

# 차세대 비휘발성 메모리를 활용한 플래시 메모리 기반 스토리지의 성능 향상 기법

삼성전자 | 이형규 · 이 철

## 1. 서론

반도체 기술의 향상과 함께 휴대용 기기에서도 데스크탑 이상의 시스템에서만 실행 가능했던 멀티미디어 응용과 같은 복잡한 응용의 실행이 가능하게 됨에 따라서 휴대용 기기라 할지라도 고성능/대용량의 메모리/스토리지를 사용하게 되었다. 따라서 기존 디스크기반 스토리지와 비교해서 상대적으로 높은 성능, 적은 전력을 소모할 뿐 아니라 차지하는 공간이 작으면서도 안정성이 뛰어나 휴대용 기기에 적합한 플래시 메모리(flash memory) 기반의 스토리지(storage)가 점차 널리 사용되기 시작하였다.

특히 낸드플래시 메모리(NAND Flash memory)는 노어플래시 메모리(NOR Flash memory)등의 다른 메모리와 비교하여 셀(cell)들간의 연결 구조가 간단하기 때문에 보다 많은 셀을 단일 칩 내에 집적시킬 수 있기 때문에 저비용으로도 대용량화가 가능하면서도 메모리형태로 되어 있기 때문에 데이터가 저장된 위치와 상관없이 접근 시간이 일정한 장치라는 장점이 있다. 기본적으로 낸드플래시 메모리칩은 2KB 혹은 4KB로 이루어지는 페이지(page), 64-128개의 페이지로 구성된 블록(block) 그리고 이러한 블록이 다수개가 모여 구성된다.

그러나 낸드플래시는 근본적으로 부동 게이트(floating gate)에 전자를 주입하거나 빼내는 원리로 동작되는 플래시 메모리 기술에 기반을 두었을 뿐 아니라 메모리 셀들은 고 용량을 위해 직렬로 연결하는 형태를 취하고 있기 때문에 이를 사용하는데 있어서 다음과 같은 제약들이 존재한다.

- 쓰기 전 소거 제약(erase before program)
- 읽기/쓰기동작의 기본 단위와 소거동작의 기본 단위가 다름 - 읽기/쓰기: 2-4KB, 소거: 128-256KB
- 내구성(endurance) - 제한된 쓰기/소거 회수

### · 순차적 쓰기 제약

이러한 제약들은 낸드플래시 메모리를 이용한 스토리지를 설계를 어렵게 하고 있다. 특히 쓰기 동작(program operation)전에 반드시 쓰고자 하는 메모리 셀을 소거(erase)해야 하는 쓰기 전 소거제약(erase-before-program constraints)과 읽기/쓰기 동작의 기본 단위와 소거동작의 기본단위가 다른 점으로 인해 낸드플래시를 스토리지로 사용하는 시스템의 경우 기존의 디스크를 기반으로 하는 시스템과는 달리 낸드플래시 메모리와 파일 시스템 사이에 플래시 변환 계층(FTL, flash translation layer)으로 불리는 추가적인 소프트웨어 계층을 사용하여야 한다[1].

플래시 변환 계층의 주요 역할은 논리적 주소공간과 물리적 주소공간의 매핑을 유지함으로써 쓰기 요구를 이미 소거 동작이 이루어져 쓰기 준비가 된 주소 공간으로 변환 하여 낸드플래시 메모리 기반 스토리지를 HDD(hard disk drive)와 같이 덮어쓰기가 가능한 블럭장치(block device)로 보이게 하는 역할을 한다. 이러한 과정에서 대부분의 쓰기동작들은 쓰기가 가능한 빈 블럭(free)을 만들어 내기 위해 블럭합병(block merge)과 폐영역 회수(garbage collection)의 과정을 동반하며 플래시 변환 계층이 이러한 동작을 제어하게 된다. 이 밖에 특정 블럭에 이상이 발생했을 때 새로운 예비된 블럭으로 교체하는 배드블럭(bad block) 관리, 모든 블럭이 균형적으로 쓰여져 제한된 쓰기/소거 회수 제약을 완화하는 마모도 평준화(wear-leveling) 등의 부가적인 기능을 하게 된다. 이러한 주요 기능 중 스토리지의 성능에 영향을 주는 가장 중요한 것은 논리적 주소 공간과 물리적 주소 공간을 매핑 하는 방법이다. 일반적으로 매핑이 이루어지는 단위에 따라서 소거 동작의 기본 단위인 낸드플래시의 블럭단위로 이루어지는 블럭기반 매핑 기법[2], 읽기/쓰기 동작의 가장 기본 단위인 페이지 단위로 이루어지는 페

이지 기반 매핑 기법[3], 그리고 두 기법을 혼합한 하이브리드 매핑 기법[4-6]으로 분류될 수 있다.

일반적으로 블럭기반 매핑 기법들은 유지해야 하는 매핑 공간이 작지만 비순차 갱신(non-sequential update)동작이나 블럭보다 작은 크기의 갱신에서는 매우 낮은 성능을 보인다. 반대로 페이지 기반 매핑 기법의 경우 매핑이 낸드플래시에서 가장 작은 단위인 페이지 단위로 매핑이 이루어지기 때문에 상대적으로 블럭매핑보다 비순차 갱신이나 작은 크기의 갱신에서 좋은 성능을 보이지만 매핑에 유지되는 공간의 낭비가 심할 뿐 아니라 연관된 논리 페이지들이 물리적으로는 서로 다른 블럭에 위치할 수가 있고 이는 블럭 합병 및 페어링 회수의 비용을 증가시켜 성능을 감소시키는 요인이 된다. 물론 두 가지 기법을 혼합하여 사용하는 하이브리드 매핑 기법의 사용으로 앞에서 제시한 문제점을 어느 정도 해결하였지만 낸드플래시를 기반으로 하는 스토리지는 앞에서 설명한 근본적인 두 가지 제약으로 인해 여전히 비순차 갱신(update) 및 작은 크기의 갱신에는 여전히 취약함을 보인다. 이러한 취약점은 높은 수준의 태스크 병렬화(task parallelism)로 인해 더욱 복잡하고 잦은 비순차 접근을 요구할 것으로 예상되는 차세대 시스템에서는 더욱 더 문제가 될 것이다.

이 밖에도 낸드플래시 메모리의 경우 다른 메모리 장치와는 다르게 반도체 기술이 발달하면서 저장할 수 있는 최대 용량은 증가하지만 칩의 절대적인 성능은 점차 악화되고 있다. 특히 메모리의 집적도를 증가시키기 위해 사용되는 다중레벨셀(MLC, multi-level-cell)기술을 사용하게 되는 경우 용량은 두 배로 증가하지만 쓰기 지연시간의 경우 평균 4배 정도(200us→800us)로 내구성은 1/10으로 (100K→10K)로 나빠지고 있을 뿐 아니라 기본적인 읽기/쓰기의 단위인 페이

지의 크기가 증가하여 작은 데이터의 처리는 더욱더 불리하게 된다. 이러한 요소들은 낸드플래시 기반 스토리지를 사용하기 위한 소프트웨어의 복잡도를 더욱 증가시키는 요인이 되고 있다.

따라서 본 연구에서는 낸드플래시 기반 스토리지의 한계점을 극복하고 소프트웨어 개발의 복잡성을 줄이기 위해 차세대 비휘발성 메모리를 활용한 낸드플래시 기반 스토리지의 성능향상에 대한 연구를 소개한다. 보다 구체적인 방법 및 차세대 비휘발성 메모리에 대한 상세한 설명은 다음 장으로 넘긴다.

