

플래시 메모리상에서 효율적인 버퍼 페이지 교체 기법

박종민^o, 박동주
 송실대학교 대학원

{ jmpark^o, djpark } @computing.ssu.ac.kr

An Efficient Buffer Page Replacement Strategy on Flash Memory

Jongmin Park^o, Dong-Joo Park
 School of Computing, Soongsil University

요 약

플래시 메모리는 오늘날 다양한 형태로 우리 생활의 일부를 차지하고 있다. 휴대 전화기, MP3 플레이어, PDA 등과 같은 모바일 제품, 이동식 저장매체, 유비쿼터스 컴퓨팅 환경 등에 광범위하게 활용되고 있다. 이처럼 많은 분야에서 사용되는 주된 이유는 플래시 메모리의 장점인 저전력, 비휘발성, 고성능, 물리적 안정성, 휴대성을 갖기 때문이다. 더불어 최근에는 Gb급 플래시 메모리도 개발되어 하드디스크의 자리를 대체할 수 있는 상황에 이르렀다. 하지만, 플래시 메모리는 하드디스크와 달리 이미 데이터가 기록된 블록에 대해 덮어쓰기(overwrite)가 되지 않는다는 특성을 갖고 있다. 덮어쓰기 위해서는 해당 블록을 지우고(즉, 소거(erase)) 쓰기 작업을 수행해야 한다. 이로 인해 플래시 메모리의 데이터 읽기/쓰기/소거에 비용이 하드 디스크와 같이 동일한 것이 아니라 서로 다르다(읽기 비용을 1로 가정할 경우 쓰기와 소거는 각각 8, 65)[1][5][6]. 따라서 OS, DBMS 등과 같은 시스템 소프트웨어에서 사용된 기존 버퍼 교체 기법은 플래시 메모리의 특성이 고려되지 않았기 때문에 플래시 메모리의 특성을 고려한 효율적인 버퍼 교체 기법이 필요하다. 본 논문에서는 플래시 메모리의 서로 다른 연산 비용 고려한 새로운 버퍼 교체 기법을 제안한다.

1. 서 론

소비전력이 적고, 전원이 꺼지더라도 저장된 정보가 사라지지 않는 비휘발성의 특성을 지닌 플래시 메모리는 하드디스크와 같은 기존의 디스크에 비해 데이터 접근 성능이 좋고 크기가 매우 작다. 또 물리적인 충격에 있어서도 하드디스크에 비해 강하고 무게 역시 가볍다는 장점을 갖고 있다. 이러한 특성을 가진 플래시 메모리는 최근 MP3 플레이어, 휴대 전화기, 개인정보단말기(PDA), 디지털 카메라/캠코더 등의 휴대용 정보 기기들의 보조기억장치로 급속도로 사용되기 시작했다[1].

최근 플래시 메모리의 사용 분야는 다양해지고, 그와 더불어 플래시 메모리의 저장 용량도 급속도로 대용량화 되어가는 추세이다. 플래시 메모리의 대용량화 추세는 기존 하드디스크의 역할이 플래시 메모리로 대체 될 수 있음을 예견한다. 하드디스크의 역할을 대체한다는 것은 운영체제(OS)나 데이터베이스 시스템(DBMS)과 같은 시스템 소프트웨어의 사용이 플래시 메모리 상에서 이루어 질 수 있음을 말한다. 일반적으로 OS나 DBMS와 같은 시스템 소프트웨어는 시스템의 메인 메모리 공간 일부를 메모리 버퍼(memory buffer) 영역으로 할당한다. 이렇게 할당된 메모리 버퍼는 프로그램 실행 시 빈번하게 접근되는 디스크 페이지를 보관하여 동일 디스크 페이지 접근 시 메모리 버퍼에 보관된 페이지로 접근하여 디스크 접근에 필요한 비용을 줄일 수 있다. 만약 메모리 버퍼에 더 이상의 여유 공간이 없다면 다른 페이지를 버퍼에 보관하기 위해 교체 대상 페이지를 선정해야 한다. 이때 어떠한 기준에 의해 교체 대상 페이지를 선정할 것인지에 따라 메모리 버퍼의 성능인 버퍼 집중률이 달라진다.

다음 2장에서는 하드디스크에서 사용되는 대표적인 페이지 교체 기법과 플래시 메모리에서 사용될 경우 발생하는 문제점을 이야기한다. 3장에서는 논문에서 제안 기법을 설명한다. 4장에서는 다양한 실험을 통해 성능 평가를 하며, 5장에서 결론

을 맺는다.

