논문 2017-12-21

메모리에서 정적 마모도 평준화를 위한 콜드 블록 추적 기법

(Tracking Cold Blocks for Static Wear Leveling in FTL-based NAND Flash Memory)

장 용 훈, 김 성 호, 황 상 호, 이 명 섭, 박 창 현*
(Yonghun Jang, Sungho Kim, Sang-Ho Hwang, Myungsub Lee, Chang-Hyeon Park)

Abstract : Due to the characteristics of low power, high durability and high density, NAND flash memory is being heavily used in various type of devices such as USB, SD card, smart phone and SSD. On the other hand, because of another characteristic of flash cell with the limited number of program/erase cycles, NAND flash memory has a short lifetime compared to other storage devices. To overcome the lifetime problem, many researches related to the wear leveling have been conducted. This paper presents a method called a TCB (Tracking Cold Blocks) using more reinforced constraint conditions when classifying cold blocks than previous works. TCB presented in this paper keeps a MCT (Migrated Cold block Table) to manage the enhanced classification process of cold blocks, with which unnecessary migrations of pages can be reduced much more. Through the experiments, we show that TCB reduces the overhead of wear leveling by about 30% and increases the lifetime up to about 60% compared to BET and BST.

Keywords : NAND flash memory, Wear leveling, Non-volatile memory, Lifetime, Reliability

Ⅰ. 서 론

낸드 플래시 메모리는 직렬 단위의 구성 방식으로 그룹화 된 페이지 (page) 혹은 블록 (block) 단위로 접근이 가능하며, 비교적 높은 직접도를 가진다. 이러한 장점으로 낸드 플래시 메모리는 이동식 장치인 USB, SD 카드, 스마트폰에 사용될 뿐만 아니라 보조 기억 장치인 SSD (Solid State Drive)에도 활용되는 등 다양한 분야에서 폭넓게 이용되고 있다 [1-2]. 이와 같은 장점에도 불구하고 낸드 플래시 메모리는 기존의 보조 기억 장치들에 비해 제한된 제자리 덮어쓰기, 짧은 수명 등의 문제점이 존재한다. 이러한 문제점을 해결하기 위해, 많은 연구자들이 낸드 플래시 메모리의 특성을 고려한 기법들을 제안하고 있으며, 메모리의 수명을 연장시키는 마모도 평준화 (wear leveling) 기법이 대표적인 예이다 [3-7].

마모도 평준화는 각 블록의 소거횟수, 콜드 (cold) 및 핫 (hot) 블록을 구분하는 정보, 각 페이지들의 참조 횟수, 참조 및 소거 경과 시간 등의 여러 정보를 테이블에 저장하고, 이를 활용하여 플래시 메모리의 수명을 연장한다. 하지만 이러한 정보들을 메모리에 저장하면 메모리의 사용량이 늘어나게 되어 저 사양 임베디드 시스템에 구현하기 힘들다는 단점이 있다. 그래서 비트의 배열로 구성되어 메모리의 사용량을 줄일 수 있는 BET (Block Erase Table)를 이용한 마모도 평준화 기법이 연구되었다 [8]. 또한 BET는 블록그룹의 크기 k가 커짐에 따라 여러 블록들을 하나의 그룹으로 묶어 각 비트에 대응시켜 사용할 수 있으며 k가 증가하면 메모리 사용량을 더욱 줄일 수 있다는 장점이 있다. 하지만 k가 증가할수록 마모도 평준화의 성능이 급격히 저하되는 단점이 있다.

이러한 BET 기법의 단점을 해결하기 위해 성능 저하의 원인을 숨겨진 콜드 블록 문제 (hidden cold block problem)로 지적하고, 이를 해결하기

*Corresponding Author (park@yu.ac.kr)
Received: May 15 2017, Revised: May 23 2017, Accepted: May 29 2017.
Y. Jang, S. Kim, S. Hwang, M. Lee, C. Park: Yeungnam University

ⓒ IEMEK J. Embed. Sys. Appl. 2017 June 12(3) 185-192
ISSN : 1975-5066
http://dx.doi.org/10.14372/IEMEK.2017.12.3.185

