

<https://doi.org/10.7236/JIIBC.2019.19.6.139>  
JIIBC 2019-6-20

## 비휘발성 메모리의 마모도 평준화를 위한 레드블랙 트리

# A Swapping Red-black Tree for Wear-leveling of Non-volatile Memory

정민성\*, 이은지\*\*

Minseong Jeong\*, Eunji Lee\*\*

**요약** 비휘발성 메모리는 높은 용량과 DRAM에 준하는 수준의 접근 성능을 제공하여 차세대 메모리 기술로 각광을 받고 있다. 최근 미국 반도체 시장을 중심으로 비휘발성 메모리가 상용화 되면서 그 활용 방법에 대한 관심은 더욱 고조되고 있다. 그러나 비휘발성 메모리는 쓰기 연산 시 셀이 마모되는 물리적 특성을 가지고 있어 마모 평준화를 수행하지 않으면 특정 셀의 과도한 마모로 메모리의 용량이 감소되는 현상이 발생할 수 있다. 본 논문은 현재 균형 이진 탐색 트리로 널리 사용되고 있는 레드-블랙 트리(Red-black tree)가 비휘발성 메모리 위에서 동작할 때 잦은 리밸런싱 동작이 트리의 상위 레벨 노드들의 빈번한 쓰기를 발생시켜 특정 셀의 마모를 가속화 시킨다는 것을 관찰하고, 이를 해결하기 위한 새로운 형태의 레드-블랙 트리를 제안한다. 실제 시스템에서 추출한 레드-블랙 트리 접근 트레이스를 활용한 성능 평가에서 제안된 레드-블랙 트리는 기존 자료구조 대비 셀 간의 쓰기 횟수 편차를 최대 12.5% 감소시킴을 보여주었다.

**Abstract** For recent decades, Non-volatile Memory (NVM) technologies have been drawing a high attention both in industry and academia due to its high density and short latency comparable to that of DRAM. However, NVM devices has write endurance problem and thus the current data structures that have been built around DRAM-specific features including unlimited program cycles is inadequate for NVM, reducing the device lifetime significantly. In this paper, we revisit a red-black tree extensively adopted for data indexing across a wide range of applications, and make it to better fit for NVM. Specifically, we observe that the conventional red-black tree wears out the specific location of memory because of its rebalancing operation to ensure fast access time over a whole dataset. However, this rebalancing operation frequently updates the long-lived nodes, which leads to the skewed wear out across the NVM cells. To resolve this problem, we present a new swapping wear-leveling red-black tree that periodically moves data in the worn-out node into the young node. The performance study with real-world traces demonstrates the proposed red-black tree reduces the standard deviation of the write count across nodes by up to 12.5%.

**Key Words** : Red-black Tree, Non-volatile Memory, System Software, Wear-leveling

\*비회원, LG CNS 정보기술연구소 (본 연구는 충북대학교 컴퓨터과학과 Received: 23 September, 2019 / Revised: 3 November, 2019 / 석사과정 중에 참여함.) Accepted: 6 December, 2019

\*\*정회원, 숭실대학교 스마트시스템소프트웨어학과 (교신저자)  
접수일자: 2019년 9월 23일, 수정완료: 2019년 11월 3일  
게재확정일자: 2019년 12월 6일

\*Corresponding Author: ejlee@ssu.ac.kr

Dept. of Smart Systems Software, Soongsil University, Korea

## 1. 서 론

Phase-Change Memory(PCM), STT-RAM 등으로 대표되는 비휘발성 메모리(Non-volatile Memory)는 지난 십여 년간 차세대 메모리 기술로 각광을 받아 왔다. 비휘발성 메모리는 기본적으로 바이트 단위의 접근이 가능하고 DRAM에 준하는 성능을 제공하면서도 전원 없이 데이터를 유지할 수 있는 영속성을 띠고 있기 때문에 메모리와 스토리지 영역에서 모두 활용될 수 있기 때문이다<sup>[1][2]</sup>. 또한 비휘발성 메모리의 높은 집적도는 현재 스케일링(Scaling) 한계에 부딪히고 있는 DRAM을 대체 또는 보완하여 데이터 중심 응용들의 높은 메모리 요구량을 충족시킬 수 있을 것으로 기대되고 있다<sup>[3]</sup>.