## 2. 차세대 비휘발성 메모리(Next Generation Non-Volatile Memory)

### 2.1 특성

최근 낸드플래시나 노어플래시 메모리를 대체할 차세대 비휘발성 메모리에 대한 연구가 진행되고 있다. 차세대 비휘발성 메모리의 가장 큰 특징은 비휘발성 이면서도 빠른 바이트 별 무작위 접근(random access)이 가능하고 쓰기 전 소거 동작 없이 직접 갱신(inplace update)이 가능하다는 것이다. 차세대 비휘발성 메모리의 개발은 스토리지를 포함하는 메모리 시스템 설계에 있어서 빠른 부팅 및 응용의 실행/종료뿐 아니라 현재 대부분의 시스템에서 다양한 요구를 만족시키기 위해 다양한 종류의 메모리가 사용되는 것을 하나의 차세대 메모리로 통합하여 사용할 수 있게 하여 혁신적인 메모리 시스템을 설계할 수 있다는 기대감을 가져오고 있다. 초기에는 차세대 비휘발성 메모리가 기존의 주 메모리로서 사용되는 DRAM뿐 아니라 노어 및 낸드플래시 메모리와 같이 스토리지로 사용되는 메모리까지 대체할 수 있을 것으로 여겨졌지만 아직은 현실적으로 불가능하다. 현재 차세대 비휘발성 메모리라는 이름으로 많은 종류의 메모리들

표 1 플래시 메모리와 차세대 비휘발성 메모리의 특성 비교

	Flash (SLC 기준)	PRAM	MRAM	FeRAM
동작원리	FG에 전자 유무	물질의 상변화	전극의 자화 방향	강유전체의 분극특성
읽기속도	76.2 us/page	20-80 ns/word	35 ns/word	110 ns/word
쓰기속도	251.2 us/page	100 ns ~ /word	35 ns/word	110 ns/word
소거속도	1.5ms/block	N/A	N/A	N/A
최대용량 (확장성)	32Gbit (낮음)	512Mbit (높음)	16Mbit (낮음)	4Mbit (낮음)
전력소모	25mA	NA	55-105mA	18mA
내구성	10 <sup>5</sup>	10 <sup>12</sup>	무한	10 <sup>14</sup>
MLC 확장	가능	가능	불가능	어려움
개발상태	제품화	제품화 예정	제품화	제품화

이 연구가 되지만 본 연구에서는 상용화 되고 있거나 상용화에 근접하여 현실에서 사용이 가능하리라 생각되는 메모리인 PRAM(phase-changed RAM), MRAM(magneto-resistive RAM) 그리고 FeRAM(ferroelectric RAM) 등을 중심으로 살펴본다. 표 1은 앞에서 언급한 차세대 비휘발성 메모리들의 특성을 기존 플래시 메모리와 비교한 결과를 보여준다. PRAM은 제품화가 되지 않아 공개된 데이터가 없기 때문에 참고문헌 [7]을 참조하였으며 다른 메모리들은 생산되는 메모리의 데이터 자료 [8]을 참조하였다.

표에서 보면 이미 상용화 된 MRAM이나 FeRAM의 경우 다른 성능은 좋으나 확장성(scalability) 측면이 가장 큰 단점으로 지적되어 그 응용에 많은 제약이 따르고 있다. 따라서 기존연구에서는 용량의 제약으로 인해 MRAM이나 FeRAM에는 전체 데이터 중 작은 부분만을 저장하고 크기가 상대적으로 큰 데이터는 여전히 디스크나 낸드플래시 기반의 스토리지에 저장하는 하이브리드 형태로 연구가 진행되어 왔다.

반면 PRAM의 경우는 아직 상용화 제품으로 출시되지는 않았지만 성능뿐 아니라 확장성 측면에서도 좋은 특성을 나타낸다. 특히 PRAM은 다중레벨셀 구조도 지원이 가능하여 상대적으로 집적도가 높은 메모리를 구현할 수 있기 때문에 차세대 비휘발성 메모리로서 가장 주목을 받고 있다. 몇몇의 시장 예측조사기관은 향후 그 시장규모가 30조원(2016년 기준) 정도까지 이를 것으로 예상하고 있으며 한국의 삼성전자, 스위스의 뉴모닉스사가 주요생산 회사로서 2009년 양사 모두 상용화 제품을 출시 할 것으로 예상하고 있다[9]. 비록 PRAM의 특성을 현재 기술수준에서 살펴보면 쓰기 성능이 타 메모리에 비해 낮을 뿐 아니라 내구성(endurance) 측면에서 다소 불리한 면이 있지만 이는 앞으로 충분히 개선될 여지가 있고, 현재 기술수준의 PRAM이라 해도 기존의 노어플래시보다 성능/용량/내구성 등에서 뛰어 날뿐 아니라 대용량 데이터의 처리에 유리한 낸드플래시와 비교해도 작은 크기의 데이터 처리에 유리하기 때문에 MRAM 및

FRAM과는 달리 PRAM만을 가지고도 노어플래시를 대체할 수 있는 스토리지를 구현할 수 있다. 또한 그림 1과 같이 PRAM을 낸드플래시 메모리를 하이브리드 형태로 사용하면 앞에서 언급한 낸드플래시의 취약점들을 충분히 보완하여 현실적인 저비용/고성능의 스토리지를 구현할 수 있을 것이다. 본 연구에서는 PRAM과 낸드플래시 메모리를 하이브리드 형태로 사용하는 방법에 대해 소개한다.

## 2.2 메모리 장치 수준에서의 성능비교

낸드플래시와 PRAM을 같이 사용할 경우 두 장치가 어떠한 부분에 각각 유리한 지에 대한 분석이 필요하다. 따라서 본 장에서는 두 메모리 장치를 상위 응용 차원이 아닌 장치(칩) 자체수준에서 비교하고자 한다. 모든 동작에서 바이트단위의 접근을 허용하는 PRAM과는 달리 기존 낸드플래시 메모리의 경우는 페이지단위의 읽기/쓰기, 블록 단위의 소거동작만을 지원하기 때문에 두 가지 형태의 메모리의 직접적인 성능을 비교하기는 어렵다. 따라서 이들 메모리의 성능을 직접적으로 비교하기 위해 각각의 메모리에 대해 접근되는 데이터의 양( $n_{data}$ )을 인자로 하는 간단한 지연(latency) 모델을 다음과 같이 구성할 수 있다.

메모리장치의 데이터 전송 시간은 크게 두 가지로 나뉜다. 첫째는 초기지연시간(initial latency)으로 메모리 셀에서 메모리 장치내의 I/O버퍼로 전송되는 시간이며, 둘째는 버스 전송시간으로 I/O버퍼로부터 버스를 통해 CPU로 전송되는데 소요되는 시간이다. 먼저 본 연구에서 차세대 비휘발성 메모리로서 주된 대상으로 하고 있는 PRAM은 읽기 동작 시 기존의 DRAM이나 노어플래시 메모리와 같이 연속읽기 모드(burst mode)를 지원하기 때문에 초기 지연시간과 버스전송 시간 모델은 각각 다음과 같다.

$$L_{NV\_RRAD}(n_{data}) = T_{NVR} \cdot \frac{n_{data}}{N_B} + T_{BUS} \cdot \frac{n_{data}}{W_{NV}} \quad (1)$$

여기서  $T_{NVR}$ ,  $N_B$ ,  $T_{BUS}$  및  $W_{NV}$ 는 각각 PRAM의 읽기동작을 위한 초기 지연시간, 버스트(또는 페이지) 모

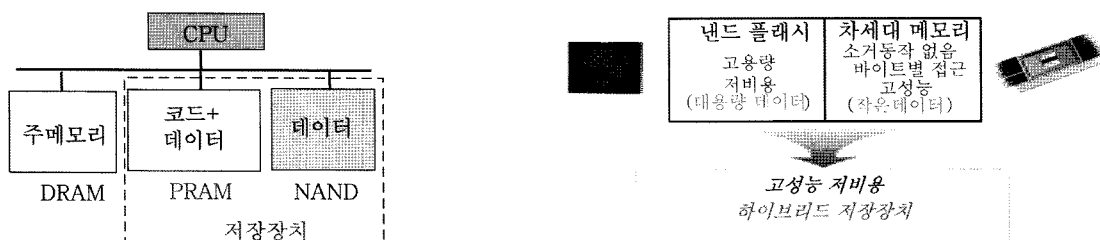


그림 1 PRAM\_NAND 하이브리드 스토리지 구조

드로 한번에 읽는 데이터 길이, 바이트당 버스 전송시간, 그리고 버스(혹은 워드)의 크기를 의미한다.

