Journal of The Korea Society of Computer and Information
Vol. 21 No. 5, pp. 1-10, May 2016
www.ksci.re.kr
http://dx.doi.org/10.9708/jksci.2016.21.5.001

EPET-WL: Enhanced Prediction and Elapsed Time-based Wear Leveling Technique for NAND Flash Memory in Portable Devices

Sung Ho Kim*, Jong Wook Kwak**

Abstract

Magnetic disks have been used for decades in auxiliary storage devices of computer systems. In recent years, the use of NAND flash memory, which is called SSD, is increased as auxiliary storage devices. However, NAND flash memory, unlike traditional magnetic disks, necessarily performs the erase operation before the write operation in order to overwrite data and this leads to degrade the system lifetime and performance of overall NAND flash memory system. Moreover, NAND flash memory has the lower endurance, compared to traditional magnetic disks. To overcome this problem, this paper proposes EPET (Enhanced Prediction and Elapsed Time) wear leveling technique, which is especially efficient to portable devices. EPET wear leveling uses the advantage of PET (Prediction of Elapsed Time) wear leveling and solves long-term system failure time problem. Moreover, EPET wear leveling further improves space efficiency. In our experiments, EPET wear leveling prolonged the first bad time up to 328.9% and prolonged the system lifetime up to 305.9%, compared to other techniques.

▸Keyword: NAND flash memory, wear leveling, elapsed time, free block allocation policy, portable device

I. Introduction

하드 디스크(hard disk)로 대표되는 자기 저장 장치는 보조 기억장치로 수십 년 동안 사용 되어 왔다. 최근에는, 플래시 메모리 기반의 SSD(Solid State Drive)의 사용이 증가하고 있다. 플래시 메모리는 비 휘발성 메모리로 낮은 전력, 빠른 동작 속도, 높은 내진성의 특성을 가지고 있어, 태블릿, 스마트폰, 임베디드 시스템과 같은 다양한 분야에서 널리 사용되고 있다. 플래시 메모리는 셀(cell)의 구성단위에 따라 바이트 단위인 노어 플래시 메모리(NOR flash memory)와 페이지 단위인 낸드 플래시 메모리(NAND flash memory)로 구분할 수 있으며, 상대적으로 낸드 플래시 메모리는 가격이 저렴하고 대용량화가 쉽기 때문에 많이 사용되고 있다.

낸드 플래시 메모리는 기존의 하드 디스크와 달리 크게 세 가지 다른 특성을 가지고 있다. 첫 번째, 읽기와 쓰기연산은 페이지(page) 단위로 수행되며 소거연산은 블록(block) 단위로 수행된다. 낸드 플래시 메모리에서의 최소 연산 단위는 페이지이며, 다수의 페이지 집합이 블록이다. 이는 낸드 플래시 메모리의 셀 구성단위 특성 때문이다. 따라서 블록 단위인 소거연산은 페이지 단위인 쓰기와 읽기연산에 비해 최소 3배에서 최대 28배까지 느리게 수행된다.

두 번째, 낸드 플래시 메모리는 기존의 하드 디스크와 달리 제자리 덮어쓰기를 수행하기 위해서 반드시 쓰기 전 소거(erase before write) 연산이 선행되어야 한다. 이는 낸드 플래시 메모리의 구조적 특성으로 인하여 페이지의 제자리 덮어쓰기 연산이 불가능하기 때문이다. 또한 이러한 쓰기 전 소거연산

• First Author: Sung Ho Kim, Corresponding Author: Jong Wook Kwak
*Sung Ho Kim(boca123@ynu.ac.kr), Dept. of Computer Engineering, Yeungnam University
**Jong Wook Kwak(kwak@yu.ac.kr), Dept. of Computer Engineering, Yeungnam University
• Received: 2016. 03. 11, Revised: 2016. 03. 28, Accepted: 2016. 04. 26.
• 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).

은 전반적인 시스템의 성능을 저하시키는 원인이 된다.

세 번째, 낸드 플래시 메모리는 하드 디스크와 비교하여 상대적으로 낮은 내구성을 가지고 있다. 이와 같은 문제를 해결하기 위해서 셀이 가지는 비트를 나누어 낮은 내구성 문제를 해결하는 방식들이 연구되었다. 낸드 플래시 메모리는 셀이 가지는 비트에 따라 다음과 같이 구분되며 각각의 내구성은 한 비트를 가지는 SLC(Single-Level Cell)의 경우 약 100,000번, 두 비트를 가지는 MLC(Multi-Level Cell)는 약 10,000번의 내구성을 가진다. 표 1은 SLC와 MLC의 상세를 보여준다[1-2]. 낸드 플래시 메모리는 기존의 하드 디스크와 비교하여 약 10배의 낮은 내구성을 가지고 있기 때문에 수명을 연장하는 기법이 매우 중요하다. 기존의 연구에서는 가비지 컬렉션(garbage collection)과 마모도 평준화(wear leveling)를 통해서 수명을 연장하고자 노력하였다. 그러나 기존의 연구에서는 수명과 오버헤드를 동시에 고려하지 않았기 때문에 이를 그대로 휴대용 장치에 적용하기 어렵다[3-12].

Table 1. Comparison of NAND flash memories

Cell Type	SLC	MLC
Read latency	$25us$	$60us$
Program latency	$200us$	$800us$
Erase latency	$700us$	$1.5ms$
# of erases per block	10^5	10^4

따라서 본 논문에서는 EPET(Enhanced Prediction and Elapsed Time) 마모도 평준화 기법을 제안한다. EPET 마모도 평준화 기법은 기존의 임베디드 시스템을 기반으로 하는 PET 마모도 평준화 기법의 일부 기능을 차용함과 동시에 PET 마모도 평준화의 장점인 작업량(workload)을 예측하는 특성을 보다 더 정교하게 구현함으로써 시스템의 수명을 더욱 연장시킨다. 본 논문에서 제안하는 EPET 마모도 평준화는 다음과 같은 차별성을 가진다.

1. EPET 마모도 평준화는 장치 적응형 마모도 평준화 기법으로, 휴대용 장치에 특화된 마모도 평준화 기법을 사용한다.
2. EPET 마모도 평준화 기법은 기존 PET에서 TH 조건에 따른 민감한 성능을 보완하여 전반적인 수명을 연장하였다.
3. EPET 마모도 평준화는 공간 효율성을 최적화시킴으로써, 다른 기법과 비교하여 우수성을 가진다.

