

# 플래시 메모리를 위한 Not-cold-page 쓰기지연을 통한 LRU 버퍼교체 정책 개선

## (Enhancing LRU Buffer Replacement Policy with Delayed Write of Not-cold-dirty-pages for Flash Memory)

정 호 영 <sup>†</sup> 박 성 민 <sup>†</sup> 차 재 혁 <sup>\*\*</sup> 강 수 용 <sup>\*\*\*</sup>  
 (Ho-young Jung) (Sung-min Park) (Jae-hyuk Cha) (Sooyong Kang)

**· 요약** 플래시 메모리는 비휘발성이며 빠른 I/O 처리 속도와 같은 많은 장점들이 있으나, in-place-update가 불가능하고 읽기/쓰기/지우기 작업의 속도가 다르다는 단점을 지니고 있다. 버퍼 캐시를 통해 플래시 메모리 기반 저장장치의 성능을 향상시키기 위해서는 수행 속도가 느린 지우기 작업의 수행 횟수에 직접적인 영향을 끼치는 쓰기 작업의 횟수를 줄이는 알고리즘이 필요하다. 본 논문에서는 기존의 LRU 버퍼교체 정책에 not-cold-dirty-page에 대한 교체를 지연하는 알고리즘을 적용한 새로운 버퍼교체 정책(LRU-Dirty Page Later-Cold Detection, 이하 LRU-DPL-CD)을 제시하고 성능을 분석한다. 트레이스 기반 시뮬레이션 실험에서 LRU-DPL-CD는 버퍼 적중률의 큰 감소 없이 쓰기 작업과 지우기 작업의 횟수를 감소시켰으며, 그 결과 전체 플래시 메모리의 I/O 수행속도가 증가하는 결과를 보였다.

**키워드** : 플래시 메모리, 버퍼 교체 정책

**Abstract** Flash memory has many advantages like non-volatility and fast I/O speed, but it has also disadvantages such as not-in-place-update data and asymmetric read/write/erase speed. For the performance of flash memory storage, it is essential for the buffer replacement algorithms to reduce the number of write operations that also affects the number of erase operations. A new buffer replacement algorithm is proposed in this paper, that delays the writes of not-cold-dirty pages in the buffer cache of flash storage. We show that this algorithm effectively decreases the number of write operations and erase operations without much degradation of hit ratio. As a result overall performance of flash I/O speed is improved.

**Key words** : Flash memory, buffer replacement algorithm

### 1. 서론

플래시 메모리는 비휘발성 기억장치로서 I/O 연산의 속도가 빠르고 외부 충격에 강하며 전력소모가 작고 무게가 가볍다는 장점을 가지고 있다. 이러한 특성으로 인해 플래시 메모리는 디지털 미디어 기기와 모바일 기기에서 데이터 저장장치로 널리 사용되고 있다.

그러나 플래시 메모리는 몇 가지의 하드웨어적인 단점을 가지고 있다. 플래시 메모리에서 I/O 작업의 단위인 페이지는 in-place-update가 불가능하다. 플래시 메모리의 쓰기가 일어난 더티(dirty) 페이지에 데이터를 다시 쓰기 위해서는 몇 개의 페이지로 구성된 블록 단위의 소거 연산(erase operation)이 선행되어야 하며 따라서 같은 페이지에 대한 재쓰기는 소거 연산의 원인이 된다. 또한 각각의 메모리 셀에 대한 지우기 횟수에는 제한이 있으며 이는 일반적으로 100,000번에서 1,000,000번 사이로 알려져 있다. 플래시 메모리의 또 다른 문제점은 읽기/쓰기/소거 연산의 수행속도간의 불균형이다. 쓰기 속도는 읽기 속도보다 느리며, 소거 연산의 속도는 읽거나 쓰기에 비해 매우 느리다[1,2].

최근의 플래시에 관한 많은 연구들은 기술한 플래시 메모리의 하드웨어적인 단점을 극복하려는데 주안점을

· 본 연구는 한국과학재단 특정기초연구(R01-2006-000-10630-0)지원으로 수행되었음

<sup>†</sup> 학생회원 : 한양대학교 정보통신학과  
 horong@hanyang.ac.kr  
 syrilo@hanyang.ac.kr

<sup>\*\*</sup> 종신회원 : 한양대학교 정보통신공학부 교수  
 chajh@hanyang.ac.kr

<sup>\*\*\*</sup> 종신회원 : 한양대학교 컴퓨터교육과 교수  
 sykang@hanyang.ac.kr

논문접수 : 2006년 5월 22일  
 심사완료 : 2006년 8월 17일

두고 있다. 제한적인 메모리 셀의 수명을 극복하기 위해서 소거 작업을 블록들에게 균등하게 분배하는 웨어 레벨링(wear leveling)알고리즘이 도입되었다. 플래시 메모리의 in-place-update가 불가능한 특성과 I/O 연산간의 속도 불균형을 극복하고 시스템의 성능을 향상시키기 위해서는 쓰기 연산과 소거 연산의 수행 횟수를 감소시키는 것이 필요하다. 이를 위하여 기존 연구에서는 다양한 가비지 콜렉션(garbage collection) 알고리즘, 블록 매핑 알고리즘(block mapping algorithm), 버퍼 교체 알고리즘이 제안되었다[1,3,4].

