플래시 메모리 상에서의 비용 예측 모델

노홍찬° 유진희 박상현
연세대학교 컴퓨터과학과
{fallsmal°, jhyou, sanghyun}@cs.yonsei.ac.kr

A Cost Estimation model for the flash memory storage system

Hongchan Roh° Jinhee you Sanghyun Park
Department of Computer Science, Yonsei University

요 약

최근 모바일 환경에서 우수한 특성으로 인해 플래시 메모리가 하드 디스크를 대체할 만한 차세대 저장 장치로서 주목을 받고 있다. 하지만 이러한 플래시 메모리는 읽기 속도에 비해 쓰기 속도가 느리고 접근 비용의 비중이 미미한 특징 때문에 하드 디스크 기반에서의 디스크 접근 횟수를 이용한 비용 예측 방법을 그대로 적용할 수 없다. 그러므로 플래시 메모리 기반의 효율적인 인덱스 설계나 질의 처리, 최적화를 위해서는 플래시 메모리에 적합한 비용 예측 방법이 필요하다. 본 논문의 목적은 플래시 메모리를 위한 파일 시스템인 플래시 전환 계층(Flash Translation Layer)을 기반으로 비용 예측 모델을 제시하는 것이다. 플래시 메모리의 저장 공간에서 데이터를 읽는 비용은 플래시 메모리의 하드웨어 특성으로부터 쉽게 유추될 수 있지만, 쓰기 비용의 경우 플래시 메모리에 대한 쓰기 작업이 초래하는 가비지 컬렉션(Garbage Collection) 때문에 예측하기가 힘들다. 본 논문은 해당 파일 시스템으로부터 전체 플래시 메모리 공간 대비 유효 데이터의 사용률을 얻어낸 후 그 정보를 이용하여 가비지 컬렉션을 포함한 쓰기 비용을 예측하는 모델을 제안한다. 이러한 예측 모델을 사용하면 인덱스를 구성하거나 질의 처리 및 최적화 알고리즘을 구성하는데 있어 플래시 메모리의 특성을 반영한 비용 효율적인 설계를 수행할 수 있다.

1. 서 론

플래시 메모리는 하드 디스크처럼 비휘발성의 특징을 가지면서도 하드 디스크보다 충격에 강하고 전력 손실이 작아 최근 모바일 환경에서 많이 사용되고 있으며 하드 디스크를 대체할만한 차세대 저장 장치로서 주목받고 있다.

하지만 플래시 메모리는 다음과 같은 중요한 단점들을 가지고 있다: 1) 느린 쓰기 속도, 2) 불가능한 페이지 단위 덮어쓰기, 3) 짧은 수명. 플래시 메모리는 다수의 블록(block)들로 구성되며 하나의 블록은 다시 다수의 페이지(page)들로 구성된다. 페이지 단위의 쓰기 작업은 페이지 단위의 읽기 작업에 비해 매우 느리게 수행되며 이를 표 1에서 확인할 수 있다. 다른 단점으로 플래시 메모리는 이미 쓰여진 페이지에 다른 데이터를 덮어 쓰기 위해서 그 페이지가 속하는 블록을 지우고 난 후 써야 한다. 그러므로 본 논문은 한 페이지를 다시 쓰기 위해 블록 전체를 지우기보다는 갱신된 데이터를 다른 블록의 빈 페이지에 쓰는 방법을 제안한다. 마지막으로 플래시 메모리는 한 블록을 지울 수 있는 횟수가 만 번에서 백만 번 사이로 제한되어 있다[1]. 특정 블록의 지우기 횟수가 제한된 횟수를 초과하게 되면 해당 블록에 데이터를 저장할 때 오류가 발생하여 더 이상 해당 블록을 사용할 수 없게 된다.

이러한 플래시 메모리의 단점들을 보완하기 위해 플래시 메모리를 위한 파일 시스템들과 그에서 파생된 많은 연구들이 제안되어 왔다. 플래시 메모리에서 동작하는 파일 시스템은 크게 두 가지로 분류된다. 플래시 메모리 특화 파일 시스템(Flash Specific File System) 방법과 블록 장치 모방(Block Device Emulation Technique) 방법이다[1].

