Journal of The Korea Society of Computer and Information
Vol. 20 No. 12, pp. 1-8, December 2015

www.ksci.re.kr
http://dx.doi.org/10.9708/jksci.2015.20.12.001

Cold Data Identification using Raw Bit Error Rate in Wear Leveling for NAND Flash Memory

Sang-Ho Hwang*, Jong Wook Kwak**, Chang-Hyeon Park***

Abstract

Wear leveling techniques have been studied to prolong the lifetime of NAND flash memory. Most of studies have used Program/Erase(P/E) cycles as wear index for wear leveling. Unfortunately, P/E cycles could not predict the real lifetime of NAND flash blocks. Therefore, these algorithms have the limited performance from prolonging the lifetime when applied to the SSD. In order to apply the real lifetime, wear leveling algorithms, which use raw Bit Error Rate(rBER) as wear index, have been studied in recent years. In this paper, we propose CrEWL(Cold data identification using raw Bit error rate in Wear Leveling), which uses rBER as wear index to apply to the real lifetime. The proposed wear leveling reduces an overhead of garbage collections by using HBSQ(Hot Block Sequence Queue) which identifies hot data. In order to reduce overhead of wear leveling, CrEWL does not perform wear leveling until rBER of the some blocks reaches a threshold value. We evaluate CrEWL in comparison with the previous studies under the traces having the different Hot/Cold rate, and the experimental results show that our wear leveling technique can reduce the overhead up to 41% and prolong the lifetime up to 72% compared with previous wear leveling techniques.

▸ Keyword : NAND Flash Memory, Wear Leveling, Garbage Collection, Cold Data, Raw Bit Error

I. Introduction

저장매체로 널리 사용되고 있는 낸드 플래시는 하드디스크와 다른 특징을 가지고 있다. 낸드 플래시의 읽기 및 쓰기 연산은 페이지 단위로 이루어지고, 소거 연산은 블록 단위로 이루어진다. 낸드 플래시는 이미 데이터가 있는 영역에는 다른 데이터를 쓰지 못하기 때문에, 데이터 수정 시 제자리 갱신이 불가능하다[1].

소거 연산이 필요하고 제자리 갱신이 불가능한 특성으로 인해 낸드 플래시를 사용하는 시스템은 별도의 플래시 전환 계층(FTL: Flash Translation Layer)을 사용한다. 플래시 전환 계층은 사상 테이블(Mapping Table)을 이용하여 논리 주소를 낸드 플래시 메모리의 물리적 주소로 연결시키고, 가비지 컬랙션(Garbage Collection)을 이용하여 업데이트 및 삭제에서 발생될 수 있는 많은 무효화 페이지를 재사용할 수 있다. 또한 FTL은 마모도 평준화(Wear Leveling) 기법을 사용하여 특정 블록이 빨리 소모되는 것을 방지하여 저장매체의 수명을 증가시킨다. 뿐만 아니라 플래시 전환 계층 안에는 공장 출하 시 발생된 배드 블록 및 사용 중에 발생되는 배드 블록을 관리하기 위한 블록 관리(Block Management) 기능도 포함되어 있다.

특정 블록이 빠르게 배드 블록이 되는 것을 방지하지 못한다면, 저장매체는 활용할 수 있는 블록의 수가 줄어들게 되고, 결국 줄어든 블록에 의해 가비지 컬랙션이 자주 수행되어 수명뿐만 아니라 저장매체의 응답 속도도 떨어지게 된다. 따라서 저장매체의 수명 및 성능을 위해 마모도 평준화에 대한 연구가 지금까지 이루어져 왔다[2-9].

• First Author: Sang-Ho Hwang, Corresponding Author: Chang-Hyeon Park
*Sang-Ho Hwang(snailcom@ynu.ac.kr), Dept. of Computer Engineering, Yeungnam University
**Jong Wook Kwak(kwak@yu.ac.kr), Dept. of Computer Engineering, Yeungnam University
***Chang-Hyeon Park(park@yu.ac.kr). Dept. of Computer Engineering, Yeungnam University
• Received: 2015. 08. 18, Revised: 2015. 10. 05, Accepted: 2015. 11. 12.
• This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(No. NRF-2014R1A1A2057146).

이러한 연구들은 대부분 P/E(Program/Erase) 수에 따라 마모도 평준화를 수행하고 있지만, 낸드 플래시 칩 내의 블록들의 수명은 일정하지 않기 때문에 실제 낸드 플래시 환경에 적용하게 되면 성능이 떨어지게 된다[5-8].

실제 블록 수명을 반영하기 위해, 최근에는 rBER(raw Bit Error Rate)을 마모도 평준화에 이용하는 연구가 이루어지고 있다[10-12]. 낸드 플래시 칩의 성능을 측정한 많은 연구에서는 P/E에 따라 블록의 rBER가 증가함을 보이고 있으며[13-16], rBER 기반의 마모도 평준화 연구들은 이러한 정보에 기초하고 있다. 따라서 rBER 기반의 마모도 평준화 연구들은 실제 블록의 수명을 예측하기 위해 ECC(Error Correction Codes)에서 이루어지는 에러 검출/에러 정정의 정보를 이용하고 있다.

