MLC-LFU : 플래시 메모리를 위한 멀티레벨 버퍼 캐시 관리 정책

(MLC-LFU : The Multi-Level Buffer Cache Management Policy for Flash Memory)

옥 동 석 † 이 태 훈 † 정 기 동 ††
(Dongseok Ok) (Taehoon Lee) (Kidong Chung)

요 약 플래시 메모리는 현재 휴대용 기기 뿐 아니라 개인용 컴퓨터와 서버용 컴퓨터에서 널리 사용되고 있다. 하드디스크를 위한 버퍼 캐시 교체 정책인 LRU(Least Recently Used)와 LFU(Least Frequently Used)는 플래시 메모리의 특성을 전혀 고려하지 않아 플래시 메모리에 적합하지 않다. 기존에 연구되었던 CFLRU(Clean-First LRU)와 그 변형인 CFLRU/C, CFLRU/E, DL-CFLRU/E는 플래시 메모리의 특성을 고려하였지만 hit ratio가 LRU와 LFU에 비하여 좋지 않다. 본 논문에서는 기존의 버퍼 캐시 교체 정책들을 보완하는 새로운 버퍼 캐시 교체 정책을 제안한다. 이 버퍼 캐시 교체 정책은 LFU를 기반으로 하고 플래시 메모리의 특성을 고려하였다. 그리고 이 새로운 버퍼 캐시 교체 정책을 기존 플래시 메모리 버퍼 캐시 교체 정책과 hit ratio와 flush 횟수를 비교하여 성능을 평가한다.

키워드 : 플래시 메모리, 버퍼 캐시 교체 정책

Abstract Recently, NAND flash memory is used not only for portable devices, but also for personal computers and server computers. Buffer cache replacement policies for the hard disks such as LRU and LFU are not good for NAND flash memories because they do not consider about the characteristics of NAND flash memory. CFLRU and its variants, CFLRU/C, CFLRU/E and DL-CFLRU/E(CFLRUs) are the buffer cache replacement policies considered about the characteristics of NAND flash memories, but their performances are not better than those of LRU. In this paper, we propose a new buffer cache replacement policy for NAND flash memory. Which is based on LFU and is taking into account the characteristics of NAND flash memory. And we estimate the performance of hit ratio and flush operation numbers. The proposed policy shows better hit ratio and the number of flush operation than any other policies.

Key words : Flash memory, Buffer cache replacement policy

1. 서 론

플래시 메모리는 최근 양적·질적으로 많은 발전을 해왔다. 플래시 메모리는 임베디드 기기와 휴대용 기기뿐만 아니라 개인용 컴퓨터와 서버용 컴퓨터에도 사용되고 있다[1]. 플래시 메모리는 무게가 가볍고, 읽기 속도가 빠르며, 충격에도 강하다는 장점 때문에 하드디스크를 대체할 것이라고 전망하고 있다. 현재 플래시 메모리가 들어가지 않는 전자 기기가 없을 정도로 수요가 많다[2].

플래시 메모리는 위와 같은 여러 가지 장점을 가지고 있지만, 심각하게 고려해야 할 단점들도 있다. 첫째, 업데이트 시 제자리 덮어쓰기가 불가능하다. 하드디스크의 경우 데이타가 변경되면 해당 부분을 다시 쓸 수 있지

· 이 논문은 2단계 두뇌한국21 사업에 의해 지원되었음

† 학생회원 : 부산대학교 컴퓨터공학과
ok3@melon.cs.pusan.ac.kr
withsoul@pusan.ac.kr
†† 종신회원 : 부산대학교 컴퓨터공학과 교수
kdchung@pusan.ac.kr

논문접수 : 2008년 8월 20일
심사완료 : 2008년 10월 21일

만 플래시 메모리는 지우기 전에 해당 페이지를 다시 쓰는 것이 불가능하다. 이를 해결하기 위하여 일반적인 플래시 메모리에서 업데이트는 플래시 메모리의 빈 영역에 업데이트할 데이타를 기록하고 이전에 기록되었던 데이타의 페이지들은 invalid 표시를 하게 된다. 둘째, invalid 표시된 영역을 다시 쓸 수 있도록 하기 위해서 가비지 컬렉션이라는 추가적인 연산이 필요하다. 플래시 메모리는 블록단위로 지움 연산을 하므로 invalid 페이지를 재사용하기 위해서는 valid 페이지들을 다른 블록으로 복사한 후 해당 블록을 지워야한다. 이때 valid 페이지를 복사하는 비용이 추가되므로 이 추가 비용을 줄이기 위해서 좋은 가비지 컬렉션 정책이 필요하다. 셋째, 플래시 메모리의 지움 횟수는 10,000~1,000,000번으로 제한된다[3]. 플래시 메모리의 지움 횟수가 이를 넘게 되면 플래시 메모리에 기록된 데이타가 제대로 유지되지 않아 데이타를 신뢰할 수 없다. 플래시 메모리의 각 블록의 지움 횟수가 균일하지 않으면 플래시 메모리의 수명을 단축시킬 수 있으므로 블록별 지움 횟수를 균일하게 유지하는 wear-leveling 정책도 필요하다.

