

# IPTV를 위한 플래시메모리에서의 내용기반 지움 정책 설계 및 구현

조원희, 양준식, 고영욱, 송재석, 김덕환\*  
인하대학교 전자공학과

e-mail:{chowon, juneseek, kyw, gentlejs82 }@iesl.inha.ac.kr,  
deokhwan@inha.ac.kr

## Design and Implementation of Content-Based Clean Policy in Flash memory for IPTV

Won-Hee Cho, Jun-Sik Yang, Young-wook Go, Jae-Seok Song Deok-Hwan  
Kim\*

Dept. of Electronic Engineering, Inha University

### 요 약

IPTV(Internet Protocol Television)는 차별화된 초고속 광대역 네트워크를 기반으로 기존 TV의 단점을 보완하여 차세대 DTV 시장을 주도할 것으로 예상된다. IPTV의 저장용량이 증가하는 추세에 따라 SSD(Solid State Disk)가 NAND 플래시 메모리를 대체 할 것으로 예상된다. 본 논문에서는 IPTV의 저장장치인 SSD의 수명을 증가시키고 플래시메모리의 특성인 마모도 제한을 고려하지 않은 지움 정책(Garbage-Collection)을 사용하는 YAFFS(Yet Another Flash FileSystem)의 문제점을 해결하기 위해 블록을 내용기반 리스트로 관리하고 블록스왑을 사용하는 내용기반 지움 정책을 제안한다. 기존 파일시스템 보다 수명을 향상 시키는 내용기반 파일시스템을 설계 및 구현하여 성능을 분석하였다.

### 1. 서론

최근 DTV시장에서 IPTV(Internet Protocol Television)가 빠르게 보급되고 있으며, 이는 디지털영상 서비스, 양방향 데이터 서비스 및 다양한 개인 맞춤형 서비스를 제공한다. 따라서 이러한 IPTV의 발전으로 IPTV의 셋톱박스에 관한 하드웨어 및 소프트웨어 연구도 활발히 진행되어 지고 있다[1].

특히 IPTV의 서비스를 제공하는데 있어서 저장장치의 역할은 IPTV셋톱박스의 핵심적인 기술 중 하나이다. 현재까지 IPTV의 저장장치는 하드디스크만을 사용해 왔으나 최근에 하드디스크와 플래시메모리를 결합한 하이브리드 디스크(H-HDD) 및 SSD(Solid-Stats-Disk) 등이 새로운 저장장치로 연구 및 사용되고 있다[2].

SSD는 플래시 메모리와 컨트롤러로 이루어진 디스크로 기계적 장치가 존재하는 하드디스크에 비해 탐색시간(Seek Time) 및 회전대기시간(Rotational Time)에 의한 지연이 없으며, 하드디스크보다 소모전력이 작은 장점이 있다. 그에 비해 SSD를 구성하는 NAND 플래시 메모리의 특성상 읽기가 쓰기보다 빠르며, 덮어 쓰기가 없어 병합 연산이 필요하다. 따라서 SSD는 하드디스크와 달리

읽기가 쓰기보다 빠른 비대칭적인 성능을 보인다[3].

SSD에서는 최종적으로 데이터를 저장하는 집적화된 NAND 플래시 메모리들을 관리하는 플래시 메모리 컨트롤러가 존재한다. SSD의 플래시 메모리 컨트롤러는 파일 시스템레벨에서 NAND 플래시 칩과의 상호 작용을 지원한다.

본 논문에서는 기존에 NAND 플래시 메모리에서 많이 사용된 YAFFS 파일시스템의 지움 정책을 보완하고 삭제작업이 많은 IPTV의 특성을 고려하여 SSD상에 내용기반 지움 정책을 설계 및 구현하였다. 이를 통해 IPTV환경에 기존의 파일시스템 지움 정책 대신에 내용기반 파일시스템 지움 정책을 사용하므로 써 수명을 향상시킬 수 있다.

