

하이브리드 메인 메모리의 성능 향상을 위한 페이지 교체 기법

(Page Replacement Algorithm for Improving Performance of Hybrid Main Memory)

이 민 호 ^{*} 강 동 현 ^{*} 김 정 훈 ^{*} 엄 영 익 ^{**}
(Minhoe Lee) (Dong Hyun Kang) (Junghoon Kim) (Young Ik Eom)

요 약 DRAM은 빠른 쓰기/읽기 속도와 무한한 쓰기 횟수로 인해 컴퓨터 시스템에서 주로 메인 메모리로 사용되지만 저장된 데이터를 유지하기 위해 지속적인 전원공급이 필요하다. 반면, PCM은 비휘발성 메모리로 전원공급 없이 저장된 데이터를 유지할 수 있으며 DRAM과 같이 바이트 단위의 접근과 덮어쓰기가 가능하다는 점에서 DRAM을 대체할 수 있는 메모리로 주목받고 있다. 하지만 PCM은 느린 쓰기/읽기 속도와 제한된 쓰기 횟수로 인해 메인 메모리로 사용되기 어렵다. 이런 이유로 DRAM과 PCM의 장점을 모두 활용하기 위한 하이브리드 메인 메모리가 제안되었고 이에 대한 연구가 활발하다. 본 논문에서는 DRAM과 PCM으로 구성된 하이브리드 메인 메모리를 위한 새로운 페이지 교체 기법을 제안한다. PCM의 단점을 보완하기 위해 제안 기법은 PCM 쓰기 횟수를 줄이는 것을 목표로 하며 실험결과에서 알 수 있듯이 본 논문의 제안 기법은 다른 페이지 교체 기법에 비해 PCM 쓰기 횟수를 80.5% 줄인다.

키워드: PCM, 하이브리드 메인 메모리, 페이지 교체 기법, 페이지 캐시

Abstract In modern computer systems, DRAM is commonly used as main memory due to its low read/write latency and high endurance. However, DRAM is volatile memory that requires periodic power supply (i.e., memory refresh) to sustain the data stored in it. On the other hand, PCM is a promising candidate for replacement of DRAM because it is non-volatile memory, which could sustain the stored data without memory refresh. PCM is also available for byte-addressable access and in-place update. However, PCM is unsuitable for using main memory of a computer system because it has two limitations: high read/write latency and low endurance. To take the advantage of both DRAM and PCM, a hybrid main memory, which consists of DRAM and PCM, has been suggested and actively studied. In this paper, we propose a novel page replacement algorithm for hybrid main memory. To cope with the weaknesses of PCM, our scheme focuses on reducing the number of PCM writes in the hybrid main memory. Experimental results shows that our proposed page replacement algorithm reduces the number of PCM writes by up to 80.5% compared with the other page replacement algorithms.

Keywords: PCM, hybrid main memory, page replacement algorithm, page cache

-
- 본 연구는 미래창조과학부 및 정보통신산업진흥원의 대학 IT 연구센터 육성 지원 사업의 연구결과로 수행되었음(NIPA-2014(H0301-14-1020))
 - 이 논문은 2014 한국컴퓨터종합학술대회에서 'PCM과 DRAM으로 구성된 하이브리드 메인 메모리 시스템을 위한 페이지 교체 기법'의 제목으로 발표된 논문을 확장한 것임

^{*} 학생회원 : 성균관대학교 정보통신대학
minhozx@skku.edu
kkangsu@skku.edu
myhuni20@skku.edu

^{**} 종신회원 : 성균관대학교 정보통신대학 교수(Sungkyunkwan Univ.)
yieom@skku.edu
(Corresponding author임)

논문접수 : 2014년 9월 5일
(Received 5 September 2014)
논문수정 : 2014년 10월 16일
(Revised 16 October 2014)
심사완료 : 2014년 10월 28일
(Accepted 28 October 2014)

Copyright©2015 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.
정보과학회 컴퓨팅의 실제 논문지 제21권 제1호(2015. 1)

1. 서론

컴퓨터 시스템에서 메인 메모리로 사용되는 DRAM은 휘발성 메모리로 저장된 데이터를 유지하기 위해 지속적인 전력공급이 필요하다. 이 때문에 DRAM의 전력 소비량은 컴퓨터 시스템 전체 전력 소비량의 상당 부분을 차지한다. 최근 어플리케이션의 메모리 워킹 셋(Memory Working Set)이 증가하면서 메인 메모리의 용량이 증가하고 있는 추세이다. 하지만, DRAM의 메모리 집적도가 한계를 맞이하면서 DRAM의 크기가 증가될 수밖에 없는 상황이며 이는 DRAM의 전력 소비량이 증가되는 요인이 되고 있다. 이런 문제는 특히 배터리 기반의 디바이스나 서버 환경에서 심각한 문제를 야기할 수 있으므로 이를 해결하기 위한 수단이 필요하다.

