

# 하이브리드 하드디스크를 위한 효율적인 데이터 블록 교체 및 재배치 기법

## (An Efficient Data Block Replacement and Rearrangement Technique for Hybrid Hard Disk Drive)

박 광 희 <sup>†</sup>                      이 근 형 <sup>†</sup>                      김 덕 환 <sup>\*\*</sup>  
(Kwanghee Park)              (Geunhyung Lee)              (Deok-Hwan Kim)

**요 약** 최근 낸드 플래시 메모리가 하드디스크 수준으로 읽기 성능이 향상되고, 전력소비가 훨씬 적음에 따라, 플래시메모리와 하드디스크를 같이 사용하는 하이브리드 하드디스크와 같은 이기종 저장장치들이 출시되고 있다. 하지만 낸드 플래시 메모리의 쓰기 및 삭제 속도가 기존 자기디스크의 쓰기 성능에 비해 매우 느릴 뿐 아니라, 사용자 층에서 쓰기 요청이 집중될 경우 CPU, 메인 메모리에 심각한 오버헤드를 발생시킨다. 본 논문에서는 비휘발성 캐시의 역할을 하는 낸드 플래시 메모리의 성능을 향상시키기 위해 읽기의 참조 빈도는 낮고, 쓰기의 갱신 빈도가 높은 데이터 블록들을 교체하는 LFU(Least Frequently Used)-Hot 기법을 제시하고, 교체 될 데이터 블록들을 재배치하여 자기디스크로 플러싱하는 기법을 제시한다. 실험 결과, 본 논문에서 제안하는 LFU-Hot 블록 교체 기법과 멀티존 기반의 데이터 블록 재배치 기법 실행시간이 기존 LRU, LFU 블록 교체 기법들보다 입출력 성능 면에서 최대 38% 빠르고, 비휘발성 캐시의 수명을 약 40% 이상 향상 시킴을 증명하였다.

**키워드** : 하이브리드 하드디스크, 낸드 플래시 메모리, 블록 교체 기법, 데이터 블록 재배치 기법, LFU-Hot

**Abstract** Recently heterogeneous storage system such as hybrid hard disk drive (H-HDD) combining flash memory and magnetic disk is launched, according as the read performance of NAND flash memory is enhanced as similar to that of hard disk drive (HDD) and the power consumption of NAND flash memory is reduced less than that of HDD. However, the read and write operations of NAND flash memory are slower than those of rotational disk. Besides, serious overheads are incurred on CPU and main memory in the case that intensive write requests to flash memory are repeatedly occurred. In this paper, we propose the Least Frequently Used-Hot scheme that replaces the data blocks whose reference frequency of read operation is low and update frequency of write operation is high, and the data flushing scheme that rearranges the data blocks into the multi-zone of the rotation disk. Experimental results show that the execution time of the proposed method is 38% faster than those of conventional LRU and LFU block replacement schemes in I/O performance aspect and the proposed method increases the life span of Non-Volatile Cache 40% higher than those of conventional LRU, LFU, FIFO block replacement schemes.

**Key words** : Hybrid Hard Disk Drive, NAND Flash Memory, Block Replacement, Data Rearrangement, LFU-Hot

· 이 논문은 지식경제부와 한국산업기술진흥원의 전략기술인력양성사업 및 ETRI, 시스템반도체진흥센터에서 수행한 IT SoC 핵심설계인력양성사업, 그리고 2008년도 정부재원(교육인적자원부 학술연구조성사업비)으로 한국학술진흥재단의 지원을 받아 연구되었음(KRF-2008-313-D00822)

<sup>†</sup> 학생회원 : 인하대학교 전자공학과  
khpark@iesl.inha.ac.kr  
ghlee@iesl.inha.ac.kr

<sup>\*\*</sup> 정 회원 : 인하대학교 전자공학부 교수  
deokhwan@inha.ac.kr  
(Corresponding author임)

논문접수 : 2009년 1월 30일  
심사완료 : 2009년 10월 15일

Copyright©2010 한국정보과학회: 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.  
정보과학회논문지: 컴퓨팅의 실제 및 레터 제16권 제1호(2010.1)

## 1. 서론

하드디스크의 성능은 CPU, 메모리의 성능과는 큰 차이가 존재해 컴퓨터시스템에서 병목현상을 발생시키는 가장 주된 요인이 되고 있다[1]. 따라서 저장장치의 성능을 향상시키기 위해 기존의 자기디스크와 함께 읽기 동작에서 탁월한 성능을 보여주는 낸드 플래시 메모리를 비휘발성 캐시(NVCache: Non-Volatile Cache)로 사용하는 하이브리드 하드디스크와 같은 이기종 저장장치가 등장하고 있다. 그림 1과 같이 하이브리드 하드디스크(H-HDD: Hybrid Hard Disk Drive)는 낸드 플래시 메모리를 2차 비휘발성 캐시로 사용하여 하드디스크 스핀들 모터의 동작 시간을 감소시켜 모바일 컴퓨터의 전력소비를 최소화하고 충격에 강한 모바일 환경을 구성하는데 효율적이다[2-4].

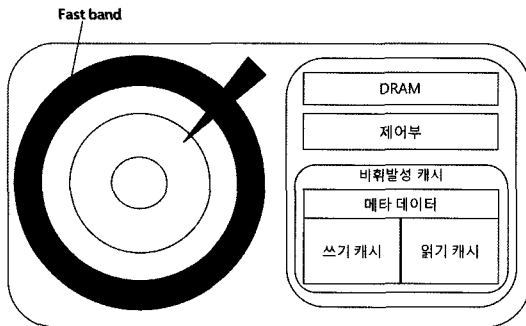


그림 1 하이브리드 하드디스크 구조

하지만 하이브리드 하드디스크를 구성하면서 다음과 같은 세 가지의 문제점을 고려해야 한다.

첫째, 운영체제에서 사용하는 페이지 캐시는 보편적으로 다양한 LRU(Least Recently Used) 기법을 사용하여 페이지 캐시의 내용을 갱신한다. 따라서 2차 캐시인 비휘발성 캐시에서도 LRU와 같은 유사 블록 교체 기법을 사용하는 경우 이중 캐싱 현상으로 인하여 자기디스크의 에너지 소비를 감소시키지 못한다. 그러므로 캐시 공해(Cache Pollution)로 인해 성능 향상 또한 미미할 수 밖에 없다[5,6]. 따라서 하이브리드 하드디스크를 사용하는 모바일 컴퓨터의 성능 향상 및 저전력 소비를 위해서는 2차 캐시인 비휘발성 캐시 내에 적합한 블록 교체 기법이 필요하다.

