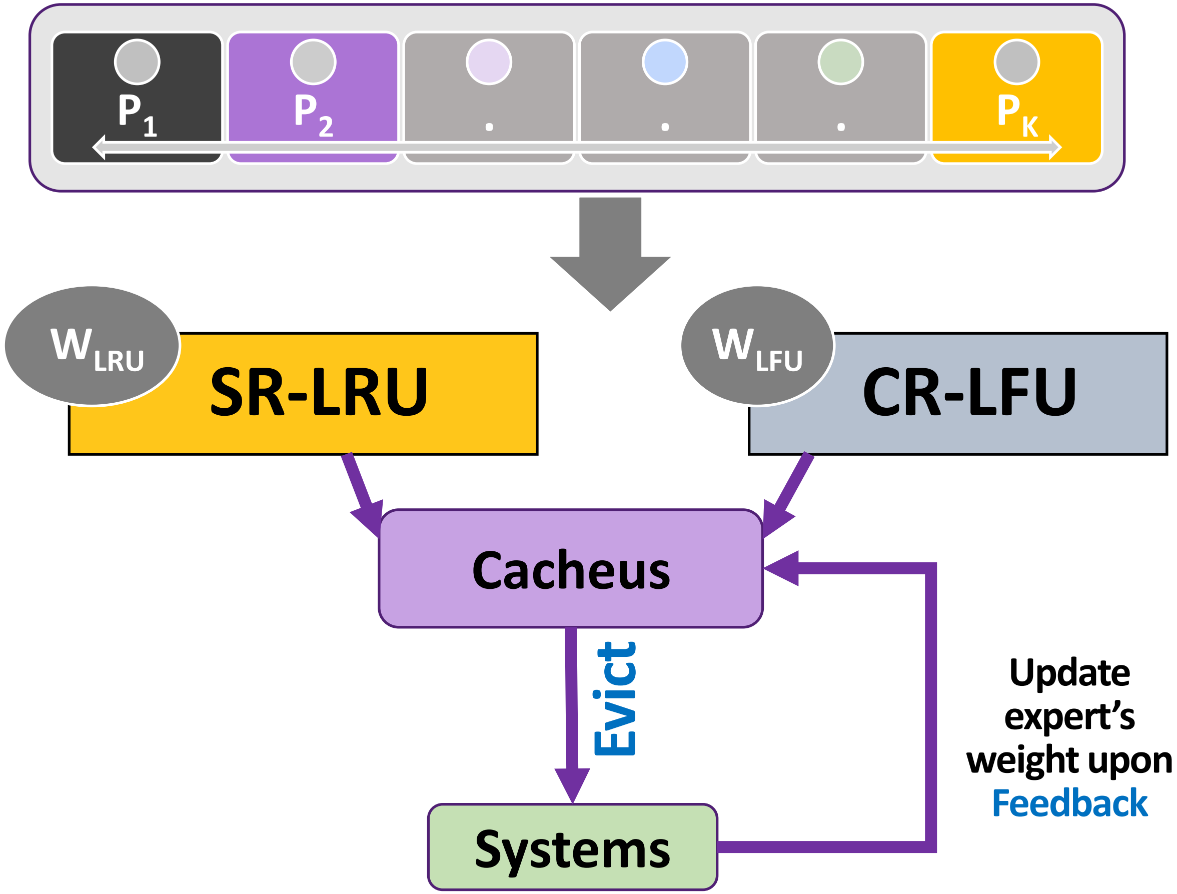
Algorithm	LRU	LFU	Churn	Scan
ARC	✓	✗	✗	✓
LIRS	✗	✗	✗	✓
LeCaR	✓	✓	✓	✗
DLIRS	✓	✗	✗	✓





CACHEUS: Improving LeCaR

- Adapt Learning Rate
- Improve experts
  - Introduce scan resistance
    - Replace LRU with
      - ARC (C1)
      - LIRS (C2)
      - DLIRS
    - Scan resistant LRU: SR-LRU (C3)
  - Improve churn resistance
    - Churn resistant LFU (CR-LFU)