이하 본 논문의 구성은 다음과 같다. 2장에서는 배경 지식과 관련 연구를 소개하고 3장에서는 기존의 PET 마모도 평준화와 제안하는 EPET 마모도 평준화의 구조적 차이를 보여주며, PET 마모도 평준화의 문제점을 해결하기 위한 새로운 기법을 서술한다. 4장은 PET 마모도 평준화와 EPET 마모도 평준화의 TH 차이에 따른 성능을 비교하고, 최적의 TH를 찾는다. 이후 기존의 다른 기법과의 비교 분석을 통하여 성능 평가를 진행한다. 마지막으로 5장에서는 결론을 맺는다.

II. Background and Related Works

1. Background

낸드 플래시 메모리는 기존의 하드 디스크와는 다른 몇가지 특성을 가지고 있기 때문에 하드 디스크 기반의 파일 시스템 및 응용 프로그램을 그대로 사용할 수 없다. 따라서 낸드 플래시 메모리는 이와 같은 단점을 해결하고자 낸드 플래시 전용 파일 시스템 및 전용 응용 프로그램을 사용하였다[13-14]. 최근에는 하드 디스크 기반의 기존 파일 시스템의 장점을 그대로 활용하기 위해서 FTL(Flash Translation Layer)을 사용한다. 휴대용 장치의 FTL은 일반적으로 그림 1과 같이 구성한다. FTL의 구성은 크게 세가지 모듈로 구성되며, 각각의 모듈에 따른 특성은 다음과 같다[15].

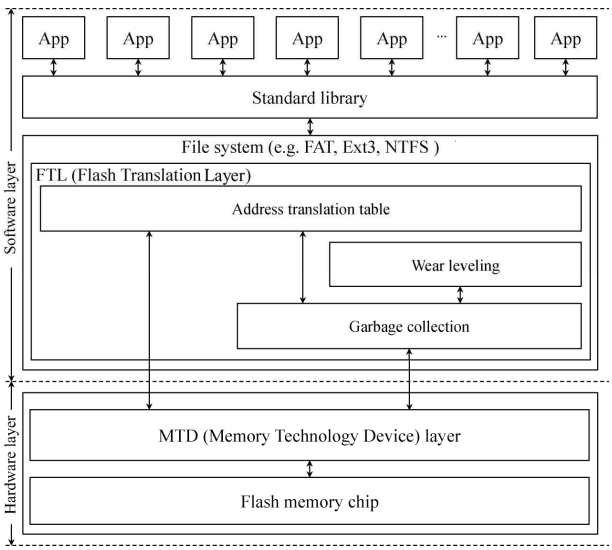

Fig. 1. System architecture based on portable devices

첫 번째, 낸드 플래시 메모리는 물리 주소에 데이터가 존재할 경우 제자리 덮어쓰기를 위해 쓰기 전 소거연산을 선행해야 한다. 하지만 앞서 언급한 바와 같이, 이와 같은 쓰기 전 소거연산은 성능 저하의 원인이 된다. 따라서 낸드 플래시 메모리의 제자리 덮어쓰기 문제를 해결하기 위해서 주소 변환 테이블(address translation table)을 사용한다. 주소 변환 테이블은 낸드 플래시 메모리에 데이터가 존재하지 않는 물리 주소를 찾아 주소 변환 테이블에 물리 주소를 매핑하는 방식으로 제자리 덮어쓰기 문제를 해결한다. 주소 변환 테이블은 물리 주소의 매핑 방법에 따라, 페이지 단위인 페이지 매핑, 블록 단위인 블록 매핑, 페이지 매핑과 블록 매핑의 장점을 고려한 하이브리드 매핑으로 구분할 수 있다.

두 번째, 낸드 플래시 메모리를 사용하는 시스템은 최소한의 빈 블록(free block)을 유지하며, 이러한 빈 블록이 일정 개수보다 작은 경우 빈 블록을 추가로 확보한다. 이와 같은 과정을 가비지 컬렉션이라 한다. 가비지 컬렉션은 빈 블록을 확보하기 위해, 사용 중인 블록(used block) 가운데 하나의 블록을 선택한다. 선정된 블록에 유효 페이지(valid page)가 존재할 경우, 가비지 컬렉션은 모든 유효 페이지를 다른 블록으로 이주시킨 후, 희생 블록의 소거연산을 수행한다.

세 번째, 시스템은 쓰기 패턴에 따라 일정한 구역만 참조하는 쓰기영역 불균형 문제를 발생시킬 수 있다. 이와 같은 문제는 블록의 내구성 불균형을 초래하며, 그 결과 특정 구역에서의 빠른 배드블록의 발생을 초래한다. 이와 같은 경우, 낸드 플래시 메모리는 배드블록으로 인하여 공간 효율성이 떨어지게 된다. 따라서 마모도 평준화는 블록 내구성 균등화를 통해 수명을 연장한다. 마모도 평준화는 참조하는 영역에서만 동작을 수행하는 동적 마모도 평준화(dynamic wear leveling)와 전체 영역에서 균등화를 수행하는 정적 마모도 평준화(static wear leveling)로 구분할 수 있다.

2. Related Works

낸드 플래시 메모리의 성능을 향상시키기 위해서 많은 연구가 진행되어왔다. 가비지 컬렉션은 희생 블록 선정 정책에 의해서 응답시간과 오버헤드를 줄이면서 수명을 연장하였다. 마모도 평준화는 자주 참조 되지 않는 데이터(프로그램 코드 데이터, 커널 파일 등)를 희생 블록 선정 정책에 참여시킴으로써 낸드 플래시 메모리의 수명을 연장시키는데 초점을 두었다[16]. 대표적인 마모도 평준화 기법은 다음과 같다.

GA(Greedy Algorithm) 기법은 가장 대표적인 가비지 컬렉션 기법으로, 전체 블록 중 무효 페이지(invalid page)의 개수가 가장 많은 블록을 희생 블록으로 선정한다. GA는 다른 기법과 비교하여 구현이 쉽고 간편하지만, 현재 비용만을 산정하기 때문에 희생 블록을 선정하는 효율이 떨어지는 단점을 가지고 있다[17].

DP(Dual-Pool) 기법은 쓰기연산 이후에 마모도 평준화를 수행하는 기법으로, 블록의 상태를 구분하기 위해 핫 블록을 가지는 핫 풀(hot pool)과 콜드 블록을 가지는 콜드 풀(cold pool)을 사용하여 마모도 평준화를 수행한다. 핫 풀과 콜드 풀 사이에 마모도 횟수가 TH를 초과 할 경우, 콜드 풀의 데이터를 핫 풀 안으로 이주시키는데, 이와 같은 과정을 더티 스왑(dirty swap)이라고 부른다. 이 과정에서 DP는 오랫동안 더티 스왑이 발생하지 않는 블록을 발생시킬 수 있다. 이러한 문제를 해결하기 위해서 더티 스왑이 발생하였을 때 더티 스왑 시간을 고려하여 문제를 해결하고자 노력했지만, 이는 근본적인 해결책이 되지 않는다[18].

