논문 2018-13-30

DRAM&PCM 하이브리드 메모리 시스템을 위한 능동적 페이지 교체 정책

(Active Page Replacement Policy for DRAM & PCM Hybrid Memory System)

정 보 성, 이 정 훈*
(Bo-Sung Jung, Jung-Hoon Lee)

Abstract : Phase Change Memory(PCM) with low power consumption and high integration attracts attention as a next generation nonvolatile memory replacing DRAM. However, there is a problem that PCM has long latency and high energy consumption due to the writing operation. The PCM & DRAM hybrid memory structure is a fruitful structure that can overcome the disadvantages of such PCM. However, the page replacement algorithm is important, because these structures use two memory of different characteristics. The purpose of this document is to effectively manage pages that can be referenced in memory, taking into account the characteristics of DRAM and PCM. In order to manage these pages, this paper proposes an page replacement algorithm based on frequently accessed and recently paged. According to our simulation, the proposed algorithm for the DRAM&PCM hybrid can reduce the energy-delay product by around 10%, compared with Clock-DWF and CLOCK-HM.

Keywords : Hybrid memory, Non-volatile memory, High-performance, Low-power, Memory characteristics

Ⅰ. 서 론

현재 대용량의 다양한 어플리케션의 등장과 IoT(Internet of Things) 시스템이의 급속한 발전으로 저전력 및 고성능 메모리가 요구되어지고 있다[1].이러한 이유로 PCM(Phase Change Memory), RRAM(Resistive RAM) 및 STT-RAM (Spin Transfer Torque RAM) 등의 비휘발성 메모리가 오늘날 차세대 메모리로 주목받고 있다. 이러한 차세대 비휘발성 메모리는 저전력, 고집적도 그리고 빠른 메모리 접근 시간으로 기존의 SRAM 및 DRAM의 대체 메모리로 많은 연구가 이루어지고 있다[2-4].

PCM은 DRAM처럼 바이트-주소 접근이 가능하며, DRAM에 비해 약 4배의 높은 집적도를 가진다. 또한 DRAM과 달리 비휘발성 메모리로 데이터를 유지하기 위한 재충전 동작(reflash)이 필요하지 않다[5].

현재 컴퓨팅 시스템에서 메인 메모리인 DRAM은 전체 전력소비에서 약 40%를 차지하고 있다. 더욱이 이러한 에너지 소비는 대부분 데이터를 유지하는 리플래시 동작 및 누설 전류가 대부분이다. 또한 DRAM은 현재 집적도의 한계를 보이고 있다. 만약 DARM의 집적도를 향상시킬 경우, 터널링 효과 및 셀간의 간섭 문제를 해결해야 한다. 따라서 DRAM은 더 이상 저전력 및 소형화를 위한 컴퓨팅 시스템에 이상적인 메인 메모리 역할을 수행 할 수 없다 [6].

이에 반해, PCM은 앞서 언급한 특성으로 현재 DRAM을 대체할 차세대 메모리로 주목 받고 있다. 더욱이, PCM은 DRAM과 유사한 공정 과정을 가지므로, 다른 비휘발성 메모리에 비해 설비투자와 개발 비용의 관점에서 유리하다.

하지만, 이러한 PCM의 특성에도 불구하고, PCM은 아직까지 DRAM을 대체를 위한 메인 메모리로 상용하기 위해 큰 문제점을 가진다. 기본적으로

*Corresponding Author (leejh@gsnu.ac.kr)
Received: July 31. 2018, Revised: Oct. 1. 2018, Accepted: Oct. 2. 2018.
B.S. Jung, J.H. Lee : GyeongSang National University(ERI)

ⓒ IEMEK J. Embed. Sys. Appl. 2018 Oct. 13(5) 261-268
ISSN : 1975-5066
http://dx.doi.org/10.14372/IEMEK.2018.13.5.261

PCM은 NAND 플래시 메모리와 같이 쓰기 횟수(10^7-10^8)가 제한되어져 있다[7]. 그리고 비록 PCM이 비휘발성 메모리로 낮은 대기 전력과 DRAM과 같은 리플래시 동작이 없지만, 쓰기 연산에 대해 높은 구동 전력과 긴 지연시간을 가진다. 이렇듯, 현재 PCM은 다양한 장점에도 불구하고, 쓰기 연산에 대한 높은 구동 전력과 지연시간으로 단독으로 현재의 메인 메모리의 역할을 수행 할 수 없다.

따라서 PCM을 메인 메모리로 사용하기 위해 이러한 단점을 해결하기 위한 대표적인 방법으로 DRAM과 PCM을 같이 사용하는 하이브리드 메모리 구조가 제안되었다 [7-9].

이러한 하이브리드 메모리 구조에서 DRAM은 빠른 메모리 접근 시간과 쓰기 연산에 대한 횟수 제한이 없기 때문에, 쓰기 연산에 대한 참조 가능성이 높은 페이지를 가지는 것이 효과적이다. 반대로 PCM은 제한된 쓰기 동작과 긴 지연시간의 단점가 지지만, 읽기 연산에 대해 DRAM과 비슷한 성능을 보인다. 따라서 PCM은 읽기 연산에 대한 참조 가능성이 높은 페이지를 저장하는 것이 유리하다. 따라서 DRAM&PCM 하이브리드 메모리 구조에서 각 메모리의 특성에 적합한 페이지 할당 및 교체 관리에 대한 정책이 중요하다.