본 논문에서는 실제 수명을 예측할 수 없는 P/E 수를 사용하지 않고, ECC에서 생성되는 rBER 정보를 활용하여 마모도 평준화를 수행하며, Hot/Cold 데이터 판별을 기존의 오버헤드가 많은 접근 빈도 측정 기법이 아닌 새로운 방법을 사용하는 CrEWL(Cold data identification using raw bit Error rate in Wear Leveling)를 제안한다. 제안하는 마모도 평준화 기법은 Hot/Cold 데이터를 구별하기 위해 HBSQ(Hot Block Sequence Queue)를 사용하고, 빈 블록 할당을 위해 최소 힙(Min Heap) 구조를 사용한다. 또한 마모도 평준화에서 발생되는 오버헤드를 최대한 줄이기 위해 블록의 에러 비트가 특정 임계값을 넘을 때까지는 마모도 평준화를 수행하지 않으며, 임계값을 넘은 블록이 할당을 위해 선택이 되면 마모도 평준화를 수행한다.

이하 논문의 구성은 다음과 같다. 2장에서는 낸드 플래시에서 발생할 수 있는 에러와 기존 연구에 대하여 기술한다. 3장에서는 본 논문에서 제안하는 마모도 평준화 기법을 소개하고, 4장에서는 실험을 통해 기존의 기법들과 비교하고, 5장에서는 결론 및 향후 연구과제에 대하여 기술한다.

II. Bakground and Related works

1. Background

낸드 플래시에서 발생할 수 있는 에러는 크게 4가지로 나눌 수 있다. 첫 번째 에러는 쓰기(Write Error)이다. 쓰기 에러는 데이터를 쓰는 과정에서 발생할 수 있는 에러로 문턱전압(Threshold Voltage)을 가지고 있는 셀에 전하를 추가로 주입하는 과 프로그래밍(Over Programming) 등으로 인하여 발생한다. 이러한 쓰기 에러는 P/E 횟수가 증가할수록 증가한다[13-16]. 두 번째 에러는 쓰기 방해(Write Disturb)이다. 쓰기 방해는 인접된 셀이 같은 워드라인에 포함되어있어 쓰기 연산 시 전압이 전가되는 현상을 말한다. 세 번째는 읽기 방해(Read Disturb)이다. 읽기 방해는 같은 셀을 반복적으로 읽어서 주변의 워드라인에 전압이 전가되어 문턱 전압 값이 정상적인 값보다 높아져서 발생한다. 마지막으로 보존에러(Retention Error)는 플로팅 게이트(Floating Gate)에 저장된 전하가 시간이 지남에 따라 빠져나와 데이터가 변경되어 발생된다. P/E 횟수가 늘어날수록 셀의 산화막은 점점 손상이 이루어지고 플로팅 게이트내의 전하가 더 많이 빠져나가게 된다.

2. Related works

지금까지 낸드 플래시 수명을 증가시키고, 오버헤드를 줄이기 위한 마모도 평준화 기법들이 많이 연구되어왔다. 대표적인 알고리즘으로는 GA(Greedy Algorithm)가 있다[2]. GA는 블록 내에 유효페이지가 가장 적은 블록을 가비지 컬렉션의 대상으로 하는 방법으로, 오버헤드가 적지만 특정 블록을 빠르게 소모하는 단점이 있다.

CB(Cost Benefit)는 GA에서 콜드 데이터가 희생블록으로 선정되지 않는 단점을 해결하기위해 제안되었으며, 식 (1)을 사용하여 가장 큰 값을 가지는 블록을 희생블록으로 선정하는 방법을 통하여 마모도 평준화를 수행한다[3].

$$age_i \times \frac{1-u}{2u} \quad (1)$$

식 (1)에서 u는 유효 페이지의 비율이며, age_i는 현재 시간에서 블록의 페이지 중에 마지막으로 무효화된 시간을 뺀 것으로 무효화 된 이후 경과 시간을 의미한다. $1-u$는 무효 페이지에서 오는 이득이며, $2u$는 유효페이지를 읽고 옮기는데 드는 비용을 의미한다.

CAT(Cost Age Time)는 CB가 콜드 데이터만 인식하는 단점을 개선하기 위해 제안되었으며, 식 (2)를 사용하여 가장 작은 값을 가지는 블록을 희생블록으로 선정한다[4].

$$\frac{u}{1-u} \times \frac{1}{age_a} \times EC \quad (2)$$

식 (2)에서 age_a는 사용을 위해 블록이 할당된 이후 지금까지 경과된 시간을 의미하며, EC(Erase Count)는 블록의 삭제 횟수를 의미한다. CAT는 가비지 컬랙션에서의 오버헤드를 줄이기 위해 Hot/Cold 데이터를 구별하여 저장하는 방법을 사용하고 있지만 여전히 콜드 데이터를 자주 이주시키는 단점이 있다.

