

# MRAM 기반 SSD 파일 시스템의 에너지 효율적 서브페이징\*

이재열\*\* · 한재일\*\*\* · 김영만\*\*\*\*

## Energy-Efficient Subpaging for the MRAM-based SSD File System\*

JaeYoul Lee\*\* · Jae-Il Han\*\*\* · Young-Man Kim\*\*\*\*

### ■ Abstract ■

The advent of the state-of-the-art technologies such as cloud computing and big data processing stimulates the provision of various new IT services, which implies that more servers are required to support them. However, the need for more servers will lead to more energy consumption and the efficient use of energy in the computing environment will become more important. The next generation nonvolatile RAM has many desirable features such as byte addressability, low access latency, high density and low energy consumption. There are many approaches to adopt them especially in the area of the file system involving storage devices, but their focus lies on the improvement of system performance, not on energy reduction. This paper suggests a novel approach for energy reduction in which the MRAM-based SSD is utilized as a storage device instead of the hard disk and a downsized page is adopted instead of the 4KB page that is the size of a page in the ordinary file system. The simulation results show that energy efficiency of a new approach is very effective in case of accessing the small number of bytes and is improved up to 128 times better than that of NAND Flash memory.

Keyword : Energy consumption, MRAM, Page, File system

논문투고일 : 2013년 10월 26일      논문수정완료일 : 2013년 12월 18일      논문게재확정일 : 2013년 12월 20일

\* 본 연구는 미래창조과학부 및 한국산업기술평가관리원의 산업융합원천기술 개발사업(정보통신)[10041753, 에너지 30% 이상 절감 가능한 범용 운영체제 핵심 원천 기술 개발]과 2013년도 국민대학교 교내연구비의 지원으로 수행되었음.

\*\* 국민대학교 대학원 컴퓨터공학과 석사과정

\*\*\* 국민대학교 컴퓨터공학과 교수, 교신저자

\*\*\*\* 국민대학교 컴퓨터공학과 교수

## 1. 서 론

클라우드 컴퓨팅이나 빅데이터 처리 등의 기술 발전에 따라 이의 활용이 점차 확산되고 있다[1]. 이러한 컴퓨터 환경의 등장은 다양하고 새로운 IT 서비스의 출현을 촉진시킬 것이며 사용자 증가와 이에 대응하기 위한 서버의 증설을 필요로 한다 [2]. 이에 따라 모바일 디바이스, PC, 데이터 센터의 서버 등 다양한 IT 기기에서의 에너지 소모가 증가할 것이며, 효율적인 에너지 사용에 대한 기술 즉 그린 컴퓨팅 기술의 발전이 요구된다. [3]은 이러한 에너지 절감을 위해서는 하드웨어보다는 소프트웨어 기술 개선이 더욱 중요한 것임을 보이고 있다.

최근 차세대 비휘발성 메모리를 활용한 하드웨어나 소프트웨어 시스템 성능 개선에 대한 연구가 활발히 진행되고 있다. 차세대 비휘발성 메모리는 대표적으로 MRAM(Magnetic Random Access Memory)을 포함해 PRAM(Phase Change RAM), ReRAM(Resistive RAM) 등이 있으며, 현재의 주 메모리인 RAM(Random Access Memory)과 같거나 근접한 빠른 읽기/쓰기 속도, 오랜 수명, 비휘발성, 바이트 단위 읽기/쓰기 가능 등의 장점을 제공한다. 차세대 비휘발성 메모리는 아직 기술적인 어려움으로 상용화되지 않았거나 고가인 이유로 크게 확산되지 않은 상태이나, 최근 PRAM, MRAM과 같은 차세대 비휘발성 메모리는 빠른 기술 발전을 보이고 있어 이에 따른 가격의 저하는 예상보다 빠르게 이 차세대 비휘발성 메모리들의 활용을 촉진할 것으로 예상된다[4].

현재 차세대 비휘발성 메모리를 활용한 파일 시스템이 많은 관심을 받고 있으며 대표적인 연구로 [5-7]를 들 수 있다. 그러나 이 연구들은 차세대 비휘발성 메모리 활용이 파일 시스템의 성능에 미치는 영향에 초점을 두고 있어 에너지 절감 기술은 다루고 있지 않으며 페이지 캐쉬의 페이지 크기가 기존 파일시스템과 같은 4KB를 사용한다.

본 연구는 에너지 절감을 위한 소프트웨어 기술

개선에 대한 것으로, 차세대 비휘발성 메모리의 하나인 MRAM 기반의 SSD(Solid State Disk)를 활용한 파일 시스템에서의 에너지 절감을 위해 기존의 페이지 기술을 개선한 서브페이징 기술을 제안한다. 또한 서브페이징 기반의 에너지 절감 모델에 대한 에너지 효율성을 측정하기 위해 Linux 커널의 저장장치 관련 입/출력 메커니즘과 디스크 시뮬레이션 도구인 DiskSim[8]을 심층 분석한 후, 시뮬레이션에 적합하도록 Linux 커널을 수정하고 DiskSim의 소스코드를 수정한 시뮬레이터를 만들었으며 대표적인 입/출력 시나리오에 대하여 실험 결과를 보였다.

본 논문의 구성은 다음과 같다. 제 2장은 비휘발성 메모리와 비휘발성 메모리를 활용한 관련 연구에 대하여 논하며, 제 3장은 에너지 절감을 위한 서브페이징 모델에 대하여 기술한다. 제 4장은 서브페이징 모델의 에너지 효율성 실험을 위한 시뮬레이터의 설계 및 구현에 대하여 설명하고, 5장은 시뮬레이션 실험 결과를 논하며, 제 6장은 결론을 기술한다.

