

# 강화 학습 기반의 프리페처와 캐시 교체 기법

진준우<sup>1</sup>, 김찬우<sup>1</sup>, 김희진<sup>1</sup>, 이미소<sup>1</sup>, 주용완<sup>1</sup>, 오현영<sup>2\*</sup>

<sup>1</sup>가천대학교 AI 소프트웨어학부 학부생

<sup>2</sup>가천대학교 AI 소프트웨어학부 교수

{wlswnsdn, cwkim010129, g579359, dyddhksdl1, miso0723, hyoh}@gachon.ac.kr

## RL based prefetcher and cache replacement method

Junwoo Jin, Chanwoo Kim, HeeJin Kim, Miso Lee, Yongwan Joo, Hyunyoung Oh

Dept. of AI Software, Gachon University

### 요 약

현대 컴퓨터 시스템에서 메모리 접근의 효율성을 극대화하는 것은 필수적이다. 기존의 캐시 교체 및 프리페칭 기법은 고정된 규칙에 기반해 설계되어 동적 접근 패턴에 제대로 적응하지 못한다. 이 논문에서는 이러한 한계를 극복하기 위해 RL 기반 기술을 통해 동적 환경에 적응하는 효율적인 기법들을 소개한다. 이러한 방식은 기존의 캐시 교체와 프리페칭이 지닌 문제를 해결하고, 효율적인 메모리 관리 방안을 제안한다.

### 1. 서론

“강화학습(Reinforcement Learning)”은 에이전트가 환경과 상호작용하며 보상을 극대화하는 최적의 행동을 학습하는 방법으로, 높은 적응성과 최적화 능력을 제공한다. 자율주행과 공격 탐지 등의 다양한 분야에서 복잡한 패턴 학습과 동적 환경 적응에 강점을 가진다[1]. 최근 컴퓨터 구조 분야에서도 강화학습의 적용이 늘어나고 있으며 특히 캐시 메모리 관리 분야에서 주목할 만한 성과를 보이고 있다. 이 논문은 강화학습을 활용한 프리페칭과 캐시 교체 방법을 탐구하고, 전통적인 방식에 비해 캐시 메모리 관리 성능 향상에 기여하는 방안을 소개한다.

### 2. 배경 지식

#### 2.1 프리페처

프리페처는 프로세서의 메모리 접근 지연을 줄이기 위해 사용되는 예측 기법이다. 프로그램 실행 중 발생하는 데이터 접근 패턴을 모니터링하고, 이를 기반으로 미래에 필요할 데이터를 예측하여 미리 캐시로 가져온다. 이를 통해 메모리 접근 시간을 단축시켜 프로세서의 성능을 향상시킨다. 하지만 이런 예측에 오류가 있을 수 있기 때문에, 이를 줄이기 위한 학습 메커니즘을 포함한 경우가 많다. 이 논문에서는 강화 학습을 사용하는 두 기법을 대표적으로 알아보겠다.

#### 2.2 캐시 교체

캐시의 제한된 용량으로 인해, 새로운 데이터를 저장하기 위해 기존 데이터를 제거하는 과정이 필요하다. 이때 어떤 데이터를 교체할지 결정하는 알고리즘을 캐시 교체 정책이라고 한다. 효과적인 캐시 교체 정책은 캐시의 히트율을 높여 시스템의 전반적인 성능 향상에 기여한다. 현존하는 캐시 교체 정책들은 설계 과정에서 많은 자원이 소모되거나, 기술의 발달로 인해 다중 코어 환경이 등장하면서 새로운 방식의 캐시 교체 정책이 요구되고 있다. 따라서 이 논문에서는 강화학습을 활용한 기존 기법 개선과 새로운 캐시 교체 정책 설계 방법을 살펴보겠다.

### 3. 강화학습 기반 프리페처

#### 3.1 알고리즘



(그림 1) 프리페처 동작 과정

기본 구조는 에이전트와 환경으로 이루어져 있다. 에이전트는 시간  $t$  마다 환경의 상태를 관찰하고, 이를 바탕으로 가능한 행동 중 가장 높은  $Q$  값(상태-행동 쌍에 대한 예상 보상값)을 가진 행동을 선택한다. 이 행동을 통해 미리 가져오기 요청을 보내고, 성공 여부에 따라 보상을 받는다. 보상에 따라  $Q$  값을 업데이트

\* 교신저자

이트하며 학습이 진행된다[2,3].

### 3.2 Pythia

Pythia는 단일 에이전트 기반 프리페처 시스템으로, 여러 프로그램의 특징 정보와 시스템 수준의 피드백을 통해 메모리 접근 패턴을 예측한다. 메모리 요청마다 프로그램의 컨텍스트를 분석하고 프리페치 결정을 내린다. 또한 요청에 대한 보상을 통해 더욱 정확히 메모리 예측을 학습하며, 시스템 자원을 고려한 성능 최적화도 진행한다. Pythia는 프리페칭을 사용하지 않은 싱글 코어 환경 대비 22%의 성능 향상을 보였다.

### 3.3 RL-CoPref

RL-CoPref는 다중 프리페처 제어 알고리즘으로, Q-러닝을 활용하여 여러 프리페처의 협업을 통해 최적의 프리페치 정책을 수립한다. 프로그램 실행 환경에 기반하여 다수의 프리페처를 동적으로 선택하고 조정함으로써 혼합된 메모리 접근 패턴 문제를 효과적으로 해결하는 강화 학습 기반의 프리페칭 컨트롤러이다. 강화학습을 사용하지 않은 기존 프리페처(BO, MISB)에 비해 각각 4.68%, 10.88%의 성능을 개선시켰다.