CACHEUS: Learning Rate Adaptation

- Learning rate changed
  - Performance change
    - Positive, reinforce latest direction
    - Negative, reverse the latest direction
- Learning rate unchanged
  - Performance change
    - Positive, no update
    - Negative, random jump
- Performance zero for 10 intervals (Einziger et. Al, Middleware '18)
  - Restart Learning

$$w_t = w_{t-1} e^{\boxed{\eta} r}$$

Learning rate

CACHEUS: Scan Resistance LRU (SR-LRU)

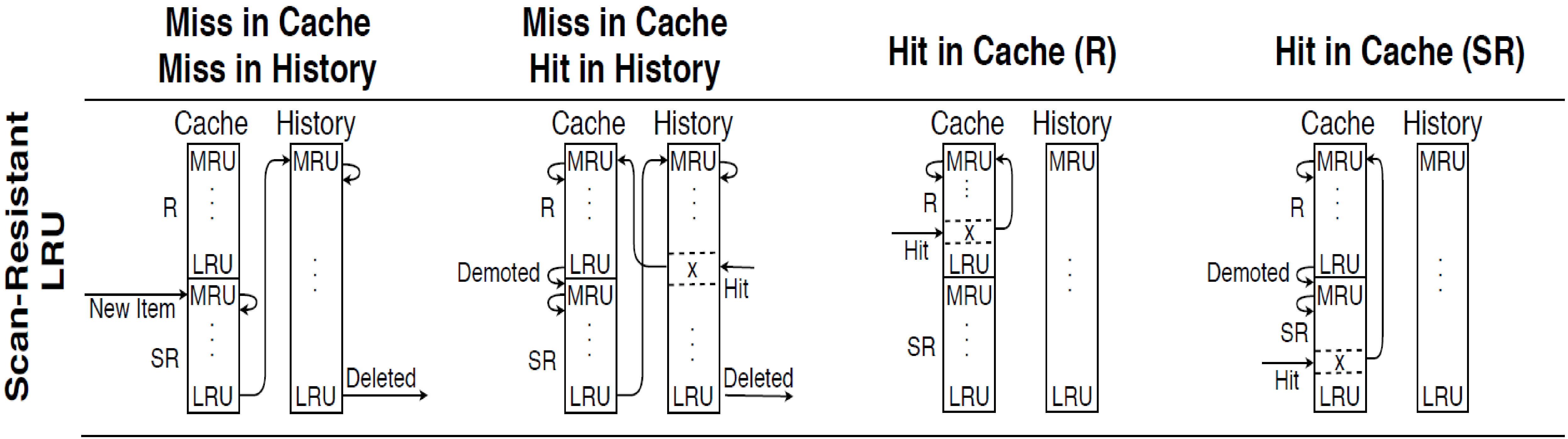


Figure: Understanding SR-LRU. Actions taken to handle request x for: cache miss, cache miss with x in history, cache hit with x in SR, and cache hit with x in R.

CACHEUS: Churn Resistance LFU (CR-LFU)

