

논문 2016-11-13

압축 기반 상변화 메모리 시스템에서 저장 위치를 고려한 하이브리드 SLC/MLC 관리 기법

(Location-Aware Hybrid SLC/MLC Management for Compressed Phase-Change Memory Systems)

박재현, 이형규*

(Jaehyun Park, Hyung Gyu Lee)

Abstract: Density of Phase-Change Memory (PCM) devices has been doubled through the employment of multi-level cell (MLC) technology. However, this doubled-capacity comes in the expense of severe performance degradation, as compared to the conventional single-level cell (SLC) PCM. This negative effect on the performance of the MLC PCM detracts from the potential benefits of the MLC PCM. This paper introduces an efficient way of minimizing the performance degradation while maximizing the capacity benefits of the MLC PCM. To this end, we propose a location-aware hybrid management of SLC and MLC in compressed PCM main memory systems. Our trace-driven simulations using real application workloads demonstrate that the proposed technique enhances the performance and energy consumption by 45.1% and 46.5%, respectively, on the average, over the conventional technique that only uses a MLC PCM.

Keywords : Phase-change memory, Memory system, Compression, Hybrid, Location-aware

1. 서론

지난 수년간 DRAM 기술이 스케일링의 한계에 부딪치면서 여러 종류의 새로운 메모리 기술들이 제안되었다 [1, 2]. 새로 연구/개발 중인 여러 메모리 기술 중에서 상변화 메모리(PCM, Phase-Change Memory)는 나노미터 수준까지 작아질 수 있는 특성으로 많은 주목을 받고 있다 [2]. 상변화 메모리는 물질의 상태(state)에 따라 전기전도도의 차이가 존재하는 상변화 물질을 이용한 비휘발성 메모리이다. 상변화 메모리의 읽기 성능은 DRAM과 유사한 성능을 보이는 것에 반해 쓰기 성능은 매우 낮다 [2]. 따라서 쓰기 성능을 개선하기 위해 소자 수준

*Corresponding Author(hglee@daegu.ac.kr)

Received: 1 Feb. 2016, Revised: 9 Mar. 2016,

Accepted: 22 Mar. 2016.

J. Park: Arizona State University

H.G. Lee: Daegu University

※ 이 논문은 2013년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업 임(NRF-2013R1A1A2063350).

에서 시스템 수준까지 다양한 연구들이 이루어져 왔다 [3, 4].

최근 많은 연구들에서 Multi-Level Cell (MLC) 상변화 메모리도 가능함을 보였다 [3, 5]. 상변화 메모리에 MLC기술을 적용함으로써 상변화 메모리의 용량을 2배 또는 4배까지 증가시키는 것이 가능해졌다. 하지만, MLC 상변화 메모리는 기존의 Single-Level Cell (SLC) 상변화 메모리에 비해 응답시간(Latency)이 느리고 에너지 소모량이 많은 단점을 가지고 있다. 따라서 MLC 상변화 메모리의 장점인 큰 용량을 살리면서 성능을 보완할 방법이 필요하다.

MLC기반의 상변화 메모리라 할지라도 저장 공간이 충분할 경우 2비트를 저장할 수 있는 MLC의 공간에 SLC 모드처럼 1비트만 데이터만을 저장할 경우 일반적인 SLC 상변화 메모리와 같은 속도로 데이터를 읽고 쓸 수 있다 [6]. 이 경우 MLC 형태로 데이터를 읽고 쓰는 방식에 비해 수행시간과 에너지 소모량이 모두 감소한다.

MLC 상변화 메모리를 이용한 메인 메모리 시스템에서 데이터 저장 방식을 SLC 형태와 MLC 형태 중 선택함으로써 메모리 시스템의 특성을 개선하는

기법들이 제안된바 있다. 하지만 기존의 논문들은 사전에 목표대상(workload)에 대한 분석이 필요하거나 [7], 저장 방식을 선택하기 위해 메모리 사용량을 실시간으로 추적하기 때문에 구현에 많은 제약이 따른다 [6]. 또한 이러한 방법들은 동적으로 SLC 형태와 MLC 형태로 바꾸는 과정에서 메모리 주소가 변경되고 이로 인해 복잡한 주소 재변환 과정(address remapping)이 요구된다. 이러한 복잡한 주소 재변환 과정은 제안된 기법들의 효과에 부정적인 영향을 주기도 한다.

주소 재변환 과정을 피하기 위해 메인 메모리 접근단위인 캐쉬라인 단위로만 압축을 허용하며, 압축 결과에 따라 SLC 형태와 MLC 형태로 데이터를 저장하는 기법이 제안된 바 있다 [8]. 이 기법은 주소 재변환 과정이 필요 없다는 장점이 있지만, 제한된 정보만 이용하여 저장 방식에 대한 판단이 이루어지 때문에 효율이 떨어지는 단점이 있다. 또한 정해진 크기의 공간에 압축된 유효한 데이터만 기록하고 남은 공간은 사용하지 않는다.

본 논문에서 제안하는 기법은 압축 후 데이터를 기록하는 과정에서 인접 위치에 존재하는 유휴 공간을 활용함으로써 보다 적극적으로 데이터를 SLC 형태로 저장하고 이를 통해 메모리 시스템의 수행 시간과 에너지 소모량을 감소시킨다. 본 기법은 상변화 메모리의 제어기만 변경하여 구현이 가능하며 사용하고자 하는 응용에 대한 사전 분석이나 복잡한 주소 재변환 과정이 필요하지 않다.

