

Recency and Frequency based Page Management on Hybrid Main Memory

Sungho Kim*, Jong Wook Kwak**

Abstract

In this paper, we propose a new page replacement policy using recency and frequency on hybrid main memory. The proposal has two features. First, when a page fault occurs in the main memory, the proposal allocates it to DRAM, regardless of operation types such as read or write. The page allocated by the page fault is likely to be high probability of re-reference in the near future. Our allocation can reduce the frequency of write operations in PCM. Second, if the write operations are frequently performed on pages of PCM, the pages are migrated from PCM to DRAM. Otherwise, the pages are maintained in PCM, to reduce the number of unnecessary page migrations from PCM. In our experiments, the proposal reduced the number of page migrations from PCM about 32.12% on average and reduced the number of write operations in PCM about 44.64% on average, compared to CLOCK-DWF. Moreover, the proposal reduced the energy consumption about 15.61%, and 3.04%, compared to other page replacement policies.

▶ Keyword: Non-Volatile Memory, Page Replacement Policy, CLOCK, Hybrid Main Memory, Recency, Frequency

I. Introduction

DRAM은 빠른 연산 속도와 높은 집적도를 가지기 때문에 많은 컴퓨터 시스템에서 메인 메모리로 채택되어 왔다. 최근 들어 어플리케이션에서 요구하는 작업 세트(working set)가 증가하면서 그에 따라 메인 메모리의 요구량도 증가했다. 그러나 DRAM은 집적도로 얻을 수 있는 용량 증가의 한계에 도달하였고 용량을 증가시키기 위해서는 이제 DRAM의 물리적 크기를 직접 증가시켜야 하는 문제점이 발생하였다. 게다가 DRAM은 데이터를 유지하기 위해 주기적인 에너지가 소모된다. 이는 서버 시스템의 경우 전체 에너지 소모의 30~40%를 차지할 정도로 크다. 결과적으로 DRAM의 용량이 증가할수록 전체 컴퓨터 시스템의 에너지 소모가 증가하게 된다[1].

최근 들어 PCM(Phase Change Memory)이 DRAM을 대체할 새로운 메인 메모리로 각광받고 있다[2]. PCM은 비휘발성 메모리(Non-Volatile Memory)의 특성을 가지고 있어, 기존의 DRAM과 달리 주기적인 재공급 연산(refresh operation)을 수행할 필요가

없다. 게다가 PCM은 DRAM과 비교하여 집적도는 최대 4배 높으며, 읽기 연산의 속도가 DRAM과 유사하다는 장점이 있다. PCM의 이러한 장점들을 이용하여 PCM을 DRAM으로 대체하기 위한 연구들이 최근 들어 활발히 진행되고 있다[1,3-5].

그러나 PCM이 DRAM을 대체하기 위해서는 해결되어야 하는 몇 가지 문제점이 존재한다. PCM은 DRAM과 비교하여 쓰기 지연시간이 길 뿐만 아니라 연산에 대한 에너지 소모가 크다. 이는 쓰기 연산을 주로 수행하는 컴퓨터 시스템에서 전반적인 성능 저하의 원인이 된다. 또한 PCM은 상대적으로 짧은 수명을 가지고 있다. 이는 PCM을 메인 메모리로 채택하는데 치명적인 문제점이라 할 수 있으며, 많은 연구에서 마모도 평준화(wear leveling) 기법을 통해 이를 해결하고자 노력했다[3,5-11].

이러한 노력에도 불구하고, PCM을 기존의 메인 메모리로 대체하여 사용하기에는 여전히 수명이 짧기 때문에 DRAM을 완전히 대체하기 어렵다.

• First Author: Sungho Kim, Corresponding Author: Jong Wook Kwak

*Sungho Kim (bocal23@ynu.ac.kr), Dept. of Computer Engineering, Yeungnam University

**Jong Wook Kwak (kwak@yu.ac.kr), Dept. of Computer Engineering, Yeungnam University

• Received: 2017. 11. 30, Revised: 2017. 12. 10, Accepted: 2018. 01. 10.

• This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2017R1D1A1A09000654)

따라서 이러한 문제점들을 해결하기 위해 기존의 연구자들은 DRAM과 PCM을 결합한 하이브리드 메인 메모리를 제안했다. 하이브리드 메인 메모리는 쓰기가 빈번한 페이지들은 DRAM에서 수행할 수 있도록 유도하고, 그와 반대로 쓰기 빈도가 낮은 페이지들은 PCM으로 유도한다. 이를 통해 하이브리드 메인 메모리는 빠른 쓰기 연산 속도, 낮은 에너지 소모, 높은 집적도와 같은 DRAM과 PCM의 모든 장점을 포함한 구조가 된다[12-13].

그러나 하이브리드 메인 메모리는 각기 다른 특성을 가지는 메모리를 사용하기 때문에, 기존의 DRAM 기반의 페이지 교체 정책을 그대로 활용하기에는 부적합하다[14-17]. 이러한 문제점을 해결하기 위해 CLOCK-DWF는 하이브리드 메인 메모리의 장점을 활용한 페이지 교체 정책을 제안했다[18].

CLOCK-DWF는 각기 다른 두 메모리의 특성을 고려하여 최초의 쓰기 연산에 의해 발생한 페이지 부재는 DRAM으로, 그와 반대로 읽기 페이지 부재는 PCM으로 상주시킨다. 이러한 과정을 통해 CLOCK-DWF는 DRAM과 PCM에 쓰기가 빈번한 페이지와 읽기가 빈번한 페이지를 구분 예측했다. 이러한 예측의 경우 몇 가지 예외 상황이 존재하는데, 이는 DRAM에서 빈 공간이 부족할 경우와 PCM에서 쓰기 연산이 발생할 경우에 해당한다. 전자의 경우 본 논문에서는 페이지를 구분하기 위해 쓰기 횟수를 기준으로 판단했다. 후자는 쓰기 연산이 발생한 경우 PCM에서 DRAM으로 즉시 이주를 수행했다. 그러나 CLOCK-DWF는 DRAM에 페이지들을 유지하는 방법으로 쓰기 횟수만을 판단 기준으로 사용한다. 또한 CLOCK-DWF의 즉시 이주 정책은 어떠한 고려 없이 PCM에서 DRAM으로 페이지를 이주한다. 이러한 CLOCK-DWF의 정책들은 DRAM과 PCM에 페이지들을 유지하는 기준으로 부정확성을 가지고 있다.

