

Soft Actor-Critic 기반 계층형 메모리 시스템에서의 적응형 메모리 관리 기법

(Adaptive Memory Management Techniques in Tiered Memory Systems
Based on Soft Actor-Critic)

박지원*, 정진만**

(Jiwon Park, Jinman Jung)

요약

본 논문은 계층형 메모리 시스템에서 발생하는 페이지 마이그레이션 오버헤드와 자원 경쟁 문제를 해결하기 위해 강화학습 기반의 동적 제어 프레임워크를 제안한다. 기존의 단일 메트릭 기반 정책은 워크로드 접근 패턴과 시스템이 급변하는 환경에서 최적 운영점을 지속적으로 추적하기 어렵다. 이를 극복하기 위해 본 연구는 다변량 상태 공간을 설계하여 시스템의 동적 변화를 보다 빠르게 반영하도록 하였다. 또한 Soft Actor-Critic(SAC)을 활용하여 처리량 향상, 혼잡도 억제, 제어 진동 완화를 함께 고려한 다목적 비용 인지형 보상 함수를 기반으로 커널의 승격 속도 제한을 동적으로 조정한다. 실제 Linux 커널 환경에서 평가한 결과, 제안 기법은 워크로드 접근 패턴 변화 시 평형점 수렴 시간을 기존 대비 약 15.6% 단축하였으며, 평균 처리량을 1.93% 향상시켰다. 특히 극심한 자원 경쟁 상황에서도 불필요한 마이그레이션을 1.47% 억제하여 시스템을 보다 안정적으로 유지할 수 있음을 보였다.

■ 중심어 : CXL ; 계층형 메모리 시스템 ; 강화학습 ; Soft Actor-Critic ; 페이지 마이그레이션

Abstract

This paper proposes a reinforcement learning-based dynamic control framework to address page migration overhead and resource contention in tiered memory systems. Existing single-metric policies have difficulty continuously tracking the optimal operating point when workload access patterns and system conditions change rapidly. To address this limitation, this study designs a multivariate state space so that dynamic system changes can be reflected more quickly. In addition, using Soft Actor-Critic (SAC), the proposed framework dynamically adjusts the kernel's promotion rate limit through a multi-objective cost-aware reward function that jointly considers throughput improvement, congestion suppression, and control oscillation mitigation. Experimental results on a real Linux kernel environment show that the proposed method reduces equilibrium convergence time by approximately 15.6% and improves average throughput by 1.93% under workload access pattern changes. Notably, even under severe resource contention, it suppresses unnecessary migrations by 1.47%, showing that the system can be maintained more stably.

■ keywords : CXL ; Tiered Memory Systems ; Reinforcement Learning ; Soft Actor-Critic ; Page Migration

I. 서론

최근 고성능 서버 시스템은 다중 소켓 기반의 NUMA(Non-Uniform Memory Access) 구조를 바탕으로, CXL(Compute Express Link) 기반의 확장 기술을 적극 도입하여 계층형 메모리 시스템을 구성하고 있다. 일반적으로 이러한 계층형

메모리 시스템은 대역폭이 높고 지연시간이 작은 로컬 메모리(Local Memory)와 용량은 크지만 지연시간 또한 큰 원격 메모리(Remote Memory)로 이루어진다. 이러한 환경에서는 접근 지역성(locality) 확보를 위한 운영 체제의 동적 페이지 마이그레이션이 응용 프로그램의 성능을 좌우한다.

* 정회원, 인하대학교 전기컴퓨터공학과

** 중신회원, 인하대학교 컴퓨터공학과

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. RS-2023-00252501, No. RS-2026-25471408).

접수일자 : 2026년 03월 03일

수정일자 : 2026년 03월 19일

게재확정일 : 2026년 03월 20일

교신저자 : 정진만 e-mail : jmjung@inha.ac.kr

그러나 페이지 이동은 본질적으로 비용을 수반한다. 마이그레이션 과정에서 발생하는 페이지 복사는 CPU 사이클을 소모하며, 노드 간 데이터 이동은 소켓 간 링크(UPI) 및 메모리 컨트롤러(IMC)의 대역폭 경합을 증가시켜 시스템 혼잡을 유발할 수 있다. 즉, 계층형 메모리 시스템에서의 동적 페이지 배치는 성능 향상을 위한 핵심 수단인 동시에, 잘못 적용될 경우 오버헤드를 증폭시켜 성능을 악화시킬 수 있는 양면성을 가진다.