제안된 기법의 효율성을 보여주기 위해 실제 응용에서 추출한 메모리 트레이스(Memory Trace)를 이용하여 시뮬레이션을 실시한 결과 압축만 적용한 MLC 기반 상변화 메모리 시스템 대비 수행 시간은 평균 45.1%, 에너지 소모량은 평균 46.5% 감소하였다.

본 논문의 구성은 II장에서 먼저 MLC 상변화 메모리에 대한 기초와 본 논문에서 제안한 하이브리드 SLC/MLC 관리 기법에 대해 설명한다. 이후 실험을 통해 제안한 기법의 효과를 기술하고 분석한 후 III장에서 결론을 맺는다.

II. 본 론

1. MLC 상변화 메모리에 대한 기초

상변화 메모리는 가열 온도와 냉각 방식에 따라 비결정질(Amorphous) 상태와 결정질(Crystalline) 상태의 두 가지 상태를 가지며 각 상태간의 저항

표 1. SLC와 MLC 상변화 메모리의 특성 비교

Table 1. Comparison of SLC and MLC PCM parameters

	SLC	MLC
Read latency	10 ns	44 ns
Write latency	100 ns	395 ns

값의 차이를 이용하여 0과 1을 표현하는 메모리 소자이다. 비결정질 상태의 저항 값은 결정질 상태의 저항 값에 비해 일반적으로 1,000~10,000배 더 큰 값을 갖는다 [9]. 비결정질 상태와 결정질 상태의 큰 저항 값 차이를 이용하여 1개의 상변화 메모리 소자에 여러 비트의 데이터를 저장하는 MLC 상변화 메모리 소자 기술도 개발되었다.

표 1은 본 논문에서 사용한 SLC 상변화 메모리와 MLC(2비트 MLC) 상변화 메모리의 특성을 보여주고 있다 [6]. MLC 상변화 메모리의 경우 데이터를 기록할 때 SLC 상변화 메모리에 비해 정밀한 저항 값 조절이 필요하다. 이를 위한 기법으로 기록 후 확인(write-and-verify, WAV) 기법이 잘 알려져 있다 [5]. 이 기법은 저항 값을 임의의 단위로 변화시킨 후 목표저항 값에 도달하였는지를 확인한다. 목표저항 값에 도달하지 못했다면 동일한 과정을 반복한다. 1개의 메모리 소자로 표현하고자 하는 데이터 비트의 수가 많아질수록 더욱 정밀한 제어가 필요하며 이에 따라 WAV 반복 횟수가 증가한다. 따라서 쓰기에 소요되는 시간과 에너지 소모 또한 증가하게 된다. MLC 메모리 소자에 저장된 데이터를 읽는 과정도 SLC 메모리 소자에 저장된 데이터를 읽는 과정에 비해 더 많은 비교 과정을 거치기 때문에 더 많은 시간과 에너지가 소모된다.

2. 압축 기반의 동적 SLC/MLC PCM 관리 기법

상변화 메모리는 기존의 DRAM에 비해 동작속도가 느리며 전력 문제로 인해 동시에 쓸 수 있는 비트 수가 제한되어 있어 쓰기에 많은 시간에 소요된다 [4]. 따라서 압축을 통해 읽고 쓰는 데이터의 양을 줄이는 것은 상변화 메모리 시스템의 성능을 향상시키는데 큰 도움이 된다.

앞에서 기술하였듯이 SLC 상변화 메모리는 성능이 좋은 반면 동일한 공간에 저장할 수 있는 데이터양은 MLC 상변화 메모리에 비해 적다. 반대로 MLC 상변화 메모리는 동일한 물리적 공간에 더 많은 데이터를 저장할 수 있는 대신 성능 및 전력소모 특성이 좋지 않다.

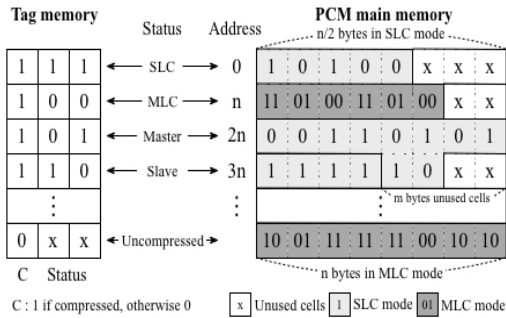


그림 1. 제안된 저장위치를 고려한 하이브리드 SLC/MLC 관리기법

Fig. 1 The proposed hybrid SLC/MLC management

두 가지 형태의 상변화 메모리의 장점만을 취하기 위하여 압축 기법을 활용하고 특정 MLC 상변화 메모리를 SLC 상변화 메모리처럼 활용하여 SLC 상변화 메모리의 성능과 MLC 상변화 메모리의 저장 용량을 동시에 가진 메모리 시스템 관리 기법이 제안된 바 있다 [8]. 본 논문에서는 [8]에서 제시된 기법을 기반으로 저장 위치를 고려하고 압축과정에서 발생하는 인접 메모리 주소의 유휴 공간을 이용하여 상변화 메모리 시스템의 성능을 더욱 향상시키는 기법을 제안하고자 한다.

2.1 동적 SLC/MLC PCM 재구성

그림 1은 본 논문에서 제안하는 기법의 상위 수준의 동작 방식을 보여주고 있다. 그림 1에서 표현된 것과 같이 메인 메모리는 n 바이트의 기본 단위로 하는 MLC 상변화 메모리를 기준으로 주소가 부여되어 있다고 가정한다. n 바이트는 메인 메모리의 기본 접근 단위인 최하위 계층 캐쉬(Last-level cache, LLC)의 캐쉬라인 크기와 동일하다. 본 논문에서는 1개의 상변화 메모리 소자를 이용하여 2비트를 표현하는 2비트 MLC 상변화 메모리를 가정하였다. 하지만 본 논문에서 제안한 기법은 3비트 이상을 표현하는 MLC(또는 TLC라고도 불림)상변화 메모리에도 적용 가능하다.