따라서 본 논문에서는 CLOCK-DWF의 문제점을 해결하는 새로운 페이지 교체 정책을 제안한다. 제안하는 기법은 DRAM에 유지할 페이지들의 정확성을 보장하기 위해 일정한 주기 동안 각 페이지의 쓰기 횟수를 측정한다. 이를 통해 제안하는 기법은 DRAM의 페이지들을 쓰기 집중적인 페이지 혹은 쫓겨날 후보자 페이지로 각각 구분한다. 또한 PCM에서의 불필요한 페이지 이주를 방지하기 위해, 제안하는 기법은 PCM에서 쓰기 연산이 발생한 경우, 해당하는 페이지가 드물게 쓰기 연산이 발생하는지 판단하고, 이러한 판단 기준에 부합할 경우 PCM으로의 쓰기 연산을 허용하여 DRAM으로의 불필요한 페이지 이주를 방지한다.

이하 본 논문의 구성은 다음과 같다. 2장에서는 본 논문의 배경 지식과 주요 관련 연구를 소개하고 CLOCK-DWF의 문제점을 제시한다. 3장은 제안하는 페이지 교체 정책을 소개하며 4장은 CLOCK, CLOCK-DWF, 제안하는 페이지 교체 정책의 성능 평가를 진행한다. 마지막으로 5장에서는 결론을 맺는다.

II. Background and Related Works

PCM은 칼코젠 유리(chalcogenide glass)에 상변화(phase

change) 물질을 활용하여 데이터를 저장하는 비휘발성 메모리이다. 이 기술은 1960년대에 입증되었지만 데이터를 저장하는데 많은 에너지가 소모되며, 빠르게 데이터를 저장할 수 있는 물질을 찾지 못하여 사용되지 않았다. 최근 PCM은 낮은 에너지 소모와 빠른 데이터 저장 속도를 가지는 GST(Ge₂Sb₂Te₅) 혹은 AIST(Ag- and In-doped Sb₂Te) 물질을 이용하여 산업계와 학계에서 활발히 연구가 진행되고 있다[19-20].

PCM은 하나의 셀이 가지는 상태에 따라 결정질(crystallization)과 비결정질(amorphous)로 구분하며, 이는 비트 정보를 저장하기 위한 최소 단위에 해당한다. 이러한 상태를 결정하기 위해 PCM은 히터(heater)를 통해 일정한 전압을 지속적으로 주입하여 결정질 상태가 되며, 반대로 높은 전압을 순간적으로 주입하여 상태를 결정하는 경우 비결정질 상태가 된다. 이러한 PCM의 결정질과 비결정질 상태는 하나의 비트를 기준으로 각각 SET(비트 1)과 RESET(비트 0)으로 상호 구분한다[12-13,20]. 그림 1은 PCM 셀의 구조와 셀의 상태를 결정하는 흐름도이다[19].

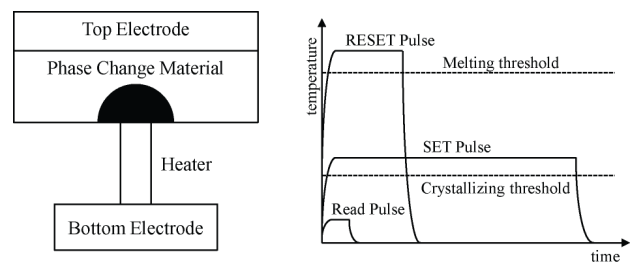
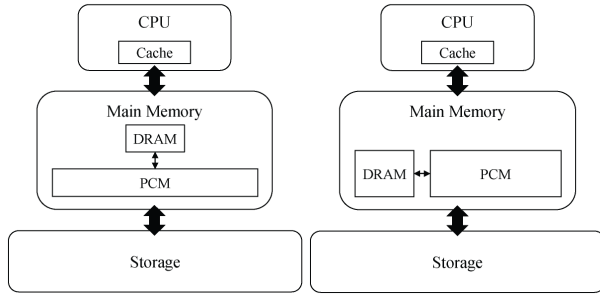


Fig. 1. Structure and two states of PCM cell

PCM은 DRAM과 달리 비휘발성 메모리 특성으로 대기 전력이 매우 낮으며 비교적 DRAM과 유사한 읽기 연산 속도를 가지고 있다. 또한 PCM은 기존의 메인 메모리 구조를 그대로 활용할 수 있기 때문에 DRAM을 대체하기 위한 연구가 활발히 진행 중이다[1,3-5]. 그러나 기존의 연구들은 DRAM을 PCM으로 완전히 대체하는데 초점을 두었으며, 이는 두 가지 문제점을 초래했다. 첫 번째, PCM은 DRAM과 비교하여 상대적으로 쓰기 지연시간이 길고, 연산을 수행하는데 에너지 소모가 크다. 이는 쓰기 연산이 집중적으로 발생하는 서버 시스템과 같은 경우, 사용자 응답 시간에 치명적인 영향을 미칠 수 있다. 두 번째, PCM은 제한적인 수명을 가지기 때문에, 지역성(locality)과 같은 컴퓨터 시스템의 특성을 고려해야 한다. 이러한 고려가 없을 경우, PCM은 이상적인 수명에 도달할 수 없게 된다.

