

인간 기억 인출 과정을 응용하여 설계된 ACT-R 기반 페이지 교체 정책

노 흥 찬[†] · 박 상 현^{††}

요 약

자주 접근되는 데이터에 대해서 빠른 접근을 보장하기 위해 사용되는 임시저장소인 캐쉬는 컴퓨터 시스템 내에서 다양한 계층에 존재하며, 이러한 캐쉬 저장 공간 내에서 효율적으로 데이터를 관리하기 위해 다양한 페이지 교체 알고리즘들이 연구되어 왔다. 대부분의 페이지 교체 알고리즘들은 얼마나 최근에 데이터가 접근 되었는가 또는 얼마나 자주 접근되었는가를 바탕으로 향후 다시 접근될 것 같은 데이터들을 캐쉬 안에 유지하는 휴리스틱 방법을 취하고 있다. 이러한 컴퓨터 내에서의 데이터의 인출 과정은 인간의 기억 인출 과정과 유사하며, 인간의 기억 인출 과정 역시 캐쉬 구조처럼 기억이 얼마나 최근에 그리고 자주 인출되었는가에 의해 그 기억 인출의 확률과 인출속도가 결정된다는 것이 최근 연구에 의해서 밝혀진바 있다. 본 연구에서는 인간의 기억 인출 과정에서의 과거 해당 기억에 대한 인출 기록들의 최근성과 빈도가 인출 확률에 영향을 미치는 관계를 분석하고 이를 페이지 교체 알고리즘에 응용하여 기존의 페이지 교체 알고리즘의 성능을 개선하였다. 또한, 실험을 통해 제안하는 페이지 교체 알고리즘이 기존에 데이터베이스 버퍼 캐쉬에서 가장 좋은 성능을 보이는 것으로 알려진 LRFU보다 파라미터에 민감하지 않고 우수한 성능을 보인다는 것을 입증하였다.

키워드 : 페이지 교체 정책, LRU, LRFU, LFU, ACT-R, 인간 기억 인출

A novel page replacement policy associated with ACT-R inspired by human memory retrieval process

Hongchan Roh[†] · Sanghyun Park^{††}

ABSTRACT

The cache structure, which is designed for assuring fast accesses to frequently accessed data, resides on the various levels of computer system hierarchies. Many studies on this cache structure have been conducted and thus many page-replacement algorithms have been proposed. Most of page-replacement algorithms are designed on the basis of heuristic methods by using their own criteria such as how recently pages are accessed and how often they are accessed. This data-retrieval process in computer systems is analogous to human memory retrieval process since the retrieval process of human memory depends on frequency and recency of the retrieval events as well. A recent study regarding human memory cognition revealed that the possibility of the retrieval success and the retrieval latency have a strong correlation with the frequency and recency of the previous retrieval events. In this paper, we propose a novel page-replacement algorithm by utilizing the knowledge from the recent research regarding human memory cognition. Through a set of experiments, we demonstrated that our new method presents better hit-ratio than the LRFU algorithm which has been known as the best performing page-replacement algorithm for DBMS caches.

Keywords : Page Replacement Policy, LRU, LRFU, LFU, ACT-R, Human Memory Retrieval

1. 서 론

자주 접근되는 데이터에 대해서 빠른 접근을 보장하기 위해 사용되는 임시저장소인 캐쉬는 컴퓨터 시스템 내에서 다양한 계층에 존재하며 그 예로는 CPU 캐쉬, Disk 캐쉬, Database Buffer 캐쉬 등이 있다. 하지만 제약된 캐쉬의 저

※ 이 논문은 2008년 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(KRF-2008-313-D00849).

† 준 회원 : 연세대학교 컴퓨터과학과 박사과정

†† 종신회원 : 연세대학교 컴퓨터과학과 부교수(교신저자)

논문접수 : 2010년 4월 6일
수정일 : 1차 2010년 9월 29일
심사완료 : 2010년 9월 29일

장 공간 내에서 가장 효율적으로 데이터를 관리하기 위해 사용되는 캐쉬 페이지 교체 알고리즘들은 이들 계층에서 모두 공통적으로 사용되고 있다.

이러한 페이지 교체 알고리즘의 이상적인 알고리즘은 Belady가 제안한 OPT[1] 알고리즘처럼 미래에 가장 오랫동안 사용되지 않을 페이지를 교체하는 것이며, 하지만 이러한 이상적인 교체 대상 페이지는 페이지 교체 시점에서는 알 수 없으며 모든 연산이 수행된 후에야 특정 과거 시점에서 가장 좋은 교체 대상 페이지에 대해서 확인할 수 있다는 모순이 있다. 그러므로 이러한 알고리즘은 online으로 수행될 수 없으며 실질적으로 적용되기에는 불가능하다. 그러므로 모든 교체 알고리즘들은 과거의 데이터에 대한 접근 기록을 기준으로 가장 앞으로 사용되지 않을 것 같은 페이지를 교체하는 휴리스틱 접근 방법을 취한다.

대표적인 캐쉬 페이지 교체 알고리즘들로는 LRU[2], LFU[2], LRFU[3] 등이 있다. 이들은 모두 과거의 데이터에 대한 접근 기록에 의존하여 가장 앞으로 사용되지 않을 것 같은 페이지를 교체하는 전략을 취하며, 그러기 위해 과거에 접근된 횟수, 접근된 후로부터 지난 시간 등을 페이지 교체의 기준으로 적용하여 작동한다. [2]는 이들 대표적인 페이지 교체 알고리즘들 중 과거의 접근된 횟수와 접근된 후로부터 지난 시간을 모두 고려하는 LRFU를 실제 환경에서 가장 효율적인 알고리즘이라고 언급하였다.