본 논문에서는 저전력 및 고집적도 차세대 메인 메모리를 위해 DRAM & PCM 하이브리드 메모리 구조에 대한 새로운 효과적인 페이지 정책을 제안하였다. 본 논문에서 DRAM과 PCM은 동일한 계층의 메모리로 구성된다. 그리고 본 논문에서 제시된 페이지 정책은 기본적으로 최근 참조된 페이지 및 빈번하게 발생하는 페이지를 하이브리드 메모리에 오랜 시간 유지하는 방법을 제시 하였다. 제안된 페이지 정책은 각 메모리의 특성을 고려하여, DRAM은 쓰기 연산 중심의 페이지 정책을 그리고 PCM은 읽기 연산 중심의 페이지 정책을 가진다.

Ⅱ. 관련연구

1. 페이지 교체 알고리즘

S벨레디 변이 현상 발견으로 최적 페이지 대체 알고리즘을 연구가 지속적으로 이루어지고 있다.

LRU 알고리즘은 과거의 데이터를 이용하여 페이지 참조를 예측하려는 통계적 개념으로 메모리의 지역성을 고려한 알고리즘이다. 가상 메모리 환경에서 페이지 참조는 시간 지역성 중심으로 실행되어지고 있다. 시간 지역성은 프로그램 실행 시 '최근 참조가 발생한 데이터는 빠른 시간에 다시 참조 가능성이 높다'라는 것을 의미한다. 하지만 이러한 LRU 알고리즘이 좋은 성능을 보장하지만, 실제 구현을 위한 하드웨어의 (계수기 혹은 스택) 지원이 필요하다. 더욱이, 모든 메모리에서 접근이 발생하면, LRU는 페이지들을 모두 최근 사용한 위치로 이동시켜야한다. 이는 충분히 메모리 시스템의 성능저하를 야기 시킬 원인이 될 수 있다.

CLOCK 알고리즘은 근사 LRU 알고리즘으로 제안되었다 [12]. CLOCK 알고리즘은 기존의 LRU 알고리즘과 FIFO 알고리즘의 병합방법이다. 메모리의 페이지 유지를 위해서, 단지 페이지가 참조 되었는지 그렇지 않는지를 모니터링 한다. CLOCK 알고리즘은 희생 페이지 선택을 위해 circular list로 구동되며, 그를 위해 페이지 프레임을 가리키는 클럭 포인터를 가진다. 비록 CLOCK 알고리즘이 정확한 LRU 시스템을 구동하지 못한다고 하여도 최근 참조 되지 않은 페이지를 선정할 수 있다.

2. Hybrid memory 관련연구

현재 DRAM & PCM 하이브리드 시스템의 구조는 DRAM의 운용에 따라 크게 2가지로 나눌 수 있다. 첫 번째로 DRAM 버퍼 & PCM 하이브리드 메모리 구조이다. 이러한 구조는 PCM에서 빈번하게 접근이 발생하는 블록을 DRAM에 저장하므로 PCM의 접근을 숨길 수 있다. 더욱이, DRAM의 버퍼는 다양한 블록 크기를 가질 수 있으므로, 프로그램 수행 시 적합한 지역성을 효과적으로 이용 할 수 있다.

Qureshi [10]는 PCM과 동일한 페이지 크기를 가지는 DRAM 버퍼 구조를 제안하였다. Selective data buffering [11]은 PCM을 위해 다양한 블록 크기를 가지는 DRAM 버퍼 구조를 제안하였다. 하지만 이러한 동작은 DRAM 버퍼로부터 PCM에 순차적으로 접근이 발생하게 된다. DRAM write-cache [12]는 캐시 시스템으로부터 PCM에 쓰기 연산을 줄이기 위한 DRAM 버퍼 구조를 제안하였다. 하지만 이 연구는 PCM에서 직접 접근에 대한 쓰기 연산이 고려되어있지 않다.

두 번째로, DRAM과 PCM이 동일한 계층의 메인 메모리로 구성되는 DRAM & PCM 하이브리드 구조이다. 이러한 구조는 기존의 페이지 테이블을 효과적으로 사용할 수 있다. DRAM & PCM 하이브리드 메모리 시스템 구조에서는 성능향상을 위해 효과적인 페이지 교체 알고리즘이 필요하다.

CLOCK-DWF [7], CLOCK-HW [8]은 DRAM

과 PCM이 동일한 계층의 메모리로 DRAM에 쓰기 요청이 빈번한 페이지를 관리함으로 하이브리드 메모리 시스템의 성능향상을 도모하였다.