기존의 휴리스틱 기반 제어 혹은 경험적 정적 튜닝은 고정된 환경에서는 유의미한 개선을 보였지만, 실제 운영 환경의 동적 변화에 대응하기 어렵다. 실제 운용 환경에서는 핫셋 접근 확률 또는 지역성 같은 워킹셋의 접근 특성 또는 메모리 인터커넥트 경합 수준이 시간에 따라 지속적으로 변화한다. 이때 동일한 마이그레이션 강도라도 어떤 구간에서는 원격 접근을 줄여 성능을 개선할 수 있으나, 다른 구간에서는 IMC/UPI 혼잡과 오버헤드를 유발하여 오히려 성능을 악화시키는 등 상태에 따라 제어 효과가 상이하게 나타난다.

따라서 본 논문의 목표는 TPP 및 Colloid 환경에서 강화학습을 통해 페이지 마이그레이션 강도를 동적으로 최적화하여, 처리량(throughput)을 향상하는 동시에 혼잡 및 오버헤드를 억제하는 것을 목표로 한다. 이를 위해 연속 제어 문제에 적합한 Soft Actor-Critic(SAC)을 적용하고, 커널 수준의 제어 노브(knob)를 실시간으로 조정하는 정책을 학습한다. 구체적으로, (i) 실시간 모니터링을 통해 수집되는 지표를 기반으로 상태(state)를 구성하고, (ii) 처리량과 혼잡도, 비용을 동시에 반영하는 cost-aware reward를 정의하여 다목적 목표를 단일 보상 함수로 정리하며, (iii) 학습된 정책을 커널 제어 루프와 통합하여 실제 운영 환경에서도 적용 가능하도록 설계하였다.

II. 관련 연구

1. 계층형 메모리 시스템에서의 페이지 이동 정책

CXL 기반 계층형 메모리 환경에서의 핵심 과제는 한정된 상위 계층에 유효 워킹셋(Working Set)을 유지하면서 페이지 이동에 수반되는 오버헤드와 혼잡을 최소화하는 것이다. 대표적으로 TPP(Transparent Page Placement)[1]는 페이지 테이블 스캔 및 힌트 폴트(Hint fault)를 통해 페이지의 접근 빈도(Hotness)를 식별한다. 이를 기반으로 상위 계층의 여유 공간을 확보하는 선제적 강등(Proactive demotion)과 하위 계층에 적재된 핫 페이지의 즉각적 승격(Prompt promotion)을 수행하면서도, 샘플링 오버헤드와 불필요한 마이그레이션을 억제한다. 이 과정에서 스캔 주기, 대상 선별 임계값, 이동 속도 제한(Rate limit) 등의 제어 노브가 정책의 방향을 결정한다. 중요한 것은 이들이 독립적으로 동작하지 않고 강한 상호 의존성을 지닌다는 것이다. 만약 스캔 주기를 늘리면 워크로드의 접근 패턴 변화를 제때 반영하지 못해 마이그레이션 지연(Migration Lag)을 초래하며, 반대로 이동 속도 제한을 과도하게 완화하면 메모리 컨트롤러(IMC) 및 인터커넥트 혼잡이 급증하게 된다. 즉, 계층형 메모리 관리는 단순한 단일 파라미터 튜닝으로 해결될 수 없으며, 성능 향상과 오버헤드 억제 간의 다목적 상충 관계(Multi-objective trade-off)를 종합적으로 고려해야 하는 복합적 최적화 문제로 귀결된다.

2. Colloid의 단일 메트릭 접근법

Colloid[2]는 계층형 메모리 환경에서 페이지 배치의 목표를 단순히 상위 계층에 hot page를 채우는 것으로 고정하기보다, 각 계층의 부하 인지형 지연시간(Loaded latency)을 균형화하는 관점에서 접근한다. Colloid는 현재 로컬 메모리에 있는 페이지들의 접근 확률의 합(p)을 핵심 지표로 정의하며, 이는 전체 메모리 요청 대비 로컬 메모리 요청의 비로 계산된다.

$$p = \frac{local_requests}{(local_requests + remote_requests)} \quad (1)$$

정책의 목표는 관측된 p 를 최적의 평형점인 p^* 부근의 허용 범위($p_{lo} \leq p \leq p_{hi}$)내로 수렴시키는 것이다.

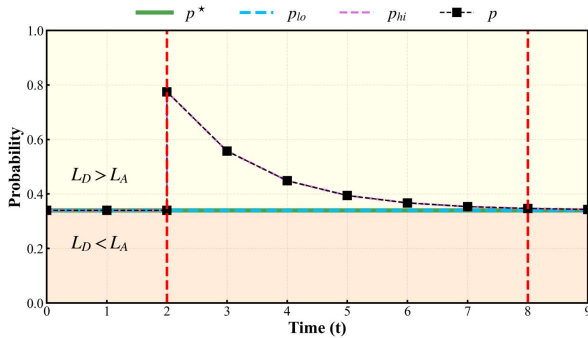


그림 1. 안정적인 환경에서의 Colloid 정책 수렴

그림 1은 시스템이 안정적인 상태에서 Colloid가 목표 평형점에 성공적으로 수렴하는 이상적인 동작 과정을 보여준다. 그러나 이러한 단일 메트릭 기반 제어에는 다음과 같은 어려움이 있다. 첫째, 워크로드 접근 패턴 변화가 빠른 환경에서 마이그레이션 지연이 발생할 소지가 있다.