블록 장치 모방 방법은 기존의 디스크 기반 파일 시스템을 에뮬레이트(emulate)하는 계층을 기존 파일 시스템과 플래시 메모리 사이에 추가함으로써 기존의 파일 시스템에서 사용되었던 상위 어플리케이션 프로그램들을 변경 없이 그대로 적용시킬 수 있는 장점이 있으며 플래시 전환 계층(Flash Translation Layer)[2], 컴팩트 플래시(CompactFlash)[3], 스마트 미디어 (SmartMedia)[4] 등이 제안되었다.

플래시 메모리 특화 파일 시스템은 블록 장치 모방 방법과 달리 기존 파일 시스템을 대체하는 플래시 메모리를 위한 새로운 파일 시스템으로서 기존의 로그 구조 기반 파일 시스템[5]의 아이디어를 기반으로 설계되었으며 블록 장치 모방 방법보다 좋은 쓰기

표 1. 1기가바이트 큰 블록 낸드 플래시 메모리 특징[1)]

페이지 크기	블록 크기	페이지 읽기 시간	페이지 쓰기 시간	블록 지우기 시간
2kbyte	128kbyte	20μs	200μs	1.5ms

1) http://www.samsung.com/Products/Semiconductor/common/product_list.aspx?family_cd=NFL0201

성능을 보인다. 플래시 메모리 특화 파일 시스템의 대표적인 종류로는 JFFS2[6], LFM[7], YAFFS[8] 등이 있다.

이러한 파일 시스템 위에서 동작하는 다양한 어플리케이션 프로그램(Application Program)들을 설계하기 위해서는 플래시 메모리의 특성을 반영한 비용 분석 모델이 필요하다. 기존의 하드 디스크 기반에서는 하드 디스크에 접근하는 시간, 즉 목표하는 실린더(Cylinder)에 디스크 헤더(Header)를 위치시킬 때까지의 시간이 디스크를 읽기, 쓰기 하는데 있어 큰 비중을 차지했기 때문에 하드 디스크에 대한 접근 횟수를 하드 디스크 기반 비용 예측 모델로 이용하였다.

하지만 앞에서 설명한 바와 같이 플래시 메모리 상에서는 플래시 메모리 페이지에 대한 쓰기 작업이 읽기 작업에 비해 매우 느리게 수행되며, 그 과정 중에 해당 페이지를 접근 하는 비용은 미미하다. 그러므로 하드 디스크와는 다른 비용 예측 모델이 필요하다.

본 논문은 플래시 전환 계층 파일 시스템을 기반으로 플래시 메모리에 대한 읽기 쓰기 작업의 비용 예측 모델을 제시한다. 이를 위해 단위 읽기 작업 및 쓰기 작업에 대한 비용 계산법을 제시한다. 특히 페이지 쓰기 비용의 경우 페이지 쓰기로 인해 추가로 발생되는 가비지 컬렉션(Garbage Collection)의 비용도 환산하여 함께 계산한다.

가비지 컬렉션은 기존에 제안된 다양한 블록 재생 정책(block-recycling policy)에 따라 발생 시점을 예측하기 어렵기 때문에 이러한 불확실성을 해결할 수 있는 방법이 필요하다.

이를 위해 본 논문은 가와구치(Kawaguchi) 등[9]이 제안한 비용-이득 정책(cost-benefit policy)[9]을 기반으로 가비지 컬렉션의 비용을 예측하여 단위 쓰기 비용에 포함시키는 방법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 가비지 컬렉션에 대해 좀 더 자세히 알아보고 3장에서 본 연구가 제안하는 비용 예측 모델을 제시한다. 마지막으로 4장에서 본 논문을 결론짓는다.