저장장치는 CPU에 비해 느리므로, 대부분의 운영체제는 버퍼 캐시를 사용하여 시스템의 성능을 향상시키고자 한다. 기존 운영체제의 버퍼 캐시는 읽기 연산과 쓰기 연산의 수행 속도가 비슷하고, in-place-update가 가능한 디스크를 가정하고 동작한다. 이러한 기존의 버퍼 캐시가 사용하는 교체 알고리즘을 플래시 메모리 저장장치에 적합하도록 개선하는 것은 플래시 기반 저장장치의 성능을 향상시키는 한 가지 방법이다. 플래시 메모리의 I/O 수행속도를 향상시키기 위해 우선적으로 고려되어야 할 사항은 소거 연산의 횟수를 줄이는 것이다. 하지만 버퍼 교체 정책을 개선하는 목적은 쓰기 연산의 수행 횟수를 감소시키는 데 있는데, 이는 버퍼 캐시 수준에서는 읽기 연산과 쓰기 연산만 고려하는 것이 가능하고, 버퍼 캐시 수준에서 쓰기 연산의 발생 횟수를 줄이는 것은 결과적으로 플래시 메모리에서의 물리적인 쓰기 연산과 소거 연산의 발생 횟수를 줄이는 결과를 가져오기 때문이다.

본 논문은 LRU 버퍼 교체 알고리즘을 플래시 메모리에 적합하도록 개선한 LRU-DPL-CD(LRU-Dirty Page Later-Cold Detection) 알고리즘을 제안한다. 제안된 알고리즘은 콜드 페이지가 아닌 더티 페이지(not-cold-dirty page)에 대한 교체 지연을 그 핵심으로 하는 버퍼 교체 정책으로써, 버퍼 캐시 적중률의 감소를 최소화하고 플래시 메모리의 물리적 쓰기를 효과적으로 감소시키는데 그 목적이 있다. 또한 LRU-DPL-CD 알고리즘은 구현 시 기존의 LRU 교체 정책에서 사용되는 자료구조에 콜드 플래그라 불리는 플래그만을 추가하면 되므로, 버퍼 교체 알고리즘에 의해 사용되어지는 자료구조의 유지와 교체시의 검색 오버헤드를 최소화하였다.

트레이스 기반의 시뮬레이션 결과 LRU-DPL-CD는 최대 50%의 물리적인 쓰기 연산 수행 횟수를 감소시켰으며 그 결과 감소된 쓰기 연산과 소거 연산으로 인해 25%의 전체 성능향상을 보였다.

논문의 구성은 다음과 같다. 2장에서는 관련 연구를 살펴보고 3장에서는 LRU-DPL-CD 알고리즘의 세부 사항에 대해 설명한다. 4장은 LRU-DPL-CD 알고리즘

의 성능 평가 결과를 기술한다. 마지막으로 5장에서는 본 논문의 결론과 앞으로의 연구 방향을 제시한다.

## 2. 관련 연구

### 2.1 플래시 메모리

플래시 메모리의 정식 명칭은 Flash EEPROM으로, 비휘발성 메모리의 한 종류이다. 상용화된 플래시 메모리는 NAND와 NOR 두 가지 타입으로 분류가 된다. 표1은 NAND 타입 플래시 메모리와 NOR 타입 플래시 메모리간의 서로 다른 특성을 보여준다.

표 1 플래시 메모리의 특성 [5]

Device	current (mA)		Access time (4kB)		
	Idle	Active	Read	Write	Erase
NOR	0.03	32	23 us	28 ms	1.2 sec
NAND	0.01	10	25 us	250 us	2 ms

NOR 플래시 메모리의 읽기 속도는 NAND 플래시 메모리보다 빠르지만, 쓰기 속도와 지우기 속도는 NAND 플래시 메모리보다 느리며 집적도 또한 낮다. 일반적으로 NOR 플래시 메모리는 코드의 저장용으로, NAND 플래시 메모리는 데이터 저장장치로 사용된다[6].

NAND 플래시 메모리는 일반적으로 512 바이트의 크기를 갖는 페이지 단위의 I/O를 지원하며 표 1에서 보는 바와 같이 쓰기 속도는 읽기 속도에 비해 약 8배 이상 느리다. 반면 소거 작업의 단위는 32개의 페이지로 구성된 블록이며, 소거 작업의 속도는 읽기/쓰기 작업에 비해 매우 느리다. 이러한 플래시 메모리의 물리적인 특성으로 인해 in-place-update가 불가능한 플래시 메모리를 기존의 하드 디스크와 같은 in-place-update가 가능한 블록 장치로 사용하기 위해서는 파일시스템의 LBA (Logical Block Address)와 플래시 메모리의 물리 페이지 주소 간의 블록 매핑 구조가 필요하다. 여러 블록 매핑 기법 중 널리 쓰이는 방식은 FTL(Flash Translation Layer)이다. FTL은 빠른 주소변환을 위해 SRAM에 블록이나 페이지에 대한 매핑 테이블을 저장하며, 그 외에 가비지 콜렉션, 배드 블록 관리 등의 역할을 수행한다. 그림 1은 FTL을 사용하는 플래시 저장 장치의 구조를 보인다.