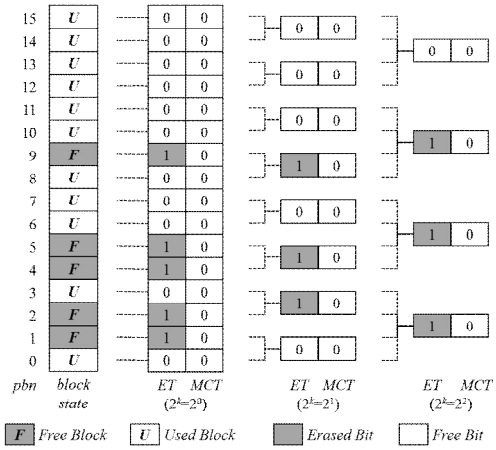

그림 1. *ET* 와 *MCT*의 구조
Fig. 1 Architecture of *ET* and *MCT*

위해 비트 설정 임계값 (BST: Bit Set Threshold)을 이용한 마모도 평준화 기법이 제안되었다 [9]. BST는 숨겨진 콜드 블록을 평균 무효 페이지수의 비율과 설정한 임계값의 비교로 인지하여 숨겨진 콜드 블록 문제를 해결하였다. 그러나 BST는 k값이 1이상인 경우에만 효율이 있다.

위와 같은 문제점들을 해결하기 위해 본 논문에서는 TCB (Tracking Cold Blocks)를 제안한다. 제안하는 기법은 콜드 블록의 인지수준을 강화하고 마모도 평준화로 인한 페이지들의 이주를 최소화함으로써 오버헤드를 감소시키고 수명을 증가시켜 낸드 플래시 메모리의 전반적인 성능향상에 기여한다.

이하 본 논문의 구성은 다음과 같다. Ⅱ장에서는 마모도 평준화 관련 연구를 소개하고, Ⅲ장에서는 본 논문에서 제안하는 TCB을 서술한다. Ⅳ장에서는 기존 기법들과 비교한 TCB의 성능을 평가하고, 마지막 Ⅴ장에서 결론을 맺는다.

Ⅱ. 관련 연구

1. 마모도 평준화 기법

플래시 메모리의 수명을 연장하기 위해 여러 마모도 평준화 기법들이 연구되었다. DP (Dual-Pool)는 콜드 블록과 핫 블록에 따라 2개의 그룹으로 나누고, 콜드 그룹의 가장 낮은 소거횟수를 가진 블록과 핫 그룹의 가장 높은 소거횟수를 가진 블록을 서로 비교하여 임계값 이상이면 두 블록을 서로 교환한다 [7]. Lazy-WL (Lazy Wear Leveling)은 블록의 소거횟수가 임계값 이상이면 삭제를 하지 않고 추후에 콜드 데이터를 해당 블록으로 옮긴다 [10]. Rejuvenator는 각 페이지 별로 참조 횟수를 저장하여 콜드 페이지와 핫 페이지로 나누고, 각각 소거 횟수가 많은 올드 블록 (old block)과 소거 횟수가 적은 영 블록 (young block)으로 이동시킨다 [11]. OWL (Observational Wear Leveling)은 최근 참조된 블록들을 리스트로 저장하고 주기적으로 감시하여 자주 참조된 블록의 데이터를 가용 블록 중 가장 소거횟수가 적은 블록에 저장한다 [12]. FeGC (Fast and endurant Garbage Collection)는 각 블록의 지워진 시간을 기록하여, 경과된 시간에 따라 각 블록의 비용을 계산하고 이를 이용해 마모도 평준화를 수행한다 [13].

위의 기법들은 블록의 지우기 횟수 제한이 동일하다는 점을 기준으로 하고 있다. 그러나 ERA (Efficient Reliability Aware) 기법은 블록들이 모두 동일한 수명을 갖는 것이 아니라 BER (Bit Error Rate)에 따라 실제 수명이 다르다는 점에 기안하여 BER를 기반으로 한 마모도 평준화를 수행한다 [14].

이상의 기법들은 마모도 평준화를 수행하기 위해 블록 지우기 횟수 테이블, 참조 횟수 테이블, BER 테이블 등의 여러 테이블들을 사용한다. 따라서 테이블을 관리하기 위한 일정 이상의 메모리 사용량이 요구되기 때문에 저 사양 임베디드 장치에서 사용하기에 비용이 많이 든다는 단점이 있다.