하드디스크를 위한 버퍼 캐시 교체 정책은 단순히 hit ratio만을 고려하고 있다. 하지만 플래시 메모리는 지움 횟수가 제한되므로 지움 연산을 유발할 수 있는 flush 연산 횟수도 최소화하여야 한다. 하드디스크를 위한 버퍼 캐시 교체 정책들은 일반적으로 이 flush 연산의 횟수를 고려하고 있지 않으므로 플래시 메모리에 적합하지 않다.

현재까지 제안된 플래시 메모리를 위한 버퍼 캐시 교체 정책들은 이를 위하여 clean 페이지를 우선적으로 교체하지만 이를 위해서 hit ratio를 희생하였다.

본 논문에서는 플래시 메모리의 특성을 반영하되 hit ratio도 개선할 수 있는 버퍼 캐시 교체 정책을 제안한다. 이 버퍼 캐시 교체 정책은 하드 디스크를 위한 버퍼 캐시 교체 정책들 보다 나은 hit ratio를 보이고 플래시 메모리를 위한 버퍼 캐시 교체 정책들 보다 낮은 flush 연산 횟수를 보여 플래시 메모리에 더욱 적합한 버퍼 캐시 교체 정책이라 할 수 있다.

본 논문의 이 후는 다음과 같이 구성된다. 2장은 하드디스크와 플래시 메모리를 위한 버퍼 캐시 교체 정책들에 대하여 설명한다. 3장에서는 버퍼 캐시 교체 정책들을 비교하기 위한 워크로드에 대하여 설명하고 그 워크로드로 성능 실험을 하였을 때 각 정책들의 성능을 분석한다. 4장은 본 논문에서 제안하는 MLC-LFU의 구조와 각 연산들에 대하여 설명하고 5장에서 다른 버퍼 캐시 교체 정책들과의 성능 비교를 통하여 본 논문에서 제안하는 기법의 우수성을 검증한다. 6장에서는 본 논문에 대하여 결론을 내린다.

2. 관련연구

2.1절에서는 하드디스크를 위한 버퍼 캐시 교체 정책인 LRU와 LFU에 대하여 설명한다. 2.2절에서는 플래시 메모리를 위한 버퍼 캐시 교체 정책인 CFLRU와 그 변종들에 대하여 설명한다.

2.1 하드디스크를 위한 버퍼 캐시 교체 정책

하드디스크를 위한 대표적인 버퍼 캐시 교체 정책으로는 LRU(Least Recently Used)와 LFU(Least Frequently Used) 정책이 있다[4]. LRU정책은 가장 최근에 참조된 페이지는 앞으로 참조될 가능성이 크고 반대로 가장 오랫동안 참조되지 않은 페이지는 앞으로도 참조되지 않을 것이라고 가정한다. 따라서 LRU 정책은 페이지 교체 시 교체 대상이 되는 페이지인 victim을 선택할 때 가장 오래 전에 참조된 페이지를 선택한다. LFU는 LRU와 다르게 많이 참조된 페이지는 앞으로 참조될 가능성이 크고 적게 참조된 페이지는 앞으로 참조되지 않을 것이라고 가정한다. 이와 같은 가정으로 LFU는 각 페이지마다 참조 횟수(R.C, Reference Counter)를 두어 가장 적게 참조된 페이지를 victim으로 선택한다.

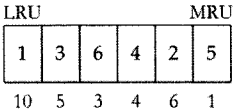

그림 1 LRU의 단점의 한 예

LRU의 경우 해당 페이지가 몇 번 참조되었는지를 전혀 고려하지 않는다. 그림 1을 참고하여 보면 페이지 리스트의 LRU에 가장 많이 참조된 1번 페이지가 있고 MRU에는 가장 적게 참조된 5번 페이지가 있다. 페이지 부재가 발생하여 교체가 필요하게 되면 가장 많이 참조된 1번 페이지가 교체된다. 이는 연산이 빈번하게 발생하는 핫 블록(hot block)을 희생하는 결과를 초래하므로 버퍼 캐시의 성능을 저하시킬 수 있다.