CLOCK-HW는 하이브리드 메모리에서 추출되는 페이지 정보를 가지는 history list를 이용하여 프로그램 수행 시 페이지의 특성을 예측하였다. 하지만, 빈번한 쓰기 요청 페이지를 제외한 페이지들이 DRAM과 PCM에 무작위 할당되어 효과적인 메모리 활용이 어려운 단점이 있다.

CLOCK-DWF는 각 메모리의 특성을 고려하여 DRAM은 쓰기 연산 요청 페이지를, PCM은 읽기 연산 요청 페이지를 관리하게 된다. 또한 PCM의 쓰기 연산을 줄이기 위해 DRAM에서 쓰기 연산이 빈번한 페이지를 오랜 시간 유지하는 알고리즘을 제안하였다. 하지만 CLOCK-DWF는 DRAM에서 빈번한 쓰기 요청 페이지를 선정을 위해 추가적인 연산비트가 필요하다.

더욱이, 이러한 하이브리드 메모리는 CLOCK 알고리즘을 바탕으로 단순히 참조가 발생한 페이지만 확인할 수 있다. 본 논문에서는 기존 CLOCK 알고리즘을 바탕으로 적극적인 근사 LRU 방법을 모색하였다.

Ⅲ. 본 론

1. 제안 동기 및 방법

본 논문에서 제안된 DRAM&PCM 하이브리드 메모리 구조는 각 메모리가 동일한 계층의 메모리 구조를 가진다. 이러한 구조는 기존의 페이지 테이블을 이용 할 수 있어, 전체적인 하이브리드 메모리에서 페이지를 쉽게 관리 할 수 있다는 장점을 가진다. 이러한 장점과 더불어 각 메모리의 특성에 적합하게 페이지 저장을 할 수 있다. 즉 DRAM은 쓰기 횟수 제한이 없고, 빠른 접근 시간을 보장하므로 쓰기 요청 페이지 및 빈번하게 접근하는 페이지를 관리하는 것이 효과적이다. 그리고 PCM은 높은 집접도와 읽기 연산이 DRAM과 유사한 성능을 가지므로 데이터 저장 및 읽기 요청이 빈번한 페이지를 관리하는 것이 효과적이다.

현재 DRAM&PCM 하이브리드 메모리 구조에서 근사 LRU 기법인 CLOCK 알고리즘을 채용하고 있다. 하지만, CLOCK 알고리즘은 단순히 참조가 발생한 페이지와 그렇지 못한 페이지만을 선택할 수 있다. 즉 페이지 체크를 위한 CLOCK-Handle을 기준으로 최근 참조가 발생한 페이지는 구분하지 못하는 단점이 있다.

CLOCK-DWF는 빈번한 쓰기 요청 페이지의 카운터 비트와 빈번한 쓰기 요청이 발생한 페이지의 추출을 지연시키는 비트를 이용하여 이러한 CLOCK 알고리즘의 단점을 보완하였다. 하지만, 이러한 동작은 추가적인 비트를 요구 할 뿐 아니라, 여전히 CLOCK 알고리즘에서 최근 참조가 발생한 페이지 선정이 어렵다.

본 논문에서는 이러한 CLOCK 알고리즘의 단점을 보완하기 위해 적극적인 근사 LRU 방법으로 CLOCK-Handle 기준으로 참조가 발생한 페이지의 접근을 나타내기 위해 DRAM과 PCM은 메모리 특성을 고려한 2비트의 참조 비트 (DRAM : W0/W1, PCM : R0/R1)를 가진다. 만약 페이지 참조가 CLOCK-Handle 이전 페이지에서 발생한다면, 제안된 하이브리드 메모리에서는 첫 번째 참조비트 (W0,R0)비트가 갱신되며, 그렇지 못하고, CLOCK-Handle 이후의 페이지에서 접근 성공이 발생하면, 두 번째 참조비트 (W1,R1)비트가 갱신이 된다. 만약 희생 페이지 선택을 위해 페이지가 검사가 일어나면, 참조 비트R0는 R1에 의해 갱신되며, R1은 '0'으로 갱신된다. 이러한 동작은 CLOCK-Handle에 의해 최근 참조가 발생한 페이지를 선정할 수 있다.

앞서 언급한 내용과 같이 DRAM&PCM 하이브리드 메모리 구조에서 DRAM은 쓰기 요청 페이지를 관리하는 것이 효과적이다. 따라서 쓰기 요청이 빈번하게 접근이 발생 할 가능성이 높은 페이지를 유지하는 것이 성능향상에 효과적이다. 본 논문에서는 DRAM에 쓰기 요청이 빈번하게 발생하는 페이지 선택을 위해 추가적으로 쓰기 요청 카운터 비트 (Write_Count, WC)를 사용하였다. 그리고 쓰기 요청이 빈번하게 발생한 페이지는 DRAM 페이지들의 WC 값의 평균값보다 높은 페이지를 선정하였다.

이때 빈번하게 쓰기 요청된 페이지의 효용성을 판단하기 위해 R 비트는 '0'으로 갱신된다. 여기서 R 비트는 DRAM의 메모리 접근을 여부를 나타내는 추가적인 참조비트 (Reference, R)이다. R비트는 페이지 요청 접근이 성공하면, '1'로 갱신된다.

