

논문 2019-14-10

비대칭적 성능의 고용량 비휘발성 메모리를 위한 계층적 구조의 이진 탐색 트리

(A Hierarchical Binary-search Tree for the High-Capacity and Asymmetric Performance of NVM)

정민성, 이미정, 이은지*

(Minseong Jeong, Mijeong Lee, Eunji Lee)

Abstract : For decades, in-memory data structures have been designed for DRAM-based main memory that provides symmetric read/write performances and has no limited write endurance. However, such data structures provide sub-optimal performance for NVM as it has different characteristics to DRAM. With this motivation, we rethink a conventional red-black tree in terms of its efficacy under NVM settings. The original red-black tree constantly rebalances sub-trees so as to export fast access time over dataset, but it inevitably increases the write traffic, adversely affecting the performance for NVM with a long write latency and limited endurance. To resolve this problem, we present a variant of the red-black tree called a hierarchical balanced binary search tree. The proposed structure maintains multiple keys in a single node so as to amortize the rebalancing cost. The performance study reveals that the proposed hierarchical binary search tree effectively reduces the write traffic by effectively reaping the high capacity of NVM.

Keywords : Non-volatile memory, Data structure, Phase-change memory

1. 서론

지난 십여 년간 비휘발성 메모리는 지속적인 기술 발전을 이루어 왔으며 지난해부터 실제 상용화 제품이 출시되기 시작했다. 인텔과 마이크론은 합작하여 개발한 3D-xpoint 기술을 기반으로 고성능·고집적도의 영속성을 제공하는 옵테인 메모리를 발표하였으며 [1], SK하이닉스 역시 비휘발성 메모리 양산 계획을 발표한 바 있다 [2]. 이에 구글과 페이스북을 비롯한 다양한 IT 서비스 제공업체들은 수

백GB급의 용량을 지원해주는 이러한 차세대 메모리를 실제 시스템에서 효율적으로 활용하기 위한 방향을 적극적으로 모색하고 있다 [3, 4].

비휘발성 메모리는 DRAM 대비 다양한 장점을 지니고 있다. 기본적으로 비휘발성 메모리는 바이트 단위의 접근이 가능하고 DRAM에 준하는 성능을 제공하면서도 전원 없이 데이터를 유지할 수 있는 영속성을 띠고 있기 때문에 메모리와 스토리지 영역에서 모두 활용될 수 있다는 가능성이 있다 [5-10]. 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 기대하고 있다 [3].

그러나 이러한 비휘발성 메모리 기술의 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.

*Corresponding Author (ejlee@ssu.ac.kr)

Received: Feb. 23, 2019, Revised: Mar. 28, 2019,

Accepted: Apr. 05, 2019.

M. Jeong: LG CNS.

M Lee: Chungbuk National Univ.

E. Lee: Soonsil Univ.

※ 본 논문은 2017년도 정보(교육부)재원으로 한국연구재단의 지원을 받아 수행되었음(No. 2017R1D1A1B03031494).

이러한 특성은 사실상 무제한의 쓰기 횟수를 제공하는 DRAM과 대조적이라고 할 수 있다. 또한 비휘발성 메모리는 읽기의 경우 DRAM과 비슷하거나 수 배 이내의 성능을 제공하지만 쓰기 연산의 경우 최대 1000배 느린 쓰기 성능을 제공한다. DRAM과 대비되는 느린 쓰기 성능 및 내구성 문제는 비휘발성 메모리를 실제 시스템에 도입할 때 의도치 않은 성능저하 및 수명단축의 현상을 일으킬 수 있기 때문에 주의 깊은 접근이 필요하다.

본 논문은 이러한 하드웨어적인 메모리 기술 변화에 효율적으로 대응하기 위하여 현재 상용 시스템에서 널리 사용되는 레드-블랙 트리 (red-black tree)를 비휘발성 메모리의 특성을 고려하여 개선하는 연구를 수행하였다. 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 원치 않는 성능저하와 내구성 문제를 초래할 수 있다. 데이터의 고속 접근을 지원하기 위해 레드-블랙 트리는 지속적으로 트리 구조를 리밸런싱 (Rebalancing)하는 작업을 수행하는데 이 작업이 쓰기 트래픽을 상당히 증폭시키기 때문이다. 이러한 특성이 DRAM에서는 크게 오버헤드로 작용되지 않았지만 쓰기 속도가 느리고 마모 문제가 존재하는 비휘발성 메모리에서는 성능과 수명 관점에서 치명적인 문제가 될 수 있다.