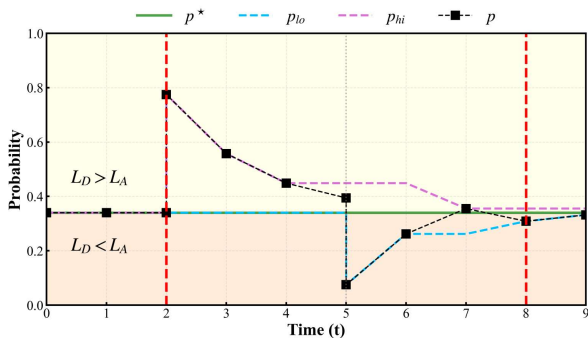


그림 2. 동적 환경에서의 Colloid 정책의 반응성 저하

그림 2와 같이 접근 패턴이 빠르게 변화하는 환경에서는 단일 메트릭 기반 제어에 한계가 발생할 수 있다. 둘째, 다목적 최적화에 제약이 있다. 실제 시스템 운영 목표는 단순히 지연시간 격차를 해소하는 것이 아니라, 처리량 보존, 인터connect 혼잡 억제, 진동(oscillation) 완화를 포괄적으로 관리해야 한다. 지연시간 중심의 단일 메

트릭 제어는 이러한 복합적인 최적화 목표를 동시에 만족시키기 어려우며, 특정 구간에서 보수적 제어로 인한 처리량 손실을 야기할 수 있다.

3. 강화학습 기반 시스템 제어 연구 동향
 강화학습(Reinforcement Learning, RL)은 비선형적이고 동적인 시스템 환경에서 다변량 상태(State)를 관측하여 연속적인 의사결정을 내리는 데 적합한 방법론이다[3]. 최근 계층형 메모리 시스템에서도 RL을 직접 적용하는 연구가 등장하고 있다. IDT[4]는 multi-tier main memory에서 RL 기반 demotion policy auto tuning을 제안하며, 유효한 모니터링이 RL 성능의 전제 조건임을 강조한다. ArtMem[5]은 계층형 메모리 시스템에서 동적으로 변화하는 워크로드에 적응하는 RL 기반 마이그레이션 프레임워크를 제시하여, 성능 개선과 불필요한 마이그레이션 감소를 동시에 지향한다. 이러한 계열의 연구는 단일 규칙 기반 정책의 한계를 넘어서, 혼잡도·오버헤드·변동성을 포함한 다변량 상태를 입력으로 받고, 다목적 비용-이득을 보상 함수로 명시화함으로써 워크로드 접근 패턴 변화와 상충 관계가 공존하는 환경에서 정책이 스스로 운영점을 찾아가도록 하는 접근이다.

III. 제안 기법

1. 데이터 분석

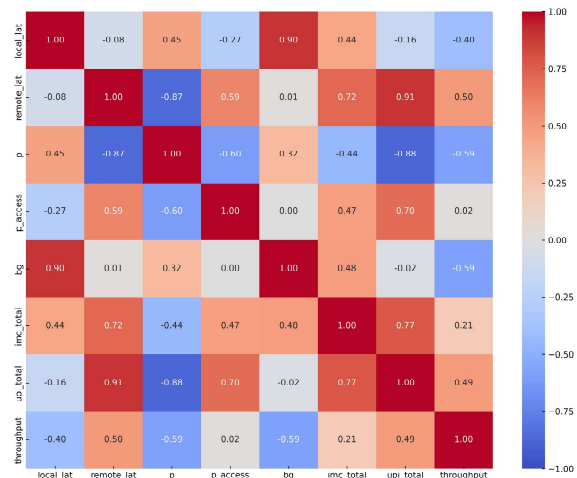


그림 3. 시스템 성능 지표 간 상관관계 분석 히트맵

그림3의 상관관계 분석 결과에 따르면, 원격 지연시간은 UPI와 강한 양의 상관관계를 보인다. 반면 로컬 접근 확률(p)을 높이는 제어는 원격 메모리 부하를 완화하지만, 로컬 지연시간을 상승시키는 상충 관계를 유발한다.