만약 WC 값이 쓰기 연산의 평균값보다 높다라면, 그 페이지는 쓰기 연산이 다시 발생 할 가능성이 높은 페이지로 두 번째 기회를 주게 된다. 따라서 본 논문에서는 빈번하게 발생하는 페이지의 오래 유지하기 위해 추가적인 기회 비트 (Second chance, S)를 가진다.

그리고 PCM은 고집적도로 읽기 연산 페이지나 쓰기 연산 요청 가능한 페이지를 관리하는 것이 효

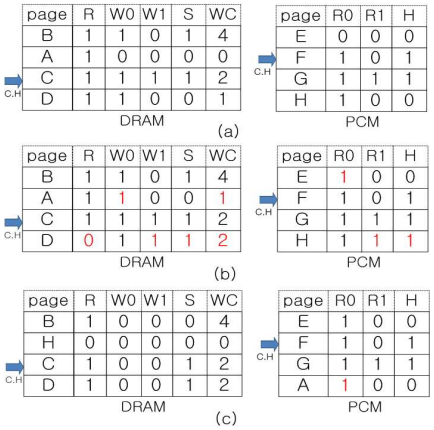

그림 1. 제안된 하이브리드 메모리의 알고리즘 동작
Fig. 1 The algorithm operation of the proposed hybrid memory

과적이다. 따라서 PCM의 참조 비트는 쓰기 연산 및 읽기 연산 페이지 요청에 의해 관리되어진다.

만약 PCM에서 쓰기 요청이 발생하면, 그 페이지는 DRAM에 저장 되어져야 한다. 하지만, 빈번한 페이지 교체는 오히려 PCM의 쓰기 연산의 증가로 전체 시스템의 성능저하의 원인이 된다. 본 논문에서 이러한 빈번한 페이지 교체를 줄이기 위해 빈번한 페이지 쓰기 요청이 발생한 페이지만 페이지 교체가 발생한다. 따라서 PCM에서 빈번한 쓰기 요청이 발생 가능성이 높은 페이지를 선택하기 위한 추가 비트 (write_hot page, H)를 사용하였다. H 비트는 DRAM으로부터 PCM으로 추출되는 페이지가 쓰기 연산의 평균값보다 높거나, PCM에서 한번 쓰기 요청이 발생한 페이지를 의미한다.

앞서 언급한 내용과 같이, DRAM은 쓰기 연산 페이지를 관리하며, PCM은 읽기 연산 페이지 및 쓰기 요청 발생 가능한 페이지를 관리하는 것이 효과적이다. 따라서 본 논문에서 제안된 하이브리드 메모리에서 최하위 계층으로부터 요청된 페이지가 쓰기 연산이면 DRAM에 저장되며, 읽기 연산이 요청된 페이지는 PCM에 저장된다.

만약 DRAM에 페이지를 저장할 여유 공간이 없다면 DRAM은 희생 페이지를 선택하게 된다. 만약 희생 페이지는 W0/W1 비트와 S 비트가 모두 '0'인 페이지를 선택하게 되며, PCM에 저장된다. 반면, R 비트 역시 '0'이라면 이 페이지는 PCM으로 이동하지 않고 최하위계층 메모리로 이동하게 된다. PCM 역시 상태비트가 모두 '0'인 페이지가 희생 페이지로 선택되어진다.

2. 제안된 페이지 관리 방법

2.1 DRAM 접근

만약, 그림 1 (a) DRAM은 현재 DRAM의 상태를 나타내고 있다. 그림 1 (a)에서 읽기 요청 접근이 발생 했다면, 페이지 접근을 나타내는 R 비트만 갱신된다. 반면, 페이지 A에 쓰기 요청이 접근이 발생 했다면, 현재 페이지A는 CLOCK-Handle (C.H) 이전 페이지로 W0 비트가 '1'로 갱신되며, WC가 '1'증가하게 된다. 이때, 페이지 A의 WC는 전체 WC의 평균값보다 작다. 따라서 페이지 A의 S 비트는 갱신되지 않는다. 반면, 페이지 B에서 쓰기 요청이 발생하면, W0비트가 '1'이기 때문에 어떠한 상태 비트의 갱신도 발생하지 않는다. 반면, 페이지 D에서 쓰기 요청이 발생하면, 페이지 D는 현재 CLOCK-Handle 이후의 위치에 있기 때문에 W1 비트를 검사하게 된다. 현재 W1 비트가 '0'이기 때문에 W1 비트는 '1'로 갱신되며, W.C 비트가 증가하게 된다. 페이지 D의 WC는 전체 W.C의 평균값보다 높기 때문에 S비트가 '1'로 갱신된다. 그리고 이 페이지 D의 효용성 여부를 판단하기 위해 R비트는 '0'으로 갱신된다.

그림 1 (b)의 DRAM은 위에서 언급한 페이지들의 쓰기 연산에 대한 결과를 나타내고 있다.