LFU는 LRU와는 반대로 참조 횟수를 고려한다. 따라서 페이지 참조에 참조 지역성이 있는 경우 좋은 성능을 보일 수 있다. 하지만 LFU의 경우 참조 횟수만을 고려하는 것이 약점이 될 수 있다. 그림 2와 같은 상황을 고려해보면, 마지막 페이지의 참조 횟수가 가장 낮으므로 victim으로 선택된다. 새로 페이지가 교체되면 역시 참조 횟수가 가장 낮으므로 이 페이지가 다시 참조되어 참조 횟수가 증가하지 않는 한 다음번에 교체가 필요할 경우 다시 교체 대상이 될 가능성이 높다. 최악의 상황에는 다른 페이지는 그대로 존재하고 가장 참조 빈도가 낮은 하나의 페이지만 계속해서 교체되는 상황

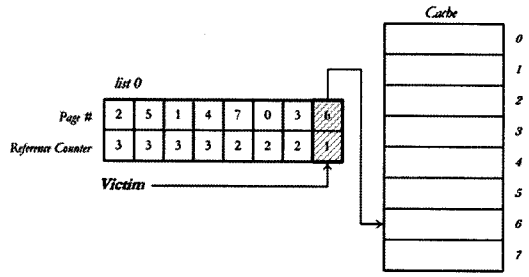

그림 2 LFU에서 발생할 수 있는 최악의 상황

이 발생할 수 있다. 이 경우 페이지가 8개이지만 1개인 것과 다름없는 상황이 될 수 있다.

2.2 플래시 메모리를 위한 버퍼 캐시 교체 정책

플래시 메모리를 위한 버퍼 캐시 교체 정책 중 CFLRU(Clean First LRU)가 있다[5]. CFLRU는 LRU 정책을 기반으로 하고 있고 flush 횟수를 줄이기 위해 버퍼 캐시 페이지 중 clean 페이지를 우선적으로 교체한다. CFLRU와 유사한 버퍼 캐시 교체 정책으로 CFLRU/C, CFLRU/E, DL-CFLRU/E 등이 있다[6]. CFLRU/C는 각 페이지에 참조 카운터를 두어 LRU 리스트 내에 Dirty 페이지만 있을 경우, 참조 횟수가 가장 작은 Dirty 페이지를 교체한다. CFLRU/E는 CFLRU/C와 유사하나 참조 횟수가 아닌 플래시 메모리의 대상 블록의 지움 횟수를 고려하는 것이다. CFLRU와 CFLRU/C, CFLRU/E의 구조는 그림 3, 그림 4, 그림 5와 같다. DL-CFLRU/E는 CFLRU/E와 유사한 교체 정책으로 Dirty 페이지와 clean 페이지를 각기 다른 리스트로 관리한다.

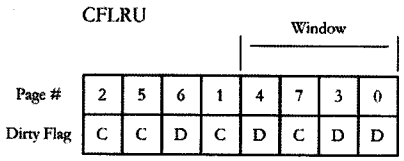

그림 3 CFLRU의 구조

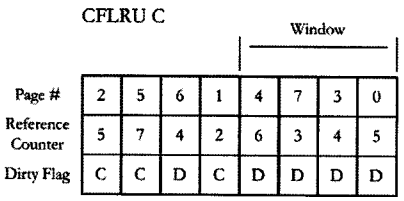

그림 4 CFLRU/C의 구조

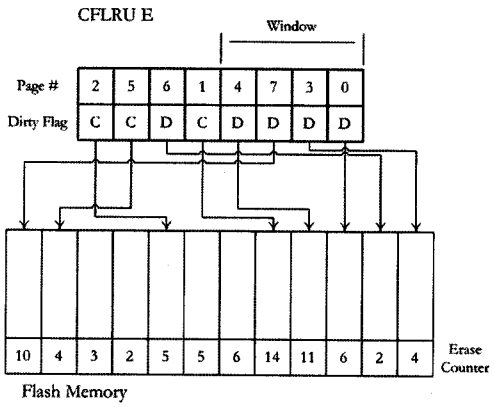

그림 5 CFLRU/E의 구조

3. 기존 버퍼 캐시 교체 기법의 성능 분석

3.1절에서는 본 논문에서 버퍼 캐시 교체 정책들의 성능 평가 시 사용하게 될 워크로드에 대해서 설명한다. 3.2절에서는 이 워크로드를 사용하여 각 버퍼 캐시 교체 정책들의 hit ratio와 flush 연산의 횟수를 측정한 결과를 토대로 버퍼 캐시 교체 정책들의 특징을 분석한다.