그 외에도 BET(Block Erasing Table), Rejuvenator, SAW(System-Assisted Wear leveling)와 같은 마모도 평준화 기법들이 제안되었으며, 최근에는 실제 블록 수명을 반영하기 위해 P/E가 아닌 rBER을 마모도 평준화에 이용하는 ERA, Equalizer와 같은 연구가 이루어지고 있다[10-12].

본 논문에서는 Cold 데이터 판별 및 쓰기 에러 정보를 활용하여 마모도 평준화를 수행한다. ECC에서는 쓰기 에러, 읽기 방해 및 보존 에러 등을 따로 구별하지 못하기 때문에, ERA 기법과 동일하게 쓰기 연산 후 바로 읽는 방법을 사용하여 쓰기 에러를 검출한다. 이렇게 함으로써 시간이 지남에 따라 발생할

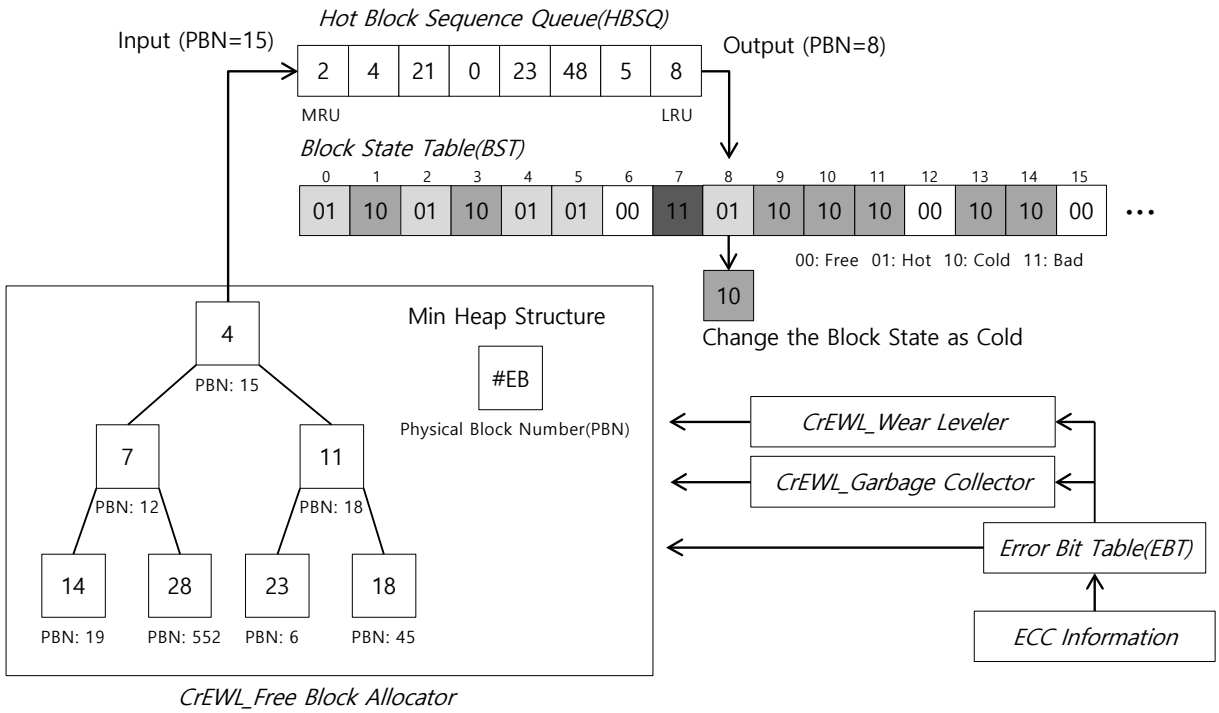


Fig. 1. The structure of CrEWL

수 있는 보존 에러 및 읽기 방해 에러 등의 다른 에러의 간섭을 최소화할 수 있다. 블록 내에 페이지마다 에러 발생 빈도가 다르기 때문에 블록의 수명은 불록 내 페이지들 중에 에러가 가장 많은 페이지를 기준으로 한다.

III. CrEWL Technique

1. 주요 구성

제안하는 기법은 에러 정보를 활용하여 마모도 평준화 및 가비지 컬랙션을 수행하고 있으며, 마모도 평준화 수행 시 발생될 수 있는 오버헤드를 줄이기 위해 블록의 에러 비트가 특정 임계 값에 도달하기 전까지는 마모도 평준화를 수행하지 않으며, 마모도 평준화를 수행할 때에는 Cold 데이터 판별을 활용한다. 그림 1은 제안하는 CrEWL를 보여주고 있다. CrEWL은 마모도 평준화를 위하여 각각의 블록에 사상되는 2개의 테이블을 사용하고 있다. 첫 번째 테이블은 EBT(Error Bit Table)로 ECC에서 읽어 들인 에러 검출 및 정정 정보가 저장된다. EBT에 저장된 정보는 빈 블록 할당, 마모도 평준화 및 가비지 컬랙션에 활용된다. 두 번째 테이블은 BST(Block State Table)로 블록 상태가 저장된다. 블록 상태는 Free(00), Hot(01), Cold(10), Bad(11) 4가지이다. 또한 CrEWL는 3가지 모듈로 이루어져 있다. 첫 번째 모듈은 FBA(Free Block Allocator)로 쓰기 연산 및 가비지 컬랙션에서 필요한 빈 블록을 할당하는 역할을 하며, 최소 힙 구조로 구성되어 있다.