2.2 PCM 접근

그림 1 (a)의 PCM은 현재 PCM을 나타내고 있다. 만약 페이지 E에서 읽기 요청이 발생하면, 페이지 E가 현재 CLOCK-Handle의 이전 위치이므로 R0가 '1'로 갱신이 된다. 반면 페이지 H가 읽기 요청이 발생하면, 페이지 H는 CLOCK-Handle 이후의 페이지로 R1 비트가 '1'로 갱신된다.

만약 PCM에서 페이지 E 혹은 페이지 H에서 쓰기 요청이 발생하면, R0, R1 및 H 비트가 모두 '1'로 갱신이 된다. 반면, 페이지 F 혹은 페이지 G에서 쓰기 요청이 발생하면, 현재 H 비트가 '1'이므로 DRAM과 PCM 사이의 페이지 교체가 발생하게 된다.

그림 1 (a)의 PCM은 페이지 E에서 읽기 요청과 페이지 H에서 쓰기 요청이 발생한 상태를 나타내고 있다.

2.3 DRAM&PCM 페이지 교체

만약 그림1 (b)의 PCM에서 페이지 H에서 쓰기 요청이 발생하면, 페이지 H의 H비트가 '1'이기 때문에 페이지 H는 DRAM에 저장되어진다. 이때 DRAM은 희생 페이지를 선택하게 된다. 그림 1 (b)

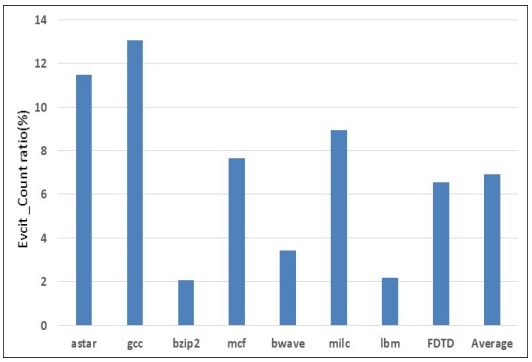

그림 2. 제안된 하이브리드 메모리의 페이지 추출 횟수 비율

Fig. 2 Evict_count ratio of the propsoed hybrid memory

의 DRAM에서 CLOCK-Handle이 가리키는 페이지 C부터 상태비트를 검사하게 된다. 페이지 C의 W0 비트가 '1'이므로 희생 페이지 후보군에 제외되며, 이때 W0 비트는 W1에 의해 '1'로 갱신된다. 그리고 W1비트는 '0'으로 갱신된다. 이러한 동작으로 페이지 W0비트 때문에 1cycle 동안 모든 페이지가 희생 페이지 후보군에서 제외된다. 그리고 다시 페이지 C와 페이지 D는 W0 비트에 의해 희생 페이지 후보군에서 제외된다. 그리고 페이지 B는 비록 W0와 W1비트가 '0'이지만, S 비트에 의해 희생 페이지 후보군에서 제외되며, 이때 S비트는 '0'으로 갱신된다. 그리고 페이지 A는 W0, W1 그리고 S비트가 모두 '0'으로 희생 페이지로 선택되어진다.

선택 되어진 희생 페이지 A가 PCM에 저장 될 때, 페이지 A의 R비트는 PCM의 R0비트에 계승되어진다.

그림 1 (c)는 (b)의 상태에서 페이지 교체가 발생한 상태는 나타내고 있다.

만약 그림 1 (b)에서 페이지 B혹은 페이지 D가 희생 페이지로 선택되어진다면, WC가 전체 WC의 평균값보다 높기 때문에 이 페이지는 쓰기 참조 가능성이 높은 페이지로 정의 되며, PCM에 저장될 때 H 비트가 '1'로 갱신된다.

2.4 메모리 접근 실패

만약 제안된 하이브리드 메모리에서 접근 실패가 발생하면, 최하위 계층 메모리로부터 페이지를 가져온다. 만약 쓰기 페이지가 요청되면, 그 페이지는 DRAM에 저장된다. 만약 페이지 저장을 위한 여유 공간이 없다면, 희생 페이지를 선택하게 된다.

표 1. DRAM과 PCM 특성[7]

Table 1. DRAM & PCM characteristic[7]

	DRAM	PCM
내구성	N/A	10^7-10^8
read latency	$50(ns)$	$50\sim100(ns)$
write latency	$50(ns)$	$350(ns)$
read energy	$0.1(nJ/b)$	$<1.0(nJ/b)$
write energy	$0.1(nJ/b)$	$<1.0(nJ/b)$

반면, 읽기 페이지가 요청된다면 그 페이지는 PCM에 저장된다. 만약 그림 1 (c)와 같이 페이지 저장 공간이 없다면, 희생 페이지 선택을 위해 페이지 F부터 상태 비트를 검사하게 된다. 페이지 F는 R0비트에 의해 희생 페이지 후보군에서 제외된다. 그리고 페이지 G 역시 R0의해 희생 페이지 후보군에서 제외된다. 이때 R0비트는 R1에 의해 '1'로 갱신되며, R1비트는 '0'으로 갱신된다. 페이지 A 역시 R0비트에 의해 희생 페이지 후보군에서 제외된다. 그리고 페이지 F는 비록 R0와 R1비트가 '0'이지만, 참조 가능성이 높은 페이지를 나타내는 H비트가 '1'이므로 희생 페이지에서 제외되며, H비트는 '0'으로 갱신된다. 이러한 동작으로 PCM에서 페이지 A는 희생 페이지로 선택되어진다.