만약 n 바이트의 데이터를 압축하여 n/2 바이트보다 작게 만들 수 있다면 해당 데이터는 같은 주소에 MLC 모드 대신 SLC 모드로 저장할 수 있다. SLC 모드로 데이터를 저장한다면 쓰기 과정에서 소요되는 시간과 에너지 소모를 줄일 수 있으며 읽기 과정에서의 시간과 에너지 소모도 줄어든다. 데이터 크기가 반 이하로 줄었기 때문에 SLC 모드로 저장할

다고 하더라도 MLC 모드로 저장할 때와 같은 수의 상변화 메모리 소자를 이용하여 데이터를 저장할 수 있으며 주소 재변환 과정도 필요하지 않다.

압축 후 데이터의 크기가 n/2 바이트보다 클 경우 인접된 메모리 주소의 유휴 공간을 활용하여 SLC 모드로 저장할 수 있는지 판단한다. 그림 1에서 메모리 주소 2n과 같이 인접된 메모리 주소의 공간 (n 또는 3n)이 SLC 모드로 관리되고 있으며, MLC 기준 m 바이트의 유휴 공간이 존재할 경우, 저장할 데이터의 압축 후 크기가 (n+m)/2 바이트보다 작으면 해당 인접 유휴공간까지 사용하여 해당 데이터를 SLC 모드로 저장할 수 있다.

만약 데이터의 압축 후 크기가 (n+m)/2 바이트보다 크다면 MLC 모드로 데이터를 저장하며 압축이 되지 않는 경우를 따로 구분하여 읽기 시 압축 해제 과정을 생략함으로써 시간과 에너지 소모를 줄인다.

데이터를 효과적으로 관리하기 위해 본 논문에서는 다음과 같이 5가지 상태(state)를 정의하고 이를 3비트의 태그 메모리(Tag Memory)를 사용하여 관리한다.

- 1) SLC 상태: 태그 메모리에 “111”로 저장, n 바이트의 데이터를 압축 후의 크기가 n/2 바이트보다 작아 데이터를 SLC 모드로 저장한 상태를 나타내며, 인접한 주소와 물리적 공간을 공유하지 않는다.
- 2) 인접 단위 마스터(Master) 상태: 태그 메모리에 “101”로 저장, 압축 후 데이터 크기가 n/2 바이트보다 크지만 인접 주소의 유휴 공간을 활용하여 데이터를 SLC 모드로 저장한 상태를 나타낸다. 인접 단위 마스터로 저장된 데이터를 복원하기 위해서는 인접 단위까지 추가로 읽어야 복원이 가능하며 이로 인한 부하가 발생한다.
- 3) 인접 단위 슬레이브(Slave) 상태: 태그 메모리에 “110”로 저장, 데이터를 SLC 모드 저장하는 과정에서 발생한 유휴 공간을 인접 단위 마스터 상태에 빌려준 상태를 표시한다. 인접 단위 마스터 상태와 달리 데이터를 복원하기 위해 인접 단위를 추가로 읽지 않는다. 다만 해당 위치에 데이터 크기가 증가한 새로운 데이터를 기록할 때 인접 단위 마스터 상태를 MLC 상태로 변경해야 하는 경우가 발생할 수 있다.
- 4) MLC 상태: 태그 메모리에 “100”로 저장, 데이터 압축은 가능하지만 압축 후 데이터 크기가 n/2 바이트보다 큰 경우를 나타내며 데이터 읽기 시 복원 과정이 필요하다.

5) 비압축(Uncompressed) 상태: 태그 메모리에 "000"로 저장, 데이터 압축이 이루어지지 않는 경우를 나타하며 읽기 시 복원 과정이 없이 데이터를 읽는다.

그림 1에서와 같이 5가지 상태 중 SLC 상태와 인접 단위 마스터 상태, 인접 단위 슬레이브 상태는 상변화 메모리를 SLC 모드로 이용하고 MLC 상태와 비압축 상태는 MLC 모드로 이용한다.

본 논문에서 제안한 기법은 n바이트로 구성된 메모리 기본 접근 단위를 기반으로 동작하기 때문에 복잡한 주소 재변환 과정 없이 SLC 상변화 메모리와 MLC 상변화 메모리의 장점을 동시에 갖춘 메인 메모리 시스템을 가능하게 한다.

2.2 압축 알고리즘

본 논문에서 제안되는 기법에 적용되는 압축 알고리즘은 무손실 압축이어야 한다. 압축이 잘 될수록 더 많은 데이터를 SLC 모드로 저장할 수 있어 메모리 시스템의 성능이 높아진다. 하지만 압축이 잘 되는 압축 알고리즘의 경우 압축 및 압축복원에 추가적인 시간손실이 따른다. 압축은 쓰기과정에서 일어나며 이때 발생하는 시간손실은 쓰기 버퍼(Write Buffer)를 사용함으로써 최소화 할 수 있다. 이에 반해 읽기 과정에서 발생하는 압축복원에 따른 시간 손실은 압축복원이 완료된 후에만 캐쉬에 데이터를 제공할 수 있으므로 메모리 시스템의 성능에 직접적으로 영향을 미치게 된다. 따라서 적절한 압축률과 함께 압축 및 압축복원에 따른 시간손실을 최소화 할 수 있는 압축 알고리즘이 선택이 필요하다.

