

경량 해시 키를 이용한 SSD에서 오프라인 중복제거 기법

박은수, 신동군
성균관대학교 정보통신공학대학
pes9488@skku.edu , dongkun@skku.edu

Offline Deduplication in SSD with Light-weight Hash Key

Eunsoo Park, Dongkun Shin
Sungkyunkwan University

요 약

중복 제거 기법은 플래시 메모리의 쓰기 양을 줄이고, 저장장치 상의 여유 공간을 확보하여 가비지 컬렉션 비용을 절약함으로써 직·간접적으로 플래시 메모리의 수명을 증가시킬 수 있는 기법이다. 본 논문에서는 경량 해시 키인 CRC32를 이용해 불필요한 중복제거 프로세스를 줄여 기존 오프라인 기법에 비해 중복제거 비용을 최대 70%를 줄이고, 기존 온라인 기법에 비해 사용자의 쓰기 요청에 대한 응답 시간을 최대 15%만큼 감소시켰다.

본 연구는 미래창조과학부 및 정보통신산업진흥원의 서울어코드활성화지원사업의 연구결과로 수행되었음(NIPA-2014-H1807-14-1005)

1. 서 론

여러 사용자나 파일의 중복되는 데이터를 찾아내어 한 개의 데이터를 공유하도록 하는 중복제거 기법은 플래시 저장장치에서 쓰기 양을 줄이고 저장장치의 여유 공간을 확보하여 가비지 컬렉션(Garbage Collection) 비용을 줄일 수 있다.

중복제거 기법 중, 플래시 저장장치의 FTL에서 사용되는 중복제거 기법은 중복제거가 수행되는 시점에 따라 온라인 혹은 오프라인 기법으로 나뉜다. 온라인 기법은 런타임에 중복 여부를 파악하여 저장장치에 기록할지를 결정한다. 반면에, 오프라인 기법은 저장장치의 유휴 시간에 중복 여부를 판단하여 중복을 제거하는 기법이다. 온라인 기법을 사용하면 쓰기 양을 줄여 플래시 메모리의 수명을 증가시키고 중복이 자주 일어나는 상황에서는 중복되는 쓰기 요청에 대하여 저장장치에 기록하는 일이 없으므로 쓰기 응답 시간을 줄일 수 있다. 반면에 중복이 적은 환경에서는 중복 검사 시간이 오버헤드로 작용할 수 있다. 또한, 시스템의 갑작스러운 정지 상황의 경우 기록하지 못한 데이터의 복구가 불가능한 단점이 있다. 오프라인 중복제거의 경우, 여유 공간을 확보함으로써 가비지 컬렉션 시 발생하는 페이지 복사 횟수를 감소시켜 전체적인 응답 시간을 줄일 수 있다.

본 논문은 개인용 컴퓨터에 사용되는 플래시 저장장치인 SSD를 대상으로 한 기존 오프라인 기법들을 개선하여 경량 해시 키인 CRC32 해시 키를 이용해 짧은 유휴 시간 동안 보다 많은 중복 데이터를 효율적으로 제거하는 기법을 제안한다.

2. 관련 기술

2.1 플래시 저장장치에서의 중복제거 기법

플래시 저장장치 내의 중복제거 기법으로는 SSD를 대상으로 한 CAFTL[1]이 있다. CAFTL의 중복제거 프로세스는 FTL의 매핑 테이블을 수정함으로써 수행된다. 중복이 발생하면 그림1과 같이 중복된 데이터의 논리 주소들은 중간 테이블을 통해 하나의 물리 주소를 가리키고 중복된 데이터는 제거된다. CAFTL은 온라인 중복제거와 오프라인 중복제거를 둘 다 지원하고 있으나 온라인 중복제거에 초점을 맞추고 있다.

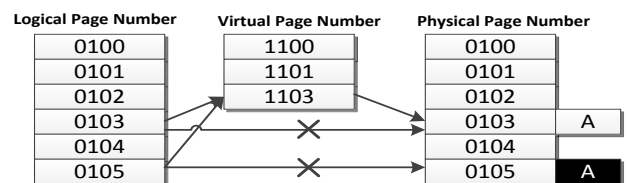


그림 1 중복제거의 예

2.2 온라인 중복제거 기법의 문제점

온라인 중복제거는 런타임 시 수행하는 중복제거 기법이다. CAFTL에서 제안하는 온라인 중복제거 기법은 쓰기 요청이 저장장치에 도착하면 저장장치 내 버퍼에 해당 데이터의 내용을 저장한다. 데이터는 중복제거를 위한 기본 단위인 청크(Chunk)로 분리되고 각 청크의 데이터를 사용하여 충돌이 없는 암호화 알고리즘을 통해 해시(Hash) 키를 생성한다. 이러한 과정을 핑거프린팅(Fingerprinting)이라 한다. 만약 동일한 핑거프린트가 기존의 핑거프린트 테이블에서 발견되면 중복이라 판단하여 해당 청크의 데이터를 기록하지 않고 그림1과 같이 매핑 테이블만을 수정하여 쓰기 동작을 종료한다.