Ⅳ. 성능 평가

본 논문에서는 제안된 페이지 관리정책의 성능 평가를 위하여 Spec CPU 2006벤치마크의 실행되는 동안 메모리에 접근하는 트레이스 파일을 추출하여 사용하였다. 트레이스 파일은 valgrind의 Cachegrind [13]를 수정하여 L1캐시와 L2캐시로부터 접근 실패 후 메모리에 접근하는 1억 개의 데이터의 주소를 사용하였다. 성능평가를 위한 DRAM과 PCM의 파라미터는 표 1과 같다.

본 논문에서 DRAM & PCM 하이브리드 메모리의 성능 평가를 위해 PCM의 크기를 기존 메인 메모리 모듈 크기인 128Mbyte로 설정하였으며, DRAM의 크기는 PCM과 DRAM의 집적도를 고려하여 32Mbyte로 설정하였다. 그리고 각 메모리의 페이지 크기는 기존 메인 메모리의 페이지 크기인 4Kbyte로 설정하였다. 그리고 본 논문에서 WC의 최대값은 3으로 정의하였다. 만약 높은 WC의 값을 가지면, 페이지의 효용성 여부를 판단하기 어렵기 때문이다.

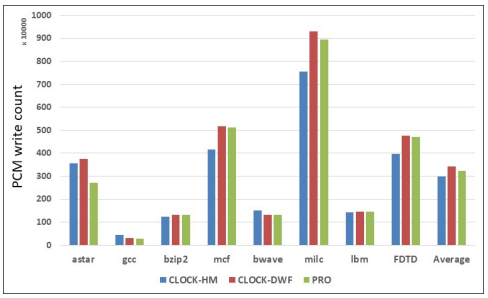

그림 3. 다양한 하이브리드 메모리의 PCM 쓰기 연산 횟수

Fig. 3 PCM write count of the various hybrid memories

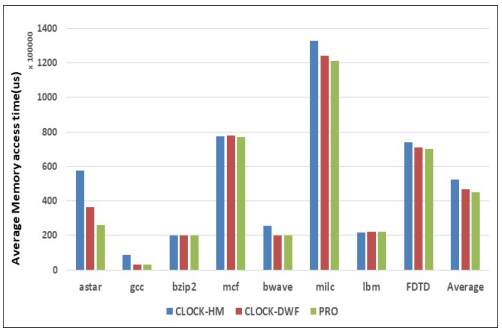

그림 4. 평균 메모리 접근 시간

Fig. 4 Average memory access time

본 논문에서 DRAM에서 접근 효용성이 낮은 페이지는 DRAM으로부터 PCM으로 페이지 이동하지 않고 최하위계층으로 바로 추출하였다. 이러한 동작은 PCM의 쓰기 연산을 줄이는 효과뿐 아니라 PCM의 페이지 요염 역시 줄 일수 있다. 그림 2는 본 논문에서 DRAM으로부터 최하위 계층으로 추출되는 페이지의 비율을 나타내고 있다.

시뮬레이션 결과에 따르면 제안된 하이브리 메모리에서 평균 약 7%의 DRAM으로부터 PCM으로 페이지 이동을 줄였다. 특히 astar과 gcc에서는 다른 벤치마크에 비해 높은 페이지 추출 비율을 보이고 있다. 이는 두 벤치마크에서 짧은 수명의 쓰기 연산 페이지가 많이 존재 한다 것을 의미한다.

본 논문에서는 제안된 DRAM&PCM 하이브리드 메모리에서 성능 평가를 위해 제안된 하이브리드 메모리와 유사한 구조 및 개념을 가지는 기존의 CLOCK-DWF와 CLOCK-HW와 성능 비교를 하였다. CLOCK-DWF와 CLOCK-HW의 메모리 크기는 본 논문에서 제안된 메모리 크기와 동일하며, CLOCK-HW의 히스토리 버퍼 (history buffer)는 DRAM의 entry와 동일하게 적용하였다.

하이브리드 메모리에서 가장 큰 문제점 하나는 PCM의 쓰기 횟수이다. 그림 3은 각 하이브리드 메모리의 쓰기 횟수를 나타내고 있다.

시뮬레이션 결과에 따르면, CLOCK-HW가 가장 좋은 PCM 쓰기 횟수를 보이고 있다. 이는 실제로 CLOCK-HW는 페이지의 효용성을 고려한 페이지 교체 알고리즘이 없다. 단지 PCM에서 쓰기 요청된 페이지를 단순히 DRAM으로 페이지 이동만 발생하기 때문이다. 그럼에도 불구하고 astar에서는 제안된 하이브리드 메모리 구조에서 가장 효과적인 PCM 쓰기 횟수를 보이고 있다. 본 논문에서는 PCM의 쓰기 횟수를 줄이기 위해 2가지 방법을 사용하였다. 하나는 DRAM으로부터 효용성이 있는 페이지만을 PCM에 저장되며, 그리고 PCM으로부터 쓰기 요청시 빈번하게 쓰기요청이 발생 할 가능성 있는 페이지만 DRAM과 교체하였다. 이런 제안된 방법에서 astar이 가장 적합한 벤치마크이기에 이러한 결과를 보이고 있다.