2. 배경지식 및 연구동기

하드디스크에서 메모리 버퍼가 필요한 시스템 소프트웨어가 사용 될 경우 시스템의 메인 메모리의 일부를 메모리 버퍼로 사용한다. 메모리 버퍼에 사용되는 페이지 교체 기법은 기존에 많이 제안되었다. 그 중 대표적인 기법들로 LRU[2], LFU[2], MFU[2]가 있다. 이들 기법은 메모리 버퍼에 보관되는 시간과 보관 후에 참조되는 횟수를 기준으로 만들어진 것이다. 하드디스크에서는 페이지를 읽어 메모리 버퍼로 보관시키는 비용과 메모리 버퍼에서 교체 대상으로 선정된 페이지를 다시 하드디스크로 쓰는데 사용되는 비용이 동일하다. 하지만 플래시 메모리는 읽기/쓰기/소거 연산의 비용이 다르고 쓰기연산의 경우 읽기연산의 약 8배의 비용이 든다. 덮어쓰기(overwrite)가 발생 할 경우 소거연산을 반드시 수행하게 된다. 이때 소거연산은 읽기연산의 약 65배의 비용이다. 따라서 하드디스크가 아닌 플래시 메모리에서 버퍼 페이지 교체 기법으로 사용될 경우에는 연산의 비용이 다르기 때문에 교체 기법에서 고려하지 않은 페이지 교체 기법의 사용은 예상치 못한 비용을 초래할 수 있다. 예를 들어 덮어쓰기가 예상되는 버퍼 페이지가 빈번하게 교체 페이지로 선정되면서 비용은 증가하게 된다.

따라서 플래시 메모리상의 버퍼 페이지 교체 기법은 기존 하드디스크에서 사용하던 페이지 교체 기법과는 달리 교체 예상 비용을 적용한 새로운 기법이 필요하다.

3. FLRU (Flash Memory LRU) 기법: 플래시 메모리 페이지 교체 전략

FLRU 기법의 제안 목적은 기존 기법들이 하드디스크가 아닌 플래시 메모리 상에서 이루어질 때는 페이지 교체 비용 부분을 고려하지 않아 부적합 하므로 각각의 예상 페이지 교체 비용이 고려되어야 올바른 페이지 교체 기법이 될 수 있음을 보이는 것이다. 즉, FLRU 기법에서는 플래시 메모리 특성으로 인해

* 본 연구는 한국학술진흥재단의 지원에 의하여 수행되었음.

(KRF-2004-005-D00172)

달라지는 페이지 연산 비용을 예측하는 것이다. 예측이 필요한 이유는 앞서 설명한 플래시 메모리의 읽기/쓰기/소거 연산의 비용이 각각 다르기 때문에 기존 기법들을 그대로 사용하는 경우에는 메모리 버퍼의 사용이 본래 목적과는 달리 큰 비용을 초래할 수 있게 된다.

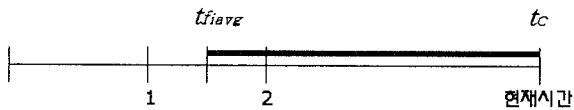
FLRU 기법은 페이지의 연산 비용을 FTL (Flash Translation Layer)을 통해 예측한다. FTL이란 시스템에서 입력되는 논리적 번지를 플래시 메모리의 물리적 번지로 변환하는 사상 계층이다. 버퍼에 보관된 페이지를 통해 해당 논리적 번지를 알 수 있고, FTL을 통해 논리적 번지의 사상되는 물리적 번지를 알 수 있다. 물리적 번지는 플래시 메모리의 상태를 알 수 있게 한다. 예를 들어 물리적 번지의 위치에 이미 저장된 데이터가 존재한다면 덮어쓰기 연산이 될 수 있음을 알 수 있다. 이러한 방식으로 교체 대상으로 선정된 페이지의 연산을 예측할 수 있다.