둘째, 비휘발성 캐시의 쓰기 및 삭제 속도가 기존 자기디스크의 쓰기 성능에 비해 상대적으로 느리므로 호스트로부터 쓰기 요청이 집중될 경우 자기 디스크에 직접 쓰는 기법을 취하고 있다[5]. 이러한 경우 자기디스크의 스핀들 모터로 인한 전력 낭비는 불가피하다. 다시 말해서 비휘발성 캐시의 수명과 전력 소비의 상관관계

를 극복해야 할 필요성이 있는데 그 해결 방법으로는 쓰기 캐시 내의 데이터를 선별하여 전송량을 감소시키는 방법이 있다. 이는 쓰기 캐시의 데이터 역시 차후 읽기 빈도가 높아 읽기 캐시로 전환될 가능성이 있고, 갱신 횟수가 낮아 자기디스크에 갱신할 필요가 없는 데이터가 존재하기 때문이다. 이러한 데이터들은 임시파일과 같은 일회성 파일들, 시스템 부팅 로그들이 있다.

셋째, 기존의 하이브리드 하드 디스크 기법은 비휘발성 읽기 캐시 미스로 인해 데이터를 자기디스크로부터 읽어와야 할 경우 스핀들 모터가 활성화되면서 데이터를 비휘발성 읽기 캐시로 반입하게 된다. 이와 함께 전력 소비를 최소화하기 위해 비휘발성 쓰기 캐시에 수정되거나 새로 저장된 데이터도 전부 자기디스크에 반영한다. 이러한 기법을 플러싱(Flushing)이라 한다[7]. 이러한 플러싱 기법은 단순히 비휘발성 쓰기 캐시의 내용을 일괄적으로 자기디스크에 반영하는 수준에 머무르고 있다. 이는 비휘발성 캐시 내의 참조정보를 전혀 반영하지 않아 차후 자기디스크의 입출력 성능을 감소시키는 문제가 발생한다.

본 논문에서는 페이지 캐시와 동일한 데이터를 중복 저장하는 이중 캐싱을 발생시키지 않으면서 비휘발성 캐시의 성능을 향상시킬 수 있는 LFU(Least Frequently Used)-Hot이라는 데이터 블록 교체 기법을 제시하고, 이를 통해 교체되는 데이터 블록을 재배치하여 자기디스크로 전송하는 기법을 제시한다.

본 논문의 구성은 2장에서 하이브리드 하드디스크에 관련된 연구를 살펴보고 3장에서는 제안하는 LFU-Hot 블록 교체 기법과 멀티존 기반의 데이터 재배치 기법에 대해서 설명하며 4장에서 구현 및 실험 결과를 분석하고 5장에서 결론을 맺는다.

## 2. 관련 연구

### 2.1 운영체제의 페이지 캐시와 2단계 비휘발성 캐시

#### 2.1.1 운영체제의 페이지 캐시

현대의 대부분의 컴퓨터 저장장치는 휘발성인 주 메모리(RAM)와 2차 저장장치인 하드디스크로 나눌 수 있다. 운영체제에서 사용하고 있는 페이지 캐시 기법은 재 참조될 가능성이 큰 하드디스크의 데이터를 주 메모리에 선반입하여 실질적인 하드디스크와의 입출력 횟수를 감소시키고 성능을 증대 시킨다[6]. 다시 말해서, 기존에 쓰였던 데이터를 주 메모리에 저장하고 있어 디스크의 직접적인 접근 빈도를 감소시키므로 디스크 지연성에서 자유로울 수 있고, 소규모의 쓰기 요청들을 하나로 통합하여 처리함으로써 오버헤드 감소 및 동일 페이지 내 반복적인 데이터 갱신을 줄일 수 있다. 추가적으로 자주 갱신되는 데이터 블록들을 주 메모리인 페이지

캐시에 먼저 적재하고 이를 나중에 디스크에 쓰기 때문에 페이지 캐시 내의 데이터가 디스크의 실제 영역 내의 데이터의 내용과 항상 일치하지 않는 특징을 포함한다. 하지만 페이지 캐시는 일반적으로 주 메모리를 사용하므로 적재할 수 있는 데이터의 양이 한계가 있다. 따라서 데이터 블록 참조 횟수, 시간 등과 같은 선형적인 (a priori) 정보를 이용하여 참조되지 않는 데이터 블록을 하드디스크로 갱신 및 퇴거시켜야 캐시 성공률을 향상 시키고 캐시 미스로 인한 페널티를 최소화할 수 있다[8]. 널리 이용되는 LRU 기법은 시간적 지역성(Temporal Locality)을 이용하여 캐시에 있는 블록 중 가장 오래 전에 참조된 블록을 교체한다. 다시 말해서 LRU 기법은 과거 정보인 최근 참조성 정보를 이용하여 페이지 캐시에 적재된 블록들의 퇴거 여부를 결정하게 된다.

2.1.2 2차 비휘발성 캐시

본 논문에서 제안하는 비휘발성 캐시는 주 메모리의 페이지 캐시와 함께 2차 캐시의 역할을 한다. 따라서 동일한 데이터를 페이지 캐시와 비휘발성 캐시에 담은 것은 동일한 데이터를 중복적으로 갖는다는 것이다. LRU 기법을 비휘발성 캐시와 페이지 캐시에 동시에 적용할 경우, 해당 데이터는 하드디스크로부터 비휘발성 캐시와 페이지 캐시에 두 번 저장되는 이중 캐싱 현상이 발생하게 되는데 이는 데이터 블록을 비휘발성 캐시와 페이지 캐시에서 중복적으로 퇴거시키고 빈 영역을 동일한 데이터를 채우므로 효율적인 블록 교체 기법이 되지 못한다. 이로 인해 하이브리드 하드디스크의 전력 소비를 최소화시킬 수 없으며, 비휘발성 읽기 캐시의 입출력 성능에 좋은 영향을 끼치지 못함이 알려졌다[2,5,7].