이와 같은 문제를 해결하기 위해 기존의 연구들은 비트, 바이트, 페이지 단위로 변경되지 않는 데이터에 대해서는 쓰기 연산을 수행하지 않음으로써 수명과 쓰기 속도를 향상시켰다[21-23]. 또한 PCM의 이상적인 수명에 도달하기 위해 마모도 평준화 기법을 활용하여, 쓰기 연산이 빈번한 페이지와 빈번하지 않은 페이지를 서로 이주하여 쓰기 횟수를 고르게 분포시키는 방법으로 수명을 연장했다[3,5-11].

최근에는 DRAM과 PCM의 장점을 결합한 하이브리드 메인 메모리에 대한 연구가 활발히 진행 중이다. 하이브리드 메인 메모리는 쓰기 연산을 DRAM에서 수행하여 PCM의 수명을 연장했고, 읽기 연산을 PCM으로 유도하여 시스템의 에너지 소모 감소, PCM의 수명 향상, 응답 시간 향상 등의 다양한 장점을 추구했다. 이러한 하이브리드 메인 메모리는 DRAM을 구성하는 방법에 따라 계층 구조와 병렬 구조로 구분한다.



(a). Hierarchical architecture (b). Parallel architecture

Fig. 2. Hybrid DRAM and PCM main memory

그림 2는 하이브리드 메인 메모리의 두 가지 구조이다. 계층 구조는 캐시와 마찬가지로 DRAM을 PCM 상위 계층에 두어 PCM의 근본적인 쓰기 횟수를 줄이기 위한 구조이다. 이는 DRAM의 계층 구조 설계와 관리를 위한 추가적인 하드웨어 비용을 초래한다. 병렬 구조는 기존의 DRAM 구조와 동일하게 하나의 페이지 매핑 테이블에 의해서 주소 접근이 가능하다. 이는 계층 구조와 달리 추가적인 하드웨어 비용을 초래하지 않는다. 그러나 병렬 구조는 기존의 DRAM 페이지 교체 정책을 활용하기에는 부적합하다. 이는 하이브리드 메인 메모리에 적합한 별도의 페이지 교체 정책이 필요함을 의미한다.

현재까지 메인 메모리의 페이지 교체 정책은 DRAM을 중심으로 활발히 연구가 진행되어 왔다. LRU(Least Recently Used)는 대표적인 페이지 교체 정책이다. LRU는 가장 최근에 참조한 페이지에 가장 높은 순위를 부여하며 가장 참조되지 않은 페이지를 희생 페이지로 선정한다. LRU는 하드웨어적으로 구현이 복잡하고 비용이 많이 들기 때문에 고성능을 요구하는 캐시에서 주로 활용된다.

CLOCK은 참조 비트를 활용하여 페이지 접근이 발생하였을 때 접근 유무를 판단하며, 페이지 부재에 의해서 페이지 교체가 요구될 때 해당하는 페이지의 참조 비트에 의해서 희생 페이지를 추정하고 선정한다[24]. 이러한 CLOCK은 하나의 참조 비트만을 사용하여 희생 페이지를 선정하기 때문에 하드웨어 비용이 적고 구성이 비교적 간단하다.

CLOCK-DWF는 하이브리드 메인 메모리에서 페이지 교체 정책을 수행하기 위해 기존 CLOCK 정책의 장점을 활용하고, 이를 추가 개선했다. CLOCK-DWF는 쓰기 연산에 의해서 페이지 부재가 발생하면 DRAM으로, 그 반대인 경우는 PCM으로 상주시켜 페이지를 구분 할당했다. 또한 CLOCK-DWF는 PCM

에 쓰기 연산이 발생하였을 때 PCM에 상주시키는 페이지를 DRAM으로 이주시켜 PCM의 쓰기 연산을 줄이고자 노력했다. 이러한 과정에서 CLOCK-DWF는 DRAM에 많은 가용 공간을 요구한다. 이를 해결하기 위해 CLOCK-DWF는 쓰기 연산이 빈번한 페이지들이 가까운 미래에도 더 많은 쓰기 연산을 발생시킬 것으로 간주하고 이를 DRAM 내에 상주시키고, 그렇지 않은 페이지들을 PCM으로 이주하여, 전반적인 쓰기 연산을 줄이고 수명을 연장했다[18].

그러나 CLOCK-DWF는 읽기 연산에 의해서 페이지 부재가 발생하였을 때 이를 PCM에 상주시키고, 다시 해당 페이지에 쓰기 연산이 발생하였을 경우 DRAM으로 재차 이주시킨다. 이는 PCM으로의 불필요한 쓰기 연산을 초래한다. 또한, CLOCK-DWF는 최근에 접근한 페이지들에 대한 판단의 비효율성이 존재한다. 페이지들의 최신성을 관리하기 위해 클록 회전 주기 내부의 쓰기 연산 발생 시점을 보다 더 정교화 하여 처리할 필요성이 있다.

III. The Proposed Page Replacement Policy

이 장에서는 CLOCK-DWF의 문제점을 해결하고 성능을 향상하기 위한 새로운 페이지 교체 정책을 제안한다. 제안하는 페이지 교체 정책은 그림 3과 같이 구성하며, DRAM과 PCM의 특성에 맞게 각각의 페이지 교체 정책을 수행한다. 제안하는 기법을 서술하기에 앞서 그림 3에서 사용한 모든 표기는 본문의 그림 4와 그림 5에도 동일하게 적용한다.