BET(Block Erase Table)는 하나 혹은 다수의 블록이 하나의 비트에 대응되는 테이블을 사용한다. BET는 가비지 컬렉션에 의해 희생 블록으로 선정되고, 대응하는 비트에 소거연산이 발생하였을 때 비트 체크(소거연산이 발생하였을 때 0에서 1로 체크)를 한다. BET는 시스템에서 지정한 기간을 나타내는 T의 값을 초과할 경우, 가비지 컬렉션에서 비트 체크되지 않은 블록을 강제로 이주시킴으로 마모도 평준화를 수행한다. BET는 단순히 비트 체크만을 사용하여 마모도 평준화를 수행하기 때문에, 블록 당 편차는 적지만 페이지 이주비용이 크다[19].

Adaptive WL(Wear Leveling) 기법은 GA의 단점인 쓰기 패턴의 선형적인 문제를 인식하고, 이로 인해 수명이 줄어드는 문제를 해결하고자 노력한 기법이다. Adaptive WL은 모든 블록의 마모도 차이 정보를 가지는 $State_{diff}$가 시스템에서 지정한 지수 쓰기 패턴 정보인 $Bound$를 초과 할 경우 마모도 평준화를 수행함으로써 수명을 연장시켰다[20].

PET(Prediction of Elapsed Time) 마모도 평준화는 무효 페이지의 변화를 통해 경과시간을 예측하는 기법이다. PET 마모도 평준화는 가비지 컬렉션이 발생하였을 때, 무효 페이지의 변화를 측정하며, 이러한 변화를 토대로 PET 마모도 평준화는 핫 블록(hot block)과 콜드 블록(cold block)을 구분한다. 또한 PET 기법은 마모도 평준화를 수행할 때 빈번한 블록 선정을 방지하기 위해서, 직전에 선정한 블록을 선정하지 않는 방식으로 마모도 평준화를 수행한 기법이다[21].

기존의 PET 마모도 평준화 기법은 메모리 사용량에 초점을 둔 기법으로 임베디드 시스템에 최적화하였기 때문에 첫 번째 배드 블록이 발생한 이후 시스템 수명 한계까지 오랜 기간이 걸리는 문제점이 존재하며, 이는 전반적인 시스템의 성능에 좋지 못한 영향을 미친다. 따라서 본 논문에서는 이러한 문제를 “long-term system failure time problem”이라 정의하며, 이를 해결하는 새로운 기법을 제안한다.

III. Enhanced Prediction and Elapsed Time-based Wear Leveling

1. Overview

이 장에서는 제안하는 EPET 마모도 평준화의 구조적 특성을 보여주고, 기존의 PET 마모도 평준화의 문제점을 해결하는 기법을 서술한다. 먼저, EPET 마모도 평준화는 마모도 예측의 주요 기준 값인 경과시간을 예측하기 위한 EPET 테이블을 사용하며, 블록의 상태 정보를 구분하기 위해 BaL(Block-aware Level) Count 테이블을 사용한다. BaL Count 테이블은 핫 블록과 콜드 블록을 구분하는 용도로 사용된다. 다음으로 EPET 마모도 평준화의 희생 블록 선정 정책을 서술하며, 해당 방식과 기존 방식의 차이를 분석한다. 마지막으로 PET 마모도 평준화의 문제점인 long-term system failure time problem을 해결하기 위한 기법을 제안한다.

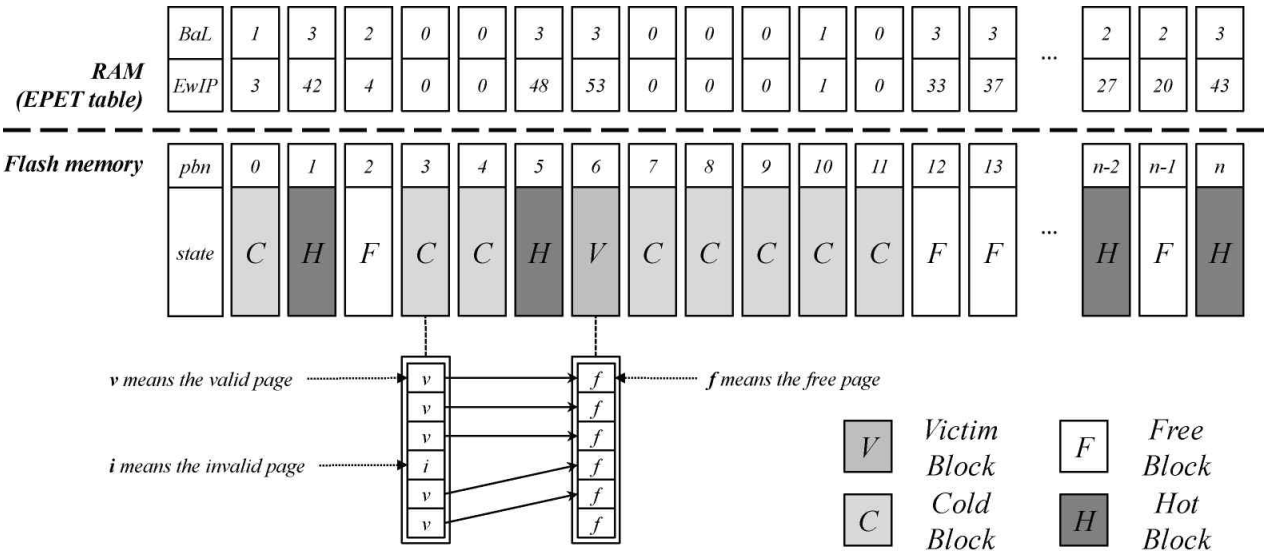

Fig. 3. Procedure for EPET wear leveling

2. EPET Table

EPET 테이블의 각 라인들은 모든 블록을 기준으로 각 블록과 1:1로 대응되며, 대응되는 최소 단위를 EPET 엔트리(Enhanced Prediction and Elapsed Time Entry)라 부른다. EPET 엔트리는 블록의 상태를 측정하기 위해서 두 가지 정보를 가진다. 첫 번째, EwIP(Exponential average with Invalid Page)는 가비지 컬렉션에 의해서 발생하는 무효 페이지 개수의 변화량을 측정하며, 이는 쓰기 패턴의 변화를 측정하는 지표로 사용하게 된다. 두 번째, BaL(Block-aware Level)은 EwIP에 따른 쓰기 패턴의 변화를 반영하여 블록의 마모도 단계를 구분하는데 사용한다. BaL은 콜드 블록과 핫 블록을 세분화하기 위해 네 단계로 구성하며, BaL의 저 단계(0, 1)는 콜드 블록을 의미하며, 고 단계(2, 3)는 핫 블록을 의미한다. 결과적으로 BaL의 최하위 혹은 최상위 단계는 매우 극히 혹은 매우 빈번히 참조되는 블록에 해당 될 것이다. 본 논문에서는 무효 페이지 개수의 변화량을 측정하기 위해서 식 (1)을 사용하여 현재의 EwIP(C_EwIP)를 구한다.