이에 본 논문에서는 계층적 구조를 통해 리밸런싱 (Rebalancing) 횟수를 줄이고 쓰기 증폭 현상을 감소시키는 새로운 형태의 레드-블랙 트리를 제안한다. 제안하는 계층적 레드-블랙 트리는 테이블 (Table) 구조와 트리 구조를 효율적으로 구성하여 트리의 리밸런싱 (Rebalancing) 실행 횟수를 감소시키는 것이 핵심 아이디어이다. 이것은 비휘발성

메모리가 쓰기 성능은 느리지만 집적도가 높다는 점을 활용하여 공간사용량은 증가하더라도 쓰기 증폭은 감소시킬 수 있도록 설계되었다. 각 노드에서 단일 키-밸류 (Key-value)를 유지하던 기존 레드-블랙 트리와 달리 계층적 레드-블랙 트리는 각 노드가 다수의 데이터 셋을 테이블 형태로 유지할 수 있도록 허용한다. 이것은 노드 간의 불균형 상황 발생빈도를 낮출 수 있기 때문에 쓰기 비용을 절감할 수 있고, 결과적으로 비휘발성 메모리에서 높은 성능과 내구성을 제공할 수 있다.

제안된 계층적 레드-블랙 트리는 C언어로 구현되었으며 다양한 워크로드 기반의 성능평가에서 인덱싱에 의해 발생하는 추가적인 쓰기 트래픽을 크게 감소시켰다. 또한 계층적 레드-블랙 트리는 노드 내에서는 테이블 형태로 데이터를 유지하여 데이터를 트리에서 검색하는 비용을 줄임으로써 읽기 횟수도 감소시켰다.

본 논문의 구성은 다음과 같다. 2장에서는 계층적 레드-블랙 트리의 구조에 대해 자세히 설명한다. 3장에서는 성능평가에 대해 논의하고 4장에서는 결론을 맺는다.

II. 계층적 레드-블랙 트리

1. 레드-블랙 트리 (Red-black Tree)

현재 사용되고 있는 레드-블랙 트리 (Red-black Tree)는 1972년도에 Rudolf Bayer 에 의해 고안되었다. 노드마다 블랙 또는 레드 컬러를 유지하도록 하여 트리의 불균형을 탐지하는 방식으로 특정 경로에 레드 노드가 연속하여 등장하는 경우 노드 회전 (rotation)을 통해 트리가 균형을 유지할 수 있도록 한다. 이러한 레드-블랙 트리의 동작은 불균형을 방지했을 때 발생할 수 있는 긴 지연시간에 대한 비용을 매 삽입 연산 분할 할당하는 방식 (Amortization)이다. 결과적으로 모든 데이터에 대해 로그 시간 내의 삽입 (Insertion), 삭제 (Deletion), 검색 (Lookup) 연산을 보장한다.

그러나 매 삽입 연산마다 트리의 구조를 변형하는 레드-블랙 트리의 동작은 추가적인 쓰기 트래픽을 과도하게 발생시켜 쓰기 성능이 느리고 마모도 문제가 존재하는 비휘발성 메모리에서는 심각한 비효율성을 초래할 수 있다. 그림 1은 삽입 연산이 발생했을 때의 레드-블랙 트리의 동작과 그에 따른 쓰기 증폭 현상을 간단한 예제를 통해 보여준다. 본 예제에서는 하나의 데이터 아이템 키 (Key)를 가

