

# 다중 코어 SSD를 위한 오프라인 중복 제거 기법

곽현호<sup>○</sup>, 신동군

성균관대학교 전자전기컴퓨터공학과

gusghrhkr@skku.edu, dongkun@skku.edu

## Offline Deduplication Architecture for Multi-Core SSD

Hyunho Gwak<sup>○</sup>, Dongkun Shin

Department of Electrical and Computer Engineering, Sungkyunkwan University

### 요 약

플래시 메모리 기반 SSD는 다른 저장장치 대비 높은 읽기, 쓰기 성능을 가지고 있으며, 모바일 장치부터 서버 시스템에서까지 널리 사용되고 있다. 그런데, SSD는 많은 수의 쓰기 요청을 처리할 경우 수명이 감소하는 문제가 있다. 중복 제거 기법은 플래시 메모리에 발생하는 쓰기 수를 줄일 수 있으며 SSD의 제한된 수명 문제를 해결할 수 있다. 본 논문에서는 기존 온라인 중복 제거 기법에서 문제되었던 쓰기 요청 지연시간 증가 문제를 해결하기 위한 오프라인 중복 제거 기법을 제안한다. 또한, 기존 FTL 기반 중복 제거 기법을 적용할 수 없는, 다중 코어 SSD를 위한 HIL 기반 중복 제거 기법을 설계하여 실제 SSD에 구현하였다.

### 1. 서 론

SSD (Solid State Drive)와 같은 플래시 메모리 기반 저장장치는 기존 주 저장장치로 사용되던 HDD에 비해 많은 기술적 장점을 가지고 있다. SSD는 빠른 임의 접근 성능, 내구성과 저전력 등의 특성을 가지고 있고, 이러한 특성 때문에 모바일 장치부터 서버 시스템에서까지 널리 사용되고 있다. 그런데, SSD는 HDD와 달리 제한된 수명 문제를 가지고 있다. 플래시 메모리는 쓰기 동작 횟수에 제한이 있기 때문에 많은 쓰기는 SSD의 수명을 감소시킨다. 이와 같은 SSD의 수명 문제를 해결하기 위해 많은 기법들이 제안되었으며, 중복 제거 기법은 그 중 하나이다. 중복 제거 기법은 SSD의 특성을 활용하여 저장장치의 여유 공간을 확보하고 플래시 메모리에 기록되는 쓰기 수를 줄일 수 있다.

중복 제거 기법은 중복제거가 수행되는 시점에 따라서 온라인 또는 오프라인 중복 제거 기법으로 나눌 수 있다. 온라인 중복 제거 기법은 쓰기 수행 도중 데이터의 중복 여부를 판단하여 중복 제거를 수행하며, 오프라인 중복 제거 기법은 저장장치의 유휴 시간에 이미 기록된 데이터의 중복 여부를 판단한다. 오프라인과 달리 온라인 중복 제거 기법은 쓰기 요청을 수행할 때 중복되는 데이터가 있을 경우 쓰기 요청을 SSD에 전달하지 않으므로써 플래시 메모리에 기록되는 쓰기 수를 줄일 수 있다. 그런데, 온라인 중복 제거 기법의 경우 중복되는 데이터가 적은 환경에서는 중복 제거 검사로 인한 오버헤드가 쓰기 요청 시간을 지연시키지만 할 수 있다. 또한, 쓰기 요청을 저장장치에 전달하지 않으므로, 중복 제거 메타데이터가 기록되지 않았을 때 복구가

불가능한 문제가 있다.

본 논문에서는 앞서 언급하였던 온라인 중복 제거 기법에서 발생하는 문제를 해결하기 위한 오프라인 중복 제거 기법을 제안한다. 그리고, 실제 다중 코어 SSD에 중복 제거 기법을 구현하여 실험하였다. 기존 FTL 기반 중복 제거 기법들은 [1, 2, 3] 다중 코어 환경에서 서로 다른 FTL에 속한 데이터에 대한 중복 검사를 수행하지 못하는 문제가 있었으며, 이를 해결하기 위해 HIL (Host Interface Layer) 기반 중복 제거 기법을 사용하였다.

