

낸드 플래시 메모리 기반 멀티미디어 파일 시스템에서의 효율적인 페이지 할당 및 회수 기법¹

한준영^o 김성조

중앙대학교 컴퓨터공학부

hanjunyeong@konan.cse.cau.ac.kr^o sjkim@cau.ac.kr

An Efficient Page Allocation and Garbage Collection Scheme for a NAND Flash Memory-based Multimedia File Systems

Junyeong Han Sungjo Kim

School of Computer Science & Engineering Chung-ang University

요 약

낸드 플래시 메모리는 특성상 덮어 쓰기가 불가능하기 때문에 유효하지 않는 데이터가 저장된 더티(Dirty) 상태의 페이지를 삭제 연산을 통해 클린(Clean) 상태로 만든 후 데이터를 써야 한다. 더티 페이지가 낸드 플래시 메모리에 많이 존재하면 파일을 쓸 때 많은 블록을 삭제해야 하기 때문에 쓰기 지연 시간이 길어지는 문제가 발생한다. 따라서 본 논문에서는 일정한 쓰기 지연 시간을 보장하는 새로운 페이지 할당 및 회수 기법을 제안한다. 파일이 삭제될 때 더티 상태인 페이지를 삭제함으로써 클린 상태로 변경하여 낸드 플래시 메모리에 쓰기 지연 시간을 길게 만드는 더티 페이지가 없는 상태로 유지한다. 또한 삭제 연산은 블록 단위로 수행되므로 삭제할 블록의 유효한 페이지를 다른 블록으로 복사해야 하기 때문에, 페이지를 할당할 때 한 블록에 가급적 적은 개수의 파일을 저장하는 알고리즘을 제시한다.

1. 서론

최근 플래시 메모리는 전자 산업의 발전으로 가전, 통신기기, 휴대기기 등 다양한 기기의 저장매체로서 사용되고 있다. 플래시 메모리는 기존의 하드디스크 저장소와 다른 특성을 가지고 있기 때문에 플래시 메모리만을 위한 파일 시스템이 개발되었다. 특히 소형 멀티미디어 기기의 저장소로 사용되는 낸드 플래시 메모리를 위한 멀티미디어 파일 시스템이 많이 개발되었는데, 이 파일시스템은 크기가 큰 파일을 지원하고 멀티미디어 파일을 순차적으로 읽어서 재생하도록 특화되어있다. 하지만 기존의 멀티미디어 파일 시스템에서 유효하지 않은 데이터가 저장된 더티 페이지가 존재하기 때문에, 쓰기 연산에서 일정한 지연 시간을 보장하지 못한다. 멀티미디어 기기에서 소리를 녹음하거나 영상을 촬영할 때, 많은 양의 데이터를 일정한 시간 간격으로 계속 써야 하기 때문에 일정한 쓰기 지연 시간이 보장되지 않는다면 녹음 또는 촬영하는 데이터를 저장하지 못하고 잃어버릴 수 있다. 이런 문제를 해결하기 위해 본 논문에서는 일정한 쓰기 지연 시간을 보장하기 위한 블록 할당 및 회수 알고리즘을 제시한다.

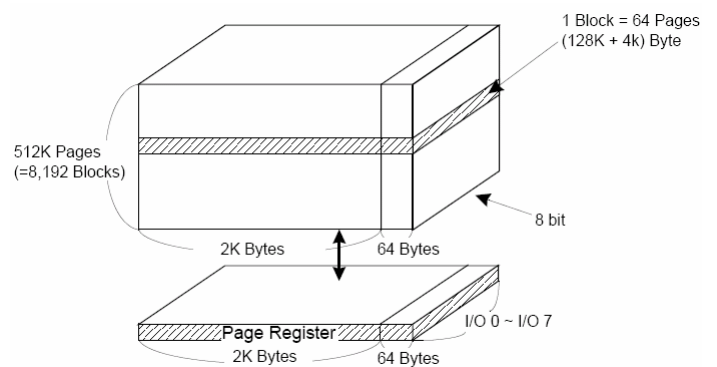
본 논문의 2장에서는 기존 낸드 플래시 메모리 파일

시스템의 블록 할당 및 해제 기법을 알아보고, 3장에서는 일정한 쓰기 지연 시간을 보장하는 블록 할당 및 해제 알고리즘을 설명한다. 4장에서는 제안한 알고리즘을 구현하여 성능을 측정하고, 5장에서는 본 논문의 결론을 맺고 향후 연구과제에 대해 기술한다.