제안된 하이브리드 메모리는 비록 CLOCK-HW에 비해 약 8%의 높은 PCM의 쓰기 횟수를 보이지만, CLOCK-DWF에 비해 약 7%의 PCM 쓰기 횟수의 성능향상을 이루었다.

그림 4는 각 하이브리드 메모리의 평균 메모리 접근 시간을 나타내고 있다. 하이브리드 메모리의 접근 실패시 최하위 메모리 접근 시간은 15ms로 선정하였다. 시뮬레이션 결과 제안된 하이브리드 메모리 구조가 평균 메모리 접근 시간에서 가장 좋은 성능향상을 보이고 있다.

비록 PCM 쓰기 횟수에서 CLOCK-HM이 가장 좋은 성능을 보이지만, CLOCK-HM은 다른 하이브리드 메모리에 비해 페이지 교체 알고리즘이 없기 때문에 효과적으로 페이지를 관리하지 못한다. 반면, 제안된 하이브리드 메모리는 DRAM으로부터 PCM에 참조 가능한 페이지만 저장하므로 페이지 오염 및 페이지 쓰기 횟수를 줄일 수 있었기 때문이다.

제안된 하이브리드 메모리는 CLOCK-DWF에 비해 약 5%, CLOCK-HM에 비해 약 14%의 평균 메모리 접근 시간의 성능향상을 이루었다.

하이브리드 메모리 시스템에서 PCM의 쓰기 연산에 의해 에너지 소비에 민감하다. 따라서 본 논문에서는 하이브리드 메모리의 에너지 소비를 측정하였다. 그림 5는 각 하이브리드 메모리의 에너지 소비를 나타내고 있다.

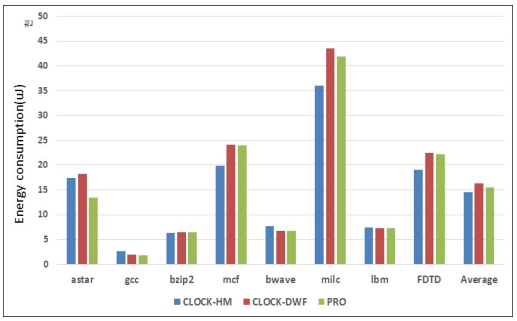

그림 5. 소비 에너지
Fig. 5 Energy consumption.

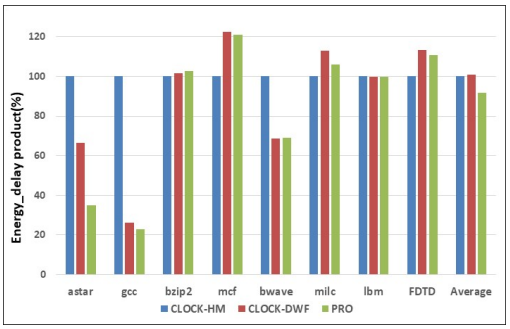

그림 6. 에너지*지연 지표
Fig. 6 Energy-delay product

시뮬레이션 결과에 따르면 CLOCK-HM이 가장 좋은 성능향상을 보이고 있다. 이러한 원인은 CLOCK-HM은 다른 하이브리드 메모리와 달리 DRAM과 PCM 사이의 페이지 교체 알고리즘이 없다. 따라서 PCM의 쓰기 연산 횟수가 다른 하이브리드 메모리에 비해 좋은 성능을 가진다. 실제 하이브리드 메모리에서 에너지 소비는 PCM의 쓰기 연산 횟수에 민감하다.

시뮬레이션에 따르면, 제안된 하이브리드 메모리는 CLOCK-HM에 비해 약 6%의 성능감소를 나타냈으며, CLOCK-DWF에 비해 약 6%의 성능향상을 보였다.

평균 메모리 접근시간에서는 제안된 하이브리드 메모리에서 좋은 성능을 보인 반면, 에너지 소비에서는 CLOCK-HM에서 가장 좋은 성능향상을 이루었다. 따라서 본 논문에서는 전체 성능평가를 에너지*지연 지표를 평가하였다. 에너지 *지연 지표는 소비 에너지와 평균 메모리 접근 시간의 곱으로 정의 된다.

그림 6은 에너지*지연 지표를 CLOCK-HM의 기준으로 평준화하여 나타내고 있다.

시뮬레이션 결과 제안된 하이브리드 메모리는 다른 비교 메모리에 비해 가장 좋은 성능을 보이고 있다. 시뮬레이션 결과에 따르면, 제안된 하이브리드 메모리는 CLOCK-DWF와 CLOCK-HM에 비해 평균 10%의 성능향상을 보였다.