2. BET

BET (Block Erase Table)는 저 사양 임베디드 시스템에 사용될 수 있도록 제안되었다 [8]. BET는 블록 그룹의 소거 이력을 비트 단위로 구성된 테이블에 저장하는 방법으로 메모리의 사용량을 크게 줄였다. k값을 임의로 지정하여 2^k개의 블록을 한 그룹으로, 테이블의 각 비트에 사상시킨다. 일정 기간 t동안 블록의 각 그룹에서 소거가 발생하면 사상된 비트에 1을 설정한다. t기간이 끝나면 값이 0인 비트에 대응되는 블록 그룹들을 콜드 블록으로 판단하고 이를 강제 이주시켜 마모도 평준화를 수행한다. 또한 k값에 따라 2^k개의 블록을 하나의 비트에 대응시켜 테이블에 소요되는 메모리를 줄일 수 있다. 하지만 k값이 커짐에 따라 여러 개의 블록을 하나의 비트에 대응시키면 그 그룹 내에 존재하는 블록들 중에서 콜드 블록을 구별하기 힘들어진다는 단점이 있다. 결국 누락되는 콜드 블록의 정보로 인하여 마모도 평준화의 성능은 급격히 떨어진다.

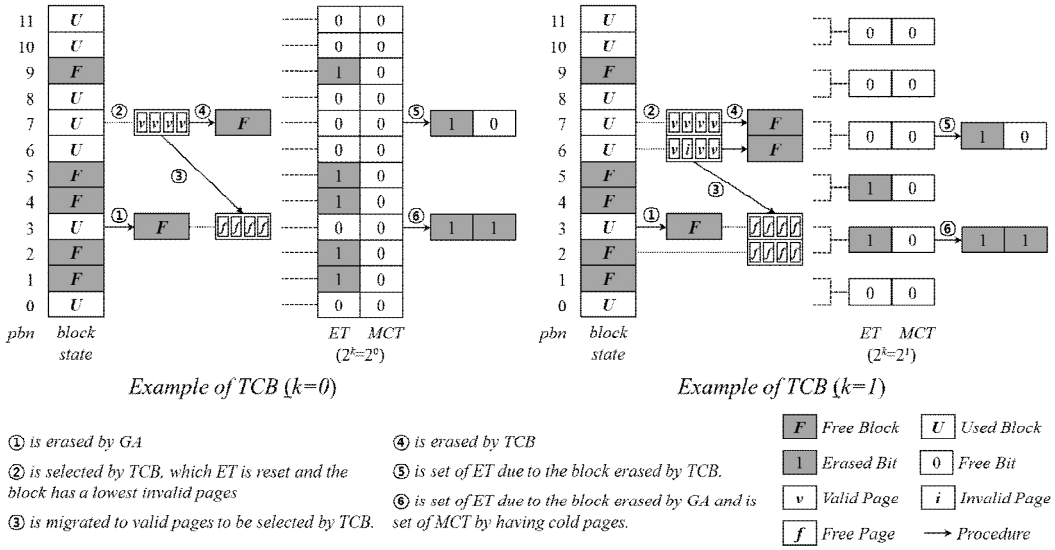

그림 2. TCB의 수행 절차
Fig. 2 Procedures of TCB

3. BST

BET에서 k값 증가에 따른 문제점을 해결하기 위한 기법으로 BST (Bet Set Threshold)이 제안되었다 [9]. BST는 k값이 1이상일 때 마모도 평준화의 효율이 떨어지는 문제점을 확인하고, 이를 숨겨진 콜드 블록 문제 (hidden cold block problem)라 하였다. BST는 숨겨진 콜드 블록 문제를 해결하기 위해 소거가 발생한 블록 그룹에 사상된 비트를 1로 설정 할 것인지에 대한 여부를, 해당 블록 그룹의 평균 무효 페이지 수가 전체 블록 그룹의 평균 무효 페이지 수 이상이면 BET의 비트를 1로 설정함으로써 k값 증가로 인한 숨은 콜드 블록 문제를 해결 하였다. 그러나 BST는 k값이 1이상인 경우에만 효율이 나타나며 BET에 종속적인 성능 개선이라는 한계가 있다.