### 3.4 두 방식의 차이점

항목	Pythia	RL-CoPref
에이전트	단일	다중
보상 지급 case	정확·적시적인 프리페치	정확·적시적인 프리페치 협력 프리페치
적합한 구조	단일, 다중 캐시 계층	다중 코어, 다중 캐시 계층

(그림 2) 두 프리페처 간의 비교표

Pythia는 단일 에이전트를 사용하여 단일 캐시 계층같은 간단한 메모리 구조에서 더 좋은 성능을 보인다. 반면 RL-CoPref는 다중 에이전트를 사용하여 복잡한 메모리 계층에서 협력을 통해 성능을 극대화하는데 중점을 둔다. Pythia는 프리페칭에 대한 보상을 지급할 때 프리페치의 정확성과 시스템 수준의 피드백에만 초점을 두지만, RL-CoPref는 각 메모리 계층이 독립적으로 학습한 결과를 공유하여 개별적인 보상뿐만 아니라 타 계층에 협력했을 때도 보상을 지급한다.

## 4. 강화학습 기반 캐시교체

### 4.1 Next-Attempt[4]

Next-Attempt는 멀티코어 환경에서 캐시 블록의 일관성을 고려한 강화학습 기반 캐시교체 기법이다. 에이전트는 블록의 일관성 상태와 공유자 비트 벡터 정보를 바탕으로 퇴출 여부를 결정하며, 캐시 미스에 따라 보상을 받아 가중치 벡터를 조정한다. 이는 기존의 캐시 교체 알고리즘의 결정을 검토하여 부적절한 경우 다음 후보를 선택해 더 효율적인 교체가 가능케 한다.

### 4.2 RLR(Reinforcement Learned Replacement)[5]

RLR 정책은 강화학습을 통해 캐시 교체 정책을 설계하는 기법이다. 정책 설계 시 에이전트의 보상에 기여

하는 주요 특징들을 확인할 수 있다. 그 후, NN(Neural Network)를 사용하여 특징들의 하이퍼 파라미터를 찾을 수 있는데, 이를 통해 낮은 우선 순위를 가지는 캐시 라인부터 효율적으로 퇴출시킬 수 있다.

### 4.3 두 방식의 차이점

항목	Next-Attempt	Reinforcement Learned Replacement
교체캐시 결정 방식	캐시 블록의 일관성 상태와 공유자 수를 고려하여 학습된 가중치 기반 의사 결정	강화 학습을 통해 찾은 주요 특징을 이용해 캐시 교체 정책 결정
하드웨어 오버헤드	디렉터리 항목당 19비트 (16비트 + 3비트), 교체 이력 테이블 2KN(T+3) 비트 필요 (K: 연관도, N: 세트 수, T: 태그 비트)	기존 알고리즘 대비 2MB/8MB LLC(Last Level Cache)에서 각 16.75KB/67KB 만큼 오버헤드 감소

(그림 3) 두 캐시교체 간의 비교표

그림 3를 통해 확인하면, 교체 캐시 결정 방식에서는 강화학습을 통해 의사 결정을 하는지 혹은 주요 특징을 확인하는지의 차이가 있다. 하드웨어 오버헤드의 경우, Next-Attempt는 추가적인 메모리가 필요하지만 RLR은 하드웨어 오버헤드를 감소시킨다는 차이가 있다. 성능의 경우 기존 알고리즘에 비해 Next-Attempt 사용시 L1, L2 캐시에서 각각 캐시 미스율이 41.2%, 27.3% 감소했고, RLR 사용시 단일 코어, 4코어 환경에서의 성능이 각각 3.25%, 4.86% 증가했다.

## 5. 결론

RL-CoPref, Pythia, Next-Attempt, RLR을 통해 동적인 환경에서도 강화학습이 최적의 행동을 선택해 성능을 극대화할 수 있음을 분석했다. 향후 하드웨어 결합 복구 메커니즘, TPU의 효율성을 높이기 위한 다양한 작업 설계, 자율 작업 할당 등 컴퓨터 구조 분야에도 강화학습으로 성능을 향상시킬 것으로 기대된다.

## 사사문구

이 논문은 2024년도 정부(산업통상자원부)의 재원으로 한국산업기술기획평가원의 지원(No. RS-2024-00406121, 자동차보안취약점기반위협분석시스템개발(R&D))과 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원(No. RS-2022-00166529)을 받고 과기정통부 정보통신기획평가원의 정보보호핵심원천기술개발사업(No. RS-2024-00337414)으로 수행한 결과임.

## 참고문헌

- [1] R. Sutton and A. Barto, "The Reinforcement Learning Problem." MIT Press, 1998.
- [2] Rahul Bera et al, "A Customizable Hardware Prefetching Framework Using Online Reinforcement Learning." MICRO, 2021
- [3] Yang, et al, "RL-CoPref: a reinforcement learning-based coordinated prefetching controller for multiple prefetchers." J Supercomput, 2024.
- [4] Matheus A. Souza et al, "Reinforcement Learning-Based Cache Replacement Policies for multicore processors." IEEE Access, 2024
- [5] Subhash Sethumurugan et al, "Designing a Cost-Effective Cache Replacement Policy using Machine Learning." HPCA, 2021