본 논문에서는 캐쉬와 메인 메모리에서 많이 사용되는 압축 알고리즘 중 자주 사용되는 패턴을 이용한 압축 알고리즘(FPC, Frequent pattern compression)을 사용한다 [10]. FPC 알고리즘은 자주 사용되는 패턴에 대한 표를 만든 후 이를 이용하여 압축 및 압축복원을 진행하므로 비교적 적은 시간 손실 및 구현 부하를 발생시킨다. 본 논문에서는 메모리 접근의 기본 단위인 캐쉬라인 단위로 FPC 알고리즘을 적용하였다. 이 경우 64 바이트를 최대 8 바이트까지 압축할 수 있다. 그림 2는 본 실험에서 사용한 트레이스의 압축률을 보여주고 있다. 압축률은 압축 후 데이터 크기를 압축 전 데이터 크기로 나눈 값으로 값이 작을수록 압축이 잘 됨을 나타낸다. 본 논문에서 사용한 트레이스 중 facesim의 압축률이 다른 응용에 비해 높은 편(압축이 잘되지 않는 것을 의미)이며 평균 0.41의 압축률을 보이고 있다. FPC

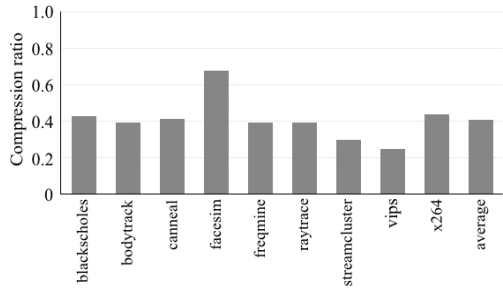


그림 2. 각 트레이스별 압축률 (높이가 낮을수록 압축이 잘되는 것을 의미)
Fig. 2 Compression ratio (lower is better)

표 2. FPC의 시간 손실 및 구현 부하
Table 2. Timing and implementation overheads of FPC

Overhead		FPC
Timing (cycles)	Compression	3
	Decompression	5
Area (gate count)		8,758
Power (mW)		4.07

알고리즘의 시간 손실 및 구현 부하는 표 2에 제시되어 있다. 공정한 비교를 위해 본 논문에서는 FPC 알고리즘의 압축 및 압축복원에 따른 시간손실 및 전력 소모를 포함하는 모든 부하를 고려하였다.

2.3 구조 및 구현

본 논문에서 제안한 기법을 실현하기 위한 기본 구성요소인 압축 및 압축복원 모듈, SLC/MLC 관리를 위한 태그 메모리는 메모리 제어기 내에 구현할 수 있다. 압축 및 압축복원 모듈이 상변화 메모리 칩 안에 구현된다면 여러 개의 상변화 메모리 칩을 이용할 때 자원이 중복되는 문제가 발생한다. 이를 메모리 제어기 내에 구현함으로써 불필요한 자원의 낭비를 막을 수 있다. 또한 태그 메모리가 상변화 메모리 칩 안에 구현된다면 항상 정해진 위치에 기록되어야 하고 이는 쓰기 횟수가 정해진 상변화 메모리의 수명에 문제가 된다.

본 기법을 지원하기 위해 추가적으로 필요한 태그 메모리의 용량은 메인 메모리의 용량과 압축에 사용되는 기본 단위 크기에 의해 결정된다. 본 논문에서는 총 4 GB의 메인 메모리를 가정하였으며 압축의 단위는 캐쉬라인의 크기인 64 바이트로 가정하였다. 이 경우 태그 메모리를 구현하기 위해 24 MB의 메

```

1: if c flag is set in tag[addr] then
2:   if status is MLC then
3:     read target data in MLC mode
4:   else if status is MASTER then
5:     read target and connected data in SLC mode
6:   else
7:     read target data in SLC mode
8:   end if
9:   decompress read data
10: else
11:   read target data in MLC mode
12: end if
13: store data to read buffer

```

알고리즘 1. 읽기 알고리즘

Algorithm 1. Algorithm of the read operation

모리가 추가적으로 필요지만 메모리 접근의 기본 단위 크기가 커질 경우 태그메모리에 필요한 용량은 감소한다.

상변화 메모리는 읽고 쓸 때 SLC 모드와 MLC 모드로 동적으로 변경하는 것을 지원하여야 한다. 이를 위해 본 논문에서는 기존 논문 [6]에서와 같이 쓰기 과정에서 SLC 모드와 MLC 모드로 SET 동작, RESET 동작, 부분 SET 동작을 지원하고 dual mode sense amplifier를 이용하여 SLC 모드와 MLC 모드의 읽기가 가능한 상변화 메모리를 가정하였다.

알고리즘 1과 2는 읽기와 쓰기 과정에서 수행되는 알고리즘을 보여준다.

메인 메모리에 읽기 요청이 도착하면 우선 태그 메모리에서 C 플래그의 값을 확인한다(1행). C 플래그는 해당 주소의 데이터가 압축되었는지 아닌지를 나타낸다. 압축되지 않은 데이터라면 MLC 모드로 읽은 후 복원 과정을 거치지 않고 읽기 버퍼에 저장한다(11행). 반면 압축된 데이터인 경우 상태 비트의 값이 MLC 상태라면 MLC 모드로 데이터를 읽고(3행) 그 외의 경우에는 SLC 모드로 읽은 후(5행, 7행) 압축복원 과정을 거친다(9행). 이 때 만약 상태 값이 인접 단위 마스터 상태라면 복원을 위해 필요한 데이터의 일부가 인접 주소에 저장되어 있으므로 상태 값이 인접 단위 슬레이브 상태에 해당하는 주소의 데이터도 읽어(5행) 복원 과정을 거친다.