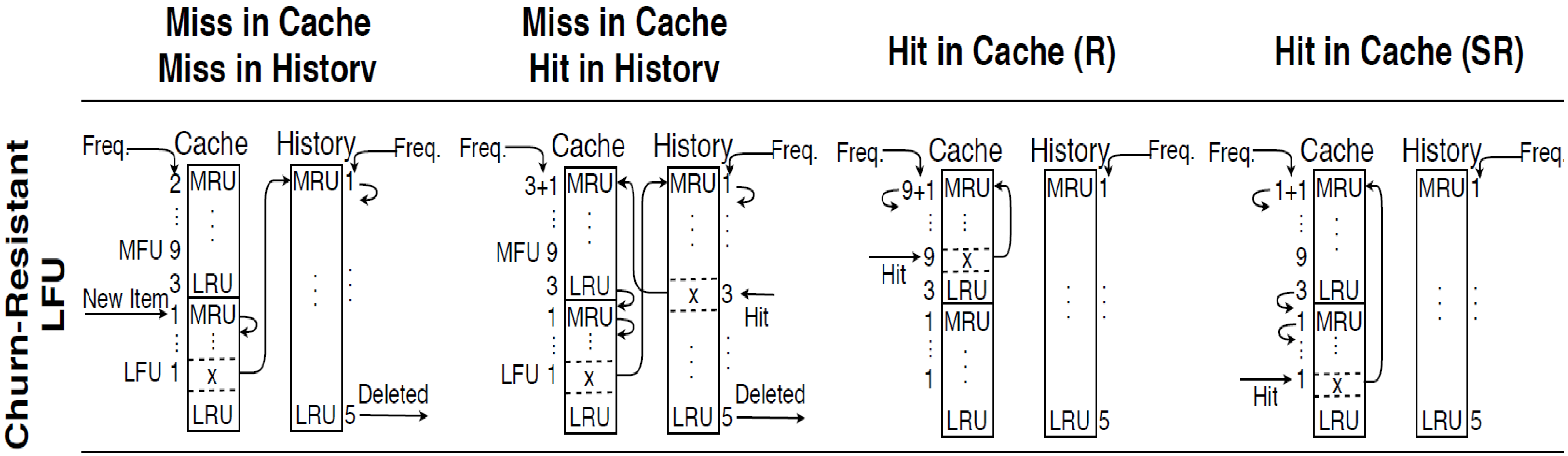


Figure: Understanding CR-LFU. Actions taken to handle request x for: cache miss, cache miss with x in history, cache hit with x in SR, and cache hit with x in R.

CACHEUS: Scan And Churn Resistance (Cont.)

