LFR기법을 이용한 블럭교체 기법에 관한 연구

오재환*, 김상수*, 김미선**
한남대학교 전자공학과*, 우송대학교 전자정보공학과**

On the study of block replacement policy using LFR

Jea-han Oh*, Sang-su Kim*, Mi-sun Kim**
Dept. of electronics, Hannam Univ.*, Dept. of electronics information, Woosong Univ.

Abstract

Most popular disk block replacement polices are LRU(Least Recently Used)policy and LFU(Least Frequently Used)policy. The LRU policy replaces blocks according to the most recent reference without considering the frequency of reference. The LFU policy replaces blocks according to the frequency of reference without considering the recently of the reference. In this thesis, a policy called LFR(Least Frequently Used & Not Used Recently) disk block replacement policy is presented. The LFR policy subsumes the LFU policy and the NUR policy. The spectrum of the LFR policy exists between the LFU policy and the NUR policy because we con give different weight to each reference of a block. The implementation shows LFR policy outperforms the previously implemented LRU policy.

1. 개 요

모든 명령어 사이클 동안에 CPU는 명령어를 읽어오기 위해 적어도 한 번은 기억장치를 액세스(Access)하며, 오퍼랜드(operand)를 인출하거나 결과를 저장하기 위하여 기억장치를 한번 또는 그 이상 엑세스한다. 기억장치의 속도가 프로세서 속도를 따라잡지 못하는 일반적인 조건하에서 기억장치의 엑세스는 명령어의 실행에서 많은 시간을 소비하며, 명령어의 실행시간을 단축해줄 해결책으로서 캐쉬(Cache)라고 불리는 CPU와 주기억장치 사이에 작고 빠른 기억장치를 설치하여 CPU가 주기억장치를 엑세스하는 시간을 줄이는 방법이 널리 사용되고 있다[1]. 최근 CPU의 동작속도가 수 백MHz정도로 고속동작을 함에 따라서 캐쉬의 내용을 변경시키는 블록 교체 기법(block replacement policy)의 선택이 매우 중요하며, 효율적인 블록 교체 기법의 고안은 운영체제[2, 4]와 데이터 베이스 분야[5, 6, 7]의 오래된 연구분야이다. 이에 본 논문에서는 새로운 블록 교체 기법인 LFR(Least Frequently used & Not Used Recently)기법을 제안하고 컴퓨터 시뮬레이션을 통하여 제안한 기법의 우수성을 증명하였다.

2. 캐쉬 메모리와 블록교체 기법

2.1 캐쉬 메모리

캐쉬는 기억장치 속도가 가능한한 가장 빠른 기억장치의 속도에 접근하도록 함과 동시에 저렴한 반도체 기억장치의 가격으로 큰 기억장치 용량을 가질 수 있도록 하기 위한 것이다.

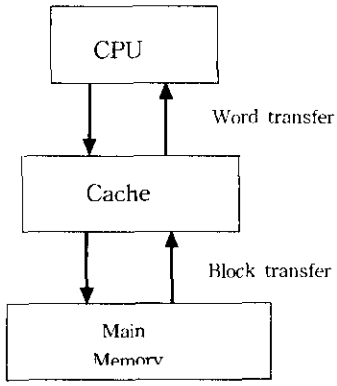

그림 2.1 캐쉬와 주기억장치

이 개념은 그림 2.1에 나타나 있는 바와 같이, 대용량의 기억장치가 고속의 캐쉬 기억장치와 함께 구성되는 것이다.

캐쉬는 주기억 장치 내용의 일부분을 복사하여 저장하고 있다. CPU가 기억장치로부터 한 단어를 읽으려고 할 때, 먼저 그단어가 캐쉬에 있는 지를 검사하고 만약 있으면, 그 단어가 즉시 CPU로 전달된다. 그렇지 않으면, 어떤 정해진 수의 단어들로 구성된 블록이 주기억장치로부터 캐쉬로 읽혀지고, 그런 다음에 그 단어가 CPU로 전달된다.

2.2 블록 교체 기법

CPU가 캐쉬로부터 한 단어를 읽으려 할 때 만약 캐쉬에 필요하는 단어가 없다면 CPU는 캐쉬의 어느 한 블럭을 버리고 CPU가 원하는 단어가 있는 블록으로 블록교체를 수행하여야 한다.

일반적으로 사용되는 블록교체 기법에는 참조 빈도수는 고려하지 않고 가장 최근의 참조만을 가지고 블록을 교체하는 LRU (Least Recently Used) 기법, 참조 시간은 고려하지 않고 빈도수만을 가지고 블록을 교체하는 LFU(Least Frequ- ently Used)기법, 최근에 참조되지 않은 블록을 교체하는 NUR(Not Used Recently)기법, 그리고, 참조빈도와 참조시간은 고려하지 않고 들어온 순서대로 블록을 교체하는 FIFO (First In First Out)기법 등이 있다.