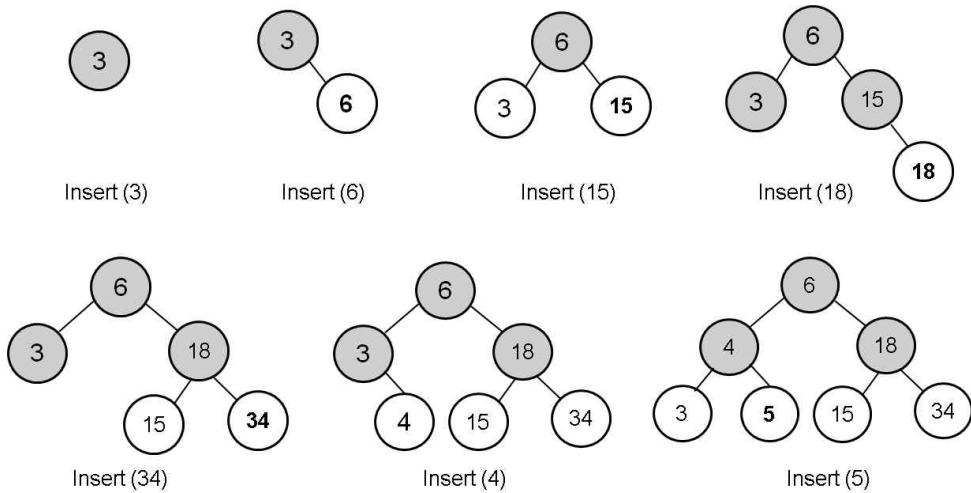


그림 1. 기존 레드-블랙 트리 구조 (회색: 블랙 노드, 흰색: 레드 노드)
Fig. 1 An original red-black tree architecture

지고 있다고 가정하고, 일련의 데이터가 아래와 같은 순서로 삽입될 때의 레드-블랙 트리의 구조를 나타낸다.

삽입순서: 3, 6, 15, 18, 34, 4, 5

최초의 회전이 발생하는 경우는 15가 삽입되는 경우로 회전하기 전에는 3, 6, 15가 하나의 경로에 존재하게 되어 루트 3을 기준으로 왼쪽 자식과 오른쪽 자식 노드의 높이 차가 2 이상이 된다. 이러한 불균형을 해소하기 위하여 노드들을 왼쪽으로 회전시켜 6을 루트로 하는 트리구조로 변환하고 6을 가진 노드는 레드에서 블랙으로 색이 변경된다. 동일한 방식으로 추후 키값 34와 5가 삽입될 때에도 회전이 발생하여 총 7개의 아이템이 삽입되는 동안 3회의 회전이 발생하였다. 총 삽입 연산 중 절반에 가까운 경우가 회전을 수반하는 것으로 상당히 높은 비율이라고 할 수 있다. 회전이 발생하면 노드들의 위상을 변경하기 위한 포인터 업데이트가 발생하고 회전된 노드의 색깔을 재설정하기 위한 쓰기도 추가적으로 발생하게 된다.

2. 계층적 구조의 이진 균형 탐색 트리 (Hierarchical Balanced Binary Search Tree)

본 논문은 상기 절에서 설명한 레드-블랙 트리의 쓰기 증폭현상을 감소시키기 위하여 계층적 구조의 이진 균형 탐색 트리를 제안한다. 제안하는 자

료구조는 레드-블랙 트리와 테이블 구조를 계층적으로 구성하여 트리의 균형 유지를 위한 빈번한 회전 연산을 발생시키지 않고도 로그 시간의 접근 성능을 제공할 수 있도록 한다. 핵심 아이디어는 기존 레드-블랙 트리가 각 노드에 하나의 키값만을 유지했던 것과 대조적으로 제안하는 트리는 하나의 노드에 다수의 데이터 셋을 저장할 수 있도록 하는 것이다. 즉, 상위 계층에서는 레드-블랙 트리 구조를 유지하고, 트리의 각 노드는 테이블 형태로 다수의 데이터를 관리하는 것이다. 이러한 방식은 이미 생성된 노드로 데이터가 삽입되는 경우 상위 레드-블랙 트리 구조에는 변동이 없기 때문에 균형 유지를 위한 오버헤드를 크게 줄일 수 있다.