SR-LRU

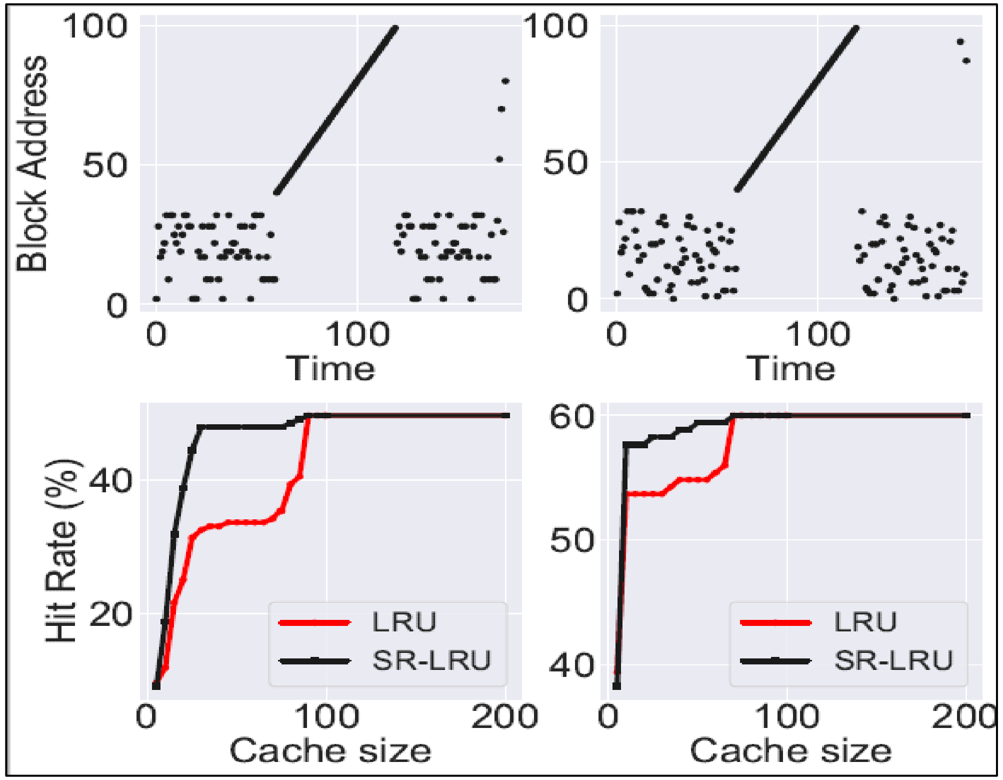


Figure: SR-LRU with the scan workload primitive type. Two synthetic workloads: LFU-friendly pattern (left column) and LRU-Friendly pattern (right column). The working set is 175 items with a single inserted scan of size 60.

CR-LFU

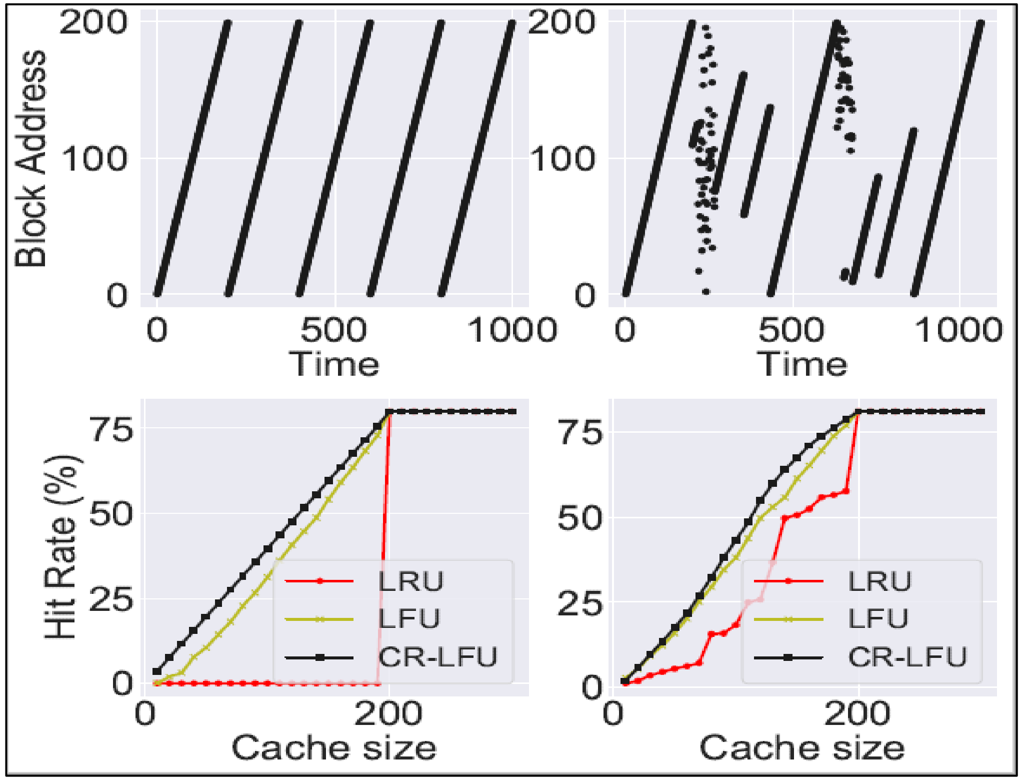


Figure: CR-LFU with the churn workload primitive type. Two synthetic workloads: a churn pattern (left column) and a combination of churn and LRU-friendly pattern (right column). The working set is 200 items.