또한 [5]에서 전력 소비 면에서는 외장 비휘발성 캐시가 최대 14% 이상 효율적임을 증명하였다. 하지만 낸드 플래시 메모리는 특성상 표 1과 같이 읽기 성능에 비해 쓰기/지우기 동작 속도가 상대적으로 느리기 때문에 PostMark와 같은 쓰기 집중적인 벤치마크에서 수행 시간이 기존 하드디스크보다 최대 14배 이상 지연됨을 증명하였다(물론 USB, CF 인터페이스를 사용하므로 조금 더 느린 경향이 있다). 전력소비를 최소화하면서 비휘발성 캐시의 수명 감소를 해결하는 방법으로 쓰기 캐시 내의 데이터를 선별하여 전송량을 감소시키는 방법이

표 1 낸드 플래시 메모리의 동작 시간

동작	동작 시간
읽기(페이지)	22us
쓰기(페이지)	80us
지우기(블록)	2ms

있다. 이는 쓰기 캐시의 데이터 역시 차후 읽기 빈도가 높아 읽기 캐시로 전환될 가능성이 있으며, 또는 갱신 횟수가 낮아 자기디스크에 갱신할 필요가 없는 데이터가 존재하기 때문이다.

2.2 낸드 플래시 메모리의 블록 교체 기법

2.2.1 CFLRU(Clean-First LRU)

플래시 메모리 기반의 페이지 캐시에서의 블록 교체 기법 정책 중 CFLRU(Clean First LRU)가 있다[9]. 운영체제에서는 참조하던 특정 페이지 캐시가 사용자 또는 운영체제에 의해 갱신이 된 경우 이를 더티 페이지(Dirty Page)라 불리고 갱신이 되지 않은 경우는 클린 페이지(Clean Page)라 한다. 그림 2와 같이 CFLRU는 LRU 정책을 기본적으로 채택하고 있고 플래시 메모리의 쓰기 횟수를 감소시키기 위해 페이지 캐시 페이지 중 갱신이 되지 않은 클린 페이지를 우선적으로 Victim으로 선택함으로써, 플래시 메모리 수명에 영향을 줄 수 있는 쓰기 요청을 감소시킨다. CFLRU와 유사한 페이지 캐시 교체 정책으로 CFLRU/C, CFLRU/E, DL-CFLRU/E 등이 있다[10].

2.2.2 FAB(Flash Aware Buffer)

FAB는 완전 연관 섹터 변환 방식인 FAST(Fully Associative Sector Translation) 기반 FTL(Flash Translation Layer)에서 블록들을 그룹화하고 이러한 그룹들은 LRU 기법을 이용하여 리스트를 유지된다[11]. 특정 블록이 참조되면 해당 블록이 속해 있는 그룹이 LRU 리스트의 맨 앞으로 이동하거나 캐시에 존재하지 않는 경우에는 해당 블록은 LRU 리스트의 가장 앞의 그룹에 포함된다. 그리고 캐시의 블록 교체를 수행하는 경우 블록을 가장 많이 가지고 있는 그룹을 선정하여 퇴거시킨다. 이는 대용량 멀티미디어 파일과 같은 반복적인 읽기보다 순차적인 읽기가 많은 PMP 시스템에서 매우 효율적이다. 연관 블록들이 하나의 그룹으로 묶여서 캐시에

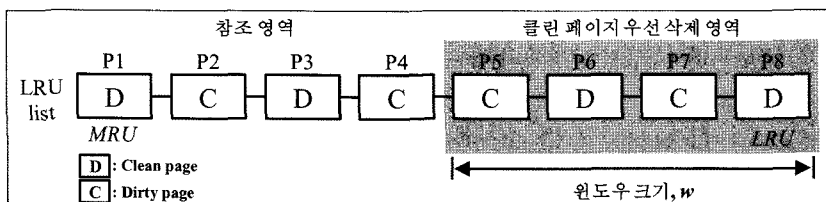


그림 2 CFLRU 알고리즘

서 동시에 삭제되기 때문이다.

2.2.3 BPLRU(Block Padding LRU)

BPLRU는 SSD 기반의 FTL에서 사용하는 기법으로 SSD 내부에 쓰기버퍼를 하나 두어 로그블록 기반의 FTL에서 머지 횟수를 감소시키고자 하는 기법이다[12]. 그림 3과 같이 LRU를 섹터 단위가 아닌 블록단위로 하여 블록단위의 지역성을 가지며, LRU에 의해서 캐시의 데이터가 교체될 때도 섹터를 포함하는 블록 전체를 퇴거 시킨다. 따라서 이는 SSD 내부의 쓰기 횟수를 감소시키고 로그 기반의 FTL에서 수행하는 머지 기법의 횟수를 감소시킨다. 하지만 BPLRU는 FTL이 로그블록 기법을 사용하고 SSD 내부적으로 동작하는 모든 방식을 이해해야만 구현할 수 있다는 단점이 있다.

2.2.4 EXCES(EXternal Caching in Energy Saving Storage System)

EXCES는 외장형 비휘발성 캐시를 이용한 저전력 소비 저장 시스템으로, 모바일 컴퓨팅 환경에서 하드디스크로 인한 전력 소비를 최소화하고자 제안된 기법이다 [5]. 그림 4와 같이 EXCES는 페이지 접근 추적기(Page Access Tracker) 컴포넌트를 통해 운영체제의 페이지 캐시의 접근 빈도를 시간 단위 별로 나누고 각 단위 별 접근 빈도와 접근 후 시간에 따른 가중치를 두어 Top-K 방식으로 해당 페이지의 순위를 연산하여 최근 성과 참조성을 모두 고려한다. 또한 경로우회자(Indirector)를 통해 데이터의 위치를 결정하고, 재설정 컴포넌트들(Reconfiguration Trigger, Reconfig Planner, Reconfigurator)은 해당 데이터의 위치가 현재 위치와 적절한지를 비교하여 에너지 소비를 최소화한다.

3. 데이터 블록 교체 및 재배치 기법

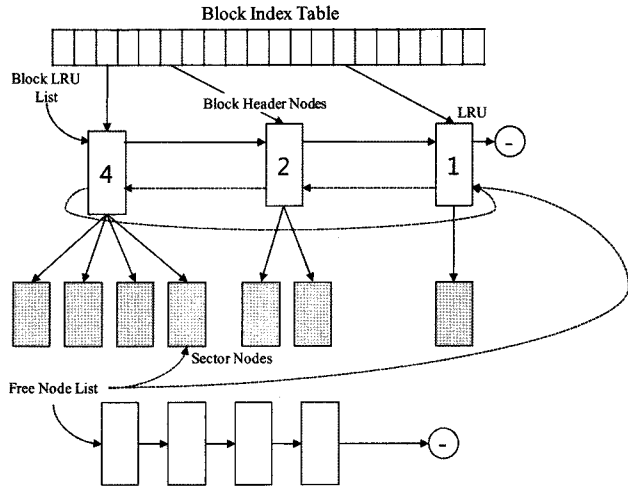
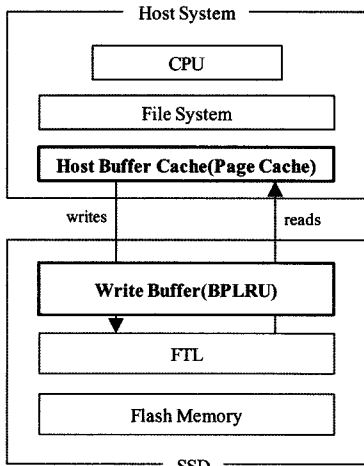