2. 관련 연구

2.1 낸드 플래시 메모리 특성과 기본 연산

큰 블록 낸드 플래시 메모리는 (그림 1)[1]과 같이 2KByte 페이지 64개를 갖는 128KByte 블록으로 구성된다.



(그림 1) 낸드 플래시 메모리 구조

¹ 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT 연구센터 (홈네트워크연구센터) 지원사업의 연구결과로 수행되었음(IIITA-2006-C1090-0603-0035)

낸드 플래시 메모리의 페이지는 유효한 파일의 데이터가 저장된 유효 페이지와, 파일이 삭제되어 더

이상 저장된 페이지가 사용되지 않는 더티 페이지 그리고 블록 삭제 연산 후 페이지의 모든 비트가 1로 초기화되어 있는 클린 페이지의 3가지 경우로 존재한다.

낸드 플래시 메모리는 하드웨어 특성 상 다음과 같은 기본 연산의 제약이 따른다.

첫째, 낸드 플래시 메모리는 한 번 쓰기 연산을 수행한 공간에 덮어 쓰기 연산을 수행할 수 없다. 즉, 이미 사용한 공간에 쓰기 연산을 수행하기 위해서는 해당 공간을 클린 상태로 만들어 주는 삭제 연산이 반드시 선행되어야 한다.

둘째, 낸드 플래시 메모리는 읽기, 쓰기 연산의 단위와 삭제 연산의 단위가 다르다. 읽기와 쓰기 연산은 페이지 단위로 수행되지만, 삭제 연산은 여러 개의 페이지로 구성된 블록 단위로 수행되기 때문에 삭제 연산 시 유효한 페이지를 다른 블록으로 복사해야 하기 때문에 이에 대해 고려를 해야 한다.

셋째, 삭제 연산의 속도는 <표 1>[1]과 같이 한 블록을 삭제하는데 1.5ms가 소요되기 때문에 쓰기 연산을 수행할 때 클린 상태인 블록이 없어서 블록을 삭제한 후 쓰기 연산을 수행하면 지연시간이 길어지게 된다. 따라서 삭제 연산을 최소화하고 삭제 연산을 수행하는 시기의 적절한 선택에 대한 고려가 필요하다.

<표 1> 4G X 8 bit 낸드 플래시 메모리 연산 수행 시간

	읽기(2KB)	쓰기(2KB)	삭제(128KB)
수행 시간	25 μ s	200 μ s	1.5ms

2.2 기존 페이지 할당 및 회수 기법

YAFFS(Yet Another Flash File System)[2]는 NAND 플래시 메모리를 위해 처음으로 개발된 파일 시스템으로 단순하고 NAND에 최적화 되었으며, 읽기, 쓰기, 마운트, 메모리 소모량 등에서 우수하다. YAFFS의 디폴트 페이지 할당 기법은 순차 할당이며 해제는 쓰기 요청을 할 때마다 두 가지 모드로 구분되어 수행된다. normal 모드에서는 200개의 블록에서 더티 상태의 페이지가 가장 많은 블록을 하나 선택하여, 이 블록에서 유효한 페이지 개수가 3보다 적으면 블록을 삭제한다. Aggressive 모드에서는 더티 상태인 페이지가 존재하는 모든 블록을 삭제한다. 파일의 쓰기와 삭제가 많이 일어나게 되면 Aggressive 모드로 설정되어 모든 블록에 대해 삭제 연산이 일어나기 때문에 지연시간은 길게 수초까지 걸릴 수 있다. 따라서 YAFFS는 일정한 쓰기 지연시간을 보장할 수 없다.

MMFS(Mobile Multimedia File System)[3]는 각각의 파일을 블록단위로 저장하고, 파일이 삭제될 때 그 파일이 저장된 블록을 삭제함으로써 일정한 쓰기 지연시간을 보장하며, 삭제 연산 시 유효한 페이지를 다른 블록으로 복사하는 연산이 없는 장점이 있다. 하지만

모든 파일을 블록 단위로 저장하기 때문에, 낸드 플래시 메모리 저장 공간의 낭비가 심하며, 블록 개수 이상의 파일은 저장할 수 없다.