현재 생산될 것으로 예상되는 PRAM은 기존 NOR 플래시 메모리와 유사한 인터페이스를 가진다. 즉 읽기 동작과는 다르게 쓰기 동작의 경우 비동기화 단일 워드 쓰기(asynchronous single word write)만을 지원하기 때문에 그 성능은 다음과 같이 간략히 표현될 수 있다.

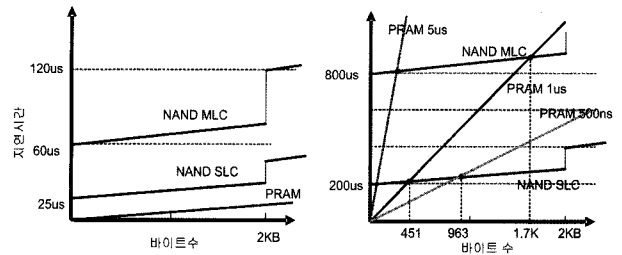
$$L_{NV\_WRITE}(n_{data}) = T_{NVW} \cdot \lceil \frac{n_{data}}{W_{NV}} \rceil \quad (2)$$

여기서  $T_{NVW}$ 는 초기 쓰기 지연시간을 의미하며 쓰기 동작 시 필요한 명령입력단계, 버스 전송단계 등은 초기 쓰기 지연 시간에 비해 상대적으로 적은 시간을 소요(수십ns 정도) 하므로 고려하지 않았다. 마지막으로 앞에서 제시한 방법을 기준으로 낸드플래시 메모리의 지연 모델을 구성하면 다음과 같다.

$$L_{ND\_READ}(n_{data}) = T_{NDR} \cdot \lceil \frac{n_{data}}{N_{psize}} \rceil + T_{BUS} \cdot n_{data} \quad (3)$$

$$L_{ND\_WRITE}(n_{data}) = (T_{NDW} + \frac{T_{NDE}}{N_{psize}}) \cdot \lceil \frac{n_{data}}{N_{psize}} \rceil + T_{BUS} \cdot n_{data} \quad (4)$$

위 식에서  $N_{psize}$  및  $N_{bysize}$ 는 각각 낸드플래시의 페이지 크기 및 블럭의 크기이며  $T_{NDR}$ ,  $T_{NDW}$  및  $T_{NDE}$ 는 각각 낸드플래시의 읽기, 쓰기, 그리고 소거동작의 지연 시간을 의미한다. 위 식에서 낸드플래시 메모리의 소거 시간에 대한 영향을 반영하기 위해 블럭단위의 소거 시간을 페이지 단위의 소거시간으로 변환한 후 이를 쓰기 시간에 포함시켰다. 보다 정확한 모델을 위해서는 낸드플래시 메모리에서 일어나는 블럭합병 및 폐영역 회수를 위한 비용이 고려되어야 하지만 이는 사용하는 플래시 변환 계층 알고리즘, 플래시 메모리의 상태에 따라 다르므로 위 식에서는 고려하지 않았다. 그림 2는 제시된 지연 모델을 기반으로 기존 낸드플래시 메모리와 PRAM에 대한 바이트 별 지연시간을 비교한 결과이다. 낸드플래시의 경우 상용의 SLC(single level cell) 및 MLC(multi-level cell) 낸드플래시의 데이터 스위트를 참조하였으며 PRAM의 경우 상세한 자료가 아직 공개되지 않았기에 세가지로(워드당  $0.5us$ ,  $1us$ ,  $5us$  쓰지연시간) 가정하여 표현하였다. 표에서 알 수 있듯이 장치의 종류, 쓰고자 하는 데이터의 크기에 따라서 손익분기점(break-even point)이 다르게 나



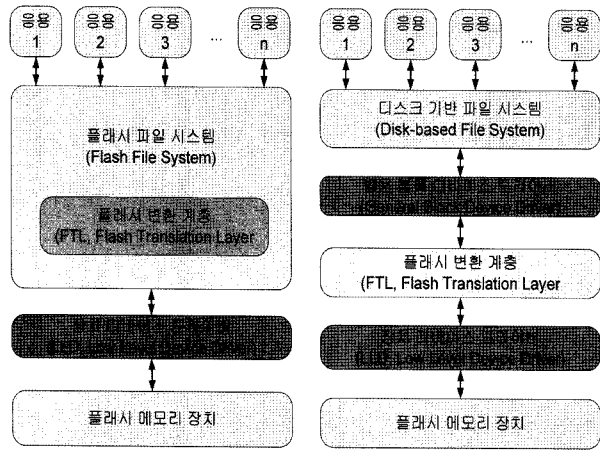
(a) 읽기 지연시간 (b) 쓰기 지연시간  
 그림 2 메모리 장치 수준에서의 읽기/쓰기 성능 비교

타난다. 예를 들어 451바이트 보다 작은 데이터를 쓰는 경우 SLC 낸드플래시 보다는 1us의 쓰기 지연시간을 갖는 PRAM에 쓰는 것이 유리하다. 이러한 손익분기점은 쓰고자 하는 데이터의 크기에 따라 낸드 혹은 PRAM으로 할당해야 할지에 대한 정책을 결정하는 중요한 요소로 사용할 수 있다.

### 3. 기존 연구 소개

낸드플래시 기반 스토리지를 사용하는 소프트웨어 구조는 그림 3과 같이 두 가지로 나누어 질 수 있다. 그림 3의 (a)와 같은 구조는 파일시스템 설계 시부터 낸드플래시를 고려하여 설계되었기 때문에 독립된 플래시변환계층이 포함되지 않는다. 반면 (b)의 구조는 일반적인 디스크 기반 파일 시스템에 낸드플래시 메모리를 디스크처럼 덮어쓰기가 가능한 일반적인 블럭장치로 보이게 하는 플래시변환 계층을 사용한다. YAFFS, JFFS, JFFS2등이 전자의 파일 시스템으로 사용되며 후자는 FAT, EXT2, 3등의 어떠한 디스크 기반 파일시스템도 사용될 수 있다. 낸드플래시를 지원하는 스토리지 소프트웨어의 두 가지 접근 방식 모두 공통적으로 낸드플래시에 기인한 성능 문제를 극복하려는 노력이 필요하며, 이를 위해 기존 연구들은 차세대 비휘발성 메모리와 낸드플래시 메모리를 혼용하여 사용하는 연구들이 진행되어 왔다.

