

# 페이지 비율 분석 기반의 NAND 플래시 메모리를 위한 가비지 컬렉션 기법

## (Garbage Collection Method for NAND Flash Memory based on Analysis of Page Ratio)

이 승 환<sup>†</sup>    옥 동 석<sup>†</sup>    윤 창 배<sup>†</sup>    이 태 훈<sup>†</sup>    정 기 동<sup>††</sup>  
 (Seung-Hwan Lee) (Dongseok Ok) (Changbae Yoon) (Taehoon Lee) (Kidong Chung)

**요 약** NAND 플래시 메모리는 부피가 작고, 가볍고, 소비전력이 낮으며 입출력이 빠르고 집적도가 높아 최근 임베디드 기기들에 널리 사용되고 있다. 그러나 NAND 플래시 메모리는 지움 연산을 수반하는 가비지 컬렉션 연산을 수행해야 한다. 게다가 지움 연산은 속도가 느리고, 각 블록마다 지움 연산 횟수가 제한이 있다. 따라서 제안하는 가비지 컬렉션 기법은 전체 지움 연산 횟수와 각 블록의 지움 횟수 편차를 감소시키고, 가비지 컬렉션 수행 시간을 최소화하는데 초점을 맞춘다. NAND 플래시 메모리는 유효 페이지, 무효 페이지, 빈 페이지로 구성되어 있다. 제안하는 기법은 페이지들의 비율을 이용해 가비지 컬렉션의 수행 시기를 결정하고 대상 블록을 선택한다. 그리고 할당 기법과 그룹 관리기법을 추가적으로 구현하였다. 실험 결과 제안한 정책은 기존의 Greedy나 CAT 기법에 비해 전체 지움 횟수를 최소화 하면서, 최대 82% 지움 횟수 편차를 감소시켰고, 최대 75%의 가비지 컬렉션 수행 시간을 단축시켰다.

**키워드** : NAND 플래시 메모리, 가비지 컬렉션, 균등화

**Abstract** NAND flash memory is widely used in embedded systems because of many attractive features, such as small size, light weight, low power consumption and fast access speed. However, it requires garbage collection, which includes erase operations. Erase operation is very slow. Besides, the number of the erase operations allowed to be carried out for each block is limited. The proposed garbage collection method focuses on minimizing the total number of erase operations, the deviation value of each block and the garbage collection time. NAND flash memory consists of pages of three types, such as valid pages, invalid pages and free pages. In order to achieve above goals, we use a page rate to decide when to do garbage collection and to select the target victim block. Additionally, We implement allocating method and group management method. Simulation results show that the proposed policy performs better than Greedy or CAT with the maximum rate at 82% of reduction in the deviation value of erase operation and 75% reduction in garbage collection time.

**Key words** : NAND flash memory, Garbage Collection, Wear-leveling

<sup>†</sup> 학생회원 : 부산대학교 컴퓨터공학과  
 cat@pusan.ac.kr  
 ok3@pusan.ac.kr  
 yunch@melon.cs.pusan.ac.kr  
 withsoul@pusan.ac.kr  
<sup>††</sup> 종신회원 : 부산대학교 컴퓨터공학과 교수  
 kdchung@psuan.ac.kr  
 논문접수 : 2008년 8월 20일  
 심사완료 : 2009년 7월 16일

Copyright©2009 한국정보과학회 : 개인 목적이거나 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 제15권 제9호(2009.9)

## 1. 서론

플래시 메모리는 비휘발성이며, 일반적인 컴퓨터 환경에서 사용되는 하드디스크보다 부피가 작고, 가볍고, 소비 전력이 낮으며, 기계적인 동작이 필요 없기 때문에 응답 시간이 빠르고, 충격에 대한 내구성이 뛰어난 장점을 가지고 있어 소형 기기의 저장장치로 적합하다[1]. 최근 유비쿼터스 컴퓨팅 환경의 발전으로 휴대폰, PDA, 자동차 네비게이션 등 이동 가능한 정보기기들이 널리 보급되고 있으며, 플래시 메모리는 이러한 휴대용 정보 기기의 저장장치로서 사용되며 지속적으로 발전하는 추세이다. 이러한 플래시 메모리는 크게 NAND 타입과

NOR 타입이 있다. 그 중 NAND 타입의 플래시 메모리는 NOR 타입의 플래시 메모리 보다 집적도가 뛰어나 용량이 크고, 쓰기 연산과 지움 연산이 빠른 특징을 가지고 있어 주로 사용자의 데이터를 저장하는 데에 사용되고 있다.

그러나 NAND 플래시 메모리는 기존 하드디스크와 다른 두 가지 단점이 있다[2,3]. 첫 번째 단점은 제자리 덮어쓰기가 불가능하여 갱신 연산을 수행할 때 동일한 주소에 바로 덮어 쓰기가 되지 않는다. 따라서 갱신 연산을 하기 위해서는 지움 연산이 추가적으로 발생한다. 두 번째 단점은 플래시 메모리를 구성하고 있는 블록은 지움 연산 횟수가 제한이 되어 있다. 따라서 제한 횟수를 초과하는 블록들은 이후 쓰기 오류가 발생하게 되고 블록은 배드 블록 처리를 해야 한다. 결국 블록들의 지움 횟수가 균등하게 유지되지 못하면 가용공간은 급격히 줄어드는 현상이 발생한다.