$$C_EwIP = (I + EwIP) \times \beta \qquad (1)$$

I는 블록의 무효 페이지 개수, $EwIP$는 현재까지 계산된 EwIP의 무효 페이지 개수의 변화량, β는 가중치, C_EwIP는 계산된 EwIP의 값이다(다음 EwIP). 본 논문에서는 현재 무효 페이지와 과거 무효 페이지를 5:5로 고려하기 때문에 β 는 0.5의 가중치를 가진다.

3. BaL Count Table

가비지 컬렉션과 EPET 마모도 평준화를 수행하는 동안에 BaL은 핫 블록과 콜드 블록을 구분한다. 이 과정에서 가비지 컬렉션은 핫 블록에 해당하는 블록을 선택하게 될 것이며, 이는 블록 간의 마모도 불균형을 초래한다. 따라서 EPET 마모도 평준화는 이러한 블록 간의 불균형을 추적하고 방지하기 위해 BaL Count 테이블을 사용한다.

```
Algorithm 1: EPET wear leveling
Input: idx, e_idx, h_blk, c_blk, a_ewip, c_ewip, h_ratio, TH, and B
Output: gc_time
   /* greedy algorithm */
1  a_ewip ← GetAvgEwIP();
2  for idx = 0 to B do
3     h_blk ← GetGreedyBlock();
      /* EPET table update */
4     c_ewip ← GetEwIPCost();
5     if c_ewip < a_ewip then
6        increase BaL[idx];
7     else
8        decrease BaL[idx];
9     end if
10    EwIP[idx] ← c_ewip;
11 end for
12 gc_time ← EraseBlock(h_blk);
   /* EPET wear leveling condition */
13 if h_ratio < TH then
14    for idx = 0 to B do
15       c_blk ← GetColdBlock();
16    end for
17    gc_time ← MigrateColdBlock(v_blk, c_blk);
18    gc_time ← EraseBlock(c_blk);
19    e_idx ← c_blk;
20 end
```

Fig. 2. Pseudo-code for EPET wear leveling

BaL Count 테이블은 BaL에 해당하는 네 단계의 마모도 횟수를 측정하고 핫 블록과 콜드 블록의 불균형 정도를 파악한다. BaL Count 테이블은 마모도 평준화를 수행할 때 빈번한 페이지 이주로 인하여 응답시간이 증가하는 문제를 방지하기 위해 사용한다. 본 논문에서는 성능 평가를 통하여 EPET 마모도 평준화를 수행함에 있어서 핫 블록과 콜드 블록의 비율(TH)에

따른 수명과 오버헤드를 측정하고 최적의 성능을 찾는다.

4. Wear Leveling Policy

EPET 마모도 평준화는 크게 두 가지 부분으로 구성된다. 가비지 컬렉션은 EwIP와 BaL의 쓰기 패턴 변화를 측정한다. EPET 마모도 평준화는 제안하는 기법의 마모도 평준화를 수행한다. 그림 2는 EPET 마모도 평준화에 대한 의사코드를 보여주고 있다.

시스템은 빈 블록이 부족할 경우 가비지 컬렉션을 호출한 뒤, EPET 테이블의 평균 EwIP(a_ewip)을 구한다(1번째 줄). 가비지 컬렉션은 모든 블록에 대해 GA 방식을 사용하여 희생블록(h_blk)을 선정한다. 이 과정에서 EPET 테이블의 현재 EwIP(c_ewip)를 구하고, 이 값이 EwIP의 평균보다 아래 일 경우 단계를 낮추고 아닐 경우 단계를 높여서 BaL의 단계 재조정을 수행한다. EPET 마모도 평준화는 PET 마모도 평준화의 테이블 갱신 알고리즘을 가비지 컬렉션 검색과정 동안에 수행함으로써 검색 성능을 향상시켰다(2~11번째 줄). 이후 GA에 의해서 선택한 희생 블록은 소거 연산을 수행한 뒤 가비지 컬렉션 지연시간(gc_time)을 갱신한다(12번째 줄).

EPET 마모도 평준화는 빈번한 이주를 방지하기 위해 BaL Count 테이블의 핫 비율(h_ratio)이 지정한 임계값(TH)을 초과할 경우에 한해 마모도 평준화를 수행한다(13번째 줄). 마모도 평준화를 수행할 때, EPET 마모도 평준화는 식(2)에 의해서 비용을 산정한다.

$$\begin{array}{ll} if\ \ (idx == e_idx) & max_cost \\ otherwise & c_{cost} = \dfrac{bal}{max_bal} + \dfrac{ewip}{max_pages} \end{array} \qquad (2)$$

idx는 블록 번호, e_idx는 직전에 선택된 콜드 블록 번호, bal은 블록의 BaL, max_bal는 최대 BaL, $ewip$는 블록의 $EwIP$, max_pages는 최대 페이지 수, c_cost는 산정한 비용이다. EPET 마모도 평준화는 가장 작은 c_cost에 해당하는 블록을 콜드 블록으로 선정한다. 결과적으로 직전에 선정된 콜드 블록을 제외하고, 가장 접근 빈도가 낮은 블록을 콜드 블록으로 선정한다(14~16번째 줄).

다음으로 EPET 마모도 평준화로 인해 선정된 콜드 블록의 모든 유효 페이지는 가비지 컬렉션에 의해 선정된 희생 블록으로 페이지 이주를 수행한 후, 가비지 컬렉션 지연시간(gc_time)을 갱신한다(17번째 줄). 이후 EPET 마모도 평준화에 의해서 선택된 콜드 블록은 소거연산을 수행한 뒤 가비지 컬렉션 지연시간(gc_time)을 갱신한다(18번째 줄).