쓰기 과정의 경우 저장할 데이터의 압축률과 해당 주소에 저장된 이전 데이터의 상태 값에 의해 쓰기 과정이 다르게 이루어진다. 데이터가 도착하면 압축이 이루어진다(1행).

```

1: Compress data
2: if c flag is set and status is MASTER then
3:   if  $S_{target} < n/2$  then
4:     change status of related SLAVE as SLC
5:     write data in SLC mode
6:   else if  $S_{target} + S_{neighbor} < n$  then
7:     write data in SLC mode
8:   else
9:     change status of related SLAVE as SLC
10:    write data in MLC mode
11:   end if
12: else if c flag is set and status is SLAVE then
13:   if  $S_{target} + S_{MASTER} < n$  then
14:     write data in SLC mode
15:   else
16:     rewrite data of related MASTER in MLC mode
17:   if  $S_{target} < n/2$  or  $S_{target} + S_{neighbor} < n$  then
18:     write data in SLC mode
19:   else
20:     write data in MLC mode
21:   end if
22: end if
23: else
24:   if  $S_{target} < n/2$  or  $S_{target} + S_{neighbor} < n$  then
25:     write data in SLC mode
26:   else
27:     write data in MLC mode
28:   end if
29: end if

```

알고리즘 2. 쓰기 알고리즘

Algorithm 2. Algorithm of the write operation

이전 상태 값이 인접 단위 마스터 상태(2행)이며 새로운 데이터의 압축 후 크기가 $n/2$ 바이트보다 작다면(3행) 해당 데이터는 SLC 모드로 기록된다. 이 경우와 인접한 주소의 유휴공간을 이용하여도 SLC 모드로 저장할 수 없을 경우(8행) 인접 주소의 유휴 공간을 더 이상 사용하지 않기 때문에 인접 주소의 상태 값이 인접 단위 슬레이브 상태에서 SLC 상태로 변경된다(4행, 9행). 이 과정에서는 데이터 변경 없이 태그 메모리 내의 상태 값 변경만 이루어진다. 데이터 저장 모드는 데이터의 압축 여부와 압축 후 크기에 따라 SLC 모드(5행, 7행) 또는 MLC 모드(10행)로 데이터를 기록한다.

이전 상태 값이 인접 단위 슬레이브 상태(12행)인 위치에 데이터를 기록할 경우 먼저 데이터 압축 후 크기와 연결된 인접 단위 마스터 상태의 데이터 크기에 따라 인접 단위 슬레이브 상태를 유지할지 판단한다(13행). 새로 발생한 유휴 공간이 충분하지 않다면 연결된 인접 단위 마스터 상태의 데이터를 MLC 모드로 다시 쓰고 상태 값도 MLC 상태로 변경한다(16행). 이후 데이터의 압축 여부와 압축 후 크기에 따라 SLC 모드(14행, 18행) 또는 MLC 모드(20행)로 데이터를 기록한다.

이전 상태 값이 인접 단위 마스터 상태나 인접 단위 슬레이브 상태 외의 경우(23행) 압축 후의 크기가 $n/2$ 바이트보다 작거나 인접한 주소의 유휴공간을 이용해 SLC 모드로 기록할 수 있다면(24행) SLC 모드로 기록하고(25행) 아니면 MLC 모드로 기록한다(27행).

인접한 주소에 존재하는 유휴 공간의 크기를 판단하기 위해서는 해당 데이터를 읽은 뒤 복원하는 과정을 거쳐야한다. 쓰기 과정은 쓰기 버퍼를 통해 이루어지기 때문에 이와 같은 추가 과정이 시간 부하를 발생시키지는 않지만 에너지 부하를 발생시킨다. 본 논문에서는 위와 같이 읽기 및 쓰기과정에서 발생하는 부하를 모두 고려하였다.

3. 실험

3.1 실험 환경

본 논문에서 제안한 기법을 평가하기 위하여 메모리 트레이스(Memory Trace) 기반의 시뮬레이터를 개발하였다. 본 실험에서 사용된 메모리 트레이스는 Simics라는 시스템 수준의 시뮬레이터에서 실제 응용 프로그램을 수행하면서 추출하였다 [11]. 시뮬레이션 대상 시스템의 자세한 구성은 표 3에 제시되어 있으며 사용된 메모리 트레이스들은 PARSEC 벤치마크 [12]에 포함된 여러 개의 응용 프로그램을 이용하여 추출하였다.

시스템에 대한 평가는 3가지 메인 메모리 시스템 구조에 대해 이루어졌다. 첫 번째 메모리 시스템 구조(“Compressed MLC_only”로 표시)는 MLC 모드로만 상변화 메모리를 사용하며 데이터 저장 시 압축을 활용한다. 각 관리 기법의 효과를 비교하기 위하여 이 시스템의 설정을 비교의 기준(Baseline)으로 활용하였다. 두 번째 메모리 시스템 구조(“Intra-cacheline”로 표시)는 압축 후 데이터 크기가 절반 이하로 줄어든다면 SLC 모드로 데이터를 저

표 3. 시뮬레이션 시스템의 구성
Table 3. Simulated system parameters

No. of CPU cores	4
CPU core type	UltraSPARC-III+, 2 GHz
L1 caches (private)	I- and D-cache 32 KB, 4-way, 64 B
L2 cache (Shared)	2 MB, 4-way, 64 B
Main memory	4 GB PCM