Ⅲ. 제안 기법

이 장에서는 BET와 BST의 단점을 개선하기 위한 기법인 TCB (Tracking Cold Blocks)를 제안한다. TCB를 설명하기에 앞서, 블록 그룹의 소거 이력을 관리하는 테이블 BET의 명칭을 4장의 성능 평가 때 비교 기법으로 사용될 BET와의 혼동을 방지하기 위해 ET (Erase Table)라는 용어로 바꾸어 설명한다.

input: e_{cnt}, f_{cnt}, tcb_{cnt}, k, f_{index}, ga_{victim}, tcb_{vimtim}, ET, MCT, and T
output: *null*

```
1  ga_victim ← GetVictimBlockByGreedy();
2  if e_cnt/f_cnt ≥ T do
3    if f_cnt ≥ size(ET) then
4      e_cnt ← 0;
5      f_cnt ← 0;
6      ResetAllFalgsInET();
7      return;
8    end
9    if tcb_cnt ≥ size(MCT) then
10     tcb_cnt ← 0;
11     ResetAllFalgsInTCB();
12     return;
13   end
14   for f_index = 0 to size(ET) do
15     if ET[f_index] = 0 and MCT[f_index] = 0 then
         // select tcb_vimtim, by equation (1)
16     end
17   end
18   for ga_victim mod k to (ga_victim mod k) + k do
19     if IsErasedBlock(ga_victim) = false then
20       EraseBlock(ga_victim);
21     end
       //migrate pages in tcb_victim to ga_victim
22     MigrateTo(ga_victim, tcb_victim);
23     EraseBlock(tcb_victim);
24     MCT[idx(ga_vimtim)] ← 1;
25   end
26   tcb_cnt ← tcb_cnt + 1;
27 end
```

알고리즘 1. TCB-정적마모도 평준화 수행절차
Algorithm 1. TCB-StaticWearLevelingProcedure

TCB에서는 기존의 ET와 함께 콜드 블록으로 판명되어 이주된 블록들을 기억하는 테이블이 추가로 사용되며. 이를 MCT (Migrated Cold block Table)라 명명한다. TCB는 ET와 MCT의 비트가 모두 0이어야만 블록이 마모도 평준화의 대상이 되도록 함으로써 페이지의 무분별한 이주를 방지하고 적절치 않은 블록이 가비지 컬렉션 [15]의 대상에서 제외 되도록 하여 낸드 플래시 메모리의 성능 향상을 유도한다.

그림 1은 TCB에서 사용되는 테이블의 구조를 보인다. 그림 1에서 2^k개의 블록으로 묶여있는 블록 그룹은 ET와 MCT의 비트에 각각 사상된다. 알고리즘 1은 ET와 MCT를 활용한 TCB의 자세한 수행 절차를 기술하고 있으며, 그림 2는 k값이 각각 0과 1일 때, TCB의 마모도 평준화 과정을 순차적으로 보인다.

알고리즘 1에서 ga_{victim} 가비지 컬렉션이 수행된 후 (그림 2. ①), 전체 블록의 소거 횟수 e_{cnt}와 소거된 블록 그룹의 수 f_{cnt}를 나눈 값이 임계값 T 이상이면 마모도 평준화가 수행된다. 만약 ET 또는 MCT에 사상된 모든 블록 그룹이 소거된 상태라면 e_{cnt}, f_{cnt}, ET, MCT를 모두 0으로 초기화 한다 (알고리즘 1. 3-13).