그림 3 BPLRU 구조

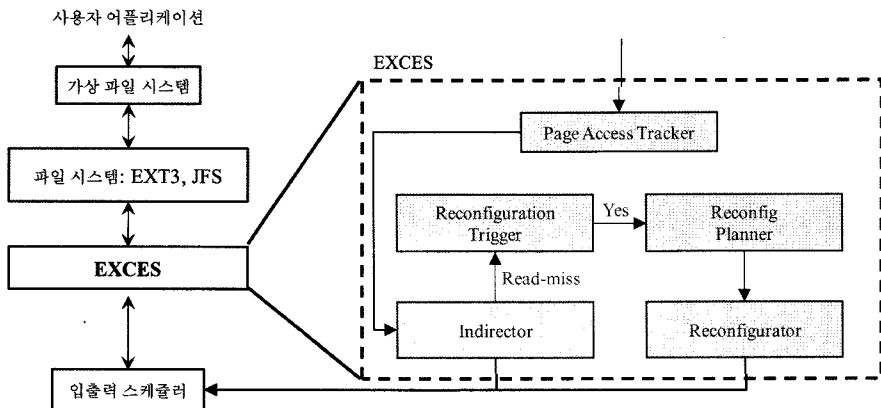


그림 4 EXCES 시스템 구조

낸드 플래시 메모리로 구성된 비휘발성 캐시는 자기 디스크와는 달리 블록이 저장되는 각 셀 별로 수명이 한정되어 있다. 또한 쓰기와 지우기 동작은 읽기 동작에 비해 많은 시간을 수반한다. 반복적인 데이터의 갱신은 비휘발성 캐시의 수명을 급격하게 감소시키고 쓰기가 집중되는 경우 기존 하드디스크보다 좋지 못한 성능을 보여준다. 따라서 낸드 플래시 메모리와 자기디스크를 같이 사용하는 하이브리드 하드디스크는 전력소비를 어느 정도 감소시킬 수 있으나 입출력 성능저하가 발생할 수 있는 상반관계에 있다. 이러한 상반관계를 극복하기 위하여 본 논문에서는 운영체제의 버퍼 캐시로 인해 이 중 캐싱이 발생하지 않도록 읽기와 쓰기의 참조 빈도를 이용하여 해당 저장장치 영역에 적합한 데이터 블록을 선정하는 LFU-Hot 블록 교체 기법과 멀티존 기반의 데이터 블록의 재배치 기법을 설명한다. 설명하기에 앞서 비휘발성 캐시의 논리적 구조는 그림 5와 같다. 비휘발성 캐시의 50%는 쓰기 캐시, 50%는 읽기 캐시로 이용하며 비휘발성 캐시의 맨 앞 부분은 전체 비휘발성 캐시의 정보를 담고 있는 메타 데이터로 사용한다. 각 캐시는 메타 블록과 데이터 블록으로 이루어져 있다. 메타 블록은 섹터번호, 논리 주소, 길이, 상태 정보로 나누어져 있다. 이에 따른 정보들은 다시 메인 메모리에서 리스트로 관리된다.

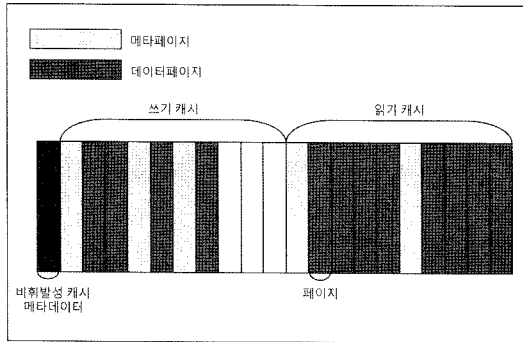


그림 5 비휘발성 캐시 구조

3.1 LFU-Hot 블록 교체 기법

LFU-Hot 블록 교체 기법은 자주 갱신되는 데이터 블록을 자기디스크로 이동시켜 비휘발성 캐시의 사용 수명을 연장 시키고, 비휘발성 캐시의 읽기 영역에 참조성이 높은 데이터 블록들만을 유지시키는 기법으로 하이브리드 하드디스크의 성능 및 비휘발성 캐시의 수명을 향상시킨다.

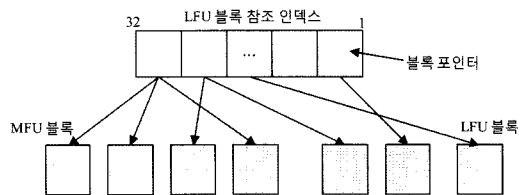
LFU-Hot 블록 교체 기법에서는 데이터 블록의 참조 빈도 및 쓰기 빈도를 측정하기 위해 LFU 비트와 Hot 비트 등을 사용한다. LFU 비트는 비휘발성 캐시의 입출

력 성능 향상을 위해 사용되며 Hot 비트는 빈번한 쓰기 동작으로 인한 비휘발성 캐시의 수명 단축 및 성능 저하를 방지하기 위해 사용한다. LFU 비트는 운영체제 버퍼 캐시에서 중요시하는 최근성과 중복이 되지 않는 참조성을 이용하며, 이 중 비휘발성 캐시의 가장 큰 장점인 읽기 성능을 극대화시킨다. Hot 비트는 쓰기가 빈번한 데이터 블록을 추출하기 위해 쓰기 요청의 참조성을 사용한다. LFU 비트와 Hot 비트는 페이지 캐시의 데이터 영역이 저장장치로 분배되는 입출력 스케줄러에서 연산되고 초기화 된다. 그 후 LFU 비트와 Hot 비트는 비휘발성 캐시의 메타 데이터 영역에 저장되어 최근성에 대한 정보도 유지하도록 한다. 그림 6은 LFU 리스트와 Hot 리스트의 자료 구조를 나타내며, 이는 32개 참조 그룹을 나타내는 인덱스이다. 또한 각 엔트리는 동일 참조 횟수를 갖는 블록들에 대한 포인터를 포함하고 있다.