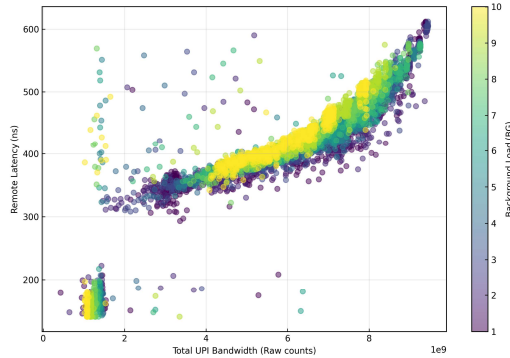


그림 4. UPI 대역폭 대비 원격 지연시간

또한 동일한 UPI 대역폭 하에서도 경합도에 따라 원격 지연시간 분포가 상이함을 보여주는데, 이는 경합도가 UPI와 비선형적인 영향을 미치고 있음을 보여준다.

이러한 분석은 정적인 휴리스틱 정책이나 단일 지표 기반 제어로는 시변하는 최적 운영점을 지속적으로 추적하기 어려움을 시사한다. 따라서 본 연구는 에이전트가 UPI와 IMC 혼잡도, 로컬과 원격 지연시간, 마이그레이션 비용 등을 통합적으로 판단하는 다변량 상태 기반 제어 프레임워크를 제안한다.

2. 전체 시스템 개요

본 연구는 TPP 및 Colloid가 적용된 커널 환경에서 페이지 마이그레이션의 개입 강도를 동적으로 최적화하기 위해, 실시간 모니터링과 정책 추론, 커널 적용이 유기적으로 결합된 폐루프(Closed-loop) 제어 구조를 설계하였다. 전체 시스템은 일정 제어 주기마다 워크로드의 실행 상태를 관측하고, 이를 바탕으로 최적의 제어 행동을 결정하여 커널에 반영하는 과정을 반복한다.

시스템 아키텍처는 크게 세 단계로 구분된다. 첫째, 모니터링 및 전처리 단계에서는 실시간 처리량(Throughput), 계층별 지연시간(Local/Remote

Latency), 인터커넥트 및 메모리 컨트롤러 혼잡도(UPI/IMC Bandwidth), 마이그레이션 통계 등을 수집한다. 이때 Colloid와 동일하게 CHA(Caching/Home Agent) 기반 메커니즘을 활용하여 occupancy 및 inserts로 로컬 및 원격 메모리의 loaded latency를 빠르게 측정하고, 그 외의 UPI/IMC 대역폭 및 마이그레이션 관련 지표는 perf 기반 계측을 통해 주기적으로 수집한다. 데이터 수집은 특히 앞선 분석 결과에 따라 UPI와 IMC 지표를 통해 경합 상황을 간접적으로 파악하며, 지연시간 비율을 통해 워크로드의 접근 패턴 변화를 식별한다. 둘째, 정책 추론 단계에서는 수집된 시계열 데이터를 정규화하여 상태(s_t)를 구성하고, 학습된 신경망을 통해 연속적인 제어 값(a_t)을 도출한다. 셋째, 커널 적용 단계에서는 도출된 행동을 커널의 promotion_rate_limit을 제어하는 핵심 노브에 매핑하여 즉각 적용한다. 이러한 구조는 급격하게 변화하는 워크로드의 특성과 시스템 경합 수준에 실시간으로 적응하며 처리량 향상과 오버헤드 억제 사이의 최적 운영점을 찾는다.

3. Soft Actor-Critic 알고리즘

강화학습 알고리즘 중 값 기반(value-based) 방법(DQN)은 주로 이산 행동 공간에서 효과적이거나, 본 연구의 제어 대상은 promotion_rate와 같이 연속적으로 조절되는 커널 노브이므로, 연속 행동을 다루는 정책 기반의 Actor-Critic 계열이 적합하다. 또한 실시간 시스템 제어에서는 한정된 실행 시간 동안 학습이 진행되므로 샘플 효율과 학습 안정성이 중요하다.

계층형 메모리의 마이그레이션 제어는 단일 임계값 튜닝으로 안정적으로 유지되기 어렵다. 워크로드의 접근 패턴과 메모리 인터커넥트 컨테이션은 시간에 따라 비정상적으로 변하며, 마이그레이션 효과 또한 즉시 나타나지 않고 지연되어 관측되기 때문이다. 본 연구는 이러한 연속 제어 문제의 복잡성을 해결하기 위해 Soft Actor-Critic(SAC) 알고리즘[6]을 채택하였다.