그림 1에서 보이는 바와 같이, 파일 시스템 계층에서는 FTL을 사용함으로써 플래시 메모리를 블록 장치로 간주하며, 논리 페이지에 대한 재쓰기가 가능해진다. 그러나 플래시 메모리의 물리적 특성상 같은 LBA(Logical Block address)에 대한 재쓰기 작업은 플래시 메모리 상에서 다른 페이지 혹은 블록에서 이루어진다. 따라서 파일 시스템 계층에서 같은 LBA에 대한 재쓰기 작업의 수행 횟수를 감소시키는 것은 플래시 메모리에서

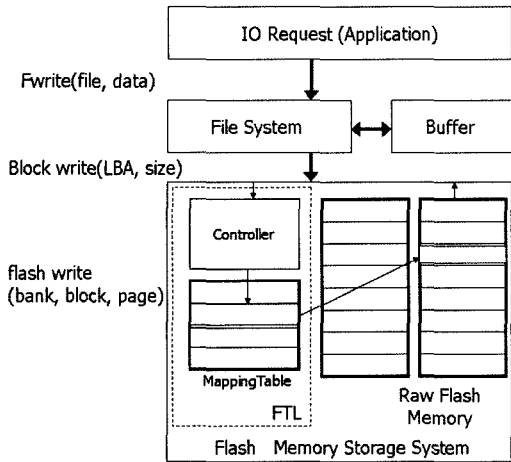


그림 1. FTL을 사용하는 NAND 플래시 저장장치 구조

의 물리적인 쓰기 및 소거 작업의 수행 횟수를 감소시키고 시스템의 성능을 향상시키며 또한 플래시 메모리의 수명을 증가시키는 효과를 가진다.

**2.2 플래시 메모리를 위한 버퍼 교체 알고리즘**

버퍼 캐시는 전체 디스크 블록의 일부를 주기억장치에 저장하는 공간으로서, 디스크 I/O의 발생횟수를 줄여 시스템의 성능을 향상시킨다. 버퍼 캐시의 공간은 디스크에 비해 한정적이므로 이를 적절히 활용하기 위해서 여러 버퍼 교체 알고리즘이 제안되었다. LRU, LFU, 클락(clock) 알고리즘과 같은 기존의 버퍼 교체 알고리즘의 목적은 페이지 적중률을 최대화하는 것이다. 이러한 기존의 알고리즘들은 디스크의 읽기와 쓰기 연산의 수행시간이 유사하다고 가정한다. 그러나 쓰기 연산의 수행속도가 읽기 연산에 비해 8배 이상 느린 플래시 메모리에서 더티(dirty) 페이지가 교체되는 것은 읽기만 발생한 페이지(clean page)를 교체하는 것보다 높은 비용을 발생시킨다. 따라서 기존의 페이지 적중률을 높이기 위한 버퍼 교체 알고리즘은 플래시 메모리 저장장치의 I/O 성능에 대한 최적화를 보장할 수 없다.

플래시 메모리를 위한 버퍼 교체 알고리즘의 한 기법으로는 기존의 버퍼 교체 알고리즘에 더티 페이지에 대한 지연 쓰기를 적용하는 기법이 있다. 이 알고리즘은 기존의 버퍼 교체 알고리즘에 의해 선택된 교체 페이지가 더티 페이지일 경우, 해당하는 페이지를 교체하지 않고 버퍼 캐시 내의 다른 페이지를 다시 찾는 방식이다. 이로 인해 기존의 버퍼 교체 알고리즘에 지연 쓰기를 적용하는 기법은 참조 빈도가 높지 않은 더티 페이지가 캐시의 저장 공간을 차지하게 됨으로써, 버퍼 캐시의 적중률이 감소할 가능성은 있으나 플래시 메모리의 쓰기와 소거 작업의 감소로 인해 전체적인 성능 향상을 유도한다.

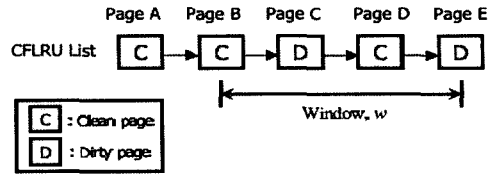


그림 2 CF-LRU 교체 예제 [2]

[2]에서는, CF-LRU(Clean First LRU) 교체 알고리즘을 제안하였다. CF-LRU는 LRU 알고리즘을 플래시 메모리 저장장치에 적합하게 개선한 알고리즘이다. 그림 2에서 페이지는 E, D, C, B, A 순으로 접근되었고, 이 중 페이지 C, E가 더티 페이지인 것을 가정한다. LRU 교체 알고리즘의 경우, 교체 페이지의 순서는 E, D, C, B가 되는데, 이는 항상 가장 오래전에 참조된 페이지를 교체하기 때문이다. 그러나 CF-LRU에서는 더티 페이지가 교체될 경우의 쓰기 및 소거 오버헤드를 감소시키기 위해, 더티 페이지인 E를 교체하는 것이 아니라, 쓰기가 발생하지 않은 D와 B가 우선적으로 교체되게 된다.