파일 시스템 전체적인 측면에서가 아닌 페이지 할당 및 해제 기법에 대한 연구도 많이 이루어졌다. 기존 연구[4]에서는 삭제 연산이 일어날 때 유효한 페이지를 복사하는 비용을 줄이기 위한 연구에 초점을 맞추고 있다. 블록 클리닝의 비용에 영향을 주는 성능 인자를 이용률, 무효율, 순수도로 수치화하여 블록 클리닝 비용이 가장 적도록 하는 순수도 기반 페이지 할당 기법을 사용하였다. 그리고 데이터를 자주 수정되는 데이터들과 그렇지 않은 데이터들을 구분하고 서로 다른 블록에 할당함으로써 클리닝 비용을 최소화하였다. 이 연구도 블록 삭제 연산의 횟수를 줄일 수 있지만 쓰기 요청 때 블록을 삭제하기 때문에 일정한 쓰기 지연시간을 보장하지 않는다.

3. 블록 할당 및 해제 알고리즘

일정한 쓰기 연산을 지원하기 위해서는 낸드 플래시 메모리에 할당된 페이지는 유효 상태이거나 클린 상태여야 한다. 이를 통해 쓰기 연산이 수행될 때 삭제 연산을 수행할 필요 없이 오직 쓰기 연산만 수행함으로써 일정한 쓰기 지연시간을 보장한다.

먼저 쓰기 요청을 버퍼에 모았다가 파일의 데이터를 세그먼트 단위로 낸드 플래시 메모리에 할당한다. 세그먼트는 현재 낸드 플래시 메모리 상태에서 한 블록에 저장할 수 있는 영역 보다 크지 않고, 한 블록 내에서 연속적인 페이지들로 이루어진 영역이다. 이렇게 세그먼트 단위로 데이터를 할당함으로써 읽기 및 쓰기 성능을 높일 수 있으며, 효율적인 블록 할당을 할 수 있다.

파일을 삭제할 때, 파일의 데이터가 할당된 블록을 지우면 그 블록에 저장된 유효한 페이지는 다른 블록으로 복사해야 한다. 따라서 한 블록에 가능한 적은 파일의 수의 페이지가 저장되어야 블록 삭제 시 유효한 페이지의 복사를 최소화 할 수 있다.

낸드 플래시 메모리에 저장되는 전체 파일의 집합인 F , 하나의 파일을 나타내는 F_i 와 파일을 구성하는 세그먼트 S 는 다음과 같이 표현된다.

$$F = \{F_1, F_2, \dots, F_n\} \text{ 일 때,}$$

$$F_i = \sum_{j=1}^m S_{ij} \quad \dots \text{ 식 (1)}$$

n : 낸드 플래시 메모리에 저장된 파일의 수
 m : 파일 i 를 구성하는 세그먼트의 수

하나의 파일 F_i 를 삭제할 때 필요한 연산의 비용

E_i 는 식 (2)와 같이 정의할 수 있다.

$$E_i = \sum_{j=1}^m (vs(B(S_{ij})) - s(S_{ij})) \cdot w + e \quad \dots \text{식 (2)}$$

$s(S_{ij})$: S_{ij} 가 할당된 페이지의 개수
 $B(S_{ij})$: S_{ij} 가 할당된 블록
 $vs(B(S_{ij}))$: $B(S_{ij})$ 의 전체 유효 페이지의 개수
 w : 페이지 한 개를 복사하는 비용(25 μ s + 200 μ s)
 e : 블록 한 개를 지우는 비용(1.5ms)

식 (2)에서 S_{ij} 가 할당된 블록의 전체 유효한 페이지의 개수 $vs(B(S_{ij}))$ 는 식 (3)과 같은 관계를 가지며, 식 (2)와 식 (3)으로부터 식 (4)를 유도할 수 있다.

$$s(S_{ij}) \leq vs(B(S_{ij})) \leq s(B) \quad \dots \text{식 (3)}$$

$$m \cdot e \leq E_i \leq \sum_{j=1}^m (s(B) - s(S_{ij})) \cdot w + e \quad \dots \text{식 (4)}$$

