

<https://doi.org/10.7236/JIIBC.2023.23.6.119>

JIIBC 2023-6-18

시간 지역성과 인기 편향성을 가진 데이터 참조의 모델링

Modeling of Data References with Temporal Locality and Popularity Bias

반효경*

Hyokyung Bahn*

요약 본 논문은 시간 지역성과 인기 편향성을 가진 데이터 참조를 나타낼 수 있는 새로운 참조 모델을 제안한다. 기존의 참조 모델 중 대표적인 LRU 스택 모델은 시간 지역성, 즉 최근에 참조된 데이터가 다시 참조될 가능성이 높은 성질을 나타낼 수 있으나, 데이터의 서로 다른 인기도를 고려할 수 없는 약점이 있다. 이와 반대로 데이터의 서로 다른 인기도를 반영할 수 있는 모델로 독립 참조 모델이 있으나, 시간에 따른 데이터 참조 성향의 변화를 모델링할 수 없는 한계가 있다. 본 논문이 제시하는 참조 모델은 이 두 모델의 한계를 극복하여 서로 다른 데이터의 인기도와 시간에 따른 참조 성향의 변화를 모두 반영할 수 있는 특징이 있다. 또한, 본 논문에서는 캐시 교체 알고리즘과 참조 모델의 연관성에 대해 살펴보고 제안한 모델의 최적성에 대해 보인다.

Abstract This paper proposes a new reference model that can represent data access with temporal locality and popularity bias. Among existing reference models, the LRU-stack model can express temporal locality, which is a characteristic that the more recently referenced data has, the higher the probability of being referenced again. However, it cannot take into account differences in popularity of the data. Conversely, the independent reference model can reflect the different popularity of data, but has the limitation of not being able to model changes in data reference trends over time. The reference model presented in this paper overcomes the limitations of these two models and has the feature of reflecting both the popularity bias of data and their changes over time. This paper also examines the relationship between the cache replacement algorithm and the reference model, and shows the optimality of the proposed model.

Key Words : Reference model, temporal locality, popularity bias, data reference, LRU-stack model

1. 서론

컴퓨터시스템 내의 메모리 및 스토리지는 계층적인 구조를 형성하고 있다. CPU에 가까운 상위 계층으로 갈수록

접근속도가 빨라야 하므로 단위 용량당 단가가 높은 매체로 구성하고 하위 계층으로 갈수록 용량이 커야 하므로 단가가 낮은 매체로 구성하여 접근속도가 느리다. 따라서, 이들 간의 속도 차이를 완충하기 위한 캐싱 기술

*정회원, 이화여자대학교 컴퓨터공학과
접수일자 2023년 9월 16일, 수정완료 2023년 11월 16일
게재확정일자 2023년 12월 8일

Received: 16 September, 2023 / Revised: 16 November, 2023 /
Accepted: 8 December, 2023

*Corresponding Author: bahn@ewha.ac.kr
Dept. of Computer Engineering, Ewha University, Korea

이 시스템 전체의 성능 개선을 위해 널리 연구된 바 있다^{11, 2)}. 스토리지와 메모리 계층 간의 속도 차이를 완충하기 위한 캐싱 기법 적용의 대표적인 사례로는 가상메모리의 페이징 시스템과 파일시스템의 버퍼 캐시가 있다^{12, 3)}.

한편, 캐싱 기법이 우수한 성능을 나타내기 위해서는 캐싱의 단위 데이터가 어떠한 참조 특성을 가지고 접근되는지를 분석하고 이를 모델링하는 연구가 필요하다. 데이터 참조의 특성으로 널리 알려진 대표적인 두 가지 성질로는 시간 지역성(temporal locality)과 인기 편향성(popularity bias)이 있다. 시간 지역성은 최근에 참조된 데이터가 다시 참조될 가능성이 높은 성질을 말하며, 인기 편향성은 여러 번 참조된 데이터일수록 또다시 참조될 가능성이 높은 성질을 말한다. 이러한 두 가지 성질을 가지는 데이터 참조를 반영하는 대표적인 참조 모델로 LRU 스택 모델(LRU-stack model)과 독립 참조 모델(independent reference model)이 있다. 다양한 캐싱 환경의 참조 데이터가 시간 지역성을 나타내기 때문에 LRU 스택 모델은 실제 데이터 참조 시 발생하는 시간적 특성을 모델링하기 위한 방법으로 널리 사용되고 있다⁴⁾. 그러나, LRU 스택 모델은 데이터 참조의 인기 편향성을 반영할 수 없다는 단점이 있다. 독립 참조 모델은 각 데이터가 서로 다른 참조 확률에 입각해서 접근되는 상황을 나타낼 수 있기 때문에 데이터별로 상이한 인기도를 반영할 수 있다는 장점이 있다. 그러나, 독립 참조 모델은 시간에 따라 데이터의 인기도가 변할 경우 이를 반영할 수 없다는 단점이 있다.

