

Regulation Pool을 활용한 플래시 메모리 마모평준화 기법

박정수^o 민상렬

서울대학교 컴퓨터공학부

jspark@archi.snu.ac.kr, symin@archi.snu.ac.kr

Flash memory wear-leveling scheme using Regulation Pool

Jeongsu Park^o Sang Lyul Min

School of Computer Science and Engineering, Seoul National University

1. 서론

플래시 메모리는 크기가 작고 진동과 충격에 강하며 성능 및 전력 특성이 우수하기 때문에 다양한 저장 장치 응용분야에서 사용되고 있다. 하지만 플래시 메모리의 구성단위인 블록은 소거횟수의 제약이 있으며, 이를 넘어가게 되는 경우 정상적인 동작을 보장하지 못한다[1]. 따라서 소거연산을 전체 플래시 메모리 영역에 골고루 분산시켜 저장 장치의 수명을 연장시키기 위한 마모평준화 (wear-leveling) 기법들이 연구되어 왔다[2]. 본 논문에서는 Regulation Pool을 활용한 플래시 메모리 마모평준화 기법을 제안한다. 기존의 마모평준화 기법들이 주로 복잡한 쓰레기 수집 (garbage collection) 정책에 의존하거나 모든 블록들의 소거횟수를 유지 관리하기 위해 과도한 비용을 지불해야 하는 것과 달리, 제안하는 기법은 FTL의 메타데이터 저장영역을 Regulation Pool로 활용하여 소거횟수가 낮은 블록에 지우기 및 쓰기 작업을 집중시키는 방식을 통해 간단하면서도 효율적인 마모평준화를 수행할 수 있다.

2. 기존 연구

기존 마모평준화 기법은 쓰레기 수집 동작의 희생블록 선정 정책을 변경하는 방식[3,4], 전체 블록간 소거횟수 차이를 계속 검사하며 최대 차이가 특정 값을 넘어가게 되는 경우, 가장 많이 지워진 블록과 가장 적게 지워진 블록의 내용을 교환하는 방식[5], 소거 동작이 일정 기간 이상 수행되지 않은 블록에 대하여 쓰레기 수집 동작을 수행하는 방식 (static wear-leveling)[6] 등이 있다.

첫 번째 방식은 쓰레기 수집 동작의 희생블록을 선정할 때 블록 내 유효한 페이지의 개수(블록 활용도)[7]만을 고려하는 것이 아니라 블록의 소거 횟수도 함께 고려하여 마모도를 평준화한다. 이러한 접근 방법은 하나의 기법으로 우수한 쓰레기 수집 성능 및 마모도 평준화 효율이라는 상충되는 목표를 동시에 추구하므로 구현이 복잡하고 많은 계산 자원을 소모하는 문제가 있다. 두 번째 방식은, 쓰레기 수집 동작과 독립적으로 전체 공간에 대한 블록 소거횟수의 편차를 계속 감시하며 이 수치가 지정된 값을 넘어서는 순간 마모평준화를 위한 작업이 수행된다. 이때 소거횟수가 높은 블록(hot block)에 포함된 데이터는 참조 확률이 높은 데이터(hot data)이고, 소거횟수가 낮은 블록(cold block)에 저장된 데이터는 참조 확률이 낮은 데이터(cold data)라는 가정에 기반하여 cold 블록의 내용과 hot 블록의 내용 및 사상 정보를 교환한다. 이러한 방식은 블록의 소거횟수와 블록에 소속된 데이터의 참조 확률간 밀접한 연관관계를 활용하지만, 동작이 진행됨에 따라 hot 블록에 cold 데이터가 존재하게 되는 경우도 발생하므로 참조 확률의 정확한 예측에는 한계가 있다. Static wear-leveling 기법은 전체 블록 중 static(cold)데이터가 소속되어 있는 블록을 효과적으로 구분할 수 있는 방안을 제시한다. 이를 통해 판별된 블록들은 특정 조건에 의해 쓰레기 수집 동작의 희생블록으로 선정되며, static 데이터가 점유하고 있던 블록에 hot 데이터가 저장되어 블록간의 소거횟수가 균일하게 유지될 것을 기대한다. 하지만 이 방법 역시 hot 데이터에 대한 구분 방식은 명확하지 않다.