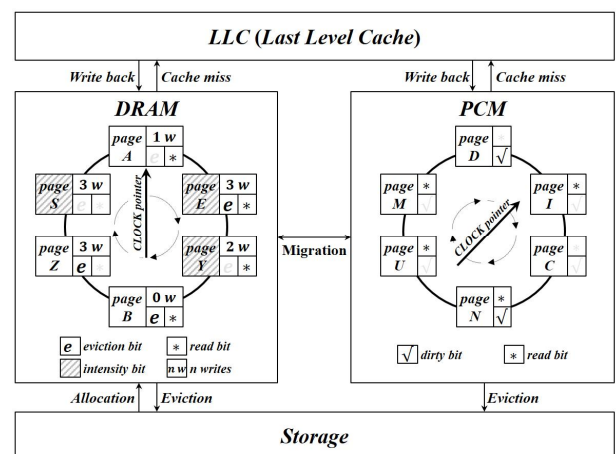


Fig. 3. Architecture of the proposed page replacement policy

제안하는 기법은 CLOCK-DWF와 다르게 페이지 할당 정책을 수행함에 있어서 페이지 부재가 발생할 경우 연산에 관계없이 해당하는 페이지를 DRAM으로 우선 할당한다. 이는 두 가지 측면에서 장점이 있다. 첫째, 페이지 부재가 발생한 페이지

는 연산에 관계없이 한 번 이상의 쓰기 연산이 발생할 가능성이 크다. 따라서 이러한 페이지들은 DRAM으로의 우선 할당을 통해 PCM으로의 쓰기 연산을 줄일 수 있다. 두 번째, 페이지 부재로 인해 메인 메모리로 할당된 페이지는 가까운 미래에 참조 될 확률이 크다. 이는 컴퓨터 시스템의 특성인 참조 지역성(locality of reference)이 존재하기 때문이다. 그러나 이러한 장점을 활용하기 위해서는 DRAM에 할당된 페이지 가운데 쓰기 연산이 빈번한 페이지와 그렇지 않은 페이지를 구분하기 위한 정책이 필요하다.

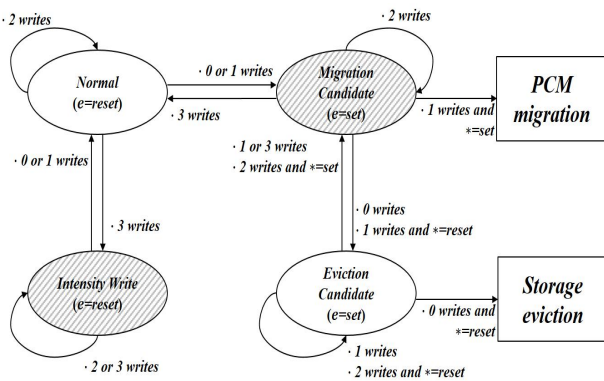


Fig. 4. State transition diagram for classifying pages in DRAM

제안하는 기법은 DRAM의 각 페이지를 쓰기 카운터, 읽기 비트, 강도 비트, 퇴거 비트를 활용하여 페이지들의 쓰기 빈도를 구분한다. 쓰기 카운터는 CLOCK 포인터가 1회전 하는 동안의 쓰기 횟수를 기록하여 각 페이지의 쓰기 빈도를 측정한다. 읽기 비트는 읽기 연산에 의한 접근 유무를 기록한다. 강도 비트는 쓰기 혹은 읽기 연산이 빈번하게 발생하는 페이지를 의미하는 용도로 활용한다. 퇴거 비트는 최근 쓰기 연산이 빈번하게 발생하지 않은 페이지를 의미한다. 이는 PCM 혹은 저장소(storage)로 쫓겨날 후보자 페이지에 해당한다. 이러한 정보들을 기반으로 제안하는 기법은 페이지 부재가 발생하였을 때 DRAM에 상주할 페이지를 그림 4에 의해서 구분한다.

그림 4에서 페이지는 네 가지의 상태 값을 가지며, 이는 강도 비트와 퇴거 비트의 조합으로 이루어진다. 먼저 강도 비트는 빈번한 쓰기 연산이 발생하거나 혹은 읽기 연산이 발생할 것이라 판단되는 페이지를 표기한다. 따라서 강도 비트는 전자의 경우 지속적으로 DRAM에 상주하게 할 페이지이며, 후자의 경우 PCM으로 이주할 후보자 페이지를 결정한다. PCM의 읽기 연산은 DRAM과 거의 유사하게 발생하기 때문에 읽기 연산이 빈번한 페이지를 PCM에서 수행할 경우, 상대적으로 DRAM은 더 많은 쓰기 연산을 수행할 수 있기 때문에 전체 시스템의 효율성을 높일 수 있다. 퇴거 비트는 쓰기 연산뿐만 아니라 읽기 연산도 거의 발생하지 않은 페이지를 구분하기 위해 사용한다. 이를 통해 제안하는 기법은 DRAM과 PCM 둘 다 거의 참조될 확률이 낮은 페이지에 해당할 경우, 이를 저장소로 내보낸다.

제안하는 페이지 교체 정책의 자세한 수행 과정은 다음과 같이 요약된다.

1. 메인 메모리에서 페이지 부재가 발생할 경우, DRAM 내에 빈 공간이 있는지 확인한다.
2. DRAM에 빈 공간이 존재할 경우, 빈 공간에 새로운 페이지를 할당한다. 이때 해당하는 모든 비트 정보는 초기화 하되, 읽기 페이지 부재의 경우 퇴거 비트와 강도 비트를 설정한다. 이후 페이지 교체 정책의 수행 과정을 마친다.
3. 만약 DRAM에 빈 공간이 존재하지 않을 경우, 새로운 공간을 확보하기 위해 쫓겨날 혹은 이주될 페이지를 선정한다. 이때 해당하는 페이지가 이러한 조건에 만족하지 못할 경우, DRAM 상태 전이도에 의해서 다음의 해당하는 비트들로 설정한 뒤 새로운 페이지로 이동한다.
4. 이러한 과정을 반복적으로 수행하여 선정된 페이지는 쫓겨나거나 혹은 이주한다. 이후 새롭게 확보한 빈 공간에 새로운 페이지를 할당하고, 페이지 교체 정책의 수행 과정을 마친다.