온라인 중복제거 기법은 오프라인 중복제거 기법과는 달리 중복이 발생하면 매핑 테이블만을 수정한다. 따라서 매핑

테이블이 저장장치에 쓰여지지 않은 채로 시스템이 정지하게 되면 파일 시스템은 해당 데이터의 쓰기 동작 완료를 보장받았으나 실제로 저장장치에는 적히지 않은 상태가 되어 데이터 일관성을 깨뜨릴 수 있다. 이런 경우에 일반적인 FTL에서는 물리 페이지의 보조 영역(Spare area)에 매핑된 논리 페이지 주소를 적어 복구 시 사용하지만, 온라인 중복제거에서는 중복된 데이터 자체를 기록하지 않으므로 복구가 불가능하다. 이를 해결하기 위해 매번 매핑 테이블을 저장장치에 기록한다면 중복제거로 인해 감소한 쓰기 양이 다시 늘어나 중복제거 효과가 사라진다. 기존 온라인 중복제거 기법에서는 SuperCap을 사용해 이 문제를 해결하고 있으나 이는 추가적인 비용을 발생시키므로 결론적으로 갑작스러운 시스템 정지에 대한 복구를 고려하면 온라인 중복제거 기법은 적합하지 않다.

2.3 오프라인 중복제거

오프라인 중복제거 기법은 유희시간을 활용하여 중복을 제거하는 기법이다. 우선 쓰기 요청이 오면 핑거프린팅을 거치지 않고 저장장치에 기록한다. 이 후, 읽기/쓰기 요청이 없는 유희시간 동안 기록된 데이터들에 대한 핑거프린팅 및 탐색을 통해 매핑 테이블을 수정하고 중복을 제거한다.

오프라인 중복제거 기법은 온라인 중복제거 기법과는 달리 쓰기 요청에 대한 응답시간을 직접적으로 줄여주지는 않는다. 그러나, 중복제거 프로세스를 통해 무효화된 데이터를 생성시켜 가비지 컬렉션에 대한 비용을 줄임으로써 전체적인 응답 시간을 간접적으로 줄일 수 있다.

오프라인 중복 제거를 위한 블록 구분 기법[2]에서는 중복제거 기법으로 인해 발생하는 무효 페이지들을 한 블록에 모아서 그 블록 안에 존재하는 유효 페이지들의 수를 줄여 페이지 복사 양을 줄인다. 이 기법에서는 CRC32 해시 키를 사용하여 블록을 구분한다. 쓰기 요청시 발생한 청크들의 데이터로 CRC32 키를 생성하고, 생성된 CRC32의 중복 여부를 검사하여 만약 동일한 CRC32 값이 발견되면 중복 가능성이 있는 것이므로 ND (Non-Determined) 블록에 저장한다. 만약 새로운 CRC32 값이라면 해당 데이터는 중복될 가능성이 존재하지 않으므로 CRC32 테이블을 갱신하고 U (Unique) 블록에 저장한다. 이 후, 유희시간 동안에 오프라인 중복제거가 수행되면 ND 블록들에서만 중복제거에 의한 무효 페이지들이 발생하므로 가비지 컬렉션 비용을 줄일 수 있다.

하지만, 유희 시간 동안 모든 페이지에 대해 핑거프린트를 생성하므로 중복 제거에 걸리는 시간이 길다. 이로 인해, 중복이 제거되기 전에 가비지 컬렉션이 발생한다면 중복이 제거되어 곧 무효화될 데이터임에도 불구하고 불필요한 페이지 복사가 발생할 수 있다.

3. 본 론

3.1 경량 해시 키를 이용한 오프라인 중복제거 기법

제안하는 기법에서는 계산량이 많은 핑거프린팅을 하지 않고, 런타임에 경량 해시 키를 이용해 중복 가능성이 존재하는 페이지를 구별해두고 유희시간 중에 이

페이지들만을 대상으로 직접 페이지를 비교하는 중복제거 프로세스를 수행함으로써 전체적인 중복제거 비용을 줄이도록 한다. 본 논문에서는 경량 해시 키로 CRC32 값을 사용하였다.