3.1 워크로드 분석

본 논문에서 버퍼 캐시 교체 정책의 성능을 비교하기 위하여 순차접근과 임의접근 워크로드 두 가지의 워크로드를 정의하였다. 순차접근 워크로드는 시간별로 접근하는 논리 주소 번호가 연속적인 입출력이 많은 워크로드이다. 그림 6의 좌측 그림은 순차접근 워크로드를 나타낸 것이다. 그래프의 x축의 값이 증가 할수록 논리주소 번호 역시 증가하는 패턴이 많은 것을 볼 수 있다. 순차접근 워크로드는 파일 전체에 대한 연산들로 이루어져있다고 이해하면 될 것이다. 이 워크로드는 20%의 파일에 80%의 연산이 발생한다.

임의접근 워크로드는 시간별로 접근하는 논리 주소 번호에 일정한 연관성이 없다. 그림 6의 우측 그림은 임

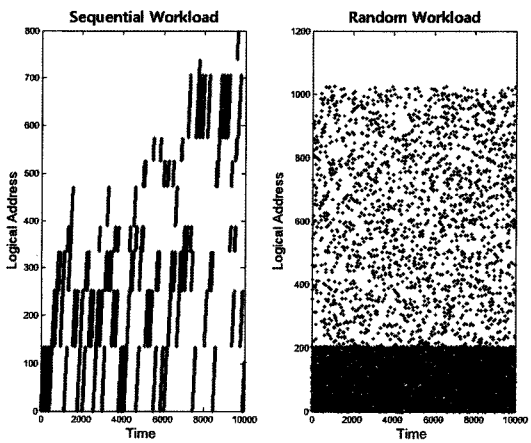

그림 6 순차접근 워크로드와 임의접근 워크로드의 시간별로 접근하는 논리 주소 번호

의접근 워크로드를 나타낸 것이다. 만약 어떤 논리주소에 접근할 확률이 일정하다면 버퍼 캐시의 성능을 측정하는데 어려움이 있으므로 참조지역성을 적용하여 논리주소의 20%에 총 연산의 80%가 발생하도록 하였다.

3.2 기존 버퍼 캐시 교체 기법의 성능 분석

기존의 하드디스크를 위한 버퍼 캐시 교체 정책과 플래시 메모리를 위한 버퍼 캐시 교체 정책의 성능 분석을 위하여 순차접근 워크로드와 임의접근 워크로드 두 가지로 나누어 실험하였다. Hit ratio는 성능의 비교를 위하여 Optimal 정책의 hit ratio를 100%라고 했을 때 상대적인 성능으로 기술하였다. Optimal 정책은 향후 가장 늦게 참조되는 페이지를 교체하는 기법으로 실제로 구현할 수는 없지만 버퍼 캐시 교체 정책의 성능 측정 시 기준이 되는 정책이다[4].

각 버퍼 캐시 교체 정책의 hit ratio 비교 결과는 그림 7과 같다. 실험 결과 순차 접근 워크로드의 경우 LRU의 hit ratio가 다른 정책들에 비해서 높았다. CFLRU와 그 변형들의 hit ratio 역시 높았으나 LRU에 비해서는 높지 못하였다. 이는 이 워크로드의 읽기 연산의 비율이 쓰기 연산의 비율에 비해서 높지만 CFLRU와 그 변형들은 clean 페이지를 우선적으로 교체하기 때문에 읽기 연산에 대한 버퍼 캐시의 페이지인 clean 페이지가 버퍼 캐시 내에 긴 시간동안 있을 가능성이 적다. 이와 같은 이유로 CFLRU와 그 변형들의 성능이 LRU에 비해서 낮다. DL-CFLRU/E의 경우 CFLRU의 극단적인 경우로 CFLRU는 일정 크기만큼 검색하여 clean 페이지를 찾지만 DL-CFLRU/E이 경우 clean 페이지를 따로 관리하기 때문에 CFLRU보다 clean 페이지가 교체될 가능성이 더 많아진다. 따라서 성능이 아주 좋지 않다. LFU는 2.1의 그림 2에서 설명한 것과 같은 이유로 성능 악화가 발생하므로 hit ratio가 낮아진다.

임의접근 워크로드의 경우 LFU가 다른 정책에 비하여 높은 hit ratio를 보여준다. 임의접근의 경우 20:80으로 참조 지역성이 있으므로 참조 횟수를 반영하는 LFU가 LRU에 비하여 좋은 성능을 보인다. DL-CFLRU/E