데이터에 대한 컴퓨터의 캐싱 정책은 인간의 기억 인출 과정과도 밀접한 연관이 있으며, 인지과학 및 인지심리학 등에서 기존에 연구해온 이슈들과 공통적인 부분을 많이 가지고 있다.

Anderson et al.에 의해 처음으로 1998년에 제안된 ACT-R은 인간의 기억 인출과정에 대해 기존 연구들을 포괄적으로 정리한 대표적인 인지 모델로서 그 후로 지속적으로 발전되어 2004년에 ACT-R ver5.0[4]이 발표되었다. ACT-R은 인간의 기억 인출 과정에서 기억 인출 속도와 확률을 결정하는 Activation level 이론을 제안하였고, 이러한 activation level은 컴퓨터의 캐쉬와 마찬가지로 특정 데이터에 대한 과거의 기록으로부터 현재의 해당 데이터의 인출 확률을 기반으로 결정되며, 인간 기억의 인출 속도와 인출 확률을 제어한다. 이러한 activation 이론은 ACT-R 이론의 가장 핵심을 이루는 부분으로서 ACT-R의 많은 응용 연구들에 의해서 실제 인간의 기억 메커니즘과 일치한다는 증거가 발견되었다.

특히, 전체 Activation 이론 중 핵심인 Base-level activation 이론은 동일 저자인 Anderson의 1991년도 연구인 [5]에서 처음 제안되었으며, 인간의 과거 기억의 빈도 및 각 기억이 인출된 후 지난 시간으로부터 현재 다시 인출될 확률을 계산하며 그것을 그대로 Base-level activation 수식으로 표현하였다. 또한 이를 실제 환경에서 측정된 통계학적인 데이터와 검증하여, 환경적인 조건에 최적으로 적용된

인간의 기억 메커니즘이라고 하였다. [5]에 따르면 일반 환경에서의 데이터들도 과거의 데이터의 출현 frequency (frequency effect)와, recency (recency effect), 그리고 각 그 data의 출현 사이의 간격(spacing effect)에 따라 현재 데이터가 다시 출현할 확률이 결정된다고 하며, [5]는 이를 수식화 하여 정리하였고 그대로 이를 인간의 기억 인출 메커니즘에 적용하여, 기존 연구들이 실패한 다양한 인간의 기억 인출 과정 현상에 대해서 설명하였다.

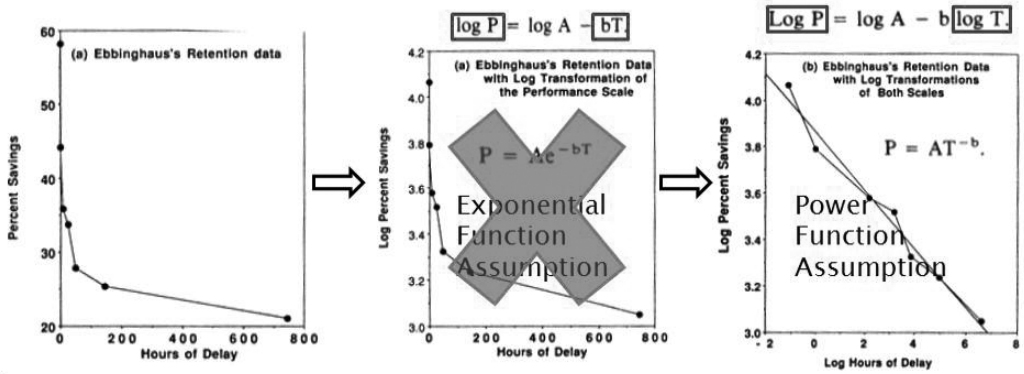
이러한 ACT-R과 [5]의 base-level activation은 Belady가 제안한 이상적인 알고리즘에 근접한 성능을 낼 수 있는 휴리스틱 방법을 설계하는데 활용될 수 있다. Belady가 제안한 것처럼 미래에 가장 오랫동안 사용되지 않을 페이지를 교체하는 정책은 현재 다시 인출될 확률이 가장 낮은 페이지를 교체하는 방법으로 근사화(approximation)될 수 있으며, 이는 base-level activation을 캐쉬에 존재하는 각 페이지에 적용하여 계산한 후 이들 중 activation level이 가장 낮은 페이지를 교체하는 것과 동일하다.

기존의 LRU, LFU는 frequency와 recency 중 하나에만 의존하여 설계된 문제점이 있으며, Frequency와 recency 둘 모두를 고려하여 설계된 LRFU 역시 recency 효과를 고려하는 데 있어서 지수 함수를 적용하여 시간에 따라 특정 페이지가 캐쉬에 있을 확률을 급격하게 감소시키는 문제점이 있다.

또한, ACT-R은 인지과학 분야의 대표적인 인간 기억 인출과정에 대한 연구로서, 컴퓨터과학의 다양한 분야에 적용되어 기존 컴퓨터과학 기술만으로 해결하지 못했던 문제를 푸는데 기여해왔다. 대표적으로, 로봇 및 소프트웨어 에이전트, HCI분야 등이 있으며 최근 연구로 로봇의 비전 제어 플랫폼을 위해 ACT-R을 응용한 [9] 및 인간의 멀티태스킹 인지과학적 모델링을 HCI에 적용하기 위해 ACT-R을 응용한 [10]이 있다.