쓰기 요청이 저장장치에 들어오면 그림2와 같이 런타임에 들어온 데이터를 청킹하고 CRC32값을 생성한다. 이 때 생성된 CRC32 값으로 구분된 블록을 할당 받고 데이터를 저장장치에 기록할 때 물리 페이지 주소를 CRC32 테이블에 저장한다. 물리 페이지 주소는 구분된 블록에 따라 중복될 수 없는 유일(Unique) 페이지인지 중복 가능성이 있는(Non-Determined) 페이지인지로 구분된다. 이 후, 유희시간에 중복 가능성이 있는 페이지의 물리주소를 CRC32테이블에서 참고하여 동일한 CRC32 값을 갖는 유일 페이지들과 비교한다. 동일한 CRC32 값에 저장된 각 4KB의 유일 페이지를 모두 읽어야 하므로 16Byte ~ 20Byte의 핑거프린트를 사용하는 기존 오프라인 중복제거 기법[2]에 비해 페이지 읽기 오버헤드가 발생할 수 있으나, [2]와는 달리 모든 페이지에 대한 핑거프린팅 비용을 절약할 수 있고 동일한 CRC32 값을 공유하는 유일 페이지들만을 비교 대상으로 선택하므로 전체적인 중복제거 비용을 절약할 수 있다. 또한 CRC32 해시는 해시 키를 충분히 다양하게 발생시키므로 하나의 CRC32 키를 공유하는 유일 페이지들의 양은 많지 않다.

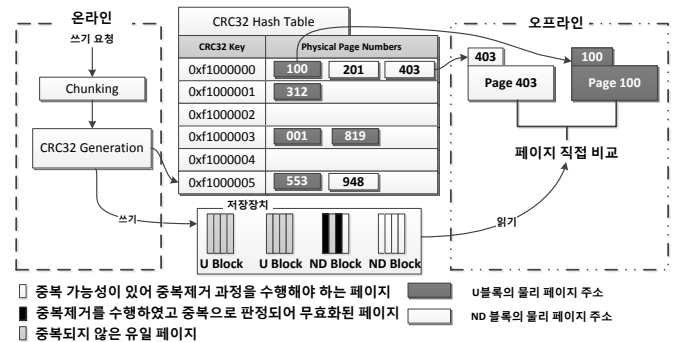


그림 2 경량 해시 키를 이용한 오프라인 중복제거 기법

4. 실험

4.1 실험 환경

본 논문에서는 SSDSim[3]을 수정하여 SSD 내의 중복제거를 시뮬레이션 하였다. 실험에는 두 가지 트레이스를 사용하였다. 첫 째는 WebVM[4]으로 웹 메일 프록시와 온라인 코스 관리에 대한 트레이스를 사용하였고 두 번째 트레이스는 어플리케이션을 버전 별로 설치하면서 발생한 읽기/쓰기 요청의 데이터를 기록한 트레이스를 사용하였다. CRC32와 핑거프린트에 사용한 MD-5의 해싱 시간은 Simple Scalar ARM버전을 이용하여 각각 13μs와 100μs를 사용하였고 SSD의 페이지 읽기, 쓰기, 지우기 연산에 드는 시간은 SSDSim 기본값을 사용하였다.

4.2 쓰기 요청에 대한 응답 시간

그림3은 온라인 기법과 오프라인 기법의 사용자 응답시간에

관련된 요소들을 측정하여 비교한 것이다. 온라인 기법의 총 응답 시간을 기준으로 정규화하여 표현하였다.

본 실험에서 중복제거 기법의 런타임 시 보여질 수 있는 오버헤드 중 비교할 수 있는 오버헤드는 쓰기 시간과 핑거프린팅 시간이다. 중복이 많이 발생하는 상황에서는 온라인 기법에서 절약할 수 있는 쓰기 시간이 증가하여 온라인 기법이 유리하고, 반면에 중복이 적게 발생하는 상황에서는 핑거프린팅 비용의 오버헤드가 발생하여 오프라인 기법의 유리하다. 위 실험에서는 두 트레이스가 각각 30%, 35%의 중복을 가지는데 이러한 상황에서는 그림3에서 보여주는 바와 같이 오프라인 기법이 각각의 트레이스에서 15%, 12%의 응답 시간을 감소시킨다.