반면, PCM은 비휘발성 메모리 중 하나로 DRAM과 달리 지속적인 전력공급 없이 저장된 데이터를 유지할 수 있다. 또한, DRAM과 같이 바이트 단위의 쓰기와 덮어쓰기가 가능하다는 장점이 있다. 이런 이유로 PCM은 차세대 메인 메모리로 주목받고 있고 이를 활용하여 컴퓨터 시스템의 성능을 개선하기 위한 연구들이 활발히 진행 중이다[1,2]. 하지만, PCM의 느린 쓰기 속도와 제한된 쓰기 횟수로 인해 당장 메인 메모리로서의 전면적인 교체는 힘든 상황이다. 따라서 PCM과 DRAM의 장점을 모두 활용하기 위해 DRAM과 PCM으로 구성된 하이브리드 메인 메모리가 등장하였다 [3, 4, 5]. 하이브리드 메인 메모리는 그림 1과 같이 두 가지 구조로 나눌 수 있다. 그림 1(a)는 DRAM을 캐시처럼 사용하고 PCM을 메인 메모리처럼 사용하는 구조이며 그림 1(b)는 DRAM과 PCM을 메인 메모리로서 동일하게 사용하는 구조이다. 그림 1(a)의 경우, 쓰기 요청이 자주 발생하는 페이지를 단순히 DRAM에 둘으로써 PCM의 쓰기 횟수를 줄일 수 있지만 그에 비해 그림 1(b)는 PCM의 쓰기 횟수를 줄이기 위해 DRAM과 PCM의 특성을 모두 고려한 페이지 교체기법이 필요하다.

본 논문에서는 DRAM과 PCM으로 구성된 하이브리드 메인 메모리를 위한 새로운 페이지 교체 기법을 제안한다. 본 논문에서 제안하는 페이지 교체 기법은 하이브리드 메인 메모리 성능에 치명적인 PCM 쓰기 횟수를 줄이기 위해 쓰기 위주의 페이지를 DRAM에 할당하고 읽기 위주의 페이지를 PCM에 두는 것을 목표로 한다. 추가적으로 PCM과 DRAM 간의 효율적인 페이지 이동을 위해 PCM에 있는 페이지에 쓰기 요청이 발생하면 어느 정도 덮어쓰기를 허용한다. 실험결과를 보면 본 논문에서 제안하는 페이지 교체 기법은 다른 페이지 교체 기법과 달리 워크로드 특성에 관계없이 높은 메모리 적중률(Memory Hit Ratio)을 가지며 PCM 쓰

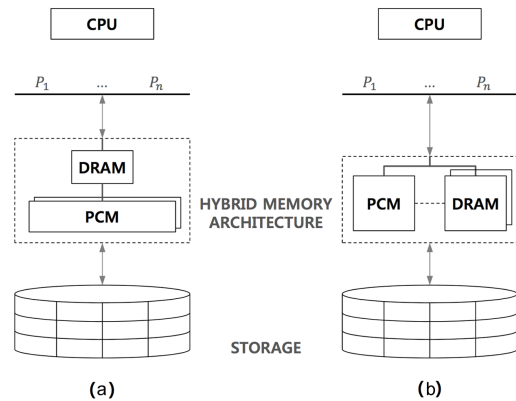


그림 1 하이브리드 메인 메모리 구조

Fig. 1 Hybrid main memory architecture

기 횟수를 다른 페이지 교체 기법에 비해 약 80%까지 줄인 것을 확인할 수 있다.

본 논문의 구성은 다음과 같이 진행된다. 2장에서 PCM의 특성과 하이브리드 메인 메모리와 관련된 연구를 살펴보고 3장에서 제안 기법에 대한 설명을 기술한다. 4장에서는 실험결과와 함께 제안 기법의 효과를 입증하고 5장에 결론으로 마무리한다.