플래시메모리 전용 파일 시스템을 사용하는 경우 표준 디스크 포맷의 제약이 없을 뿐 아니라 해당 소프트웨어 등은 공개소프트웨어(open source) 등을 통해 쉽게 접근이 가능하기 때문에 기존의 연구들은 대부분 플래시 전용 파일 시스템을 수정하는 방식으로 이루어져 있으며 이에 대한 연구들을 본 장에서 간략히 소개를 한다. 반면 대부분의 상용의 제품에서는 시스템간의 호환성을 중요시하기 때문에 기존 디스크 기반 스토리지와 호환성을 유지 할 수 있는 그림 3의 (b)와 같은 구조를 채택하여 사용하고 있다. 그러나 이러한 구조에서는 기존 호환성을 유지하면서도 차세



(a) 플래시 전용 파일 시스템 기반의 저장장치 (b) 디스크 기반 파일 시스템+플래시 변환 계층 기반의 저장 장치

그림 3 플래시 메모리 기반 스토리지용 소프트웨어 구조

대 비휘발성 메모리를 추가하여 보완하는 방식에[10] 제약이 많기 때문에 많은 연구들이 이루어지지 못했다. 따라서 그림 3의 (b)와 같은 구조에서 호환성을 유지하면서도 낸드플래시 기반 스토리지의 단점을 차세대 메모리를 활용하여 보완할 수 있는 연구 기법에 대해서는 4장에서 보다 자세하게 설명할 것이다.

### 3.1 하이브리드 파일 시스템

차세대 비휘발성 메모리를 사용한 하이브리드 파일 시스템은 기존의 낸드플래시에 기반한 임베디드 스토리지 시스템의 성능을 개선하는 것을 목표로 하고 있다. 주로 빈번한 무작위 위치 기록으로 인해 낸드플래시 스토리지 시스템의 성능이 제한되는데, 파일 시스템 메타 데이터 갱신이 그 주요한 원인이라는 점에 착안하여 메타 데이터를 차세대 비휘발성 메모리에 위치시키는 기법들이 제안되어 왔다.

FRASH[11]와 MiNVFS[12]는 낸드플래시용 파일 시스템인 YAFFS의 마운트 지연 시간을 절약하기 위해 차세대 비휘발성 메모리인 FRAM을 사용하였다. YAFFS는 낸드플래시의 제약사항을 극복하고 성능을 개선하고자 파일 데이터 블록 포인터와 디렉토리 구조를 휘발성 램에서 관리한다. 이로 인해서 마운트시 낸드플래시의 전 영역을 주사(scan)하여 흩어져 있는 메타 데이터 블록을 찾고 휘발성 램에 파일 데이터 블록 포인터와 디렉토리 구조를 재구성해야 한다. 전체 영역 주사 방식을 사용함에 따라 자연스럽게 파일 시스템의 크기가 커짐에 따라 마운트 당시 주사해야 하는 영역도 증가하고 마운트 지연시간도 선형적으로 증가한다. 최신의 YAFFS는 이와 같은 마운트 지연 시간 문제를 극복하고자 언마운트시 휘발성 램에 구성되어 있는 파일 시스템 구조의 스냅샷을 낸드플래시에 일

괄 기록하는 기법을 사용하고 있지만, FRASH에서는 이를 위해 비휘발성 메모리를 활용하는 기법을 제안하였다. 메타 데이터 블록을 FRAM에 위치시켜 마운트시 제한된 비휘발성 메모리 영역만을 주사하면 되기 때문에 마운트 지연시간을 대폭 감소시킬 수 있었다. 하지만 여전히 FRAM 영역의 주사가 필요하며 파일 시스템의 확장에 따라 주사하는 영역 또한 선형적으로 증가하여 마운트 시간과 메모리 사용량 측면에서 문제가 있다. 반면 MiNVFS는 휘발성 램에서 구성되어 하는 YAFFS의 파일 시스템 구조를 비휘발성 메모리에 위치시킴으로써, 마운트시 전 낸드플래시 영역에 대해 주사하는 과정을 제거하였고 휘발성 램 사용량을 절약하였다.

PFFS[13]는 PRAM과 낸드플래시를 혼용하는 파일 시스템이다. 바이트 수준의 접근이 가능한 PRAM에 파일 시스템 메타데이터를 영구적으로 위치시키는 구조로 마운트 시간과 주 메모리 사용량 문제를 극복하였다. 또한 PRAM의 내구성 향상을 위해 세그먼트 변환(segment switching)에 기반한 마모도 평준화 기법을 제안하였다.

이와 같이 하이브리드 파일 시스템은 낸드플래시의 성능 문제를 개선하기 위해 차세대 비휘발성 메모리를 혼용하여 사용하였고 차세대 비휘발성 메모리는 바이트 별 접근이 가능하여 메타 데이터 저장소로 적합하다고 연구되어 왔다. 이를 통해서 파일 시스템의 마운트 시간과 메모리 요구량을 크게 절약하였다.

### 3.2 비휘발성 메모리 전용 파일 시스템

기존 차세대 비휘발성 메모리 전용 파일 시스템들은 공간 효율성을 향상 시키는 노력에 집중하였다. 비휘발성 메모리는 바이트 수준의 주소 매김에 의한 읽기 및 쓰기가 가능하기 때문에 데이터를 메모리에 로딩하여 가공할 필요가 없고 부분 읽기 쓰기가 가능하다. 이러한 장점은 파일 시스템 사용시 체감 속도를 크게 향상시킬 수 있다. 하지만, 현재 기술 수준의 비휘발성 메모리는 집적도 측면에서 현재 널리 사용되는 낸드플래시에 비해 현격히 부족하다. 이를 극복하기 위해 소용량의 비휘발성 메모리를 공간 효율적으로 사용하는 비휘발성 메모리 전용 파일 시스템들이 제안되었다.

NEBFS[14]는 비휘발성 메모리의 공간 효율성을 향상 시킨 파일 시스템을 제안한다. 첫째, 데이터 블록의 내부 단편화로 인한 공간 낭비를 방지하기 위해서 블록의 크기를 가능한 작게 유지한다. 둘째, 파일 데이터 블록들을 범위(extent)기반으로 관리함으로써 파

일 메타 데이터가 차지하는 공간을 절약하였다. 기존의 파일 시스템들은 데이터 블록에 대한 위치 포인터를 기록하는 정책으로 매 데이터 블록에 대응하는 포인터를 저장 관리해야 한다. 하지만 파일 데이터 블록들을 저장매체에 연속되게 저장한다면 연속된 데이터 블록들의 시작 위치와 연속된 블록 수를 관리하면 된다. 이와 같은 기법을 제안하여 같은 규모의 파일들을 관리하는데 요구되는 비휘발성 메모리 영역의 크기를 크게 절약하였다.

MRAMFS[15]는 비휘발성 메모리인 MRAM을 위한 파일 시스템이다. MRAM의 낮은 집적도로 인하여 공간 효율성을 향상 시키려고 압축기법을 제안했다. 첫째, 메타 데이터는 상대적으로 높은 효율로 압축될 수 있다. 압축율에 대한 연구 분석을 바탕으로 메타 데이터는 감마 압축법(gamma compression)이 적합하다. 반면 파일 데이터의 압축은 LZO 또는 LZRW1 압축법을 사용하였다.

PRAMFS[16]는 리눅스 운영체제에서 구현된 비휘발성 메모리 전용 파일 시스템이다. PRAMFS가 가정하는 비휘발성 메모리는 비휘발성이 되 휘발성 램과 같은 주소 매김 인터페이스를 사용하고 접근 지연 시간 측면에서도 유사한 특징을 가진다. 따라서, 파일 시스템의 데이터를 휘발성 램의 페이지 캐시에 로딩하지 않고 비휘발성 메모리에서 직접 접근하여 사용한다. 또한 디스크 기반의 파일 시스템에서 사용되어온 디스크 최적화 성능 기법들이 제거되어 파일 시스템의 복잡도가 매우 낮아지게 되었다.