NAND 플래시 메모리 기반 임베디드 기기는 성능과 신뢰성을 만족시키기 위해 앞에서 언급한 두 가지 단점을 극복하는 것이 필요하다. 첫 번째 단점을 극복하기 위해서는 갱신 연산이 발생했을 때 다른 공간에 새로운 데이터를 쓰고 이전의 데이터는 무효화시키는 방법을 일반적으로 사용하고 있다. 그리고 새로운 공간 확보를 위해서는 무효화된 데이터를 가지고 있는 블록들에 대해서 '가비지 컬렉션(Garbage Collection)'을 수행한다. 이때 가비지 컬렉션 연산의 시간을 최소화하여야 시스템의 성능을 극대화 할 수 있다. 두 번째 단점을 극복하기 위해서는 블록의 지움 횟수를 고려하여 블록을 관리해야 한다. 지움 횟수가 지나치게 높은 블록의 활용을 줄이고 비교적 낮은 지움 횟수를 가지는 블록의 활용도를 높임으로써 전체 블록의 지움 횟수에 대한 균등화를 고려한 가비지 컬렉션 기법이 필요하다. 따라서 주로 지움 횟수가 낮은 블록에 대한 할당 연산과 가비지 컬렉션 연산을 수행해야 한다. 하지만 위의 두 가지 방법을 모두 만족 시키는 것은 쉽지 않다. 가비지 컬렉션의 연산 시간을 최소화하기 위해서는 무효율이 높은 블록을 대상블록으로 선정해야 하는데 이는 전체 블록의 지움 횟수를 균등하게 유지하지 않는다. 또는 지움 횟수를 균등화하기 위해서 지움 횟수가 낮은 블록에 가비지 컬렉션 연산을 수행하면 무효율이 높지 않은 블록들에 대해서는 연산 시간을 증가시키기 때문이다.

본 논문에서는 이와 같이 상충되는 방법들의 장점만을 최대한 활용할 수 있는 기법을 제안하여 시스템의 성능 감소 없이 신뢰성을 유지 하는 것을 목표로 한다. 이를 위해 균등화를 위한 빈 블록들의 할당 기법을 제안하고 균등화를 위한 가비지 컬렉션 기법을 제안한다. 특히 가비지 컬렉션 기법은 유효 페이지, 무효 페이지,

빈 페이지의 비율을 가지고 수행 시기와 수행 방법을 결정하여 최적의 연산 시간과 균등화를 모두 지원할 수 있도록 하였다. 추가로 가비지 컬렉션 대상 블록을 찾기 위한 탐색 영역도 최소화하면서 균등화 성능의 감소 없이 가비지 컬렉션 수행 속도를 향상시켰다.

본 논문의 구성은 다음과 같다. 2장에서는 NAND 플래시 메모리의 특징과 대표적인 NAND 플래시 전용 파일 시스템인 YAFFS, 그리고 기존의 가비지 컬렉션 연구에 대해 기술하고, 3장에서는 본 논문에서 제안하는 가비지 컬렉션 기법에 대해 설명한다. 4장에서는 제안하는 기법의 성능을 실험을 통해 평가하고, 5장에서는 결론과 향후 과제를 제시한다.

## 2. 관련연구

그럼 본 논문의 관련 연구로 NAND 플래시 메모리의 특징, NAND 플래시 전용 파일 시스템 및 기존 가비지 컬렉션 기법에 대해서 알아본다.

### 2.1 NAND 플래시 메모리 특징

플래시 메모리는 일종의 EEPROM(Electrically Erasable and Programmable ROM)으로 비휘발성의 메모리 반도체로 관련 기술도 활발히 연구되고 있다.

NOR 플래시 메모리는 바이트 단위 접근이 가능하며 집적도가 낮고 고가이기 때문에 소스코드 저장용으로 사용되며, NAND 플래시 메모리는 대용량이며 비용이 저렴하기 때문에 데이터 저장용으로 사용하고 있다. 따라서 최근 임베디드 기기에서 NAND 플래시 메모리를 많이 채택하고 있다.

특히 NAND 플래시 메모리는 한 셀 당 1비트를 저장하는 SLC 타입과 2비트 이상을 저장하는 MLC 타입으로 나눌 수 있고, MLC는 SLC 보다 대용량화가 쉽고, 가격이 낮아 널리 보급이 되고 있다[4]. 하지만 SLC은 블록 당 지움 횟수가 10만 번 정도 보장되는 반면 MLC는 5천~1만 번으로 SLC 보다 낮다. 이 지움 횟수를 넘는다면 배드 블록이 되어 일관성을 보장하지 못한다. 때문에 플래시 메모리의 균등화를 고려한 가비지 컬렉션 기법이 무엇보다도 요구되고 있다.

### 2.2 NAND 플래시 전용 파일 시스템

NAND 플래시 메모리는 기존의 하드디스크와의 하드웨어 및 연산의 차이가 있어 기존의 하드디스크용 파일 시스템과 호환성을 유지하는 FTL이나 전용 파일 시스템이 필요하다.

NAND 플래시 메모리를 위한 대표적인 전용 파일 시스템으로는 YAFFS가 있다. YAFFS[5]는 2002년 Alphone사에서 개발된 NAND 플래시 전용 파일 시스템이다. YAFFS는 NAND 플래시 메모리의 제자리 덮어쓰기가 되지 않는 문제점을 해결하기 위하여 로그 구조

파일 시스템을 기반으로 하였다[4]. 따라서 파일에 대한 갱신 연산을 추가 쓰기 연산으로 변형하여 새로운 페이지를 할당하는 외부 갱신 기법을 사용해 FTL에 비해 전체 블록을 고루 사용 할 수 있다. 그리고 빈 블록을 확보하기 위해서 가비지 컬렉션을 수행을 하여 플래시 메모리의 빈 공간을 확보한다. 하지만 YAFFS의 할당과 가비지 컬렉션 연산은 플래시 영역을 순차적으로 탐색하여 수행하므로 일반적으로 전체 블록의 지움 횟수의 균등화는 지원하지 않는다. 본 논문에서 제안하는 가비지 컬렉션 기법과 기존 기법을 YAFFS 기반의 전용 파일 시스템에서 구현하여 성능을 측정한다.