EPET 마모도 평준화에 대한 자세한 수행과정은 그림 3과 같다. 그림 3에서 6번 블록이 GA에 의해서 희생블록으로 선정되었고, 3번 블록이 EPET 마모도 평준화에 의해서 콜드 블록으로 선정되었다. 6번 블록은 GA에 의해서 소거연산이 발생하였기 때문에 모든 페이지는 사용가능한 페이지(free page)를 가지게 된다. 따라서 3번 블록에 있는 모든 유효 페이지를 6번 블록으로 이주시킨다. 결과적으로 이 과정은 핫 블록에 콜드 블록을 이주시킴으로써 블록의 효율성을 높이는 결과를 가져온다. 마지막으로 EPET 마모도 평준화는 빈번한 콜드 블록 선정을 방지하기 위해 콜드 블록을 갱신한다(19번째 줄).

5. Long-term System Failure Time Problem

앞서 서술한 바와 같이, Long-term system failure time problem의 경과 시간을 단축하면 공간 효율성을 증가시킬 뿐만 아니라 시스템의 수명을 연장시킬 수 있다. 이와 같은 문제를 해결하기 위해서, 본 논문에서는 기존의 블록 할당 정책의 문제점을 인식하였다. 기존의 블록 할당 정책은 FIFO(First In First Out)를 사용한다. FIFO는 가장 단순하고 구현이 쉬운 알

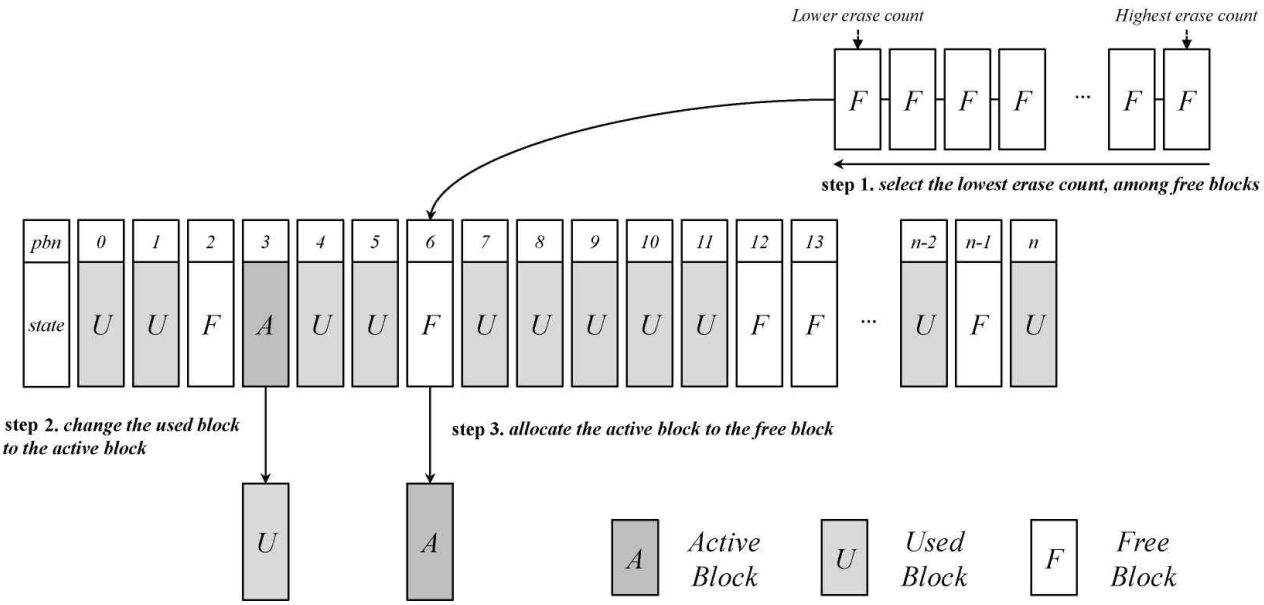

Fig. 4. Free block allocation policy of EPET wear leveling

고리즘으로써 소수의 블록만을 자주 할당하는 문제점을 가지고 있어 효율성이 떨어진다. 제안하는 블록 할당 정책은 기존의 블록 할당 정책을 해결하기 위해 블록의 편차를 비교하여 적은 마모도를 가지는 블록을 할당하여 할당하면서 블록 당 편차를 줄인다.

그림 4는 제안하는 블록 할당 정책을 보여주고 있다. EPET 마모도 평준화는 활성화 블록(active block)에서 사용 가능한 페이지가 부족할 경우 빈 블록 중에 마모도 횟수가 가장 적은 블록을 선정한다(step 1). 이후 활성화 블록을 사용 중인 블록으로 변경시킨다(step 2). 마지막으로 제안하는 블록 할당 정책에 의해 선정된 빈 블록을 활성화 블록으로 변경시킨다(step 3). 따라서 제안하는 블록 할당 정책은 콜드 블록을 빈 블록으로 먼저 할당하게 되어 효율적으로 내구성 균등화를 수행 할 수 있다. 또한 이를 통해서 기존의 PET 마모도 평준화의 문제점인 *TH*에 민감한 문제점을 제안하는 블록 할당 정책을 통해 해결 할 수 있으며, 성능 평가를 통해서 *TH*에 따른 성능 차이를 비교 분석할 것이다.

IV. Performance Evaluation

1. Experiment Setup

Table 2. Parameters of simulation environment

Parameter	Value and Description
Total capacity	4Gb
Reserved free blocks	15%
Garbage Collection Trigger	# of free blocks < 5%
Flash chip elements	1
Planes per elements	1
Blocks per plane	2048
Pages per block	64
Page size	4KB
Page read latency	60us
Page program latency	800us
Block erase latency	1.5ms
Blocks per lifetime	10^4

본 논문에서 제안하는 EPET 마모도 평준화의 성능 평가를 위해 DiskSim Simulation 4.0에 마이크로소프트사에서 개발한 SSD 모델을 추가한 SSD Extension for DiskSim Simulation Environment을 사용하여 성능 평가를 진행하였다[22-23]. 모의실험을 진행하기 위해서, 삼성전자에서 개발한 MLC 낸드 플래시 메모리의 스펙에 해당하는 지연시간, 페이지, 블록, 내구성 등을 기반으로 실험 환경을 구축하였으며, 자세한 매개변수는 표 2와 같다[2]. 시뮬레이션에 사용한 트레이스(trace)는 정규 분포를 사용하였다. 정규 분포는 가장 일반적인 확률 분포이다. 자세한 분포는 그림 5와 같으며, 표 3과 같은 쓰기 패턴을 따른다.