2장에서는 SSD내부 구조와 기존 중복 제거 기법의 구조를 설명하며, 3장에서는 본 논문에서 제안하는 오프라인 중복 제거 기법이 어떻게 구현되었는지 다룬다, 그리고, 4장에서는 실험을 통해 오프라인 중복 제거의 효과를 보여주며, 마지막 5장에서 결론과 향후 계획을 정리한다.

### 2. 배경 지식

#### 2.1 SSD 내부 구조

SSD는 크게 컨트롤러, 캐시 메모리, 플래시 메모리로 구성되어 있다. 플래시 메모리는 여러 개의 다이 (Dies)로 구성되어 있고, 다이는 여러 개의 플레인 (Planes), 그리고 플레인은 수천 개의 블록으로 구성되어 있다. 블록은 페이지로 구성되어 있으며, 플래시 메모리는 페이지 단위로 읽기와 쓰기를 수행한다. 이러한 플래시 메모리의 페이지는 덮어쓰기가 불가능하고 지우기 동작 전에는 쓰기가 불가능한 특성을 가지고 있다. 또한, 지우기 동작에는 횟수 제한이 있기 때문에 많은 쓰기는 플래시 메모리의 수명을 감소시키게 된다. 이와 같은 플래시 메모리의 특성 때문에 기존 시스템에서 사용하는 블록 계층을 사용하여 플래시 메모리에 접근할 수 없다. 그래서, SSD는 내부 컨트롤러에서 FTL (Flash Translation Layer)을 실행시킴으로써 사용자가 SSD를 HDD와 같은 블록 저장장치로써 사용할 수 있게 한다.

이 논문은 2016년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2016R1A2B2008672)

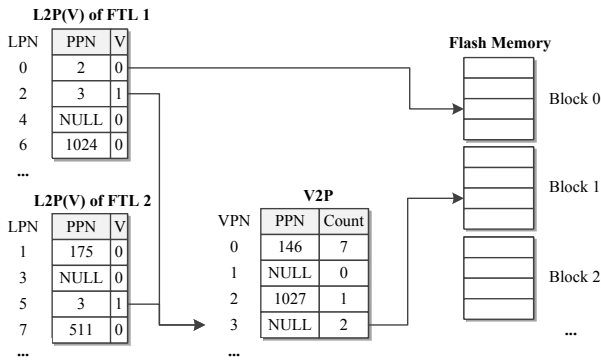


그림 1. 이중 매핑 구조

FTL은 호스트의 논리 페이지 주소를 플래시 메모리의 물리 페이지 주소로 매핑하며, 이 매핑 구조는 여러 논리 페이지가 하나의 물리 페이지를 가리키는 중복 제거 기법을 가능하게 한다. 그리고, FTL은 주기적으로 가비지 컬렉션 (Garbage Collection)을 수행하여 데이터를 기록하기 위한 프리 (Free) 블록을 확보한다. 가비지 컬렉션은 한 블록에 속한 유효한 페이지들을 복사하는 작업으로, 가비지 컬렉션이 수행되는 동안 FTL은 읽기 및 쓰기 요청을 처리할 수 없기 때문에 SSD의 성능에 크게 영향을 끼친다. 중복 제거 기법은 FTL의 가비지 컬렉션에서 복사하는 페이지의 수를 줄일 수 있기 때문에 SSD의 성능을 증가시킬 수 있으며, 플래시 메모리에 기록되는 쓰기 수를 감소시키므로 SSD의 수명 문제 또한 해결할 수 있다.

다중코어 SSD에서는 다른 코어를 조절하는 HIL을 위한 코어가 하나 동작하며, 나머지 코어에서는 FTL이 동작하여 플래시 메모리에 접근한다. HIL은 호스트의 읽기 및 쓰기 요청을 FTL에게 전달하며, FTL은 앞서 언급한 매핑 주소 관리나 가비지 컬렉션 등을 수행하며 플래시 메모리에 대한 접근을 관리한다. 그런데, 각 FTL은 서로 다른 물리 주소 영역을 관리하기 때문에, FTL 기반 중복 제거 기법을 사용할 경우 서로 다른 FTL에 속한 중복된 데이터를 감지 및 제거할 수 없는 문제가 발생한다.