2. 플래시 전환 계층상의 가비지 컬렉션

일반적으로 알려진 낸드 플래시 메모리의 문제점은 페이지 덮어쓰기의 불가능이다. 즉, 낸드 플래시 메모리의 읽기, 쓰기 작업의 단위는 페이지이고 지우기의 단위는 블록이므로 페이지 단위의 데이터 수정이 발생할 때 해당 페이지만 덮어쓰지 못하고 전체 블록을 지워야만 해당 페이지의 데이터를 수정할 수 있다.

이러한 문제점을 해결하기 위해 플래시 전환 계층은 최대한 지우는 작업을 뒤로 미루고 수정이 필요한 페이지에 대해서는 해당 페이지의 내용을 아직 쓰여 지지 않은 다른 페이지에 복사한다. 이러한 플래시 전환 계층의 수정 작업을 외부 공간 갱신이라 명명한다. 외부 공간 갱신에 의해 아직 쓰여 지지 않은 페이지 즉, 플래시 메모리가 동작 후 한 번도 쓰여 지지 않았거나 해당 페이지에 대해 지우기 작업이 수행된 후 쓰여 지지 않은 페이지를 프리 페이지(free page)라 명명한다. 또한 외부공간갱신이 수행된 후 올바르지 못한 데이터를 가지고 있는 기존의 페이지를 데드 페이지(dead page)라 이름 붙이고 기존 페이지의 데이터가 복사된 페이지는 라이브 페이지(live page)라고 명명한다.

시간이 지나면 데드 페이지가 플래시 메모리에 계속 쌓이게 되고 수정된 데이터를 복사할 프리 페이지가 남지 않게 된다. 이렇게 되면 플래시 메모리의 데이터에 대한 더 이상의 수정이나 추가가 불가능하다. 그러므로 이러한 데드 페이지를 프리 페이지로 바꿔줄 수 있는 작업이 필요하다.

가비지컬렉션(garbage collection)은 특정 블록을 선택하고 그 블록을 지움으로써 프리 페이지를 생산해내는 작업이다. 여기서 중요한 것은 블록 선택의 기준이다. 어떤 블록을 선택해서 그 블록을 지우고 블록 내의 페이지들을 프리 페이지로 생성해낼 것인가 하는 문제이다. 이러한 블록 선택의 기준을 블록 재생 정책이라고 한다[1].

앞에서 언급한 바와 같이 블록 재생 정책에 관한 연구로서 가와구치 등은 비용-이득 정책을 제안하였다. 이는 해당 블록의 데드 페이지의 개수와 최근 사용 후 지난 시간을 기준으로 휴리스틱(heuristic) 함수를 정의하고 해당 함수의 값이 가장 큰 블록을 선택하여 가비지컬렉션을 수행하는 하는 방법이다.

$$\frac{a(1-u)}{u} \quad (1)$$

비용-이득 정책은 식 (1)의 값이 큰 블록을 선택하여 해당 블록을 지우고 프리 페이지를 생산 해내는 그리디 방식(greedy approach)이다.

식 (1)에서 a는 각 블록의 나이(age)를 나타내는 값으로서 가장 최근에 블록이 업데이트(update)된 후로 시간이 얼마나 지났는지를 나타내는 변수이고 u는 해당 블록의 전체 페이지 개수 대비 라이브 페이지의 비율을 나타내는 것이다. 이 식에 의해 시스템은 라이브 페이지가 가장 적은(데드 페이지가 가장 많은) 블록을 선택하는 것을 가장 높은 우선수위로, 얼마나 오래 전에 사용되었는지를 그 다음 우선순위로 하여 가비지 컬렉션을 수행한다.

이렇게 그리디 함수에 기반을 두어 블록 재생을 수행하는 시점은 플래시 메모리 상의 전체 프리 페이지의 개수(ϕ)가 특정 임계 값(T) 이하로 내려갔을 때이다. 즉, 플래시 메모리에 대한 업데이트(update)나 쓰기(write) 작업이 수행되었을 때 T 이하로 ϕ가 내려가게 되면 가비지 컬렉션이 수행되는 것이다.