그림 2는 그림 1의 예와 동일한 순서로 키가 삽입될 때 제안하는 계층적 구조의 이진 균형 트리의 구조를 보여준다. 계층적 이진 트리는 각 노드가 저장하는 데이터의 범위를 그룹 크기 (group size)로 정의한다. 위의 예제에서는 키값이 정수라고 가정하고 그룹의 크기를 10으로 설정하였다. 즉, 데이터를 그룹의 크기로 나누어 각 범위에 속하는 데이터 셋은 동일한 노드에서 관리된다. 예를 들면, 최초의 키값인 3이 삽입되었을 때는 해당 데이터가 속한 그룹 (키값 0~9)을 담고 있는 노드가 생성되지 않았기 때문에 새로운 노드를 생성하고 삽입한다. 이때 해당 노드가 유지할 수 있는 최대의 데이터 셋 크기만큼의 배열을 할당하여 유지한다. 그 후 키값 6이 도착하면 해당 키값이 속한 그룹을 담고 있는

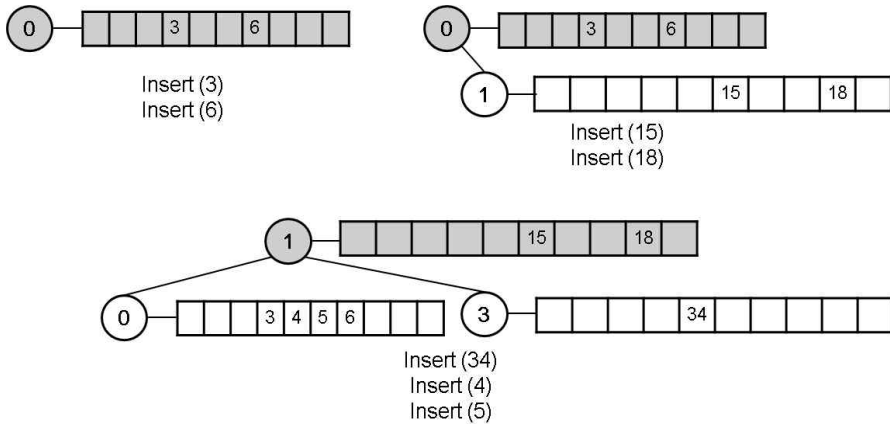


그림 2. 계층적 구조의 이진 균형 트리 구조 (회색: 블랙 노드, 흰색: 레드 노드)
 Fig. 2 A Hierarchical big-node red-black tree architecture

노드의 테이블에 값을 저장한다. 각 그룹을 대표하는 키를 주요 키 (Primary Key)라고 정의하며 레주요 키들은 레드블랙 트리로 구성된다. 위의 예제와 같이 그룹 사이즈를 10으로 설정한 경우에는 킷값의 마지막 자리수를 제외한 나머지 접두 숫자 (prefix)가 주요키가 된다. 이러한 구조를 통해 제안하는 구조에서는 회전 연산이 발생하지 않았다.

계층적 균형 이진트리는 쓰기를 유발하는 회전 연산 대신 비휘발성 메모리의 고집적도를 활용하여 높은 성능을 보장해 주는 원리이다. 테이블의 경우 자료구조의 제조정 연산 등은 존재하지 않지만 실제 삽입되지 않은 데이터에 대한 엔트리를 유지해야하기 때문에 공간 사용량이 증가하게 된다. 그러나 비휘발성 메모리의 경우 DRAM 대비 수배의 높은 집적도를 가지고 있기 때문에 공간과 성능을 트레이드 오프하는 전략은 비휘발성 메모리 환경에서는 적절한 것으로 생각할 수 있다.

성능 관점에서 계층적 균형 이진 트리를 살펴보면 실제 데이터를 삽입 및 검색하는 연산은 오히려 기존 레드-블랙 트리보다 높을 수 있다. 총 데이터의 아이템 개수가 N이라고 가정할 때 기존 레드-블랙 트리는 모든 데이터를 하나의 노드로 저장하여 $O(\log N)$ 시간복잡도를 지니는 반면, 계층적 이진 트리 구조는 그룹의 주요키만을 레드-블랙 트리 구성하고 실제 각 노드에서는 데이터를 $O(1)$ 시간에 접근할 수 있기 때문에 전체적인 성능은 $O(\log(N/G))$ (G : group size) 로 표현될 수 있다. 그룹의 사이즈는 메모리 용량에 따라 변경가능한데 그룹 크기가 1일 때는 기존 레드-블랙 트리와 동일

하고 그룹 크기가 최대 데이터 셋 크기일 때는 테이블과 동일한 구조이기 때문에 메모리의 용량 및 접근 성능에 따라 선택적으로 자료구조를 운용할 수 있는 스펙트럼을 제공한다고 하겠다.