이에 본 연구에서는 ACT-R의 base-level activation의 접근방법을 캐쉬 페이지 교체 정책에 적용하여 인간의 기억 메커니즘과 유사하면서도 기존 페이지 교체 알고리즘들보다 효율적인 페이지 교체 정책을 제안하고자 한다. Base-level activation에서 제안된 frequency, recency, spacing 효과 모두를 고려한 수식을 페이지 교체 알고리즘의 교체 기준이 되는 척도로 변형하여 적용하며 기존의 LRFU의 문제인 recency 효과와 관련된 지수 함수(exponential function)를 ACT-R에서처럼 멱함수(power function)로 적용하여 페이지 교체 알고리즘을 설계하고 이를 실제 데이터베이스 환경에서의 workload를 이용하여 실험 해보려고 한다.

본 논문은 다음과 같이 구성되어 있다. 2장에서 관련 연구로서 ACT-R과 LRFU에 대해서 소개하고 3장에서 제안하는 페이지 교체 알고리즘에 대해서 기술한다. 4장에서 실험결과를 제시한 뒤, 5장에서는 결론을 맺으며 및 향후 연구를 제안한다.



(그림 1) Anderson의 멱함수 가정

2. 관련 연구

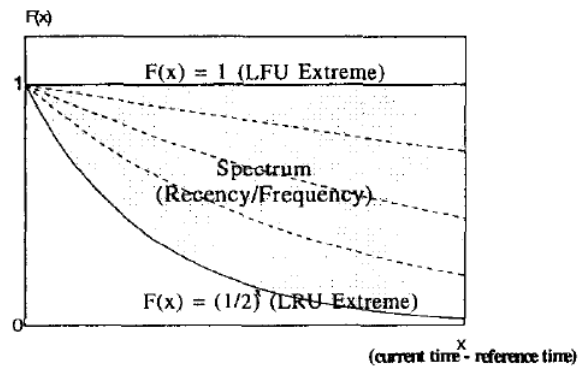
LRFU는 데이터베이스에서 수행되는 워크로드의 특성을 분석해본 결과, 특정 페이지가 다시 접근될 확률은 과거 해당 페이지에 대한 접근의 recency 또는 frequency와 연관이 있지만 그 영향 크기가 캐쉬에 할당된 메모리의 크기에 따라 동적으로 변한다는 사실을 밝혀냈다. 이러한 사실을 기반으로 접근의 recency만을 고려한 LRU 또는 접근의 frequency만을 고려한 LFU는 최적의 페이지 교체 알고리즘이 될 수 없다고 주장하였다.

이러한 이유로 LRFU는 LRU와 LFU의 특징을 모두 반영하여 접근의 recency와 frequency를 모두 고려한 페이지 교체 알고리즘을 제안하였다. 페이지 교체 대상을 선정하기 위해 다음과 같은 식에 의해 개별 페이지의 활성도를 계산하고 그 활성도가 가장 낮은 페이지를 캐쉬에서 쫓아내는 방식을 취하였다.

$$C_{t_n}(p) = \sum_{i=1}^k F(t_n - t_i) \quad (1)$$

식 (1)에서 $C_{t_n}(p)$ 는 특정 페이지 p의 활성도 값을 나타내는 것이고 이러한 특정 페이지의 활성도는 해당 페이지에 대한 과거의 접근 시간들(t_i)과 현재 시간(t_n)의 차이에 의해 계산된다. 총 접근횟수가 k개인 각각의 접근 기록에 대해서 현재 시간과 해당 과거의 접근 시간의 차이 값을 특정 함수($F(x)$)에 입력하여 그 결과 값을 계산하고, 해당 페이지의 모든 접근 기록들에 대해 계산된 값을 더하여 전체 활성화 값을 계산하게 된다. 여기서 이용되는 특정 함수는 다음과 같이 정의된다.

$$F(x) = \left(\frac{1}{2}\right)^{\lambda x} \quad (2)$$



(그림 2) LRFU 특정함수의 λ 에 따른 변화

식(2)는 파라미터 λ 에 의해 조절되는 지수함수로 각각의 접근 기록들의 recency를 반영하여 과거의 접근 기록이 오래되면 오래될수록 함수 값이 급격하게 감소하는 성질을 가진다. 다시 전체 식을 해석해보자면, (2)의 경우 접근의 recency를 활성화 값에 반영하기 위해 지수함수를 사용한 것이고, (1)에서 접근 기록들의 수열의 합을 취한 것은 frequency에 대한 효과를 활성화 값에 반영하기 위해 사용된 것으로 볼 수 있다.

식 (2)의 값은 파라미터 λ 에 의해서 (그림 1)과 같이 변화된다. 식 (2)는 λ 가 1인 경우, (그림 1)의 가장 아래쪽 실선으로 표시된 $1/2$ 에 대한 지수 함수가 되어 활성화 값에 frequency보다는 recency가 압도적으로 반영되게 된다. 이러한 성질은 다음과 같은 $1/2$ 를 지수로 하는 지수함수의 속성에 근거한다.

$$F(i) > \sum_{j=i+1}^{\infty} F(j) \quad (3)$$

식 (3)에 나타난 바와 같이 λ 가 1인 경우, 현재로부터 i 번째 이전의 단일 접근에 대한 함수 값은 $(i+1)$ 번째 이전의 모든 접근에 대한 함수의 값들의 합보다 크게 되므로, 한 time unit 차이로 더 최근에 접근된 페이지는 그전에 아무리 많

이 접근된 페이지가 있을지라도 활성화 값이 항상 크게 된다. 그러므로 frequency보다는 recency만이 고려되어 페이지 교체 정책이 수행되는 것이고 이는 LRFU가 LRU 정책과 일치하는 동작을 보이게 한다.