2장 관련연구에서는 SSD의 구조와 YAFFS의 지움 정책에 관하여 기술한다. 3장에서는 내용기반 블록관리 기법에 관하여 기술한다. 4장에서는 블록관리 기법에 기반을 둔 내용기반 지움 정책을 제시한다. 5장에서는 내용기반 지움 정책을 통해 기존의 지움 정책 보다 향상 된 수명을 확인한다. 6장에서는 결과 및 향후 연구방향을 기술한다.

### 2. 관련연구

#### 2.1 SSD (Solid Stats Disk) 구조

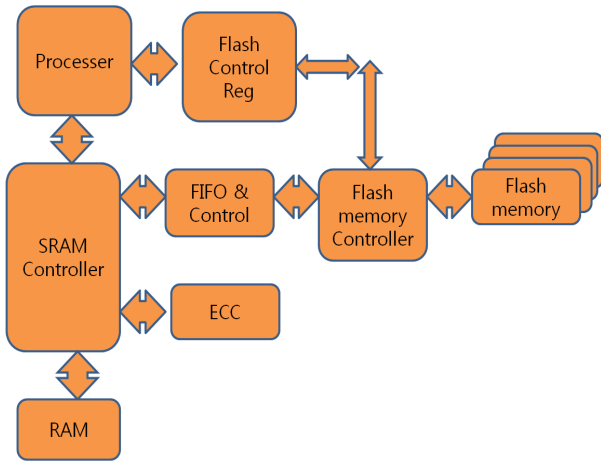
SSD는 다수의 NAND플래시메모리를 부착하고 호스트에서 요구하는 명령을 받아 수행하는 대용량 저장장치이다. 기존의 하드디스크와는 달리 플래시메모리에서 사용하는 블록기반 구조에 마모도에 따른 수명제한이라는 특성을

\* 정회원-교신저자, 인하대학교 전자공학과

※ 본 논문은 정보통신부 출연금으로 ETRI, SOC산업 진흥센터에서 수행한 IT SOC 핵심설계인력양성사업의 연구결과이며, 지식경제부와 한국산업기술 재단의 전략기술인력양성사업으로 수행된 연구결과임.

가진다.

그림 1은 SSD의 구조[3]를 블록 다이어그램으로 나타내었다.



(그림 1) SSD 구조 블록 다이어그램

SSD 내부 NAND 플래시 메모리에 저장된 데이터는 플래시 메모리 컨트롤러를 거쳐 FIFO & Control (First In, First Out)이 적용되어 SARM Controlle에 접근 한다. 이 데이터는 SRAM 컨트롤러에서 프로세서가 내린 명령에 따라 RAM에 접근이 결정된다.

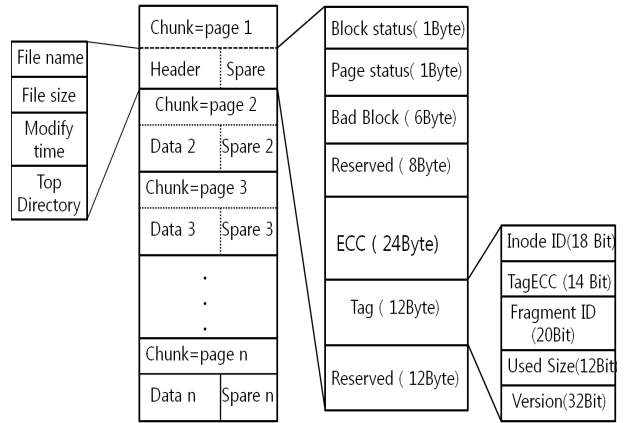
### 2.2. YAFFS(Yet Another Flash File System)

YAFFS는 페이지의 크기가 512Byte인 플래시 메모리만이 사용가능하며 최대 파일 크기 512MB, 최대 파일 수 260,000, 최대 파일 시스템 크기 1GB인 제약을 가진다[4]. 플래시 메모리에서 읽기와 쓰기는 페이지 단위로 수행되며, 삭제는 블록 단위로 수행된다. YAFFS는 그림 2와 같이 파일 데이터를 하나의 페이지와 동일한 크기인 Chunk로 나누어 플래시 메모리에 저장한다[8].