### 2.3 기존의 가비지 컬렉션 기법

플래시 메모리 기반 시스템과 관련된 연구들은 대부분 가비지 컬렉션 연산에 소요되는 비용을 최소화 목표를 두고 있다[6-8]. 가비지 컬렉션 연산 시 기법에서는 사용률이 가장 낮은 블록을 가비지 컬렉션 대상 블록으로 선택하여 가비지 컬렉션 연산 시 블록 내 유효페이지를 복사하는 오버헤드를 최소화하여 공간 확보율이 높다. 하지만 블록의 균등화를 고려하지 않아 특정 블록에 가비지 컬렉션 연산이 집중되는 최악의 경우가 발생할 수 있고, 블록 간 지움 횟수의 편차가 크게 되어 배드 블록이 빨리 발생 하는 단점이 있다. Greedy 기법은 단순히 무효화 페이지가 가장 적은 블록을 선택하는 기법이며 이는 균등화를 고려하지 않는다. [9]는 Greedy 기법을 개선하여 플래시 메모리를 이용하여 최신 데이터를 캐싱하여 전체 지움 연산 횟수를 낮추었다. [10]은 가비지 컬렉션에 영향을 끼치는 요소를 이용률, 무효율, 순수도로 정의하고 순수도를 높게 유지하는 할당 정책을 제안하여 가비지 컬렉션 연산 시간을 낮추었다. [8]은 데이터를 Hot/Cold로 구분하여 할당정책을 수행한다. 하지만 [5]는 지움 횟수를 고려한 할당 정책을 통해 블록의 균등 사용을 지원 하지만 할당된 블록들에 대해서는 균등화 정책이 없는 한계가 있고, 역시 가비지 컬렉션 시 균등화 정책에 대한 고려가 없다. 할당된 블록들의 지움 횟수를 고려한 균등화 정책으로는 CAT 기법이 있다. CAT 기법은 블록의 사용률(u)과 블록이 할당된 시간(age), 그리고 블록의 지움 횟수(ec)를 고려해 블록의 지움 비용을 계산하며, 비용이 가장 작은 값을 가지는 블록을 가비지 컬렉션 대상 블록으로 선택을 하기 때문에 블록 간 지움 횟수의 편차가 작은 장점이 있다. 하지만 가비지 컬렉션 연산 시 복사하는 유효 페이지의 수가 Greedy기반 기법에 비해 많아 회수되는 빈 페이지의 수가 적어 전체 지움 횟수가 많아 질 수 있고, 전체 블록에 대해 비용을 계산할 시 연산 시간이 오래 걸린다는 단점이 있다. 따라서 본 논문에서는 가비지 컬렉션 연산 시 전체 지움 횟수를 최소화함과 동시에 블

록의 지움 횟수를 균등화를 고려한 가비지 컬렉션 기법을 제안하였다.

### 3. 페이지 비율을 고려한 가비지 컬렉션 기법

가비지 컬렉션 연산의 주요 목적은 플래시 메모리의 가용 공간을 확보하는 것이므로 가비지 컬렉션 연산을 자주하면 플래시 메모리의 가용 공간을 최대한으로 확보할 수 있다. 하지만 빈번한 가비지 컬렉션 연산은 전체 블록의 지움 횟수를 높게 되고 블록의 수명에 영향을 미친다. 따라서 가비지 컬렉션 연산 횟수를 최소화해서 전체 지움 횟수를 낮추어야 한다. 또한 가비지 컬렉션 대상 블록이 특정 블록에 집중되면 해당 블록의 지움 횟수를 증가시켜 수명을 단축시키는 결과를 초래할 수 있기 때문에 전체 블록의 지움 횟수를 균등하게 유지할 수 있는 가비지 컬렉션 정책이 필요하다.

이 절에서는 가비지 컬렉션의 수행 조건을 결정하고, 서로 다른 수행 조건에서 가비지 컬렉션 대상 블록을 효율적으로 선택하여 전체 블록의 지움 횟수와 전체 블록의 지움 횟수 편차를 최소화하는 기법을 제안한다. 그리고 가비지 컬렉션 연산 시 탐색 범위를 제한하여 가비지 컬렉션 수행 시간을 최소화 할 수 있는 기법을 제안한다.

#### 3.1 페이지 비율에 따른 수행 조건

플래시 메모리의 페이지는 사용되지 않은 빈 페이지, 데이터를 저장하고 있는 유효 페이지, 삭제가 필요한 무효 페이지로 구분할 수 있다. 각 페이지들이 차지하는 비율은 파일연산 시마다 항상 바뀌게 된다.

그림 2의 (a) 상태에서 새로운 파일이 생성이 되면 데이터가 늘어나므로 유효페이지 수는 증가하고 빈 페이지 수는 감소하여 (b) 상태가 된다. 또한 (c) 상태에서 파일이 삭제가 되면 유효페이지 수는 감소하고, 무효페이지 수는 증가한다. 따라서 (d) 상태가 되는 것을 알 수 있다.

본 논문에서는 파일이 생성 또는 삭제 시 달라지는 페이지들의 비율을 이용해서 다음 두 가지 조건을 만족하면 가비지 컬렉션을 수행한다.