2. 관련 연구

본 장에서는 PCM의 특성을 살펴보고 PCM과 DRAM으로 구성된 하이브리드 메인 메모리를 위한 페이지 교체 기법들을 살펴본다.

2.1 PCM의 특성

표 1[6]에서 보는 것과 같이, PCM은 DRAM과 달리 비휘발성을 가지므로 저장된 데이터를 유지하기 위한 전력량이 DRAM에 비해 약 10배 적다. 하지만, PCM은 다른 쓰기/읽기 속도를 가지며, 특히 PCM의 쓰기 속도는 DRAM의 쓰기 속도보다 약 7배 느리다는 단점이 있다. 또한, PCM은 약 10^7 번의 제한된 쓰기 횟수를 가진다. 이런 이유로 DRAM과 PCM으로 구성된 하이브리드 메인 메모리 시스템에서 PCM 쓰기 횟수를 줄이는 것은 하이브리드 메인 메모리의 성능과 PCM의 수명에 직접적인 영향을 끼친다.

표 1 PCM과 DRAM 특성
Table 1 Characteristics of DRAM and PCM

	PCM	DRAM
Non-volatility	O	X
Write latency	350ns	50ns
Read latency	100ns	50ns
Static power	1.0 W/GB	0.1 W/GB
Endurance	10^7	N/A

2.2 관련 페이지 교체 기법

기존 연구에서는 PCM의 느린 쓰기 속도와 제한된 쓰기 횟수를 고려해 DRAM과 PCM으로 구성된 하이브리드 메인 메모리에서 PCM의 쓰기 횟수를 줄이는 것에 집중하였다. PDRAM[5]의 경우, PRAM과 DRAM으로 구성된 하이브리드 메인 메모리에서 PRAM에 할당된 페이지의 쓰기횟수를 기록하면서 특정 페이지의 쓰기횟수가 한계점을 넘으면 DRAM에 페이지를 이동하는 방식으로 PRAM의 쓰기 횟수를 줄였다. 하지만, 이 기법은 페이지의 특성(쓰기/읽기)을 파악하지 않고 PRAM에 페이지를 할당함으로써 불필요한 PRAM 쓰기를 발생시킨다는 단점이 있다. 4개의 LRU 큐를 이용한 페이지 교체 기법[4]의 경우, 페이지의 특성이 반영된 weighting value를 페이지가 참조될 때마다 계산하여 페이지의 위치를 조정함으로써 PCM 쓰기 횟수를 줄였다. 하지만, 페이지가 참조될 때마다 weighting value를 계산하는 것에 대한 오버헤드가 있다. CLOCK-DWF[3]의 경우, 페이지 폴트가 발생하면 페이지 접근 방식(쓰기/읽기)에 따라 페이지를 DRAM 또는 PCM에 할당하고 PCM에 있는 페이지의 쓰기 요청이 발생하면 해당 페이지를 DRAM으로 이동한 뒤 쓰기 요청을 처리함으로써 PCM의 쓰기횟수를 줄였다. 하지만, 이는 읽기가 많은 워크로드에서는 쓰기 위주의 페이지를 할당하기 위한 DRAM의 공간을 충분히 활용하지 못한다는 단점이 있다. 또한 DRAM과 PCM이 모두 여유 공간이 없을 때, PCM의 쓰기요청을 DRAM에서 처리하기 위한 페이지 이동이 DRAM에서 PCM으로 페이지 이동을 초래하면서 불필요한 페이지 이동과 오버헤드를 발생한다는 단점이 있다.

3. 제안 기법

PCM의 쓰기횟수를 줄이기 위해 본 논문의 제안 기법은 쓰기 요청이 많은 페이지를 DRAM에 읽기 요청이 많은 페이지를 PCM에 두는 것을 목표로 한다. 그리고 PCM과 DRAM 간의 불필요한 페이지 이동을 줄이기 위해 PCM에 있는 페이지에 쓰기 요청이 발생했을 때 덮어쓰기를 어느 정도 허용함으로써 PCM의 쓰기 횟수를 효율적으로 처리한다. 본 기법의 동작은 크게 세 가지로 나눌 수 있다. 1) 페이지 폴트가 발생하면 스토리지에서 읽어온 페이지를 DRAM에 할당한다. 2) DRAM과 PCM 간의 페이지 이동은 제안 기법의 Lazy migration 정책에 의해 결정된다. 3) 불필요한 페이지는 PCM에서만 스토리지로 내보낸다. 본 논문의 제안 기법은 기존의 CLOCK 알고리즘 [6]에서 사용되는 참조/쓰기 비트 외에 Lazy Count (이하 LC)를 각 페이지마다 설정한다. DRAM 또는 PCM에서의 페이지 관리는 기존의 CLOCK 알고리즘과 유사하게 페이지가 참조되면

참조비트를 설정하고 페이지에 쓰기요청이 발생하여 페이지가 업데이트되면 쓰기비트를 설정한다.