이에 많은 반도체 제조업체들은 지속적인 비휘발성 메모리 기술 발전을 이루어 왔으며 지난해부터는 실제 상용화 제품이 출시되기 시작했다. 인텔과 마이크론은 합작 프로젝트로 3D-xpoint 기술을 기반으로 한 고성능·고집적도의 영속성을 제공하는 옵테인 메모리를 발표하였으며<sup>[4]</sup>, SK하이닉스 역시 비휘발성 메모리 양산 계획을 발표한 바 있다<sup>[5]</sup>. 구글과 페이스북을 비롯한 다양한 IT 서비스 제공업체들은 수백GB급의 용량을 지원해주는 이러한 차세대 메모리를 실제 시스템에서 효율적으로 활용하기 위한 방향을 적극적으로 모색하고 있다<sup>[3][6]</sup>.

그러나 이상적인 메모리로 일컬어지고 있는 비휘발성 메모리의 장점 이면에는 다양한 제약점 또한 존재한다. 먼저 비휘발성 메모리 셀은 최대 쓰기 횟수에 제한이 있는 마모성 소자이다. 특정 횟수 이상으로 쓰기를 수행할 시에는 해당 셀에 저장된 데이터를 읽어내지 못하거나 데이터를 올바르게 프로그램하지 못하게 된다. 이러한 특성은 사실상 무제한의 쓰기 횟수를 제공하는 DRAM과 대조적이라고 할 수 있다. 현재 Phase-change Memory 의 경우 최대 쓰기 횟수가  $10^8$ 으로 알려져 있는데<sup>[7]</sup>, 비휘발성 메모리가 상대적으로 많은 접근 횟수를 발생시키는 메인 메모리로 사용되는 경우에는 내구성에 문제가 있을 수 있다.

한편, 절대적인 쓰기 횟수 뿐 아니라 셀 간 마모도에 균형을 이루는 것도 매우 중요하다. 특정 셀이 과도하게 마모될 경우 전체 메모리의 수명이 다하기 전에 용량이 줄어드는 현상이 발생하여 성능이 크게 저하될 수 있기 때문이다.

본 논문은 이러한 쓰기 내구성에 제한이 있는 비휘발성 메모리를 효율적으로 활용하기 위하여 비휘발성 메모리의 마모도 평준화를 제공하는 새로운 레드-블랙 트리

(red-black tree)를 제안한다. 레드-블랙 트리는 대표적인 균형 이진 탐색 트리로 데이터의 삽입(Insertion), 검색(Lookup), 삭제(Deletion) 연산을  $O(\log N)$  시간에 제공한다. 자료구조가 간결하고 비교적 구현이 쉽기 때문에 레드-블랙 트리는 빠른 데이터 저장 및 검색이 필요한 환경에서 빈번히 활용된다.

예를 들면 리눅스 커널은 프로세스들의 가상 메모리 영역을 관리하기 위한 자료구조로 레드-블랙 트리를 사용하고 있고 CPU 스케줄링을 위해 프로세스를 관리하는 용도로도 레드-블랙 트리를 활용한다. 또한 최근 비정형 데이터를 처리하기 위한 NoSQL 시스템에서도 데이터의 메타데이터 등을 관리하기 위한 용도로 레드-블랙 트리를 사용한다. 레드-블랙 트리의 이러한 높은 활용도는 다른 자료구조 대비 메모리의 접근횟수 및 사용용량 측면에서 효율성이 높기 때문이다.

그러나 레드-블랙 트리는 비휘발성 메모리 위에서 동작할 경우 특정 노드의 마모로 내구성 문제를 초래할 수 있다. 데이터의 고속 접근을 지원하기 위해 레드-블랙 트리는 지속적으로 트리 구조를 리밸런싱(Rebalancing)하는 작업을 수행하는데 이 때 노드 간의 쓰기 트래픽이 균형을 이루지 못하고 삽입 및 회전이 빈번히 발생하는 노드에 쓰기 트래픽이 집중되기 때문이다. 특히 트리 내부에 오랫동안 존재하는 노드가 잦은 리밸런싱 연산에 노출되어 빈번한 업데이트가 발생하는 경우가 관찰되었다. 이러한 동작은 비휘발성 메모리의 특정 셀의 수명을 단축시켜 메모리의 수명이 다하기 전에 용량을 감소시키는 치명적인 문제가 될 수 있다.