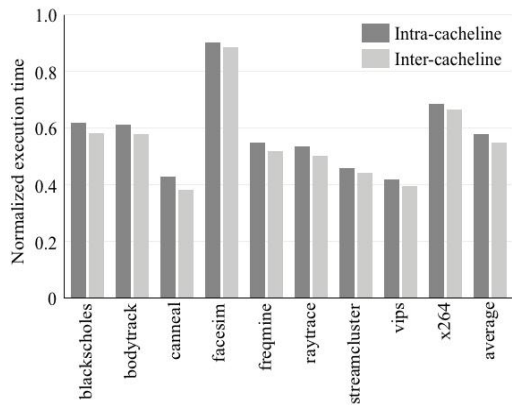


그림 3. 수행시간 비교

Fig. 3 Comparison of execution time (normalized to compressed MLC_only)

장한다 [8]. 이 관리기법은 데이터의 저장 위치를 고려하지 않는다. 세 번째 시스템 구조(“Inter-cacheline”로 표시)는 본 논문에서 제안한 메모리 시스템 구조로 데이터의 저장 위치를 고려하여 인접한 단위의 유휴 공간까지 활용함으로써 SLC 모드로 데이터를 저장하는 비율을 최대한으로 높일 수 있는 구조이다.

상변화 메인 메모리를 SLC와 MLC 형태로 읽고 쓰는데 소요되는 시간은 표 1에 제시한 값을 사용하였으며 압축 알고리즘으로 앞에서 제시한 바와 같이 FPC 알고리즘을 사용하였다.

3.2 실험결과

먼저 그림 3과 같이 각 시스템 구성에 따른 수행시간의 변화를 평가하였다. 각 시스템의 수행 시간은 비교기준인 “Compressed MLC_only”의 수행시간을 기준으로 정규화(Normalize)하여 표현하였다.

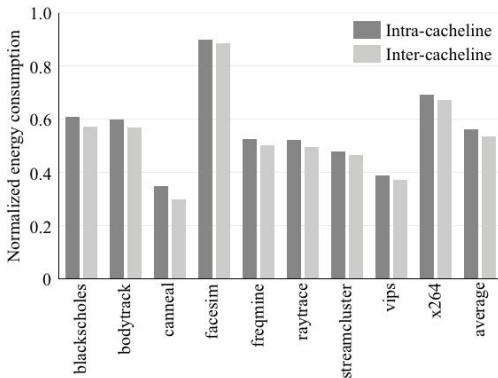


그림 4. 에너지 소모량 비교

Fig. 4 Comparison of energy consumption (Normalized to compressed MLC_only)

진한 회색의 막대그래프는 “Intra-cacheline”의 수행시간을 나타내며 연한 회색의 막대 그래프는 “Inter-cacheline”의 수행 시간을 나타낸다.

“Intra-cacheline”은 비교기준과 비교하여 수행 시간을 평균 42.2% 줄일 수 있는 반면 본 논문에서 제안한 “Inter-cacheline”은 평균 45.1% 줄일 수 있다. 각 응용별 수행시간을 살펴보면 그림 2에서 제시된 압축률과 수행시간이 비례한 것을 볼 수 있다. 압축률이 클수록 SLC 모드로 데이터를 저장할 수 있는 확률이 줄어들어 수행시간을 감소시킬 수 없기 때문이다. 여러 응용 중에서 facesim과 x264은 압축률에 비해 수행시간의 감소폭이 적고 canneal의 경우 반대의 경우를 보이는데 이에 대해서는 뒤에서 자세하게 분석하도록 하겠다.

다음으로 그림 4와 같이 각 메모리 시스템의 에너지 사용량을 비교하였다. 그림 3과 동일하게 “Intra-cacheline”의 에너지 소모량은 짙은 회색으로 표시하였으며 비교기준 시스템의 에너지 소모량에 비해 평균 44.0% 적은 에너지 소모량을 보인다. 이에 비해 연한 회색으로 표시된 본 논문에서 제안한 “Inter-cacheline”은 평균 46.5%의 에너지 소모 감소를 보여준다. 에너지 소모량은 수행시간과 비슷한 경향성을 보여준다. 기존 논문과 비교하여, 본 논문에서 제안한 기법의 추가 에너지 감소량은 추가 수행 시간 감소량보다 적는데 이는 인접 주소의 유희 공간을 활용하는 과정에서 발생하는 추가 메모리 접근 중 SLC 형태로 저장된 데이터를 MLC 형태로 바꿔 쓰는 동작 같이 일부 동작이 추가적인 수행시간은 요구하지 않지만 추가적인 에너지 소모하기 때문으로 분석된다.

3.3 실험결과 분석

실험결과를 분석하기 위하여 그림 5와 같이 각각의 응용을 수행하는 과정에서 발생하는 읽기와 쓰기 동작의 횟수를 분석하였다. 3개의 누적 막대 그래프 중 첫 번째 막대그래프는 “Compressed MLC_only”의 읽기 및 쓰기 횟수의 분포를 보여주며 두 번째 막대그래프는 “Intra-cacheline”의 읽기 및 쓰기 횟수의 분포, 세 번째 막대그래프는 본 논문에서 제시한 “Inter-cacheline”의 읽기 및 쓰기 횟수의 분포를 보여주고 있다. 각 시스템의 읽기 및 쓰기 횟수는 비교기준인 “Compressed MLC_only”의 총 읽기와 쓰기 횟수를 이용하여 정규화 하였다.