2.2 중복 제거 기법

중복 제거 기법은 저장장치에 중복되어 기록되어 있는 데이터들을 제거함으로써 저장장치의 여유공간을 확보하는 기법이다. SSD에서는 FTL에서 관리하는 매핑 구조를 사용하여 중복 제거 기법을 적용할 수 있다. 같은 데이터를 가지는 여러 개의 논리 페이지들을 하나의 물리 페이지로 매핑시킴으로써 중복되는 물리 페이지만큼 저장장치의 여유공간을 확보할 수 있다.

2.2.1 이중 매핑 구조

FTL에서는 L2P (Logical to Physical) 테이블을 사용하여 논리 페이지 주소를 물리 페이지 주소로 매핑한다. 중복 제거 기법은 L2P 테이블 이외에 V2P (Virtual to Physical) 테이블을 추가로 관리하여 중복된 데이터를 가지는 논리 페이지들을 하나의 물리 페이지로 매핑시킨다. 본 논문에서 제안하는 중복 제거 기법 또한 그림 1과 같은 이중 매핑 구조를 사용한다. L2P 테이블에서 같은 가상 페이지 주소로 매핑된

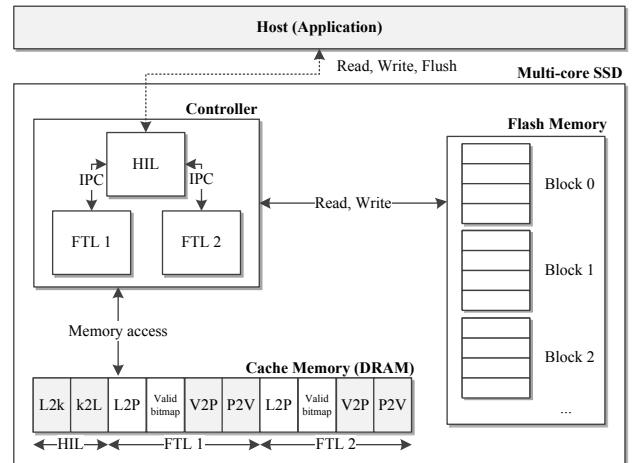


그림 2. 다중 코어 SSD 구조

페이지들은 V2P 테이블을 통해 하나의 물리 페이지로 매핑된다. 이러한 이중 매핑 구조는 FTL의 가비지 컬렉션으로 인해 물리 페이지 주소가 변경되었을 때, V2P 테이블에서 하나의 물리 페이지 주소만 수정하면 되는 이점이 있다. 이중 매핑 구조를 사용하지 않고 모든 중복된 페이지들이 L2P 테이블에서 하나의 물리 페이지 주소를 저장할 경우에는, L2P 테이블에서 모든 중복된 페이지의 물리 페이지 주소를 변경해야 하는 오버헤드가 발생한다.

2.2.2 해시 엔진의 사용

중복된 페이지를 찾기 위해서 모든 페이지를 비교하는 것은 비효율적이다. 따라서, 기존 중복 제거 기법들에서는 해시 엔진을 사용하여 페이지 비교에 소모되는 비용을 줄이고 있다. 각 물리 페이지에 대한 해시 값을 생성하여 같은 해시 값을 가지는 페이지들만을 비교함으로써 페이지 비교 수를 줄일 수 있다. 특히, 충돌 방지 (Collision-Free) 해시 엔진을 사용할 경우 해시 값을 비교하는 것으로 페이지의 중복 여부를 판단할 수 있다. 그러나, 충돌 방지 해시 엔진의 경우 해시 값을 계산하기 위한 오버헤드가 문제로 작용한다. 또한, 온라인 중복 제거 기법의 경우에는 중복된 페이지가 없을 경우 해시 엔진으로 인해 쓰기 요청의 지연시간을 크게 증가시킬 수 있다. 본 논문에서는 하드웨어 경량 해시 엔진을 사용함으로써 적은 오버헤드로 효율적으로 페이지 비교를 수행한다.