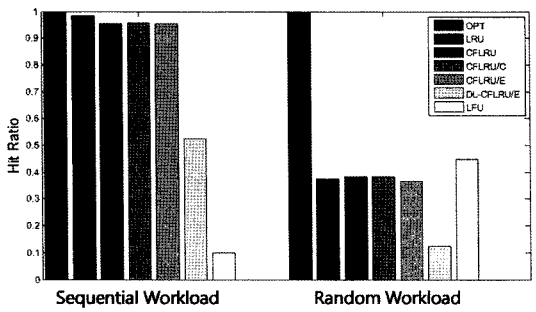

그림 7 기존 버퍼 캐시 교체 기법의 hit ratio 비교

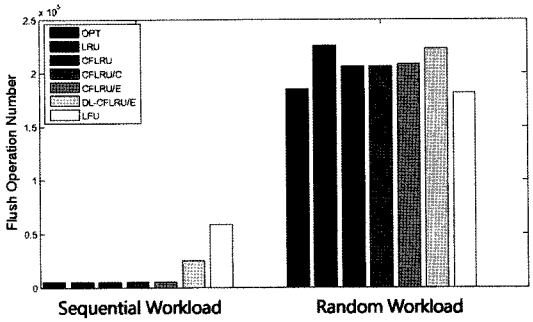

그림 8 기존 버퍼 캐시 교체 기법의 flush 횟수 비교

의 경우 순차접근과 마찬가지의 이유로 좋지 않은 hit ratio를 보인다.

Flush 횟수 비교 결과는 그림 8과 같다. Flush 횟수 비교는 hit ratio 비교와 유사한 결과를 보인다. 순차접근은 LRU가 가장 좋은 성능을 보였고 LFU가 가장 좋지 않은 성능을 보였다. 임의접근 워크로드는 LFU가 가장 좋은 성능을 보여 hit ratio와 유사함을 알 수 있었다.

4. 새로운 버퍼 캐시 교체 정책

플래시 메모리를 위한 새로운 버퍼 캐시 교체 정책은 두 가지 사항을 고려한다. 하나는 플래시 메모리를 위한 버퍼 캐시 교체 정책이므로 flush 횟수를 줄이기 위하여 clean 페이지를 우선적으로 교체한다. 다른 하나는 LFU와 같이 참조 횟수를 기반으로 하여 교체 페이지를 선택하도록 한다. LFU를 기반으로 하는 이유는 참조 횟수를 반영하면 참조 지역성이 있는 접근 패턴에 대해서 좋은 hit ratio를 기대할 수 있기 때문이다. 실험결과 임의 접근 워크로드에서 LFU의 hit ratio가 다른 버퍼 캐시 정책이 비하여 높았고 순차 접근 워크로드는 LFU의 hit ratio가 좋지 않았으나 그림 2에서 나타날 수 있는 상황을 막을 수 있으면 순차접근 워크로드에서도 좋은 성능을 보일 수 있을 것이다.

본 논문에서 제안하는 버퍼 캐시 교체 정책인 MLC-LFU(Multi-Level Clean-first Least Frequently Used)는 그림 9와 같다. 버퍼 캐시의 페이지를 관리하기 위한 리스트는 다단계 리스트로 구성되어 있다. 각 리스트는 참조 횟수 순으로 정렬되어 있고 각 페이지는 clean과 dirty를 구별할 수 있다. 각 단계의 리스트는 참조 횟수 순서로 구성된다. 리스트가 구성되는 과정은 4.2에서 다룬다.

4.1 Victim 선택

본 논문에서 제안하는 victim 선택 기법은 CFLFU (Clean First Least Frequently Used)라고 말할 수 있다. 즉 LFU와 동일하지만 clean 페이지를 우선적으로

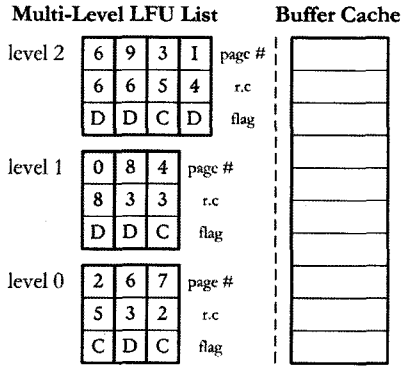

그림 9 MLC-LFU의 구조

```
function selectVictim()
targetList = list[level0]
victim = searchCleanPage(targetList)
if victim is NULL
    victim = targetList[last]
return victim
```

그림 10 victim 선택 알고리즘

선택하는 기법이다. clean 페이지를 우선적으로 교체함으로써 dirty 페이지가 플래시 메모리로 flush되는 것을 최대한 지연시킨다.

victim을 선택하기 위하여 최하위 단계의 리스트에서 참조 횟수가 가장 적은 clean 페이지를 찾는다. 만약 리스트 내에 clean 페이지가 없을 경우 참조 횟수가 가정 적은 dirty 페이지가 victim으로 선택된다. 그림 9에서 만약 페이지 교체가 필요하다면 level 0의 가장 우측 페이지가 clean 페이지이므로 교체 대상이 된다. clean 페이지가 모두 dirty 페이지로 교체된다면 level 0의 dirty 페이지들 중 참조 횟수가 가장 적은 페이지가 가장 우측에 위치할 것이므로 이 페이지가 교체 대상이 된다. 그림 10은 victim을 선택하는 알고리즘을 pseudo code로 나타낸 것이다. 이 알고리즘에서 searchCleanPage는 리스트를 역순으로 탐색하므로 이 알고리즘의 계산 복잡도는 $O(n)$이다.