이렇듯 가비지 컬렉션은 플래시 메모리에 대한 쓰기 작업에 의해 초래 되므로 가비지 컬렉션이 수행되는 동안 시스템이 멈추게 되어 쓰기 작업에 대한 시간 비용을 증가시키게 된다.

가비지 컬렉션은 여러 요인에 의해 특정 시점에 수행되므로 그 실행 시점을 예측하기가 어려우며 소모비용도 일정치 않아 이것이 다음 장에서 설명할 플래시 전환 계층상에서 읽기, 쓰기 작업에 대한 비용 예측 모델을 구축하는데 있어 어려움으로 작용한다.

3. 플래시 전환 계층상의 비용 예측 모델

서론에서 언급하였듯이 플래시 메모리의 경우 하드 디스크와 달리 단위 페이지 읽기 비용이나 페이지 쓰기 비용에 있어서 접근 비용은 미미하다. 하드 디스크가 전동 모터(motor), 헤더와 같은 기계적인 장치를 사용하여 특정 데이터가 위치한 실린더를 헤더를 움직여서 기계적으로 찾아가는데 비해 플래시 메모리는 기계 장치의 도움 없이 메인 메모리와 유사하게 특정 주소로 큰 비용 없이 직접 접근이 가능하다.

하드 디스크 기반에서는 하드 디스크를 읽고 쓰는데 있어서 이러한 접근 비용이 다른 비용들(데이터 전송 비용, 회전 지연 비용)에 비해 월등히 크므로 디스크에 대한 접근 횟수를 비용 분석의 기준으로 사용한다.

하지만 플래시 메모리는 접근 비용이 미미하므로 접근 횟수가 이러한 비용 예측에 기준이 될 수 없고 실제 읽기 비용 및 쓰기 비용을 좀 더 자세하게 따져봐야 한다.

읽기 비용의 경우 간단하게 해당 플래시 메모리의 하드웨어적인 특성으로 간단히 예측이 가능하다. 플래시 메모리에 대한 읽기 작업이 초래하거나 병행하여 수행되는 다른 작업이 없으므로 플래시 메모리 자체의 하드웨어적인 특성인 페이지 읽기 속도를 비용 예측 모델에 그대로 사용할 수 있다.

즉, 특정 프로세스가 N개의 페이지를 읽는 다면 전체 비용은 해당 플래시 메모리의 단위 페이지 읽기 시간(P_r)에 N을 곱한 값이 된다. 표 1에 근거하여 1기가 낸드 플래시 메모리에 이를 수행할 경우, P_r은 20μs가 되며 전체 비용은 N * 20μs가 되는 것이다. 그러므로 플래시 메모리에서의 비용 예측을 위한 페이지 단위 읽기 비용(C_r)은 P_r이 된다.

$$C_r = P_r \tag{2}$$

반면, 쓰기 비용의 경우는 그리 간단하지가 않다. 플래시 메모리에서의 비용 예측을 위한 페이지 단위 쓰기 비용(C_w)을 계산하기 위해서는 기본적으로 해당 플래시 메모리의 단위 페이지 쓰기 시간(P_w)을 쓰기 비용으로 포함해야 하고 그 외에 가비지 컬렉션이 수행될 경우에 추가되는 비용도 포함시켜야만 한다. 표 1의 경우 P_w는 200μs로 P_r에 비해 10배의 시간 비용을 가진다.

가비지 컬렉션은 일반적으로 여러 개의 페이지 쓰기 작업이 이루어진 이후에 전체 프리 페이지 개수(ϕ)가 가비지 컬렉션을 위한 임계값(T) 이하로 내려간 경우에 수행된다. 그러므로 복수 개의 페이지 쓰기 작업 후에 한 번의 가비지 컬렉션 작업이 수행되게 되므로 한 번의 페이지 쓰기 작업에 대한 가비지 컬렉션 코스트는 가비지 컬렉션 코스트를 수행된 쓰기 작업 횟수로 나눈 값으로 환산해야 한다.