## 2. 관련연구 및 문제점

본 장에서는 낸드플래시 메모리와 차세대 비휘발성 메모리의 기능 및 특징에 대해 소개하고, 낸드플래시 메모리와 차세대 비휘발성 메모리를 활용한 연구에 대하여 기술한다.

### 2.1 비휘발성 메모리 소개

#### 2.1.1 낸드플래시

낸드플래시는 오늘날 사용되는 가장 대표적인 비휘발성 메모리다. 이는 EEPROM(Electronically Erasable Programming Read-Only Memory)의 일종이지만, 기존 EEPROM에 비해 빠른 읽기, 쓰기 속도 및 대용량화가 쉽다는 장점을 가지고 있다. 또한 낸드플래시는 현재 보편적으로 사용되는 저장장치인 하드 디스크와 달리 기계적 움직임 없

이 전기적으로만 동작하므로 접근 속도가 빠르고, 전력소모가 적으며, 충격에 강하고, 크기나 모양에 대한 제약도 적은 편이다[9].

낸드플래시에서의 읽기, 쓰기는 페이지 단위로 이루어지고 한 페이지는 낸드플래시의 종류에 따라 512바이트~8K바이트의 크기를 갖는다. 한번 데이터가 써진 페이지에는 덮어쓰기를 할 수 없으며, 쓰기 연산을 위해서는 지우기 연산을 통해 데이터를 삭제한 후 써야 한다[10]. 한편 지우기 연산은 페이지 단위로는 할 수 없으며 블록 단위로만 가능하다. 읽기 연산은 약 25~60 $\mu$ s의 시간이 걸리며 쓰기 연산은 이보다 큰 200 $\mu$ s에서 800 $\mu$ s 정도의 시간이 걸린다. 지우기 연산이 가장 느리며 약 1.5~3ms의 시간이 걸린다. 삭제 및 지우기 연산은 일종의 트랜지스터 셀 내부의 플로팅게이트에 전하를 저장하거나 방출하는 형태로 이루어진다.

### 2.1.2 PRAM

PRAM(Phase Change RAM)은 OUM(Ovonic Unified Memory)라는 이름으로 미국 Ovonyx[11]가 처음으로 소개한 메모리 기술이다. CD-ROM이나 DVD-RAM과 같이 칼코게나이드 합금이라고 하는 특수한 얇은 박막 소재를 사용하고 있다[12]. 칼코게나이드 합금은 비결정 상태에서 저항치가 높아지고, 결정 상태에서는 저항치가 낮아지는 특성을 가진다. 이 2개의 상태를 제어해서 바꾸는 것으로 '1'과 '0'의 논리를 기억한다.

PRAM에서의 읽기, 쓰기는 바이트 단위로 이루어지고 바이트의 크기는 PRAM의 종류에 따라 2~64바이트의 크기를 갖는다. 또한 데이터가 써진 영역에 덮어쓰기가 가능하며, 삭제연산이 필요 없다. 읽기 연산은 약 50~70ns의 시간이 걸리며 쓰기 연산은 이보다 큰 90~400ns 정도의 시간이 걸린다[13, 14].

### 2.1.3 MRAM

MRAM(Magnetic Random Access Memory)

[15]은 자기저항 이라는 양자역학적 효과를 이용한 기억소자로서, 전원이 공급되지 않아도 정보가 지워지지 않는 차세대 비휘발성 메모리의 한 종류이다.

MRAM은 PRAM과 유사하게 읽기, 쓰기는 바이트 단위로 이루어지고 데이터가 써진 영역에 덮어쓰기가 가능하며, 삭제연산이 필요 없다. 그러나 최소 접근 단위는 8B이다. 읽기와 쓰기 연산은 32ns 정도의 시간이 걸린다. MRAM은 자기저항을 이용하기 때문에 PRAM과는 다르게 내구성에 제한이 없다[16].

## 2.2 비휘발성 메모리 기반 연구

비휘발성 메모리를 활용한 파일 시스템에 대한 연구는 낸드플래시 메모리를 포함한 비휘발성 메모리 기반의 SSD 사용에 대한 연구[5, 7, 17]와 주기억장치에서의 차세대 비휘발성 메모리 사용에 대한 연구[6]가 보이고 있다. PFFS[5]는 PRAM을 낸드플래시와 함께 사용하는 하이브리드 방식 파일 시스템으로써 이전 파일 시스템과 같이 사용자 데이터는 낸드플래시에 저장하고 기존의 낸드플래시에 저장하던 메타데이터를 PRAM에 저장하는 방법을 제시한다. PFFS는 데이터 영역에 기존 데이터가 있어도 덮어쓰기가 가능하고, 바이트 단위 접근이 가능한 PRAM을 사용하여 낸드플래시에서 발생한 오버헤드 문제를 크게 줄였다. PFFS는 최대 25%의 성능 향상이 된 것으로 나타났다. 그러나, PFFS는 단순히 메타데이터를 PRAM에 저장하는 것만을 제안했을 뿐, 잦은 메타데이터 업데이트로 인한 내구성 감소와 에너지 사용 문제를 보이고 있다. BPFSS[6]는 차세대 비휘발성 메모리인 PRAM을 캐시 메모리로 사용하는 파일 시스템이다. 캐시 메모리를 작은 바이트 단위로 접근이 가능하기 때문에, copy on write를 필요로 하지 않는다. 또한, BPFSS는 open, read, write, seek 과 같은 기존의 시스템 콜들을 수정하지 않고 그대로 사용이 가능하다. BPFSS는 기존 파일 시스템보다