만약 최근에 참조된 클린 페이지가 교체될 경우, 페이지 폴트(page fault)가 발생할 수 있기 때문에, CF-LRU는 오프라인으로 결정된 윈도 크기 w 내에서만 더티 페이지에 대한 교체를 지연한다. 만약 윈도 내의 모든 페이지가 더티 페이지일 경우, CF-LRU의 교체 방식을 LRU 알고리즘처럼 가장 오래전에 참조된 페이지를 교체하게 된다[2]. 일반적인 LRU보다 버퍼 캐시의 적중률이 감소됨에도 불구하고, CF-LRU는 플래시 메모리의 쓰기와 소거 작업의 수행 횟수를 감소시키기 때문에 전체적인 성능이 향상 될 수 있다. 하지만, CF-LRU는 오프라인으로 윈도 사이즈 w를 결정해야 하며 따라서 다양한 워크로드에 적응적이지 못하다. 무엇보다도, CF-LRU 교체 알고리즘 하에서 버퍼 캐시는 자주 사용되는 더티 페이지(hot dirty page) 뿐만 아니라, 참조 빈도가 낮은 더티 페이지(cold dirty page)까지 버퍼 캐시에 장시간 유지할 가능성이 높고, 이로 인해 LRU보다 더 많은 페이지 폴트(page fault)를 발생시켜 I/O 성능저하를 가져올 수 있다. 이러한 문제점을 해결하기 위해서는 온라인으로 w를 결정하는 알고리즘과 윈도 내의 콜드 더티 페이지에 대한 쓰기 지연을 피하는 방법을 고려할 필요가 있다.

**2.3 플래시 메모리를 위한 핫-콜드 페이지 식별**

플래시 메모리 저장장치를 위한 핫 데이터 식별(hot data identification)은 가버지 콜렉션과 I/O 성능, 그리고 수명의 향상에 많은 영향을 끼친다[8]. 기존의 연구에서 핫 데이터 식별은 플래시 메모리 블록 내의 핫 데이터와 콜드 데이터를 분류하는데 적용되었다. 분리된 핫 데이터와 콜드 데이터는 각각 다른 블록에 저장되고,

따라서 핫 데이터가 저장된 핫 블록은 블록 내에 쓰기가 빈번하게 발생하게 된다. 플래시 메모리 저장장치에서 새로운 블록을 할당하기 위해 기존의 블록에 대한 가비지 콜렉션을 수행할 경우, 핫 블록을 소거하면 유효한 페이지의 수가 적어 소거의 효과가 커지므로, 가비지 콜렉션의 대상이 되는 블록의 수를 줄일 수 있다.

[9]에서는 플래시 메모리의 핫 쓰기 페이지(hot write page)를 식별하는 알고리즘을 제안하였다. 제안된 알고리즘은 핫 쓰기 페이지를 식별하기 위해 그림 3과 같이 LBA에 대한 2개의 고정된 크기의 LRU 리스트를 사용한다. 두 개의 리스트 중 처음의 리스트는 핫 리스트이며 두 번째 리스트는 후보 리스트이다. 페이지에 대한 쓰기가 발생하면 해당하는 페이지는 후보 리스트에 추가된다. 후보 리스트에 존재하는 페이지에 대한 쓰기가 발생했을 경우 해당하는 페이지는 핫 리스트에 추가되며, 각 리스트가 찼을 경우에는 LRU 알고리즘에 의해 가장 오래전에 참조된 페이지가 버려지게 된다. 2 단계 LRU 리스트 중 핫 리스트에 존재하는 페이지는 핫 쓰기 페이지로 간주되며, 그 외의 경우는 콜드 쓰기 페이지로 간주된다.

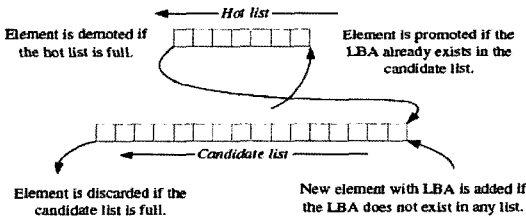


그림 3 2단계 LRU 리스트 [9]

핫-콜드 쓰기 페이지 식별을 쓰기 지연 버퍼 교체 알고리즘에 적용할 경우, 콜드 더티 페이지에 의해 발생하는 버퍼 캐시 저장 공간의 비효율적인 사용을 줄일 수 있다. 하지만, [9]의 식별 알고리즘은 워크로드에 따라 2 단계 LRU 리스트의 크기를 결정해 주어야 하며, 자료 구조의 오버헤드로 인해 이를 버퍼 교체 알고리즘에 적용하기에는 적합하지 않다. 버퍼 교체 알고리즘에 핫-콜드 식별을 적용하기 위해서는 자료구조 및 검색의 오버헤드가 적고, 다양한 워크로드에 적응적인 알고리즘이 필요하다.

### 3. 제안하는 쓰기 지연 알고리즘

본 논문에서는 플래시 메모리 저장장치의 성능 향상을 위해 기존의 LRU 교체 정책을 개선한 LRU-DPL-CD(LRU-Dirty Page Later-Cold Detection) 버퍼 교체 알고리즘을 제안한다. 본 알고리즘은 페이지 교체가 발생했을 때, 더티 페이지를 플래시 메모리로 플러시

(flush)하는 횟수를 감소시켜 물리적인 쓰기와 소거 작업의 수행 횟수를 감소시킨다. 페이지 교체가 발생했을 때, LRU 교체 알고리즘은 가장 오래 전에 참조된 페이지가 항상 교체 페이지로 선정되어, 페이지 교체의 대상이 되지만, LRU-DPL-CD 알고리즘 하에서는 교체 페이지로 선택된 페이지가 참조 빈도가 높은 더티 페이지일 경우에는 이를 교체하지 않고 다른 교체 대상 페이지를 버퍼 캐시에서 검색한다.