3.1 DRAM에서 PCM으로의 페이지 이동

DRAM에 여유 공간이 없을 때, 기존의 CLOCK 알고리즘과 유사한 방식으로 DRAM에서 PCM으로 이동될 페이지를 찾는다. 먼저, CLOCK-hand가 가리킨 페이지의 참조비트를 확인한다. 해당 페이지의 참조비트가 설정되어 있다면 CLOCK-hand는 이를 리셋하고 다음 페이지를 가리키지만 반대의 경우 해당 페이지의 쓰기비트를 확인한다. 만약 쓰기비트가 설정되어 있지 않으면 해당 페이지를 PCM으로 이동시키지만 설정되어 있다면 본 기법은 페이지 이동을 결정하기 위해 마지막으로 해당 페이지의 LC를 확인한다. LC가 Threshold보다 작으면 LC를 증가시킨 뒤, 다음 페이지를 가리키지만 LC가 한계점보다 크다면 이는 충분히 DRAM에 상주할 기회를 준 것으로 간주하고 해당 페이지를 PCM으로 이동시킨다. PCM으로 이동된 페이지는 PCM에서의 Lazy migration을 위해 LC가 초기화된다.

3.2 PCM에서 DRAM으로의 페이지 이동

본 기법에서는 하이브리드 메인 메모리 내에서 불필요한 페이지 이동을 줄이기 위해 PCM에 있는 페이지에 쓰기요청이 발생하면 덮어쓰기를 어느 정도 허용한다. 하지만 특정 페이지에 쓰기요청이 자주 발생할 경우를 대비하여 DRAM에서와 같이 해당 페이지의 LC를 사용하여 PCM에서 DRAM으로의 페이지 이동을 결정한다. PCM에서 DRAM으로의 페이지 이동은 다음과 같은 과정을 통해 이뤄진다. 1) DRAM에 여유 공간이 있는지 확인한다. 만약, DRAM의 여유 공간이 있다면 쓰기 요청이 발생한 페이지를 PCM에서 DRAM으로 이동시킨 후 쓰기 요청을 처리한다. 2) 만약 DRAM의 여유 공간이 없다면 쓰기 요청이 발생한 페이지의 LC를 확인한다. 만약 LC가 0일 경우, 본 기법은 해당 페이지의 덮어쓰기를 한번 허용하여 요청된 크기만큼 쓰기 요청을 처리한다. 하지만 반대로 LC가 1일 경우, 해당 페이지는 다시 쓰기 요청을 받을 가능성이 크다고 간주하여 DRAM으로 해당 페이지를 이동시켜 쓰기 요청을 처리한다.

3.3 PCM에서 스토리지로 페이지 이동

PCM의 여유 공간이 없는 경우, 본 논문에서 제안하는 기법은 CLOCK 알고리즘과 동일하게 CLOCK-hand를 이용해 참조비트가 설정 안 된 페이지를 해당 페이지를 스토리지로 페이지를 이동함으로써 새로운 페이지를 위한 여유 공간을 만든다. 그림 2는 제안 기법의 예를 표현한 것이다. 먼저, 페이지 폴트가 발생했고 DRAM에 여유 공간이 없는 상황을 가정하자. 이때 DRAM에 여유 공간을 만드는 과정은 다음과 같다. 우선 CLOCK-hand가 가리키는 3번 페이지는 참조비트와 쓰기비트가 설정