성능을 13% 향상시켰다. 그러나, 메인 메모리와 저장장치에는 차세대 비휘발성 메모리를 적용하지 않았으며, 에너지 소비 문제 또한 고려하지 않았다. Moneta[7]는 비휘발성 메모리 기반 가상화 저장장치에 접근하기 위해 최근에 개발된 저장 시스템이다. 파일 시스템의 입/출력을 향상시키기 위해 비휘발성 메모리를 직접 접근하는 방법과 512B(Byte) 단위의 비휘발성 메모리 접근을 제안한다. Moneta는 기존 파일 시스템보다 지연시간을 60%, 대역폭을 7.2배 향상 시켰다. 그러나, 이 연구는 차세대 비휘발성 메모리 특성인 바이트 단위 접근을 완벽히 활용하지 않았고, 에너지 소비 문제 또한 고려하지 않았다. PTL[17]은 PRAM을 저장장치로 사용할 때 PRAM의 수명을 늘리기 위한 웨어 레벨링(Wear-leveling) 방법이다. PRAM은 덮어쓰기가 가능해 관리에 있어서 별도의 웨어 레벨링을 필요로 하지 않는다. 그러나 웨어 레벨링을 고려하지 않으면 지역성(Locality)으로 인해 같은 데이터 영역이 자주 접근될 가능성이 크게 되며 PRAM의 특성상 특정 데이터 영역이 많아 없어지게 된다. 이러한 문제를 해결하기 위해 PTL은 페이지 레벨 매핑 방법을 사용하고 메타데이터 영역에 내구성 정보를 추가하는 방식을 사용한다. PRAM은 바이트 단위 접근이 가능한데도 페이지 레벨 매핑 방법을 사용하는 이유는 바이트 단위 접근의 경우 바이트 매핑 정보만큼 추가적으로 내구성 정보가 필요하기 때문이다. 내구성에 대한 메타데이터를 추가적으로 저장하는 오버헤드는 전체 사용자 데이터 영역의 4%로 나타났다. 평균 성능은 낸드플래시에 비해 최대 300%까지 개선됐으며, 수명은 낸드플래시에 비해 25% 이상 개선된 것으로 나타났다. 그러나, 이 연구는 PRAM의 수명을 연장하는 방법만을 제안했을 뿐, 서브페이지를 활용한 에너지 소비에 대해서는 고려하지 않았다.

차세대 비휘발성 메모리의 바이트 단위 읽기/쓰기 기능은 차세대 비휘발성 메모리 기반 SSD와의 입/출력 데이터의 크기를 줄여 에너지 절감에 도

움이 될 수 있다. 그러나 위 연구들은 페이지 크기가 기존의 파일 시스템과 같은 4KB이며 에너지 절감에 초점을 두지 않아 이러한 가능성을 다루고 있지 않다. 본 연구는 차세대 비휘발성 메모리의 장점을 활용하여 입/출력 데이터의 크기를 줄이기 위해 최초로 페이지 크기를 4KB보다 작은 서브페이지 방법을 제시하고 시뮬레이션을 통해 에너지 절감이 가능함을 보였다.

### 3. 에너지 효율적 서브페이징

낸드플래시가 하드 디스크에 비해 에너지를 적게 소비하고 접근 속도 또한 빠르기 때문에 최근 낸드플래시를 저장장치로 많이 사용하고 있다. 그러나, 낸드플래시 또한 에너지를 낭비하게 하는 두 가지 특성이 있다. 첫째는 쓰기전 지우기(Erase Before Write)이다. 낸드플래시는 데이터가 이미 존재하는 영역에는 덮어쓰기가 불가능하며, 데이터가 이미 존재하는 영역을 쓰기 위해서는 반드시 데이터 영역을 삭제한 후 사용해야 한다. 이런 동작은 결국 소비하지 않아도 될 에너지를 소비 하는 결과를 초래하게 된다. 둘째는 페이지 단위 접근이다. 낸드플래시의 접근속도는 하드 디스크보다 빠르지만 최소 접근 단위는 여전히 페이지이다. [그림 1]과 같이 응용 프로그램이 페이지크기보다 작은 소수의 바이트 쓰기를 요청하더라도 파일 시스템은 반드시 최소 접근단위인 페이지로 낸드플래시를 접근 해야만 한다. 이 경우 페이지에서 변경되지 않은 부분만큼 에너지를 낭비하게 된다. 위와 같은 에너지 낭비 문제를 개선하기 위해 본 연구는 MRAM의 장점인 바이트 단위 접근 특성을 이용한다.

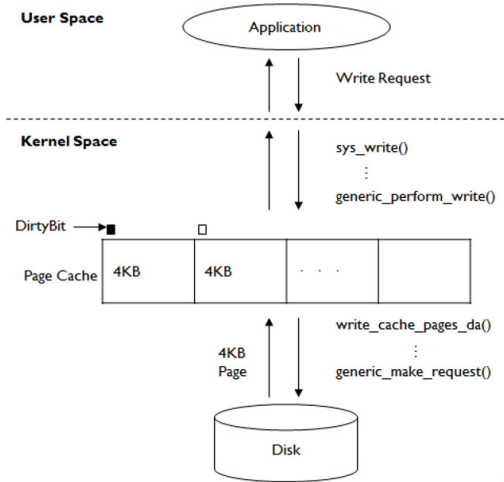
[그림 2]와 같이 페이지를 보다 작은 크기의 서브페이지로 분할하여 응용 프로그램이 실제로 접근한 서브페이지만을 저장장치로 요청하게 된다. 바이트 단위 접근을 그대로 사용하지 않고 페이지를 서브페이지로 나누어 접근하는 이유는 페이지