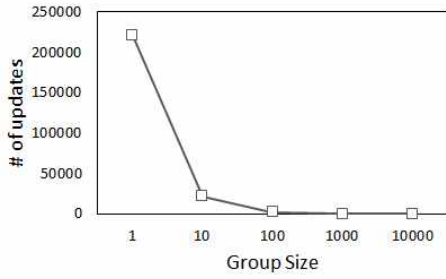
III. 성능평가

1. 실험환경

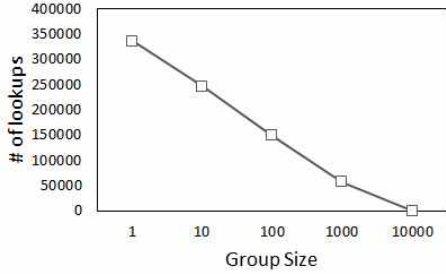
제안된 계층적 구조의 이진 탐색 트리는 성능평가를 위해 C언어로 구현되었으며 합성 워크로드 (Synthetic Workload)를 사용하여 성능을 측정하였다. 워크로드는 임의의 키를 삽입 (Insert), 검색 (Search), 삭제 (Delete) 순서로 발생시키도록 하였다. 비휘발성 메모리는 DRAM을 사용하여 에뮬레이션 하였다.

본 논문에서 탐색하고자 하는 자료구조의 성능은 워크로드의 공간 지역성에 따라 달라질 수 있다. 데이터가 발생 가능한 범위는 큰데, 실제 저장되는 데이터가 작은 영역에 밀집해 있는 경우 테이블 기반의 자료구조는 메모리 낭비가 커질 수 있다. 반면 데이터가 발생 가능한 영역에 촘촘히 분포되어 있는 경우에는 테이블 구조가 가장 효율적이며 트리 구조는 오히려 인덱싱을 위한 추가 비용이 들어 비효율적일 수 있다. 제안된 계층적 구조의 이진 탐색 트리는 레드-블랙 트리와 테이블 양극단의 자료구조 사이에서 메모리 사용량과 성능 간의 적절한 균형을 이룰 수 있는 방법을 제공한다.

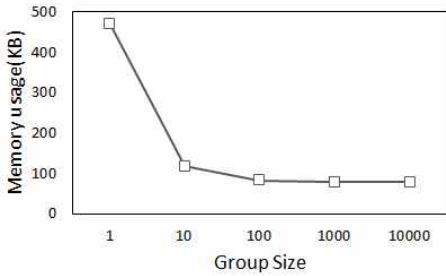
상기 기술된 효과를 평가하기 위하여 우리는 데이터의 공간 지역성 지수를 변화시키며 워크로드를



(a) Insert/Delete 연산에 대한 쓰기 횟수
(a) Number of updates for insertion/deletion



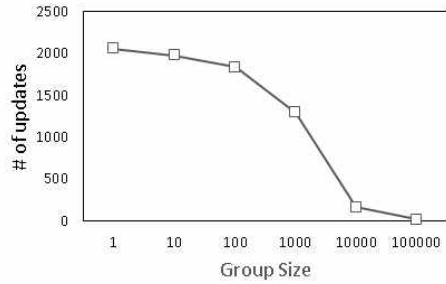
(b) Search 연산에 대한 참조 횟수
(b) Number of lookups for search