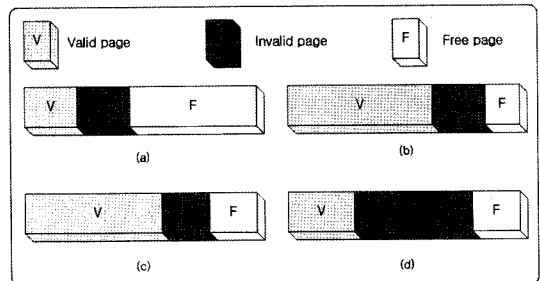


그림 1 플래시 메모리의 페이지 사용률

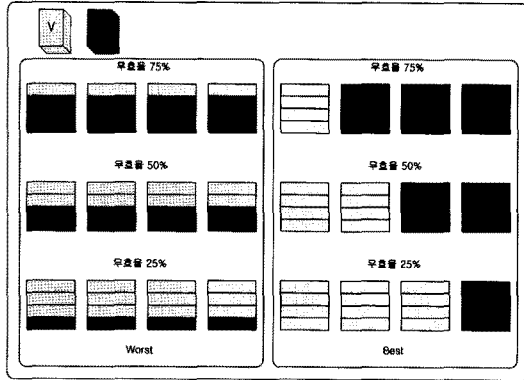


그림 2 무효율과 블록 상태의 관계

첫 번째 조건은 그림 2의 (b) 상태와 같이 새로운 파일 생성 시 감소하는 빈 페이지의 비율이 임계값 이하일 때 가비지 컬렉션을 수행한다. 임계값이 Free/(Valid+Invalid+Free)의 이하일 경우에 수행하며 임계값은 실험을 통해 설정하였다. 즉 새로운 빈 공간 확보가 필요 하게 될 때까지 가비지 컬렉션 연산을 최대한 미루어 전체 지움 횟수의 증가를 억제할 수 있다.

두 번째 조건은 그림 2의 (d) 상태와 같이 기존 파일 삭제 시 증가하는 무효 페이지의 비율이 임계값 이상일 때 가비지 컬렉션을 수행한다. 무효 페이지의 비율은 식 (1)을 통해 확인하며 모든 페이지 단위 I/O에 대해 무효 페이지 비율을 계산하게 된다. 첫 번째 조건과 같이 가비지 컬렉션 연산을 최대한 미루기만 하면 지움 횟수의 증가를 어느 정도 억제할 수 있지만 할당을 하기 위한 빈 블록은 많이 확보할 수가 없다. 따라서 추가적인 가비지 컬렉션을 수행하여 빈 블록의 개수를 충분히 유지할 수 있도록 한다. 가비지 컬렉션 연산 시 대상 블록 내 유효 페이지 복사 비용은 오버헤드로 작용하는데 이는 무효율에 따라 달라진다. 무효율은 식 (1)과 같이 사용된 페이지(nInvalid\_page+nValid\_page) 중 무효 페이지(nInvalid\_page) 비율이며 사용된 페이지는 데이터가 기록 중이거나 기록되었던 유효 페이지와 무효 페이지의 합으로 나타낼 수 있다.

$$Invalidity\_rate = \frac{nInvalid\_page}{nInvalid\_page + nValid\_page} \quad (1)$$

할당된 블록 중 무효율에 따른 블록의 상태는 그림 3과 같다. 무효율이 증가하면 블록전체가 무효 페이지로만 구성된 블록의 수가 많이 존재 하거나 블록내의 유효 페이지 수가 낮게 유지된다. 따라서 무효율이 임계값 이상으로 증가했을 때 가비지 컬렉션을 수행하면 오버헤드로 작용하는 유효 페이지 복사비용을 최소화 할 수 있다.

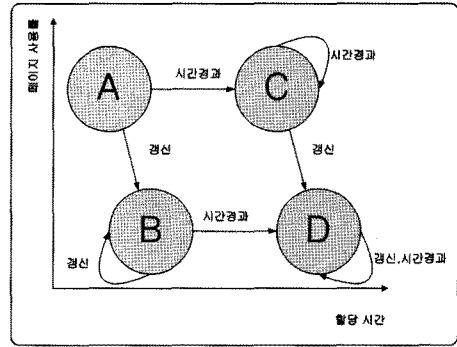


그림 3 블록의 상태도

### 3.2 대상 블록

할당된 블록은 그림 4와 같이 페이지 사용률과 할당 시간에 따라 4가지 상태가 존재한다. 그림 4에서 A는 블록이 할당되고 데이터가 쓰인 직후의 상태를 나타낸다. 할당 직후 블록 내 갱신 또는 전체 블록의 지움 횟수를 삭제 연산이 발생하게 되면 B와 같은 상태가 되고 짧은 시간 안에 갱신 연산이 발생했으므로 핫 블록이라 정의할 수 있다. 블록이 할당되고 오랫동안 갱신 또는 삭제 연산 없이 유지된 블록은 C의 상태가 되고 오랫동안 갱신 또는 삭제 연산이 일어나지 않으므로 콜드 블록이라고 정의할 수 있다. D의 상태는 B와 C의 상태에서 시간이 지나거나 갱신 연산이 발생했을 때 나타난다. 따라서 할당 시간이 길며 적은 수의 유효 페이지를 유지하고 있는 D 상태가 가비지 컬렉션 대상 블록으로 가장 적합하다. 하지만 상황에 따라서는 B와 C에 대해서도 가비지 컬렉션을 해야 한다.

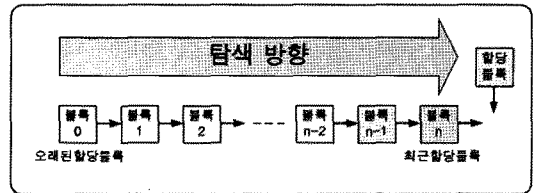


그림 4 가비지 컬렉션 대상 블록 탐색 방향

3.2.1에서 제시한 첫 번째 조건에서는 빈 페이지 비율이 일정 값 이하로 떨어지면 가비지 컬렉션을 수행해야 되므로 공간 확보가 가장 요구된다. 때문에 첫 번째 조건에 의한 가비지 컬렉션 시 블록 내에 유효 페이지가 가장 적은 블록을 선택하는 Greedy 정책을 적용한다. 따라서 가비지 컬렉션 대상 블록은 페이지 사용률이 낮은 D와 B가 될 확률이 높다.