Experiments

- **Datasets**: 5 different sources
- **Cache sizes**: 0.05, 0.1, 0.5, 1, 5, 10%
- 6+1 **Algorithms compared**
  - LRU, LFU, ARC, LIRS, LeCaR, DLIRS, CACHEUS (3 variants)
- **Total experiments: 17,766**

Dataset	# of Traces
FIU	184
MSR	22
CloudPhysics	99
CloudVPS	18
CloudCache	6
Total	329

Evaluation of CACHEUS

- Paired t-test used
- **Significance**: p-value
  - Green: Significantly better
  - Red: Significantly worse
  - Gray: Not significant
- **Magnitude**: Effect size (Cohen’s d-measure)
  - Dark: High effect
  - Light: Low effect

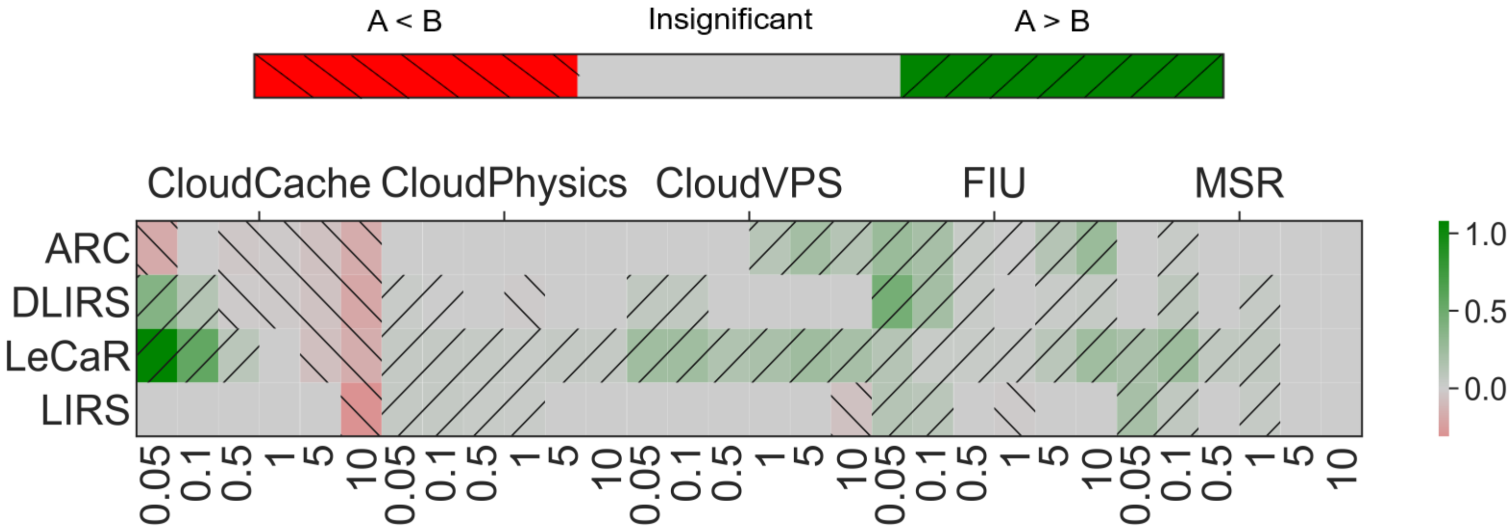


Figure: CACHEUS vs. others

	Effect size	Better	Insignificant	Worse
CACHEUS	[-0.31 , **2.08**]	47%	40%	13%

Contributions

- **Workload Primitive Types**
  - LRU-friendly, LFU-friendly, Churn, Scan
- **CACHEUS**: Improved Cache replacement algorithm
  - **Adaptive** learning rate
  - **Improved** experts: LRU and LFU algorithm
    - SR-LRU and
    - CR-LFU
  - **Comprehensive** evaluations (17,766 simulations)
  - **Outstanding** Performance

THANKS!!!

- SyLab Team, SCIS, FIU
- BioRG Team, SCIS, FIU
- Kenneth Salem
- FAST Organizing Committee

NSF NetApp®

fyusu003@fiu.edu

https://sites.google.com/fiu.edu/farzana