알고리즘 1은 FBA 모듈의 의사코드를 보여주고 있다. 빈 블록 할당은 최소 힙 구조를 이용하여 에러 비트가 가장 적은 블록부터 순차적으로 이루어진다. 할당이 이루어진 블록은 Cold 블록과 구별하기 위해 BST에 Hot으로 설정되고 HBSQ(Hot Block Sequence Queue)에 블록의 물리적 주소를 저장한다. HBSQ는 일정한 크기의 버퍼공간이고, 원형 큐로 되어 있다. 블록 물리 주소가 순차적으로 삽입되고 버퍼에 빈 공간이 없을 경우에는 가장 먼저 삽입된 물리주소가 빠져나온다. 빠져나온 물리 주소의 블록은 BST에 Cold로 설정된다. 즉, 할당된 블록은 일정 시간이 흐른 뒤에는 Cold로 설정이 이루어지며, 여기서 시간은 HBSQ의 크기이다. 전체 가용블록들 중에 에러 비트가 임계값을 넘은 블록의 비율이 80%가 넘는 경우에는 마모도 평준화 및 가비지컬랙션에서 발생될 수 있는 오버헤드를 줄이기 위해 알고리즘1의 6번째 줄처럼 임계값을 증가시킨다. 임계값 설정은 다음의 식 (3)에 따라 결정된다. 식 (3)의 r은 라운드를 의미하고, EB_{MAX}는 배드 블록으로 판단되는 최대 에러 비트 수를 의미한다. 마모도 평준화 실행을 최대한 늦추기 위하여, r의 초기 값은 3으로 설정되며, 1씩 증가된다.

$$TH = EB_{MAX} \times \left(1 - \frac{1}{2^r}\right) \quad (3)$$

두 번째 모듈은 WL(Wear Leveler)이며, 알고리즘 2는 이 WL 모듈의 의사코드를 보여주고 있다. FBA를 수행할 때, 할당

Algorithm 1: CrEWL_FreeBlockAllocator

```
Input: BST, HBSQ, EBT
Output: blk
1:  blk ← GetMinHeap()
2:  if EBT[blk] > TH then
3:      blk ← CrEWL_WearLeveler(blk)
4:  end
5:  if the ratio of blocks which have more
        error bit then TH is 80% then
6:      increase TH
7:  end
8:  BST[blk] ← Hot
9:  outputBlk ← InputHBSQ(blk)
10: BST[outputBlk] ← Cold
11: return blk
```

될 블록의 에러 비트가 특정 임계값을 초과한 경우에는 마모도 평준화가 수행된다. 이 때 선택된 할당 블록은 P/E가 상당히 많이 이루어진 블록이므로 이 블록에 핫 데이터를 저장하게 되면 빠르게 배드 블록이 될 수 있다. 따라서 WL 모듈을 통해, 이 블록에는 콜드 데이터가 저장될 수 있도록 한다. 알고리즘 2의 1번째 줄에서 선택될 콜드 데이터는 BST에 블록의 상태가 Cold이면서 특정 임계값을 넘지 않는 블록 중에서, EBT의 에러 비트가 가장 적은 블록이 대상이 되며, 만약 선택 대상의 블록이 2개 이상 존재한다면 블록 내에 유효페이지가 가장 많은 블록이 선택된다. 블록이 선정되면, 해당 블록은 가지고 있는 데이터를 모두 FBA에 의해 선택되었던 기존의 블록에 옮기고 삭제 연산을 수행한다. 삭제된 블록은 FBA에 의해 할당 블록으로 선택된다.

세 번째 모듈은 GC(Garbage Collector)이며, 알고리즘 3은 CG 모듈의 의사코드를 보여주고 있다. GC 모듈은 GA기반으로 동작하지만, 기존의 방식은 희생블록으로 선정할 때 블록 내의 무효 페이지 수만 평가하여 효율이 낮다. 따라서 제안하는 알고리즘에서는 무효 페이지 수뿐만 아니라 HBSQ를 사용하여 구별할 수 있는 Hot/Cold의 정보를 활용하고 있다. HBSQ에 삽입된 블록은 Hot 데이터를 가지고 있을 확률이 높으므로, 같은 무효 페이지 수를 가지고 있는 경우에는 HBSQ에 삽입되지 않은 블록을 선택하는 방법이 오버헤드를 더 줄일 수 있다. 왜냐하면 Hot 블록의 페이지들은 Cold 블록의 페이지들에 비해 빠르게 무효화가 될 수 있기 때문이다. 따라서 알고리즘 3의 2번째 줄에서 선택될 희생블록은 식 (4)의 값이 가장 큰 값이 된다.