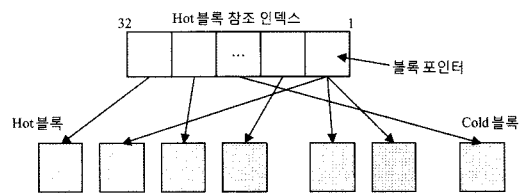
Hot 비트의 참조 빈도 값이 최상위로 높은 경우, 해당 데이터 블록은 자기디스크의 입출력 성능이 가장 높은 최외곽 영역인 패스트밴드로 이동하게 된다. 패스트 밴드는 ZBR(Zone Bit Recording) 방식의 하드디스크 중 가장 데이터의 밀집도가 높은 영역을 호칭한다[13-15].

그림 7은 LFU 비트와 Hot 비트 값에 따라 영역 별로 해당 데이터 블록의 위치가 결정되는 것을 보여준다. 하지만 이러한 데이터 블록 교체 기법이 무분별하게 이루어진다면 교체로 인한 비용만이 증가하여 성능에 나쁜 영향을 끼칠 수 있다. 따라서 본 논문에서는 해당 데이터 블록의 이동으로 인한 비용을 계산하여 하이브리드 하드디스크 입출력 성능의 효율성을 제고한다.

비휘발성 캐시와 자기디스크의 영역 간의 데이터 블록 재배치의 비용은 다음과 같으며 해당 심볼의 설명은 표 2와 같다.



(a) LFU 블록 리스트 구조



(b) Hot 블록 리스트 구조

그림 6 LFU, Hot 리스트 자료 구조

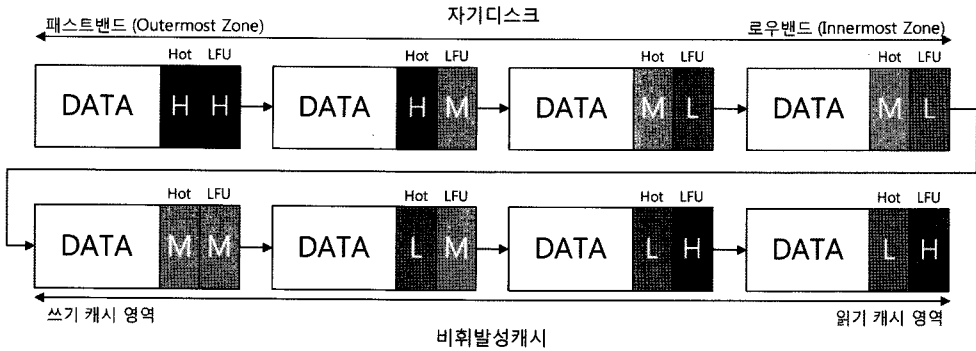


그림 7 LFU-Hot 블록 교체 기법을 적용한 데이터 배열 레이아웃

표 2 데이터 블록의 재배치 비용 산정을 위한 심볼

$C_{NV}$	비휘발성 캐시 쓰기 비용
$C_{MD}$	자기디스크 영역에 쓰기 비용
$T_r$	읽기 동작 시 걸리는 시간
$T_u$	갱신 시 걸리는 시간
$T_e$	비휘발성 캐시의 데이터 블록 삭제 비용
$T_h$	자기디스크의 탐색 지연 시간
$LFU_n$	블록 n의 정규화된 읽기 빈도
$Hot_n$	블록 n의 정규화된 쓰기 빈도
$Z_n$	자기디스크의 해당 영역 번호

$$C_{NV} = (T_r \times LFU_n) + (T_u \times Hot_n) + T_e \quad (1)$$

$$C_{MD} = (T_{r[zn]} \times LFU_n) + (T_{u[zn]} \times Hot_n) + T_n \quad (2)$$

예를 들어, 8번의 읽기와 8번의 갱신을 가진 데이터 블록이 있다면 데이터 블록이 비휘발성 캐시에 위치한 경우 표 1을 참고하여 블록이 소모하는 비용을 다음과 같이 계산할 수 있다.

$$C_{NV} = (22\mu s \times 8) + (80\mu s \times 8) + \frac{8}{32} \times 2ms = 1.17$$

또한 비휘발성 캐시는 각 셀 별로 쓰기 횟수의 한계가 있어 빈번한 쓰기를 제어해야 한다. 따라서 본 논문에서는 각 블록 교체 기법이 어느 정도 비휘발성 캐시의 수명에 영향을 주는지 측정하기 위하여 다음과 같은 공식을 사용한다. 이와 같은 공식은 추후 블록 교체 기법으로 인해 발생하는 입출력량을 이용하여 해당 NVCache의 수명을 측정하는데 매우 유용하다.

$$Y_{NV} = \frac{SZ_w}{T_{current} - T_{start}} \quad (3)$$

$Y_{NV}$ 는 비휘발성 캐시의 수명을 년으로 표시한 것이며,  $T_{current}$ 는 현재 시간이고  $T_{start}$ 는 쓰기 동작을 시작한 시간이다.  $SZ_w$ 는 해당 시간 동안 비휘발성 캐시에 쓰여진 데이터의 전체 크기이다. 따라서 쓰기 동작을 수행하는 시간 동안 쓰여진 데이터의 총량을 이용하여 비휘발성 캐시의 수명을 측정한다.

### 3.2 멀티존 기반의 데이터 블록 재배치 기법

하이브리드 하드디스크는 기본적으로 전력 소비를 최소화하기 위해서 자기디스크의 스피들 모터를 정지시켜 대기 상태로 진입시킬 수 있다. 자기디스크의 스피들 모터가 정지된 경우, 운영체제로부터 읽기, 쓰기 요청이 발생하면 데이터 블록은 비휘발성 캐시의 읽기, 쓰기 캐시에서 적절하게 쓰기, 읽기 동작을 수행한다.

예컨대, 쓰기 요청이 들어오고, 이러한 쓰기 요청이 비휘발성 쓰기캐시의 용량 이상 존재하는 경우, 하이브리드 하드디스크는 스피들 모터를 동작시켜 비휘발성 쓰기 캐시의 데이터를 자기디스크에 모두 갱신한다. 이렇게 일괄적으로 이동하는 것을 플러싱이라고 한다[7].

본 논문에서는 비휘발성 쓰기 캐시의 데이터 블록을 자기디스크에 효과적이면서 일괄적으로 쓰기 위해 앞서서도 언급했던 데이터 블록 교체 기법인 LFU-Hot 블록 교체 기법을 같이 적용한다.