이에 본 논문에서는 비휘발성 메모리의 쓰기 편중 문제를 해결하고 셀 간의 마모도 균형을 이루기 위한 스왑 기반 마모도 평준을 위한 레드-블랙 트리(Swapping Red-black Tree)를 제안한다. 제안하는 스왑 기반 레드-블랙 트리는 각 노드에 쓰기 횟수를 유지하고 노드의 쓰기 횟수가 특정 횟수를 초과하는 경우 새로운 메모리 공간을 할당받아 메모리 공간을 스왑한다. 이는 마모도가 심한 노드의 메모리 영역을 새로운 메모리 셀에 옮겨 적음으로써 전체적인 메모리 마모도 평준화를 이룰 수 있다.

한편, 스왑 기반 마모도 평준 레드-블랙 트리가 실제 시스템에서 실효성을 나타내기 위해서는 물리 메모리를 할당하는 운영체제의 메모리 관리 시스템도 함께 고려가 되어야 한다. 현재 리눅스의 메모리 할당 시스템은 메모리의 쓰기 횟수를 고려하지 않기 때문에, 마모도 평준화를 위해 새로운 메모리 공간을 할당 받더라도 이미 노화

된 메모리가 재활당될 수 있기 때문이다. 이에 본 논문에서는 쓰기 횟수 인지형 버디 시스템을 함께 사용하여 비휘발성 메모리의 마모도를 평균화하였다.

성능평가를 위해 리눅스의 메모리 할당 시스템과 스왑 기반 마모도 평균 레드-블랙 트리의 시뮬레이터를 구현하였으며 다양한 워크로드를 사용하여 트레이스 기반 시뮬레이션을 수행하였다. 성능평가 결과 스왑 기반 레드-블랙 트리는 기존 레드-블랙 트리 대비 노드들의 쓰기 횟수의 표준 편차를 7% - 12.5% 감소시켰다.

본 논문의 구성은 다음과 같다. 2장에서는 스왑 기반 마모도 평균화 레드-블랙 트리의 구조에 대해 자세히 설명한다. 3장에서는 메모리 쓰기 횟수를 고려한 메모리 할당 시스템에 대하여 설명한다. 4장에서는 성능평가에 대해 논의하고 5장에서는 결론을 맺는다.

## II. 스왑 기반 마모 평균 레드-블랙 트리

레드-블랙 트리는 기본적으로 탐색 속도를 개선하기 위해 데이터의 삽입과 삭제 시 리밸런싱에 의해 트리 탐색 경로의 균형을 맞춘다. 이로 인해 추가적인 쓰기 증폭 현상으로 쓰기 횟수가 증가하는 현상이 시스템에 영향을 줄 수 있다. 또한 쓰기 증폭 현상이 발생하는 것 뿐만 아니라 트리 내부에 생성된 오래된 노드들은 리밸런싱에 빈번하게 노출되어 노드 내부의 포인터가 자주 변경된다. 이와 같이 특정 노드의 빈번하게 포인터가 변경되는 것은 노드가 소유하고 있는 비휘발성 메모리의 특정 소자만을 마모시켜 해당 소자가 속한 메모리를 사용할 수 없게 되는 치명적인 결함이 발생한다.

본 논문에서 제안하는 스왑 기반 마모 평균화 레드-블랙 트리는 이러한 마모의 불균형을 해소하기 위하여 각 노드별로 쓰기 횟수를 유지한다. 각 노드에 쓰기가 발생할 때 마다 저장된 쓰기 횟수를 참고하여 허용된 최대 쓰기 횟수를 초과한 경우 메모리 공간을 재활당 하여 새로운 메모리 영역에 해당 노드의 데이터를 옮겨 적어 특정 물리 메모리 셀의 집중적인 쓰기를 방지한다.

그림 1은 제안하는 레드-블랙 트리의 동작을 보여준다. 이 예제는 노드별로 최대 허용 쓰기 횟수가 100회라고 가정한다. 이 경우 그림 1에서 킷값이 6인 노드의 총 쓰여진 횟수가 100회에 도달하였으므로 해당 노드에게 새로운 메모리 영역을 할당하여 주고 기존에 가지고 있던 메모리 영역을 해제 한다.