이와 같이 제안하는 페이지 교체 정책은 퇴거 비트와 강도 비트를 효과적으로 활용하며, 단순하게 쓰기 혹은 읽기 연산의 횟수만을 고려하여 페이지를 구분하지 않는다는데 주목한다. 또한 이러한 방법을 통해 DRAM에 상주하는 페이지는 클럭 주기 동안에 쓰기 카운터를 기반으로 퇴거 비트와 강도 비트를 지속적으로 갱신함으로써 해당하는 페이지의 최신성을 보장한다. 제안하는 기법은 최근에 접근하는 페이지의 비효율성을 완화하고, 이를 통해 쓰기 연산이 빈번하게 발생하는 페이지들을 DRAM에 상주시킨다. 이러한 과정을 통해 제안하는 페이지 교체 정책은 PCM에 쫓겨난 페이지는 가까운 미래에 읽기 연산으로 인해 참조 확률이 클 것이라고 판단하여 PCM으로 이주한다. 만약 이러한 페이지에서 쓰기 연산이 빈번하게 발생하는 경우, PCM의 수명과 성능에 치명적인 영향을 미칠 수 있다.

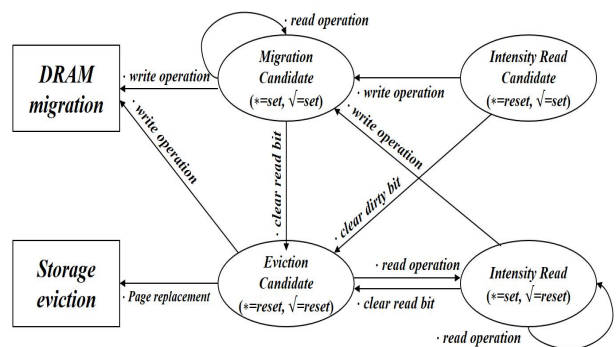


Fig. 5. State transition diagram for the page migrated to DRAM or the page replacement in PCM

이와 같은 문제를 해결하기 위한 그림 5는 PCM에서의 쓰기 빈도(frequency)를 판단하고 이를 DRAM으로 이주하기 위한 정책이다. 제안하는 기법은 쓰기가 빈번하게 발생하는 페이지를 판단하기 위해 PCM의 각 페이지를 더티 비트, 읽기 비트를 활용하여 구분한다. 쓰기 연산이 발생하였을 때 읽기 비트가 설정되어있다면 한 번의 덮어쓰기를 허용한다. 만약 해당하는 페

이지의 더티 비트와 읽기 비트가 둘 다 설정되어있거나 혹은 그 반대의 경우, 이는 DRAM으로 이주하는 조건에 해당한다. 이 때, 두 개의 비트가 설정되어 있는 경우에는 가까운 미래에 참조할 확률이 크다고 판단하고, 반대의 경우는 해당하는 페이지에 마지막 기회를 준다. PCM에서 DRAM으로 페이지를 이주하는 자세한 수행 과정은 다음과 같다.

1. 페이지에 읽기 연산이 발생한 경우, 해당하는 페이지의 읽기 비트를 설정한다.

2. 쓰기 연산이 발생할 경우, 해당하는 페이지의 비트 설정 여부를 확인한다. 만약 두 개의 비트가 모두 설정되어 있거나 설정되어 있지 않은 경우, DRAM으로 이주한다. 이때, 두 개의 비트가 모두 설정되어 있지 않은 페이지는 더티 비트를 설정한다.

이를 통해 제안하는 기법은 PCM에 빈번하게 쓰기 연산을 발생시키는 페이지들을 DRAM으로 이주하고, 거의 드물게 쓰기 연산이 발생하는 페이지들은 PCM 내에 유지한다. 이는 DRAM으로의 빈번한 페이지 이주의 횟수를 줄이게 한다. 마지막으로 PCM에 페이지 교체 정책이 발생하였을 경우, 더티 비트와 읽기 비트 둘 다 설정되지 않은 페이지를 저장소로 쫓아낸다. 이러한 조건에 만족하지 않는 경우, 해당하는 페이지의 모든 비트를 초기화하며, 이러한 과정을 반복적으로 수행하여 조건에 부합하는 페이지를 찾는다.

IV. Performance Evaluation

1. Experimental Setup

이 장에서는 본 논문에서 제안하는 페이지 교체 정책의 성능 평가를 진행하기 위해 gem5 시뮬레이터를 사용하여 모의실험을 수행 한다[25]. gem5 시뮬레이터는 M5 시뮬레이터에서 제공하는 폭넓은 프레임워크 환경과 GEMS의 메모리 시스템 환경 구성의 유연성을 결합한 새로운 시뮬레이터이다[26-27]. 또한 gem5 시뮬레이터는 SE(System-call Emulation) 모드와 FS(Full System) 모드를 제공한다. SE 모드는 운영체제의 기능을 배제한 하드웨어 단위의 모의실험 환경을 제공하며, FS 모드는 하드웨어와 운영체제의 전반에 걸쳐서 모든 기능을 활용 할 수 있다. 본 논문에서는 gem5 시뮬레이터의 FS 모드를 사용하여 성능 평가를 진행했다.

Table 1. Simulation parameters

Description	DRAM	PCM
Capacity	256MB	1GB
Density	1x	2-4x
Read/Write latency	50/50 (ns)	50/350 (ns)
Read/Write energy	0.1/0.1 (nJ/bit)	0.2/1.0 (nJ/bit)
Static power	1(W/GB)	0.1(W/GB)
Endurance	∞	10^7

Table 2. Workload characteristics

Workload	Benchmarks	Memory footprint (MB)	Ratio of operations (Read : Write)
mix1	401.bzip2	309.8	1.3:1
	410.bwaves	1,129	3.03:1
mix2	433.milc	699	2.4:1
	437.leslie3d	200	2.78:1
mix3	450.soplex	178.2	2.34:1
	459.GemsFDTD	1,071	1.64:1
mix4	458.sjeng	278.1	1.07:1
	470.lbm	635	1.34:1
mix5	471.omnetpp	170.3	2.19:1
	473.astar	513	1.72:1