3. HIL 기반 오프라인 중복 제거 기법 설계

본 논문에서는 다중 코어 SSD를 위한 HIL 기반 오프라인 중복 제거 기법을 제안한다. 중복 제거는 저장장치의 유휴 시간에 HIL에서 수행되며, 서로 다른 FTL에서 관리하는 물리 페이지에 대해서도 중복 제거를 수행한다. 그림 2는 다중 코어 SSD 구조와 HIL기반 오프라인 중복 제거 기법에서 사용하는 메모리 구조를 보여준다.

3.1 중복된 페이지 검색

본 논문에서 제안하는 오프라인 중복 제거 기법에서는 SSD에서 ECC를 위해 하드웨어로 생성하는 CRC 값을 사용한다. HIL은 쓰기 요청이 수행될 때 각 페이지의 해시 값을 L2k (Logical to Hash Key) 테이블에 저장하며, 같은 해시

값을 가지는 물리 페이지에 대해서만 페이지 비교를 수행한다. 이때, 같은 해시 값을 가지는 페이지를 빠르게 찾기 위해서 L2k 테이블을 기반으로 k2L (Hash Key to Logical) 트리를 생성한다. k2L 트리는 쓰기 요청의 지연시간을 증가시키지 않기 위해 중복 검사가 수행되는 저장장치의 유휴 시간에만 생성된다.

### 3.2 이중 매핑 관리

저장장치의 유휴 시간에 중복된 페이지를 찾아낸 경우 이중 매핑을 사용하여 그림 1과 같이 논리 페이지 주소들을 하나의 물리 페이지 주소로 매핑시킨다. 먼저, V2P 테이블에서 가상 페이지 주소를 할당 받고 중복되는 페이지 중 하나의 물리 페이지 주소를 저장한다. 그리고, L2P 테이블에 할당 받은 가상 페이지 주소를 저장하고 virtual flag를 설정한다. 이때 V2P 테이블에 기록된 물리 페이지를 제외한 나머지 물리 페이지는 FTL의 valid bitmap을 수정하여 무효화시킨다. 이렇게 무효화된 물리 페이지는 FTL의 가비지 컬렉션을 수행할 때 복사되지 않는다.

그런데, 그림 2와 같이 각 FTL은 L2P 테이블, valid bitmap 등 자료 구조를 구분하여 관리한다. 하지만, 중복 제거 과정은 HIL에서 수행하며, HIL은 FTL의 메모리 영역에 접근할 수 있다. 또한, 중복 제거 과정은 FTL이 읽기 및 쓰기 요청을 처리하지 않는 저장장치의 유휴 시간에만 동작하므로 FTL 코어간 동기화 문제가 발생하지 않는다.

V2P 테이블은 각 FTL에 나누어서 저장되며, P2V (Physical to Virtual) 트리와 함께 관리된다. P2V 트리는 물리 페이지 주소를 기반으로 가상 페이지 주소를 빠르게 찾기 위한 자료 구조로써, FTL의 가비지 컬렉션으로 인해 물리 페이지 주소가 변경되었을 때 V2P 테이블을 수정하는 오버헤드를 줄이기 위해 사용된다.

### 3.3 중복된 페이지에 대한 읽기 및 쓰기 요청 처리

읽기 및 쓰기 요청은 HIL을 거쳐 각 FTL로 전달되며, HIL은 L2P 테이블의 virtual flag를 확인하여 페이지가 이중 매핑되었는지 확인할 수 있다. 만약 이중 매핑된 페이지에 대한 요청이 올 경우 HIL은 V2P 테이블에 추가로 접근하게 된다. 읽기 요청의 경우에는 HIL은 V2P 테이블에 기록된 물리 페이지 주소를 기반으로 적절한 FTL에게 읽기 요청을 전달한다. 쓰기 요청의 경우에는 해당 페이지에 대한 이중 매핑을 해지하는 과정을 거친다.