4.2 버퍼 캐시 관리 리스트 업데이트

4.2절에서는 MLC-LFU의 버퍼 관리 리스트의 구성과 그 리스트를 효율적으로 관리하기 위한 합병 연산에 대하여 설명한다.

4.2.1 버퍼 캐시 관리 리스트의 구성

참조 횟수가 많은 clean 페이지를 오랫동안 유지하기 위하여 참조가 많이 된 페이지들은 상위 레벨의 리스트로 이동한다. 상위 레벨로 이동하기 위한 기준 참조 횟수인 Threshold(TH)값은 버퍼 캐시의 페이지 수로 한다. 어느 페이지의 참조 횟수가 TH와 같거나 이를 초과하게 되면 이 페이지는 상위 level의 리스트로 이동하게 된다. 만약 level 0의 페이지 참조 횟수가 모두 TH보다 같거나 초과하여 level 0가 비게 되면 level 1의 리스트가 level 0 리스트가 되고 상위의 리스트는 한 단계 아래의 리스트로 바뀌게 된다. 그림 11은 level 1 리스트에 존재하는 한 페이지의 참조 횟수가 TH를 초과하여 level 2 리스트로 이동하는 모습이다. TH가 10이라고 했을 때 level 1 리스트의 첫 번째 페이지에 2번의 참조가 추가적으로 발생하여 TH와 같아졌으므로 이 페이지를 상위 레벨인 level 2 리스트로 이동시킨다. 상위 레벨로 이동하게 되면 참조 횟수는 1로 초기화한다.

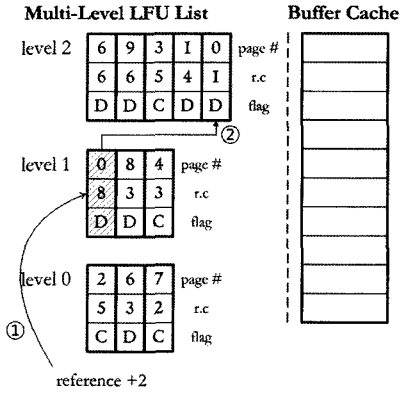

그림 11 버퍼 캐시 관리 리스트 업데이트

4.2.2 버퍼 캐시 관리 리스트의 합병 연산

다중 레벨로 리스트를 관리하게 되면 참조 횟수의 차이가 각 페이지 별로 많이 나서 다중 리스트의 각 리스트에서 관리하는 페이지가 하나인 상황이 발생할 수 있다. 각 리스트에서 페이지를 하나씩 관리하는 것은 MLC-LFU의 레벨을 과도하게 늘릴 수 있으므로 이 리스트들을 관리하는 것은 버퍼 캐시 시스템에 부담이 될 수 있다. 따라서 MLC-LFU는 이 경우 리스트들 간의 합병 연산을 수행한다. 합병 연산은 그림 12와 같다. 버퍼 캐시 관리 리스트 업데이트 시 새로운 상위 레벨로의 페이지 이동이 발생하면 최상위부터 아래 레벨로 리스트들을 순회하면서 리스트가 관리하는 페이지의 수를 검사한다. 만약 관리하는 페이지가 1인 리스트가 연속으로 3개 이상이면 이 리스트를 하나의 리스트로 합치게 된다. 이를 합병 연산이라고 정의한다. 합병 연산이 결과 세 리스트가 하나로 합쳐지고 그 리스트는 세 리스트 중 가장 하위의 리스트가 된다. 그리고 이후 상위 리스트는 리스트 level이 조정되게 된다.

합병 연산이 수행되면 참조횟수에서 문제가 발생한다. 한 페이지의 참조 횟수가 TH보다 높아져 상위레벨로 이동하면 참조 횟수가 1로 초기화되는데 합병하고 참조