제안하는 페이지 교체 정책의 성능 평가를 진행하기 위해 CLOCK-DWF에서 사용한 매개변수와 동일하게 모의실험 환경을 구축했고, 자세한 매개변수는 표 1과 같다. 하이브리드 메인 메모리의 용량은 일반적으로 DRAM보다 PCM이 크기 때문에 1:4의 비율로 메인 메모리를 구성했다. 본 논문에서는 모의실험의 신뢰성을 보장하기 위해 SPEC CPU2006 벤치마크 프로그램을 사용했다 [28-29]. SPEC CPU2006 벤치마크는 메인 메모리에 접근하는 분포가 매우 다양한 응용 프로그램들을 포함하고 있으며, 그 가운데 메모리로의 접근 분포가 넓은 벤치마크와 국부적으로 접근하는 벤치마크를 선정했다. 각 벤치마크를 기반으로 모의실험을 수행하기 위해서 본 논문에서는 듀얼코어 시스템 환경을 고려했으며, 해당하는 벤치마크의 접근 분포는 표 2와 같다.

2. Experimental Results

하이브리드 메인 메모리의 성능 평가를 진행하기 위해 기존의 DRAM 기반의 페이지 교체 정책인 CLOCK과 하이브리드 메인 메모리 기반의 CLOCK-DWF, 그리고 제안하는 페이지 교체 정책의 성능을 상호 비교했다.

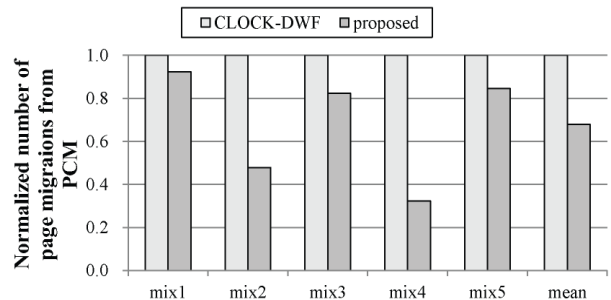


Fig. 6. Normalized number of page migrations from PCM

성능 평가를 진행하기에 앞서 제안하는 페이지 교체 정책과 CLOCK-DWF의 페이지 이주 횟수를 비교했다. 그림 6의 결과는 CLOCK-DWF의 페이지 이주 횟수를 기준으로 정규화 했다. 제안하는 페이지 교체 정책은 CLOCK-DWF와 비교하여 평균적으로 32.12%에 해당하는 PCM에서 DRAM으로의 페이지 이주 횟수가 감소했다. CLOCK-DWF는 읽기 연산으로 인해 메인 메모리상에서 페이지 부재가 발생하였을 때, 해당 페이지를

PCM에 할당 한다. 이는 해당 페이지로의 추가적인 쓰기 연산에 의해서 재차 해당 페이지가 DRAM으로 이주될 확률이 크다는 것을 의미한다. 결과적으로 CLOCK-DWF에서 페이지를 구분 할당하는 정책은 효율성이 낮다는 것을 보여주는 결과이다.

제안하는 페이지 교체 정책의 성능 평가는 두 가지 지표를 통해 비교 분석하였다. 첫 번째 지표는 PCM의 쓰기 횟수이다. 이는 하이브리드 메인 메모리에서 PCM 수명에 직접적인 영향을 끼친다. 하이브리드 메인 메모리의 에너지 소모는 두 번째 지표로 비교했다. 이는 DRAM과 PCM의 장점을 결합하는데 가장 주된 요인 중 하나이다. 메인 메모리에서의 DRAM은 전체 컴퓨터 시스템에서 데이터를 유지하기 위해 상당한 에너지가 소모되기 때문에 에너지 소모를 줄이는 것은 매우 중요하다.

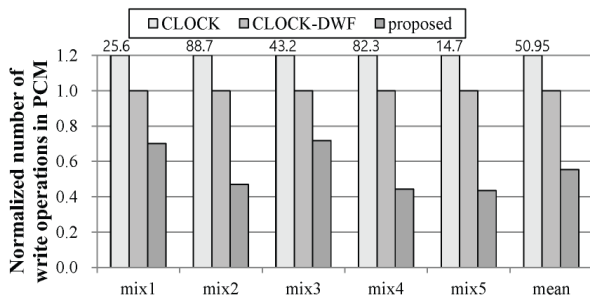


Fig. 7. Normalized number of write operations in PCM

먼저 PCM으로의 쓰기 횟수를 측정하기 위해 1011개의 명령어를 수행한 뒤 쓰기 횟수를 비교했다. 그림 7은 CLOCK-DWF의 PCM에 대한 쓰기 연산 횟수로 정규화 했다. CLOCK은 DRAM 기반의 페이지 교체 정책을 사용하기 때문에 사실상 하이브리드 메인 메모리 환경에서는 적합하지 않다. 한편, 제안하는 페이지 교체 정책은 CLOCK-DWF와 비교하여 평균적으로 44.64%의 쓰기 연산 횟수를 감소시켰다. 제안하는 페이지 교체 정책은 CLOCK-DWF에서 발생하는 PCM으로의 불필요한 쓰기 연산 없이 DRAM에 쓰기 연산이 빈번한 페이지를 유지하여 성능 향상을 보여주고 있다.