Chunk는 파일의 데이터와 스페어 영역으로 구분되며, 데이터 영역에 파일의 정보를 관리 하는 헤더(ObjectHeader)가 저장된다. 헤더가 저장되는 경우는 스페어 영역의 ChunkID가 0으로 되어 있다. 이 헤더는 파일의 이름과 파일의 크기, 수정 시간, 상위디렉터리에 포인터 등으로 구성된다. ChukID가 0이 아니면 파일의 데이터라고 간주한다[8]. 또한 각 Chunk마다 스페어 영역이 존재한다. 스페어 영역에는 블록상태, 페이지상태, ECC영역 및 Tag가 존재한다. 페이지 상태 영역에는 유효(valid)/무효(invalid)페이지를 나타내는 정보가 있다. Tag영역은 페이지에 들어있는 데이터의 크기, 파일 업데이트 횟수 및 아이노드 식별자 등의 자료가 들어 있다.

### 2.4. 파일시스템 지움 정책(Clean-Policy)

플래시 메모리는 한번 사용한 블록을 다시 사용하기 위해서 항상 삭제작업을 거쳐야 한다. 삭제작업은 읽기, 쓰

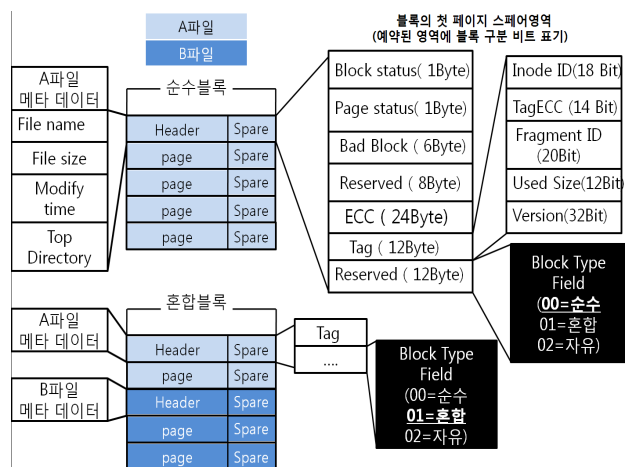


(그림 2) YAFFS 플래시 메모리 구조

기 속도에 비해 지우는 속도가 상당히 느리며, 삭제작업은 블록단위로 이루어진다. 또한 마모도 제한이 존재하여 삭제 시 마모도를 고려하지 않으면 플래시 메모리 수명이 줄어들 수 있다[7].

### 3. 내용기반 블록관리 기법

YAFFS와 같은 파일 시스템에서는 각 블록 안에 파일을 구분 할 수 있는 헤더가 저장되어 있기 때문에 한 블록 안에 포함된 파일 개수를 파악할 수 있다. 블록에서 파일 한 개당 하나의 헤더를 가지기 때문에 헤더의 개수를 통하여 블록 안에 들어있는 내용이 하나의 파일로만 이루어져 있는지 여러 파일이 혼합되어 들어있는지를 알 수 있다. 이를 기반으로 내용기반 YAFFS에서는 블록안의 파일 개수를 기반으로 블록을 세 가지(순수, 혼합, 자유)로 나눈다. 순수블록은 블록 안에 파일이 한 가지 내용의 파일로만 가득 차 이루어진 블록이며, 혼합블록은 두 가지 이상의 파일이나 자유페이지를 포함한 블록을 의미하며, 자유블록은 자유페이지만을 포함한 블록이다[6].



(그림 3) 순수블록과 혼합블록의 구조

그림 3은 순수블록과 혼합블록이 어떻게 구성되어 있는지

나타낸다. 순수 블록과 혼합블록의 첫 번째 페이지의 데이터 영역에는 헤더가 들어간다. 헤더에는 파일에 대한 정보(메타 데이터)가 저장되어 있다. 또한 내용기반 블록관리를 위하여 블록의 첫 페이지 스페어 영역의 예약된 영역(Reserved)에 블록종류필드(Block Type Field) 정보를 추가한다.