$$Cost_i = I_i + \frac{HBSQ_i}{HBSQ_{size}+1}$$
$$HBSQ_i = \begin{cases} position & \text{if } i \in HBSQ \\ HBSQ_{size} & otherwise \end{cases} \quad (4)$$

식 (4)에서 i는 블록의 물리 주소, I는 무효 페이지의 수이며, $HBSQ_{size}$는 설정된 $HBSQ$의 크기를 의미한다. $positon$은 $HBSQ$내에서의 위치를 의미하며, MRU에 위치하는 블록은 0, LRU에 위치하는 블록은 $HBSQ_{size}$ - 1의 값을 가진다. 따라서 식 (4)에 의해 $HBSQ$에 포함된 블록의 경우에는 같은 무효 페이지를 가지는 Cold 블록에 비해 항상 적은 값을 가지므로 Cold 블록보다 늦게 희생블록으로 선정된다. GC에 의해 선택되지 않는 블록들은 WL의 이주 대상 블록들이 된다.

Algorithm 2: CrEWL_WearLeveler

```
Input: blk, EBT, BST
Output: blk
1: cold_blk ← GetColdBlock(EBT, BST)
2: if exist cold_blk then
3:     migrate data from cold_blk to blk
4:     Erase(cold_blk)
5:     return cold_blk
6: end
7: return blk
```

Algorithm 3: CrEWL_GarbageCollector

```
1: for i = 1 to N do
2:     Cost_i = I_i + HBSQ_i / (HBSQ_size + 1)
3:     blk ← block i has the largest cost value
4: end
5: move valid pages in blk to active block
6: Erase(blk)
7: BST[blk] ← Free
8: Insert_Minheap(blk)
```

IV. Performance Evaluation

1. Experiment Setup

Table 1. Experimental environment

Chip	1 chip
Page Size	8192 + 640(Spare)byte
Block Size	2M + 160Kbyte, 256pages
Chip Size	1024 + 84(Extended)blocks
Random Read Time	60μs
Page Program Time	1500μs
Block Erase Time	5.0ms
P/E cycles	1000
ECC capability	256 bit
Free Block trigger	5%

본 논문에서는 CrEWL의 성능을 평가하기 위해 마이크로소프트사에서 개발한 SSD Extension for DiskSim 시뮬레이터를 사용하였다[17]. 실험에 적용한 ECC 용량은 256bit로 가정하였고, 256bit가 넘는 에러가 발생하게 되면 배드 블록으로 판단하였다. 실험에는 Hot/Cold 데이터의 비율에 따른 성능차이를 확인하기 위하여, Hot/Cold 비율이 그림 2와 같이 가우스 분포를 가지는 트레이스 파일을 사용하였다. 데이터는 평균 접근 횟수보다 많은 것을 Hot이라하고, 평균 접근 횟수보다 적은 것을 Cold라 한다. Cold 데이터가 적은 경우에는 마모도 평준화 성능이 잘 드러나지 않기 때문에 실험에서는 Cold 데이터 비율을 70% ~ 90%로 설정하였다. 전체 용량의 85%가 데이터로 채워져 있으며, 트레이스 파일에서 읽은 LPN(Logical Page Number)에 해당하는 PPN(Physical Page Number)을 업데이트하는 방법으로 실험을 진행하였다. 자세한 실험환경은 표 1과 같다. 마모도가 있는 블록의 경우 보존 에러 및 읽기 방해 에러가 발생할 수 있지만, 발생 빈도가 낮으므로 본 논문에서는 고려하지 않았다. 실험에서 제안하는 CrEWL의 HBSQ 크기는 32로 설정하였다.

제안한 CrEWL의 성능 비교를 위해 GA, CB, CAT, BET와 비교를 하였다. 실험에서는 각 기법들의 첫 번째 배드 블록이 발생하는 시점 및 마모도 평준화에서 발생되는 오버헤드를 측정하였다.