접근 횟수와 메모리 오버헤드가 트레이드 오프관계에 있기 때문이다. 페이지 크기가 작아질수록 실제 변경된 페이지를 쓰기에 요구되는 크기와 에너지는 감소하는 반면 더티 비트(Dirty Bit)와 페이지 테이블에 요구되는 메모리가 증가하는 오버헤드가 나타난다. 그러나 미래의 주메모리 용량은 4GB 이상의 대용량일 것이며 서브페이징에 요구되는 수십 MB 이하의 오버헤드는 에너지 절감과

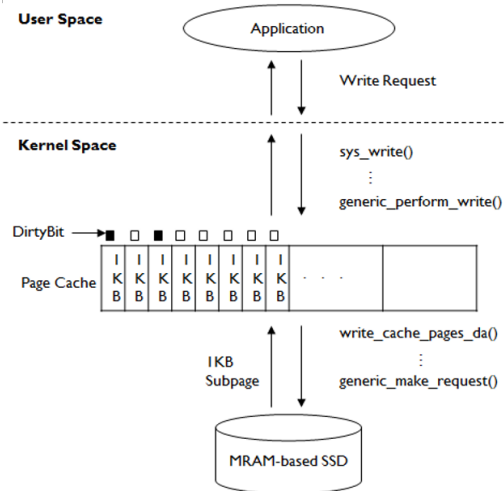
같은 이점을 고려할 때 감내할 만한 수준으로 예상된다. 본 연구에서는 이러한 추세를 감안하고 의미 있는 페이지 접근 단위를 유지하기 위해 최소 페이지 크기를 기존 페이지의 1/64인 64B로 제한하였다.

### 4. 시뮬레이터 설계 및 구현

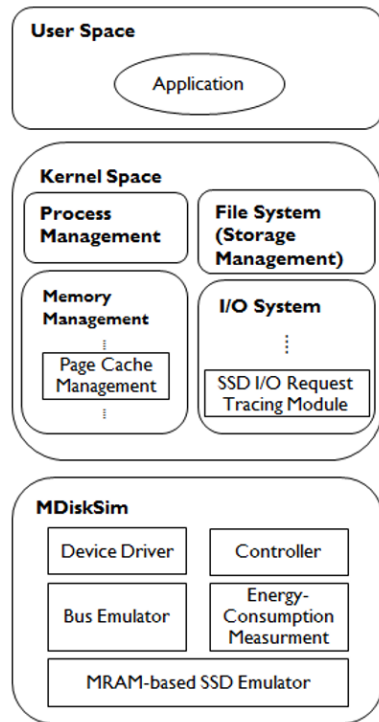
위에서 제시한 에너지 효율적 서브페이징 모델의 실제 효용성을 측정하기 위해 비휘발성 메모리 MRAM 기반의 SSD 이뮬레이터와 입/출력 요청을 하는 응용 프로그램, 응용 프로그램으로부터 입/출력 요청을 받아 서브페이징 모델을 파일 시스템에 실제 실현한 운영체제가 필요하다. 본 연구는 이러한 목적으로 MRAM 기반의 SSD 이뮬레이터(Emulator)인 MDiskSim을 구현하고, 서브페이징 모델이 실현되도록 Linux 운영체제의 파일



[그림 1] 기존 페이징 기법



[그림 2] 서브페이징 기법



[그림 3] 시뮬레이션 시스템 구성도

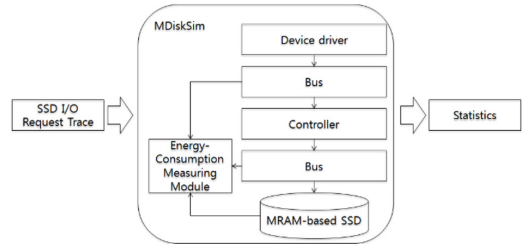
시스템과 관련된 커널(Kernel) 소스코드를 수정하였으며, 실험 결과를 얻기 위한 입/출력 응용 프로그램을 작성하였다. 본 장에서는 시뮬레이터의 핵심인 MDiskSim과 수정된 Linux 파일 시스템의 입/출력 부분에 중점을 두고 시뮬레이터의 설계 및 구현에 대하여 설명한다.

시뮬레이션 시스템의 전체 구성은 [그림 3]과 같다. 응용 프로그램이 어떤 데이터의 쓰기 요청을 발생시키면 데이터가 페이지 캐시에 기록되고 쓰기 요청이 종료된다. 서브페이징 기법에 의해 요청은 작은 크기의 서브페이지 단위로 더티 비트가 설정된다. 이후 응용 프로그램이나 운영체제에 의해 서브페이지가 디스크로 플러시 될 때, SSD 입/출력 추적 모듈은 더티 비트가 설정된 서브페이지들에 대한 입/출력 요청 정보를 기록한다. 서브페이지들에 대한 입/출력 요청정보는 MDiskSim의 입력으로 주어지며, MDiskSim은 MRAM 기반 SSD 이물레이터를 사용하여 요청에 대한 입/출력 동작을 시뮬레이션하고 에너지 사용량을 측정한다. 각 모듈의 자세한 설명은 다음과 같다.