FLRU 기법에서 사용하는 요소들로는 참조시간, 참조 횟수와 예측 비용이 있다. 먼저 교체 페이지의 연산 비용을 예측하기 위해서 플래시 메모리의 FTL을 추적하게 된다(이때, FTL은 플래시 메모리의 S-RAM에 저장 되므로 I/O비용으로 간주하지 않는다).

$$C = \left(\frac{t_c - t_{fiavg}}{f_i \times ERC} \right)$$

<수식 1. FLRU 기법>

<수식 1>은 FLRU 기법을 수식으로 표현한 것이다. 먼저 ERC (Expected Replacement Cost)는 버퍼에 보관된 페이지가 교체 대상으로 선정되었을 때 해당 페이지가 쓰일 섹터의 상태에 따라 크기가 달라진다. 메모리 버퍼에 읽기 연산으로 보관된 상태의 페이지가 교체 대상 페이지로 선정되면 교체 비용은 발생하지 않는다. 하지만 쓰기 상태면 플래시 메모리의 상태에 따라 쓰기/덮어쓰기 두 경우가 발생할 수 있다. 앞서 설명했듯이 쓰기일 경우는 읽기의 8배의 비용이 발생하므로 ERC 값은 8에 상태를 추적하는데 발생하는 비용을 더한 값이 된다. 덮어쓰기 경우에는 ERC의 값 65에 상태 추적하는데 발생하는 비용을 더해준다. 분모 부분은 ERC 값과 버퍼 페이지 참조 횟수(f_i)를 곱해 각 페이지의 가중치를 갖게 된다.



<그림 1. 버퍼 페이지 시간 개념도>

<그림 1>은 특정 페이지가 시점 1일 때 처음 버퍼에 보관되고 시점 2일 때 참조가 발생했을 경우 두 번의 참조 시간의 평균 시점(t_{fiavg})과 현재 시점(t_c)의 차이를 실선으로 표현한 것이다.

<수식 1>에서 분자 부분에 해당하는 값(실선)을 앞서 구한 페이지의 가중치로 나누어 값 C를 얻는다. 이렇게 얻은 C 값이 교체 대상 페이지 선정 여부를 가리는 값이 된다. 다시 말해 C 값이 큰 페이지를 교체 대상으로 선정한다. FLRU 기법의 가장 큰 변수가 되는 ERC 값은 웹 캐싱 기법의 대다수가 중요 변수로 웹 문서의 크기를 적용하는 방식에서 착안했다 [4].

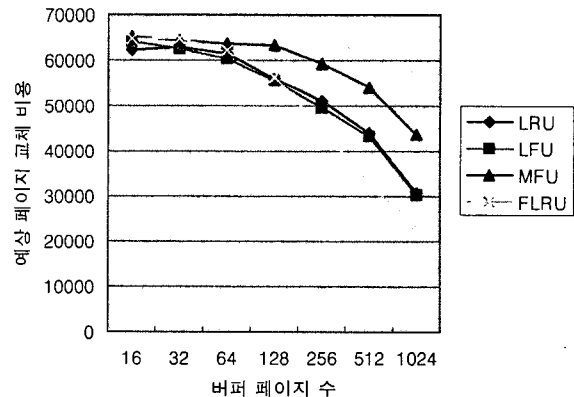
FLRU 기법에서는 페이지가 플래시 메모리에 저장되는데 사

용되는 예상 비용을 ERC로 측정하고, ERC 값을 적용한 후 FLRU 기법을 통해 구해지는 C 값은 버퍼에 보관된 모든 페이지마다 존재하게 된다. 새로운 버퍼 페이지가 필요한 시점에서 보관된 모든 페이지 중 C 값이 가장 큰 페이지가 교체 대상 페이지로 선정된다. 교체 대상 페이지가 플래시 메모리에 저장되는데 사용되는 비용은 ERC가 된다.

4. 성능 평가

2장에서 언급한 페이지 교체 기법들과 FLRU의 성능 비교 평가를 위해 사용한 FTL은 미쓰비시사에서 제안한 FTL을 사용했다. 이전 FTL은 동일 논리적 번지에 쓰기 연산을 수행하고 덮어쓰기가 일어나면 반드시 블록 교체와 소거연산이 일어나며 비용 역시 증가한다. 미쓰비시 FTL은 덮어쓰기 시 발생하는 블록 소거 연산을 지연시키기 위해 미리 지정된 수의 스페어 섹터(spare sector)에 덮어쓰기 데이터를 입력한다. 그래서 종전의 덮어쓰기로 인해 발생했던 블록 소거를 지연시키므로 소거 연산의 비용도 스페어 섹터 수만큼 줄일 수 있다. 하지만 블록마다 지정해 둔 스페어 섹터를 유지하기 때문에 플래시 메모리의 공간 활용률이 떨어진다. 이러한 문제점은 본 실험의 결과에 크게 적용하지 않는 부분이다.