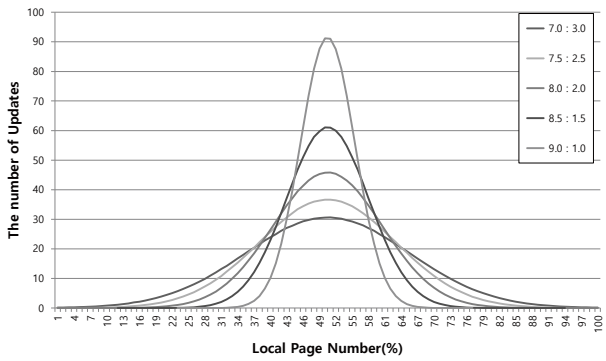
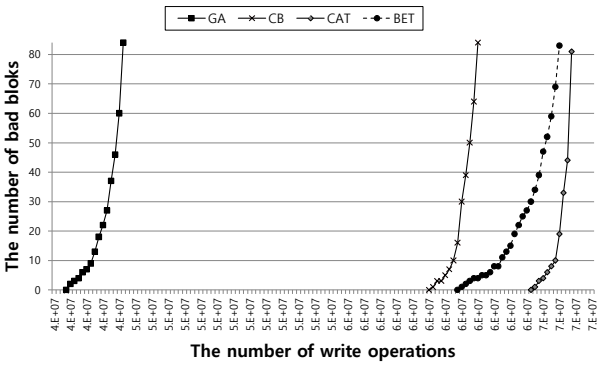


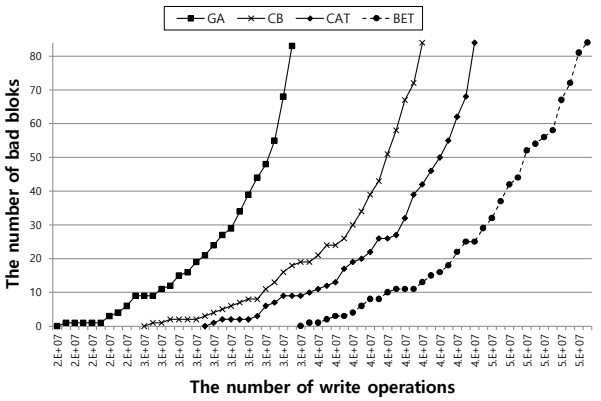
Fig. 2. Characteristics of Data set

2. Performance Evaluation

그림 3은 낸드 플래시 칩 내의 블록들의 수명이 일정할 때와 일정하지 않고 가우스 분포로 수명이 분포하고 있을 때의 (Cold 데이터 비율: 85%) 각 알고리즘의 성능을 비교한 그림이다. 대부분의 마모도 평준화 알고리즘들은 마모 지표로 삭제 횟수를 사용하고 있다. 하지만 낸드 플래시 블록들의 수명은 일정하지 않기 때문에 마모 지표로 삭제 횟수를 사용하는 것은 적절하지 않다. 수명이 일정할 경우에는 그림 3a에서와 같은 성능을 보였던 알고리즘들이 그림 3b와 같이 수명이 일정하지 않았을 때는 효율이 떨어지는 것을 확인할 수 있다. 즉, 실제 환경의 차이로 인하여 기존에 마모도 평준화에 중요한 지표로 사용되었던 블록 삭제 횟수는 실제로 적용하기에는 적절치 않은 것을 확인할 수 있다.



a. Uniform distribution



b. Gauss distribution(Variation = Average * 0.2)

Fig. 3. First failure time

그림 4는 쓰기 연산수에 따른 첫 번째 배드 블록이 발생하는 시점을 GA를 기준으로 평균 몇 배의 수명차이를 보이는지를 나타내고 있으며, 제안한 CrEWL을 기존의 알고리즘을 비교하고 있다. CrEWL은 평균적으로 GA와 비교하여 72%, CB와 비교하여 50%, CAT와 비교하여 41% 그리고 BET와 비교하여 27%정도 수명이 더 길다.

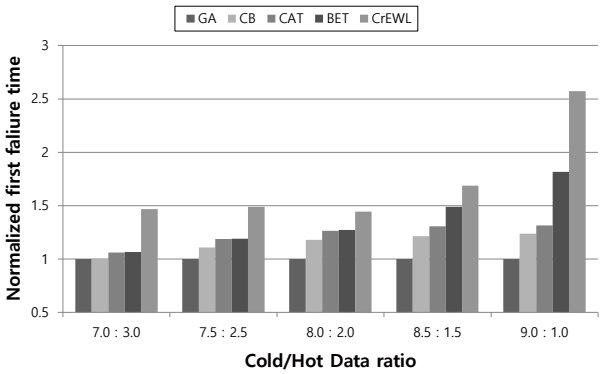


Fig. 4. Normalized first failure time

Table 2. Comparison of the average response time

	GA	CB	CAT	BET	CrEWL
7:3	316ms	317ms	311ms	318ms	317ms
8:2	306ms	299ms	293ms	308ms	294ms
9:1	252ms	235ms	228ms	264ms	240ms
Relative comparison of CrEWL	102.7%	100.0%	97.8%	104.5%	100%