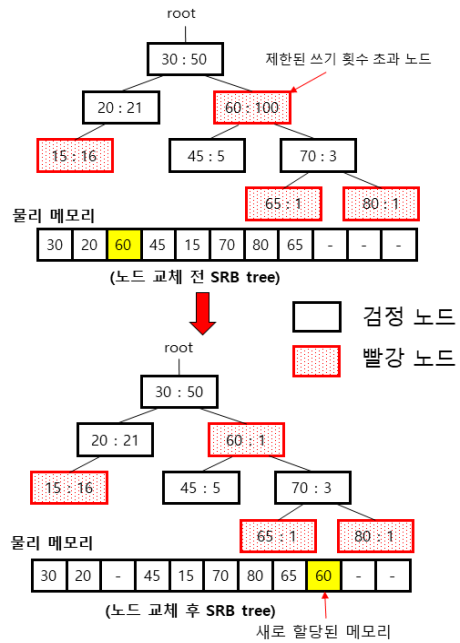


그림 1. 스왑 기반 마모도 평균 레드-블랙 트리 동작  
 Fig. 1. Swapping Wear-leveling Red-black Tree

노드 교체 레드-블랙 트리의 전체적인 동작 과정을 순서대로 나타내면 그림 2와 같다. 레드-블랙 트리에 리밸런싱이 일어날 때마다 해당 노드의 쓰기 횟수가 값이 제한된 값을 초과하는지 체크한다. 만약 초과하는 경우 해당 노드는 교체 대상 노드를 담고 있는 스왑 리스트(Swap list)에 추가된다. 레드-블랙 트리의 리밸런싱은

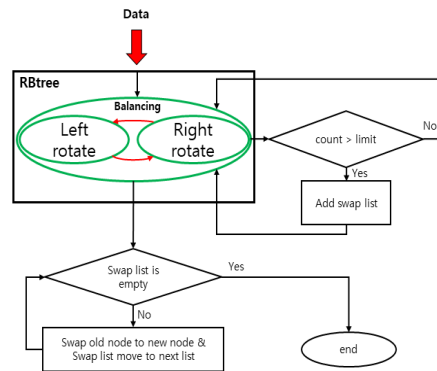


그림 2. 노드 스왑핑 알고리즘  
 Fig. 2. Node-swapping algorithm

상위 노드로의 재귀적인 동작이 요구되기 때문에 모든 리밸런싱 작업이 끝나면 스왑 리스트에 추가된 각 노드를 탐색하며 새로운 메모리 공간으로 스왑핑 해준다. 스왑핑 작업이 완료된 노드들은 스왑 리스트에서 삭제된다.

### III. 쓰기 횟수를 고려한 메모리 할당

기존 리눅스의 메모리 할당 시스템은 가용 메모리 공간을  $2^N$  크기로 구분하여 관리한다. 버디 시스템으로 알려져 있는 메모리 관리 시스템은 메모리 크기 별로 연결 리스트를 유지하고, 새로운 메모리 공간에 대한 할당 요청이 발생한 경우 요청된 메모리 크기를 만족시킬 수 있는 가장 작은 크기의 메모리 조각을 찾아 할당하게 된다. 문제는 새로운 메모리 할당 시 쓰기 횟수에 대한 고려가 없다는 것이다. 이로 인해 집중적인 쓰기를 회피하기 위해 메모리 재할당을 시도 하더라도 버디 시스템에서 쓰여진 횟수를 고려하지 않고 사용 가능한 메모리를 할당하여 주기 때문에 이미 마모가 많이 진행된 메모리 공간이 재할당 될 수 있다.

이와 같은 내부적인 시스템 문제는 스왑 기반 마모 평균 레드 블랙 트리를 사용하더라도 실제 물리적 메모리의 마모도 평균화를 이룰 수 없게 된다. 이에 본 논문은 리눅스에서 새로운 메모리를 할당할 때 쓰기 횟수가 작은 메모리 영역을 할당해 주도록 개선한다.