두 번째 조건에서는 무효율이 일정 수준 이상으로 올라가면 가비지 컬렉션을 수행한다. 무효율이 높으면 블

록 내에 무효 페이지의 수가 많은 것이므로 가비지 컬렉션 오버헤드가 감소한다. 따라서 지움 횟수를 고려하여 가비지 컬렉션 대상 블록을 찾아야 하므로 B, C, D 중 지움 횟수를 고려하여 비용이 낮은 블록을 선택해야 한다.

$$Cost\_2 = u \times ec \quad (2)$$

$$Cost\_1 = u \quad (3)$$

위에서 말한 것처럼 조건에 따라 가비지 컬렉션 대상 블록을 다르게 선정을 해야 되므로 조건에 맞는 대상 블록을 선택하기 위해 조건에 따라 블록의 가비지 컬렉션 비용을 다르게 계산한다. 따라서 첫 번째 조건에서는 공간 확보를 위해 식 (2)와 같이 블록의 페이지 사용률(u)을 지움 비용으로 하고 두 번째 조건에서는 균등화를 위해 식 (3)과 같이 페이지 사용률(u)과 지움 횟수(ec)를 동시에 고려한 지움 비용을 계산 하여 가장 작은 비용의 블록을 선택한다.

또한 비용 계산 시 할당된 지 오래된 블록부터 탐색을 하여 같은 비용을 가지는 경우 할당된 지 오래된 블록을 우선적으로 가비지 컬렉션 연산을 수행하여 빈번하게 갱신되는 블록에 대한 지움 연산을 방지할 수 있도록 한다.

따라서 플래시 메모리에서 공간 확보가 필요한 조건과 가비지 컬렉션을 하여도 오버헤드가 적은 조건을 판단하여 조건에 따라 대상 블록 선정을 다르게 고려하여 추가적인 오버헤드를 최소화하면서 블록의 지움 횟수를 균등화 할 수 있다.

### 3.3 그룹 관리 기법

최적의 가비지 컬렉션 대상 블록을 선정하기 위해서는 할당되어 있는 전체 블록의 가비지 컬렉션 비용을 계산하여 최적의 비용을 가지는 블록을 선택하는 것이 가장 좋다. 그러나 할당된 블록의 수에 비례하여 탐색시간이 증가하는 오버헤드가 발생 한다. 이러한 오버헤드를 줄이기 위해 탐색 범위를 최소화하고 최적의 대상 블록을 선정하는 블록의 그룹 관리 기법을 제안한다.

그룹은 플래시 메모리의 블록들이 순차적으로 나열했을 때 일정 개수 블록 단위로 묶는다. 각 그룹은 그룹에 속하는 블록들의 정보를 유지하여 가비지 컬렉션 시 이를 판단한다. 그 정보는 그룹에 속하는 블록 중 할당

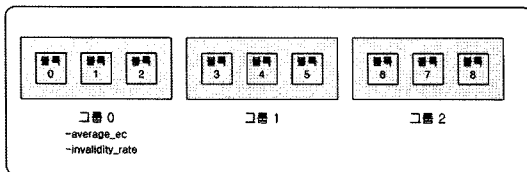


그림 5 그룹 관리 기법

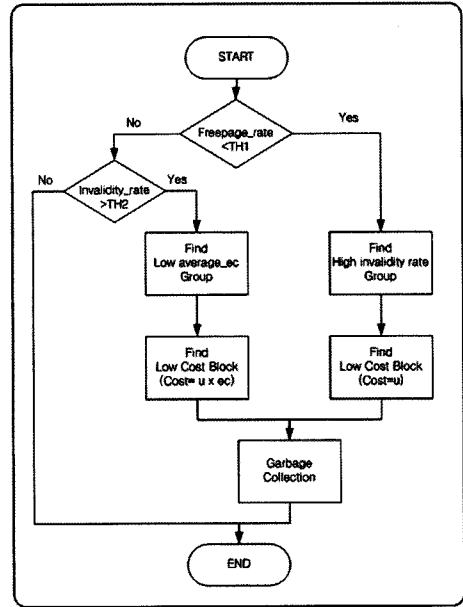


그림 6 가비지 컬렉션 순서도

된 블록들의 지움 횟수 평균값과 무효율이다. 가비지 컬렉션 수행 조건에 따라 대상 블록 판단을 달리 하는 것처럼 그룹 역시 가비지 컬렉션 조건에 따라 탐색 대상 그룹을 다르게 한다. 따라서 빈 페이지 비율이 일정 수준 이하일 때 공간 확보를 위해서 그룹의 무효율 값을 이용하고, 플래시 메모리의 균등화를 위해서는 그룹의 지움 횟수 평균값을 사용하여 탐색 대상 그룹을 선택한다. 탐색 대상 그룹이 선정되면 그룹 내 속하는 블록만을 탐색하여 가비지 컬렉션 대상 블록을 선정한다.

블록의 그룹 관리 기법은 가비지 컬렉션 대상 블록 선정 시 그룹의 개수와 그룹 내 할당된 블록의 수만큼만 탐색하여 할당 된 블록 전체를 탐색하는 오버헤드를 최소화 할 수 있다. 전체 그룹의 개수는 성능 측정을 통해 최적의 값을 선택해야 한다.