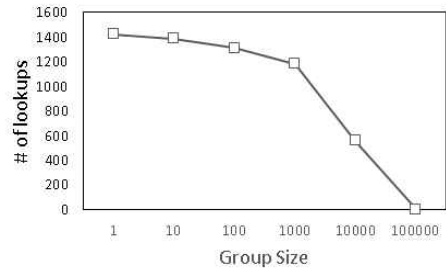
(c) 메모리 사용량
(c) Memory Usage

그림 3. 지역성 지수 1인 워크로드의 성능평가
Fig. 3 Performance of a random workload for the full dataset

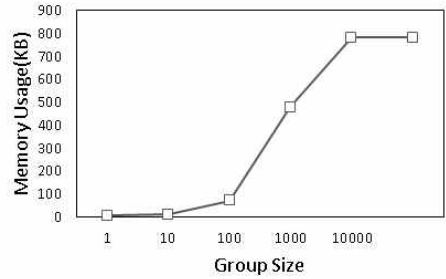
발생시켰다. 지역성 지수가 1인 것은 발생 가능한 데이터 셋과 실제 삽입된 데이터의 영역이 동일한 워크로드이다. 지역성 지수가 0.0001인 워크로드는 발생 가능한 데이터 셋의 0.1%의 영역에만 실제 데이터가 삽입된 경우이다. 예를 들어 데이터가 가질 수 있는 키값이 1부터 10000까지인데 이 중 데이터가 모든 영역에 고르게 발생하였다면 지역성 지수가 1인 것이고, 1에서 10까지의 특정 영역에 대해서만 발생하였다면 지역성 지수가 0.0001인 것이다. 지역성 지수가 낮을수록 강한 공간 지역성을



(a) Insert/Delete 연산에 대한 쓰기 횟수
(a) Number of updates for insertion/deletion



(b) Search 연산에 대한 참조 횟수
(b) Number of lookups for search



(c) 메모리 사용량
(c) Memory Usage

그림 4. 지역성 지수 0.0001인 워크로드의 성능평가
Fig. 4 Performance of a random workload for the partial dataset

나타내는 워크로드이다. 지역성 지수가 1인 워크로드는 최대 10000 이하의 키값을 10000개 발생시켰으며 지역성 지수가 0.0001인 워크로드는 최대 100000까지 키값을 발생시킬 수 있을 때 그 중 100개의 키값만 임의로 발생시켰다. 모든 키값은 지역성 특성을 유지한 범위 내에서 랜덤으로 발생되었다.

2. 성능분석

그림3은 지역성 지수가 1인 워크로드에서 계층

적 구조의 이진 탐색 트리에서 그룹 크기를 변화시키며 측정한 성능평가 결과를 보여준다. 그룹의 크기가 1인 경우는 하나의 노드가 하나의 키값만을 유지할 수 있는 상태이므로 기존 레드-블랙 트리와 동일하며 그룹 크기가 데이터 셋과 동일한 10000은 기존 테이블 구조와 동일한 상태이다. 그림3 (a)에서 보여주는 바와 같이 그룹 크기가 10이상일 때 쓰기량이 크게 감소하는 것을 볼 수 있다. 또한 그룹의 크기를 키우면 타겟 키 탐색을 위한 읽기 횟수도 감소한다. 이것은 그룹의 대표키만을 이진 트리로 유지하고 실제 노드 내에서는 테이블을 활용해 O(1)으로 타겟 키를 검색할 수 있기 때문이다.

그림3 (c)에서 지역성 지수가 1인 워크로드에서는 그룹의 크기를 증가시켰을 때 오히려 메모리의 사용량도 줄어드는 것을 볼 수 있다. 밀집도가 높은 워크로드에서는 테이블로 데이터를 유지할 때 빈슬롯이 발생하지 않아 공간낭비가 없고 트리 유지를 위한 인덱싱 정보도 유지할 필요가 없기 때문이다. 결과적으로 모든 데이터 셋이 균등하게 데이터가 저장되는 경우에는 테이블로 데이터를 유지하는 것이 가장 효율적이다.