본 논문에서는 LRU 스택 모델과 독립 참조 모델의 단점을 극복하여 시간 지역성과 인기 편향성을 모두 표현할 수 있는 새로운 참조 모델을 제안한다. 본 논문이 제안하는 참조 모델은 각 데이터가 서로 다른 인기도를 가지고 참조되는 성향을 반영할 수 있는 동시에 그러한 인기도가 시간에 따라 변하는 상황을 고려할 수 있다.

멀티미디어, 웹 등 다양한 환경의 데이터 참조 시 시간 지역성과 인기 편향성을 동시에 가지고 있는 것으로 알려져 있어^{5, 6)}, 본 논문이 제안하는 참조 모델은 이와 같은 환경의 캐싱 기법을 위한 참조 모델링에 효과적으로 이용될 수 있을 것으로 기대된다.

본 논문의 이후 구성은 다음과 같다. II장에서는 참조 모델과 대응 관계를 이루는 캐시 교체 알고리즘에 대해 설명한다. III장에서는 본 논문이 제안하는 참조 모델을 설명하고, IV장과 V장에서는 이 모델에 최적인 교체 알고리즘에 대해 설명하고 이를 이론적 및 실험적으로 입증한다. 끝으로 VI장에서는 본 논문의 결론을 제시한다.

II. 캐시 교체 알고리즘

캐싱 기법의 성능은 교체 알고리즘에 의해 크게 좌우된다. 교체 알고리즘이란 한정된 빠른 캐시 공간에 어떠한 데이터를 담고 어떠한 데이터를 쫓아낼지를 동적으로 결정하는 온라인 알고리즘이다. 이러한 교체 알고리즘에서 데이터의 참조 가능성을 예측하는 것은 캐싱 기법의 성능에 중요한 영향을 미친다. 참조 가능성에 영향을 미치는 요소 중 시간 지역성과 인기 편향성을 고려하는 대표적인 알고리즘으로 LRU(Least Recently Used) 알고리즘과 LFU(Least Frequently Used) 알고리즘이 있다⁴⁾. 캐시에서 빈 공간이 필요한 경우 LRU 알고리즘은 가장 오래전에 참조되었던 데이터를 삭제하며, LFU 알고리즘은 참조횟수가 가장 적었던 데이터를 삭제한다.

일반적으로 하나의 참조 모델이 주어지면 그에 최적인 교체 알고리즘을 생각해 볼 수 있다. LRU 스택 모델에 최적인 교체 알고리즘은 LRU 알고리즘으로 알려져 있으며, 독립 참조 모델에 최적인 교체 알고리즘은 LFU 알고리즘으로 알려져 있다^{4, 7)}. 한편, LFU 알고리즘이 참조횟수를 카운트하는 방법은 서로 다른 두 가지를 생각해 볼 수 있다. 그 첫째는 캐시에서 삭제된 후 다시 진입한 데이터의 경우 이전의 참조횟수를 모두 합산하는 방식이고, 둘째는 데이터가 캐시에서 삭제될 때 그 참조횟수를 유지하지 않아 새롭게 캐시에 진입한 이후의 참조횟수만을 카운트하는 방식이다. 일반적인 LFU 알고리즘은 후자를 의미하지만, 독립 참조 모델에 최적인 LFU 알고리즘은 전자를 의미한다. O'Neil 등은 과거 참조 시각이 k 개까지 주어진다 가정 하에 LRU- k 알고리즘이 독립 참조 모델에서 최적임을 증명하는 바 있다⁸⁾.