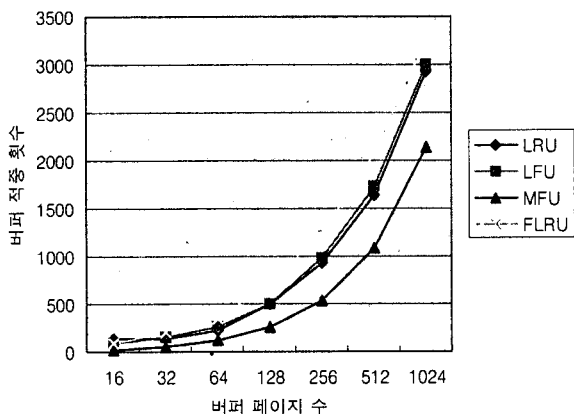
성능 평가에서 사용된 워크로드(workload)는 지프 분포(Zipfian distribution)[3]를 통해 실험 공간의 최대 주소번지를 감안한 0부터 5000사이의 수 1만개를 생성했다. 지프 분포는 지프 값에 따라 달라지며, 0.1, 0.5, 0.9 세 가지의 지프 값을 사용했다. 0.1일 경우는 생성되는 수가 균등하게 분포된다. 다시 말해 0부터 5000사이 대부분의 수가 1회 이상 나타난다. 이와 반대로 0.9일 경우는 0부터~5000사이 일부분의 수가 50% 이상 나타난다. 두 경우 중간 수준으로 반복 되는 수를 0.5에서 볼 수 있다. 이렇게 지프 분포에 의해 생성된 수들을 연산의 입력으로 사용되는 논리적 섹터 번지(LSN: Logical Sector Number)로 사용했다. 플래시 메모리는 연산에 따라 비용이 다르기 때문에 읽기/쓰기 연산을 나타내는 구분자가 필요하여 이 역시 지프 값으로 생성하여 1만개의 LSN에 표시 후 실험했다. 다양한 버퍼 페이지를 설정 후 측정된 값은 교체 대상으로 선정된 페이지들의 교체 예상 비용 ERC의 누적 값과 버퍼 페이지 적중 수를 측정했다.



<그림 2. 지프 값 0.5일 경우 예상 페이지 교체 비용>

<그림 2>는 지프 값 0.5의 실험 결과 중 예상 페이지 교체 비용을 보인다. 128개 이하의 버퍼 페이지 수를 가지는 경우에는 기법들의 성능 차이가 크지 않았다. 이러한 결과는 워크로드

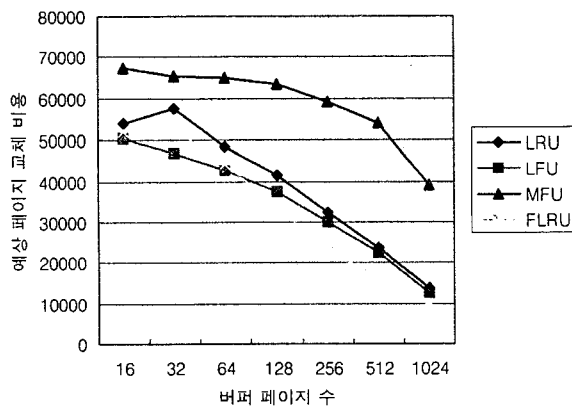
드 패턴 중 중복 LSN이 버퍼 페이지의 수 보다 큰 간격으로 나타나기 때문이다. 그러나 중복 LSN을 포함할 수 있는 버퍼 페이지 수가 128개 이상이 되면서부터는 FLRU가 최소의 페이지 교체 비용을 가지는 것을 알 수 있다.



<그림 3>은 지프 값 0.5일 경우 버퍼 적중 횟수

<그림 3>은 지프 값 0.5일 경우 버퍼 적중 횟수를 보여주는 그래프다. 모든 버퍼 페이지에서 FLRU가 높은 버퍼 적중 횟수를 보인다. 증가율로 보면 버퍼 페이지 256개의 경우 FLRU가 두 번째 높은 LFU에 비해 약 21% 증가했다. 지프 값 0.5일 경우에서 예상 페이지 교체 비용과 버퍼 적중 횟수에서 다른 기법들에 비해 FLRU가 우수함을 알 수 있다.