$s(B)$: 블록을 구성하는 페이지의 개수

식 (2)와 식 (3)에서 세그먼트를 할당할 때, $s(S)$ 가 클수록 파일의 삭제 연산 비용이 최소화 됨을 알 수 있다. 따라서 세그먼트 S 를 할당할 때, $vs(B(S)) - s(S)$ 의 값이 최소인 블록을 찾아서 할당한다. $vs(B(S)) - s(S)$ 의 값이 최소인 블록은 세그먼트 S 를 저장하고자 하는 블록이 비어있거나 할당된 유효한 페이지가 최소인 블록을 뜻한다. 즉, 모든 페이지가 클린한 블록부터 세그먼트가 할당되고, 모든 페이지가 클린한 블록이 없으면, 유효한 페이지가 가장 적은 블록에 세그먼트가 할당된다.

위와 같은 세그먼트 할당 알고리즘에 의해 저장된 파일이 삭제될 때, 그 파일의 모든 세그먼트가 할당된 페이지를 클린 상태로 만든다. 이를 위해 파일의 세그먼트가 저장된 블록마다 유효한 페이지를 다른 블록으로 복사하고 그 블록을 삭제하는 연산을 수행한다. 유효한 페이지를 복사할 때도 $vs(B(S)) - s(S)$ 가 최소가 되는 블록을 찾아서 복사를 수행한다.

식 (4)의 3번째 항은 최악(worst case)의 경우 파일 삭제 시 복사와 블록 삭제 연산에 필요한 비용이다.

4. 성능 측정

본 논문에서 제시한 알고리즘을 멀티미디어 파일 시스템인 NAMU(NAnd flash memory based Multimedia file system)에 적용하여 IOZone[5]을 이용해 성능을 측정하였다.

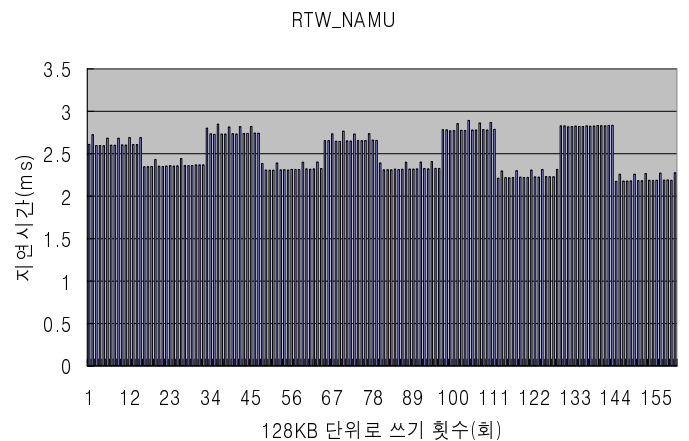
NAMU는 파일을 세그먼트 단위로 할당하여 파일

시스템에 저장하는데 세그먼트를 할당할 때, 모든 블록이 균일한 지움 횟수를 갖도록 지움 횟수가 적은 블록이 우선적으로 할당된다. 그리고 쓰기 연산을 수행할 때 세그먼트를 저장할 공간이 없으면 더티한 페이지가 저장된 블록을 삭제하여 저장 공간을 확보한 후 쓰기 연산을 수행한다. 그리고 파일 시스템에 일정 시간 읽기와 쓰기 연산이 없는 유휴 상태가 되면 가비지 컬렉션이 수행되어 더티 페이지를 클린 페이지로 변경한다.

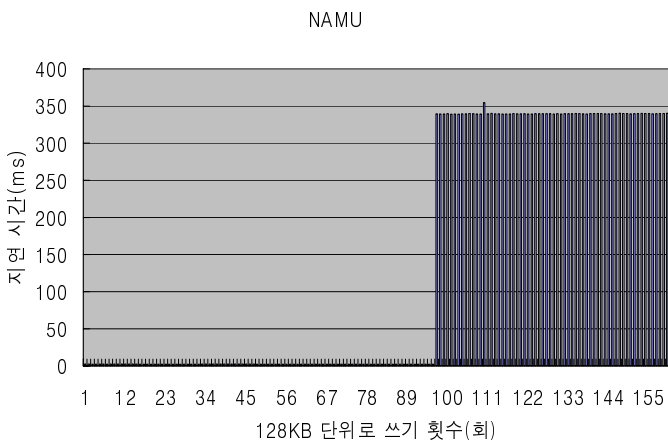
이런 알고리즘으로 수행되는 NAMU의 페이지 할당 및 회수 기법을 본 논문에서 제시한 기법으로 변경하여 일정한 쓰기 지연시간을 보장하는 RTW(Real Time Write) NAMU를 기존의 NAMU, YAFFS2와 비교하였다.