그룹 관리 기법까지를 모두 포함한 가비지 컬렉션 수행 흐름은 그림 7과 같으며 이를 페이지 단위의 I/O마다 수행한다. 최초 현재 남은 빈 페이지 비율이 일정 수준 이하인지를 판단하여 조건을 만족하면 공간 확보를 위한 가비지 컬렉션, 그렇지 않으면 전체 무효율 값을 판단하여 일정 수준 이상의 무효율을 가질 때 균등화를 위한 가비지 컬렉션 연산을 수행한다.

## 4. 성능평가

### 4.1 실험 환경

제안된 기법의 성능을 평가하기 위해 NAND 플래시 시뮬레이터를 사용하였고, 표 1과 같은 파일 연산 조건

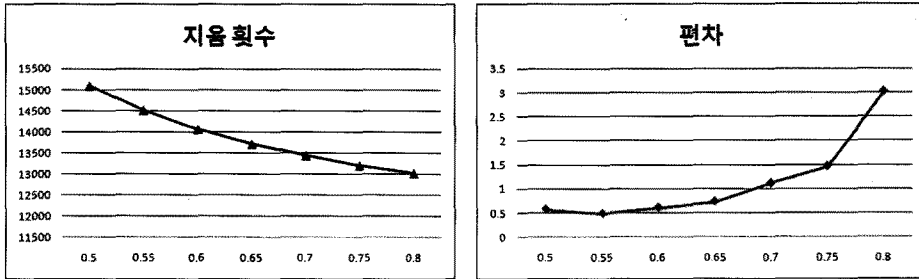


그림 7 무효율 TH에 따른 전체 지움 횟수 및 편차

을 가지고 성능을 평가하였다. 로컬리티는 x/y에서 전체 파일 중 x% 파일에 전체 연산 중 y% 연산이 집중되는 것으로 50/50은 낮고 10/90은 높다고 말할 수 있다.

가비지 컬렉션 수행 조건 중 첫 번째 조건을 결정하기 위한 빈 페이지 비율 임계값은 빈 페이지가 64개 이하로 낮아질 때로 설정한다. 그 이유는 할당할 블록과 가비지 컬렉션 시 복사 대상 블록의 여분을 가져야하므로 최소 2개 블록은 유지해야 되기 때문이다.

두 번째 조건을 결정하기 위한 무효율 임계값을 설정하기 위해서 20/80의 실험 조건에서 5만번의 입력 연산을 설정하고, 무효율의 임계값 변화에 따라 지움 횟수와 편차를 측정하였다. 측정 결과 그림 8과 같이 무효율 임계값이 높을수록 지움 횟수는 낮아지나 지움 횟수 편차가 높아져 균등화가 되지 않고 임계값이 낮을수록 지움 횟수 편차는 낮아져 블록의 균등화가 되지만 지움 횟수가 높게 나타났다. 무효율 임계값은 파일 입력 트레이스에 따라 최적의 값이 약간씩 달라지지만 일반적으로 0.6으로 설정했을 때 높은 성능을 나타냄을 알 수 있어 이후 무효율 임계값은 0.6으로 고정하였다. 그리고 제안하는 기법 외에는 무효율을 사용한 가비지 컬렉션을 수행하지 않아 추가적인 지움 연산이 일어나지 않도록 하였다.

표 1 실험 조건 표

조건	값
블록 개수	64
블록당 페이지 개수	32
파일 개수	200
파일 크기	0.5~16.5KB
파일 연산 횟수	10만번
로컬리티	50/50, 40/60, 30/70, 20/80, 10/90

4.2 비용 계산 기법에 대한 성능 비교

가비지 컬렉션의 조건에 상관없이 Cost\_1 또는 Cost\_2만을 이용한 기법들과 첫 번째 조건에는 Cost\_1을 두 번째 조건에는 Cost\_2를 혼합 적용한 기법의 성

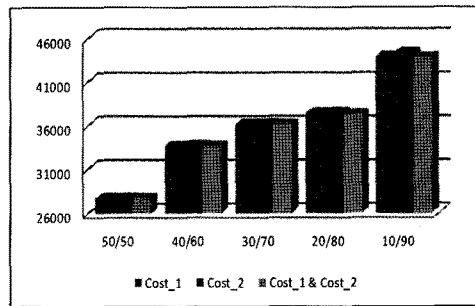


그림 8 비용 계산에 대한 전체 지움 횟수

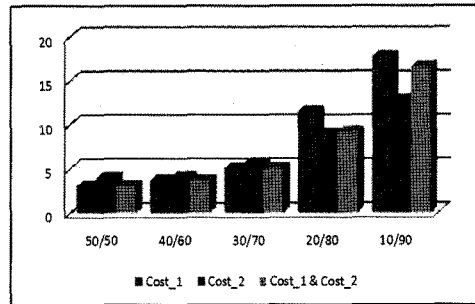


그림 9 비용 계산에 대한 지움 횟수 편차

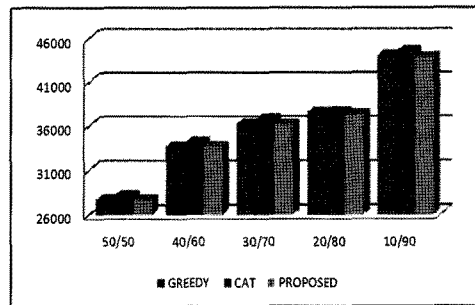


그림 10 전체 지움 횟수

능을 다양한 로컬리티 기반으로 측정하였다. 그림 9,10과 같이 Cost\_1은 페이지 사용률만을 고려해 지움 횟수

는 낮게 측정되지만 편차가 매우 높아지는 결과를 보이고 Cost\_2를 고려한 기법은 지움 횟수의 편차는 낮아지거나 전체 지움 횟수가 높게 유지되었다. Cost\_1과 Cost\_2를 혼용한 기법은 지움 횟수는 낮게 유지되면서 지움 횟수의 편차도 중간 정도로 유지가 되어 효과적이라는 결론을 내릴 수 있다.