$$\cos t = \sum_{i=0}^{2^k} 1 - v_i \qquad (1)$$

위의 상황에 해당하지 않는다면 즉, 마모도 평준화를 수행해도 될 상황이라면 같은 색인의 ET와 MCT의 비트가 모두 0인 블록 그룹들 중에서 식 (1)을 통해 이주 될 블록 그룹 tdb_{victim}을 선정한다 (알고리즘 1. 14-17, 그림 2. ②). 식 (1)에서 v_i는 블록 그룹 내 i번째 블록의 유효 페이지 비율을 의미하며, $\cos t$가 가장 낮은 블록 그룹을 tdb_{victim}으로 선정한다. tdb_{victim}가 선정되면 tdb_{victim}의 데이터들이 가장 최근에 가비지 컬렉션이 수행 된 블록 그룹으로 이주된다 (알고리즘 1. 18 - 25, 그림 3. ③). 이때 tdb_{victim}는 소거되고 사상된 ET의 비트에 1이 설정된다 (알고리즘 1. 20, 그림 3. ④, ⑤). 동시에 콜드 블록이라 판단되어 이주된 블록 그룹을 표시하기 위해 ga_{victim}에 사상된 MCT의 비트를 1을 설정하고 (그림 3. ⑥) 이주된 콜드 블록 그룹 수 tdb_{cnt}를 증가시킨다 (알고리즘 1. 26).

알고리즘 2는 소거에 의한 ET의 업데이트 과정을 기술한다. 알고리즘 2에서 ET의 업데이트는 가비지 컬렉션이나 마모도 평준화로 인해 소거가 발생되었을 때 수행된다. 블록 그룹 내의 블록에 소거가 발생하면 블록 그룹에 사상된 ET의 비트 색인을 계산하고 해당 ET의 비트를 1로 설정한다. 그리고 전체 블록의 소거횟수 e_{cnt}와 소거된 블록의 수 f_{cnt}를 증가시킨다 (알고리즘 2. 1-5).

```
input: e_cnt, f_cnt, k, b_index, and ET
output: e_cnt, f_cnt
/* ET are updated based on b_index and k in the ET
   mapping when the block is erased.          */
1 e_cnt ← e_cnt + 1; // Increase the total erase count.
  // Update the ET if needed
2 if ET[b_index/2^k] = 0 do
3 |  ET[b_index/2^k] ← 1;
4 |  f_cnt ← f_cnt + 1;
5 end
```

알고리즘 2. TCB-ET갱신
Algorithm 2. TCB-ETUpdate

TCB에서는 가비지 컬렉션이 소거횟수가 많은 블록을 수행 대상으로 선택하는 기법인 GA [15]를 가장 일반적으로 사용한다는 점에 착안하여, 가비지 컬렉션이 수행된 블록들을 핫 블록이라 판단하였고 또한 ET와 MCT의 비트가 모두 0인 블록 즉, 소거가 발생하지 않았고 유효 페이지 비율이 높은 블록 그룹을 콜드 블록이라 판단하였다.

결과적으로 TCB는 가비지 컬렉션으로 핫 블록을 인지하고 ET 그리고 MCT를 이용하여 콜드 블록의 인지도를 강화하였으며 또한 이들의 무분별한 페이지 이주를 방지하고 핫 블록의 빈번한 가비지 컬렉션 참여를 최대한 제외하였다.

Ⅳ. 성능 평가

1. 실험 환경설정

TCB의 성능 평가를 위해 DiskSim 4.0 기반의 SSD (Extension for DiskSim)를 사용하여 모의실험을 진행하였고, 워크로드로 SPC와 FIU를 사용했다 [16-18]. SPC는 OLTP (OnLine Transaction Processing)의 특성을 가지고 있으며, FIU는 데이터 센터의 호스트로부터 터미널까지의 서비스 동작을 포함하고 있다. 이러한 특성을 가지는 워크로드의 자세한 정보는 표 1과 표 2에 보이며, 실험 환경 설정에 사용된 매개변수는 표 3에 보인다.

표 1. SPC 워크로드의 종류와 특성
Table 1. types and characteristics of SPC workload

types \ characteristics		ratio of request (%)		avg. request (byte)
		read	write	
SPC	financial1	23.16	76.84	4.8
	financial2	82.28	17.72	4.3

표 2. FIU 워크로드의 종류와 특성
Table 2. types and characteristics of FIU workload

types \ characteristics		ratio of request (%)		avg. request (byte)
		read	write	
FIU	home1	1.12	98.88	7.2
	home2	12.40	87.60	13.1
	home3	3.94	96.06	51.5
	home4	0.03	99.97	16.6
	online	35.55	64.45	17.6
	webmail	26.03	73.97	18.0
	webresearch	0.14	99.86	22.4
	webusers	29.24	70.76	17.0