LFU 기법은 각 블록의 참조 빈도수를 유지하며 블록 교체시 참조 빈도수가 가장 낮은 블록을 선택한다. 이 기법은 과거에 자주 참조되었던 블록은 가까운 미래에 다시 참조될 것이라는 가정에 기반하고 있다. 이러한 특성으로 인해 LFU 기법은 과거의 모든 참조를 고려하기는 하지만 참조 시간은 고려하지 못한다. 즉, 각각 다른 시간에 발생한 참조를 모두 같게 취급한다. 이러한 특성은 변화하는 작업에 대해 효과적으로 대처할 수 없다. 예를 들면 LFU 기법은 참조 빈도수에 기반하기 때문에 과거에는 자주 참조되었지만 현재에는 참조되지 않는 블록 대신 현재 자주 참조되는 블록을 교체할 수 있다는 단점이 있다.

또한, NUR 기법은 LRU 기법과 유사한 기법으로 최근에쓰여지지 않은 블록을 교체하는 기법이다. 즉, 가장 최근의 참조만을 고려하는 기법이다. NUR 기법에서 최근에 쓰이지 않은 블록은 가까운 미래에도 쓰이지 않기 쉬우며 따라서 이러한 블록들이 새로 들어오는 페이지들과 교체되는 것이다. 이러한, 특성때문에 NUR 기법은 과거에 참조되었던 블록과 그렇지 못한 블록을 구분 할 수 없다. 이 기법의 장점은 다른 기법에 비해 변화하는 작업에 대해 효과적으로 대처할 수 있다는 것이다. 또 하나의 장점은 아주 적은 오버헤드만이 생성되며, 구현이 매우 쉽다는 것이다. 그러나, 하나의 프로그램이 여러 페이지로 구성되는 커다란 루프(Loop)를 가지고 있을 때는 가장 오랫동안 사용되지 않은 블록이 바로 가장 가까운 장래에 사용되어질 블록이 바로 가장 가까운 장래에 사용되어질 블록이 된다는 것이 큰 단점으로 작용된다[1].

3. LFR(Least Frequently used & Not Used Recently)블록 교체 기법

참조 빈도수만을 고려하는 LFU 기법이나 가장 최근의 참조 시간만을 고려하는 NUR 기법과는 달리 LFR 기법은 한 블록을 교체하려 할 때 참조 빈도수와 참조 시간을 동시에 고려한다. 또한, LFR 기법은 한블록에 대해 미래에 참조될 확률을 계산하기 위해 그 블록에 대한 과거의 모든 참조를 고려한다.

LFR기법에서 블록의 재 참조 가능성은 블록의 가치로 표현되는데, 디스크 블록의 가치는 블록에 대한 참조가 발생할 때마다 증가하며 시간이 지남에 따라 감소 한다. 이렇게 시간에 따라 감소하는 비율은 감소 함수 F(x)로 계산되는데, x는 참조의 최근성을 의미하며 참조가 발생한 시간과 현재와의 시간 간격이다. 예를 들어 블록 b가 3, 5 그리고 8이라는 시간에 참조되고 현재 시간이 10이라면 블록 b의 가치($V_{lc}(b)$)는 다음과 같다.

$$V_{lc}(b) = F(10-3) + F(10-5) + F(10-8)$$
$$= F(7) + F(5) + F(2).$$

과거의 참조보다 최근의 참조에 가중치를 두기 위하여 F(x)는 x가 클수록 감소하는 함수이다. 시간이 흘러감에 따라 가치를 감소시키는 감소 함수 F(x)는 LFR기법의 교체 형태를 결정한다.

LFR기법은 LRFU기법과 유사하게 LFU와 NUR 조건을 모두 만족하는 함수인 $F(x) = (\frac{1}{2})^{\lambda x}$를 감소함수로 사용한다. 이감소 함수에서 λ는 제어 변수로 0부터 1 사이의 값을 가진다. 먼저 λ가 0인 경우 F(x)=1이 되어 LFR기법은 LFU기법과 동일하게 된다 λ가 1이면 $F(x) = (\frac{1}{2})^{x}$이며, 이 경우 LFN기법은 NUR기법과 동일하다 그림 3.1은 λ의 변화에 따라 $F(x) = (\frac{1}{2})^{\lambda x}$ 함수를 보여준다.

그림 3.1에서 Spectrum(Recency/Fruquency)로 표현된 부분에서 LFR기법은 기존의 NUR 및 LFU 기법과 다르게 되며 LFN기법의 고유영역이다[8].