먼저 각 트레이스별 메모리 접근의 특성을 분석해보면 facesim과 x264의 경우 전체 메모리 접근 중 압축이 되지 않는 데이터에 대한 읽기 및 쓰기 접근이 차지하는 비중이 큰데 반해 canneal은 해당 비중이 작다. 압축되지 않는 데이터에 대한 접근은 수행 시간과 에너지 소모를 감소시킬 수 없으므로 그림 3과 그림 4에서 살펴본 바와 같이 facesim과 x264의 경우 압축률에 비해 수행시간과 에너지 소모량의 감소량이 적은 반면, canneal의 경우 압축률 대비 수행 시간 및 에너지 소모 감소량이 많은 것을 볼 수 있다.

각 시스템별 읽기 및 쓰기 횟수 분포를 살펴보면 “Compressed MLC_only”의 경우 상변화 메인 메모리를 MLC 모드로만 사용하기 때문에 메모리 접근이 압축된 MLC 형태의 읽기 및 쓰기와 압축이 되지 않은 MLC 형태의 읽기 및 쓰기로만 이루어져 있다.

“Intra-cacheline”의 경우 압축 후 데이터의 크기가 압축 전 데이터 크기의 절반 이하일 경우 SLC 형태로 데이터를 기록하기 때문에 압축된 MLC 형태의 메모리 접근 중 상당 부분이 SLC 형태의 읽기 및 쓰기로 변환되어 시스템의 수행시간과 에너지 소모를 줄인다. 메모리 관리 시 기본 메모리 접근 단위인 캐쉬라인 내의 정보만을 사용하기 때문에 메모리 접근 횟수는 비교기준인 “Compressed MLC_only”의 메모리 접근 횟수와 동일하다.

이에 비해 본 논문에서 제안한 저장 위치를 고려한 하이브리드 SLC/MLC 관리기법이 적용된 “Intra-cacheline”의 경우 읽기 과정에서 해당 주소의 상태 값이 인접 단위 마스터 상태인 경우 인접 단위 슬레이브 상태에 저장된 데이터를 추가로 읽어야만 복원이 가능하며 쓰기 과정에서도 인접 위치에 저장된 데이터를 확인하는 과정에서 추가적인 읽기가 필요하다. 이러한 추가 읽기 동작은 수행시간에 증가를 가져온다. 쓰기 과정에서 데이터의 압축 후 크기가

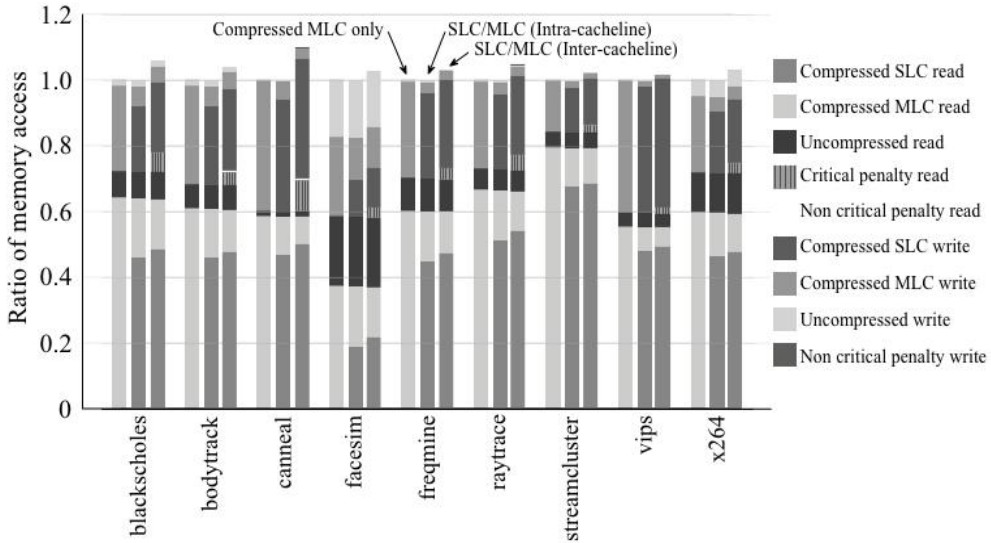


그림 5. 기존 시스템의 메모리 접근 횟수에 정규화된 메모리 접근의 종류별 비율

Fig. 5 Breakdown of the type of memory access (normalized to the compressed MLC only)

의 변화에 따라 SLC 형태로 저장된 데이터를 MLC 형태로 변경하여 저장해야 할 경우가 발생하기도 한다. 이 과정에서 발생하는 읽기와 쓰기는 수행 시간의 요인은 아니지만 에너지 소모 증가의 주요 요인이 되기도 한다.

위와 같이 저장 위치를 고려하여 SLC/MLC 관리를 수행하는 과정에서 추가적인 읽기 및 쓰기 동작이 발생하여 전체 메모리 접근 횟수는 다른 시스템의 메모리 접근 횟수에 비해 증가한 것을 확인할 수 있다. 하지만 더 많은 데이터를 접근 속도가 빠르고 에너지 소모가 적은 SLC 형태로 저장할 수 있어 전체 메모리 접근 횟수 중 SLC 형태의 메모리 접근 횟수가 차지하는 비중이 증가한다. 이로 인해 결과적으로 시스템 전체의 수행시간과 에너지 소모가 줄어드는 것을 확인할 수 있다.

III. 결론

상변화 메모리는 MLC 기술로 인해 저장 용량을 2배 이상 늘리는 것이 가능해졌지만 이 경우 SLC 상변화 메모리에 비해 응답시간과 에너지 사용량이 늘어나는 단점이 있다. 본 논문에서는 압축 기반의 상변화 메모리 시스템에서 저장 위치를 고려한 SLC/MLC 하이브리드 관리 기법을 제안하였다. 본 논문에서 제안한 기법은 큰 추가 부하 없이 시스템의 수행 시간과 에너지 소모량을 감소시켰으며 실

제 응용들에서 추출한 트레이스를 이용한 트레이스 기반 시뮬레이션 결과 하이브리드 SLC/MLC 관리 기법 없이 압축만 적용한 시스템 대비 수행 시간은 평균 45.1%, 에너지 소모량은 평균 46.5% 감소하였다.