본 알고리즘 1은 [15]의 기법에서 착안되었으며 본 알고리즘의 상대 중요도(RI: Relative Importance)를 구하기 위해 LFU 비트와 Hot 비트의 값을 정규화하여 계산한다. 본 알고리즘에 해당하는 심볼의 상세한 내용은 표 3에 나와 있다. 각 자기디스크의 존 별로 빈 공간을 지속적으로 관측하기 위해 각 자기디스크 내 존 별 사용량(ZUI: Zone Usage Indicator)을 측정한다.

존 별 사용량인 ZUI는 현재 해당 존의 사용량인  $Usage_{nz}$ 에 현재 쓰고자 하는 데이터의 크기인  $SZ_{data}$ 를 합한 것을 해당 존의 전체 용량인  $ZC_{nz}$ 으로 나눈 값이다.

$$ZUI = \frac{Usage_{nz} + SZ_{data}}{ZC_{nz}} \quad (4)$$

$C\_ZR$ 은 누적 존 비율로 해당 존마다 전체 용량에 따른 존 용량 누적 비율을 의미하며 이 누적비율이 증가하여 상대 중요도인 RI보다 크고 ZUI 값에 의해 데이

표 3 데이터 재배치를 위한 자기디스크 존 선택 알고리즘의 심볼

ZR	해당 존의 전체 존 대비 용량 비율
C_ZR	누적된 존 비율
RI	해당 데이터 블록의 상대적 중요도
NZ	자기 디스크의 존 개수
ZUI	해당 존의 전체 용량 대비 적재 데이터 비율
S_ZONE	해당 데이터를 쓰기로 결정된 존 (0: 최외곽존 NZ: 최내각존)
DATA_SZ	해당 데이터의 크기

터가 쓰여질 공간이 해당 존에 존재한다면 데이터를 해당 존에 쓰게 된다. 다시 말해서 제안된 알고리즘은 자기디스크의 각 존의 사용률과 데이터의 상대적인 중요도에 따라 쓰여질 존이 확정된다. 따라서 제안하는 알고리즘은 데이터 블록의 상대적인 중요성에 따라 디스크 입출력 성능을 효과적으로 증가시킬 수 있다.

```

MultiZoneFlushing(LFU, HOT, DATA, DATA_SZ)
{
    double C_ZR;    // Cumulative Zone Ratio

    // RI (Relative Importance) of data
    RI = Normalized_LFU + Normalized_Hot
    for(C_ZR = 0, i = 0; i < NZ; i++) {
        ZUI = (USAGE[NZ-i-1] + DATA_SZ) / ZC[NZ-i-1];
        C_ZR += ZR[NZ-i-1];    // Zone Ratio
        if(RI <= C_ZR && ZUI <= 1) {
            S_ZONE = NZ-i-1;    // Selected Zone
            break;
        }
    }
    Move_Rotation_Media(S_ZONE, DATA, DATA_SZ)
}
    
```

알고리즘 1 데이터 이동을 위한 자기디스크 존 선택 알고리즘

#### 4. 실험 결과

제안하는 LFU-Hot 블록 교체 기법이 하이브리드 하드디스크의 성능을 어느 정도 향상할 수 있는지를 측정하기 위해, 기존 하드디스크와 플래시 메모리 시뮬레이터를 병합하여 몇 가지 실험을 실행하였다. 본 실험의 환경은 리눅스 커널 2.6.24 버전에서 구축하였다. LFU 비트와 Hot 비트는 각각 32비트를 사용했으며, 실험을 위해 그림 8과 같이 블록 장치 영역에 리다이렉션 및 선반입이 가능하도록 설계 및 개발하고 이와 함께 자기 디스크와 SSD(Solid State Drive)를 같이 사용하여 하이브리드 하드디스크 시뮬레이터를 구현하였다. SSD는 여기서 NVCache 역할을 하며 각각 HDD와 SSD의 성능은 표 4와 같다. 또한 기존 방법들과의 성능 측정을 위

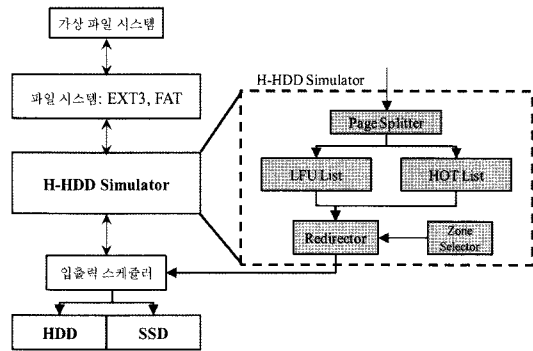


그림 8 하이브리드 하드디스크 시뮬레이터 구조

표 4 HDD와 SSD의 전송률 비교

	HDD(160GB)	SSD(8GB)
Maximum	48.7MB/sec	29.4MB/sec
Average	36.1MB/sec	27.6MB/sec
Burst rate	87.0Mb/sec	35.4MB/sec
Interface	S-ATA2	S-ATA2

해 IOzone 파일시스템 벤치마크와 Netapp사의 Postmark 벤치마크를 사용하였다[17,18]. LFU-Hot 기법의 성능을 객관적으로 평가하기 위해 기존 제안된 방법인 CFLRU, EXCES과 전통적인 LRU, LFU 기법을 비교하여 성능을 측정하였다. 관련 연구에서 제시한 FAB, BPLRU는 낸드 플래시 메모리의 성능을 향상시키기 위한 블록 교체 기법이지만 FTL 수준에서 해결을 하고자 한 연구로서, 운영체제 수준에서 해당 기술 비교에 비교적 많은 어려움이 따르므로 실험에서 이를 제외한다.

#### 4.1 데이터 블록 교체 및 재배치 기법의 성능 비교 및 분석

본 실험은 제안한 LFU-Hot 블록 교체 기법과 멀티존 기반의 데이터 재배치 기법이 입출력 성능에 어느 정도 영향을 주는지 측정하기 위한 것이다. 우선, 그림 9는 LFU-Hot 블록 교체 기법이 LFU 비트와 Hot 비트를 이용하여 벤치마크에 사용된 데이터 블록들의 참조도와 쓰기 빈도 비트에 따라 분포하는지 보여준다.

이러한 분포를 가진 LFU-Hot 블록 교체 기법과 기존의 블록 교체 기법을 하이브리드 하드디스크 시뮬레이터를 통해 벤치마크의 전체 수행시간을 서로 비교하였다. 일반 하드디스크의 실행시간을 100% 기준으로 삼고 측정된 결과 그림 10에서의 같이 LFU-Hot 블록 교체 기법이 기존 하드디스크 대비 최대 38% 이상의 성능 향상을 보였으며, LFU 기법보다 최대 23%, CFLRU 보다 최대 27%, EXCES보다 최대 19% 이상의 성능 향상을 가져옴을 알 수 있다.