그러나 실제 워크로드는 이러한 경우가 많지 않으며 가능한 범위의 일부만이 유입되는 경우가 대부분이다. 예를 들어 CPU 스케줄링의 경우 프로세스ID는 64비트로 표현가능한 최대 숫자까지 가질 수 있으나 실제 실행 중인 프로세스의 ID는 그중 극히 일부에 해당하는 데이터만이 저장될 것이다. 지역성 지수 0.0001 워크로드는 이러한 실제 워크로드를 모사하며 그림 4는 그에 대한 성능평가 결과이다. 그림 3의 결과와는 대조적으로 그룹 사이즈를 증가시킬수록 읽기/쓰기 트래픽은 감소하는 반면 메모리 사용량은 증가한다. 즉, 제안하는 계층적 이진 탐색 트리는 메모리의 용량과 내구성에 따라 적절히 트레이드-오프를 수행함으로써 상황에 맞게 데이터를 관리할 수 있도록 해준다.

IV. 관련연구

비휘발성 메모리의 특성을 고려하여 기존 메모리에서 널리 사용되는 자료구조를 개선하려는 연구가 최근 다양하게 진행되어 왔다. 비휘발성 메모리의 느린 쓰기 성능을 보완하기 위해 쓰기 최적화된 형태로 기존 자료구조를 향상시키는 한편, 영속적인 데이터 저장이 가능하다는 특성을 이용해 메모리 내의 자료구조가 데이터의 일관성도 보장해줄 수 있는 형태로 개선하는 연구 등이 수행되었다.

Venkataraman et al. 은 메모리의 비휘발성 특성을 이용해 B-tree에서 버진 정보를 유지하는 형태로 로깅 없이 데이터의 일관성을 제공해주는 자료구조를 제안하였다 [11]. Hwang et al. 은 비휘발성 메모리가 블록 디바이스와는 달리 바이트 단위의 업데이트가 가능하다는 점을 이용해 B+ tree에서 발생하는 업데이트를 점진적으로 수행함으로써 갑작스러운 전력 소실에도 데이터가 손상된 부분을 감지하고 회복할 수 있도록 하는 연구를 수행하였다 [12]. Chi et al. 은 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 [13]. Yang et al. 은 비휘발성 메모리에서 B+ Tree를 운용할 때 데이터의 일관성을 위해 쓰기 순서를 보장하는 것이 심각한 성능저하를 일으킨다는 것을 관찰하고, CPU 캐쉬 플러시를 최소화할 수 있는 형태로 B+ Tree를 재설계하여 기존 자료구조 대비 성능을 향상시켰다 [14]. 최근 Bittman 등이 수행한 연구에서는 비휘발성 메모리에서 비트 플리핑 에러를 줄이기 위해 XOR 연산을 이용하여 자료구조에서 유지하는 포인터의 개수를 절감하는 방식을 제안한 바 있다 [15].

본 연구에서 제안하는 계층적 구조의 이진 탐색 트리는 이전 연구들과 비슷한 동기를 가지고 있으나 단순히 쓰기량을 감소시키는 것이 아니라 비휘발성 메모리가 제공해주는 높은 공간과 느린 쓰기 성능 사이에서 적절한 트레이드 오프를 할 수 있도록 해준다는 점에서 차별적이다. 제안하는 레드-블랙 트리는 집적도와 쓰기 성능이 다양한 비휘발성 메모리 스펙트럼에서 최적의 성능을 제공하는 데에 사용될 수 있을 것으로 사료된다.

IV. 결론

본 논문은 기존의 레드-블랙 트리를 기반으로 차세대 메모리인 비휘발성 메모리에서 효율적으로 동작하는 계층적 구조의 이진 탐색 트리를 제안한다. 기존 레드-블랙 트리의 균형 유지를 위해 쓰기량 증가가 쓰기 속도가 느리고 내구성에 제한이 있는 비휘발성 메모리에서는 성능 및 수명에 부정적인 영향을 끼칠 수 있다는 점을 관찰하고 이를 개선하기 위해 비휘발성 메모리의 높은 집적도를 활용하여 쓰기량을 감소시킬 수 있는 자료구조를 고안

하였다. 삽입/삭제/탐색의 연산을 임의의 데이터에 수행하는 워크로드 기반의 성능평가에서 제안된 계층적 이진 탐색 트리는 실제 환경과 유사한 워크로드에서 메모리 공간과 쓰기량 간의 적절한 트레이드오프를 제공하는 것으로 관찰되었다.