이에 반해 식(2)에서 λ 가 0인 경우, (그림 1)의 가장 상위에 표시된 실선으로 표시된 상수 함수가 되어 recency를 나타내는 x축에 상관없이 함수 값이 일정한 LFU에 근접한 성질을 보이게 된다. x 축에 표시된 과거 접근시간과 현재 시간의 차이가 얼마가 될지라도 F(x)의 값은 항상 1이 되므로 접근의 recency는 무시되고 접근의 frequency만 활성화 값에 기여하게 된다. 그리하여 이러한 경우 LRFU는 LFU와 동일한 페이지 교체 알고리즘으로 동작하게 된다. 이러한 LRFU의 성질이 (그림 1)에 잘 표시되어 있으며 0과 1사이의 λ 에 따라 LRFU가 어떻게 변화하는지도 (그림 1)의 중앙에 표시된 점선에 의하여 알 수 있다. LRFU는 이러한 LRU와 LFU 두 극단에서 λ 값의 조절을 통해 얼마나 recency와 frequency를 반영할지를 결정할 수 있게 된다.

ACT-R은 사람의 인지과정을 모델링하기 위해 설계된 인지과학 아키텍처로서 Anderson에 의해 1998년에 제안되었으며 그 후로 여러 연구자들에 의해 보완되고 개선되어 계속 발전되어 왔으며 2004년에는 ACT-R 5.0[4]이 발표되었다. Anderson은 그 자신 및 기존의 많은 인지과학 및 인지심리학, 뇌과학자들이 밝혀낸 인간 인지 과정의 부분적인 지식들을 하나의 통합된 시스템으로 만들고자 하였고, 이를 컴퓨터 언어로 구현하여 보다 현실적인 환경에서 실제로 외부 세계와 상호 작용을 하는 시스템을 구축하고자 하였다.

이렇게 인지과학 모델이 성공적으로 컴퓨터 언어로 구현됨에 따라 더 많은 연구자들이 이를 응용한 연구를 수행할 수 있게 되었고 이를 통해 ACT-R 이론의 많은 부분들이 검증되어 왔다.

Anderson은 [4]에서 인간의 장기 기억 중 서술 기억(declarative memory)을 인출하는 데 있어서 사용되는 activation level 공식을 많은 응용 연구들에서 가장 성공적으로 검증된 이론이라고 언급하였다. 이러한 activation level은 서술 기억의 인출 과정에 있어 기억 인출 속도와 확률을 결정하며 이 이론을 구성하는 핵심 요소인 base-level activation 이론은 Anderson이 1991년에 발표한 [5]에 근거한다.

[5]에서 Anderson은 인간의 기억 인출 과정을 설명하는 recency, frequency, spacing 효과가 기존의 기억 인출 과정을 설명하는 다른 이론들에 의해 완전하게 설명되지 못한다는 문제점을 제기하였고, 이를 해결하기 위해 특수한 가정을 하였다. 먼저 Anderson은 인간이 환경적인 조건에 최적으로 반응하도록 여러 세대를 거쳐서 진화했다고 가정하며, 그러므로 인간의 기억 인출 및 저장 과정을 알아보기 위해서는 인간의 기억에 영향을 미치는 환경적인 조건이 어떠한가를 분석해봐야 한다고 주장하였다. 그리하여 환경적인 조

건을 분석해본 결과 최근에 접근된 객체일수록 그리고 자주 접근된 객체일수록 향후에 다시 접근될 확률이 높다는 것을 여러 실험을 통해서 밝혀내었다. 또한 그러한 recency 효과가 시간에 따라 멱함수의 형태로 감소한다는 것, 그리고 frequency 효과는 단순히 그러한 recency 효과를 더하는 효과만이 있다는 것을 밝혀내었다.

더불어, 기존의 연구자들이 기억의 recency 효과에 대해서 recency 효과가 감소하는 비율이 기억의 마지막 인출 시간으로부터 지수함수와 유사하게 감소한다는 주장을 했던 것에 대해서 지수함수가 아니라 멱함수에 더 가깝게 감소한다는 사실을 밝혀내었다. 이는 (그림 2)에 잘 나타나 있다. 인간 기억의 유지 비율(P)을 시간(T)에 따라 실제 측정된 실험 결과를 이용하여 P를 각각 T에 대한 지수함수와 멱함수를 표현해보고 얼마나 두 함수가 실제 실험결과에 잘 맞는가를 비교해본 것이다. 이를 위해 지수 함수와 멱함수 가정에 의해 P를 T에 대한 수식으로 나타낸 후, 양 변에 log를 취해보았다. 지수함수의 경우, 그 결과로 log P와 T간에 선형적인 관계가 성립해야하므로 실험결과에서 P만을 log scale로 변형해서 다시 그래프를 그려보았다. 그 결과 변형된 그래프는 (그림 2)의 가운데 그래프처럼 예상과 달리 선형적이지 않은 결과가 나왔다. 이를 토대로 Anderson은 지수함수 가정이 틀렸음을 설명하였다. 반대로 멱함수의 경우 log를 양변에 취한 수식에 의해 log P와 log T의 선형적인 관계가 형성되어야 하는데, 실험결과와 P와 T 모두를 log scale로 변형하여 나타내본 결과, (그림 2)의 가장 오른쪽 그림처럼 선형적인 성질이 근사하게 나타남을 확인할 수 있었다. 이를 근거로 Anderson은 인간 기억의 recency 효과가 지수함수가 아닌 멱함수 성질을 가진다는 사실을 밝혀냈다.