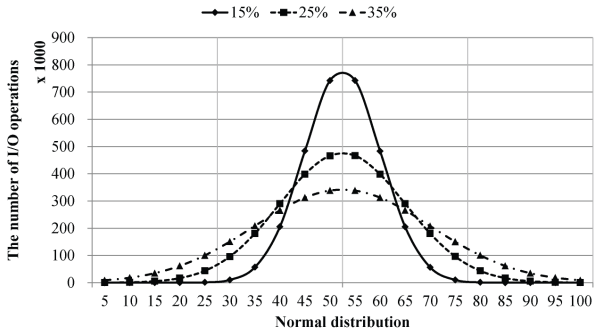

Fig. 5. Normal distribution of each trace

성능평가는 다음과 같이 네 단계로 구분한다. 첫 번째, 기존의 PET 마모도 평준화의 문제점인 *TH*에 민감한 문제점을 제시하고, 이를 EPET 마모도 평준화 기법과 비교하여 최적의 *TH*를 찾는다. 두 번째, EPET 마모도 평준화의 최적 *TH*를 이용하여 다른 기법들과 시스템 수명을 비교 분석하고, long-term system failure time problem을 확인한다.

Table 3. A dataset of simulation environment

trace	hot ratio	cold ratio
trace1	15%	85%
trace2	25%	75%
trace3	35%	65%

세 번째, 마모도 평준화에 의해서 발생하는 추가적인 오버헤드를 측정한다. 마지막으로, 마모도 평준화를 수행함에 있어서, EPET 마모도 평준화와 다른 기법과의 표준 편차를 비교 분석한다.

2. Experiment Results

EPET 마모도 평준화의 성능을 입증하기 위해서, 기존의 PET 마모도 평준화와 제안하는 기법의 *TH*에 따른 성능을 비교하였다. 그림 6은 EPET 마모도 평준화와 PET 마모도 평준화를 10^8번 쓰기연산을 수행하였을 때의 오버헤드와 첫 번째 배드 블록 발생시간이다. *TH*의 성능을 비교하였을 때, 핫 블록 비율이 90% 일 때 가장 높은 성능을 보였으며, EPET 마모도 평준화는 핫 블록의 비율이 95%까지 오버헤드는 줄고 성능은 유사하게 보였다. 이후에는, 성능과 오버헤드 둘 다 줄어드는 것을 볼 수 있다. 따라서 본 논문에서는 최고의 성능을 보여준 핫 블록 비율 90%, 비교적 높은 성능과 낮은 오버헤드를 보여준 핫 블록 비율 95%, 97%를 사용하였다. EPET 마모도 평준화는 PET 마모도 평준화와 비교하여 유사한 오버헤드를 발생시킴에도 *TH*에 따라 성능차이가 적어 PET 마모도 평준화에서 *TH*에 민감한 문제점을 해결하였다. 다음으로 EPET 마

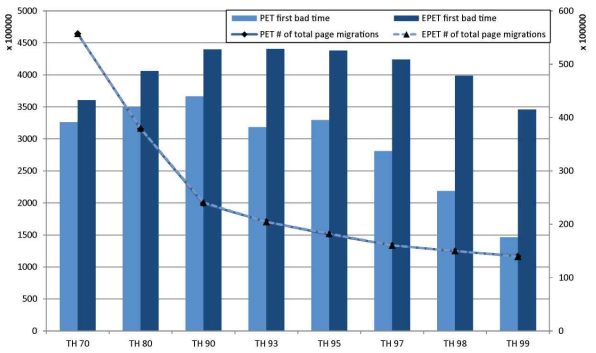

Fig. 6. Performance comparison of EPET and PET

모도 평준화의 최적 *TH*를 이용하여 관련 연구와의 성능 차이를 분석한다.

먼저 낸드 플래시 메모리의 중요한 성능 지표 중 하나인 시스템 수명을 측정하였다. 그림 7은 10^3의 블록 내구성을 가정하고 시스템 수명을 측정하였다. EPET 마모도 평준화는 GA, DP, BET, Adaptive WL, PET-TH 90, PET-TH 95, PET-TH 97과 비교하여 첫 번째 배드 블록 발생 시간을 평균적으로 328.9%, 326.8%, 271.0%, 124.3%, 23.9%, 36.8%, 60.1%까지 연장시켰으며, 시스템 불능 시간은 평균적으로 275.2%, 284.9%, 242.8%, 94.3%, 4.5%, 6.3%, 11.4%까지 연장시켰다. 또한 시스템 수명은 302.1%, 305.9%, 256.9%, 109.3%, 14.2%, 21.6%, 35.8% 연장하였다.

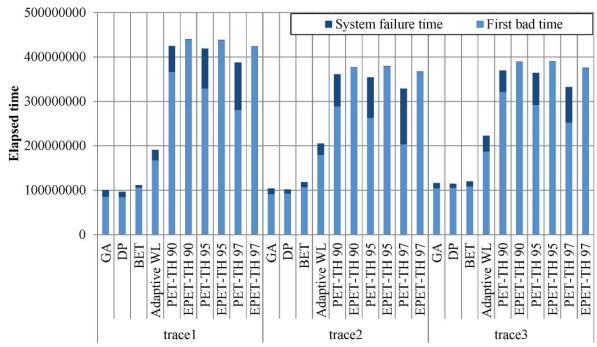

Fig. 7. System lifetime

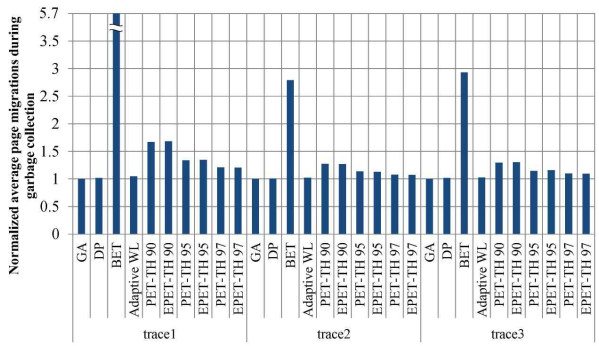

Fig. 9. Normalized average page migrations

본 논문에서는 3장 5절에서 long-term system failure time problem의 문제점을 언급했다. EPET 마모도 평준화 기법이 이와 같은 문제점을 해결하였는지 확인하기 위해서 전체 시스템 수명에서 long-term system failure time이 차지하는 비율을 측정하였다. 그림 8은 첫 번째 배드 블록이 발생한 이후부터 시스템 불능 시간까지의 경과시간을 전체 시스템의 수명을 기준으로 차지하는 상대적 비율로 보여주고 있다. 본 논문에서 제안한 EPET 마모도 평준화는 약 0.1%의 경과시간을 차지하는 것을 보여주고 있으며, 특히 PET 마모도 평준화는 다른 기법과 비교해 평균 22.6%의 높은 경과 시간을 차지하고 있다. 결과적으로, EPET 마모도 평준화는 제안하는 블록 할당 정책에 의해서 long-term system failure time problem을 해결하였다.