이와 같이 비휘발성 메모리 전용 파일 시스템에서는 현재 비휘발성 메모리의 집적도 기술 수준이 낮음에 집중하여, 공간 효율성을 향상시키기는 기법들이 제안되어 왔다. 이러한 기법에는 메타 데이터를 압축하거나 블록들을 범위 기반으로 관리하는 방법들이 있다.

#### 4. FAT-호환성을 유지하는 하이브리드 스토리지

그림 3의 (b)에서 보여준 바와 같이 낸드플래시를 기반으로 하는 소프트웨어는 크게 플래시 변환 계층과 파일 시스템으로 분류 된다. 본 장에서는 차세대 메모리를 활용하여 플래시 변환 계층에서 개선하는 방법과 파일 시스템 차원에서 개선하는 두 가지 방법으로 나누어 설명한다.

##### 4.1 하이브리드 플래시 변환 계층

낸드플래시에 비해 차세대 비휘발성 메모리의 이점은 빠른 읽기 성능을 보이면서도 바이트 별 접근이 가능하기 때문에 작은 크기의 데이터 갱신에 유리하다는 것이고 비록 쓰기 지연시간이 상대적으로 크지만 쓰기 전 소거제약이 없기 때문에 항상 일정한 성능을 보일 수 있다는 것이다. 이러한 관점에서 출발하여 본 절에서는 페이지 매핑 기반의 플래시 변환 계층 구현 및 파일시스템 같은 상위 계층으로부터 내려오는 사용자 데이터를 낸드플래시 뿐 아니라 요구에 따라서 PRAM에 저장할 수 있는 요구기반 섹터 페이징(on-demand sector paging)에 대해 소개한다.

##### 4.1.1 페이지 매핑 기반 플래시 변환 계층 및 요구기반 섹터 페이징

플래시 변환 계층의 가장 중요한 역할은 주소 변환이다. 서론부분에서 제시한 것과 같이 논리적 주소 공간을 물리적 주소공간으로 매핑하는 것은 그 수준에 따라 요구되는 메모리의 크기가 달라질 뿐 아니라 매핑에 대한 정보는 빠른 주소변환을 위해 주 메모리(main memory)에 저장해 왔다. 따라서 비순차갱신 동작에서도 비교적 높은 성능을 보이는 페이지단위의 매핑 방법은 많은 메모리 요구량 때문에 그 사용에 제약을 받게 된다. 또한 매핑테이블의 경우 시스템의 전원이 갑자기 끊기더라도 전원 회복 후 그 내용을 복구할 수 있어야 하기 때문에 많은 부가적인 작업을

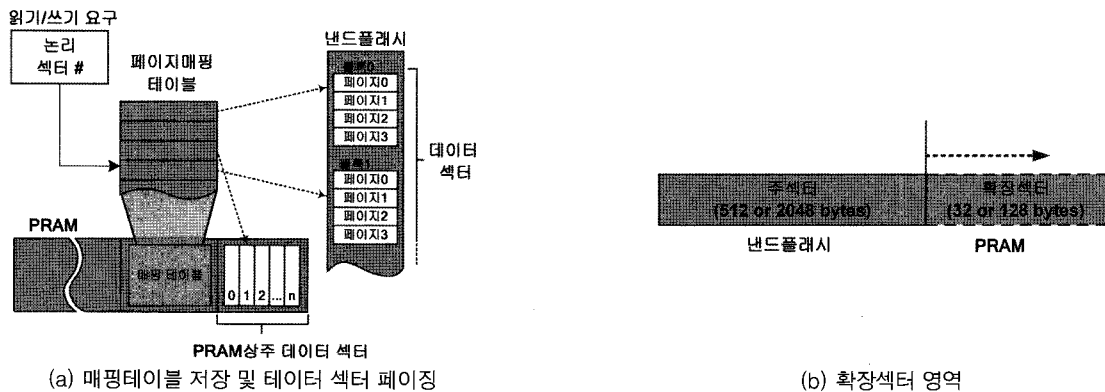


그림 4 플래시 변환계층 수준에서의 차세대 비휘발성 메모리 활용

해야 하기 때문에 플래시 변환 계층을 설계하는데 있어서 복잡도를 증가시키는 요인이 된다. 따라서 이러한 매핑정보를 포함하는 플래시변환계층의 메타데이터를 비휘발성이고 비용이 높은 주 메모리보다 그림 4의 (a)와 같이 낮은 비용의 비휘발성 메모리인 PRAM 같은 차세대 비휘발성 메모리에 저장하면 앞에서 언급한 대용량 주 메모리 요구 문제, 비정상 전원공급문제 등에 비교적 쉽게 대응할 수 있을 뿐 아니라 소프트웨어 개발의 복잡도를 감소시킬 수 있다.

또한 플래시변환계층의 매핑 정보뿐 아니라 데이터의 특성에 따라 파일 시스템 같은 상위 계층에서 내려오는 사용자 데이터라 할지라도 그 특성에 따라서 낸드플래시 보다 PRAM에 저장하는 것이 유리한 경우가 발생한다. 데이터를 PRAM에 저장하는 경우 데이터를 디스크기반의 시스템에서 최소의 데이터 입출력 단위인 섹터형태로 유지하면서도 상황에 따라서 필요한 바이트만 접근하여 읽고 쓰는 것이 가능하도록 설계한다면 상대적으로 갱신되는 양이 적은 상위 계층의 데이터를 처리하는데 유리할 수 있다. 이 경우 상위 계층으로부터 접근 요구가 왔을 경우 PRAM에 저장되는 매핑테이블에 1비트의 정보만을 추가적으로 사용하여 효과적으로 매핑을 관리할 수 있기 때문에 추가적인 부하는 크지 않다.

#### 4.1.2 확장영역 제공

디스크 또는 낸드플래시 메모리 기반의 스토리지들은 섹터(혹은 페이지) 단위의 입출력만을 지원하기 때문에 상위의 소프트웨어에서는 섹터(혹은 페이지) 단위로 입출력 되도록 이루어졌다. 따라서 바이트 별 접근이 가능한 저장공간을 지원하더라도 이를 활용하기 위해 파일 시스템 같은 시스템 소프트웨어뿐 아니라 스토리지에 기록되는 디스크 포맷(on-disk format)의 구조를 바꾸지 않으면 이를 제대로 활용하기 어렵다. 또 이러한 변경이 가능하다 하더라도 기존의 시스템과는 호환성을 유지할 수 없기 때문에 이러한 변환은 현실적으로 쉽지 않은 상태이다. 따라서 기존 시스템과 호환성을 잃지 않으면서도 적극적으로 바이트 별 접근이 가능한 비휘발성 메모리를 활용할 수 있도록 하는 것이 필요한데, 그림 4의 (b)와 같이 낸드 플래시에 저장되는 주 섹터영역은 호환성을 위해 그대로 유지하면서 주 섹터영역에 대한 요약정보 혹은 확장 정보를 기록할 수 있는 확장섹터영역을 제공하면 파일시스템 등에서는 이를 응용에 특화된 공간으로 활용을 할 수가 있을 것이다. 물리적으로는 낸드 플래시와 PRAM공간이 분리되어 있지만 이들간의 매

핑을 플래시변환 계층에서 자동적으로 관리를 하게 되고 파일시스템 같은 상위 계층의 소프트웨어에서 요구 시에만 공간을 제공토록 한다면 추가적인 공간의 제공으로 인한 공간 낭비도 최소화 할 수 있을 것이다. 이 확장섹터영역에 대한 활용은 다음 장에서 설명할 파일 시스템 차원에서의 최적화 기법에서 설명한다.