References

- [1] Available on : <http://www.micron.com/products/advanced-solutions/3d-xpoint-technology>.
- [2] Available on : <https://www.eenewsembedded.com/news/iedm-sk-hynix-will-make-3d-crosspoint-memory>
- [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," Proceedings of the 9th European Conference on Computer Systems, No. 42, 2018.
- [4] Available on : https://www.theregister.co.uk/2018/08/10/optane_dimm_ceremonially_ships_but_1_acks_any_xeon_support/
- [5] S.R. Dulloor, S. Kumar, A. Keshavamurthy, P. Lantz, D. Reddy, R. Sankaran, J. Jackson, "System Software for Persistent Memory," Proceedings of the 9th European Conference on Computer Systems (Eurosys), No. 15, 2014.
- [6] E. Lee, K. Koh, H. Bahn, "P2FS: Supporting Atomic Writes for Reliable File System Design in PCM Storage," Proceedings of EICE Electronics Express, Vol. 11, pp. 1-6, 2014.
- [7] J. Condit, E. B. Nightingale, C. Frost, E. Ipek, B. Lee, D. Burger, D. Coetzee, "Better I/O Through Byte-addressable, Persistent Memory," Proceedings of the ACM SIGOPS 22nd Symposium on Operating Systems Principles (SOSP), pp. 133-146, 2009.
- [8] E. Lee, H. Bahn, S.H. Noh, "Unioning of Buffer Cache and Journaling Layers," Proceedings of the 11th USENIX Conference on File and Storage Technologies (FAST), pp. 73-80, 2013.
- [9] E. Lee, J.E. Jang, T. Kim, H. Bahn "On-demand Snapshot: An Efficient Versioning File System for Phase Change Memory," Journal of IEEE Transactions on Knowledge and Data Engineering, Vol. 25, No. 12, pp. 2841-2853, 2013.
- [10] J. Xu, S. Swanson, "NOVA: A Log Structured File System for Hybrid Volatile/Non-volatile Main Memories," Proceedings of the 14th USENIX Conference on File and Storage Technologies (FAST), pp. 323 - 338, 2016.
- [11] S. Venkataraman, N. Tolia, P. Ranganathan, R. H. Campbell, "Consistent and Durable Data Structures for Non-Volatile Byte-Addressable Memory," Proceedings of the 9th USENIX Conference on File and Storage Technologies (FAST), Vol. 11, pp. 61-75, 2011.
- [12] D. Hwang, W. Kim, Y. Won, B. Nam, "Endurable Transient Inconsistency in Byte-Addressable Persistent B+-Tree," Proceedings of the 18th USENIX Conference on File and Storage Technologies (FAST), pp. 187-200, 2018.
- [13] P. Chi, W.C. Lee, Y. Xie, "Adapting B+-Tree for Emerging Nonvolatile Memory-Based Main Memory," Journal of IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 35, No. 9, pp. 1461-1474, 2016.
- [14] 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," Journal of IEEE Transactions on Computers, Vol. 65, No. 7, 2169-2183, 2016.
- [15] D. Bittman, D. D. E. Long, E. L. Miller, "Optimizing Systems for Byte-Addressable NVM by Reducing Bit Flipping," Proceedings of 17th USENIX Conference on File and Storage Technologies, pp. 17-30, 2019.

Minseong Jeong (정민성)



He received B.S. in Dept. of Computer Science, Chungbuk National University (2016), M.S. in Dept of Computer Science, Chungbuk National

University (2018), He is currently a full-time researcher in LG CNS (2019).

Email: msjeong482@gmail.com

Mijeong Lee (이미정)



She is currently pursuing the undergraduate course in Dept. of Computer Science, Chungbuk National University (2019).

Email: mijeong082191@gmail.com

Eunji Lee (이은지)



She received B.S. in Dept. of Computer Science, Ewha University (2005), PhD. in Dept of Computer Engineering, Seoul National University

(2012). She is currently an assistant professor in the Department of Smart Systems Software, Soongsil University.

Email: ejlee@ssu.ac.kr