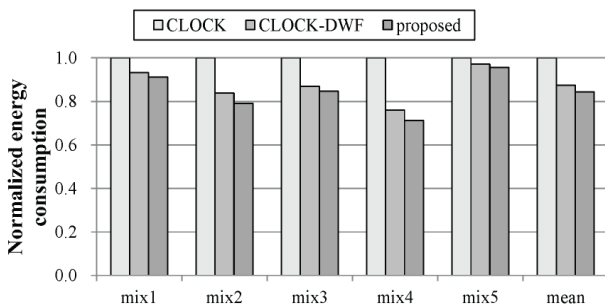


Fig. 8. Normalized energy consumption

다음으로 하이브리드 메인 메모리의 에너지 소모를 비교했다. 그림 8은 CLOCK에 해당하는 하이브리드 메인 메모리의 에너지 소모로 정규화한 결과이다. 제안하는 페이지 교체 정책

은 CLOCK과 CLOCK-DWF와 비교하여 평균적으로 15.61%, 3.04%에 해당하는 에너지 소모를 감소시켰다. CLOCK-DWF는 하이브리드 메인 메모리의 특성을 활용하였기 때문에 CLOCK과 비교하여 12.58%에 해당하는 하이브리드 메인 메모리의 에너지를 줄일 수 있었다. 하지만 CLOCK-DWF는 제안하는 페이지 교체 정책과 비교하여 PCM으로의 쓰기 연산 횟수가 많기 때문에 에너지 소모가 3.04% 더 많은 것을 확인할 수 있었다. 이와 같은 결과는 PCM의 쓰기 연산이 DRAM의 연산보다 평균 10배의 추가적인 에너지 소모가 발생하기 때문이다. 따라서 PCM의 쓰기 횟수는 시스템의 수명과 함께 전체 컴퓨터 시스템의 에너지 소모에 큰 영향을 미친다는 것을 확인했다.

결과적으로 제안하는 페이지 교체 정책은 CLOCK-DWF와 비교하여 불필요한 페이지 이주를 줄였다. 또한 페이지 교체 정책에 있어 쓰기 연산이 빈번한 페이지를 DRAM으로, 쓰기 연산이 빈번하지 않은 페이지를 PCM으로 위치시킴으로써 전반적인 쓰기 연산 횟수 및 에너지 소모를 줄였다.

V. Conclusions

대표적인 비휘발성 메모리인 PCM은 낮은 대기전력과 높은 직접도로 인하여 기존의 DRAM을 대체할 새로운 메인 메모리로 각광받고 있다. 그러나 PCM은 긴 쓰기 지연 시간과 제한적인 수명 때문에 완전히 DRAM을 대체하기 어렵다. 최근에는 DRAM과 PCM의 장점을 결합한 하이브리드 메인 메모리에 대한 연구가 활발히 진행 중이다. 하지만 하이브리드 메인 메모리는 기존의 DRAM 기반의 페이지 교체 정책을 활용하기에는 부적합하다. 이러한 문제점을 해결하기 위해 CLOCK-DWF는 하이브리드 메인 메모리 기반의 페이지 교체 정책을 제안했다. 그러나 CLOCK-DWF는 불필요한 쓰기 연산 및 DRAM에 상주하는 페이지에 대한 부정확한 예측 등의 문제점이 존재한다.

본 논문에서는 이와 같은 CLOCK-DWF의 문제점을 해결하는 새로운 페이지 교체 정책을 제안했다. 제안된 페이지 교체 정책은 일정한 주기 시간 동안의 쓰기 연산 발생 빈도를 활용하여, 페이지들의 최신성 여부를 결정한다. 이를 활용하여 제안하는 기법은 쓰기 연산이 빈번하고 최신성이 보장되는 페이지들을 DRAM에 상주시키고, 이러한 조건에 부합하지 않은 페이지들은 PCM에 구분하여 상주시킨다. 또한 PCM으로의 빈번한 페이지 이주를 감소시키기 위해, 제안하는 기법은 가까운 미래의 쓰기 연산이 빈번하게 발생할 페이지와 접근 빈도가 낮은 페이지를 구분했다.

성능 평가에서 본 논문에서 제안하는 페이지 교체 정책은 하이브리드 메인 메모리 기반의 CLOCK-DWF와 비교하여 PCM의 페이지 이주 횟수와 쓰기 연산 횟수를 평균적으로 32.12%, 44.64% 감소 시켰다. 또한 하이브리드 메인 메모리의 에너지 소모는 CLOCK 및 CLOCK-DWF와 비교하여 평균적으로

15.61%, 3.04% 감소 시켰다. 향후 연구에서는 실제 컴퓨팅 환경에서 하이브리드 메인 메모리를 적용하였을 때 발생할 수 있는 세부적인 고려 사항에 대한 연구를 진행할 예정이다.