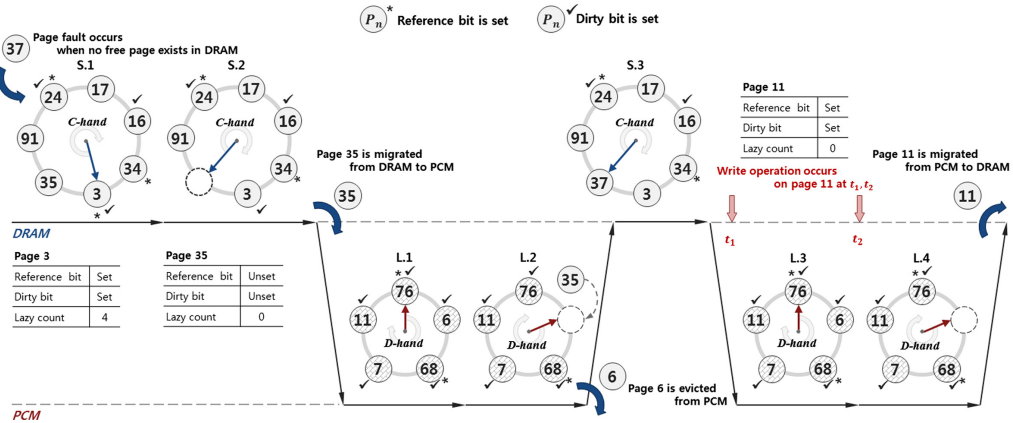


그림 2 제안한 페이지 교체 기법의 예

Fig. 2 The example of proposed page replacement algorithm

되어있기 때문에 LC를 확인한다. 페이지 3번의 경우, LC는 4이며 이는 Threshold보다 작기 때문에 CLOCK-hand는 이를 1 증가시키고 다음 페이지(페이지 35번)를 가리킨다. 반면 35번 페이지의 경우, 참조비트가 설정되어 있지 않으므로 해당 페이지는 DRAM에서 PCM으로 이동된다. PCM의 경우, 일반적인 CLOCK 알고리즘과 같이 참조비트만으로 페이지를 스토리지로 방출함으로써 그림 2와 같이 6번 페이지가 방출될 페이지로 선정된다. 하지만, PCM에 있는 페이지(페이지 11번)에 쓰기요청이 발생할 경우 해당 페이지의 LC를 확인한다.

4. 실험 및 평가

본 논문의 제안 기법을 검증하기 위해 DRAM 기반의 하이브리드 메인 메모리 시뮬레이터를 구현하였으며 이를 이용해 CLOCK 알고리즘[7]과 CLOCK-DWF[3], 제안 기법의 메모리 적중률과 PCM의 쓰기횟수를 비교하였다. 실험에 사용된 메모리 트레이스는 기존 연구에서와 같이 valgrind[8]의 cachegrind를 수정하여 Ubuntu 12.04에서 gedit가 실행되는 동안 참조된 메모리 주소와 페이지 접근 방식(쓰기/읽기)을 추출하였다. 또한, 추출된 메모리 트레이스의 최대 메모리 사용량을 하이브리드 메인 메모리의 용량으로 설정하고 DRAM의 용량은 전체 용량의 1%~95%로 조절하면서 실험하였다. 본 논문의 제안 기법에서 필요한 LC의 Threshold는 DRAM에서는 8로 설정하고 PCM의 경우 LC를 비트처럼 사용하기 때문에 Threshold를 따로 설정하지 않았다. CLOCK의 경우, DRAM과 PCM에 각각 적용하였으며 다른 페이지 교체 기법과 달리 PCM에서 DRAM으로의 페이지 이동이 없다. CLOCK-DWF는 해당 논문에서와 같이 각각의 파라미터에 대한 Threshold를 설정하였다.

4.1 메모리 적중률

그림 3은 각각의 페이지 교체 기법이 적용된 하이브리드 메인 메모리의 메모리 적중률을 나타낸 것이다. 그림 3에서 보는 것처럼 DRAM의 크기가 50%일 때까지는 각각의 페이지 교체 기법의 메모리 적중률은 비슷하다. 하지만, DRAM의 크기가 커질수록 제안 기법을 제외한 나머지 페이지 교체 기법의 메모리 적중률은 감소한다. 이는 다른 페이지 교체 기법과 달리 제안 기법은 페이지 폴트가 발생했을 때 DRAM에 먼저 페이지를 할당하고 DRAM에 여유 공간이 없을 때 DRAM에서 PCM에 페이지를 이동시키기 때문이다. 이렇게 함으로써 본 논문의 제안 기법은 하이브리드 메인 메모리의 공간을 충분히 사용할 수 있다. 그림 3을 보면 본 논문의 제안 기법이 다른 페이지 교체 기법과 달리 워크로드의 특성에 관계없이 하이브리드 메인 메모리의 용량을 충분히 사용하여 CLOCK에 비해 약 4.3%, CLOCK-DWF에 비해 약