References

- [1] H.G. Lee, C. Lee, "Enhancing Flash Memory-based Storage Using Next Generation Non-Volatile Memory Devices," Communications of the Korea Information Science Society, Vol. 27, No. 5, pp. 47-57, 2009 (in Korean).
- [2] R.F. Freitas, W.W. Wilcke, "Storage-class memory: The next storage system technology." IBM Journal of Research and Development, Vol. 52, No. 4/5, pp. 439-447, 2008.
- [3] T. Nirschl, J.B. Phipp, T.D. Happ, G.W. Burr, B. Rajendran, M.H. Lee, A. Schrott, M. Yang, M. Breitwisch, C.F. Chen, E. Joseph, M. Lamorey, R. Cheek, S.H. Chen, S. Zaidi, S. Raoux, Y.C. Chen, Y. Zhu, R. Bergmann, H.L. Lung, C. Lam, "Write strategies for 2 and 4-bit multi-level phase-change memory,"

- Proceedings of the IEEE International Electron Device Meeting, pp. 461-464, 2007.
- [4] Y. Choi, I. Song, M.H. Park, H. Chung, S. Chang, B. Cho, J. Kim, Y. Oh, D. Kwon, S. Jung, J. Shin, Y. Rho, C. Lee, M.G. Kang, J. Lee, Y. Kwon, S. Kim, J. Kim, Y.J. Lee, Q. Wang, S. Cha, S. Ahn, H. Horii, J. Lee, K. Kim, H. Joo, K. Lee, Y.T. Lee, J. Yoo, G. Jeong "A 20nm 1.8V 8Gb PRAM with 40MB/s program bandwidth," Proceedings of IEEE International Solid-State Circuits Conference, pp. 46-48, 2012.
- [5] D.H. Kang, J.H. Lee, J.H. Kong, D. Ha, J. Yu, C.Y. Um, J.H. Park, F. Yeung, J.H. Kim, W.I. Park, Y.J. Jeon, M.K. Lee, J.H. Park, Y.J. Song, J.H. Oh, G.T. Jeong, H.S. Jeong, "Two-bit cell operation in diode-switch phase change memory cells with 90nm technology," Proceedings of the Symposium on VLSI Technology, pp. 10-12, 2008.
- [6] X. Dong, Y. Xie, "Adams: adaptive MLC/SLC phase-change memory design for file storage," Proceedings of Asia and South Pacific Design Automation Conference, pp. 31-36, 2011.
- [7] M.K. Qureshi, M.M. Franceschini, L.A. Lastras-Montaño, "Morphable memory system: a roust architecture for exploiting multi-level phase change memories," Proceedings of the 37th annual international Symposium on Computer Architecture, pp. 154-162, 2010.
- [8] H.G. Lee, S. Baek, J. Kim, "A compression-based hybrid MLC/SLC management technique for phase-change memory systems," Proceedings of IEEE Computer Society Annual Symposium on VLSI, pp. 386-391, 2012.
- [9] S. Raoux, G.W. Burr, M.J. Breitwisch, C.T. Rettner, Y.C. Chen, R.M. Shelby, M. Salinga, D. Krebs, S.H. Chen, H.L. Lung, C.H. Lam, "Phase-change random access memory: a scalable technology," IBM Journal of Research and Development, Vol. 52, No. 4/5, pp. 465-479, 2008.
- [10] A.R. Alameldeen, D.A. Wood, "Frequent pattern compression: a significant-based compression scheme for L2 caches," Technical Report 1500, Computer Sciences Department, 2004.
- [11] P.S. Magnusson, M. Christensson, J. Eskilson, D. Forsgren, G. Hallberg, J. Hogberg, F. Larsson, A. Moestedt, B. Werner, "Simics: a full system simulation platform," Computer, Vol. 35, No. 2, pp. 50-58, 2002.
- [12] C. Bienia, S. Kumar, J. P. Singh, K. Li, "The PARSEC benchmark suite: characterization and architecture implications," Proceedings of the 17th International Conference on Parallel Architectures and Compilation Technique, pp. 50-58, 2002.

Jaehyun Park (박재현)



Jaehyun Park received the B.S. degree in Electrical Engineering and Ph.D degree in Electrical Engineering and Computer Science from Seoul National University, in 2006 and 2015, respectively. He was joining Ming Hsieh Department of Electrical Engineering at the University of Southern California as a Visiting Scholar from 2009 to 2010. Currently he is an Exchange Scholar with the School of Electrical, Computer and Energy Engineering from 2015. His research interests include low-power embedded system and non-volatile memory system.

Email: jaehyun.park.2@asu.edu

Hyung Gyu Lee (이형규)



Hyung Gyu Lee received the M.S and Ph.D degree from the School of Computer Science and Engineering, Seoul National University in 2001 and

2007, respectively. He was a Senior Engineer with Samsung Electronics from 2007 to 2010 and a Post-Doctoral Research Fellow with the Georgia Institute of Technology, Atlanta from 2010 to 2012. Currently he is an assistant professor with the School of Computer and Communication Engineering, Daegu University /Information and Communication Research Center, Korea. His current research interests include low power system, Flash-based storage system, and energy harvesting technology.

Email: hglee@daegu.ac.kr