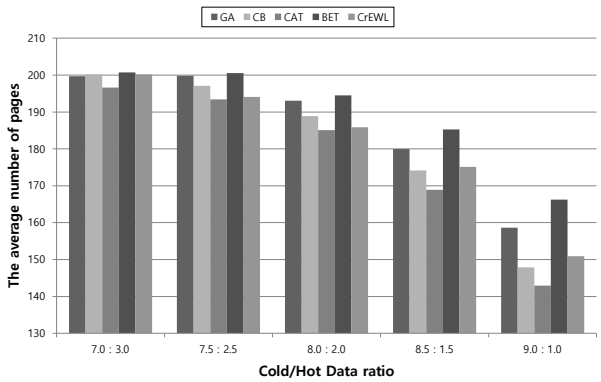


Fig. 5. The average number of page copies per erase operation

그림 5는 가비지 컬랙션 및 마모도 평준화에서 발생하는 삭제 연산 시 발생되는 유효 페이지의 평균 복사 횟수를 보여주고 있다. 평균적으로 유효 페이지의 복사 연산은 BET가 가장 빈번하게 일어났다. BET는 마모도 평준화를 위해 Cold 데이터의 존재로 인하여 삭제되지 않는 블록을 강제로 삭제하게 되는데, 이때 삭제 대상블록에 대한 평가가 이루어지지 않아 오버헤드가 많이 발생하는 것을 볼 수 있다. CrEWL은 유효 페이지의 복사에서 평균적으로 GA와 비교하여 3%, CB와 비교하여 1%, BET와 비교하여 5%정도 적게 발생한다. 반면 CAT와의 비교에서는 복사가 약 2%정도 더 발생한다.

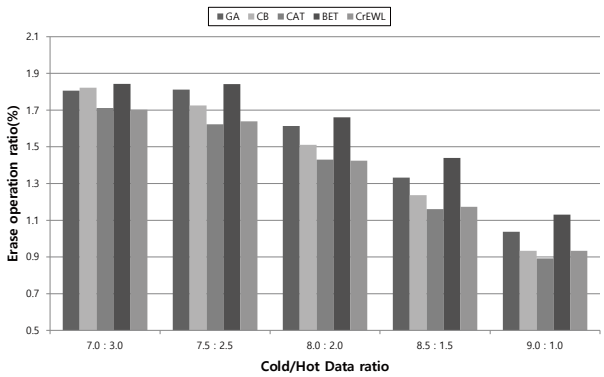


Fig. 6. Erase operation ratio per write operation

그림 6은 쓰기 연산 수에 따른 삭제 연산을 발생 빈도를 각 마모도 평준화 기법별로 나타낸 것이다. 마모도 평준화 알고리즘 및 가비지 컬랙션에서 발생되는 복사는 읽기 및 쓰기 연산만 발생되는 것이 아니라, 추가된 쓰기 연산에 의해 그림 6과 같이 더 많은 삭제 연산을 초래한다. CrEWL은 삭제 연산이 평균적으로 GA와 비교하여 10%, CB와 비교하여 5%, BET와 비교하여 15%정도 적게 발생한다. 반면 CAT와의 비교에서는 삭제 연산이 약 0.8%정도 더 발생한다.

가비지 컬랙션 및 마모도 평준화 수행 시 블록에서 발생되는 삭제 연산에서의 페이지 이주에 대한 평균 응답 시간은 아래의 식과 같다.

$$O = \sum_{n=1}^{v} (R_n + W_n) + E \quad (5)$$

식 (5)에서 v는 데이터 이주가 이루어지는 블록내의 유효 페이지 수를 의미한다. R_n, W_n 및 E은 각각 읽기 연산 속도, 쓰기 연산 속도 그리고 삭제 연산 속도를 의미한다. Hot/Cold 비율이 9:1인 데이터를 예로 들어 각각의 기법들의 평균 응답 시간은 GA는 252ms, CB는 235ms, CAT는 228ms, BET는 264ms, CrEWL은 240ms가 된다. Hot/Cold 비율에 따른 응답 속도는 표 2와 같다. 비록 CAT에 비해 약 2.2%정도 응답시간이 느리지만, 제안하는 CrEWL은 가장 중요한 지표인 수명 측면에서 CAT에 비해 약 41% 길다.

V. Conclusions

본 논문에서는 에러 검출 및 정정 정보를 이용한 마모도 평준화 기법을 제안하였다. 제안한 기법은 Cold 데이터 판단을 위한 HBSQ와 에러 발생 확률 정보를 이용하여 낸드 플래시 블록의 실제 수명을 예측하여 마모도 평준화를 수행하였다. 실험을 통하여 제안한 기법이 다른 마모도 평준화 기법들에 비하여 수명이 최대 72% 향상되었음을 확인하였으며, 오버헤드 또한 최대 41% 적은 것을 확인할 수 있었다.

다만 읽기 연산이 빈번한 데이터에 대한 고려가 이루어지지 않아 마모도가 높은 블록으로 옮겨진 Cold 데이터에 읽기 방해 에러 등이 빈도가 낮지만 축적될 가능성이 있다. 향후 읽기 방

해 및 보존 에러 등을 고려한 마모도 평준화 기법에 대한 추가 연구가 필요하다.

REFERENCE

[1] Chen, Feng, David A. Koufaty, and Xiaodong Zhang. "Understanding intrinsic characteristics and system implications of flash memory based solid state drives." ACM SIGMETRICS Performance Evaluation Review. Vol. 37. No. 1. pp. 181-192, June 2009.