그림 3은 쓰기 횟수를 고려한 메모리 할당 버디 시스템을 보여준다. 그림에서 보여주는 바와 같이 각각의 가용 메모리 청크마다 쓰기 횟수를 유지한다. 이 때 청크 사이즈가 1 이상인 경우에는 모든 블록의 쓰기 횟수의 합산 값을 유지하도록 한다. 메모리 요청에 따라 해당 크

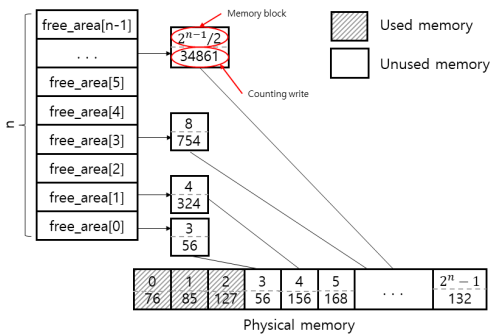


그림 3. 쓰기 횟수를 고려한 메모리 할당 시스템  
Fig. 3. Write-count aware memory allocation system

기의 블록이 할당되는 경우에 동일한 크기를 가진 가용 메모리 청크 중에 쓰기 횟수가 가장 작은 것을 할당한다. 이를 통해 새로운 메모리 할당은 상대적으로 마모수준이 낮은 공간이 할당되도록 하여 상위 계층에서의 마모도 평준화 정책이 실효성을 갖도록 한다.

### IV. 성능평가

제안된 스왑 기반 마모 평균 레드-블랙 트리의 성능을 검증하기 위해 제안된 알고리즘을 실제 구현하고 기존의 레드 블랙 트리와 성능을 비교하였다. 또한 제안된 쓰기 횟수를 고려한 메모리 할당 시스템도 함께 구현하여 물리 메모리의 마모도 평준화 효과를 측정하였다. 본 논문에서 수행된 실험은 Intel Core i5-6600에서 32GB 메모리를 사용하여 수행되었다. 시뮬레이터는 C로 구현하였고 트레이스는 실제 레드 블랙 트리가 사용되고 있는 리눅스 커널의 가상 메모리 영역에서 filebench 벤치마크를 수행했을 때 발생하는 삽입 연산을 추출하여 사용하였다.

그림 4는 레드-블랙 트리에 삽입 연산을 수행하면서 스왑 기반 마모 평균 레드 블랙 트리(RB-SW)와 기존 레드 블랙 트리(RB)의 쓰기 횟수를 측정된 결과이다. 제안하는 RB-SW는 비휘발성 메모리의 마모도 평준화를 위하여 노드의 메모리 공간 스왑핑을 수행하기 때문에 추가적인 쓰기가 발생한다. 그림 4에서 보는 바와 같이 삽입 연산의 수를 1,000개씩 증가 시켰을 때 쓰기 횟수는 기존 레드 블랙 트리 대비 제안하는 자료구조는 전체적으로 평균 4.0% 만큼의 추가적인 쓰기가 발생하였다. 이와 같이 소량의 쓰기 증가가 발생하였으나 제안하는 레드-블랙 트리는 메모리 셀 간의 마모도를 평준화 시켜 메모리를 사용하는 동안 용량이 감소하는 것을 막을 수 있다.

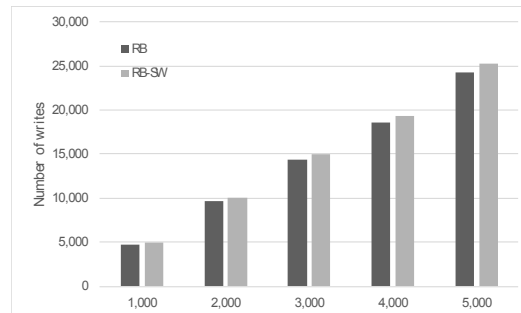


그림 4. 레드-블랙 트리에 발생한 총 쓰기량  
Fig. 4. Total memory write-traffic

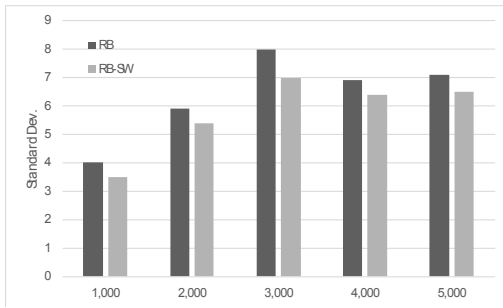


그림 5. 레드-블랙 트리의 노드별 메모리 마모도 표준편차  
 Fig. 5. Standard deviation of the write count

그림 5는 스왑 기반 마모 평균 레드-블랙 트리를 사용했을 때의 메모리 셀 간의 쓰기 횟수 편차를 기존 레드-블랙 트리를 사용했을 때의 값과 비교하여 보여준다. 그림 5에서 보여주는 바와 같이 삽입 연산의 수가 1000 단위로 증가할 때 표준편차가 7.2%에서 12.5%까지 감소하는 것을 관찰할 수 있다.