#### 4.1 MDiskSim

DiskSim은 최소 접근 단위가 페이지인 하드 디스크나 낸드플래시 기반 SSD는 시뮬레이션 할 수 있지만, 최소 접근 단위가 바이트인 차세대 비휘발성 메모리는 시뮬레이션 할 수 없기 때문에 본 연구는 차세대 비휘발성 메모리를 위한 MDiskSim을 구현하였다. [그림 4]는 에너지 측정 모듈(Energy-Consumption Measuring Module), SSD 입/출력 요청 추적 모듈(SSD I/O Request Tracing Module), MRAM 기반 SSD 이물레이터를 DiskSim에 추가한 MDiskSim의 구조를 보여준다. 시뮬레이션을 위해서 MDiskSim은 SSD 입/출력 요청 추적 모듈의 입/출력 요청 정보(SSD I/O Request Trace)를 입력 값으로 받는다. 시뮬레이터의 에너지 측정 모듈은 각 장치에서 소비하는 에너지를 측정한다. 시뮬레이션이 끝나면 각 장치의 에

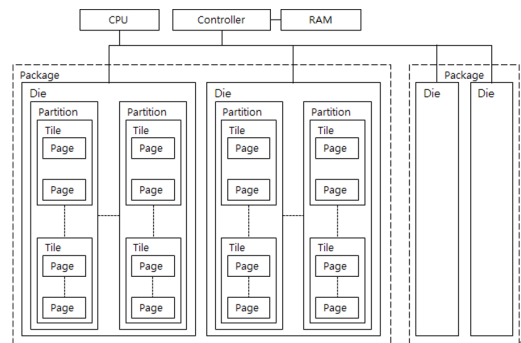
너지 소비량, 접근 횟수, 접근 시간 등을 출력한다.



[그림 4] MDiskSim의 구조

#### 4.2 MRAM 기반 SSD 이물레이터

MRAM 파일 시스템 모델은 MRAM 기반 SSD 시뮬레이터 설계 및 구현을 통해 도출되었다. MRAM에서는 하드웨어적으로 페이지 개념이 없으며 워드 기반(Word Line)과 비트 기반(Bit Line)으로 접근을 한다. 최소 구성단위는 타일이고, 하나의 파티션은 64개의 타일로 이루어지며, 하나의 MRAM은 16개의 파티션으로 이루어진다. 그러나 본 연구는 개념적 모델을 제시하는 것이기 때문에 낸드플래시 기반 SSD 구조에 MRAM의 구성과 특성을 적용한 구조를 제안하였다[그림 5].



[그림 5] MRAM 기반 SSD의 구조

위에 언급하였듯이 MRAM 자체는 페이지 개념이 없으나 기존 파일 시스템과의 호환성 및 관리를 위해 논리적 페이지 개념을 적용하였다. MRAM 기반 SSD 이물레이터는 불필요한 복잡성을 피하

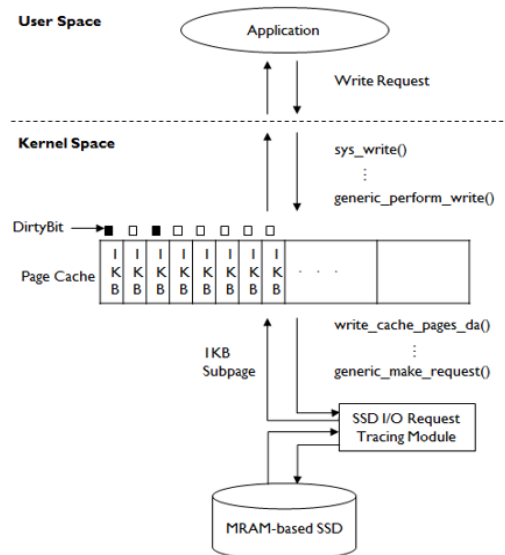
기 위해 본 연구와 관련된 MRAM의 특성만을 반영하여 설계 및 구현되었다.

### 4.3 SSD 입/출력 요청 추적 모듈

MRAM은 기존 하드 디스크와 낸드플래시와는 다르게 DRAM과 같이 바이트 단위 주소 접근이 가능하다. 그러나 기존 파일 시스템의 최소 접근 단위는 페이지이며 일반적인 페이지의 크기는 4KB이다. 기존 SSD 입/출력 요청 추적 모듈의 경우 디바이스 드라이버에서 보조 기억장치로의 접근 요청만을 기록하기 때문에 응용 프로그램이 페이지보다 작은 소수의 바이트 단위 읽기/쓰기 요청을 하더라도 실제 바이트 단위 요청을 기록하지 못하고 페이지 단위 접근 요청을 기록한다. 그러나 MRAM의 특성인 바이트 단위 주소를 활용하기 위해서는 반드시 응용 프로그램의 바이트 단위 접근 요청정보를 알 수 있어야 한다. 이런 바이트 단위 접근 요청정보를 기록하기 위해 본 연구는 응용 프로그램이 디스크로 요청을 플러시(Flush)하기 전까지 모든 입/출력 시스템 콜의 페이지 접근정보를 기록한다. 입/출력 시스템 콜의 파라미터는 페이지 내부 오프셋 정보와 실제 요청크기를 포함한다. 페이지 내부 오프셋 정보와 실제 요청 크기정보를 활용하여 입/출력 시스템 콜이 해당 페이지의 몇 번째 서브페이지를 접근하는지를 비트맵 플래그(bitmap flag)로 기록한다. 구체적으로, Linux에서는 응용 프로그램이 쓰기 요청을 할 경우 `sys_write()` 함수부터 시작하여 여러 커널 함수를 거쳐 페이지 캐시에 실제 쓰기 작업을 수행하는 `generic_perform_write()` 함수를 호출한다. `generic_perform_write()` 함수에서는 파일 구조체 포인터와, 위치를 바탕으로 접근 할 페이지 내의 오프셋과 요청 크기를 계산한다. 이 때, 기존 페이지 기법에서는 접근한 페이지의 더티 비트만 설정하는 반면, 제안하는 서브페이징 기법에서는 페이지를 보다 작은 몇 개의 고정된 크기로 분할하고, 접근한 서브페이지의 더티 비트를 설정한다. 위의

과정을 보조 기억장치로 플러시되기 전까지 반복한다. 플러시 될 때에는 `write_cache_pages_da()` 함수부터 `generic_make_request()` 함수가 디바이스 드라이버에서 호출된다. 이 때, SSD 입/출력 요청 추적 모듈은 더티 비트가 설정된 서브페이지에 대한 입/출력 요청 정보만 기록한다.