#### 4.1.3 바이트 별 접근 가능한 인터페이스 지원

파일시스템 같은 상위계층의 소프트웨어가 스토리지로 사용되는 차세대 비휘발성 메모리를 효과적으로 접근하기 위해서는 블록단위 뿐 아니라 바이트단위의 접근을 허용하는 인터페이스의 지원이 필수적이다. 비록 PRAM이 바이트 별 접근을 허용하지만 주 메모리가 아닌 스토리지용도로써 사용될 뿐 아니라 쓰기 동작의 경우 단순한 DRAM처럼 사용할 수 없기 때문에 파일 시스템에서 보다 효율적으로 PRAM영역을 접근할 수 있도록 인터페이스를 지원해야 한다. 이를 위해 *IOCTL()* 함수를 이용할 수 있다. 읽기 동작의 경우 응용은 *IOCTL()* 함수를 사용하여 플래시변환계층에 해당 섹터들이 저장된 차세대 비휘발성 메모리의 주소를 요구하고 플래시변환계층에서는 해당하는 비휘발성 메모리의 물리적 주소를 제공할 수 있다. 이렇게 되면 응용은 제공된 물리적 주소를 이용하여 비휘발성 메모리로부터 직접 필요한 데이터를 얻어 올 수가 있다. 반면 쓰기 동작은 PRM의 경우 읽기 동작과는 다른 형태로 이루어지기 때문에 응용이 직접 스토리지 공간을 수정할 수 없다. 대신 응용은 플래시변환계층에 *IOCTL()* 함수를 사용하여 해당 섹터 번호 혹은 수정하려는 데이터의 상대적 위치와 함께 데이터를 전송하면 플래시 변환 계층이 스토리지에 대한 직접적인 쓰기 동작을 수행하게 할 수 있다. 이러한 방법은 스토리지 공간의 비정상적인 수정을 예방하면서도 스토리지에서 전체 섹터영역이 아닌 필요한 영역의 바이트들만 접근하게 함으로서 효율적인 스토리지 구성을 가능하게 한다.

#### 4.2 하이브리드 파일 시스템

본 절에서는 파일 시스템 차원에서의 하이브리드 스토리지 구조를 활용하기 위한 기술들을 제안한다. 제안된 파일 시스템은 앞장에서 설명한 하이브리드 플래시변환 계층에서 제공하는 API를 사용하여 비휘발성 메모리 영역을 블록수준 또는 바이트 수준으로 접근할 수 있으며, 이를 통해 쓰기 성능과 폐영역 회수 오버헤드가 크게 개선된다.

#### 4.2.1 비휘발성 메모리 상주 메타 데이터

제안한 파일 시스템은 하이브리드 플래시 변환 계층의 요구기반 섹터 페이징 기법을 활용하여 파일 시스템의 메타 데이터를 비휘발성 메모리에 위치시킨다. 파일 시스템 메타 데이터의 빈번한 갱신은 전체 파일 시스템 성능 문제의 주요 원인으로 인식되고 있다. 따라서, 메타 데이터 갱신 성능을 개선시키는 것은 전체 파일시스템의 성능을 개선하는데 중요한 비중을 차지하고 있다.

메타 데이터를 비휘발성 메모리에 위치시키는 것으로 크게 두 가지 점에서 성능 개선이 가능하다. 첫째, 빈번한 메타 데이터 갱신에 의한 낸드플래시 메모리 상의 무작위 위치 쓰기 발생을 차단하여 폐영역 회수 부하를 경감한다. 낸드플래시에서 무작위 쓰기는 폐영역 회수 부하 발생의 주요 원인이며 이를 감소시키는 것은 성능 개선에 큰 도움이 된다. 특히 메타 데이터 갱신은 단편화되고 분산된 영역에서 빈번하게 발생하는 것이 특징으로 무작위 쓰기의 발생원인 중 주요한 비중을 차지한다. 따라서, 메타 데이터를 낸드플래시에서 분리하여 무작위 위치 쓰기에 강인한 비휘발성 메모리에 저장함에 따라 파일 시스템 성능 개선에 도움이 된다. 둘째, 비휘발성 메모리는 바이트 주소 매김이 가능하기 때문에 이를 이용하면, 개별 메타 데이터 섹터 갱신 시 요구되는 지연 시간을 감소시킬 수 있다. 메타 데이터 섹터 갱신 시 대부분의 경우 매우 일부분만 수정되는 특성이 있다.

그림 5에서 보는 바와 같이 메타 데이터 쓰기 요청 시 실제 수정되는 영역의 크기는 평균적으로 섹터 크기(512바이트)의 11% 정도이다. 이외의 영역은 대부분 수정되지 않음에도 낸드플래시와 같은 블록 기반 스토리지에서는 이를 개별적으로 접근할 수 없기 때문에 다시 기록되어야 해서 비효율적이다. 바이트 수준

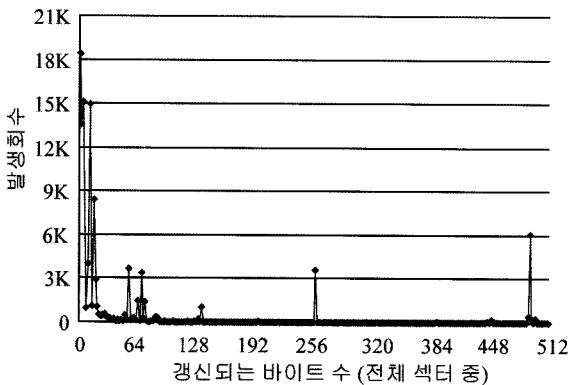


그림 5 메타 데이터 섹터 쓰기 요구당 갱신되는 데이터 크기 분포

의 주소 매김이 가능한 비휘발성 메모리는 블록내의 데이터를 바이트 단위로 기록할 수 있기 때문에, 메타 데이터 갱신 시 수정된 바이트에 대해서 선택적으로 기록하는 기법을 제안하였다. 이와 같은 선택적 블록 기록 방식은 메타 데이터 블록 갱신 시 지연 시간은 매우 단축시킨다.

수식 1~4에 기반하여 메타데이터 블록을 차세대 메모리에 기록하는 기법의 평균 성능 개선 정도를 예상해보면 아래와 같다.

$$\frac{L_{NV\_Meta\_WRITE}}{L_{ND\_Meta\_WRITE}} = \frac{L_{NV\_WRITE} (0.11 * sectorsize)}{L_{ND\_WRITE} (sectorsize)} = \frac{145us}{827us} = 0.166. \quad (5)$$

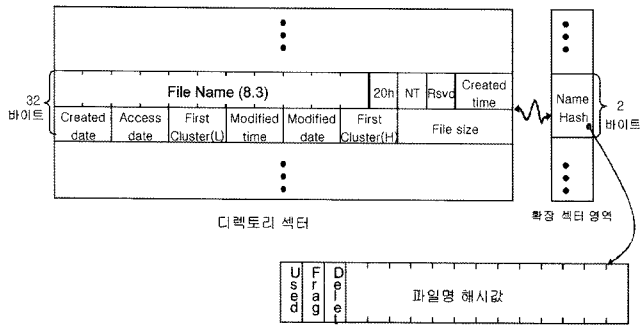
위 수식은 512바이트의 섹터 크기와 MLC 낸드플래시, 그리고 5us/word의 PRAM 환경을 가정하여 계산된 것으로, 메타 데이터 갱신이 차세대 메모리상에서 이루어지면 낸드플래시 상에서 이루어지는 것보다 약 5.7배의 성능향상이 예상됨을 보여준다.