한편, 캐시 교체 알고리즘 중 시간 지역성과 인기 편향성을 모두 고려하는 알고리즘으로 LRFU(Least Recently/Frequently Used) 알고리즘과 LUV(Least Unified Value) 알고리즘이 제안된 바 있다^{9, 10)}. 이들 알고리즘은 시간 지역성과 인기 편향성의 두 가지 성질을 어느 정도 반영할지를 제어하기 위해 파라미터를 두어 그 값이 0과 1 사이에서 변함에 따라 각각 양쪽 극단에서 인기 편향성과 시간 지역성만을 고려하는 알고리즘으로 설계되며 그 사이 값을 가질 경우 두 성질을 모두 반영하게 된다. 즉, LFU와 LRU가 각각 참조횟수와 마지막 참조 시각의 최근성만을 고려하는 반면, LRFU와 LUV는 이 두 가지 요소를 복합적으로 고려하여 데이터의 참조 가능성을 예측한다^{9, 10)}.

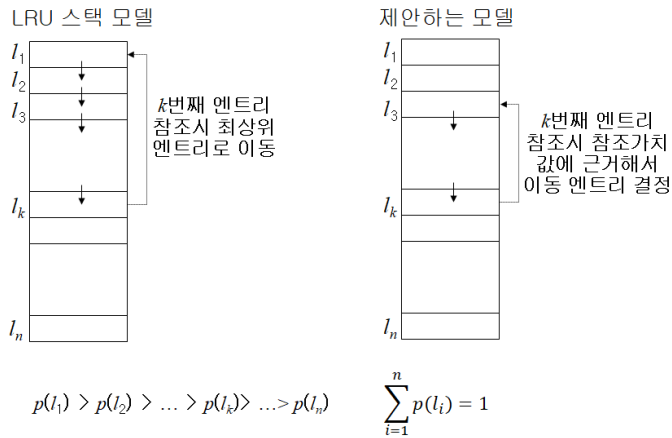


그림 1. LRU 스택 모델과 제안하는 참조 모델의 비교
 Fig. 1. Comparison of the LRU-stack model and the proposed reference model.

III. 참조 모델의 설계

본 논문이 제안하는 참조 모델은 시간 지역성과 데이터의 인기 편향성을 모두 고려하여 데이터의 참조 순서를 생성하는 모델로 그 기본적인 구조는 LRU 스택 모델과 유사하다. LRU 스택 모델의 경우 그림 1에서 보는 것처럼 총 n 개의 참조 대상 데이터를 위해 엔트리의 수가 n 인 스택의 각 엔트리에 참조 확률을 정의하며, 확률값은 스택의 낮은 위치로 갈수록 낮아진다. n 개의 데이터는 참조 시간 순서에 따라 스택의 최상위 엔트리부터 하나씩 위치하며, 스택에 주어진 참조 확률에 따라 새롭게 참조되는 데이터가 결정된다. 참조가 이루어진 데이터는 스택의 현재 엔트리에서 최상위 엔트리로 이동하며, 해당 데이터보다 상위 엔트리에 있던 데이터는 엔트리를 한 자리씩 아래로 이동한다. 이러한 방식으로 데이터의 참조 순서를 생성하여 최근에 참조된 데이터가 다시 참조될 가능성이 높은 성질을 반영한다.

본 논문의 참조 모델은 다음과 같이 정의된다. 데이터의 집합이 $D = \{d_1, d_2, \dots, d_n\}$ 으로 주어진다고 할 때, 참조 확률이 정의된 n 개의 엔트리 l_1, l_2, \dots, l_n 으로 구성된 스택이 존재하며 각 엔트리의 참조 확률은 LRU 스택 모델과 마찬가지로 스택의 낮은 위치로 갈수록 낮아진다. 각 엔트리에 정의된 확률의 총합은 1이며, 새로운 참조 데이터의 생성은 각 엔트리에 결합된 확률에 근거한다. 이때, 새롭게 참조된 데이터를 LRU 스택 모델처럼 스택의 최상위 엔트리로 이동시키는 것이 아니라 그 데이터의 참조가치 값에 근거해서 적정 엔트리로 이동시킨다.

참조가치 값은 시간 지역성과 인기 편향성을 복합적으로 고려하는 값으로 LRFU 및 LUV 알고리즘에서 제안한 데이터의 평가 방식과 유사한 방식으로 다음과 같이 정의된다.