LRU-DPL-CD 알고리즘은 LRU 알고리즘에 비해 적용률이 낮아질 가능성이 있고, 자료 구조 및 교체 대상 페이지 검색의 오버헤드가 LRU 알고리즘에 비해 높다는 단점이 있으나, 플래시 메모리의 쓰기 작업과 소거 작업의 효과적인 감소로 인해 전체적인 I/O 성능은 향상되는 결과를 보인다.

#### 3.1 LRU-DPL 알고리즘

[10]에서 제안된 LRU-DPL(LRU-Dirty Page Later) 알고리즘은 기존의 LRU 알고리즘을 개선하여 플래시 메모리의 I/O 성능을 향상키는 알고리즘이다. 버퍼 캐시의 교체가 필요한 상황에서 시스템은 버퍼 리스트를 검색하여 가장 오래 전에 참조된 위치(LRU position)의 페이지가 더티 페이지일 경우, 해당 페이지는 버퍼 리스트의 가장 최근에 참조된 위치(MRU position)로 이동하며, 다시 교체 대상 페이지를 버퍼리스트에서 검색하게 된다. 버퍼가 불필요한 더티 페이지로 완전히 채워지는 것을 방지하기 위해, 이 과정은 최대  $k$  번 반복하게 되며  $k$  번 반복한 후에는 LRU 위치에 존재하는 페이지가 더티 페이지라고 하더라도, 교체 대상 페이지가 된다.

LRU-DPL 알고리즘은, 플래시 메모리의 쓰기 및 소거 작업의 발생 횟수를 줄이는데 있어 효과적이거나, 더티 페이지 최대 지연 횟수인  $k$ 를 오프라인으로 결정해야 하므로 워크로드에 적응적이지 못하다. 또한, LRU-DPL 알고리즘은 참조 빈도가 낮은 콜드 더티 페이지를 버퍼 캐시에 유지함으로써, 참조 빈도가 높은 클린 페이지를 교체 페이지로 선택하게 되고 결국 페이지 플러시 횟수를 증가시키게 되어 플래시 메모리의 읽기 작업 횟수를 증가시키는 단점을 가지고 있다.

이러한 단점을 극복하기 위해 본 논문에서는 LRU-DPL-CD 알고리즘을 제안한다.

#### 3.2 LRU-DPL-CD 알고리즘

LRU-DPL이나 CF-LRU 알고리즘은 더티 페이지를 참조 빈도에 관계없이 일정시간 동안 버퍼 캐시에 유지하는 버퍼 교체 알고리즘이다. 그러나 앞 장에서 기술한 바와 같이, 모든 더티 페이지에 대해 플래시 메모리의 플러시(flush)를 지연하고, 이를 일정 기간 동안 버퍼 캐시에 저장하는 것은, I/O 성능을 향상시키는데 있어 효과적이지 못하다.

LRU-DPL-CD 알고리즘은 LRU-DPL 알고리즘의 이러한 문제점을 해결하기 위해서, LRU-DPL을 개선하여 콜드 쓰기 페이지 감지(cold write page detection)를 적용한 알고리즘이다. 콜드 쓰기 페이지 감지에 의해 LRU-DPL-CD 알고리즘은 미래에 참조가 발생할 확률이 낮은 콜드 더티 페이지에 대한 지연 쓰기를 피한다.

그림 4와 같이 LRU-DPL-CD는 기존의 LRU 교체 알고리즘을 구현하기 위한 LRU 리스트에 콜드 플래그(cold flag)라 불리는 플래그를 사용한다. 콜드 플래그는 더티 페이지에 대한 메타 데이터로서, 해당 플래그가 설정된 더티 페이지를 콜드 더티 페이지로 간주한다. 버퍼 캐시 내에서 페이지 교체가 필요할 때 LRU-DPL-CD는 먼저 버퍼 리스트의 LRU 위치에 있는 페이지가 더티 페이지인지를 검사한다. 해당 페이지가 더티 페이지일 경우에는 다시 콜드 플래그를 검사하여 콜드 플래그가 설정되어 있지 않을 경우 콜드 플래그를 설정하고 버퍼 리스트의 MRU 위치로 이동시키고 새로이 LRU 위치에 있는 페이지에 대해 위의 작업을 반복 수행한다. LRU 위치에 있는 페이지의 콜드 플래그가 설정되어 있거나, 더티 페이지가 아닌 경우, 해당 페이지는 교체 대상 페이지가 된다. 또한 버퍼 내의 페이지가 다시 참조가 된 경우에는 해당 페이지의 콜드 플래그가 다시 초기화된다.

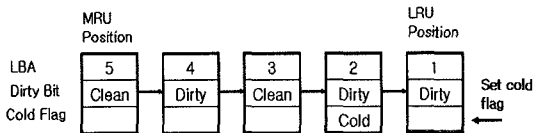


그림 4 LRU-DPL-CD 구조

제한된 콜드 쓰기 페이지 감지 알고리즘은 유사한 방식으로 핫 쓰기 페이지 감지 알고리즘으로도 구현될 수 있다. 그러나 플래시 메모리 저장장치에 대한 실험을 분석한 결과 핫 쓰기 페이지 감지 알고리즘을 적용하여 해당하는 페이지만 버퍼 캐시에서 지연 쓰기를 적용하는 기법보다는 콜드 쓰기 페이지 감지 알고리즘을 통해 해당하는 페이지에 대한 지연 쓰기를 피하는 알고리즘이 더 좋은 성능을 보였다. 이는 핫 쓰기 페이지에 대해서만 지연쓰기를 적용할 경우, 버퍼 적중률의 감소는 매우 작지만, 쓰기 횟수의 감소가 충분하지 않기 때문이다.