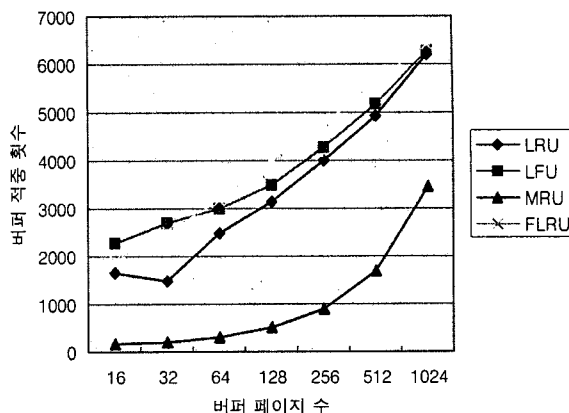
그림에는 없지만 지프 값이 0.1인 경우의 결과를 보게 되면 전체적으로 비슷한 수준으로 교체 비용과 버퍼 적중 횟수를 보여준다. 이러한 이유는 지프 값 0.1의 워크로드가 균등하게 존재하므로 다양한 수의 LSN이 적은 수의 반복을 갖는다. 그러므로 버퍼의 활용도가 떨어져 실험대상의 모든 기법들의 결과가 비슷한 수준을 보여준다.



<그림 4>는 지프 값 0.9일 경우 예상 페이지 교체 비용

지프 값 0.9인 경우는 0.1과 0.5에 비해 일부 LSN이 많은 수의 중복 횟수를 갖는다. <그림 4>는 지프 값 0.9의 페이지 교체 예상 비용이 0.1과 0.5의 결과에 비해 전체적으로 줄어들었음을 알 수 있다. 버퍼 페이지가 256개의 경우 FLRU가 LRU의 약 71%, LFU의 약 76% 수준으로 페이지 교체 비용이 감

소함을 보여준다.



<그림 5>는 지프 값 0.9일 경우 버퍼 적중 횟수

<그림 5>에서 지프 값 0.9의 버퍼 적중 횟수는 128개의 버퍼 페이지일 경우 LFU의 약 13% 증가한다. 버퍼 페이지 수 1024개의 경우는 워크로드의 전체 LSN 중 특정수들이 빈번하게 중복되기 때문에 MRU가 아닌 다른 기법들의 결과가 비슷하게 나타난다. 하지만 그 이하의 경우는 거의 모든 경우 모든 교체 기법보다 향상됨을 보인다.

5. 결론 및 향후 연구

플래시 메모리를 하드디스크 대응으로 사용하게 될 때 메모리 버퍼의 페이지 교체 기법은 플래시 메모리 특성에 맞게 페이지의 교체 비용을 고려해야 효율적인 메모리 버퍼 활용이 가능하게 된다. 페이지 교체 비용을 예측하지 않고 메모리 버퍼를 사용하면 예기치 못한 비용으로 인하여 메모리 버퍼 사용의 본래 목적을 벗어나게 된다. FLRU 기법을 통해 효율적인 메모리 버퍼 활용이 가능함을 알 수 있었다. 앞으로 최신 FTL인 로그 블록 기법[5]을 사용하여 실험하고, 다양한 실제적 패턴의 워크로드를 사용하여 실험할 계획이다.

참고 문헌

1. Tae-Sun Chung, Dong-Joo Park, Yeonseung Ryu, Sugwon Hong: LSTAFF: System Software for Large Block Flash Memory. AsiaSim 2004: 704-712
2. A. Silberschatz, P. B. Galvin, and G. Gagne, Operating System Concepts sixth edition, Wiley, 2003
3. E. J. O'Neil, P. E. O'Neil, and G. Weikum, The LRU-K Page Replacement Algorithm For Database Disk Buffering. SIGMOD Conference 1993: 297-306
4. S. Podlipnig, L. Böszörményi, A Survey of Web cache replacement strategies. ACM Comput. Surv. 35(4): 374-398 (2003)
5. J.Kim, J.M. Kim, S.H. Noh, S.L. Min, and Y. Cho, A Space-Efficient Flash Translation Layer for Compact Flash Systems, IEEE Transactions on Consumer Electronics, Vol. 48 No. 2, pp 366-375, 2002
6. Chin-Hsien Wu, Li-Pin Chang, and Tei-Wei Kuo, An Efficient B-Tree Layer for Flash-Memory Storage Systems, RTCSA 2003, 409-430