#### IV. 관련연구

최근 비휘발성 메모리의 상용화가 가시화 되면서 비휘발성 메모리의 특성을 고려하여 기존 자료구조를 개선하려는 연구가 최근 다양하게 진행되고 있다. 예로 비휘발성 메모리가 블록 디바이스와는 달리 바이트 단위의 업데이트가 가능하다는 점을 이용해 B+tree에서 발생하는 업데이트를 점진적으로 수행함으로써 갑작스러운 전력 소실에도 데이터의 손상 부분을 복구하는 연구가 수행되었다<sup>[8]</sup>. 비휘발성 메모리에서 B+tree를 사용할 때 발생하는 연산의 비용을 모델링하고 노드 내에서 데이터를 정렬형태로 유지하는 것이 성능저하를 일으킨다는 점을 고려하여 일정 부분 비정렬 상태를 유지하도록 하여 성능을 향상시키는 기법을 제안하였다<sup>[9]</sup>. 비휘발성 메모리에서 B+Tree를 운용할 때 데이터의 일관성을 위해 쓰기 순서를 보장하는 것이 심각한 성능저하가 일으킨다는 것을 관찰하고, CPU 캐쉬 플러시를 최소화할 수 있는 형태로 B+Tree를 재설계하여 기존 자료구조 대비 성능을 향상시켰다<sup>[10]</sup>. 최근 연구에서는 비휘발성 메모리에서 비트 플리핑 에러를 줄이기 위해 XOR 연산을 이용하여 자료구조에서 유지하는 포인터의 개수를 절감하는 방식을 제안한 바 있다<sup>[11]</sup>. 이 밖에도 비휘발성 메모리를 메인 메모리로 활용하여 플래시 메모리에서 발생하는 쓰기 증폭

현상을 줄이려는 연구도 수행된 바 있으며<sup>[12]</sup>, 스마트폰 환경에서 증가되는 메모리 용량에 대한 요구에 적절히 대응하기 위하여 비휘발성 메모리를 활용하는 연구도 수행되었다<sup>[13]</sup>. 또한 동일하게 마모도 문제가 존재하는 플래시 메모리에서도 블록 재활용 시 블록 간의 마모도 균등화를 위한 연구가 수행되었으며<sup>[14]</sup>, RAID-5에서 플래시의 쓰기 오버헤드를 절감하기 위한 패러티 관리 기법도 연구된 바 있다<sup>[15]</sup>.

본 연구에서 제안하는 마모 평균 레드-블랙 트리는 이전 연구들과 비슷한 동기를 가지고 있으나 비휘발성 메모리에서 발생하는 쓰기 트래픽의 감소가 아니라 메모리 셀 간의 마모 평균화를 목표로 하고 있다는 점에서 차별적이다. 이는  $10^8$ 으로 최대 쓰기 횟수가 제한되어 있는 비휘발성 메모리 환경에서 시간이 흐름에 따라 용량이 감소하는 것을 막기 때문에 메모리의 수명을 연장하는 효과를 가져올 수 있다.

#### V. 결론

본 논문에서는 최대 쓰기 횟수가 제한되어 있는 비휘발성 메모리에서 레드-블랙 트리가 사용될 때 빈번한 리밸런싱 동작으로 인해 특정 셀이 마모되는 현상을 관찰하고 이를 해결하기 위한 기법을 제안하였다. 주기적인 스왑핑을 통해 특정 메모리 영역의 마모도를 제한하는 한편 쓰기 횟수를 고려한 메모리 할당 정책을 함께 사용하여 물리 메모리의 마모도를 평균화하였다.