LRU-DPL-CD 알고리즘은 기존의 LRU에서 사용하는 리스트 구조 외에 콜드 플래그만이 추가적으로 필요하므로, 자료구조와 검색 시의 오버헤드가 매우 작다. 그림 5는 LRU-DPL-CD의 교체 페이지 선택 알고리즘을 나타낸다.

```

Victim Selection(L)
{
    victim = the page at LRU position in L
    while (victim is dirty) {
        if (victim is already visited one)
            exit while
        else
            move victim to MRU position in L
            set cold-flag of victim
            victim = the page at LRU position in L
    }
    remove victim from L
    return victim
}
    
```

\* L : buffer list of LRU

그림 5 LRU-DPL-CD 교체 알고리즘

#### 4. 성능 평가

여기서는 LRU-DPL-CD, LRU, CF-LRU, LRU-DPL의 히트 적중률과 I/O 수행시간을 비교한다. 성능평가 방법으로 트레이스 기반 시뮬레이션을 사용했으며 시뮬레이션 데이터는 요청된 디스크 블록의 번호, 읽기/쓰기 여부로 구성된다.

##### 4.1 실험환경

실험에 사용된 트레이스들은 SMDK 2410 임베디드 평가 보드[12]와 리눅스 운영체제를 기반으로 하여 PostgreSQL RDBMS[11] 응용을 수행시키면서 수집되었다. 해당 운영체제와 RDBMS 응용을 위한 저장장치로는 스마트 미디어 타입 삼성 64MB NAND 플래시 메모리 K9S-1208V0M[13]가 사용되었다. 표 2와 그림 6은 주어진 응용의 트레이스 데이터의 접근 패턴과 특성을 나타낸다. 표 2에서 p%/g% 지역성은 전체 참조 횟수의 g%가 전체 페이지 중 p%에 집중되어 있다는 것을 의미한다. 그러므로 실험에 사용한 응용은 읽기 작업보다 쓰기 작업에 대한 지역성이 높다는 것을 알 수 있다.

표 2 응용 트레이스의 특성

파일 시스템	YAFFS
응용	PostgreSQL RDBMS
물리 페이지 크기	512 바이트
논리 페이지 크기	4 KB
I/O 요청 횟수	51893
쓰기 요청 횟수	5751 (11.08%)
읽기 지역성	30% / 70%
쓰기 지역성	15% / 85%

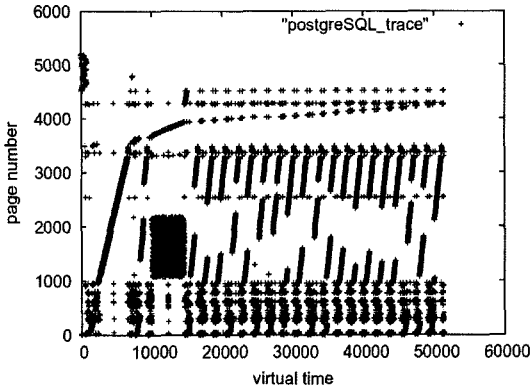


그림 6 PostgreSQL 응용 트레이스의 참조 패턴

더티 페이지를 지연하는 버퍼 교체 알고리즘의 경우, 쓰기 지역성의 정도가 성능에 미치는 영향은 매우 크다. 쓰기 지역성이 낮을 경우, 더티 페이지를 버퍼에 장기간 유지하게 되는 CF-LRU와 LRU-DPL의 경우는 감소되는 쓰기로 인한 이득보다, 버퍼 공간 낭비로 인한 읽기 횟수의 증가로 인한 비용이 더 커질 가능성이 존재한다.

4.2 버퍼 적중률

그림 7은 버퍼 적중률에 대한 실험 결과이다. 예상한 바와 같이 콜드 더티 페이지를 버퍼에 유지하는 LRU-DPL 알고리즘과 CF-LRU 알고리즘의 적중률이 낮게 나타났으며, 특히 LRU-DPL 알고리즘은 더티 페이지를 LRU 위치에서 MRU 위치로 옮기게 되므로, 시간이 지남에 따라 버퍼에 콜드 더티 페이지의 비율이 증가하여 적중률이 여러 알고리즘 중 가장 나쁜 것으로 나타났다.

CF-LRU는 기술한 바와 같이  $w(0 < w < 1)$ 의 크기를 조절함에 따라 적중률이 변하는 것으로 나타났다. 전체 버퍼의 크기를 B라고 할 경우 원도의 크기는  $w \cdot B$ 가 되며,  $w$ 가 0에 가까울수록 일반 LRU와 유사한 결과를 보이고 1에 가까울수록 LRU-DPL과 유사한 결과를 나

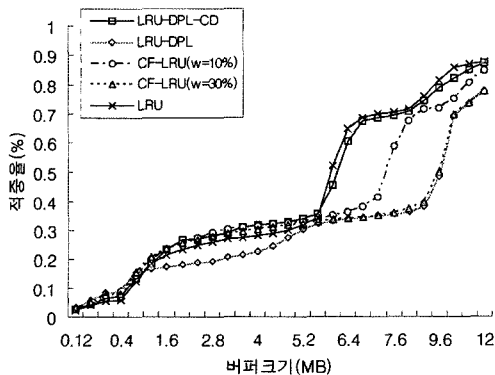


그림 7 버퍼 적중률

타냈다. 실험에는  $w=0.1B$ 인 경우와  $w=0.3B$ 인 경우의 결과를 그래프로 나타내었다.