전체 저장 공간이 64MB인 낸드 플래시 메모리에 RTW NAMU와 NAMU, YAFFS2를 IOZone을 이용하여 20MB 파일을 각각 4KB, 32KB, 128KB 단위의 레코드 단위로 3회를 써서 더티 페이지가 많이 존재하게 한다. 레코드 단위를 세 개로 나누어 쓰는 이유는 실제 낸드 플래시 메모리의 사용 환경을 만들기 위해서 이다. 그리고 다시 20MB 파일을 128KB의 레코드 단위로 쓰면서 쓰기 지연시간을 비교하였다. 128KB는 테스트를 수행하는 낸드 플래시 메모리의 블록 단위이므로, 128KB 이상의 단위로 쓰더라도 128KB 이하의 세그먼트로 할당되기 때문에 제안한 알고리즘을 정확히 테스트 할 수 있다. 또한 일정한 파일을 쓰고 지우는 연산을 반복하면서 파일 삭제 시 걸리는 시간을 비교하였다.

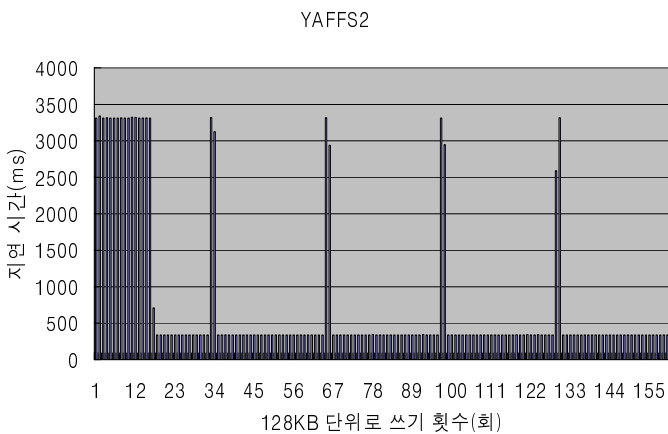
이 테스트 결과를 (그림 2)부터 (그림 5)까지의 그래프로 나타내면 다음과 같다.



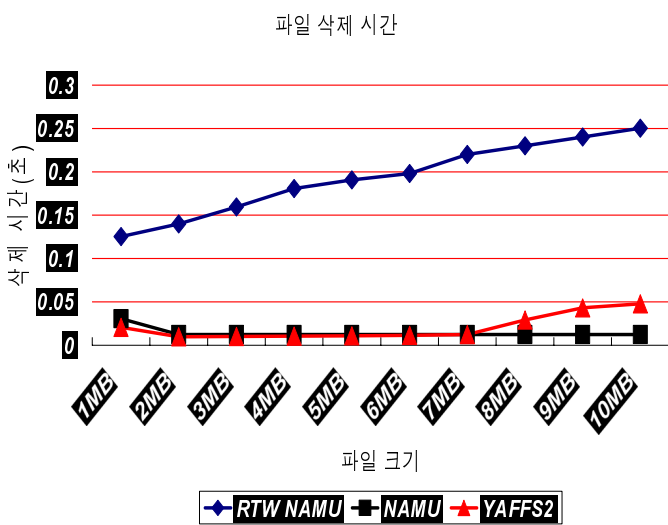
(그림 2) RTW_NAMU의 쓰기 지연시간



(그림 3) NAMU의 쓰기 지연시간



(그림 4) YAFFS2의 쓰기 지연시간



(그림 5) 파일 삭제 시간

(그림 2)는 RTW NAMU가 일정한 쓰기 지연 시간을 보장하는 것을 보여준다. RTW NAMU의 평균 쓰기 지연 시간은 2ms로 거의 쓰기 지연이 발생하지 않는다.