표 3. 모의실험의 매개변수
Table 3. Simulation parameters

Description	Value
Total capacity	2GB
Reserved free blocks	15%
Garbage Collection Trigger	# of free blocks < 5%
Flash chip elements	1
Planes per elements	1
Blocks per plane	4096
Pages per block	128
Page size	4KB
Page read latency	60us
Page program latency	800us
Block erase latency	1.5ms
P/E cylce	10^3

2. 실험 결과 및 분석

TCB의 성능을 워크로드의 쓰기 연산을 1억 회 수행했을 때, 유효 페이지 이주에 대한 오버헤드와 최초로 한계 수명에 도달한 블록이 발생했을 시점의 시간과 블록 간의 편차를 측정하여 GA, DP, BET 그리고 BST와 비교 분석하였다.

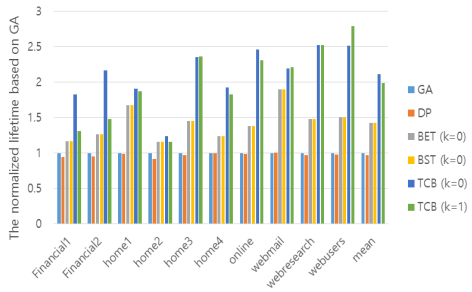

그림 3. 낸드 플래시 메모리의 수명
Fig. 3 The lifetime of NAND flash memory

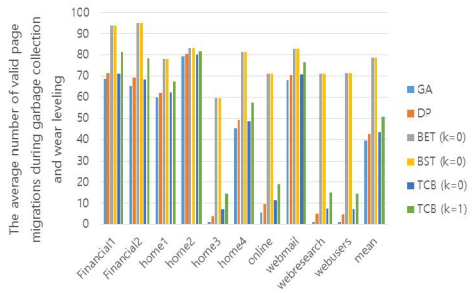

그림 4. 유효 페이지의 평균 이주 횟수
Fig. 4 The average of valid pages migrations during garbage collection and wear leveling

그림 3은 각 워크로드들을 GA로 수행한 낸드 플래시 메모리의 수명을 1이라 했을 때, 동일한 작업을 수행한 각 기법들의 수명을 나타낸 것이다. 그림 3의 실험 결과, TCB는 GA, DP, BET, BST와 비교하여 각각 평균적으로 약 111%, 114%, 69%, 69%만큼 수명을 증가시켰다. 이러한 결과는 TCB가 가비지 컬렉션에 의해 선정되는 핫 블록을 인지하고 마모도 평준화의 대상으로 선정될 빈도를 감소시킬 수 있었기 때문이다. 그러나 핫 블록의 가비지 컬렉션 최소화는 유효 페이지 이주를 이용한 방법이며, 이러한 방법은 반드시 오버헤드가 존재하기 때문에 이주 횟수에 대한 오버헤드를 측정하여 기존 기법들과 비교할 필요가 있다.

그림 4는 각 워크로드들을 1억 회 수행하는 동안 유효 페이지의 평균 이주 횟수를 각 기법별로 나타낸 것이다. 그림 4의 실험결과, TCB의 유효 페이지 이주 횟수가 GA, DP와 비교하여 평균 약 2.4회 더 많은 이주 횟수가 측정되었고, BET, BST와 비교해 평균 약 35.3회 더 적은 수치가 측정되었다. 이러한 결과는 TCB의 페이지 이주 정책 오버헤드

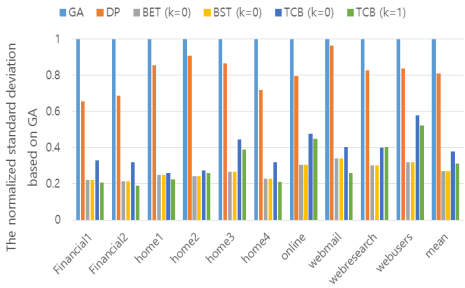

그림 5. 마모도 편차
Fig. 5 The normalized standard deviation

가 마모도 평준화를 수행하지 않는 GA와 비교했을 때와 거의 차이가 없으며, BET와 BST의 오버헤드와 비교해서도 약 30%의 오버헤드를 감소시켰다.