#### References

- [1] S.R. Dulloor, S. Kumar, A. Keshavamurthy, P. Lantz, D. Reddy, R. Sankaran, J. Jackson, "System Software for Persistent Memory," In Proceedings of the 9th European Conference on Computer Systems (Eurosys), No. 15, 2014.
- [2] J. Xu, S. Swanson, "NOVA: A Log structured File System for Hybrid Volatile/Non-volatile Main Memories," In Proceedings of the 14th USENIX Conference on File and Storage Technologies (FAST), pp.323-338, 2016.
- [3] A. Eisenman, D. Gardner, I. AbdelRahman, J. Axboe, S. Dong, K. Hazlewood, C. Petersen, A. Cidon, S. Katti, "Reducing DRAM footprint with NVM in Facebook," In Proceedings of the 9th European Conference on Computer Systems (Eurosys), No. 42, 2018.

- [4] <http://www.micron.com/products/advanced-solutions/3d-xpoint-technology>.
- [5] <https://www.eenewsembedded.com/news/iedm-sk-hynix-will-make-3d-crosspoint-memory>
- [6] [https://www.theregister.co.uk/2018/08/10/optane\\_dimm\\_ceremonially\\_ships\\_but\\_lacks\\_any\\_xeon\\_support/](https://www.theregister.co.uk/2018/08/10/optane_dimm_ceremonially_ships_but_lacks_any_xeon_support/)
- [7] X. Zhang, G. Sun, "Toss-up wear leveling: Protecting Phase-Change Memories from inconsistent write patterns," In Proceedings of the 54th ACM/EDAC/IEEE Design Automation Conference (DAC), 2017
- [8] D. Hwang, W.-H. Kim, Y. Won, B. Nam, "Endurable Transient Inconsistency in Byte-Addressable Persistent B+-Tree," In Proceedings of the 18th USENIX Conference on File and Storage Technologies (FAST), 2018.
- [9] P. Chi, W.-C. Lee, Y. Xie, "Adapting B+-Tree for Emerging Nonvolatile Memory-Based Main Memory," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol 35, No. 9, Sept. 2016.
- [10] J. Yang, Q. Wei, C. Wang, C. Chen, K. L. Yong, B. He, "NV-tree: a consistent and workload-adaptive tree structure for non-volatile memory," IEEE Transactions on Computers, Vol. 65, No. 7, July, 2016.
- [11] D. Bittman, D. D. E. Long, E. L. Miller, "Optimizing Systems for Byte-Addressable NVM by Reducing Bit Flipping," 17th USENIX Conference on File and Storage Technologies, Feb. 2019.
- [12] E. Lee, M. Jeong, H. Bahn, "NVM-based Write Amplification Reduction to Avoid Performance Fluctuation of Flash Storage," The Journal of The Institute of Internet, Broadcasting and Communication (JIIBC), Vol.. 16 No. 4, pp.15-20, 2016.
- [13] H. Choi, H. Bahn, "Buffer Cache Management based on Nonvolatile Memory to Improve the Performance of Smartphone Storage," The Journal of The Institute of Internet, Broadcasting and Communication (JIIBC), Vol.. 16 No. 3, pp.7-12, 2016.
- [13] J. Park., "Performance Analysis of K-set Flash Memory Management," Journal of the Korea Academia-Industrial cooperation Society, Vol. 5, No. 5, pp. 389-394, 2004.
- [14] S. Lim, "Metadata Log Management for Full Stripe Parity in Flash Storage Systems," Journal of Advanced Information Technology and Convergence, pp. 56-65, 2019. 10

**저 자 소 개**

**정 민 성(비회원)**



- 2016년 충북대학교 소프트웨어학과 학사.
- 2018년 충북대학교 소프트웨어학과 석사.
- 2019년 현재 LG CNS 정보기술연구소 연구원

**이 은 지(정회원)**



- 2005년 이화여자대학교 컴퓨터학과 학사
- 2012년 서울대학교 컴퓨터공학부 박사
- 2019년 숭실대학교 스마트시스템 소프트웨어학과 조교수
- 주관심분야: 데이터 시스템, 차세대 메모리, 분산 시스템, 클라우드 컴퓨팅

※ 이 연구는 교육부의 재원으로 한국연구재단(No.2017R1D1A1B03031494)의 지원을 받아 수행된 연구임.