(그림 3)과 같이 NAMU는 유향 시간에 모든 블록에 대해 가비지 컬렉션을 수행하고, 본 테스트의 경우와 같이 계속 쓰기 연산이 일어날 때는 데이터를 저장하기 위해 필요한 공간 크기만큼의 블록을 대상으로 가비지 컬렉션을 수행한다. 따라서 저장 공간이 있을 때는 쓰기 지연 시간이 RTW NAMU와 비슷하지만 일단 데이터를 쓸 클린 페이지가 부족하게 되면 데이터를 쓸 때마다 300ms 정도의 쓰기 지연 시간을 갖는다. (그림 4)의 YAFFS2는 쓰기 연산이 일어날 때 낸드 플래시 메모리의 상태에 따라 가비지 컬렉션을 다르게 수행하기 때문에 최대 3초 정도의 쓰기 지연 시간을 갖는다.

(그림 5)과 같이 RTW NAMU는 파일을 삭제할 때 낸드 플래시 메모리의 블록을 삭제하는 연산을 수행하기 때문에 파일의 크기에 비례하여 삭제 시간이 증가한다. 그러나 10MB 파일을 지울 때 걸리는 시간은 0.3초 미만이기 때문에 멀티미디어 응용 사용자의 요구를 만족시킨다.

성능 측정 결과 RTW NAMU는 파일의 삭제 시간은 파일의 크기에 비례하여 길어지지만 사용자가 받아들일 만한 결과를 가지면, 멀티미디어 파일 시스템의 중요한 서비스 요소인 실시간 음성 녹음 및 영상 촬영을 위해 일정한 쓰기 지연 시간을 보장한다.

5. 결론 및 향후 연구과제

멀티미디어 응용의 조건을 만족하기 위해 멀티미디어 파일 시스템은 일정한 쓰기 지연 시간을 제공해야 한다. 이를 위해 낸드 플래시 메모리의 페이지는 유효 상태이거나 클린 상태여야 하고, 이것은 파일이 삭제될 때 생성되는 더티 페이지를 모두 클린 페이지로 바꿈으로써 가능하다. 더티 페이지를 클린 페이지로 바꾸는 연산은 한 블록에 유효한 페이지를 다른 블록으로 복사하고 그 블록을 삭제하는 비용이 든다. 이 비용을 최소화하기 위해 본 논문에서는 유효 페이지가 적은 블록에 세그먼트를 할당하고 파일 삭제 시에 더티한 페이지를 모두 클린 페이지로 만드는 알고리즘을 제시하였다. 본 알고리즘을 NAMU에 적용하여 RTW NAMU로 개선하여, NAMU와 RTW NAMU 그리고 YAFFS2의 쓰기 지연 시간과 삭제 시 걸리는 시간을 비교하였다. RTW NAMU는 NAMU와 YAFFS2에서 지원하지 않은 일정한 쓰기 지연 시간을 보장하며, 이를 통해 소형 멀티미디어 응용의 멀티미디어 콘텐츠의 녹음이나 촬영의 QoS를 지원할 수 있음을 보여준다.

본 논문에서 제안한 알고리즘에서 파일 삭제 시 연산 비용은 최악의 경우 세그먼트가 할당된 블록마다 블록을 구성하는 페이지보다 하나 작은 페이지를 복사해야 하는 비용을 가진다. 최악의 경우 파일 삭제 시 연산 비용을 낮추기 위한 연구가 이루어져야 한다.

6. 참고 문헌

- [1] Samsung Electronics, "NAND Flash Memory," <http://www.sec.co.kr/>.
- [2] Aleph One Company, "Yet Another Flash Filing System," <http://www.aleph1.co.uk/yaffs/>.
- [3] Hyojun Kim, Youjip Won, "MMFS: Mobile Multimedia File System for NAND Flash based Storage Device," in the IEEE CCNC 2006 proceedings, pp. 208-212.
- [4] 백승재, 최종무, "플래시 메모리 파일 시스템을 위한 순수도 기반 페이지 할당 기법에 대한 연구," 정보처리학회논문지 A 제13-A권 제5호(2006.10).
- [5] IOzone Filesystem Benchmark, http://www.iozone.org/docs/IOzone_msword_98.pdf