[그림 6]은 요청이 발생했을 때 SSD 입/출력 요청 추적 모듈이 요청을 기록하는 위치를 보여준다. 예를 들어, 4KB 크기의 페이지를 256B 크기의 16개 서브페이지로 분할하고, 응용 프로그램의 입/출력 시스템 콜이 오프셋 10, 요청크기 100바이트로 페이지에 읽기/쓰기를 하면, SSD 입/출력 요청 추적 모듈은 계산을 통해 입/출력 시스템 콜이 0번째 서브페이지를 접근한다는 것을 인지하고, 0번째 서브페이지의 비트맵 플래그를 설정한다. 계속해서 응용 프로그램의 입/출력 시스템 콜이 오프셋 256, 요청크기 300바이트로 페이지를 읽기/쓰기를 하면, 1번째와 2번째 서브페이지의 비트맵 플래그를 설정한다. 위의 과정을 페이지가 보조 기억장치로 플러시되기 전까지 반복한다. 플러시된 요청들은 MDiskSim의 매개변수로 입력된다.



[그림 6] SSD 입/출력 요청 추적 모듈

## 5. 실험 결과

본 장에서는 실험 환경과 MDiskSim을 사용하여 시뮬레이션 한 실험 결과에 대해 설명한다.

### 5.1 에너지 효율성 측정

에너지 소비 측정 모듈은 에너지 소비를 측정하기 위해서는 SSD 외부버스, SSD 내부버스, MRAM 칩 등에서 사용되는 모든 에너지를 측정한다. E를 에너지, P를 전력, T를 시간이라 할 때 일반적으로 에너지 소비는 다음 식에 의해 계산된다.

$$E = P \times T$$

따라서 응용 프로그램의 입출력 요청에 따라 페이지 캐시의 페이지가 MRAM 기반 SSD로 입/출력될 때 소비되는 총 에너지는 다음 식에 의하여 계산할 수 있다.

$$E = I_B V_B \sum_1^n T_{Bn} + V_P \left( I_R \sum_1^k T_{Rk} + I_W \sum_1^m T_{Wm} + I_{SB} \sum_1^l T_{SBi} + I_I \sum_1^j T_{Ij} \right)$$

E = MRAM에서 사용한 총 에너지

$I_B$  = 외부 버스에 인가된 전류

$V_B$  = 외부 버스에 인가된 전압

$T_B$  = 버스를 통한 데이터 전송시간

$V_P$  = MRAM에서 인가된 전압

$I_R$  = MRAM에서 읽기 동작 때 인가된 전류

$T_R$  = MRAM에서 읽기 동작이 걸린 시간

$I_W$  = MRAM에서 쓰기 동작 때 인가된 전류

$T_W$  = MRAM에서 쓰기 동작이 걸린 시간

$I_{SB}$  = SSD 내부 버스를 통해 데이터를 전송 할 때 인가된 전류

$T_{SB}$  = SSD 내부 버스를 통한 데이터 전송시간

$I_I$  = 유휴 시간 동안 인가된 전류

$T_I$  = 유휴 시간

본 연구에서는 각 저장장치의 쓰기 연산 에너지 사용량만을 측정 및 비교하기 때문에 실제 계산에서는 유휴 에너지와 읽기 연산 에너지는 제외한다. 또한, 모델을 단순화하기 위해 하드웨어의 중요한 전기적 특성만을 고려한다.

### 5.2 실험 환경

MDiskSim은 MRAM의 특성을 입력 값으로 받아 사용 에너지를 측정한다. 낸드플래시[18]와 MRAM[16]의 페이지 크기는 4KB이며, 용량은 32GB이다. 32GB MRAM은 실제로 존재하지 않지만 낸드플래시와 같은 실험환경을 위해 임의로 용량을 설정하였다. <표 1>은 낸드플래시와 MRAM의 사양을 보이고 있으며 MRAM의 사양은 공개된 자료를 기반으로 추출하였다. <표 1>에서 보듯이 MRAM은 삭제 연산이 없고, 읽기와 쓰기 연산에서 낸드플래시보다 적은 지연시간을 가지고 있다.