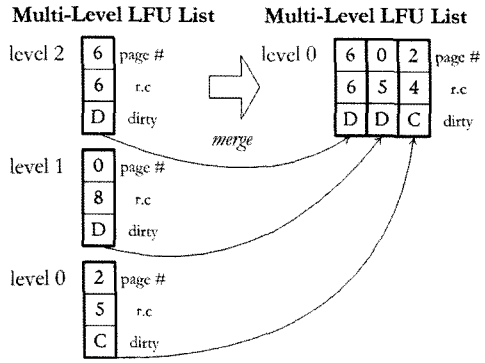

그림 12 버퍼 캐시 리스트의 합병 연산

횟수를 수정하지 않으면 합병된 리스트 내에서 참조 횟수가 뒤섞이게 된다. 이는 그림 12에서도 볼 수 있다. level 2에 있는 페이지의 참조 횟수는 6이고 level 1에 있는 페이지의 참조 횟수는 8이다. 이 리스트들을 합병하게 되면 level 2의 페이지가 분명 참조 횟수가 많음에도 리스트 상에서는 level 1의 페이지가 level 2보다 참조 횟수가 많은 것으로 나타나게 된다. 따라서 합병 후 참조 횟수 변경이 필요하다. 본 논문에서는 참조 횟수를 수정하는 방법을 간단하게 하기 위하여 가장 상위 레벨의 참조 횟수는 그대로 유지하고 하위 레벨의 참조 횟수들은 상위 레벨의 참조 횟수보다 작게 하는 방법을 사용한다. 이러한 방법을 사용하려면 합병 대상 중 최상위 레벨의 참조 횟수는 3보다 크거나 같아야 한다. 3보다 작은 경우 참조 횟수를 3으로 조정하고 하위 레벨의 참조 횟수는 3보다 작게 함으로써 해결한다.

5. 성능 평가

본 논문에서 버퍼 캐시 교체 정책들의 성능을 평가하기 위하여 시뮬레이션을 통하여 실험을 하였다. 실험을 위하여 다음과 같이 가정하였다. 먼저 저장장치인 NAND 플래시 메모리는 1024개의 페이지로 구성되어 있다. 편의를 버퍼 캐시의 페이지와 NAND 플래시 메모리의 페이지 크기는 같다고 가정하였다. 버퍼 캐시의 페이지 프레임의 수는 16~40개로 2개씩 늘려서 실험을 하였다. 실험 대상이 되는 버퍼 캐시 교체 정책은 Optimal, LRU, CFLRU, CFLRU/C, CFLRU/E, DL-CFLRU/E, LFU와 본 논문에서 제안하는 기법이다. 위의 정책들을 순차접근 워크로드와 임의접근 워크로드로 나누어 버퍼 크기를 조절하면서 실험을 하였다. 그림 13은 버퍼 캐시 크기 변화에 따른 각 정책들의 hit ratio를 비교한 것이다.

그림 13의 hit ratio는 optimal 정책의 hit ratio를 1이라고 했을 때 상대적인 hit ratio이다. 순차접근 워크로드에서 본 논문에서 제안하는 기법의 hit ratio가 다

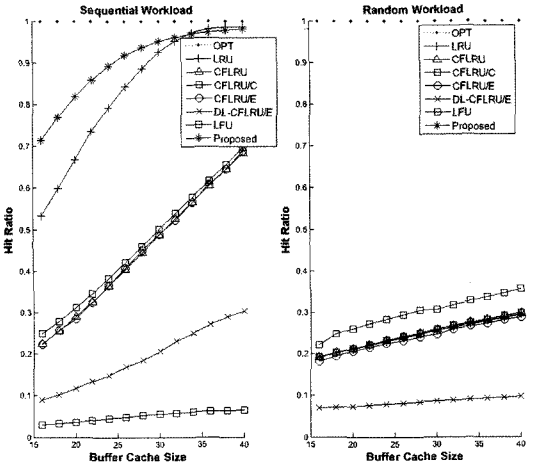

그림 13 버퍼 캐시 크기 변화에 따른 hit ratio 비교

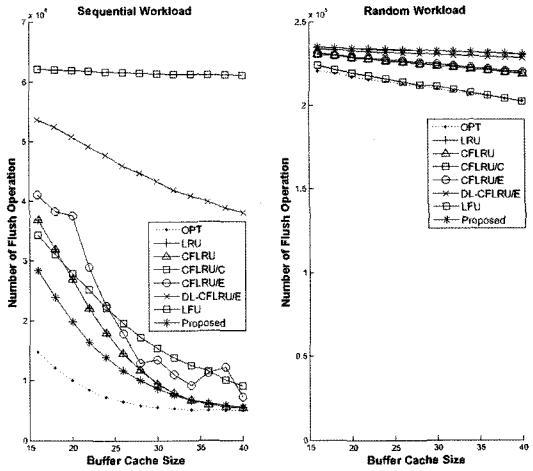

그림 14 버퍼 캐시 크기 변화에 따른 flush 횟수 비교

른 기법에 비하여 순차접근 워크로드에서 실험결과 본 논문에서 제안하는 기법의 hit ratio가 높은 것을 볼 수 있다. 이 차이는 버퍼 캐시의 크기가 작을수록 높게 나타났고 버퍼 캐시 크기가 커지면 LRU가 높은 hit ratio 증가율을 보여 제안하는 기법과 비슷한 hit ratio를 보임을 알 수 있다. 하지만 LRU의 hit ratio가 커지기 위해서는 버퍼 캐시의 크기도 충분히 커야하므로 일반적으로 제안하는 기법의 더 용이하다고 할 수 있다.