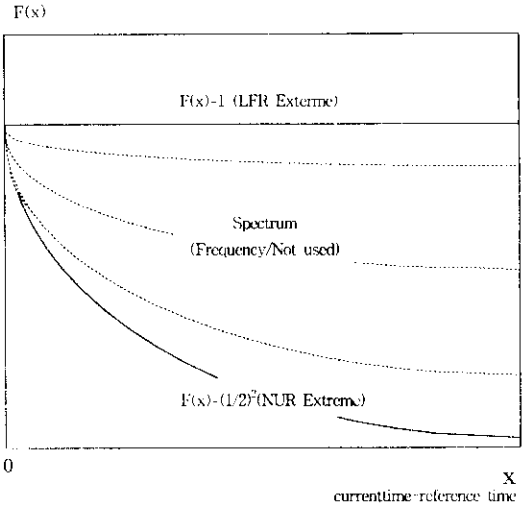

그림 3.1 가중치 함수 $F(x) = (\frac{1}{2})^{\lambda x}$ 일 때 LFR기법의 스펙트럼

4. 컴퓨터 시뮬레이션

제안한 LFR 기법의 시뮬레이션은 Awesim tool을 이용하여 수행하였다.

그림 4.1은 LFR기법과 LRU기법의 Miss rate를 최상의 성능을 보이는 λ와 연관참조 기간 c를 가지고 측정한 것이다. 그림 4.1을 보면 LRU기법은 버퍼 캐쉬의 크기가 커져도 미스율이 거의 낮아지지 않지만 LFR기법은 버퍼 캐쉬의 크기가 커질수록 미스율이 계속 낮아짐을 알 수 있다. 이것은 LFR기법은 참조 시간뿐만 아니라 참조 빈도수도 고려하기 때문에 LRU기법보다 나은 성능을 보이기 때문이다. 그러므로, LFR기법은 기존의 구현된 LRU기법에 비해 더 효율적이라는 것을 알 수 있다.

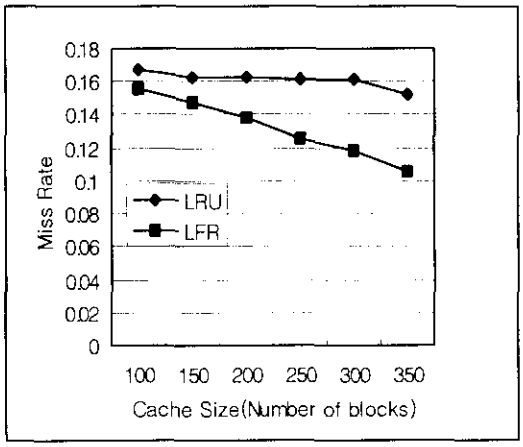

그림 4.1 LFR기법과 LRU기법의 Miss rate비교

5. 결 론

본 논문에서는 새로운 블록 교체기법으로 LFR (Least Frequently Used & Not Used Recently)기법을 제안하였다. LFR기법은 LFR기법과 NUR기법을 모두 포함하여 가중치 함수 $F(x)=(\frac{1}{2})^{\lambda x}$를 사용할 때 LFR기법과 NUR기법 사이에 스펙트럼을 이루며 존재한다. 여기에서 λ는 각 참조의 중요 정도를 결정하는 LFR기법의 제어인자이다. 블록교체 결정에 있어서 한정된 참조 정보만을 고려하던 기존의 기법들과는 달리 LFR기법은 각 블록에 대한 모든 참조 정보를 이용함으로써 블록교체시 향상된 성능을 얻을 수 있었다.

6. 참고문헌

[1] 조유근, 고건, 운영체제론, 홍릉과학출판사.

[2] R. Karedla, J. S. Love, and B. G. Wherry, "Caching Strategies to Improve Disk System Performance," IEEE Computer, vol. 27, No. 3, pp. 38-46, March 1994.

[3] M. J. Bach, The Design of the Unix Operating System. Prentice-Hall, Englewood Cliffs, NJ, 1986.

[4] P. Cao, E. W. Felten, and K. Li, "Application - Controlled File Caching Policies,"in Proceedings of the Summer 1994 USENIX Conference, pp. 171-182, 1994.

[5] J. T. Robinson and N. V. Devarakonda, "Data Cache Management Using Frequency-Based Replacement", in Proceedings of the 1990 ACM SIGMETRICS Conference, pp.134-142, 1990.

[6] E. J. O'Neil, P. E. O'Neil, and Weikum, "The LRU-K Page Replacement Algorithm For Database Disk Buffering", in Proceedings of the 1993 ACM SIGMOD Conference, pp. 297-306, 1993.

[7] T. Johnson and D. Shasha, "2Q: A Low Overhead High Performance Buffer Management Replacement Algorithm", In Proceedings of the 20th International Conference on Very Large Data Bases, pp. 439-450, 1994.

[8] 최홍기 "LRFU 디스크 블록 교체 기법의 구현 및 성능 측정" 서울대학교, 1997.