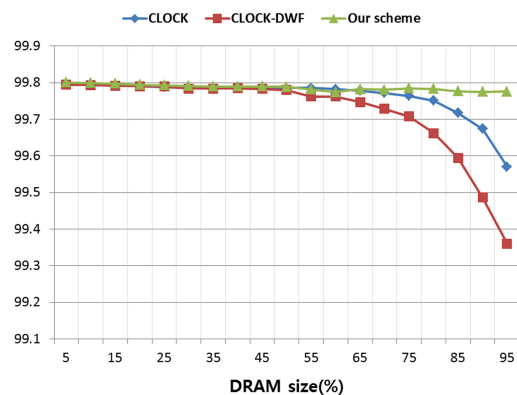


그림 3 메모리 적중률

Fig. 3 Memory hit ratio

2.4% 높은 메모리 적중률을 가진다는 것을 알 수 있다.

4.2 PCM 쓰기 횟수

그림 4는 각각의 페이지 교체 기법이 적용된 하이브리드 메인 메모리의 PCM 쓰기횟수를 비교한 것이다. CLOCK의 경우, PCM에 쓰기 횟수를 줄이기 위한 동작이 없기 때문에 위의 그래프처럼 가장 많은 PCM 쓰기 횟수를 가진다. CLOCK-DWF는 DRAM의 크기가 약 60%일 때까지는 본 논문에서 제안하는 페이지 교체 기법과 비슷한 PCM 쓰기 횟수를 가지지만, DRAM의 크기가 60% 이상일 때는 제안 기법에 비해 더 많은 PCM 쓰기횟수를 가진다. 이는 페이지 폴트가 발생할 때, CLOCK-DWF는 페이지 접근 타입이 쓰기일 때 페이지를 DRAM에 할당하고 읽기일 때 PCM에 페이지를 할당하기 때문에 읽기 요청이 많은 워크로드에서는 PCM의 공간만 활용하기 때문이다. 또한, CLOCK-DWF는 PCM의 쓰기 횟수를 줄이기 위해 PCM에 있는 페이지에 쓰기 요청이 발생하면 해당 페이지를 DRAM에 옮겨 쓰기 요청을 처리하는 방식을 사용한다. 하지만 이는 DRAM과 PCM 모두 공간이 없을 때, PCM에서 DRAM으로의 페이지 이동으로 인해 반대 방향으로의 페이지 이동이 발생시킨다는 점에서 본 논문의 제안 기법과 달리 매번 페이지 크기 (4KB)만큼 PCM을 쓰기 한다는 단점이 있다. 그림 4에서와 같이 본 논문의 제안 기법은 CLOCK에 비해 약 72.3%, CLOCK-DWF에 비해 약 80.5%까지 PCM의 쓰기횟수를 줄인 것을 확인할 수 있다.

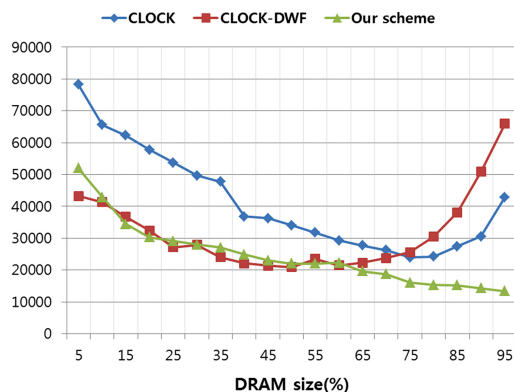


그림 4 PCM 쓰기 횟수

Fig. 4 The number of PCM writes

5. 결론

본 논문에서는 PCM의 단점인 느린 쓰기 속도와 제한된 쓰기횟수를 보완하기 위해 DRAM과 PCM으로 구성된 하이브리드 메인 메모리를 위한 새로운 페이지 교체 기법을 제안한다. PCM의 쓰기 속도는 DRAM의 약