정의 1. 데이터의 참조가치 값 정의

데이터 d 가 과거에 k 번 참조되었고 각각의 참조 시간이 t_1, t_2, \dots, t_k 라고 하자. 본 논문에서의 시간은 데이터가 참조될 때마다 1씩 증가하는 논리적 시간을 뜻한다. 그러면 데이터 d 의 현재 시간 t 에서의 참조가치 값 $Value(d)$ 는 다음과 같이 정의된다.

$$Value(d) = \sum_{i=1}^k F(t - t_i) \quad (1)$$

이 때, 함수 $F(x)$ 는 $F(x) = \left(\frac{1}{2}\right)^{\lambda x}$ ($0 < \lambda \leq 1$)로 정의된다.

LRU 스택 모델에서는 각 데이터의 가장 최근 참조 시점에 근거해서 스택 엔트리 상의 위치를 결정하므로 시간 지역성만을 반영할 수 있는 반면, 본 논문의 참조 모델에서는 참조 시점의 최근성과 참조횟수를 복합적으로 고려한 참조가치 값의 크기대로 데이터의 스택 내 위치가 결정되므로 시간 지역성과 데이터의 인기 편향성을 모두에 반영할 수 있다.

정리 1. $\lambda=1$ 인 경우 본 논문의 참조 모델은 LRU 스택 모델과 동일하다.

증명. 임의의 시점에 두 모델이 데이터를 스택 상의

동일 엔트리에 위치시킨다는 것을 보이면 된다. 즉, 데이터를 참조가치 값을 기준으로 정렬한 순서와 직전 참조 시점을 기준으로 정렬한 순서가 일치함을 보이면 된다. 임의의 두 데이터 x 와 y 에 대해 x 의 직전 참조 시점이 y 의 직전 참조 시점보다 더 최근이라고 하자. 그러면, x 의 참조가치 값이 y 의 참조가치 값보다 크다는 것을 보이면 된다. 한편, 임의의 i 와 $i \geq i+1$ 인 임의의 j 에 대해

$$\left(\frac{1}{2}\right)^i > \sum_{k=i+1}^l \left(\frac{1}{2}\right)^k \quad (2)$$

이 성립하므로 데이터 y 가 과거에 아무리 많이 참조되었다 하더라도 이보다 더 최근에 참조된 데이터 x 가 항상 더 큰 참조가치 값을 가지게 된다.

IV. 최적성 분석

본 장에서는 III장에서 제시한 참조 모델에 의해 생성된 데이터의 참조 순서에 대한 최적의 교체 알고리즘이 LRFU 알고리즘임을 증명한다. 이는 LRU 스택 모델에 최적인 알고리즘이 LRU 알고리즘임을 증명한 것과 유사한 방식을 따른 것이다^[4]. 데이터의 집합이 $D = \{d_1, d_2, \dots, d_n\}$ 이고, 주어진 \mathcal{A} 에 의한 참조 모델이 생성한 참조 데이터의 순서가 $r_1, r_2, \dots, r_t, \dots$ 이라 하자. 이때, $C_k(S, \text{Value})$ 를 참조 순서 $r_{t+1}, r_{t+2}, \dots, r_{t+k}$ 를 처리하는 동안 발생하는 캐시 미스의 최소 기대값이라 하자. 여기에서 '최소값'이 아닌 '최소 기대값'이라는 용어를 사용한 것은 본 논문의 최적성 분석이 확률적인 참조 모델 하에서 이루어지기 때문이다. 이때, S 와 $\text{Value} = \langle \text{Value}(d_1), \text{Value}(d_2), \dots, \text{Value}(d_n) \rangle$ 는 각각 시간 t 에서의 캐시 내용 S 와 데이터의 Value_t 값의 벡터 $\langle \text{Value}(d_1), \text{Value}(d_2), \dots, \text{Value}(d_n) \rangle$ 을 나타낸다. Value_x 가 데이터 x 가 참조된 후 새롭게 바뀐 Value 벡터를 나타낸다고 하면, 최적성(optimality)은 다음 식으로 표시될 수 있다.