비휘발성 메모리로 저장되는 메타 데이터 섹터는 FAT와 디렉토리로 구분된다. 초기에 파일 시스템의 포맷이 수행 될 때 생성되는 FAT 영역은 모두 비휘발성 메모리에 위치시키며, 운영시 FAT 영역에 대한 접근은 모두 비휘발성 메모리에서 이루어진다. 디렉토리 섹터의 경우 파일 시스템이 사용됨에 따라서 생성되는 것으로 생성시에 하이브리드 플래시변환계층이 제공하는 페이징 섹터 사용 인터페이스를 사용하여 해당 디렉토리 섹터에 대한 비휘발성 메모리 영역을 요구한다. 메타 데이터 섹터를 위한 비휘발성 메모리영역 요구가 실패하는 경우 다시 낸드플래시를 사용한다. 따라서 파일 시스템은 디렉토리 섹터를 접근 시 위치 판별을 위해 하이브리드 플래시 변환계층이 제공하는 인터페이스를 이용한다.

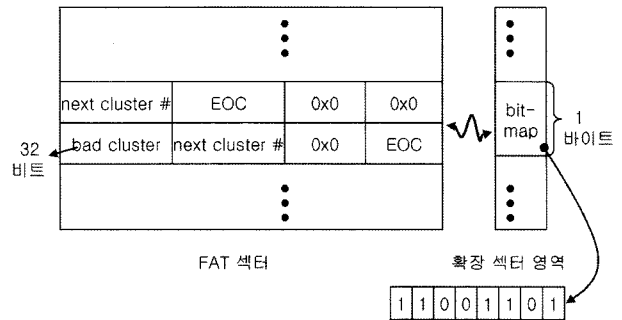
#### 4.2.2 확장 섹터 영역 활용

제안한 파일 시스템은 메타 데이터 섹터를 비휘발성 메모리에 위치시키는 기법 외에 하이브리드 플래시변환계층이 제공하는 비휘발성 메모리상의 확장 섹터 영역을 활용하여 특정 메타 데이터 섹터에 대한 요약 정보를 저장한다. 비휘발성 메모리의 확장 섹터 영역은 한 개의 정규 섹터에 대응하는 32바이트의 확장된 영역을 제공하고, 파일 시스템의 요구에 의해 특정 섹터에 대해 할당 및 해제한다. 확장 섹터 영역에 저장되는 요약 정보는 파일 시스템의 검색 및 할당 성능을 크게 개선되는 데 유용하다. 본 연구에서 활용한 요약 정보는 다음과 같이 크게 두 가지이다.





(a) 파일명 해싱(hashing) 기법



(b) 클러스터 비트맵의 활용

그림 6 확장 섹터 영역의 활용

첫째, 디렉토리 섹터에 대응하는 확장 섹터 영역에 파일명의 해시값을 저장한다. 디렉토리 섹터 내에서 산재한 디렉토리 엔트리를 모두 순차적으로 검색하는 대신, 요약 정보 공간의 파일명 해시값을 먼저 비교 검색함으로써, 파일 생성 및 열기 시 필요한 파일 검색의 지연시간을 크게 감소시킬 수 있다. 그림 6의 (a)에서 보는 바와 같이 디렉토리 섹터에 위치한 32 바이트의 디렉토리 엔트리에 대응하는 2바이트의 파일명 해시값이 확장 섹터 영역에 상주한다. 파일 검색 시 확장 섹터 영역의 파일명 해시값을 먼저 비교 검색하게 되면 검색시간을 크게 단축할 수 있다.

둘째, 클러스터 할당을 위해서 필요한 비트맵을 FAT 섹터의 확장 섹터 영역에 영구적으로 저장 가능하다. 기존의 FAT 스펙에서는 클러스터 비트맵을 저장 매체에서 영구적으로 보존하지 않기 때문에 매번 메모리 상에서 구성을 해야 했다. 이를 구성하기 위해서는 모든 FAT 섹터에 대해 읽기 수행을 해야 하는데, 메모리 사용량과 지연시간 측면에서 비효율적이었다. 따라서 하이브리드 기능을 제공하는 FAT은 모든 FAT 섹터에 비휘발성 메모리의 요약 정보 공간을 사상시키며 해당 요약 정보 공간에는 해당 섹터가 포함하는 클러스터의 비트맵 정보를 유지 관리한다. 따라서, 클러스터 할당 시 빈 클러스터를 검색하

기 위해 단지 비트맵 정보를 접근하면 되므로 그 과정이 매우 단순해지는 장점이 있다. 그림 6의 (b)에서 보는 바와 같이 FAT 섹터에 위치한 32 비트의 클러스터 정보에 대응하는 각 1비트의 비트맵 정보가 확장 섹터 영역에 상주한다. 유휴 클러스터 검색 시 확장 섹터 영역의 비트맵을 통해 연산하게 되면 FAT 섹터 전체를 검색하는 것보다 지연 시간을 크게 단축할 수 있다.

### 4.3 실험 및 평가

앞장에서 제시한 기법들을 평가하기 위해 다음의 세가지로 평가 시스템을 구성하였다.

- 미적용 - 기존 파일 시스템 + 기존 플래시변환계층
- FTL 적용 - 기존 파일 시스템 + 개선된 플래시변환계층
- FTL+FS적용 - 개선된 파일 시스템 + 개선된 플래시변환계층

기존 파일시스템으로는 상용화된 FAT파일 시스템을 사용하였으며 기존 플래시 변환계층으로 역시 상용제품에 사용되는 페이지 매핑 기반 플래시변환계층을 사용하였다. 평가에 사용된 워크로드(workload)는 스토리지의 성능을 측정할 때 일반적으로 널리 사

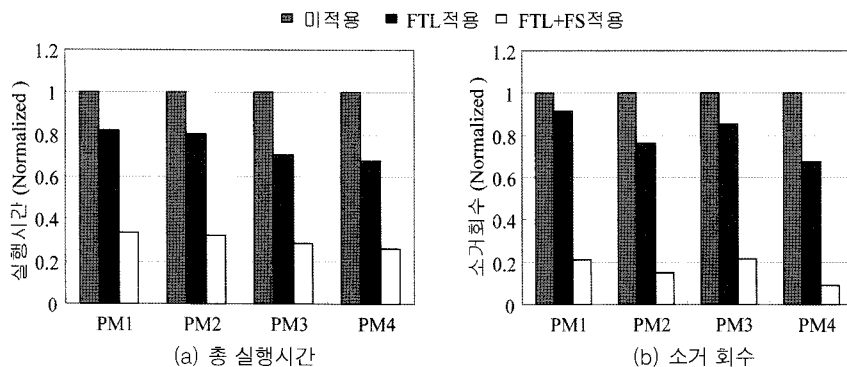


그림 7 성능 및 내구성 평가

용되는 Postmark[17] 벤치마크를 사용하였으며 다양한 워크로드를 표현하기 위해 초기 생성되는 파일의 개수, 크기, 트랜잭션의 수 및 총 쓰기량을 달리한 네 가지 구성 PM1~PM4로 실험을 하였다. 실험에 사용된 스토리지는 모두 256MB크기로 설정하였다.