SAC는 확률적 정책과 엔트로피 정규화를 통해

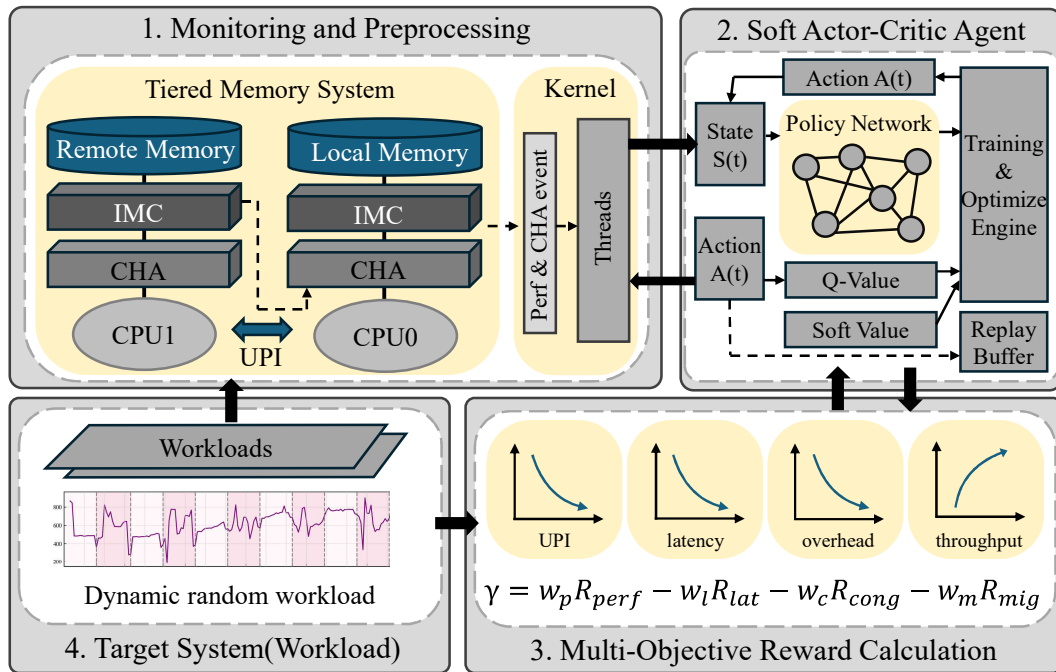


그림 5. 계층형 메모리 시스템 최적화를 위한 Soft Actor-Critic 기반 동적 워크로드 관리 프레임워크

탐색을 유지하면서도 성능을 개선하는 방향으로 학습하며, off-policy 학습을 수행하므로 리플레이 버퍼를 통해 동일한 경험을 반복 활용할 수 있어 샘플 효율이 높다. 또한 동적 워크로드처럼 상태 변동이 큰 환경에서도 정책의 과도한 진동을 완화하며 보다 안정적인 수렴을 유도할 수 있다. 따라서, 본 연구는 연속 노브 제어, 동적 환경, 샘플 효율 및 안정적 학습 요구 조건을 만족시키기 위해 SAC를 선택하였다.

4. 마르코프 결정 과정(MDP)

본 연구는 시스템 제어 문제를 MDP(Markov Decision Process) $\langle S, A, P, R, \gamma \rangle$ 로 정식화 하였다. 시스템 제어 주기는 시스템 부하를 고려하여 1초로 설정하였으며, 단순 이동 평균 대신 최신 데이터에 더 높은 가중치를 부여하는 지수 가중 이동 평균(EWMA) 기반의 상태 공간을 설계하였다.

가. 상태 공간(S)

시스템의 다차원적 상태를 인지하기 위해 다양한 벡터로 구성하였다. 이 공간은 크게 두 가지로 나뉜다. 첫째는 기초 및 파생 지표이다. 로컬, 원격 지연시간 등 기초 성능 지표와 함께 UPI와 IMC의 혼잡도를 통해 경합도를 간접 인지하고, 계층

별 지연시간 비율 및 캐시 미스 비율을 통해 워크로드의 접근 패턴을 파악한다. 모든 기초 지표는 노이즈 억제를 위해 EWMA 필터를 거친 후 로그 스케일로 정규화한다.

둘째는 추세 및 변동성 지표이다. 워크로드의 급변을 에이전트가 즉각 인지할 수 있도록 중심값, 편차, 변화율을 계산하여 추가 상태를 구성한다. 이를 통해 짧은 주기로 발생하는 워크로드의 상변이 전조도 민감하게 포착이 가능하다.

나. 행동 공간(A)

에이전트는 $[-1, 1]$ 범위의 연속적인 스칼라 값 a_t 를 선택한다. 이 값은 커널의 실제 마이그레이션 노브인 promote_rate_limit_MBps($0 \sim 10,000$ MB/s)로 선형 매핑되어 즉각 적용된다. 극한의 경합 상황에서는 0에 수렴하여 마이그레이션을 완전히 차단할 수 있도록 설계하였다.