이러한 결과를 종합하여 Anderson은 다음과 같은 base-level activation 이론을 제시하였다.

$$S = A \sum_{i=1}^k s(t_n - t_i) \tag{4}$$

$$s(x) = x^{-d} \tag{5}$$

S가 하나의 특정 기억에 대한 활성도를 나타내는 것이고 이는 식 (4), (5)에 의해 정의된다. t_n 은 현재의 시간이고 t_i 는 과거의 해당 기억에 접근 시간을 나타낸다. 그러므로 수식은 총 k개의 기억 인출에 대한 각 $s(x)$ 값의 수열이 된다. A는 scaling factor를 조절하는 파라미터이며 d 역시 recency의 감소 정도의 급격함을 조절하는 파라미터이다.

3. Base level activation 기반 페이지 교체 알고리즘

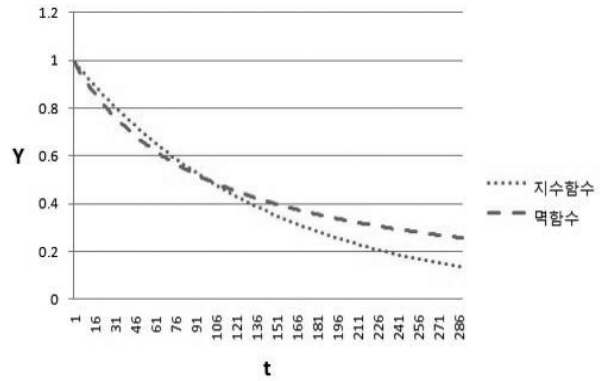
특이하게도, LRFU의 개별 페이지에 대한 활성화 함수와 ACT-R의 개별 기억에 대한 활성화 함수인 식(1)과 식(4)가

각각 지수 함수와 멱함수로 다른 것을 제외하고는 거의 일치한다. LRFU가 현재까지 가장 효율적으로 동작하는 페이지 교체 알고리즘인 것을 고려할 때 이러한 둘 간의 유사성은 [5]에서의 Anderson의 가정을 되돌아보게 한다. Anderson의 가정처럼 인간은 환경적인 조건을 인간의 기억 모델에 최적으로 반영했을 때 환경에 가장 최적으로 반응할 수 있으며, 이와 마찬가지로 컴퓨터의 경우에도 역시 환경적인 조건을 제대로 반영하게 되면 가장 효율적인 페이지 교체 정책이 설계될 수 있다.

하지만, 식 (2)와 식 (5)의 차이는 무시할 정도가 못 되므로 오랜 시간 동안 환경에 최적으로 적응하도록 진화해온 인간의 기억 모델을 컴퓨터의 캐쉬 페이지 교체 알고리즘을 개선하는데 적용해볼 여지가 있다. 또한 Anderson은 [5]에서 직접적으로 환경적인 조건을 분석하여, 특정 객체 출현의 recency 효과가 해당 객체가 다시 출현할 확률에 미치는 영향은 시간에 따라 멱함수의 형태로 감소한다는 것을 밝혀내었고 이를 인간의 기억모델을 추론하는 데 사용하였다. 그러므로 이는 인간의 기억모델 뿐만이 아니라 환경적인 조건에서의 recency 효과가 모두 멱함수의 형태로 감소함을 시사한다. 이러한 Anderson의 연구 결과에 의거할 때 페이지 교체 정책 역시, 최적의 성능을 내기 위해서는 환경적인 조건을 제대로 반영해야 하므로, 식 (2)와 같은 지수함수가 아닌 식 (5)와 같은 멱함수를 recency 효과를 구성하는데 사용하는 것이 더 바람직함을 알 수 있다.

또한 recency와 frequency를 실제로 반영하는 측면에서 두 함수를 비교해 볼 때에도 지수함수보다는 멱함수가 두 성질 모두를 더 잘 반영해주는 것을 알 수 있다. (그림 3)은 지수 함수와 멱함수를 비교한 것이다. 멱함수의 경우 $Y=(\lambda t+1)^{-d}$ 을, 지수함수의 경우 $Y=(1/2)^{t/d}$ 을 사용하였다. λ 는 모두 0.01을 사용하였고 멱 함수의 경우 d 를 1로하였다. 곡선의 모양새를 분석하여 보면 멱함수가 지수함수에 비해 초반에는 급격하게 감소하고 나중으로 갈수록 천천히 감소하는 것을 알 수 있다. recency 효과는 가장 최근일수록 급격하게 커지는 것이 바람직하며 이러한 면에서 지수함수보다 멱함수가 보다 나은 recency 특징을 가졌다고 할 수 있다. 또한 frequency 효과에 있어서는 과거의 인출 기록이 현재의 인출 확률에 영향을 미치기 위해서 recency 함수가 장기간으로 갈수록 감소되는 비율이 작아져야 한다. 그러한 면에서 멱함수는 지수함수보다 천천히 감소되므로 frequency 효과에 더 유리하다고 할 수 있다. 또한, 식 (5)처럼 멱함수를 식 (1)의 $F(x)$ 로 사용했을 경우에, d 가 1일 때에는 식 (1)이 조화수열의 합이 되어 식 (3)이 더 이상 성립하지 않게 된다. 이는 과거의 인출 기록들이 활성화 값에 더 많은 영향을 미치게 되는 것을 의미한다.

이를 바탕으로 제안하는 ACTR 페이지 교체 알고리즘의 활성화 값을 설계해보면 다음과 같이 표현할 수 있다.