LRU-DPL-CD는 예상한 바와 같이 다른 교체 알고리즘보다 높은 적중률을 보이며, 특히 LRU와 비교해도 크게 떨어지지 않는 적중률을 보임을 알 수 있다. 버퍼의 크기가 작은 구간에서는 LRU나 LRU-DPL-CD가 오히려 다른 알고리즘보다 나쁜 적중률을 보이는데, 이는 LRU 알고리즘의 경우 트레이스가 순환 패턴이나 일정한 확률로 참조될 경우, 이를 잘 감지 못하기 때문이다. 반면 다른 알고리즘은 더티 페이지를 장기간 저장하는 특성으로 인해, 이들이 적중되면서, 적중율이 약간 높게 나타난 것으로 분석된다.

4.3 쓰기 감소 횟수

그림 8은 버퍼 교체 시 발생하는 플래시 메모리로의 쓰기 횟수를 비교한 실험의 결과이다. 쓰기 횟수는 더티 페이지를 가장 장시간 버퍼에 저장하게 되는 LRU-DPL이 가장 효과적으로 쓰기 횟수를 감소시키는 것으로 나타났으며, CF-LRU는  $w$ 를 버퍼의 30%로 한 경우가  $w$ 를 10%로 한 경우보다 효과적으로 쓰기 횟수를 감소시켰다. 한편 LRU-DPL-CD는 버퍼의 크기가 충분히 커짐에 따라 쓰기 횟수의 감소효과가 점점 커지는 것으로 나타났는데, 이는 적중률이 다른 알고리즘보다 높기 때문인 것으로 분석된다. 한편 해당하는 워크로드에서 쓰기는 1233번 이하로는 더 이상 감소하지 않았는데, 이는 각각 독립적으로 반드시 써야하는 페이지들의 수 때문이다.

4.4 수행시간

그림 9는 각 버퍼 교체 알고리즘의 수행시간을 비교한 실험의 결과이다. 수행시간은 표 2에서 나타난 플래시 메모리의 물리적 특성을 참조하여 플래시 메모리에 발생한 물리적인 읽기, 쓰기, 지우기 작업의 합으로 정의하였다.

버퍼의 크기가 4MB이하의 경우 LRU를 제외한 다른

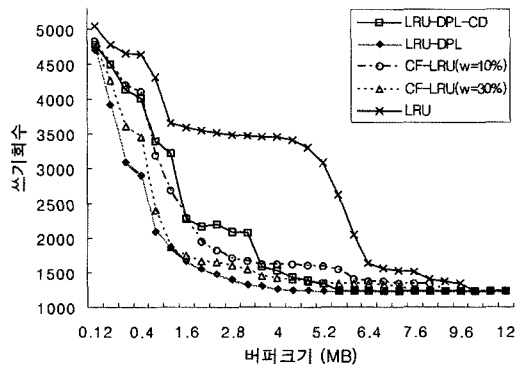


그림 8 쓰기 발생 횟수

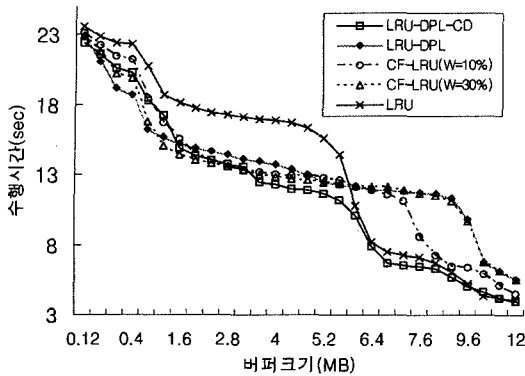


그림 9 전체 수행시간

알고리즘들은 유사한 성능을 보이나, 대체로 LRU-DPL과 CF-LRU(w=30%)인 경우가 조금 더 높은 성능을 나타내는 것으로 보인다. 이는 그림 6과 그림 7에서 보는 바와 같이 LRU-DPL과 CF-LRU(w=30%)인 경우가 쓰기 횟수를 다른 알고리즘보다 효과적으로 감소시켰으며, 부가적으로 버퍼의 크기가 워킹셋의 크기보다 작을 경우 순환 패턴을 잘 처리하지 못하여 LRU 알고리즘의 적중률이 낮기 때문이다.

버퍼의 크기가 4MB 이상이 된 경우, LRU-DPL-CD 알고리즘이 가장 성능이 좋은 것으로 나타났다. 특히 버퍼의 크기가 4MB 이상 6.4 MB 이하의 경우는 다른 알고리즘에 비해 높은 성능을 보이며, 기존의 LRU 알고리즘에 비해서는 약 30%이상 성능을 향상을 보이는 것으로 나타났다. 버퍼의 크기가 6.4 MB 이상인 경우에는 LRU와 유사한 성능을 보이는데, 이는 쓰기 횟수의 감소가 더 이상 일어나지 않기 때문이다. 그러나 타 알고리즘들은 버퍼 크기가 6.4 MB 이상인 경우 LRU에 비해 성능이 감소하는 것으로 나타났다.