4.3 기존 기법과의 성능 비교

제안하는 기법과 기존의 Greedy 기법과 CAT 기법과 성능을 비교하였다. 제안하는 기법은 4.2절에서 가비지 컬렉션 대상 블록 선정 시, 수행 조건에 따라 Cost\_1과 Cost\_2를 혼용해서 사용하였고 할당 기법을 추가하였다. Greedy 기법과 CAT 기법은 빈 페이지 개수가 64개 이하일 때 가비지 컬렉션을 수행하였고 그림 11,12와 같이 결과가 나왔다. Greedy 기법은 공간 확보율이 좋아 CAT 기법에 비해 전체 지움 횟수가 낮았고, CAT 기법은 Greedy 기법에 비해 지움 횟수 편차가 낮았다. 제안하는 기법은 두 기법의 장점만을 나타내어 전체 지움 횟수는 Greedy 기법에 근접하였으며, 특히 로컬리티가 높을 때는 오히려 가장 낮은 지움 횟수가 나타났다. 블록의 지움 횟수 편차 역시 할당 기법의 영향으로 CAT 기법에 비해 8%, Greedy 기법에 비해 82% 정도 감소시켰다.

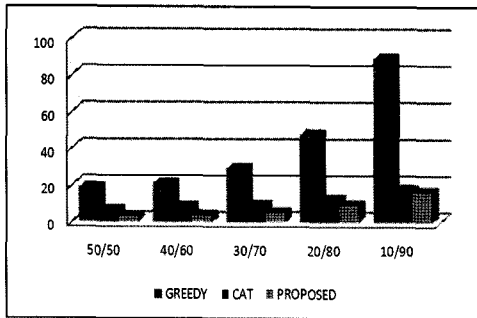


그림 11 각 블록의 지움 횟수 편차

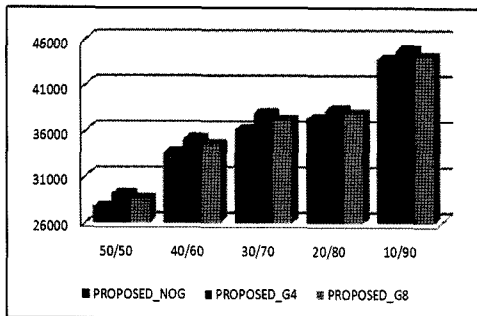


그림 12 그룹 내 블록 개수에 대한 전체 지움 횟수

4.4 그룹 관리 기법의 성능 비교

본 논문에서는 가비지 컬렉션 대상 블록 탐색 시 소요 되는 오버헤드를 최소화하기 위해 그룹 관리 기법을 사용 하였고 성능의 감소가 없는지 측정 해보았다. 그룹을 설정하지 않은 기법과 그룹 내 블록 수를 4, 8개로 설정한 기법을 비교하였고 가비지 컬렉션 대상 그룹 및 블록은 첫 번째 조건일 때 공간 확보를 위해 무효율이 가장 높은 그룹을 선택하고 Cost\_1을 적용해 대상 블록을 선택한다. 두 번째 조건일 때 균등화를 위해 지움 횟수 평균값이 가장 낮은 그룹을 선택하고 Cost\_2를 적용해 대상 블록을 선택한다.

그림 13,14에서 로컬리티가 낮은 실험에서는 그룹 기법의 지움 횟수가 비교적 높게 나타나 최적의 블록이 선정되지 않는 경향을 보였고 지움 횟수 편차 역시 그룹 기법을 사용하지 않은 것보다 대체적으로 높게 나타났다. 하지만 로컬리티가 높은 실험에서는 그룹을 8개의

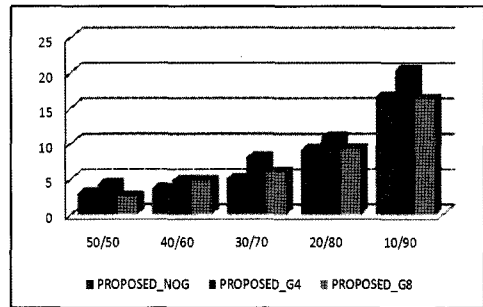


그림 13 그룹 내 블록 개수에 대한 지움 횟수 편차

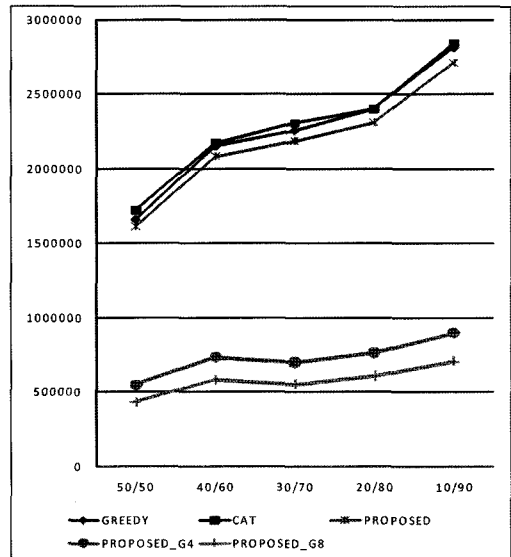


그림 14 가비지 컬렉션 시 탐색 시간

블록을 유지하는 기법이 그룹 기법을 사용하지 않은 것  
과 거의 유사한 전체 지움 횟수를 나타냈고, 편차는 오  
히려 약간 낮게 나타나 성능의 차이가 적음을 보였다.  
일반적으로 파일 사용 시 특정 파일에 연산이 집중되는  
경우가 많기 때문에 실제 사용 시 성능의 감소가 거의  
없다고 볼 수 있다.