(그림 3) 지수함수 멱함수의 비교

$$C_{t_n}(p) = \sum_{i=1}^k F(t_n - t_i) \quad (5)$$

$$F(x) = (\lambda x + 1)^{-d} \quad (6)$$

base-level activation의 수식처럼 $F(x)$ 를 멱함수를 사용하되 LRFU에서처럼 시간의 scaling 파라미터인 λ 를 수식에 추가하였다. λx 대신 $(\lambda x + 1)$ 을 사용한 이유는 1이 없을 경우, 컴퓨터 연산 시에 divide by zero의 오류가 발생할 수 있으며, 0과 1사이에서 멱함수는 ∞ ~1사이의 범위의 값을 가지므로 활성화 값의 범위를 지수함수와 같이 1~0사이로 제한하기 위함이다. 식 (5)는 식 (1)과 동일하다.

[알고리즘 1] ACTR의 페이지 교체 알고리즘

```

Input  $t_c$  (current time),  $p$  (requested buffer page)
IF  $p$  exists in the buffer cache THEN
     $C_{t_c}(p) = \sum_{i=1}^k F(t_c - t_i(p))$ 
    FOR  $i=k$  to 2 DO
         $t_i(p) = t_i - 1(p)$ 
    END FOR
     $t_1(p) = t_c$ 
    Update the heap structure by reflecting  $C_{t_c}(p)$ 
ELSE IF buffer is full THEN
    victim = the root of the heap structure
    IF the victim is dirty THEN
        write-back the victim to the disk
    END IF
    fetch the missed page  $p$  from the disk
     $C_{t_c}(p) = F(t_c)$ 
ELSE
    fetch the missed page  $p$  from the disk
     $C_{t_c}(p) = F(t_c)$ 
END IF
    
```

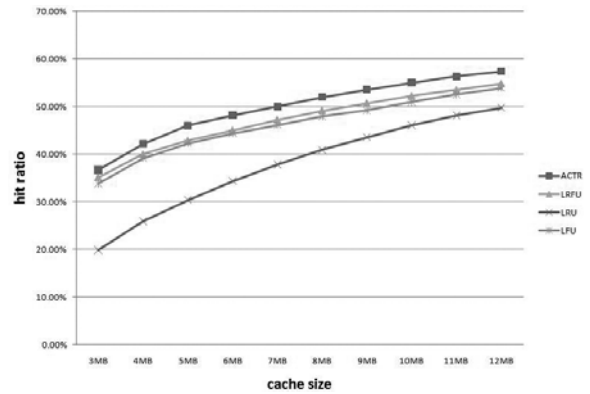
본 논문이 제안하는 새로운 페이지 교체 정책인 ACTR은 식(5)에서 기술한 새로운 활성화 값을 바탕으로 버퍼 미스(buffer miss) 발생 시 가장 활성화 값이 낮은 페이지를 버퍼 캐쉬(buffer cache)에서 쫓아내며 디스크로부터 해당 버퍼를 읽어온다. 이러한 ACTR의 페이지 교체 알고리즘을 알고리즘 1에 기술하였다. ACTR은 [알고리즘 1]에 기술된 바와 같이 요청된 버퍼가 버퍼 캐쉬에 존재할 경우, 식(5)로 표현되는 해당 버퍼의 활성화 값을 현재 시간(t_c)를 반영하여 업데이트 하며, 업데이트된 버퍼의 활성화 값을 토대로, 버퍼 캐쉬에 존재하는 페이지들의 활성화 값으로 구성되는 힙구조(heap structure)를 업데이트 한다. 또한 해당 버퍼의 접근 시간 리스트 $\{t_i(p) | 1 \leq i \leq k\}$ 에 현재 시간(t_c)을 추가한다. 반대로, 버퍼 캐쉬에 요청된 버퍼가 존재하지 않을 경우, 힙구조의 루트에 있는 가장 활성화 값이 낮은 버퍼 페이지를 디스크로 출력하고 요청된 버퍼를 디스크로부터 읽어온다. 앞서 언급된 힙구조는 최소힙(min heap)을 사용하여, 힙구조의 루트노드가 항상 가장 작은 활성화 값을 가지는 페이지를 참조하도록 한다.

4. 실험 결과

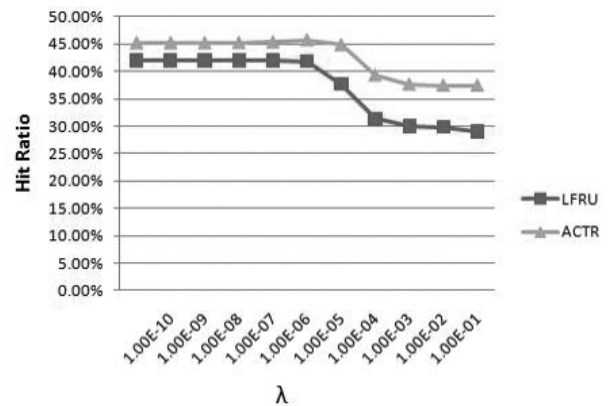
이러한 새로운 활성화 값을 기반으로 하는 페이지 교체 알고리즘을 ACTR이라 하고 이를 바탕으로 데이터베이스 버퍼 캐쉬에서 페이지 교체를 수행하는 실험을 수행해보았다.