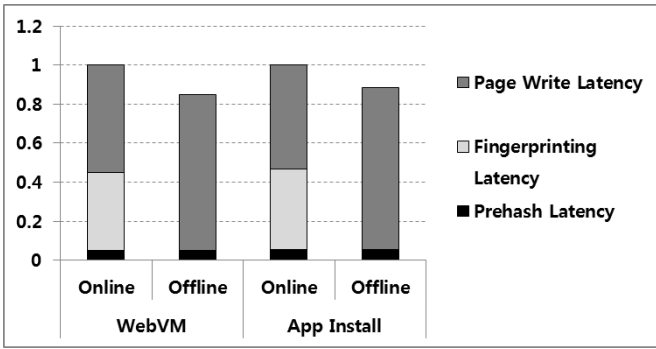


그림 3 쓰기 요청에 대한 응답 시간

4.3 중복제거에 걸린 시간

그림4는 오프라인 시 중복제거 프로세스를 수행하는 데 걸린 시간을 각각 트레이스의 기존 오프라인 중복제거 기법을 기준으로 정규화한 그래프이다. 블록 구분 기법[2]에서는 기존 기법과 동일하게 핑거프린트를 생성하고 블록을 구분하기 위하여 CRC32 해시 키를 사용하므로 CRC32 해싱을 하는 시간이 오버헤드로 발생하여 기존 오프라인 중복제거 기법에 중복제거에 걸린 시간이 길었다. 제안 기법에서는 중복제거 프로세스 당 최소 2개의 페이지를 읽어서 비교해야 하지만, 각 트레이스에서 CRC32로 구분한 중복가능성이 있는 페이지들의 수는 기존 기법들에 비해 절반에 그쳐, 중복제거를 검사한 페이지의 수가 줄어, 전체적인 페이지 읽기 오버헤드는 기존 기법에 비해 크게 늘어나지 않았다. 또한 핑거프린팅 오버헤드가 없어 그림4와 같이 중복제거에 걸린 시간을 70% 이상 절약하였다.

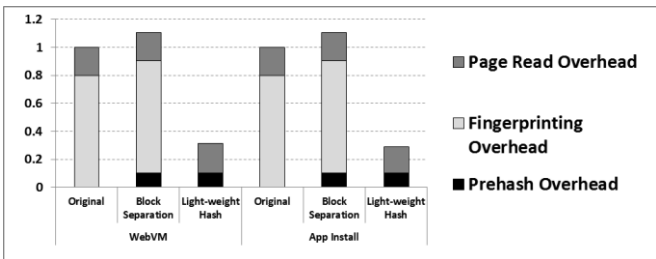


그림 4 오프라인 중복제거에 걸린 시간

4.4 가비지 컬렉션으로 인해 발생한 페이지 이동 횟수

그림5는 시뮬레이션 시 발생한 모든 가비지 컬렉션의 전체 페이지 이동 횟수를 중복된 데이터에 대해 쓰기 동작을 하지 않는 온라인 기법을 기준으로 정규화한 그래프이다. 페이지 이동 횟수는 중복제거가 된 양에 따라, 블록 구분 기법의 적용 여부에 따라 기존 오프라인 기법에 비해 줄어드는 것을 확인하였다. 본 논문의 기법에서는 짧은 유희 시간 동안에 더욱 많은 중복제거 프로세스를 수행할 수 있기 때문에 쓰기 요청에 따른 포어그라운드(Foreground) 가비지 컬렉션이 발생하기 전에 중복제거로 인한 무효 페이지를 많이 생성할 수 있었고, [2]와 같은 블록 구분 기법도 사용하므로 가비지 컬렉션에서 발생하는 페이지 복사 양을 가장 많이 줄일 수 있었다.

Application Install에서는 읽기/쓰기 요청 사이의 간격이 짧아 유희시간이 짧으므로 보다 짧은 시간에 중복제거를 할 수 있는 본 논문의 기법에서 가비지 컬렉션 전에 더욱 많은 중복된 페이지들을 무효화할 수 있어 더욱 큰 가비지 컬렉션 비용 절약 효과를 얻었다.

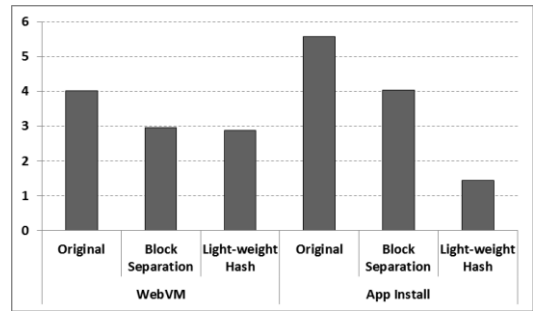


그림 5 오프라인 중복제거 기법들의 페이지 이동

5. 결론

본 논문에서는 오프라인 중복제거를 위한 블록 구분 기법을 개선하여 경량 해시 키를 이용한 오프라인 중복제거 기법을 제안하였다. 기존의 오프라인 중복제거 기법에서 블록 구분에 사용된 CRC32 키를 사용하여 핑거프린팅을 생략하고 최소한의 중복제거 프로세스를 수행함으로써 중복제거 비용을 줄이는 기법이다. 실험 결과, 쓰기 요청 중 보다 많은 무효 페이지를 발생시켜 포어그라운드 가비지 컬렉션의 비용을 기존 기법에 비해 최대 385% 줄일 수 있었다.

참고문헌

- [1] F. Chen 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] N. Agrawal et al., "Design Tradeoffs SSD Performance," USENIX ATC, 2008.
- [4] R. Koller et al., "I/O Deduplication: Utilizing Content Similarity to Improve I/O Performance," FAST, 2010.