7배이며 DRAM과 달리 PCM의 쓰기 횟수는 제한되어 있다는 점을 고려하여 본 논문에서 제안하는 페이지 교체 기법은 PCM의 쓰기 횟수를 줄이는 것을 목표로 하였다. 따라서 본 논문의 제안 기법은 쓰기 요청이 많은 페이지를 DRAM에 상주시키고 읽기 요청이 많은 페이지를 PCM에 상주시키는 것에 집중한다. 또한 PCM에 있는 페이지에 쓰기 요청이 발생할 경우, DRAM의 여유 공간 유무와 본 기법에서 설정한 LC에 따라 효율적으로 페이지를 PCM에서 DRAM으로 이동시켜 쓰기 요청을 처리함으로써 기존의 다른 알고리즘에 비해 PCM의 쓰기횟수를 효과적으로 줄인 것을 확인할 수 있다. 본 논문에서 제안하는 기법은 LC라는 추가적인 변수를 기존의 CLOCK 알고리즘에 추가하여 하이브리드 메인 메모리를 위한 페이지 교체 기법 알고리즘을 설계하였다. 하지만 추가적인 변수를 설정함으로써 각 워크로드에 적절한 한계값을 설정해야한다. 따라서 추가적인 변수 없이 하이브리드 메인 메모리의 페이지를 관리할 수 있는 페이지 교체 기법에 대한 추가적인 연구가 필요하다.

References

- [1] D. Kim and H. Bahn, "Improving Smartphone Storage Performance through SQLite Journaling with Non-volatile RAM," *Journal of KISS: Computing Practices and Letters*, Vol. 19, No. 5, pp. 283-287, May 2013. (in Korean)
- [2] H. Kim, J. Choi, D. Lee, and S. H. Noh, "Toward Efficient Management of Highly Durable NVM+DRAM Hybrid Main Memory Systems," *Journal of KISS: Computer Systems and Theory*, Vol. 40, No. 2, pp. 61-67, Apr. 2013. (in Korean)
- [3] S. Lee, H. Bahn, and S. H. Noh, "CLOCK-DWF: A Write-History-Aware Page Replacement Algorithm for Hybrid PCM and DRAM Memory Architectures," *IEEE Transactions on Computers*, Vol. 63, No. 9, pp. 2187-2200, Apr. 2013.
- [4] H. Seok, Y. Park, K. W. Park, and K. H. Park, "Efficient Page Caching Algorithm with Prediction and Migration for a Hybrid Main Memory," *ACM SIGAPP Applied Computing Review*, Vol. 11, No. 4, pp. 38-48, Dec. 2011.
- [5] G. Dhiman, R. Ayoub, and T. Rosing, "PDRAM: A Hybrid PRAM and DRAM Main Memory System," *Proc. Design Automation Conference*, pp. 664-669, Jul. 2009.
- [6] E. Sean, M. Leinwander, and G. Crisenza, "Phase Change Memory: A New Memory Enables New Memory Usage Models," *Proc. International Memory Workshop*, pp. 1-2, May 2009.
- [7] A. S. Tanenbaum, A. S. Woodhull, *Operating Systems: Design and Implementation*, Prentice-Hall Englewood Cliffs, 1987.

- [8] N. Nethercote and J. Seward, "Valgrind: A Program Supervision Framework," *Elsevier Electronic Notes in Theoretical Computer Science*, Vol. 89, No. 2, pp. 44-66, Oct. 2003.



이 민 호

2013년 성균관대학교 전자전기공학부 학사. 2013년~현재 성균관대학교 전자전기컴퓨터공학과 석사과정. 관심분야는 운영체제, 파일시스템, 가상화 기술



강 동 현

2007년 한국산업기술대학교 컴퓨터공학과 학사. 2010년 성균관대학교 전자전기컴퓨터공학과 석사. 2013년~현재 성균관대학교 전자전기컴퓨터공학과 박사과정. 관심분야는 스토리지 시스템, 운영체제



김 정 훈

2010년 성균관대학교 컴퓨터공학과 학사
2012년 성균관대학교 휴대폰학과 석사
2012년~현재 성균관대학교 IT융합학과 박사과정. 관심분야는 시스템 소프트웨어, 운영체제, 가상화 기술, 파일 시스템



엄 영 익

1983년 서울대학교 계산통계학과 학사. 1985년 서울대학교 전산학과 석사. 1991년 서울대학교 전산학과 박사. 1993년~현재 성균관대학교 정보통신대학 교수. 관심분야는 시스템 소프트웨어, 운영체제, 가상화 기술, 파일 시스템