나. 보상 함수(R)

보상 함수는 처리량 이득을 최대화하는 동시에, 시스템 혼잡도와 마이그레이션 오버헤드에 대한 페널티를 부과하는 다목적 함수로 정의하였다.

$$r_t = k_{adv} \cdot adv_t - \lambda_{lat} \cdot g_t^2 - \lambda_{mig} \cdot c_t - \lambda_{\Delta a} \cdot \Delta a_t^2 + Shaping_t$$

(2)

여기서 adv_t 는 현재의 시스템 혼잡도 환경에서 기대되는 기준치 대비 상대적 처리량 이득을 의미하며, k_{adv} 는 이 처리량 이득에 대한 가중치를 의미한다. k_{adv} 는 다른 페널티 항들과의 균형을 맞추어 에이전트가 오버헤드 억제와 성능 극대화 사이에서 어느 정도의 비중을 둘지 결정하는 매개변수 역할을 한다. 또한 λ_{lat} , λ_{mig} , $\lambda_{\Delta a}$ 는 시스템 안정성을 유지하기 위해 각 제약 조건의 반영 비율을 조절하는 페널티 계수이다. g_t 는 노이즈로 인한 민감한 반응을 방지하기 위해 설정된 허용 범위를 초과한 계층 간 지연시간 격차이다. c_t 는 시스템 혼잡도에 비례하여 가중되는 동적 마이그레이션 페널티 항으로, 경합이 심한 구간에서의 불필요한 페이지 이동을 강력히 억제한다. Δa_t^2 는 제어 진동(Oscillation)을 방지하기 위한 행동 변화량 페널티이다. 마지막으로 보상 셰이핑(*shaping*)항은 지연시간 격차가 감소하는 방향으로의 상태 전이에 보상을 부여하여 정책의 학습 및 수렴 속도를 가속한다.

IV. 성능 평가

1. 하드웨어 및 운영 체제 환경

본 연구는 다중 소켓 기반의 NUMA 구조를 활용하여 CXL 기반 계층형 메모리를 애플리케이션 환경에서 수행되었다. 프로세서는 Intel Xeon Gold 6138을 사용하였고, 메모리 계층은 로컬 메모리 16GB와 리모트 메모리 32GB로 구성하였다. 운영 체제 환경으로는 기존 Colloid 프레임워크가 구현된 Linux 6.3-colloid 커널을 활용하였다. 이를 기반으로, 유저 공간에서 동작하는 SAC 에이전트가 실시간으로 시스템 지표를 모니터링하고, 추론된 제어 값을 커널 공간의 승격 속도 제한 노브(*promote_rate_limit_MBps*)에 즉각 반영할 수 있도록 제어 환경을 구축하였다.

2. 워크로드 및 경합(contention) 설정
 실험에는 무작위 메모리 접근을 유발하는 GUPS(Giga Updates Per Second) 벤치마크를 사용하였다. GUPS의 워크로드 설정은 총 워킹셋 크기 24GB, 핫셋 크기 8GB로 구성하였다. 이러한 설정은 워킹셋이 로컬 메모리를 초과하여 페이지 마이그레이션이 필수적으로 요구되는 동시에, 핫셋 전체는 로컬 메모리에 수용 가능하므로 최적 배치 시 성능 향상 차이를 명확히 관찰할 수 있다.

3. 워크로드 접근 패턴 변화

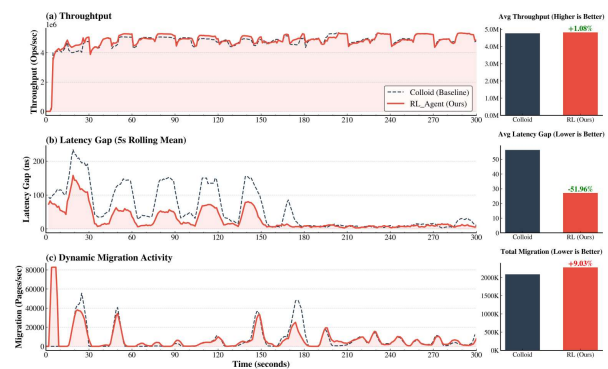


그림 6. 워크로드 접근 패턴 변화에 따른 성능 분석

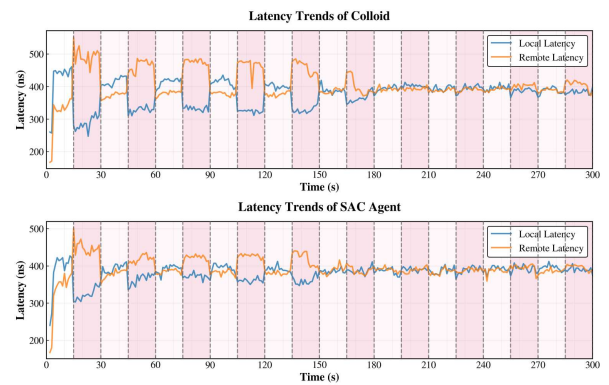


그림 7. 워크로드 접근 패턴 변화에 따른 지연시간 분석

첫 번째 실험은 GUPS의 핫셋 접근 확률을 15초마다 10%와 90%를 번갈아 동적으로 변화시켜 공간적 지역성이 급변하는 상황을 모사하였다. 이하의 성능 수치는 각 조건별 반복 실험의 평균 값이다. 지연시간 단일 지표만 사용하는 기존 Colloid 정책은 접근 패턴 변화를 사후적으로 감지함에 따라 새로운 평형점에 도달하기까지 평균 180초의 수렴 시간이 소요되었으며, 계층 간

지연시간 격차가 크게 벌어지는 경향을 보였다. 그림 6과 그림 7은 이러한 경향을 보여주는 대표 사례이다.