Ⅳ. 결 론

본 논문은 오늘날 대용량의 어플리케이션의 등장과 IoT 시스템의 급속한 발달로 인한 저전력 고성능 컴퓨팅 시스템이 요구되어지고 있다. 이에 차세대 비휘발성 메모리를 이용한 DRAM&PCM 하이브리드 메모리를 위한 효과적인 페이지 교체 정책을 제안하였다. 제안된 페이지 교체 정책은 PCM의 가장 큰 문제점인 쓰기 연산을 줄이기 위해 DRAM으로부터 유용한 페이지만을 PCM에 저장하였으며, PCM으로부터 쓰기 참조 발생 가능한 페이지만을 정의하여 DRAM과 PCM의 페이지 교체를 이루었다. 이러한 방법은 하이브리드 메모리에서 페이지 오염을 줄이는 효과적인 방법이라 할 수 있다.

또한 DRAM과 PCM의 특성을 고려하여 DRAM은 쓰기 연사에 대한 페이지를 관리하며, PCM은 읽기 연산 및 쓰기 연산에 대한 참조 가능성이 높은 페이지를 관리함으로 전체적으로 효과적인 페이지 관리 방법을 제시 하였다.

시뮬레이션 결과 본 논문에서 제안된 하이브리드 메모리와 유사한 구조 및 개념의 CLOCK-DWF와 CLOCK-HM에 비해 에너지 *지연 지표에서 약 평균 10%의 성능향상을 이루었다.

References

[1] T.J. Pack, W.Y. Jang, "Large-scale Last-level Cache Design Based on Parallel TLC STT-MRAM," Jounal of KIIT, Vol. 15, No, 12, pp. 77-89, 2017.

[2] P.P. Palangappa, J.Li, K. Mohanram, " WOM-Code Solutions for Low Latency and High Endurance in Phase Change Memory," IEEE Transactions on Computers, 2016, Vol. 65, Issue 4, pp. 1025-1040.

[3] A. Awad, B. Kettreing, and Y. Solihin, "Non-volatile memory host controller interface performance analysis in high-performance I/O systems," ISPASS, 2015 IEEE International Symposium on, 2015, pp. 145-153.

[4] J.C. Mogul, E. Argollo, M. Shah, and P. Faraboschi, “Operation system support for NVM+DRAM hybrid main memory,” HotOS′09 Proceedings of the 12th conference on Hot topics in operating systems. 2009, pp.4-14.

[5] H. Seok, Y. Pack, K. W. Park, K. H. Pack, "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, 2011.

[6] S. I. Jang, S. K. Yoon, “Data Classification Management With its Interfacing Structure for Hybrid SLC/MLC PRAM Main Memory,” Jounal of the Computer Journal, Vol. 58, No. 11, pp. 2852 - 2863, 2015.

[7] S. Y. Lee, H. K. Bahn 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, 2014.

[8] X. Cai, L. Ju, M. Zhao, Z. Sun, Z. Jia, “A Novel Page Caching Policy for PCM and DRAM of Hybrid Memory Architecture,” Proceedings of 13th ICESS., pp.67-73, 2016.

[9] K. M Lee, J. H. Choi, J. W. Kwak, " WAP_LRU: Write Pattern Analysis Based Hybrid Disk Buffer Management in Flash Storage Systems," IEMEK J. Embed. Sys. Vol. 13, No. 3, pp.151-160, 2018 (in Korean).

[10] M. K. Qureshi, S. Vijayalakshmi, J. A. Rivers, “Scalable High Performance Main Memory System Using Phase-Change Memory Technology,” Proceedings of the 36th annual international symposium on Computer architecture, pp. 24-33, 2009.

[11] K. Y. Park, S. K. Yoon, S. D. Kim, “ Selective Data Buffering Module for Unified Hybrid Storage System,” Proceedings of 14th International Conference on Computer and Information Science, PP. 173-178, 2015.

[12] C. Chen J. An, “DRAM Write-only-cache for Improving Lifetime of Phase Change Memo ry,” Proceedings of International Midwest Symposium on Circuits and Systems, pp. 1-4, 2016.

[13] N. Nethercote, J. Seward, "Valgrind: A Program Supervision Framwork“, Elsevier Electonc Notes in Theoretical Computer Science, Vol. 89, No. 2, pp. 44-66, 2003.

[14] Micron Tech, “Phase Change Memory: A new Memory Technology to Enable new Memory Usage Models,” Available on: http://www.numonyx.comDocuments/WhitePapers/Numonyx_phaseChangeMemory_WhitePaper.pdf.

Bo-Sung Jung (정 보 성)

He received M.S. and Ph.D. degrees from GyeongSang National University in 2008 and 2018 respectively. His research interests include advance computer architecture and next generation memories system, and Non-volatile memory.

Email: blueking80@gnu.ac.kr

Jung-Hoon Lee (이 정 훈)

He received the M.S. and Ph.D. degree in Computer Science from Yonsei University, Seoul, Korea, in 2001 and 2004, respectively. He is currently a professor in GyeongSang National University (ERI). His research interests include advanced computer architectures and next flash memory.

Email: leejh@gnu.ac.kr