#### 4. 내용기반 지움 정책

##### 4.1. 파일 삭제와 파일 쓰기

파일 쓰거나 파일 삭제 연산이 발생하면 해당 파일이 들어있는 페이지들이 무효화된다. 따라서 순수블록의 경우에는 동일한 파일의 내용을 담고 있으므로 파일 삭제 시 모든 페이지가 무효화 된다. 반면에 혼합 블록의 경우 해당 파일이 들어있는 페이지들만 무효화된다. 또한 파일 기록 시에 마모도 평준화(wear-leveling)를 고려하여 기록한다[5].

##### 4.2 내용기반 가비지 컬렉션

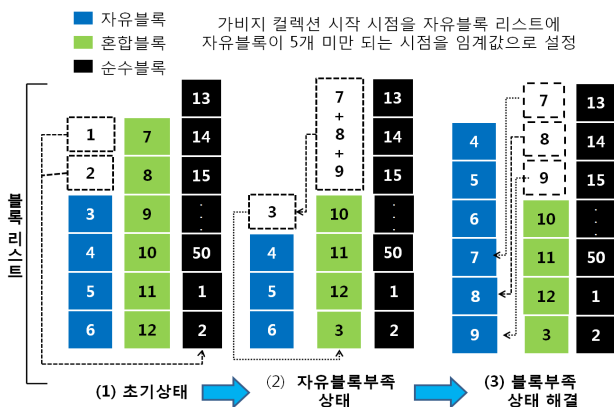
가비지 컬렉션은 자유블록(가용공간)이 부족할 때 수행되며 무효화 페이지들로 이루어진 블록에 삭제 연산을 수행하여 블록을 초기화함으로써, 다시 새로운 데이터를 저장할 수 있는 자유 블록을 만드는 과정이다. 가비지 컬렉션은 모든 페이지가 무효화된 블록에 대해서 삭제 연산을 수행하는 것이 원칙이다. 그러나 자유블록이 일정한 임계 값 이하로 떨어지고 더 이상 모든 페이지가 무효화된 블록이 존재하지 않는 경우에는 일부 유효 데이터를 포함하고 있는 블록을 대상 블록으로 선정하여 유효 데이터를 다른 블록으로 복사한 후 무효화 된 블록을 삭제하는 방식으로 수행 된다. 제안한 내용기반 가비지 컬렉션 방법은 2단계로 구성된다.

1단계: 가용공간이 부족한 경우 자유블록 확보.

2단계: 블록 스왑을 통한 마모도 평준화.

##### 4.2.1 자유블록 확보(가용 공간이 부족한 경우)

그림 4는 내용기반 가비지 컬렉션의 수행 순서를 나타낸다.



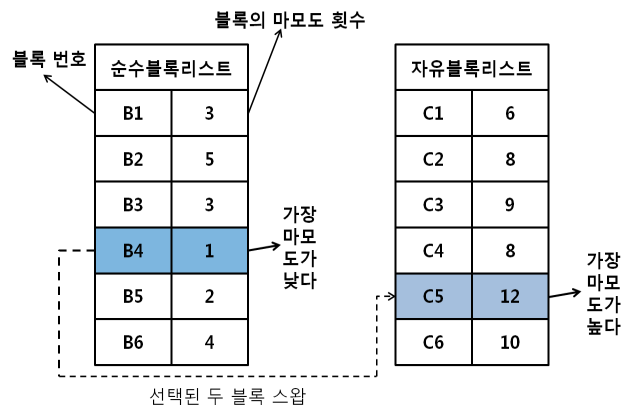
(그림 4) 자유 블록(가용공간) 확보 과정

데이터 기록 시 그 데이터가 차지하는 블록의 개수가 2개라고 가정하면 그림4의 (1)은 자유 블록 리스트의 블록 두개를 할당하며, 자유블록의 개수가 정해 놓은 임계값(5개 블록)에 미달하게 된 상태를 보여준다. 임계값은 남아있는 최소의 자유블록 개수를 의미하며 전체 블록개수의 10%를 임계값으로 설정하였다. 임계값에 미달하면 자유블록 부족 상태가 되며, 이런 경우에는 혼합 블록들에 흩어져 있는 여러 데이터들을 하나의 혼합 블록으로 모아 자유블록을 확보할 수 있는지 확인한다. (2)는 혼합 블록들 중 유효페이지 수가 가장 적은 블록을 선택하여 자유블록에 옮겨주는 상태를 나타낸다. 예를 들면 혼합블록(7,8,9)을 자유 블록 3에 모으면 2개의 자유블록이 확보되어 부족한 자유블록 문제를 해결할 수 있다. (3) 자유블록이 6개가 되어 블록부족 상태가 해결된 상황을 보여준다.