<표 1> 낸드플래시 메모리와 MRAM의 특성

Characteristics	NAND	MRAM
Page(KB)	4	4
Capacity(GB)	32	32
Chip xfer latency(mS)	0.000025	0.000025
Page read latency(mS)	0.025	0.016384
Page write latency(mS)	0.2	0.016384
Block erase latency(mS)	1.5	N/A
Chip input Voltage(V)	3.3	3.3
Page read current(mA)	25	60
Page write current(mA)	25	152
Block erase current(mA)	25	N/A

본 실험에서는 저장장치에 이상적인 작은 바이트 단위 접근과 실제 접근 비교를 위해 가상 테스트 워크로드와 실제 워크로드로 시뮬레이션 하였다. <표 2>는 테스트 프로그램에서 인위적으로 생성한 워크로드이다. 테스트 프로그램은 이상적인 작은 바이트 단위 접근을 발생시키기 위해 페이지



당 각각 64B, 128B, 256B, 512B, 1024B 쓰기를 하며, 결과 비교를 쉽게 하기 위해 요청 횟수를 동일하게 20개로 제한하였다. <표 3>은 실제 응용 프로그램을 실행시켜 얻은 워크로드이다. Load1은 Youtube를 이용하여 41분간 동영상 스트리밍을 재생, Load2는 GrooveShark를 이용하여 48분간 음악 스트리밍을 재생, Load3는 38분간 파이어폭스를 이용한 웹브라우저, Load4는 Dropbox를 이용한 파일 동기화이고, Load5는 벤치마킹 툴인 PostMark를 사용하여 얻은 워크로드이다. PostMark는 웹 메일 서버를 벤치마킹하기 위한 툴로써 작은 바이트 단위 접근에 대해 벤치마킹이 가능하다. Load5의 경우 요청사이즈는 페이지당 4B~2048B, 파일 수는 50000, 트랜잭션 수는 50000으로 설정하였다.

<표 2> 테스트 워크로드

워크로드 (요청크기)	Trace duration(mS)	Total Requests handled	Average number of I/O per sec
64B	0.43	20	0.0075
128B	0.43	20	0.0075
256B	0.43	20	0.0075
512B	0.43	20	0.0075
1024B	0.43	20	0.0075

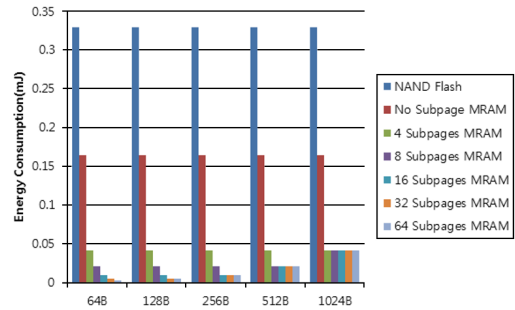
<표 3> 실제 워크로드

워크로드	Trace duration(Min)	Total Requests handled	Average number of I/O per sec
Load1	41	75484	21.82
Load2	48	15115	4.36
Load3	38	108494	31.36
Load4	7.3	56065	80.07
Load5	15	8132	2.35
Load6	7	56065	80.07

모든 워크로드들은 SSD 파일 시스템 추적 모듈을 사용하여 수집하였다.

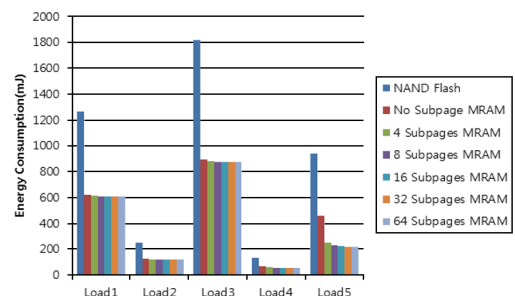
### 5.3 에너지 소비 비교

테스트 워크로드와 실제 워크로드를 MDiskSim을 사용하여 시뮬레이션한 결과는 다음과 같다.



[그림 7] 테스트 워크로드 에너지 비교

[그림 7]은 테스트 워크로드에서 낸드플래시의 에너지 사용량과 MRAM 기반 SSD를 사용한 서브페이징 모델의 에너지 사용량을 보여준다. 이 실험에서는 서브페이징 모델이 낸드플래시보다 최대 128배까지 에너지를 적게 소모하는 것으로 나타났다. 페이지당 요청 바이트 크기가 작고 페이지당 서브페이지가 많을수록 에너지를 적게 사용한다. 이 결과는 서브페이징 모델에서는 페이지당 요청 바이트 크기가 작을수록 에너지를 절감할 수 있음을 보여준다.



[그림 8] 실제 워크로드 에너지 비교

[그림 8]은 실제 워크로드에서 낸드플래시와 서브페이징 모델의 에너지 사용량을 보여준다. 이 실험에서도 서브페이징 모델이 낸드플래시보다 에

너지를 적게 소모하는 것으로 나타났다. 그러나, 서버페이징 모델의 에너지 사용량을 비교해 보면 차이가 많이 나지 않는 것을 확인 할 수 있다. 결과 값이 차이가 많이 나지 않는 이유는 실제 워크로드의 경우 응용 프로그램이 일반적인 파일시스템을 통해 4KB 크기의 페이지 단위로 접근하도록 최적화 되어있어 바이트 단위 접근의 이점을 살릴 수 없기 때문이다.

## 6. 결 론

본 논문에서는 MRAM 기반 SSD 파일 시스템을 위한 에너지 절감 모델을 제안하였다. MRAM 기반 SSD 파일 시스템 모델은 기존 파일 시스템과는 다르게 바이트 단위 접근이 가능하다. 본 연구는 이러한 특성을 활용하여 서버페이징 기반의 에너지 절감 모델을 제시하였다.

기존 파일 시스템은 실험 결과 MRAM 기반 SSD를 활용하더라도 에너지 소비 효율이 좋지 않았다. 그러나 바이트 단위 접근을 고려한 서버페이징 기반의 에너지 절감 모델은 소수의 바이트 단위 접근이 많은 경우 에너지 효율성이 높아짐을 보였다.