각 페이지 단위 쓰기 비용으로 환산할 가비지 컬렉션 비용(G_u)을 계산하기 위해서는 가비지 컬렉션의 다음 실행 주기를 예측할 수 있어야 한다. 이를 위해 일단 시스템의 프리 페이지 개수가 T이하로 떨어져서 가비지 컬렉션이 수행된 상황을 가정해본다. 이제 가비지 컬렉션이 다음에 수행될 시점은 현재 가비지 컬렉션이 수행되어 생산해낸 프리 페이지 개수를 다 소모하고 난 이후이다. 그러므로 생산된 프리 페이지들의 개수만큼의 페이지 쓰기 작업들이 수행되면 블록 재생 정책에 의해 다음 가비지 컬렉션 작업이 초래 된다.

위의 가정을 이용하면 현재 플래시 메모리의 상태로부터 다음 가비지 컬렉션의 시점과 그동안 수행될 페이지 쓰기 작업의 개수를 알아낼 수 있고 그러한 페이지 쓰기 작업들에 전체 가비지 컬렉션의 비용(G_t)을 분할하여 환산할 수 있다. 즉, 단위 가비지 컬렉션 비용(G_u)은 G_t를 해당 가비지 컬렉션 작업으로부터 생산된 프리 페이지의 개수(F_G)로 나누어주면 되며 다음의 식으로 정리될 수 있다.

$$G_{u,i+1} = \frac{G_{t,i}}{F_{G_i}} \tag{3}$$

위의 식이 의미하는 바는 다음번에 수행될 단위 가비지 컬렉션 비용($G_{u,i+1}$)은 현재의 가비지 컬렉션의 전체 비용($G_{t,i}$)을 현재의 가비지 컬렉션에서 생산한 프리 페이지의 개수(F_{Gi})로 나눔으로써 계산된다는 것이다.

식 (3)으로 표현된 $G_{u,i+1}$로 부터 앞에서 언급한 페이지 단위 쓰기 비용(C_w)을 P_w와 $G_{u,i+1}$의 합으로 나타낼 수 있다.

$$C_w = P_w + G_{u,i+1} \tag{4}$$

이와 같이 복수의 페이지 쓰기 작업 당 수행되는 가비지 컬렉션 비용을 각 페이지 쓰기 작업 별로 분할 상환하는 것은 분할상환 분석법(Amortized analysis)[10] 차원에서 이해될 수 있다. 임의의 N개의 페이지 쓰기 비용을 산출하기 위해서 예측하기 힘들게 수행되는 가비지 컬렉션에 대한 비용을 개별 페이지 쓰기 비용에 분할하여 환산함으로써 각 페이지 쓰기가 잠재적으로 일으키는 가비지 컬렉션에 대해서도 비용을 책임지도록 하는 것이다. 이는 분할상환 분석법 중 계산법(Accounting method)[10] 에 해당한다.

C_w에 대한 비용 분석을 마치기 위해서는 전체 가비지 컬렉션 과정에 드는 비용(G_t)과 가비지 컬렉션 작업으로부터 생산되는 프리 페이지의 개수(F_G)가 필요하다.

전체 가비지 컬렉션 과정에서의 비용(G_t)은 두 가지로 나누어 생각해볼 수 있다. 먼저, 첫 번째 비용은 블록 재생 정책에 의해 선택된 페이지에서 라이브 페이지들을 골라내어 이들을 다른 블록의 프리 페이지들에 복사하는 비용(G_c)이고 그 다음은 라이브 페이지들을 복사한 후 해당 블록을 삭제하는 비용(G_d)이다. G_d는 해당 플래시 메모리의 블록 삭제 비용(B_d)을 그대로 따른다. 예를 들어 표1의 경우 B_d는 1.5ms이다. G_c의 경우 해당 블록의 라이브 페이지들을 읽어서 그 개수만큼의 다른 프리 페이지들에 해당 내용을 써야 하므로 단위 페이지 읽기 비용(P_r)과 단위 페이지 쓰기 비용(P_w) 둘을 합한 것에 라이