다음으로 마모도 평준화를 수행하는 동안의 오버헤드를 측정하였다. 오버헤드의 측정으로, 본 논문에서는 10^8번의 쓰기연산을 수행하였을 때 가비지 컬렉션의 평균 페이지 이주와 페이지 응답시간을 측정하였다. 그림 9는 평균 페이지 이주를 GA 기법을 기준으로 정규화한 것을 보여주고 있다. EPET 마모도 평준화는 PET 마모도 평준화와 유사하게 페이지 이주가 발생하였으며, BET보다 최대 4배 낮은 오버헤드를 보였다. Adaptive WL은 GA와 유사한 결과를 보였다. 이는 해당 기법의 특성상, Adaptive WL이 가비지 컬렉션을 수행함에 있어서 희생 블록을 선정하는데 간접적으로 영향을 주기 때문에 이와 같은 결과를 가져왔다. 하지만 EPET 마모도 평준화는 핫 블록

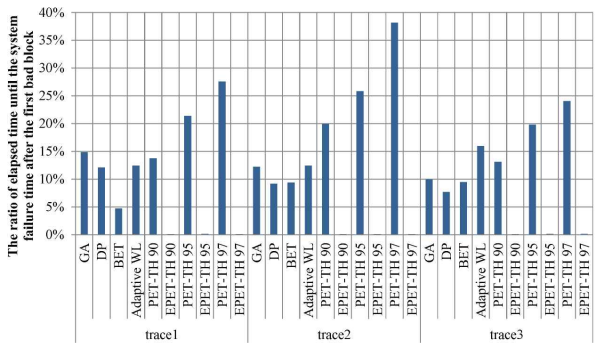

Fig. 8. The ratio of elapsed time until the system failure time after the first bad block

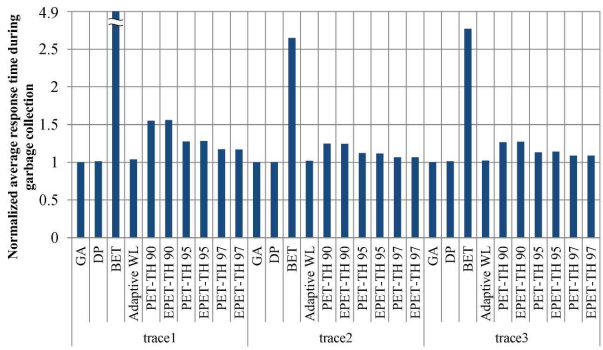

Fig. 10. Normalized average response time

의 비율이 97%일 경우, 비교적 낮은 오버헤드를 보여주며 EPET 마모도 평준화는 최대 약 8.7%의 오버헤드로 305.3%까지의 성능을 향상 시킬 수 있기 때문에 낮은 오버헤드로 마모도 평준화를 수행한다. 따라서 EPET 마모도 평준화는 최소한의 오버헤드로 최적의 성능 향상을 보인다고 할 수 있다.

그림 10은 가비지 컬렉션을 수행하였을 때의 평균 응답 시간을 정규화하여 보여주고 있다. 그림 9에서의 페이지 이주 감소로 인해 응답 시간도 함께 비례하여 감소됨을 확인할 수 있다. Adaptive WL의 경우 응답시간 측면에서 우수하나 기본적으로 시스템 수명 측면에서 매우 불리하여, 우수성의 효과가 없다고 할 수 있다.

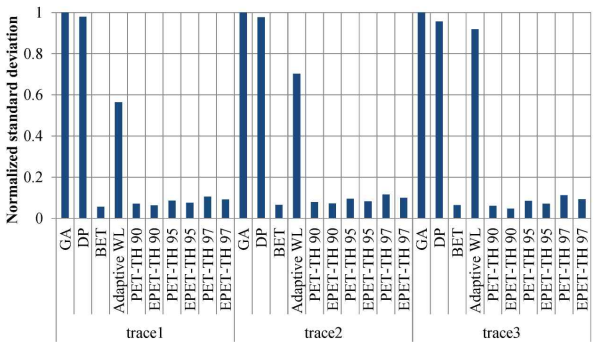

Fig. 11. Normalized standard deviation

마지막으로 108번의 쓰기연산을 수행하였을 때의 표준 편차를 측정하였다. 표준 편차는 마모도 평준화의 성능을 측정하는데 중요한 지표가 된다. 시스템의 수명은 블록 당 내구성의 균등화를 통해 이루어지기 때문이다. 그림 11은 표준 편차를 GA 기법을 기준으로 정규화한 것을 보여주고 있다. EPET 마모도 평준화는 GA, DP, Adaptive WL과 비교하여 낮은 편차를 보여주고 있으며, BET와 비교하여 거의 유사하다. 이 지표에서 볼 수 있듯이 BET는 표준 편차에 치중한 나머지 수명의 이득보다 상대적으로 더 큰 오버헤드를 초래한 것을 확인 할 수 있으며, EPET 마모도 평준화는 다른 기법들과 비교하여 효율적으로 블록을 선택하면서 편차를 줄임과 동시에 수명을 연장하는 것을 확인 할 수 있었다. 결과적으로 EPET 마모도 평준화는 PET 마모도 평준화의 단점인 long-term system failure time problem를 해결함과 동시에 적은 페이지 이주 및 오버헤드로 수명을 연장하였다.

이상으로 본 논문에서 제시한 EPET 마모도 평준화와 기존의 기법들과의 상세한 성능 차이를 표 4를 통해 요약하였다. 표 4에서는 각 기법의 가장 최고 성능 지표를 기반으로 GA와 비교하여 변화량(difference)을 측정하였으며, EPET 마모도 평준화는 약 8.7%의 오버헤드만으로 305.3%의 성능을 향상시켰다. 또한 마모도 평준화에 있어서는 약 84.8%의 표준 편차를 줄임으로써, 비교적 낮은 오버헤드로 마모도 평준화를 수행하는 것을 확인 하였다.