[2] Wu, Michael, and Willy Zwaenepoel. "eNVy: a non-volatile, main memory storage system." ACM SigPlan Notices. Vol. 29. No. 11. pp. 86-97, Nov. 1994.

[3] Kawaguchi, Atsuo, Shingo Nishioka, and Hiroshi Motoda. "A Flash-Memory Based File System." USENIX, pp. 155-164, Jan. 1995.

[4] Chiang, M-L., and R-C. Chang. "Cleaning policies in mobile computers using flash memory." Journal of Systems and Software Vol. 48, No.3, pp. 213-231, Nov. 1999.

[5] Chang, Yuan-Hao, Jen-Wei Hsieh, and Tei-Wei Kuo. "Improving flash wear-leveling by proactively moving static data." Computers, IEEE Transactions on Vol 59, No. 1, pp. 53-65, Jan. 2010.

[6] Murugan, Muthukumar, and David HC Du. "Rejuvenator: A static wear leveling algorithm for NAND flash memory with minimized overhead." Mass Storage Systems and Technologies (MSST), 2011 IEEE 27th Symposium on. IEEE, pp. 1-12, May 2011.

[7] Chang, Li-Pin. "On efficient wear leveling for large-scale flash-memory storage systems." Proceedings of the 2007 ACM symposium on Applied computing, pp. 1126-1130, March 2007.

[8] Wang, Chundong, and Weng-Fai Wong, "Observational wear leveling: an efficient algorithm for flash memory management." Design Automation Conference (DAC), pp. 235-242, June 2012.

[9] Wang, Chundong, and Weng-Fai Wong. "SAW: System-assisted wear leveling on the write endurance of NAND flash devices." Design Automation Conference (DAC), pp. 1-9, May 2013.

[10] Pan, Yangyang, Guiqiang Dong, and Tong Zhang. "Error rate-based wear-leveling for NAND flash memory at highly scaled technology nodes." Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, Vol. 21, No. 7, pp. 1350-1354, July 2013.

[11] Yang, Ming-Chang, et al. "New ERA: new efficient reliability-aware wear leveling for endurance enhancement of flash storage devices." Design Automation Conference (DAC), pp. 163, May 2013.

[12] Woo, Yeong-Jae, and Jin-Soo Kim. "Diversifying wear index for MLC NAND flash memory to extend the lifetime of SSDs." Proceedings of the Eleventh ACM International Conference on Embedded Software. IEEE Press, pp. 6, Sep. 2013.

[13] Yang, Chengen, Yunus Emre, and Chaitali Chakrabarti. "Product code schemes for error correction in MLC NAND flash memories." Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, Vol. 20, No. 12, pp. 2302-2314, Dec. 2012.

[14] Mielke, Neal, et al. "Bit error rate in NAND flash memories." Reliability Physics Symposium, 2008. IRPS 2008. IEEE International, pp. 9-19, April 2008.

[15] Cai, Yu, et al. "Flash correct-and-refresh: Retention-aware error management for increased flash memory lifetime." Computer Design (ICCD), 2012 IEEE 30th International Conference on. IEEE, pp. 94-101, September 2012.

[16] Cai, Yu, et al. "Error patterns in MLC NAND flash memory: Measurement, characterization, and analysis." Design, Automation & Test in Europe Conference & Exhibition (DATE), 2012, pp. 521-526, March 2012.

[17] V. Prabhakaran and T. Wobber, “"SSD Extension for DiskSim Simulation Environment,” http://research.microsoft.com/en-us/downloads/b41019e2-1d2b-44d8-b512-ba35ab814cd4/, 2009.

Authors

Sang-Ho Hwang received the B.S. and M.S. degrees in Computer Engineering from Yeungnam University, Korea, in 2009 and 2013 respectively. His current research interests include embedded systems and non-volatile memory systems.

Jong Wook Kwak received a B.S. degree in Computer Engineering from Kyungpook National University, Taegu, Korea in 1998, a M.S. degree in Computer Engineering from Seoul National University, Seoul, Korea in 2001, and a Ph.D. degree in Electrical Engineering and Computer Science from Seoul National University, Seoul, Korea in 2006. From 2006 to 2007, he worked as a senior engineer in the SoC R&D Center, at Samsung Electronics Co., Ltd. He is currently an associate professor in the Department of Computer Engineering, Yeungnam University. His research interests include advanced processor architecture, low-power mobile embedded system, and high performance parallel computing.

Chang-Hyeon Park received the B.S. degree in Electronics Engineering from Kyungpook University, Korea, in 1986 and M.S. and Ph.D degrees in Computer Science from Seoul University, Korea, in 1988 and 1992, respectively. Dr. Park joined the faculty of the Department of Computer Engineering at Yeungnam University, Gyeongsan, Korea, in 1993. He is currently a Professor in the Department of Computer Engineering at Yeungnam University. He is interested in artificial intelligence, data mining, and embedded system.