$$\begin{aligned} C_0(S, \text{Value}) &= 0 \\ C_{k+1}(S, \text{Value}) &= \sum_{x \in S} p(x) C_k(S, \text{Value} \cdot x) \\ &+ \sum_{x \notin S} p(x) \{1 + \min_{z \in S} C_k(S+x-z, \text{Value} \cdot x)\} \quad (3) \end{aligned}$$

이 때, $p(x)$ 는 주어진 Value 벡터 하에서 데이터 x 의 스택 위치에 주어진 참조 확률을 나타낸다. 이제 LRFU 알고리즘이 최적의 알고리즘임을 보이기 위해서는 LRFU 알고리즘에 의한 캐시 운영이 위의 식을 만족함을 보이면 된다. 캐시 내에서 Value 값이 가장

작은 데이터를 d_{\min} 이라고 할 때, 이는 임의의 k ($k \geq 0$)에 대해 다음 식을 만족하는 것을 보이는 것으로 충분하다.

$$\begin{aligned} \min_{z \in S} C_k(S+x-z, \text{Value} \cdot x) \\ = C_k(S+x-d_{\min}, \text{Value} \cdot x) \quad (4) \end{aligned}$$

이는 d_{\min} 이 $S+d_{\min}$ 중에서 Value 값이 가장 작은 원소라 할 때 다음을 보이는 것과 동일하다.

$$C_k(S+d_{\min}, \text{Value}) - C_k(S+d, \text{Value}) \geq 0 \quad (5)$$

이 식이 $k=0$ 에서 성립하는 것은 자명하다. 이제 임의의 k ($k \geq 0$)에 대해 성립한다고 가정하고 $k+1$ 일 때에도 성립함을 보이겠다.

$$\begin{aligned} &C_{k+1}(S+d_{\min}, \text{Value}) - C_{k+1}(S+d, \text{Value}) \\ &= \sum_{x \in S} p(x) [C_k(S+d_{\min}, \text{Value} \cdot x) - C_k(S+d, \text{Value} \cdot x)] \\ &+ [p(d) - p(d_{\min})] \\ &+ p(d) \left[\min_{z \in S+d_{\min}} C_k(S+d_{\min}+d-z, \text{Value} \cdot d) \right. \\ &\quad \left. - C_k(S+d, \text{Value} \cdot d) \right] \\ &+ p(d_{\min}) [C_k(S+d_{\min}, \text{Value} \cdot d_{\min}) - \\ &\quad \min_{z \in S+d} C_k(S+d+d_{\min}-z, \text{Value} \cdot d_{\min})] \\ &+ \sum_{x \notin S+d+d_{\min}} p(x) [\min_{z \in S+d} C_k(S+d_{\min}+x-z, \text{Value} \cdot x) \\ &\quad - \min_{z \in S+d_{\min}} C_k(S+d+x-z, \text{Value} \cdot x)] \quad (6) \end{aligned}$$

첫 번째 항은 k 일 때 성립한다는 귀납법 가정에 의해 0 이상의 값이다. 두 번째 항은 Value 값이 큰 데이터가 스택 상에서 참조 확률이 더 높은 엔트리에 위치하므로 0 이상의 값이다. 세 번째 항은 가정에 의해 $z = d_{\min}$ 일 때 최소 비용을 가지므로 소거된다. 네 번째 항에서는 $z = d$ 인 경우 항 전체가 소거되므로 적어도 0 이상의 값을 가진다. 마찬가지로 다섯 번째 항 역시 0 이상의 값을 가지며, 따라서 전체 식도 0 이상의 값을 가진다.

V. 시뮬레이션

본 논문에서 제안한 참조 모델의 검증을 위해 본 장에서는 모델에 의해 생성되는 데이터 참조의 특성을 살펴보고 캐시 시뮬레이션을 통해 그 타당성을 입증한다. 그림 2는 엔트리 크기가 2이고 각 엔트리에 결합된 참조 확률이 0.7과 0.3인 경우 LRU 스택 모델과 제안한 모델로 데이터 참조를 100만회 생성했을 때 데이터의 참조 확률 변화를 보여주고 있다. 그림에서 보는 것처럼

럼 LRU 스택 모델의 경우 참조 확률이 0.7 또는 0.3 중 하나의 값을 가지며 지속적으로 진동하는 모습을 나타낸 반면 제안한 모델은 참조 확률이 시간에 따라 증감하면서 확률 변화가 매끄럽게 나타나는 것을 확인할 수 있다.