기존의 지움 정책들은 모든 자유블록이 사용되어 부족하게 되면 무효화 된 페이지가 많은 블록을 선택하여 삭제한다. 따라서 긴 삭제시간을 소비하는 단점을 가지고 있다. 하지만 내용기반 지움정책은 자유블록의 임계값을 정해놓고 임계값 미만의 자유블록이 남았을 경우에는 혼합블록을 병합하여 임계값 수치 위로 상향시켜 긴 삭제 대기시간을 줄이는 삭제 블록 선택 방법을 사용하여 기존의 삭제 블록 선택방법의 문제점을 해결한다.

##### 4.2.2 스왑을 통한 마모도 평준화 방법

내용기반 가비지 컬렉션의 두 번째 단계로 블록 스왑을 통한 마모도 평준화 과정이 진행된다. 그림 5는 마모도 평준화를 위한 블록 스왑과정을 나타낸다. 먼저 순수블록리스트 중 마모도가 가장 작은 블록을 선택 한다. 그 후 자유블록 리스트 중 마모도가 가장 높은 블록을 선택한 후 두 블록의 마모도를 비교하여 차이가 일정 범위(예:10)를 넘게 되면 두 블록을 스왑한다. 이 과정은 오버헤드이므로 유희시간에 수행되어야 한다[6].



(그림 5) 블록의 스왑과정

#### 5. 실험결과

본 논문의 실험 환경은 다음과 같다. CPU는 Intel Core2 Duo 2.34GHz, 2GB RAM을 사용하였고, IPTV에서 사용

되는 40GB의 삼성 SSD디스크(MCCOE64G5MPP-OVA)를 기반으로 운영체제 Linux 2.6.17환경에서 제안한 내용기반 가비지 컬렉션 시 마모도 관리를 통한 플래시 메모리의 수명 효율을 측정 하였다. 즉, SSD에서 제안하는 내용기반 YAFFS와 기존의 YAFFS의 성능을 비교하였다. 그림 6은 IPTV에서 사용된 VOD 멀티미디어 파일(약 700MB) 25개의 데이터를 나타내며, 이를 사용하여 수명 효율을 측정하였다.

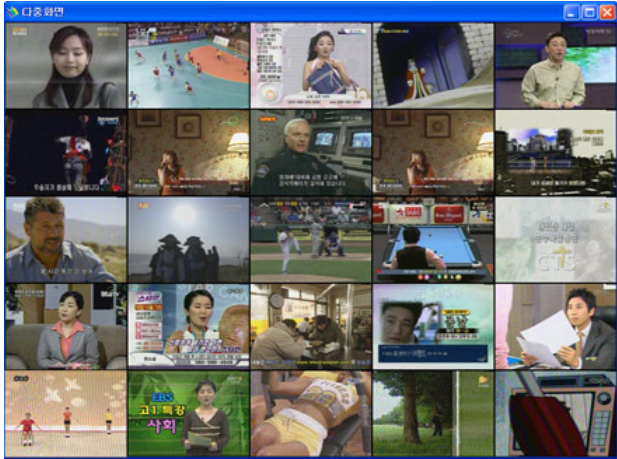


그림 6 VOD 멀티미디어 파일 실험 데이터

실험에는 다음과 같은 시뮬레이션을 사용하였다.