실험을 위해 workload로서 OLTP(OnLine Transaction Processing) 데이터베이스 환경에서 대표적 벤치마크인 TPC-C[6]를 선택하였으며, MYSQL 데이터베이스를 사용하여 트레이스 기반 시뮬레이션 실험을 진행하였다. TPC-C 생성 대표적인 공개 소스 툴인 HAMMERORA [8]를 사용하여 TPC-C 입력 파라미터로서 다음 항목들을 사용하였다 (SCALE FACTOR: 1, 사용자 수: 10, 사용자당 트랜잭션 수: 1000개). MYSQL의 경우 데이터베이스의 버퍼의 간섭을 최소화한 트레이스를 얻기 위해 버퍼 풀(buffer pool) 크기를 최소 크기인 1MB로 설정한 후 트레이스를 수집하였다. 트레이스는 DiskMon [7]툴을 사용하여 저장장치 드라이버 수준에서 저장장치에 대한 읽기, 쓰기 작업에 대해 모두 수집하였다. 전체 트레이스에서 약 30만개의 읽기 작업을 추출해냈다. 이렇게 얻은 디스크 접근 데이터를 기반으로, 제안하는 페이지 교체 정책인 ACTR과 비교하고자 하는 페이지 교체 정책들인 LRFU, LRU, LFU에 대해서 시뮬레이션 기반으로 실험을 진행하였고 hit ratio를 측정해보았다.

λ , d , k 를 각각 $4e-5$, 1, 60으로 고정시키고 cache의 크기에 따른 각 알고리즘의 성능을 측정해보았고 이는 (그림 4)와 같이 나타났다. 캐쉬 크기에 비례하여 모든 알고리즘들이 일정하게 성능이 증가하였으며, 캐쉬 크기에 관계없이 제안하는 ACTR이 타 캐쉬 알고리즘들에 비해 가장 좋은



(그림 4) 캐쉬 크기에 따른 성능



(그림 5) lambda에 따른 LRFU와 ACTR의 성능 변화

hit ratio를 보였다. ACTR은 TPC-C 워크로드에 있어 전체 DBMS 크기의 5%~6%에 해당하는 DBMS cache를 사용했을 때, LRFU, LRU, LFU 각각에 대해서 평균적으로 3.17%, 14.71%, 3.77% 성능 향상을 보이는 것으로 나타났다. 이러한 ACTR 페이지 교체 정책의 성능 우위가 DBMS 캐쉬 크기 변화에 따라 계속 유지되는 것을 보였다. 본 실험에서 기존에 DBMS cache에서 가장 좋은 성능을 보이는 것으로 알려진 LRFU가 LFU에 비해서 DBMS cache 크기와 상관없이 평균적으로 1.02%의 hit ratio 성능 우위를 가진데 비해, ACTR의 경우는 LRFU에 비해 2.68%, LFU에 비해 3.7%의 성능 우위를 가진다.

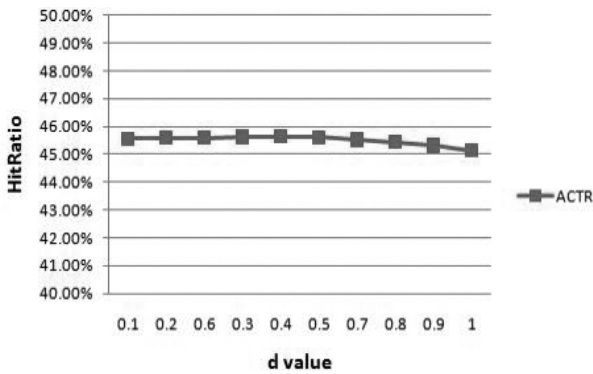
또한 각 파라미터가 알고리즘 성능에 미치는 영향을 분석해 보기 위해 lambda와 d에 따른 성능도 측정해 보았고 이에 대한 결과를 (그림 5와 6)에 나타내었다.

캐쉬 크기를 5MB로 d를 1로 고정하고 lambda에 따른 성능 변화에 대해서 실험한 결과 (그림 5)와 같이 나타났다. LRFU의 경우 lambda에 비해 매우 민감하게 성능이 변화하였으며 최대 성능과 최소 성능의 변동 폭이 약 13%까지 벌어지는 결과를 보였다. 그와 반대로 ACTR의 경우 상대적으로 LRFU에 비해 lambda에 덜 민감하게 성능이 변화하였으며 최대 변동 폭이 약 8%로 나타났다. 또한 ACTR은 lambda에 상관없이

이 LRFU에 비해서 좋은 성능을 나타내었고 최대의 성능 차이는 약 8% 정도로 나타났다.

d 에 따른 ACTR의 성능 변동 폭을 알아보기 위해 λ 를 $4e-5$ 로 고정시키고 d 에 따른 성능 변화를 측정해보았고 그 결과 (그림 6)과 같이 나타났다. (그림 6)에서 알 수 있는 바와 같이 0~1사이의 d 값에 크게 관계없이 ACTR의 성능이 거의 일정하게 나타났다. Anderson은 [4]에서 ACT-R의 base-level activation의 경우 많은 응용 연구들에서 d 의 값이 0.5일 때 가장 인간 기억 인출과정의 많은 현상들에 적합한 결과를 보인다고 언급하였다. 종합해볼 때 d 의 값을 0~1사이의 값으로 선택하는 것이 가능하며 이 중 멱함수의 계산 비용이 적은 1을 선택하는 것이 일반적인 환경에서 추천할만하다고 할 수 있다.

ACTR은 k 값이 40 이상일 때에 대해서는 큰 변화 폭을 보이지 않았으며 k 값이 60일 때 접근 시간의 리스트를 기록하기 위한 추가적인 메모리 공간을 최소화시키면서도 성능이 최적에 근접한 실험 결과를 보였다.