브 페이지개수(L) 곱한 비용이 된다. 이를 종합하여 G_t는 다음 식으로 정리될 수 있다.

$$G_t = B_d + L(P_r + P_w) \quad (5)$$

가비지 컬렉션 작업으로부터 생산되는 프리 페이지의 개수(F_G)는 한 블록을 구성하는 페이지의 개수(π)로부터 그 블록에 존재하는 라이브 페이지의 개수(L)과 기존에 그 블록에 존재했던 프리 페이지의 개수(F)를 뺀 값이 된다. 이는 한 블록을 지우게 되면 그 블록을 구성하는 모든 페이지들이 프리 페이지로 생산이 되지만 라이브 페이지의 복사를 위해서 L 만큼의 프리 페이지가 다시 소모되어야 하고, 삭제할 블록에 존재했던 프리 페이지의 경우 새로 생산된 것이 아니므로 제외시켜야 하는 이유에서다. π는 각 플래시 메모리마다 고유하게 정해져 있으므로 그 값을 사용하면 된다. 예를 들어 표1의 경우 한 블록 128kbyte이고 한 페이지가 2kbyte이므로 π는 64개이다. 또한 F의 경우, 가비지 컬렉션이 수행되는 시점이 전체 플래시 메모리에 프리 페이지의 개수가 T 아래로 내려갈 때이므로 임의의 선택된 블록에서 가질 수 있는 F의 기대 값은 평균적으로 식 (6)과 같다.

$$F = T/\text{전체 블록의 개수} \quad (6)$$

이는 보통 매우 작은 값이 되므로 무시할 수 있다. 예를 들어 표1의 경우 전체 블록의 개수는 전체 플래시 메모리의 용량이 1GB이고 한 블록의 크기는 128kbyte 이므로 8000개가 된다. 이때 T의 값이 8000개 미만일 경우 F 값은 1에도 못 미치게 된다. 그러므로 가비지 컬렉션 작업으로부터 생산되는 프리 페이지의 개수(F_G)는 다음 식으로 정리될 수 있다.

$$F_G = \pi - L \quad (7)$$

이제까지 단위 쓰기 비용(C_w) 분석을 위해 몇 가지 식들을 세워보았고 이들을 종합하여 식 (4)에 식 (3), (5), (6), (7)을 연립하면 C_w에 대한 다음 식이 완성된다.

$$C_w = P_w + \frac{B_d + L(P_r + P_w)}{\pi - L} \quad (8)$$

C_w를 표현한 식 (8)에서 P_r, P_w, B_d, π 모두는 표1과 같은 해당 플래시 메모리 고유의 하드웨어적인 특성으로 플래시 메모리 종류에 따라 고정된 값이다. C_w 산출을 위해 유일하게 도출되어야 하는 값은 L이다. L은 블록 재생 정책에 의해 선택된 블록의 라이브 페이지의 개수로 어느 블록이 선택되느냐에 따라 변하기 때문에 쉽게 예측하기 힘들다.

하지만 전체 플래시 메모리의 사용률(O) 즉, 해당 플래시 메모리의 전체 페이지 개수 대비 라이브 페이지의 비율을 안다면 C_w에 대한 최악의 경우 분석(worst case analysis)을 수행할 수 있다. 블록 재생 정책이 전체 블록들 중 가장 L이 작은 블록을 선택하게 되므로 최악의 경우 L의 기댓값은 $\pi * O$와 같다. 최악의 경우 즉, 선택된 블록의 L 값이 가장 크게 되는 경우는 시스템의 라이브 페이지들이 모든 블록에 정확하게 균일하게 분포할 경우이며 이와 같은 경우 전체 시스템의 라이브 페이지의 비율(O)과 임의의 블록의 라이브 페이지의 비율이 같게 된다. 그러므로 최악의 경우의 L 의 기댓값은 $\pi * O$와 같게 된다. 이를 이용하여 식 (8)을 다시 정리하면 최악의 경우의 C_w는 다음과 같이 정리될 수 있다.