REFERENCES

- [1] Dhiman. G, Ayoub. R, & Rosing. T, "PDRAM: a hybrid PRAM and DRAM main memory system," DAC'09, 46th ACM/IEEE, Design Automation Conference, pp. 664-669, July, 2009.
- [2] International Roadmap Committee, "International technology roadmap for semiconductors," 2008.
- [3] Zhou, P, Zhao. B, Yang. J, & Zhang. Y, "A durable and energy efficient main memory using phase change memory technology," ACM SIGARCH computer architecture news. Vol. 37, No. 3, pp. 14-23, June, 2009.
- [4] Qureshi. M. K, Srinivasan. V, & Rivers. J. A. "Scalable high performance main memory system using phase-change memory technology," ACM SIGARCH Computer Architecture News. Vol. 37, No. 3, pp. 24-33, 2009.
- [5] Qureshi. M. K, Karidis. J, Franceschini. M, Srinivasan. V, Lastras. L, & Abali. B, "Enhancing lifetime and security of PCM-based main memory with start-gap wear leveling," Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture, pp. 14-23, December, 2009.
- [6] Ferreira. A. P, Zhou. M, Bock. S, Childers. B, Melhem. R, & Mossé. D, "Increasing PCM main memory lifetime," Proceedings of the conference on design, automation and test in Europe. European Design and Automation Association. pp. 914-919, March, 2010.
- [7] Long, Linbo, et al. "A compiler assisted wear leveling for morphable PCM in embedded systems." Journal of Systems Architecture 71, pp. 32-43, 2016.
- [8] Khouzani, Hoda Aghaei, Fateme S. Hosseini, and Chengmo Yang. "Segment and Conflict Aware Page Allocation and Migration in DRAM-PCM Hybrid Main Memory." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 36, No. 9, pp. 1458-1470, 2017.
- [9] Shirinzadeh, Saeideh, et al. "Endurance management for resistive Logic-In-Memory computing architectures." 2017 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 1092-1097, 2017.
- [10] Yu. H, & Du. Y, "Increasing Endurance and Security of Phase-Change Memory with Multi-Way Wear-Leveling," IEEE Transactions on Computers, Vol. 63, No. 5, pp. 1157-1168, 2014.
- [11] Yu. H, & Du. Y, "Increasing Endurance and Security of Phase-Change Memory with Multi-Way Wear-Leveling," IEEE Transactions on Computers, Vol. 63, No. 5, pp. 1157-1168, 2014.
- [12] Xia. F, Jiang. D. J, Xiong. J, & Sun. N. H. "A survey of phase change memory systems," Journal of Computer Science and Technology, Vol. 30, No. 1, pp. 121-144, 2015.
- [13] Mittal. S. A "survey of power management techniques for phase change memory," Memory, 1223, Vol. 41, No. 1, 2015.
- [14] Jiang. S, & Zhang. X, "LIRS: an efficient low inter-reference recency set replacement policy to improve buffer cache performance," ACM SIGMETRICS Performance Evaluation Review, Vol. 30, No. 11, pp. 31-42, 2002.
- [15] Megiddo. N, & Modha. D. S, "ARC: A Self-Tuning, Low Overhead Replacement Cache," FAST, Vol. 3, pp. 115-130, March, 2003.
- [16] Bansal. S, & Modha. D. S. "CAR: Clock with Adaptive Replacement," FAST, Vol. 4, pp. 187-200, March, 2004.
- [17] Jiang. S, Chen. F, & Zhang. X, "CLOCK-Pro: An Effective Improvement of the CLOCK Replacement," USENIX Annual Technical Conference, General Track, pp. 323-336, April, 2005.
- [18] Lee. S, Bahn. H, & Noh. S. H, "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, 2014.
- [19] Raoux. S, Burr. G. W, Breitwisch. M. J, Rettner. C. T, Chen. Y. C, Shelby. R. M, ... & Lam. C. H. "Phase-change random access memory: A scalable technology," IBM Journal of Research and Development, Vol. 52, No. 4.5, pp. 465-479, 2008.
- [20] Lee. B. C, Ipek. E, Mutlu. O, & Burger. D, "Architecting phase change memory as a scalable dram alternative," ACM SIGARCH Computer Architecture News. Vol. 37, No. 3, pp. 2-13. June, 2009.
- [21] Yang. B. D, Lee. J. E, Kim. J. S, Cho. J, Lee. S. Y, & Yu. B. G, "A low power phase-change random access memory using a data-comparison write scheme," 2007 IEEE International Symposium on Circuits and Systems, pp. 3014-3017, May, 2007.
- [22] Cho. S, & Lee. H, "Flip-N-Write: a simple deterministic technique to improve PRAM write performance, energy and endurance," 2009 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO),

pp. 347-357, December, 2009.

- [23] Zhao. M, Shi. L, Yang. C, & Xue. C. J, "Leveling to the last mile: Near-zero-cost bit level wear leveling for PCM-based main memory," 2014 IEEE 32nd International Conference on Computer Design (ICCD), pp. 16-21, October, 2014.
- [24] Corbato. F. J, "A paging experiment with the multics system (No. MAC-M-384)," MASSACHUSETTS INST OF TECH CAMBRIDGE PROJECT MAC. 1968.
- [25] Binkert. N, Beckmann. B, Black. G, Reinhardt. S. K, Saidi. A, Basu. A, ... & Sen. R, "The gem5 simulator," ACM SIGARCH Computer Architecture News, Vol. 39, No. 22, pp. 1-7, 2011.
- [26] Binkert. N. L, Dreslinski. R. G, Hsu. L. R, Lim. K. T, Saidi. A. G, & Reinhardt. S. K, "The M5 simulator: Modeling networked systems," IEEE Micro. Vol. 26, No. 4, pp. 52-60, 2006.
- [27] Martin. M. M, Sorin. D. J, Beckmann. B. M, Marty. M. R, Xu. M, Alameldeen. A. R, ... & Wood. D. A, "Multifacet's general execution-driven multiprocessor simulator (GEMS) toolset," ACM SIGARCH Computer Architecture News. Vol. 33, No. 4, pp. 92-99, 2005.
- [28] Henning. J. L, "SPEC CPU2006 benchmark descriptions," ACM SIGARCH Computer Architecture News. Vol. 34, No. 4, pp. 1-17, 2006.
- [29] Phansalkar. A, Joshi. A, & John. L. K, "Analysis of redundancy and application balance in the SPEC CPU2006 benchmark suite," ACM SIGARCH Computer Architecture News. Vol. 35, No. 2, pp. 412-423, 2007.

Authors



Sungho Kim received a B.S. degree in Department of Computer Engineering from Yeungnam University College, Daegu, Korea in 2012. He is currently a Ph.D. candidate in Department of Computer Engineering from Yeungnam University.

His current research interests include embedded systems and non-volatile memory systems.



Jong Wook Kwak received the B.S. degree in Computer Engineering from Kyungpook National University, Taegu, Korea in 1998, the M.S. degree in Computer Engineering from Seoul National University, Seoul, Korea in 2001, and the Ph.D. degree in

Electrical Engineering and Computer Science from Seoul National University, Seoul, Korea in 2006, respectively. From 2006 to 2007, he was a Senior Engineer with the system-on-chip (SoC) Research and Development Center, Samsung Electronics Company, Ltd., Suwon, South Korea. During 2011-2012, he was a Guest Researcher at the Research Institute of Advanced Computer Technology, Seoul National University. During 2012-2013, he was a Visiting Scholar at the Georgia Institute of Technology, Atlanta, GA, USA. He is currently an Associate Professor with the Department of Computer Engineering, Yeungnam University, Gyeongsan, South Korea. His current research interests include SoC design, advanced processor architecture, low-power mobile embedded system design, and high-performance parallel and distributed computing.