임의접근 워크로드에서는 LFU가 가장 높은 hit ratio를 보였고 DL-CFLRU/E는 가장 낮은 hit ratio를 보였다. 제안하는 기법의 hit ratio는 LFU보다 약 4% 낮게 나타났고 LRU나 CFLRU 등과는 비슷한 성능을 보였다.

그림 14는 버퍼 캐시 변화에 따른 각 정책들의 flush 연산 횟수를 비교한 것이다. 순차접근 워크로드에서 제안하는 기법의 flush 횟수가 다른 정책에 비하여 20~

120% 낮게 나타났다. 하지만 임의접근 워크로드에서는 제안하는 기법이 다른 정책들에 비하여 1~12% 높은 flush 횟수를 보였다. 순차접근 워크로드에서는 본 논문에서 제안하는 기법이 플래시 메모리의 특성을 잘 반영하였으나 임의접근 워크로드에서는 반대로 제안하는 기법의 한계가 드러났다고 할 수 있다. 하지만 순차접근 워크로드에 비하여 증가하는 flush 횟수의 차이가 적고 실제 입출력 연산의 경우 임의접근 워크로드 만으로 이루어지는 것이 아니라 순차접근과 임의접근이 섞여있는 형태이기 때문에 실제 워크로드에서는 얻을 수 있는 이득이 클 것이라 예상된다.

6. 결 론

본 논문에서 플래시 메모리를 위한 버퍼 캐시 교체 정책을 제안하고 실험/평가를 하여 기존의 CFLRU나 그 변종들보다 좋은 성능을 가짐을 보였다. 이 기법은 순차접근 워크로드에서는 플래시 메모리를 위한 버퍼 캐시 교체 정책 뿐 아니라 LRU나 LFU보다 좋은 hit ratio와 낮은 flush 횟수를 보였고 임의 접근에서도 LRU와 유사한 성능을 보였다. 향후 연구를 통하여 임의 접근에서도 LFU보다 좋은 성능을 보이는 버퍼 캐시 교체 정책으로 발전할 수 있을 것이다. 그리고 실제 시스템에서 구현하고, 실제 워크로드로 실험을 하여 기존의 정책과 연산량의 증가 및 버퍼 캐시 교체 정책의 효율성을 검증하도록 하겠다 본 논문에서 제안하는 버퍼 캐시 교체 정책을 발전시켜 플래시 메모리 뿐만 아니라 하드디스크나 이종 저장장치(Heterogeneous Storage)를 위한 버퍼 캐시 교체 정책에도 응용할 수 있을 것이다.

참 고 문 헌

[1] ONS HS2500, http://www.ons.co.kr/

[2] R. Bez, E. Camerlenghi, A. Modelli, and A.Visconti, "Introduction to Flash Memory," *Proceedings of the IEEE*, Vol.91, No.4, Apr. 2003.

[3] E. Gal, S. Toledo, "Algorithms and data structures for flash memories," ACM Computing Surveys, 37(2), 2005.

[4] A. Silberschatz, P. Garvin, G. Gagne, "Operating System Concepts," *John Wiley & Sons*, 2003.

[5] S. Park, D. Jung, J. Kang, J. Kim, J. Lee, "CFLRU: a replacement algorithm for flash memory," *2006 international conference on Compilers, architecture and synthesis for embedded systems*, 2006.

[6] Y. Yoo, H. Lee, Y. Ryu, H. Bahn, "Page Replacement Algorithms for NAND Flash Memory Storages," *ICCSA 2007, LNCS 4705*, Part I, 2007.

옥 동 석

2008년 부산대학교 컴퓨터공학과 졸업(학사). 2008년~현재 부산대학교 컴퓨터공학과 석사과정. 관심분야는 내장형 시스템, 파일 시스템

이 태 훈

2004년 부산대학교 정보컴퓨터공학과(학사). 2006년 부산대학교 컴퓨터공학과 석사. 2006년~현재 부산대학교 컴퓨터공학과박사과정. 관심분야는 내장형 시스템, 파일 시스템

정 기 동

1973년 서울대학교 졸업(학사). 1975년 서울대학교 대학원 졸업(석사). 1986년 서울대학교 대학원 계산통계학과 졸업(이학박사). 1978년~현재 부산대학교 컴퓨터 공학과 교수. 관심분야는 멀티미디어 시스템, 멀티미디어 통신, 병렬처리