그림 3(a)는 상기의 참조 모델에 의해 데이터 참조를 생성한 경우 초기 참조 확률이 0.7인 데이터의 참조 확률 변화를 x 축 횟수만큼 독립 시행한 경우 그 평균 값의 변화를 표시한 그래프이다. 그림에서 보는 것처럼 많은 시행을 통해 해당 데이터의 참조 확률이 초기 참조 확률에 수렴하는 것을 확인할 수 있다. 그림 3(b)는 이런 방식으로 10,000번의 독립 시행 후 얻어진 데이터의 참조 확률값을 확률 구간별로 표시한 것이다. 그림에서 보는 것처럼 초기 참조 확률인 0.7이 가장 높은 빈도를 나타내었으며, 이를 중심으로 정규 분포 형태를 나타내었다.

그림 4는 모델에 의해 생성된 데이터 참조에 대해 IV장에서 최적성을 입증한 교체 알고리즘이 실제로 어떤 성능을 나타내는지를 보여주고 있다. 그림 4(a)는 그림 3(a)에 의해 생성된 데이터 참조에 대해 LRFU 알고리즘으로 캐시를 운영했을 때 캐시 적중률을 보여주고 있다. 그림에서 x 축은 시행 횟수를 나타내며 y 축은 해당 횟수만큼 캐시를 운영한 후 캐시 적중률의 평균을 보여주고 있다. 그림에서 보는 것처럼 캐시 적중률은 참조 모델에서 상대적으로 높은 확률을 가진 데이터의 참조 확률과 거의 유사한 결과를 얻은 것을 확인할 수 있다. 그림 4(b)는 그림 3(b)와 동일한 방법으로 10,000번 독립 시행으로 생성한 데이터 참조에 대해 LRFU 알고리즘으로 캐시를 운영한 결과 캐시 적중률의 분포를 보여주고 있다. 그림에서 보는 것처럼 캐시 적중률은 데이터 참조 확률과 동일한 0.7을 중심으로 정규 분포 형태를 나타낸 것을 확인할 수 있다.

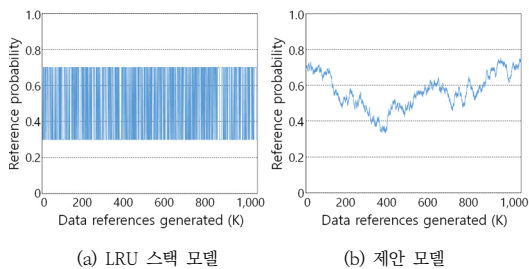


그림 2. 시간에 따른 참조 확률 변화
 Fig. 2. Changes in reference probability over time

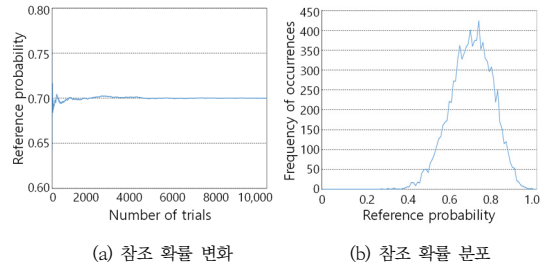


그림 3. 제안 모델에 의한 참조 확률 변화 및 분포
 Fig. 3. Reference probability changes and distributions based on the proposed model.

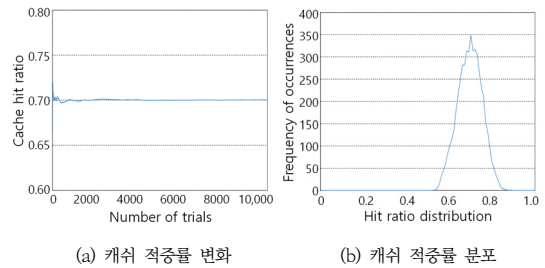


그림 4. LRFU에 의한 캐시 적중률 변화 및 분포
 Fig. 4. Cache hit ratio changes and their distributions based on LRFU.