그룹 관리 기법은 가비지 컬렉션 시 탐색시간을 줄이  
는 것이 목표이다. 따라서 그룹 관리 기법을 사용하지  
않은 기법들과 가비지 컬렉션 시 참조하는 블록 수를  
측정 해보았다. 그림 15와 같이 그룹 관리 기법을 사용  
했을 때, 그룹 관리 기법을 사용하지 않은 기법의 탐색  
시간을 75% 정도 단축 시켜 3~4배의 성능향상이 있어,  
가비지 컬렉션 연산 시 응답 속도가 낮음을 보였다.

## 5. 결론

본 논문에서는 플래시 메모리의 수명을 최대한 유지  
하기 위한 균등화 정책을 제안하였다. 균등화 정책의 목  
표는 전체 지움 횟수와 지움 편차를 최소화 하는  
것이다. 따라서 지움 횟수를 낮추기 위해 페이지들의 비  
율을 판단하여 가비지 컬렉션 수행 조건을 제안하였고,  
지움 횟수 편차를 낮추기 위해 할당 기법과 수행 조건  
에 따라 공간 확보 또는 균등화에 따라 비용 계산을 달  
리하여 상황에 맞는 블록 선택 기법을 적용하였다.

성능 측정 결과 기존의 기법과 비교해 전체 지움 횟  
수를 공간 확보율이 좋은 Greedy 기법과 유사하게 유  
지하면서 전체 블록의 지움 횟수 편차는 최대 82% 정  
도 감소하여 블록의 지움 횟수 균등화를 보장하였다. 또  
한 가비지 컬렉션 연산 시간을 최소화하기 위해 그룹  
기법을 제안하였고. 성능 측정 결과 기존 기법들에 비해  
75% 시간 단축을 할 수 있었다.

향후 본 연구는 Hot/Cold 데이터를 효율적으로 구분  
하여 할당 기법을 보완하여 유효 페이지 복사 횟수 및  
지움 횟수를 더욱 감소하는 기법 연구와 무효를 임계값  
을 적용적으로 변화시켜 로컬리티가 변하는 입력에도  
추가적인 오버헤드 없이 보다 효율적으로 균등화를 지  
원할 수 있는 방향으로 나아 갈 것이다.

## 참 고 문 헌

- [1] M-systems Flash Disk Pioneer, "Implementing MLC NAND Flash for Cost-Effective, High-Capacity Memory," [http://www.data-io.com/pdf/NAND/MSystems/Implementing\\_MLC\\_NAND\\_Flash.pdf/](http://www.data-io.com/pdf/NAND/MSystems/Implementing_MLC_NAND_Flash.pdf/)
- [2] F.Douglis, R. Caceres, F. Kaashoek, K. Li, B. Marsh and J.A Tauber, "Storage Alternatives for Mobile Computers," *Proceedings of the 1st Symposium on Operating Systems Design and Implementation(OSDI)*, pp.25-37, 1994.
- [3] Samsung Electronics, "NAND Flash data sheet," [http://www.samsung.com/global/business/semiconductor/products/flash/Products\\_NANDFlash.html/](http://www.samsung.com/global/business/semiconductor/products/flash/Products_NANDFlash.html/)
- [4] Mei-Ling Chiang, Paul C. H. Lee, Ruei-Chuan Chang, "Cleaning policies in mobile computers using flash memory," *Journal of Systems and Software*, vol.48, 1999.
- [5] Alepha One company, "The Yet Another Flash Memory File System(YAFFS)," <http://www.yaffs.net/>
- [6] Intel Corporation, "Flash File Systems Overview," <http://www.intel.com/>
- [7] M. Rosenblum, and J. K. Ousterhout, "The Design and Implementation of a Log-Structured File System," *ACM Transactions on Computer Systems*, vol.10, no.1, pp.26-52, 1992.
- [8] Ohhoon Kwon, Kern Koh, "Swap-aware Garbage Collection for NAND Flash Memory Based Embedded Systems," *Seventh International Conference on Computer and Information Technology*, pp. 787-792, 2007.
- [9] 김경운, 김영필, 송인준, 유혁, "Greedy 방법을 개선한 플래시 메모리 지움 정책", *한국정보처리학회 춘계학술발표대회 논문집*, 제11권 제1호, pp.1685-1688, 2004.
- [10] 백승재, 최중무, "블록 클리닝 비용 분석에 기초한 MODA할당 정책 설계 및 구현", *정보과학회논문지 시스템 및 이론*, 제34권 제11호, pp.599-609, 2007.



이 승 환

2007년 부산대학교 컴퓨터공학과 졸업(학사). 2009년 부산대학교 컴퓨터공학과 졸업(석사). 관심분야는 내장형 시스템, 파일 시스템

옥 동 석

정보과학회논문지: 컴퓨팅의 실제 및 레터 제 15 권 제 1 호 참조



윤 창 배

2008년 부산대학교 컴퓨터공학과 졸업(학사). 2008년~현재 부산대학교 컴퓨터공학과 석사과정. 관심분야는 내장형 시스템, 파일 시스템



이 태 훈

정보과학회논문지 : 컴퓨팅의 실제 및 레터  
제 15 권 제 1 호 참조

정 기 동

정보과학회논문지 : 컴퓨팅의 실제 및 레터  
제 15 권 제 1 호 참조