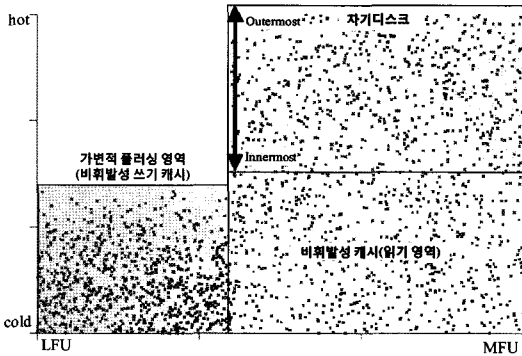


그림 9 LFU-Hot 블록 교체 및 배치 기법을 이용한 할당 영역

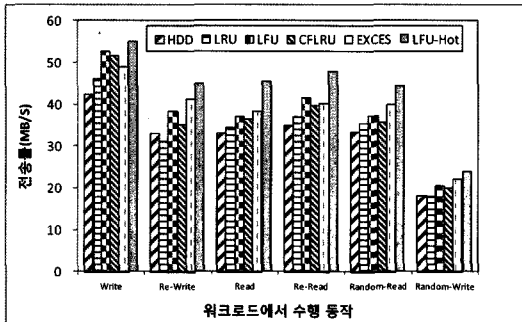


그림 10 IOzone 벤치마크 수행 결과

4.2 비휘발성 캐시의 수명 연장

LFU-Hot 블록 교체 기법은 하이브리드 하드디스크의 성능을 향상시킬 뿐만 아니라 잦은 갱신을 발생시키는 데이터 블록을 자기디스크로 이동시키므로 비휘발성 캐시의 수명을 어느 정도 향상시키는지 확인하기 위한 실험을 하였다. 본 실험에서 수명 측정을 위해 쓰기 집중적인 Postmark 벤치마크를 사용했으며 각 블록 교체 기법들이 하이브리드 하드디스크의 비휘발성 캐시 수명에 영향을 주는 것을 측정하기 위해 사용했다. 쓰여진 블록 교체 기법은 LFU-Hot, CFLRU, EXCES, LFU이다. 이러한 수명을 측정하기 위해 식 (3)을 사용하여 수명을 측정하였다. 그림 11에서처럼 256MB의 비휘발성 캐시를 쓰는 경우 LFU는 수명이 약 3.4년, EXCES는 약 4.5년 CFLRU가 약 5.3년으로 측정되었으며, 본 논문에서 제시한 LFU-Hot 블록 교체 기법이 이 7.1년으로 비교적 수명이 긴 것을 알 수 있었다. 비휘발성 캐시가 최대인 8GB인 경우 LFU-Hot 블록 교체 기법의 경우 최대 약 21년으로 나타났으며 LFU, EXCES, CFLRU는 각각 10.5년, 14.0년, 16.6년으로 나타났다. 결과적으로 LFU-Hot블록 교체 기법이 비휘발성 캐시의 수명연장에 도움을 주는 것으로 보인다.

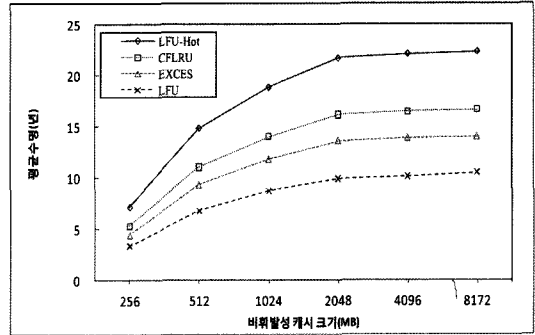


그림 11 블록 교체 기법에 따른 비휘발성 캐시의 예상 수명

4.3 평균 스핀 다운 주기

본 실험은 비휘발성 캐시로 인해 자기디스크의 스핀들 모터를 정지시켰을 경우(스핀다운 상태), 평균 입출력 횟수를 측정한 것이다. 이는 자기디스크가 동작하지 않는 기간이 비례적으로 증가하는 결과이므로, 비휘발성 캐시로 인해 전력 소비가 어느 정도 줄 수 있는지를 알 수 있는 척도가 된다. 블록 교체 기법에 따라 자기디스크의 스핀들 모터를 동작 주기를 결정할 수 있다. 또한 스핀다운 상태에서 평균 입출력 횟수가 많다는 의미는 스핀들 모터를 동작하지 않고 비휘발성 캐시에서 대부분의 데이터를 처리하는 것을 의미한다. 그림 12와 같이 LFU-Hot 블록 교체 기법은 LRU와 FIFO 방식보다 스핀다운 상태에서 처리하는 입출력 양이 LRU보다 약 4배 정도 EXCES보다는 약 3배 정도 많은 것을 알 수 있다. LRU와 LFU-Hot의 스핀 정지 시에 평균 입출력 횟수 많은 이유는 페이지 캐시와 함께 비휘발성 캐시의 퇴거 패턴이 동일하게 중복적으로 일어나기 때문이다.

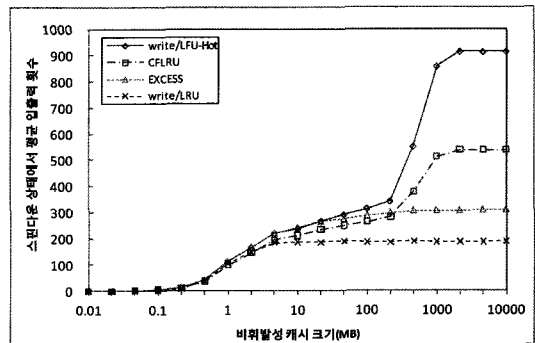


그림 12 자기디스크의 스핀 정지 시에 평균 입출력 횟수

5. 결론

본 논문에서는 비휘발성 캐시를 이용하는 하이브리드 저장장치 중의 하나인 하이브리드 하드디스크에 적합한



LFU-Hot 블록 교체 기법을 보였다. LFU-Hot 블록 교체 기법은 비휘발성 캐시와 자기디스크를 사용하는 하드디스크에서 비휘발성 캐시의 전력소비를 낮추면서 입출력 성능 또한 향상시키는 장점을 갖고 있다. 본 연구에서 차후 계속 해야 할 연구로는 첫째, LFU 비트와 Hot 비트의 가중치를 적응적으로 조절하여 ZBR로 구성된 자기디스크의 입출력 성능을 극대화하는 방법을 연구하는 것이다. 둘째, 본 논문에서는 비휘발성 읽기, 쓰기 캐시의 크기를 정적으로 설정하였다, 하지만 해당 데이터의 요청 패턴 또는 응용 프로그램의 패턴에 따라 동적으로 읽기, 쓰기 캐시의 크기를 조절해야 할 필요성이 있다. 따라서 동적으로 데이터 패턴에 따라 캐시의 크기를 동적으로 할당하는 알고리즘을 연구하여 입출력 성능을 향상시키는 방법을 고찰하는 것이다. 이러한 연구를 통하여 낸드 플래시 메모리의 낮은 전력소비와 빠른 읽기 성능, 그리고 자기디스크의 높은 전송률을 이용한다면 추후 하이브리드 저장장치의 입출력 성능을 극대화시킬 수 있을 것이다.