반면, 제안하는 SAC 기반 기법은 EWMA와 지표 변화율을 통해 접근 패턴 변화를 보다 빠르게 포착하고, 이에 따라 마이그레이션 강도를 조절하였다. 제안 기법은 평균 수렴 시간을 150초로 약 15.6% 단축하였으며, 평균 지연시간 격차 역시 기존 대비 19.4% 감소하였다.

이 과정에서 신속한 워킹셋 재배포를 위해 총 마이그레이션 발생량이 소폭 증가하였지만, 결과적으로 제안 기법은 동적 환경에서의 성능 저하를 완화하며 전체 평균 처리량을 기존 대비 1.93% 향상시켰다.

4. 시스템 경합 변동

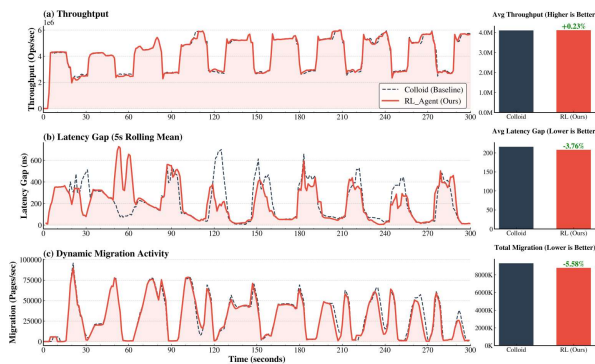


그림 8. Contention 변화에 따른 성능 분석

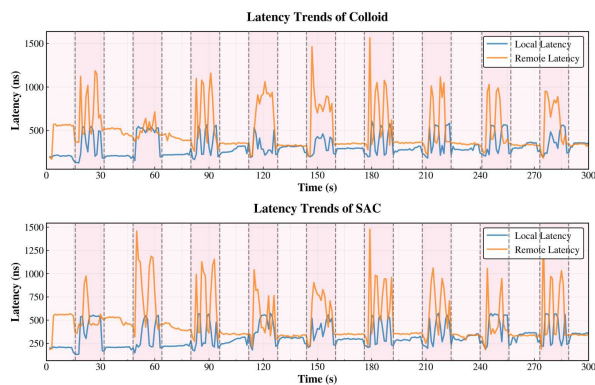


그림 9. Contention 변화에 따른 지연시간 분석

두 번째 실험은 대역폭 집약적인 STREAM 벤치마크의 활성화 코어 수를 15초마다 3개 코어와 9개 코어를 번갈아 변화시켜 인터커넥트 및 메모리 컨트롤러의 물리적 경합 수준을 동적으로

로 변화시켰다. 배경 부하가 증가하여 가용 대역폭이 극도로 포화된 상황에서는 페이지 이동 자체가 추가적인 병목을 유발하므로, 마이그레이션을 통한 절대적인 성능 향상이 물리적으로 제한된다. 그림 8과 그림 9는 이러한 상황에서의 대표적인 동작 양상을 보여준다.

SAC를 적용한 제안 기법은 시스템 혼잡도를 인지하여 불필요한 마이그레이션 발생량을 평균 대비 1.47% 감소시켰다. 이는 극심한 대역폭 결합 상황에서 불필요한 마이그레이션으로 인한 추가 병목 가능성을 줄이고, 평균 지연시간 격차를 소폭 완화(3%)하며 시스템을 보다 안정적으로 유지할 수 있음을 의미한다.

V. 결론

본 논문에서는 CXL 기반 계층형 메모리 시스템에서 발생하는 페이지 마이그레이션 오버헤드와 자원 경합 문제를 해결하기 위해, 강화학습(SAC) 기반의 동적 마이그레이션 제어 프레임워크를 제안하였다. 기존의 지연시간 단일 메트릭에 의존하는 정책은 워크로드의 접근 패턴이나 인터커넥트 혼잡이 급변하는 환경에서 마이그레이션 지연과 대역폭 포화를 유발하는 구조적 한계가 존재했다. 이를 극복하기 위해, 본 연구는 지수 가중 이동 평균(EWMA)과 지표 변화율을 포함한 다변량 상태 공간을 설계하여 시스템의 동적 변화 전조를 보다 빠르게 인지하도록 하였다. 또한, 처리량 향상과 인터커넥트 혼잡도, 제어 진동 억제를 포괄적으로 고려한 다목적 비용 인지형 보상함수를 도입하여 커널의 승격 속도 제한 노브를 실시간으로 최적화하였다.