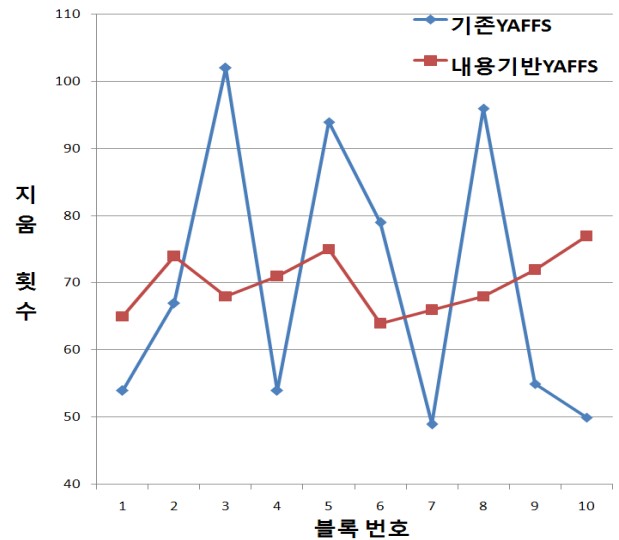
- 마모도 측정 시뮬레이션: 블록 삭제 명령어 횟수 측정, 다중 블록 삭제 명령어 횟수 측정, 지우기 요청된 블록 수 측정, 지우기 요청된 연속된 블록 수 측정 등의 4단계로 이루어진다[5].
- Postmark: 파일들에 대한 쓰기, 읽기, 변경, 삭제 등의 트랜잭션을 수행한다[9]. 파일의 개수는 100개이며, 수행 할 트랜잭션은 700으로 설정하였다.

그림 7은 기존의 YAFFS와 내용기반 지움 정책을 사용하는 내용기반 YAFFS의 마모도 평균화에 대한 실험 결과를 보여준다. 실험에서는 전체 40GByte SSD디스크를 한 블록을 80MByte로 설정하여 512개의 블록으로 나누었다. 그 중 10개의 블록을 무작위로 선택하여 지움 횟수를 비교하였다. 기존의 YAFFS는 블록들 마다 마모도 횟수가 큰 차이를 보이며 일부 사용되는 블록만이 계속해서 사용되지만 내용기반 YAFFS는 블록들의 마모도가 65내지 78회 내로 균일하게 유지되는 것을 알 수 있다.

## 6. 결론

본 논문에서는 기존의 YAFFS보다 효율적인 마모도 평균화를 제공하는 내용기반 YAFFS 파일시스템을 구현하였다. 제안된 내용기반 YAFFS는 블록을 내용기반으로 관리하여 스왑기법을 통하여 마모도를 효율적으로 관리하는 장점을 가지고 있다. 따라서 기존의 YAFFS보다 마모도

평균화 기법을 통하여 플래시메모리의 수명을 35% 증가시킨다.



(그림 7) 700회의 트랜잭션 반복 시 마모도 평균화 비교

## 참고문헌

- [1] 성정식, 이성용, 송호영, 김봉태, "IPTV 서비스 및 표준화 동향," [ETRI]전자통신동향분석, 2008.
- [2] 최진수, 정세윤, 차지훈, 문경애, 이태진, 강경옥, 이영일, "모바일 리치미디어 서비스 기술 특허 동향 분석," [ETRI]전자통신동향분석, 2008.
- [3] 문성엽, 김상우, 이상원, "Flash SSD상의 RAID 구성 의고찰," 한국인터넷정보학회 학술발표대회 논문집, 제9권, 제2호, pp.237-242, 2008.
- [4] 이태훈, 박송화, 정기동, "YAFFS를 위한 파일 연산 최적화 기법," 정보과학회논문지, 제34권, 제1호(B), pp.401-405, 2007.
- [5] 이동환, 조원희, 김덕환, "다중 블록 지우기 기능을 적용한 퓨전 플래시 메모리의 FTL 성능 측정 도구 설계 및 구현," 대한전자공학회 하계 종합 학술대회, 제31권, 제1호, pp.647-648, 2008.
- [6] 조원희, 이동환, 김덕환, "NAND플래시 파일 시스템을 위한 내용기반 블록관리기법을 이용한 마운트 시간 감소와 지움 정책," 대한전자공학회 논문지, 제46권, 제3호, 2009.
- [7] Kwanghee Park, Junsik 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, 2008.
- [8] Samsung Electronics Co., "NAND Flash Memory & SmartMedia," Data Book, 2002.
- [9] Postmark, <http://packages.debian.org/stable/utils/postmark>