그림 5는 각 워크로드들을 GA로 수행한 낸드 플래시 메모리의 블록 간 마모도 편차를 1이라 했을 때, 동일한 작업을 수행한 각 기법들의 블록 간 마모도 편차를 나타낸 것이다. 그림 5에서 TCB는 GA, DP와 비교해 블록 간 마모도 편차를 평균적으로 약 50%이상 줄였고 BET와 BST와 비교해 약 12% 증가하였다. 그러나 낸드 플래시 메모리의 수명 향상과 유효페이지 이주로 인한 오버헤드의 최소화를 고려했을 때 이는 충분히 감수할 수 있는 수치이다. 그리고 모든 실험에서 TCB의 k값이 증가되었을 때 성능이 저하되는 현상을 확인할 수 있다. 이는 BET를 기반으로 하는 기법들은 모두 k값 증가로 인한 부정적인 영향을 받을 수 있는 부분이나, TCB는 k값이 1일 때에도 k값이 0인 BET, BST보다 우수한 성능을 보였다. 이로써 가비지 컬렉션의 효율을 높였고 페이지 이주 정책으로 인한 오버헤드를 감소시켰으며 향상된 마모도 평준화로 수명을 증가시켜 낸드 플래시 메모리의 전반적인 성능 향상을 기대할 수 있다.

V. 결 론

본 논문은 낸드 플래시 메모리에 적용된 기존 기법들의 문제점을 개선하기 위해 TCB를 제안했다. TCB는 기존의 BET에 MCT를 추가하여 콜드 블록의 추적조건을 강화하는 것으로 마모도 평준화의 불필요한 페이지 이주를 감소시켰다. 그리고 콜드 블록을 가비지 컬렉션이 수행된 블록 즉, 핫 블록으로 이주시키는 것으로 콜드 블록의 활용 빈도를 높였고 또한 핫 블록의 가비지 컬렉션 최소화를 유도하였다.

TCB의 성능평가 결과, 기존의 기법들과 비교해 낸드 플래시 메모리의 수명을 69%에서 114%까지 증가 시켰으며, 페이지 이주 정책에 대한 오버헤드도 BET, BST에 비해 약 70% 수준으로 줄였다. 마모도 편차는 BET와 BST에 비해 약 12% 높았지만 이는 감수 할 수 있는 수치이며 수명 향상과 페이지 이주에 대한 오버헤드를 줄였다는 점을 고려하면 낸드 플래시 메모리의 전반적인 성능 향상이 이루어졌음을 확인할 수 있었다. 실험결과에서 개선의 여지가 있었던 마모도 편차에 대한 보완은 향후 연구과제로 남긴다.

References

[1] L. Wilson, “International Technology Roadmap for Semiconductors (ITRS),” Semiconductor Industry Association, 2013.

[2] K. Suh, B. Suh, Y. Lim, J. Kim, Y. Choi, Y. Koh, S. Lee, S. Kwon, B. Choi, J. Yum, J. Choi, J. Kim, H. Lim, “3.3 V 32 Mb NAND Flash Memory With Incremental Step Pulse Programming Scheme,” IEEE Journal of Solid-State Circuits, Vol. 30, No. 11, pp. 1149-1156, 1995.

[3] D. BWoodhouse, “JFFS: The Journalling Flash File System,” Ottawa linux symposium, Vol. 2001, 2001.

[4] B. Chang, Z. Wang, B. Chen, F. Zhang, “Mobipluto: File System Friendly Deniable Storage for Mobile Devices,” Proceedings of the ACM 31st Annual Computer Security Applications Conference, pp. 381-390, 2015.

[5] D. Agrawal, D. Ganesan, R. Sitaraman, Y. Diao, S. Singh, “Lazy-adaptive Tree: An Optimized Index Structure for Flash Devices,” Proceedings of the VLDB Endowment, Vol. 2, No. 1, pp. 361-372, 2009.

[6] M. Athanassoulis, A. Ailamaki, “BF-tree: Approximate Tree Indexing,” Proceedings of the VLDB Endowment, Vol. 7, No. 14, pp. 1881-1892, 2014.