먼저, L2P 테이블의 virtual flag를 해지하며, V2P 테이블에 기록된 페이지 수를 감소시킨다. 그림 1과 같이 V2P 테이블은 각 가상 페이지 주소에 매핑된 페이지의 수를 관리한다. 한 가상 페이지 주소에 매핑된 페이지가 하나라도 존재할 경우 해당 가상 페이지 주소에 매핑된 물리 페이지 또한 유효한 상태로 남아있게 된다. 그리고, 해당 가상 페이지 주소에 매핑된 모든 페이지가 해지될 때 이중 매핑을 해지한다. 이중 매핑이 해지되면 V2P 테이블에 기록된 물리 페이지 또한 무효화시키며, 해당 가상 페이지 주소 또한 할당 받을 수 있는 상태가 된다.

## 4. 실험

### 4.1 실험 환경

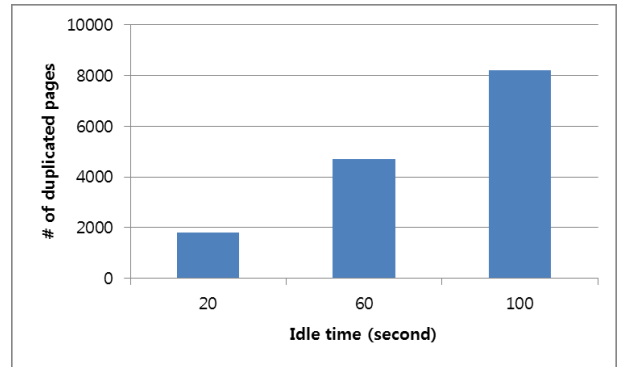


그림 3. 오프라인 중복 제거 기법의 효과

본 논문에서는 HIL 기반 오프라인 중복 제거 기법을 삼성 843 SSD에 구현하여 실험하였다. 해당 SSD는 다중 코어를 사용하며, 하나의 HIL 코어와 두 개의 FTL 코어가 동작한다. 실험은 리눅스 커널 3.13에서 진행되었으며, SSD를 ext4 파일 시스템으로 포맷 및 마운트한 뒤 사용하였다. 실험을 위해 SSD에 리눅스 커널 소스를 복사한 뒤 저장장치에 일정시간 동안 유휴시간을 발생시켰다. SSD는 주어진 시간 동안 본 논문에서 제안하는 중복 제거 기법을 수행하였으며, 이때 중복 제거 기법으로 찾아내어 제거한 중복된 페이지의 수를 측정하였다.

### 4.2 실험 결과

그림 3은 주어진 유휴 시간 동안 중복 제거 기법을 실행시켰을 때 찾아낸 중복된 페이지 수를 보여준다. 해당 페이지 수만큼 플래시 메모리에 여유 공간을 확보하여 FTL에서 수행하는 가비지 컬렉션의 오버헤드를 감소시킬 수 있다. 실험에 사용한 SSD에서 한 블록은 128개의 페이지로 구성되어 있으며, 100초동안 중복 제거를 수행하였을 때 96개의 프리 블록을 생성할 수 있었다.

## 5. 결론

본 논문에서는 다중 코어 SSD를 위한 오프라인 중복 제거 기법을 제안하며 실제 SSD에서 구현하였다. 실험 결과, SSD에 기록된 중복된 데이터를 제거함으로써 저장장치의 여유 공간을 확보하는 것을 확인하였다. 저장장치의 여유 공간을 확보함에 따라 FTL에서 가비지 컬렉션이 복사하는 페이지 수가 감소하게 된다. 이로 인해, FTL에서 가비지 컬렉션을 수행함으로써 읽기 및 쓰기 요청이 지연되는 시간이 감소하고 플래시 메모리에 발생하는 쓰기 수도 감소한다. 따라서, 중복 제거 기법으로 SSD의 성능과 수명을 모두 개선시킬 수 있다.

## 참고 문헌

- [1] C. Feng et al., "CAFTL: A Content-Aware Flash Translation Layer Enhancing the Lifespan of Flash Memory based Solid State Drives," FAST, 2011.
- [2] A. Jeongcheol, S. Dongkun, "Offline Deduplication-Aware Block Separation for Solid State Disk," FAST, 2013.
- [3] P. Eunsoo, S. Dongkun, "Offline Deduplication for Solid State Disk Using a Lightweight Hash Algorithm," Journal of Semiconductor Technology and Science, 15.5: 539-545, 2015.