$$C_w = P_w + \frac{B_d + l(P_r + P_w)}{\pi(1 - O)} \quad (9)$$

이에 본 논문은 O를 플래시 메모리 전환 계층에서 관리하여 상위 어플리케이션 프로그램에게 시스템 콜(system call)등의 형태로 제공할 것을 제안한다. 플래시 메모리 전환 계층은 블록 재생 정책을 위해서 이미 전체 시스템 내의 프리 페이지의 개수(ϕ)를 관리하고 있으므로 전체 라이브 페이지의 개수를 관리하는 것은 큰 부하가 아니다. O를 어플리케이션 계층에 제공하면 어플리케이션은 직접 블록 재생 정책에 의해 어떤 블록이 선택되어 해당 블록의 L 값이 얼마인지 알아내거나 그에 대한 예측을 하지 않고서도 C_w를 계산할 수 있게 된다.

지금까지 페이지 단위 읽기 비용(C_r)과 페이지 단위 쓰기 비용(C_w)을 분석하여 보았다. 플래시 메모리 기반의 어플리케이션들은 하드 디스크 기반과는 달리 디스크 접근 횟수만으로 그 비용을 예측하는 것이 아니라 해당 어플리케이션의 페이지 읽기 횟수(N_r)와 페이지 쓰기 횟수(N_w)를 예상하여 다음과 같이 최악의 경우 전체 비용을 예측할 수 있다.

$$N_r C_r + N_w C_w \quad (9)$$

4. 결론

본 논문에서는 플래시 메모리 상에서 동작하는 어플리케이션 프로그램의 비용 예측 모델에 대해 제시하였다. 플래시 메모리 전환 계층과 비용-이득 블록 재생 정책에 기반을 두어 전체 비용을 페이지 단위 읽기 비용과 페이지 단위 쓰기 비용으로 나누어 접근하였다. 또한 전체 플래시 메모리의 사용률을 이용하여 최악의 경우 분석이 가능한 비용 예측 모델을 제안하였다.

이러한 비용 예측 모델은 플래시 메모리 상에서 동작하는 데이터베이스 시스템 등의 상위 어플리케이션 프로그램들을 개발하는 데 있어, 플래시 메모리에서 비용 효율적인 새로운 알고리즘 및 데이터 구조를 설계할 수 있는 기반이 될 것이다.

차후에는 다른 플래시 메모리 파일 시스템 환경을 기반으로 하는 비용 예측에 대한 연구도 진행 하려고 한다. 궁극적으로는 다양한 플래시 메모리 기반 파일 시스템에 일반적으로 적용할 수 있도록 현재의 연구를 확장하고자 한다.

참고 문헌

[1] E. Gal, and S. Toledo, "Algorithms and Data Structures for Flash Memories,"ACM Computing Surveys, pp. 138-163, 2005.
[2] Intel Corporation, "Understanding the Flash Translation Layer(FTL) Specification".
[3] Compact Flash Association, "CompactFlashTM 1.4 Specification," 1998.
[4] SSFDC Forum, "SmartMediaTM Specification,"1999.
[5] M. Rosenblum, and J. K. Ousterhout, "The Design and Implementation of a Log-Structured File System," ACM Transactions on Computer Systems, Vol.10, No.1, pp. 26-52, 1992.
[6] D. Woodhouse, Red Hat, Inc., "JFFS: The Journalling Flash File System".
[7] Intel Corporation, "LFS File Manager Software: LFM".
[8] Aleph One Company, "Yet Another Flash Filing System".
[9] A. Kawaguchi, S. Nishioka, and H. Motoda, "A Flash-Memory Based File System," USENIX Technical Conference on Unix and Advanced Computing Systems, 1995.
[10] M.T. Goodrich, and R. Tamassia, Algorithm Design, John Wiley & Sons, pp. 34-42.