실제 Linux 커널 환경에서 GUPS 및 STREAM 벤치마크를 활용한 평가 결과, 제안 기법은 워크로드 접근 패턴이 급변하는 환경에서 기존 대비 지연시간 수렴 시간을 약 15.6% 단축하고, 평균 지연시간 격차를 19.4% 감소시켜 동적 변화에 대한 적응성을 향상시켰다. 이에 따라 평균 처리량은 기존 대비 1.93% 향상되었다.

또한 대역폭 포화로 인해 절대적인 성능 향상이 제한되는 극한의 자원 경합 상황에서도, 불필요한 마이그레이션을 1.47% 억제하고, 평균 지연 시간 격차를 3% 완화함으로써 추가적인 병목을 줄이며 시스템을 보다 안정적으로 유지할 수 있음을 확인하였다.

결론적으로 본 연구는 복잡한 상층 관계가 존재하는 계층형 메모리 관리 문제에 강화학습을 도입함으로써 급변하는 환경 속에서 단순한 평균 처리량 향상뿐 아니라 시스템의 반응성과 안정성을 함께 고려한 제어가 가능함을 보였다. 향후 연구로는 제안한 강화학습 모델의 경량화를 통해 백그라운드 추론 오버헤드를 최소화하고, 보다 광범위한 실제 산업군 워크로드와 다양한 계층형 메모리 시스템 환경으로의 범용성을 검증할 계획이다. 특히, 다중 NUMA 및 다중 CXL 노드가 복합적으로 얽힌 확장된 메모리 풀링(Memory Pooling) 환경으로 본 프레임워크를 확장 적용하여, 차세대 서버 자원 관리를 위한 효율적인 제어 기술로 고도화하고자 한다.

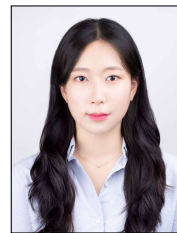
REFERENCES

- [1] Hasan Al Maruf, Hao Wang, Abhishek Dhanotia, Johannes Weiner, Niket Agarwal, Pallab Bhattacharya, Chris Petersen, Mosharaf Chowdhury, Shobhit Kanaujia, Prakash Chauhan, "TPP: Transparent Page Placement for CXL-Enabled Tiered-Memory," *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2023)*, pp. 742-755, Vancouver, Canada, Mar., 2023.
- [2] Midhul Vuppalapati, Rachit Agarwal, "Tiered Memory Management: Access Latency is the Key!," *Proceedings of the ACM SIGOPS 30th Symposium on Operating Systems Principles (SOSP 2024)*, pp. 79-94, Austin, TX, USA, Nov., 2024.
- [3] Hongzi Mao, Mohammad Alizadeh, Ishai Menache, Srikanth Kandula, "Resource Management with Deep Reinforcement Learning," *Proceedings of the 15th ACM Workshop on Hot Topics in Networks (HotNets-XV)*, pp. 50-56,

Atlanta, GA, USA, Nov., 2016.

- [4] Juneseo Chang, Wanju Doh, Yaebin Moon, Eojin Lee, Jung Ho Ahn, "IDT: Intelligent Data Placement for Multi-tiered Main Memory with Reinforcement Learning," *Proceedings of the 33rd International Symposium on High-Performance Parallel and Distributed Computing (HPDC 2024)*, pp. 69-82, Pisa, Italy, Jun., 2024.
- [5] Xinyue Yi, Hongchao Du, Yu Wang, Jie Zhang, Qiao Li, Chun Jason Xue, "ArtMem: Adaptive Migration in Reinforcement Learning-Enabled Tiered Memory," *Proceedings of the 52nd Annual International Symposium on Computer Architecture (ISCA 2025)*, pp. 405-418, Tokyo, Japan, Jun., 2025.
- [6] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, Sergey Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," *Proceedings of the 35th International Conference on Machine Learning (ICML 2018)*, pp. 1861-1870, Stockholm, Sweden, Jul., 2018.

저자 소개



박지원(정희원)

2024년 인하대학교 컴퓨터공학과 학사 졸업.

2024년~현재 인하대학교 전기컴퓨터공학과 석사과정 재학.

<주관심분야 : 지능형 임베디드 소프트웨어, 운영 체제, 메모리 시스템>



정진만(종신회원)

2008년 서울대학교 컴퓨터공학과 학사 졸업.

2014년 서울대학교 전기컴퓨터공학과 박사 졸업

2014년~2021년 한남대학교 정보통신공학과 부교수

2021년~현재 인하대학교 컴퓨터공학과 교수

<주관심분야 : 운영 체제, 임베디드 시스템, 시스템 소프트웨어>