본 연구는 기존의 파일 시스템과 달리 MRAM의 바이트 단위 접근 특성을 활용하기 위해 입/출력 단위 페이지의 크기를 소형화한 서버페이징 모델을 제시하여 에너지 절감의 가능성을 최초로 보였다는 측면에서 의미가 있다. 그러나 본 연구의 결과는 시뮬레이션을 통해 얻은 결과로써 MRAM 기반의 SSD를 실제 하드웨어로 만들어 시스템에 적용할 때 나타나는 문제는 예상할 수 없고, 에너지 측정을 단순화하기 위해 유틸리티 에너지와 읽기 연산 에너지를 제외하는 등 추후 보다 정밀하게 소비 에너지를 측정할 수 있는 방법이 필요하며, 이러한 문제는 에너지 절감의 효율성에 영향을 미칠 수 있다. 또한 본 연구는 MRAM 기반 SSD와 페이지 크기의 소형화를 통한 입/출력 최적화에 초점을 두었을 뿐 비휘발성 메모리의 주메모리 활용, 메모리 계층구조의 개선 등의 여러 방법을 통한 종합적이고 체

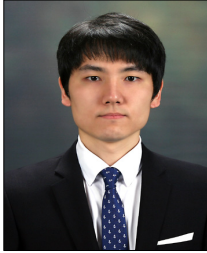
계적인 에너지 절감은 다루고 있지 않다. 앞으로 페이지 크기를 소형화함에 따라 부수될 수 있는 성능 문제, 서버페이징과 동시에 주메모리에 비휘발성 메모리를 사용할 경우 에너지 절감 및 성능에 미치는 영향 등에 대한 많은 연구가 필요하다.

## 참 고 문 헌

- [1] Agrawal, D., S. Das, and A. El Abbadi, "Big data and cloud computing : current state and future opportunities", In : *Proceedings of the 14th International Conference on Extending Database Technology, ACM*, (2011), pp.530-533.
- [2] Armbrust, M. et al., "A view of cloud computing", *Communications of the ACM*, Vol. 53, No.4(2010), pp.50-58.
- [3] Bellasi, P. et al., "Constrained power management : application to a multimedia mobile platform", In : *Proceedings of the Conference on Design, Automation and Test in Europe, European Design and Automation Association*, (2010), pp.989-992.
- [4] Song, Y. J., G. Jeong, G. I. Baek, and J. D. Choi, "What Lies Ahead for Resistance-Based Memory Technologies?", *IEEE Computer*, (2013), pp.30-36
- [5] Park, Y. W. et al., "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.
- [6] Condit, J. et al., "Better I/O through byte-addressable, persistent memory", In : *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles*. (2009), pp.133-146.

- [7] Caulfield, A. M. et al., "Providing safe, user space access to fast, solid state disks", In : *ACM SIGARCH Computer Architecture News*, (2012), pp.387-400.
- [8] <http://www.pdl.cmu.edu/DiskSim/>.
- [9] Barr, M., *Embedded Systems Programming*, O'Reilly, (2001), pp.103-104.
- [10] Lee, J. D., S. H. Hur, and J. D. Choi, "Effects of floating-gate interference on NAND flash memory cell operation", *Electron Device Letters, IEEE*, Vol.23, No.5(2002), pp. 264-266.
- [11] <http://www.ovonyx.com>.
- [12] Ovshinsky, S., "Amorphous Materials-the Key to New Devices", *IEEE Proc. Of CAS*, Vol.1(1998), p.33
- [13] Benjamin, C., E. Lee, I., O. Mutlu, and D. Burger, "Architecting Phase Change Memory as a Scalable DRAM Alternative", *Proceedings of the 36th International Symposium on Computer Architecture (ISCA)*, Austin, TX, (2009), pp.2-13,
- [14] Choi, Y. D. et al., "A 20nm 1.8 V 8Gb PRAM with 40MB/s program bandwidth", In : *Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, (2012), pp.46-48.
- [15] Engel, B. N. et al., "A 4-Mb toggle MRAM based on a novel bit and switching method", *Magnetics, IEEE Transactions on*, Vol. 41, No.1(2005), pp.132-136.
- [16] Everspin MR4A16B Datasheet.
- [17] Choi, G. S., B. W. On, K. H. Choi, and S. W. Yi, "PTL : PRAM Translation Layer", *Microprocessors and Microsystems*, Vol.37 (2013), pp.24-32.
- [18] Samsung, K9XXG08XXM Datasheet.

## ◆ 저 자 소 개 ◆

**이 재 열 (jylee195@gmail.com)**

국민대학교 컴퓨터 공학과를 졸업하고, 동 대학원에서 컴퓨터공학 전공으로 석사과정에 재학 중이다. 주요 관심분야는 정보보안, 클라우드 컴퓨팅, 운영체제 등이다.

**한 재 일 (jhan@kookmin.ac.kr)**

연세대학교에서 이학사, 미국 Syracuse University에서 전산학 석사와 박사학위를 취득하고, 국민대학교 컴퓨터학부 교수로 재직 중이다. 현재 분산처리, 빅데이터, 클라우드 시스템, 객체지향 시스템과 RFID/USN 미들웨어를 연구 중이다. 관심분야는 서비스 사이언스, 서비스 컴퓨팅, 분산 시스템, 빅데이터, 클라우드 시스템, 객체지향 시스템, 미들웨어, 컴퓨터 및 네트워크 보안, 지능형 시스템, 공개 소프트웨어 등이다.

**김 영 만 (ymkim@kookmin.ac.kr)**

서울대학교 기계공학과에서 학사, Ohio State University 전산과학과에서 석사와 박사학위를 취득하고, 국민대학교 컴퓨터공학과 교수로 재직 중이다. 현재 컴퓨터 네트워크와 미들웨어를 연구 중이다. 관심분야는 컴퓨터 및 네트워크 보안, SaaS 플랫폼, USN 등이다.