[7] L. P. Chang, “On Efficient Wear Leveling for Large-scale Flash-memory Storage Systems,” Proceedings of the ACM symposium on

Applied computing, pp. 1126-1130, 2007.
[8] Y. Chang, J. Hsieh, T. Kuo, "Improving Flash Wear-leveling by Proactively Moving Static Data," IEEE Transactions on Computers, Vol. 59, No. 1, pp. 53-65, 2010.
[9] S.H. Kim, J.H. Choi, J.W. Kwak, "BST: Hidden Cold Block-Aware Wear Leveling Using Bit-Set Threshold for NAND Flash Memory," IEICE Transactions on Information and Systems, Vol. 99, No. 4, pp. 1242-1245, 2016.
[10] L. Chang, L. Huang, "A Low-cost Wear-leveling Algorithm for Block-mapping Solid-state Disks," Proceedings of the 2011 SIGPLAN/SIGBED conference on Languages, compilers and tools for embedded systems, Vol. 46, No. 5, pp. 31-40, 2011.
[11] M. Murugan, D. H. Du, "Rejuvenator: A Static Wear Leveling Algorithm for NAND Flash Memory With Minimized Overhead," Proceedings of the IEEE 27th Symposium on Mass Storage Systems and Technologies, pp. 1-12, 2011.
[12] C. Wang, W. Wong, "Observational Wear Leveling: an Efficient Algorithm for Flash Memory Management," Proceedings of IEEE Design Automation Conference, pp. 235-242, 2012.
[13] O. Kwon, K. Koh, J. Lee, H. Bahn, "FeGC: An Efficient Garbage Collection Scheme for Flash Memory Based Storage Systems," Journal of Systems and Software, Vol. 84, No. 9, pp. 1507-1523, 2011.
[14] M. Yang, Y. Chang, C. Tsao, P. Huang, "New ERA: New Efficient Reliability-aware Wear Leveling for Endurance Enhancement of Flash Storage Devices," Proceedings of the ACM 50th Annual Design Automation Conference, pp. 163, 2013.
[15] M. Wu, W. Zwaenepoel, "eNVy: a Non-volatile, Main Memory Storage System," Proceedings of the ACM sixth international conference on Architectural support for programming languages and operating systems, Vol. 28, No. 5, pp. 86-97, 1994.
[16] J.S. Bucy, J. Schindler, S. W. Schlosser, G. R. Ganger, "The Disksim Simulation Environment Version 4.0 Reference Manual (cmu-pdl-08-101)," Parallel Data Laboratory, 2008.
[17] V. Prabhakaran, T. Wobber, "SSD Extension for DiskSim Simulation Environment," Microsoft Reseach, 2009.
[18] Trace, Exchange, "SNIA IOTTA Repository," 2010.

Yonghun Jang (장 용 훈)

He received a B.S. degree in Department of Computer Engineering from Yeungnam University College of Science and Technology, Daegu, Korea in 2012. He is currently a Ph.D. candidate in Department of Computer Engineering from Yeungnam University. He is interested in flash memory, image processing and bigdata analysis.
Email: killerwise@ynu.ac.kr

Sungho Kim (김 성 호)

He received a B.S. degree in Department of Computer Engineering from Yeungnam University College of Science and Technology, Daegu, Korea in 2012. He is currently a Ph.D. candidate in Department of Computer Engineering from Yeungnam University. His current research interests include embedded systems and non-volatile memory systems.
Email: boca123@ynu.ac.kr

Sang-Ho Hwang (황 상 호)

He 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.
Email: snailcom@ynu.ac.kr

Myungsub Lee (이 명 섭)

He received his Ph.D. degree in Computer Science from Yeungnam University, Gyungsan, Korea, in 2003. He also worked as a full-time instructor from 2002 to 2003 at Gyungdong College, Gyungsan, Korea. He was a Research Professor at Yeungnam University from 2004 to 2007. Subsequently, he was a Senior Researcher with the Regional Innovation Center at Yeungnam University. Since 2008, he has been Professor at the Yeungnam University College of Science and Technology. His research interests include wireless and sensor networks, location-based systems.
Email: skydream@ync.ac.kr

Chang-Hyeon Park (박 창 현)

He 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.
Email: park@yu.ac.kr