V. Conclusions

본 논문에서는 EPET 마모도 평준화 기법을 제안하였다. EPET 마모도 평준화는 PET 마모도 평준화 기법을 차용함과 동시에 long-term system failure time problem를 해결한 기법이다. EPET 마모도 평준화는 PET 마모도 평준화의 작업량 예측의 특성을 향상시킴으로써 전반적인 성능을 향상시켰으며, long-term system failure time problem을 줄여 이를 최적화하고 공간 효율성을 높였다. 다른 기법과 비교하여 EPET 마모도 평준화는 첫 번째 배드 블록의 발생 시간을 GA기법 대비 최대 328.9% 연장시켰으며, 시스템은 수명은 최대 305.9% 연장시켰다. 또한 EPET 마모도 평준화는 약 8.7%의 오버헤드로

Table 4. Summaries of performance comparison for each technique

Technique	First bad time		System lifetime		Page migrations		SE (Standard Deviation)	
	sec	difference	sec	difference	pages	difference	SE	difference
GA	85314	0.00	100184	0.00	12880201	0.00	1582.86	0.00
DP	84667	-0.76	96297	-3.88	13102908	+1.73	1551.59	-1.98
BET	117864	+38.15	125625	+25.40	73719886	+472.35	87.86	-94.45
Adaptive WL	161880	+89.75	184870	+84.53	13495139	+4.77	893.02	-43.58
PET-TH 90	366227	+329.27	424662	+323.88	23895839	+85.52	111.79	-92.94
EPET-TH 90	439613	+415.29	440030	+339.22	24083658	+86.98	98.53	-93.78
PET-TH 95	329382	+286.08	418866	+318.10	18097126	+40.50	136.36	-91.39
EPET-TH 95	437973	+413.37	438584	+337.78	18236994	+51.59	119.97	-92.42
PET-TH 97	280763	+229.09	387702	+286.99	16057669	+24.67	166.41	-89.49
EPET-TH 97	424042	+397.04	424362	+323.58	16042489	+24.55	144.31	-90.88
EPET-TH 98	398697	+367.33	424362	+298.28	15013225	+16.56	177.74	-88.77
EPET-TH 99	345774	+305.30	346106	+245.48	14001054	+8.70	240.86	-84.78

305.3%까지의 성능을 향상시켰기 때문에, 비교적 낮은 오버헤드로 마모도 평준화를 수행하는 것을 확인 하였다. 향후 연구에서는 낸드 플래시 메모리의 수명을 연장하기 위해서 메모리 혹은 디스크 캐시를 활용하여 쓰기 연산의 횟수를 줄이는데 초점을 맞추어 연구를 진행할 예정이다.

REFERENCES

[1] MT29F4G08ABADAWP 8Gbit SLC NAND Flash Memory Data Sheet, http://www.micron.com, Micro Technology, 2009

[2] K9GAG08U0M 2G x 8bit MLC NAND Flash Memory Data Sheet, Samsung Electronics, https://www.samsung.com, Feb, 2007

[3] Wang, Chundong, and Weng-Fai Wong. "Observational wear leveling: an efficient algorithm for flash memory management." Design Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE. IEEE, 2012.

[4] Lee, Junghee, et al. "A semi-preemptive garbage collector for solid state drives." Performance Analysis of Systems and Software (ISPASS), 2011 IEEE International Symposium on. IEEE, 2011

[5] Chang, Li-Pin, and Li-Chun Huang. "A low-cost wear-leveling algorithm for block-mapping solid-state disks." ACM SIGPLAN Notices. Vol. 46. No. 5. ACM, 2011.

[6] Lin, M. W., and Song Yan Chen. "Swap time-aware garbage collection policy for NAND flash-based swap system." Electronics Letters 49.24, pp. 1525-1526, 2013.

[7] Lim, Seung-Ho, and Kyu-Ho Park. "An efficient NAND flash file system for flash memory storage." Computers, IEEE Transactions on 55.7, pp. 906-912, 2006.

[8] Ji, Seunggu, and Dongkun Shin. "Locality and duplication-aware garbage collection for flash memory-based virtual memory systems." Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on. IEEE, 2010.

[9] 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, 2011.

[10] Jeong, Jaehyeong, and Yong Ho Song. "A technique to improve garbage collection performance for NAND flash-based storage systems." Consumer Electronics, IEEE Transactions on 58.2, pp. 470-478, 2012.

[11] Chung, Ching-Che, Duo Sheng, and Ning-Mi Hsueh. "A high-performance wear-leveling algorithm for flash memory system." IEICE Electronics Express 9.24, pp. 1874-1880, 2012.

[12] Yang, Ming-Chang, et al. "New ERA: new efficient reliability-aware wear leveling for endurance enhancement of flash storage devices." Proceedings of the 50th Annual Design Automation Conference. ACM, p. 163, 2013.

[13] Sang-Ho Hwang, Jong Wook Kwak, "CL-Tree : B+ tree for NAND Flash Memory using Cache Index List", Journal of The Korea Society of Computer and Information, Vol. 20, No. 4, pp. 1-10, Apr. 2015.

[14] Seon Hwan Kim, Jong Wook Kwak, "Garbage CollectionMethod using Proxy Block considering Index Data Structure based on Flash Memory", Journal of The Korea Society of Computer and Information, Vol. 20, No. 6, pp. 1-11, Jun. 2015.

[15] Ma, Dongzhe, Jianhua Feng, and Guoliang Li. "A survey of address translation technologies for flash memories." ACM Computing Surveys (CSUR) 46.3, pp. 36, 2014.

[16] Yang, Ming-Chang, et al. "Garbage collection and wear leveling for flash memory: Past and future." Smart Computing (SMARTCOMP), 2014 International Conference on. IEEE, pp.66-73, 2014.

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

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

[19] Chang, Yuan-Hao, Jen-Wei Hsieh, and Tei-Wei Kuo. "Improving flash wear-leveling by proactively moving static data." Computers, IEEE Transactions on 59.1, pp. 53-65, 2010.

[20] Liao, Jianwei, et al. "Adaptive wear-leveling in flash-based memory." Computer Architecture Letters 14.1 pp. 1-4 2015.

[21] Kim, Sung Ho, and Jong Wook Kwak. "Prediction of Elapsed Time based Wear Leveling for NAND Flash Memory in Embedded Systems." International Journal of Applied Engineering Research 11.1, pp. 578-585, 2016.

[22] V. Prabhakaran and T. Wobber, “SSD Extension for DiskSim Simulation Environment”, 2010.

[23] BUCY, J., SCHINDLER, J., SCHLOSSER, S., AND GANGER, G. “The DiskSim Simulation Environment Version 4.0” http://www.pdl.cmu.edu/DiskSim, 2010

Authors

Sung Ho Kim received a B.S. degree in Department of Computer Engineering from Yeungnam University College, 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.

Jong Wook Kwak received a B.S. degree in Computer Engineering from Kyungpook National University, Daegu, 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.