(그림 6) d 에 따른 ACTR의 성능 변화

6. 결론 및 향후 연구

자주 접근되는 데이터에 대해서 빠른 접근을 보장하기 위해 사용되는 임시저장소인 캐쉬 구조는 컴퓨터 과학의 다양한 분야에서 오랫동안 연구되어온 주제이며, 특히 제한된 캐쉬의 저장 공간 내에서 효율적으로 데이터를 관리하기 위해 다양한 페이지 교체 알고리즘들이 연구되어 왔다. 대부분의 캐쉬 알고리즘들은 얼마나 최근에 데이터가 접근되었는가 또는 얼마나 자주 접근되었는가를 바탕으로 향후 데이터가 다시 접근될 것 같은 데이터들을 캐쉬 안에 저장하는 휴리스틱 방법을 취하고 있다. 이러한 컴퓨터 내에서의 데이터의 인출 과정은 인간의 기억 인출 과정과 유사하며, 인간의 기억 인출 과정 역시 캐쉬 구조처럼 기억이 얼마나 최근에 그리고 자주 인출되었는가에 의해 그 기억 인출의 확률과 인출속도가 결정된다는 것이 최근 연구에 의해서 밝혀진 바 있다.

본 연구에서는 인간의 기억 인출 과정에서의 과거 해당 기억에 대한 인출 기록들의 recency와 빈도가 인출 확률에 영향을 미치는 관계를 분석한 기존의 인지과학 분야의 연구 결과를 활용하였다. 해당 연구에서 recency 효과가 지수함수 형태가 아닌 멱함수의 형태로 감소한다는 것을 밝혀낸 것을 페이지 교체 정책 설계에 적용하여 ACTR을 제안하였다. 또한 실험을 통해 제안한 ACTR의 성능이 기존에 데이터베이스 버퍼 캐쉬에서 가장 좋은 성능을 보이는 것으로 알려진 LRFU보다 우수한 성능을 보인다는 것을 입증하였다. ACTR은 LRFU에 비해 파라미터 값에 관계없이 성능이 우수하며 최대 약 8% 정도의 성능 이점을 가지는 것으로 드러났다. 이러한 결과는 본 논문에서 제안하는 새로운 페이지 교체 정책인 ACTR이 기존의 단순한 LRU나 LFU와 같은 페이지 교체 정책보다 페이지 교체 시에 계산 시간이 더 소모되지만, CPU cache에 비해 상대적으로 연산 시간보다는 I/O 수행 시간이 중요한 DBMS cache에서 적용되어 DBMS의 성능을 개선할 수 있음을 시사한다.

향후 연구로서, 보다 많은 인지과학적 아키텍처의 특징들을 데이터베이스 기술에 이식하는 연구들을 수행하려고 한다. ACT-R[4]에서 사용하는 지식의 기본 단위인 chunk를 데이터베이스 기술에 적용하여 자주 같이 접근되는 페이지들을 ACT-R의 activation level 이론에 근거하여 chunk로 묶고 prefetching 단위나 페이지 재배치 정책에 chunk 개념을 활용하는 등의 연구가 이루어질 수 있다. 이러한 향후 연구들을 수행하여, 현재 컴퓨터과학의 인공지능 분야에 많이 활용되고 있는 인지과학적 지식들을, 현재는 그 적용이 거의 전무한 상태인 데이터베이스 분야의 기술들에도 활용하여, 데이터베이스 시스템 관련 기술의 발전에 새로운 계기를 제공하고자 한다.

참고 문헌

- [1] L. A. Belady, "A study of replacement algorithms for a virtual-storage computer," IBM Systems Journal, Vol. 5, No. 2, pp. 78-101, 1966
- [2] Y. Zhou et al., "The Multi-Queue Replacement Algorithm for Second Level Buffer Caches," Proceedings of the 2001 USENIX annual Technical Conference, pp 91-104, 2001
- [3] D. Lee et al., "On the Existence of a Spectrum of Policies that Subsumes the Least Recently Used (LRU) and Least Frequently Used (LFU) Policies," SIGMETRICS, pp 134-143, 1999.
- [4] J. R. Anderson et al., "An Integrated Theory of the Mind," Psychological Review, Vol. 111, No. 4, pp 1036-1060, 2004
- [5] J. R. Anderson et al., "Reflections of the environment in memory," Psychological Science, Vol. 2, pp 396-408, 1991
- [6] TPC BenchmarkTM C Standard Specification Revision 5.10.1.

Transaction Processing Performance Council, 2009.

- [7] <http://technet.microsoft.com/en-us/sysinternals/bb896646.aspx>
- [8] <http://hammerora.sourceforge.net/>
- [9] M. Sridharan, J. Wyatt, R. Dearden, "Planning to see: A hierarchical approach to planning visual actions on a robot using POMDPs," Vol. 174, No. 11, pp. 704-725, Artificial Intelligence, 2010
- [10] Y. Liu, "QN-ACES: Integrating Queueing Network and ACT-R, CAPS, EPIC, and Soar Architectures for Multitask Cognitive Modeling," International Journal of Human-Computer Interaction, Vol. 25, No. 6, pp 554-581, 2009



노 홍 찬

e-mail : fallsmal@cs.yonsei.ac.kr
 2006년 연세대학교 컴퓨터과학과 졸업(학사)
 2008년 연세대학교 컴퓨터과학과 졸업(석사)
 2010년~현재 연세대학교 컴퓨터과학과
 박사과정
 관심분야: 데이터베이스 시스템, 플래시
 메모리



박 상 현

e-mail : sanghyun@cs.yonsei.ac.kr
 1991년 서울대학교 컴퓨터공학과 졸업(석사)
 2001년 UCLA 컴퓨터과학과 졸업(박사)
 2002년 IBM T.J Watson 연구소 연구원
 2003년 포항공과대학교 컴퓨터과학과
 조교수

2010년~현재 연세대학교 컴퓨터과학과 부교수
 관심분야: 데이터베이스 시스템, 플래시 메모리