또한 앞서 기술된 방식들은 공통적으로 플래시 메모리의 전체 블록에 대한 소거횟수를 별도의 자료구조로 관리해야 하기 때문에 메모리 사용량이 증가하고, 이를 관리하기 위한 추가적인 비용 부담이 필요하다.

3. Regulation Pool을 활용한 마모평준화 기법 소개

본 논문에서 제안하는 기법은 FTL이 관리하는 플래시 메모리 블록들이 관리 목적에 따라 서로 다른 용도로 사용될 수 있다는 점에 착안한다. FTL은 플래시 메모리 블록을 용도에 따라 사용자로부터 전달되는 데이터를 보관하는 Data 영역, 상태 변화에 관련된 주요 동작이 수행되기 전 복원에 필요한 상태 정보를 미리 순차적으로 기록하는 메타데이터 저장영역, 동작 중 발생할 수 있는 배드 블록을 처리하기 위해 할당된 Reserved 영역 등으로 구분할 수 있다. 이때 각 영역은 용도에 따라 각기 다른 접근 양상을 보인다. Data 영역의 경우 쓰기 및 소거 동작이 수행되는 블록은 사용자의 workload에 의해 결정되기 때문에, FTL은 hot 데이터 혹은 cold 데이터가

속하는 블록의 위치를 예측하기 어렵다. 반면 메타데이터 저장영역의 경우 쓰기 및 소거 동작의 대상 블록은 FTL이 결정한다. 즉 해당 영역에 속한 블록들에 대해 workload를 균일하게 분산시킬 수도 있지만, 경우에 따라서는 특정 블록에 집중시키는 방법을 통해 영역 내에 hot 블록과 cold 블록을 자유롭게 생성하는 것이 가능하다.

본 논문에서는 메타데이터 저장영역과 같이 FTL이 workload 패턴을 자유롭게 변경하여 대상 블록을 선택할 수 있는 영역을 Regulation Pool이라 정의한다. FTL은 Regulation Pool에 소속된 블록들의 소거횟수를 메모리에 보관하고 블록간 소거횟수의 편차를 감시한다. 편차가 일정 수준을 벗어날 경우 소거횟수가 낮은 블록에 대해 연산을 집중하여 소거횟수를 증가시키고, 반대로 소거횟수가 낮은 블록에 대해서는 연산을 방지하여 소거횟수 증가를 막아 Regulation Pool에 포함된 블록들의 소거횟수를 균일하게 유지할 수 있다. 또한 Regulation Pool에 속한 블록들 간에 블록 소거횟수의 평균화가 이루어지면, 이들을 다른 영역에 속한 블록들과 교체하여 전체 블록에 대한 소거횟수 평준화 역시 유도할 수 있다.

Regulation Pool 내부에서 소거횟수의 평준화를 이루는 방식은 블록에 소속된 데이터의 참조 확률을 예측하는 것이 아닌, 데이터의 참조 확률을 직접 조정하는 방식이기 때문에 그 효과가 명확하다. 또한 블록의 소거횟수만을 기준으로 동작이 수행되어 다른 마모 평준화 기법에서 요구되는 복잡한 쓰레기 수거 정책의 적용에 비해 계산 부담이 적다. 마지막으로 블록의 소거횟수를 관리하기 위한 메모리 공간 역시 절약할 수 있다. 메타데이터 저장 공간과 같이 Regulation Pool로 지정이 가능한 영역의 크기는 전체 플래시 메모리 크기에 비해 일반적으로 매우 작기 때문이다.