## 5. 결론

본 논문에서는 플래시 메모리 기반 시스템을 위하여 LRU 버퍼 교체 알고리즘을 개선한 LRU-DPL-CD 알고리즘을 제안하였다. 제안된 LRU-DPL-CD 알고리즘은 버퍼 캐시에 저장된 콜드 페이지가 아닌 더티 페이지에 대한 쓰기를 지연한다. 제안된 알고리즘에서는 LRU 리스트에 콜드 플래그를 추가하여 콜드 페이지를 인식하는 기법을 제안하고 적용하였다.

트레이스 기반 시뮬레이션에서는 제안된 알고리즘이 효과적으로 쓰기와 지우기 작업의 횟수를 감소시켜 전체적인 성능이 향상되는 것을 보였다.

향후 연구로 LRU 알고리즘뿐만 아니라, 현재까지 연구된 LIRS나 ARC와 같은 여러 LRU 기반 버퍼 교체 알고리즘에 DPL-CD 정책을 적용하고 평가할 계획이

며, 또한 다양한 플래시 파일 시스템과 FTL 알고리즘을 통한 워크로드를 통해 테스트를 수행할 계획이다.

## 참고 문헌

- [1] H. Kim and S. Lee., "A New Flash Memory Management for Flash Storage System," In 32rd Annual Intl. Computer Science and Applications Conference, October 1999.
- [2] Chanik Park, Jeong-Uk Kang, Seon-Yeong Park, Jin-Soo Kim, "Energy-aware demand paging on NAND flash-based embedded storages," Proceedings of the 2004 international symposium on Low power electronics and design table of contents, pp.338-343, 2004.
- [3] A. Kawaguchi, S. Nishioka, and H. Motoda, "A Flash Memory based File System," Proceedings of the USENIX Technical Conference, 1995.
- [4] M. L. Chiang, C. H. Paul, R. C. Chang, "Manage flash memory in personal communicate devices," Proceedings of IEEE International Symposium on Consumer Electronics, 1997.
- [5] Samsung Electronics: NAND flash memory & SmartMedia data book, 2004.
- [6] Arie Tal, "Two Technologies Compared: Nor vs. NAND White Paper," [http://www.m-sys.com/NR/rdonlyres/24795A9E-16F9-404A-857C-C1DE21986D28/77/NOR\\_vs\\_NAND6.pdf](http://www.m-sys.com/NR/rdonlyres/24795A9E-16F9-404A-857C-C1DE21986D28/77/NOR_vs_NAND6.pdf)
- [7] Eran Gal, Sivan Toledo, "Mapping Structures for Flash Memories: Techniques and Open Problems," Proceedings of the IEEE International Conference on Software-Science, Technology and Engineering, 2005.
- [8] Jen-Wei Hsieh, Li-Pin Chang, Tei-Wei Kuo, "Embedded systems: applications, solutions and techniques (EMBS): Efficient on-line identification of hot data for flash-memory management," Proceedings of the 2005 ACM symposium on Applied computing, 2005.
- [9] Li-Pin Chang, Tei-Wei Kuo, "An Adaptive Striping Architecture for Flash Memory Storage Systems of Embedded Systems," Proceeding of the 8th IEEE Real-Time and Embedded Technology and Applications Symposium, 2002.
- [10] Hoyoung Jung, "A Buffer Replacement Algorithm for Embedded Database System on NAND Flash Storage," Master Thesis, Hanyang University.
- [11] <http://www.postgresql.org>
- [12] Aiji Systems, "Overview of SMDK 2410 board," <http://www.aijisystem.com/korea/product/evboard/SMDK2410.htm>.
- [13] Samsung Elec., "NAND-type Flash Memory," <http://www.samsung.com/Products/Semiconductor/Flash/index.htm>.



정 호 영

2004년 한양대학교 재료공학부(공학사)  
2006년 한양대학교 정보통신대학원(공학석사). 2006년~현재 한양대학교 대학원 정보통신학과 박사과정. 관심분야는 데이터베이스, 플래시 메모리 기반 저장장치, 임베디드 시스템



박 성 민

2005년 한양대학교 컴퓨터교육과(이학사). 2005년~현재 한양대학교 대학원 정보통신학과 석사. 관심분야는 파일시스템, 플래시 메모리 기반 저장장치



차 재 혁

1987년 서울대학교 계산통계학과(학사)  
1991년 서울대학교 컴퓨터공학과(석사)  
1997년 서울대학교 컴퓨터공학과(박사)  
1997년~1998년 한국학술진흥재단 부설 첨단기술정보센터선임연구원. 1998년~2001년 한양대학교 사범대학 컴퓨터교육과 조교수. 2001년~현재 한양대학교 정보통신대학 정보통신학부 교수. 관심분야는 XML, 데이터베이스, 플래시 메모리 기반 저장시스템, 멀티미디어 콘텐츠 적용화, e-Learning



강 수 용

1996년 서울대학교 수학과 이학사. 1998년 서울대학교 전산학과 이학석사. 2002년 서울대학교 전기컴퓨터공학부 공학박사. 2000년 2월~2002년 2월 (주)시그엔 네트워크 보안장비 개발 팀장(선임연구원) 2002년 8월~2003년 2월 서울대학교 전기컴퓨터공학부 박사후연구원. 2003년 3월~현재 한양대학교 컴퓨터교육과 조교수, 정보통신대학 정보통신학부 교수. 관심분야는 멀티미디어 시스템, 분산시스템, 플래시 메모리 기반 저장시스템, e-Learning