그림 7은 제시된 세가지 시스템 구성에서 PM1~PM4로 설정된 Postmark 벤치마크를 수행하였을 때의 성능과 관련된 총 수행 시간과 내구성에 관련된 소거회수를 정규화한(normalized) 값으로 표시하였다. 기존 시스템에서 플래시변환계층에만 차세대 비휘발성 메모리를 활용하는 방법을 적용했을 경우 기존 방법에 비해 벤치마크 구성에 따라 약간 차이는 보이지만 전체적으로 실행시간은 25% 정도 감소하는 것을 볼 수 있다. 이는 플래시변환계층에서 매핑정보를 포함하는 플래시변환계층의 모든 제어용 데이터를 PRAM에 저장함으로써 상대적으로 낸드플래시에 대한 접근을 효율적으로 할 수 있도록 했기 때문이다. 다음으로 차세대 비휘발성 메모리를 플래시변환계층 및 파일 시스템에 모두 적용한 경우 미적용 시스템과 비교하여 70% 정도 실행 시간이 단축되었다. 이는 파일시스템의 메타데이터를 PRAM에 저장함으로써 절대적인 낸드플래시 메모리로의 접근 회수를 줄였을 뿐 아니라 메타데이터로 인한 사용자데이터의 단편화 현상을 줄인 결과이다. 또한 확장영역을 적절히 활용함으로써 파일 시스템에서는 주 섹터영역을 바로 접근하지 않고 접근비용이 적은 확장섹터영역을 접근하게 하여 주 섹터영역의 데이터가 필요한 경우에만 낸드플래시를 접근하도록 하였기 때문에 전체적인 성능이 크게 향상될 수 있었다.

마지막으로 낸드플래시의 수명과 관계되는 소거회수의 경우 차세대 비휘발성 메모리의 사용은 성능향상의 요인과 마찬가지로 절대적인 낸드플래시 메모리로의 접근 회수가 줄었을 뿐 아니라 적은 비순차쓰기, 작은 크기의 데이터 쓰기 등을 효과적으로 제거하여 플래시변환계층에서 발생하는 블럭합병 및 폐영역의 회수의 비용을 획기적으로 줄였기 때문에, 수명은 플래시변환계층만을 적용하였을 경우 1.25배, 파일시스템까지 모두 적용하였을 경우 평균 6배 정도까지 증가시킬 수 있었다.

## 5. 결론

낸드플래시 메모리 기반의 스토리지가 저비용, 고용량, 저전력, 안정성 등의 많은 이점으로 인해 휴대용 기기에서 널리 사용되고 있지만 낸드플래시 메모리의 쓰기 전 소거제약과 읽기/쓰기동작의 단위와 소

거동작의 단위가 다른 점 등의 특성으로 인해 비순차쓰기와 작은 크기의 데이터 갱신에서는 상대적으로 낮은 성능을 보인다. 이러한 것들을 해결하기 위해 플래시변환계층, 파일시스템 등의 스토리지 관련 시스템 소프트웨어 등에서 이루어진 많은 연구들 중 본 연구에서는 차세대 메모리를 활용한 스토리지의 성능향상에 관한 연구들에 대해 살펴보았다. 특히 제품화 생산을 앞두고 있는 PRAM을 활용한 플래시 기반 스토리지의 성능 및 수명을 높이는 방법에 대해 보다 상세히 살펴보았으며, 이를 통해 차세대 비휘발성 메모리와 낸드플래시의 병행 사용은 낸드플래시 메모리에서 발생하는 비순차 쓰기뿐 아니라, 작은 크기의 데이터 접근을 줄임으로써 많은 성능향상이 있음을 알 수 있었다.

본 연구에서는 차세대 비휘발성 메모리를 스토리지 관점 특히 플래시변환계층, 파일 시스템의 최적화 측면에서 살펴보았지만 차세대 비휘발성 메모리의 이점은 데이터베이스 관리 시스템(DBMS, database management system)과 같이 스토리지를 집중적으로 사용하는 응용차원의 문제점까지 해결할 수 있을 것이다. 이 밖에도 그 성능이 더 개선되어 앞으로 DRAM과 같은 주 메모리를 보완하거나 대체할 수 있는 수준이 되면 시스템에서의 활용도 및 이점은 더욱더 커질 것으로 예상하고 있다.

## 참고문헌

- [1] Intel Corp., "Understanding the flash translation layer(FTL) specification," <http://developer.intel.com>, 1998.
- [2] A. Ban, "Flash file system," United States Patent, No.5,404,485, April 1995.
- [3] A. Birrel, M. Isard, C. Thacker, and T. Wobber, "A design for high-performance flash disks," ACM SIGOPS Operating Systems Review, vol. 41, no. 2, April 2007.
- [4] J. Kim, J. M. Kim, S. Noh, S. L. Min, and Y. Cho, "A space-efficient flash translation layer for CompactFlash systems," IEEE Transactions on Consumer Electronics, vol. 48, no. 2, pp. 336-375, 2002.
- [5] S. Lee, D. Park, T. Chung, L. D. S. Part, and H. Song, "A log buffer-based flash translation layer using fully - associative sector translation," ACM Transactions on Embedded Computing Systems, vol. 6, no. 3, 2007.
- [6] C.-H. Wu and T.-W. Kuo, "An adaptive two-level management for the flash translation layer in em-

bedded systems,” in Proceedings of ICCAD’06, November 2006.

[7] 유병근, 류상욱, 윤성민, “유비쿼터스용 유니버설 메모리 기술(MRAM, FeRAM, PRAM),” 한국전자통신연구원 1월호, 2006.

[8] NAND(K9F2G08U0M–Samsung Electronics), FeRAM (FM22L16, Ramtron), MRAM(MR4A08B–Everspin Technologies) Datasheet.

[9] EETIMES, “Samsung introduces working prototype of PRAM,” <http://www.eetimes.com> (2006/9/11), September 2006.

[10] J. Kim, H. Lee, S. Choi, and K. Bahng, “A PRAM and NAND Flash hybrid architecture for high-performance embedded storage subsystems,” in Proceedings of ACM & IEEE EMSOFT’ 08, October 2008.

[11] E. Kim, H. Shin, B. Jeon, S. Han, J. Jung, and Y. Won, “FRASH: Hierarchical file system for FRAM and flash,” in Proceedings of International Workshop on Data Storage Device and Systems, August 2007.

[12] I. H. Doh, J. Choi, D. Lee, and S. H. Noh, “Exploiting non-volatile RAM to enhance flash file system performance,” in Proceedings of ACM & IEEE EMSOFT’ 07, October 2007, pp. 164–173.

[13] Y. Park, S. Lim, C. Lee, K. Park, J. Kim, and K. Bahng, “PFFS: a scalable flash memory file system for the hybrid architecture of phase-change RAM and NAND flash,” in Proceedings of the 2008 ACM Symposium on Applied Computing, 2008, pp. 1498–1503.

[14] S. Baek, C. Hyun, J. Choi, D. Lee, and S. Noh, “Design and analysis of a space conscious nonvolatile-RAM file system,” in Proceedings of the IEEE Region 10 Conference.

[15] N. K. Edel, D. Tuteja, E. L. Miller, and S. A. Brandt, “MRAMFS: A compressing file system for non-volatile RAM,” in Proceedings of the 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems.

[16] “A protected and persistent RAM-based special filesystem in linux,” <http://pramfs.sourceforge.net/>.

[17] “Postmark-Benchmark,” <http://www.devone.org/linux/postmark.html>.



### 이형규

1999 동국대학교 컴퓨터공학과 학사  
 2001 서울대학교 전기컴퓨터공학부 석사  
 2007 서울대학교 전기컴퓨터공학부 박사  
 2007~현재 삼성전자 종합기술원 S/W선행연구소 전문 연구원  
 관심분야: 내장형 시스템, 저전력 시스템, 스토리

지 시스템

E-mail : hyunggyu.lee@samsung.com



### 이철

2000 KAIST 전기및전자공학과 학사  
 2002 KAIST 전자전산학과 석사  
 2008 KAIST 전자전산학과 박사  
 2008~현재 삼성전자 기술원 S/W선행연구소 OS 그룹 전문 연구원  
 관심분야: 컴퓨터 구조, 운영체제

E-mail : chur.lee@samsung.com