4. 기법 평가 및 결론

제안하는 기법을 RS(Region Shift)-FTL에 적용하여 마모평준화 효율을 확인해 보았다. RS-FTL은 메모리 등의 자원 사용량이 매우 제한된 환경에서도 효과적으로 동작하는 FTL의 구현을 목표로 개발되었으며, Data 영역, Metadata Logging 영역, Reassembly Buffer 영역, Reserved 영역 등으로 구성된다. RS-FTL은 메모리 사용 양을 줄이기 위해 모든 블록의 소거횟수를 유지하는 대신 전체 플래시 메모리 영역에 가해진 총 소거횟수만을 유지하며, 이 값이 미리 지정된 값을 넘어가면 각 영역의 경계에 위치하고 있는 블록들을 하나씩 서로 교체하는 영역이동을 수행한다. 영역이동에 의해 플래시 메모리에 속한 모든 블록들은 workload의 특성이 다른 영역들을 한번씩 경험할 수 있고, 이를 통해 영역간 특성의 차이가 상쇄되어 모든 블록들의 소거횟수 역시 균일하게 유지되는 것을 기대한다. 하지만 마모평준화를 규칙적인 영역이동에만 의존하기 때문에, 사용자 workload의 주소 접근 패턴이 일부 영역에 편중되거나 동적으로 변화하는 경우에 대해서는 효과적으로 대응하기 어렵다.

RS-FTL은 영역이동 기법을 사용하고 있기 때문에 Regulation Pool에 속한 블록의 교체 과정이 자연스럽게 이루어진다. 따라서 Regulation Pool의 내부 동작만을 마모평준화의 목적에 맞게 수정해 주기만 하면 되며, 이를 위해 Regulation Pool로 지정된 Metadata Logging 영역의 내부 동작을 수정하였다.

시뮬레이션에 기반한 실험을 수행한 결과, Uniform, Hot-cold, LRU 세 가지의 synthetic workload에 대하여 최소한의 크기의 Regulation Pool 설정 만으로도 전체 플래시 메모리의 블록간 소거횟수의 편차가 기법 적용 이전에 비해 약 40% 정도 줄어드는 것을 확인할 수 있었다.

5. 결론

본 기법은 복잡한 쓰레기 수집 정책을 적용하거나 모든 블록의 소거횟수를 유지 관리해야 하는 기존의 마모평준화 기법에 비해 비교적 단순한 연산 및 적은 자원 사용으로 마모평준화를 수행할 수 있는 방안을 제시하였다. 본 논문에서는 RS-FTL에 대한 적용 결과만을 확인하여 보았으나, 이러한 기법이 일반적으로 사용되기 위해서는 Regulation Pool에 소속된 블록을 다른 영역에 속한 블록들과 동적으로 교체하는 방식이 다른 FTL에서도 효율적으로 수행되어야 한다. 향후 다양한 FTL에 본 기법을 적용하여 기법의 효용성을 검증하는 연구가 진행될 예정이다.

참고 문헌

- [1] S. K. Lee, S. L. Min, Y. K. Cho, "Current trends on flash memory technology," *Journal of KIISE*, vol.24, no.12, pp.99-106, Dec. 2006. (in Korean)
- [2] E. Gal, S. Toledo, "Algorithms and data structures for flash memories," *ACM Computing Surveys*, vol.37, no.2, pp. 138-163, Jun. 2005.
- [3] H.-J. Kim, S.-G. Lee, "A new flash memory management for flash storage system," *Proc. of the 23rd International Computer Software and Applications Conference*, pp. 284-289, 1999.
- [4] M.-L. Chiang, P. C. H. Lee, R.-C. Chang, "Managing flash memory in personal communication devices," *Proc. of the 1997 IEEE International Symposium on Consumer Electronics(ISCE'97)*, pp.177-182, 1997.
- [5] L.-P. Chang, "On efficient wear-leveling for large-scale flash-memory storage systems," *Proc. of the 2007 ACM symposium on Applied computing*, pp.1126-1130, 2007.
- [6] Y.-H. Chang, J.-W. Hsieh, T.-W. Kuo, "Improving flash wear-leveling by proactively moving static data," *IEEE Transactions on Computers*, vol.59, no.1, pp.53-65, Jan. 2010.
- [7] A. Kawaguchi, S. Nishioka, H. Motoda, "A flash-memory based file system," *Proc. of the USENIX 1995 Technical Conference*, 1995.