### 참고 문헌

- [1] Windsor W. Hsu, Alan Jay Smith, and Honesty C. Young, "The automatic improvement of locality in storage systems," *ACM Transactions of Computer Systems*, no.23(4), pp.424-473, 2005.
- [2] Timothy Bisson, Scott A. Brandt, and Darrell D.E. Long, "NVCache: Increasing the effectiveness of disk spin-down algorithms with caching," In *Proc. of the 14th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, pp.422-432, Sep. 2006.
- [3] Kwanghee Park, Jun-Sik Yang, Joon-Hyuk Chang, and Deok-Hwan Kim, "Anticipatory I/O Management for Clustered Flash Translation Layer in NAND Flash Memory," *ETRI Journal*, vol.30, no.6, pp.790-798, Dec. 2008.
- [4] Jun-Sik Yang, Young-Wook Ko, Chan-Gun Lee, and Deok-Hwan Kim, "Design and Implementation of Hybrid Hard Disk I/O System based on n-Block Prefetching for Low Power Consumption and High I/O Performance," *Journal of the KIISE : Computer Systems and Theory*, vol.36, no.6, pp. 451-462, Dec. 2009. (in Korean)
- [5] Luis Useche, Jorge Guerra, Medha Bhadkamar, Mauricio Alarcon and Raju Rangaswami, "EXCES: EXternal Caching in Energy Saving Storage Systems," In *Proc. of the IEEE International Symposium on High-Performance Computer Architecture*, pp.89-100, Feb. 2008.
- [6] Avi Silberschatz, Peter Baer Galvin, and Greg Gagne, "Operating System Concepts," John Wiley & Sons, Inc, 2008.
- [7] Timothy Bisson, Scott A. Brandt, "Flushing policies for NVCache enabled hard disks," In *Proc. of the 24th IEEE Conference on Mass Storage Systems and Technologies*, pp.299-304, Sep. 2007.
- [8] Donghee Lee, Sam H. Noh, Sang Lyul Min, Yookun Cho, "LRFU: A Block Replacement Policy which exploits Infinite History of References," *Journal of the KIISE(A)*, vol.24, no.7, pp.632-641, July 1997. (in Korean)
- [9] Seon-yeong Park, Dawoon Jung, Jeong-uk Kang, Jin-soo Kim, and Joonwon Lee, "CFLRU: A Replacement Algorithm for Flash Memory," In *Proc. of the 2006 International Conference on Compilers, Architecture and Synthesis for Embedded systems (CASE)*, pp.234-241, Oct. 2006.
- [10] Yun-Seok Yoo, Hyejeong Lee, Yeonseung Ryu, and Hyokyung Bahn, "Page Replacement Algorithms for NAND Flash Memory Storages," In *Proc. of the 2007 International Conference Computational Science and Its Applications (ICCSA)*, pp.201-212, Oct. 2007.
- [11] Heeseung Jo, Jeong-Uk Kang, Seon-Yeong Park, Jin-Soo Kim, Joonwon Lee, "FAB: flash-aware buffer management policy for portable media players," *IEEE Transactions on Consumer Electronics*, vol.52, Issue 2, pp.485-493.
- [12] Hyojun Kim and Seongjun Ahn, "BPLRU: A Buffer Management Scheme for Improving Random Writes in Flash Storage," *The 6th USENIX File And Storage Technologies (FAST'08)*, San Jose, California, USA. Feb. 26-29, 2008.
- [13] Fran Wang, Y. Deng, N. Helian, S. Wu, V. Khare, C. Liao, and A. Parker, "Evolutionary storage: speeding up a magnetic disk by clustered frequent data," *IEEE Transactions on Magnetics*, vol.43, Issue 6, pp.2295-2297, June 2007.
- [14] Shahram Ghandeharizadeh, Douglas J. Terardi, Dongho Kim, and Roger Zimmermann, "Placement of data in multi-zone disk drives," In *Proc. of the 2nd International Baltic Workshop on Database and Information Systems*, June 1996.
- [15] Byung-gu Yu and Seon-ho Kim, "An efficient zoning technique for multi-dimensional access methods," In *Proc. of the VLDB Workshop on Trends in Enterprise Application Architecture*, Aug. 2005.
- [16] Ruston Panabaker, "Hybrid hard disk and Ready-Drive technology: Improving performance and power for Windows Vista mobile PCs," Microsoft Windows Hardware Engineering Conference, 2006.
- [17] IOZone Filesystem Benchmark, <http://www.iozone.org>
- [18] Network Appliance, "Postmark: a new file system benchmark," Technical Report TR-3022, October 1997.



#### 박 광 회

2005년 용인대학교 컴퓨터정보처리학과 학사. 2004년 인프니스 기술연구소 연구원. 2004년~2005년 인테그라 정보통신 기업부설연구소 연구원. 2008년 인하대학교 전자공학과 석사. 2008년~현재 인하대학교 전자공학과 박사과정. 관심분야는 임베디드 시스템 소프트웨어, 하이브리드 저장장치, 차량용 소프트웨어 등



#### 이 근 형

2008년 인하대학교 컴퓨터공학과 학사  
2008년~현재 인하대학교 전자공학과 석사과정. 관심분야는 임베디드 시스템 소프트웨어, 대용량 저장장치, 감시 영상 기기 등



#### 김 덕 환

1987년 서울대학교 계산통계학과 학사  
1995년 한국과학기술원 정보통신공학과 석사. 2003년 한국과학기술원 정보통신공학과 박사. 1987년~1997년 LG전자 통신기기연구소 선임연구원. 1997년~2006년 동양공업전문대학 전산경영기술공학부 부교수. 2004년 University of Arizona, Tucson 박사 후연구원. 2006년~현재 인하대학교 전자공학과 부교수. 관심분야는 임베디드 시스템, 시스템 소프트웨어, 멀티미디어, 패턴인식, 데이터마이닝 등