VI. 결 론

본 논문에서는 시간 지역성과 인기 편향성을 동시에 반영하는 새로운 참조 모델을 제시하였다. 전통적인 참조 모델인 LRU 스택 모델의 경우 시간 지역성을 가지는 데이터의 참조 성향을 표현할 수 있는 반면 데이터의 서로 다른 인기도를 표현할 수 없고, 독립 참조 모델의 경우 데이터의 인기 편향성을 표현할 수 있지만, 시간에 따른 참조 성향의 변화를 나타낼 수 없었다. 본 논문이 제시한 참조 모델은 이러한 점을 극복하여 데이터의 서로 다른 인기도와 시간에 따른 참조 성향의 변화를 모두 반영할 수 있는 특징이 있다. 또한, 본 논문에서는 제안한 참조 모델에 최적인 캐시 교체 알고리즘에 대해 설명하고 그 최적성을 증명하였다. 최근 그 중요성이 날로 증가하고 있는 모바일 멀티미디어 데이터 및 인공지능 데이터 등에 나타나는 참조 성향이 시간 지역성과 인기 편향성을 모두 가지고 있어 본 논문의 참조 모델은 이와 같은 환경의 데이터 모델링에 효과적으로 이용될 수 있을 것으로 기대된다.

References

- [1] B. Kim, Y. Woo, and I. Kim, "Aggressive cooperative caching scheme in mobile ad hoc networks," *Journal of KIIT*, vol. 18, no. 1, pp. 41-48, 2020.
DOI: <https://doi.org/10.14801/jkiit.2020.18.1.41>
- [2] S. Park, "RFJ: A reliable and fast journaling mechanism," *Journal of the Korea Academia-Industrial cooperation Society(JKAIS)*, vol. 20, no. 7, pp. 45-51, 2019.
DOI: <https://doi.org/10.5762/KAIS.2019.20.7.45>
- [3] S. Lee, H. Bahn, and S. H. Noh, "CLOCK-DWF: A write-history-aware page replacement algorithm for hybrid PCM and DRAM memory architectures," *IEEE Trans. Computers*, vol. 63, no. 9, pp. 2187-2200, 2014.
DOI: <https://doi.org/10.1109/TC.2013.98>
- [4] E. G. Coffman and P. J. Denning, *Operating Systems Theory*, Prentice-Hall, 1973.
- [5] V. Almeida, A. Bestavros, M. Crovella, and A. Oliveira, "Characterizing reference locality in the WWW," 4th IEEE Conf. Parallel and Distributed Information Systems, pp. 92-103, 1996.
DOI: <https://doi.org/10.1109/PDIS.1996.568672>.
- [6] M. Arlitt and C. Williamson, "Web Server Workload Characterization: The Search for Invariants," *ACM SIGMETRICS Conference*, pp.126-137, 1996.
DOI: <https://doi.org/10.1145/233008.233034>
- [7] G. S. Gao, "Performance Analysis of Cache Memories," *Journal of the ACM*, vol. 25, no. 3, pp.378-395, 1978.
DOI: <https://doi.org/10.1145/322077.322081>
- [8] E. O'Neil, P. O'Neil, and G. Weikum, "An Optimality Proof of the LRU-K Page Replacement Algorithm," *Journal of the ACM*, vol.46, no.1, pp.92-112, 1999.
DOI: <https://doi.org/10.1145/300515.300518>
- [9] D. Lee, J. Choi, H. Choe, S. Noh, S. Min, and Y. Cho, "Implementation and performance evaluation of the LRFU replacement policy," 23rd Euromicro Conf. New Frontiers of Information Technology, pp. 106-111, 1997.
DOI: <https://doi.org/10.1109/EMSCNT.1997.658446>
- [10] H. Bahn, S. Noh, S. Lyul, and K. Koh, "Efficient replacement of nonuniform objects in Web caches," *IEEE Computer*, vol. 35, no. 6, pp. 65-73, 2002.
DOI: <https://doi.org/10.1109/MC.2002.1009170>.

저 자 소 개

반 효 경(정회원)



- 1997년 2월 : 서울대학교 계산통계학과 학사
- 1999년 2월 : 서울대학교 전산과학과 석사
- 2002년 2월 : 서울대학교 컴퓨터공학부 박사.
- 2002년 9월 ~ : 이화여자대학교 컴퓨터공학과 교수.
- 주관심분야 : 운영체제, 스토리지시스템, 임베디드시스템

※ This work was supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No.2021-0-02068, Artificial Intelligence Innovation Hub).