

## 차세대 비휘발성 메모리가 추가된 버퍼캐쉬에서 성능 측정 방법의 재조명\*

이규형<sup>○</sup> 최중무 이동희 노삼혁  
홍익대학교 컴퓨터 공학과<sup>○</sup> 단국대학교 컴퓨터 공학과 서울시립대학교 컴퓨터 공학과 홍익대학교 컴퓨터 공학과  
lkh@cs.hongik.ac.kr<sup>○</sup> choijm@dankook.ac.kr dhl@venus.uos.ac.kr samhnoh@hongik.ac.kr

### Reconsidering Performance Measurement when Non-Volatile RAM is used in the Buffer Cache

Kyuhyung Lee<sup>○</sup> Jongmoo Choi Donghee Lee SamH. Noh  
Hongik University Dankook University University of Seoul Hongik University

#### 요 약

영속적인 데이터 저장이 가능한 차세대 비휘발성 메모리를 휘발성 메모리와 혼용하여 버퍼캐쉬로 사용하면, 안정성과 성능향상의 효과를 얻을 수 있다. 본 연구에서는 기존의 연구에서 제시한 캐쉬관리 정책을 시뮬레이터를 이용하여 실험하고 실험 결과를 분석하여 비휘발성 메모리가 추가된 캐쉬의 새로운 특성을 밝혀냈다. 비휘발성 메모리가 캐쉬에 포함되면 읽기, 쓰기의 요청의 종류, 미스(miss)되었을 경우 캐싱될 블록의 더티(dirty)여부, 읽기 요청이 적중(hit)되었을 때, 적중된 블록의 메모리 종류에 따라 각각의 요청을 처리하기 위한 디스크 접근횟수가 달라지는 특성을 나타낸다. 이 특성 때문에 비휘발성 메모리가 추가된 버퍼캐쉬는 적중률(hit rate) 보다는 디스크 접근횟수를 측정하는 것이 정확한 성능측정을 가능하게 한다.

#### 1. 서 론

최근 삼성, IBM, Motorola등이 차세대 비휘발성 메모리를 경쟁적으로 출시 하고 있다. DRAM에 버금가는 접근 속도와 데이터 저장의 영속성을 가지는 차세대 비휘발성 메모리를 대표하는 FRAM (Ferro-electric RAM), MRAM(Magneto-resistive RAM), PRAM(Phase-change RAM)은 현재 DRAM으로 대표되는 컴퓨터 시스템의 주기억장치를 전부 혹은 일부 대체할 수 있으므로 전체 시스템의 성능향상을 기대할 수 있다. 지금까지의 일반적인 컴퓨터 시스템은 주기억장치로 속도가 빠르나 영속적인 데이터 저장이 불가능한 휘발성 메모리(DRAM 등)를 사용하고, 영속적인 저장공간을 위해 속도가 느린 디스크 혹은 플래쉬 메모리를 사용하고 있다. 즉, 대부분의 시스템에서 프로그램 수행을 위한 일시적인 저장공간과 데이터의 저장을 위한 공간이 분리되어 사용되고 있다. 그러나 비휘발성 메모리를 주기억장치로 사용할 수 있게 됨으로써, 이 구분은 없어진다. 데이터의 처리를 위해 주기억장치(비휘발성 메모리)에 저장된 데이터를 디스크 혹은 플래쉬 메모리로 옮겨야 할 필요가 없어지기 때문에 성능향상을 이룰 수 있다.

전원공급이 없이도 영속적으로 데이터를 저장할 수 있는 비휘발성 메모리를 이용하여 시스템의 성능을 향상시키려는 연구는 여러가지 분야에서 이루어져 왔다 [1,2,3,4,5,6,7,8]. 특히 비휘발성 메모리를 버퍼 캐쉬로 사용하여 시스템의 성능을 향상시키는 방법은 작은 크기의 비휘발성 메모리 만으로도 휘발성 메모리와 함께 사용할 때 안정성과 함께 큰 성능향상을 얻을 수 있다 [6,7,8]. 비휘발성 메모리를 쓰기 캐쉬로 사용하게 되면, 비휘발성 메모리 공간의 더티블록을 디스크로 써야 하는 과정이 필요 없게 되므로 시스템의 성능을 향상시킬 수 있다. 위의 특성으로 인하여 휘발성 메모리만으로 구성된 버퍼캐쉬와는 다른 특성을 나타낸다.

본 논문은 [6]의 논문에서 제시한 정책을 시뮬레이터를 이용한 실험을 하였는데, [6]의 논문에서는 실험 결과에 대한 분

석의 부족하였기 때문에 이번 연구에서는 실험결과를 비휘발성 메모리의 특성을 고려하여 분석한 결과 비휘발성 메모리가 포함된 버퍼캐쉬의 새로운 특성을 밝혀냈다. TPC-C 트레이스를 이용한 실험 결과 버퍼 캐쉬에서의 적중률(hit rate)이 성능에 직접적으로 영향을 미치던 이전까지와는 달리 휘발성 메모리와 비휘발성 메모리를 함께 사용한 캐쉬에서는 적중률이 낮은 정책이 오히려 적중률이 높은 정책보다 좋은 성능을 낼 수도 있음을 확인하였다. 그 원인은 비휘발성 메모리의 특성상 읽기와 쓰기요청이 버퍼캐쉬에서 적중(hit) 또는 미스(miss)될 때 서로 다른 디스크 접근 횟수를 가지기 때문이다. 이러한 특성들을 종합해 볼 때, 비휘발성 메모리를 사용하는 버퍼 캐쉬의 성능측정은 적중률로 측정을 하는 것 보다는 실제로 디스크 접근이 몇 번이나 이루어 지는지를 측정해야 정확한 성능 측정을 할 수 있음을 알 수 있다. 휘발성 메모리만으로 이루어진 버퍼캐쉬의 성능을 향상시키기 위해서 단지 적중률을 높이면 그것이 성능향상과 직결되므로 버퍼캐쉬에 관한 연구는 대부분 적중률 증가에 초점이 맞추어져 있었다. 하지만, 버퍼캐쉬에 비휘발성 메모리가 도입되면서, 버퍼캐쉬의 성능을 높이기 위한 노력은 적중률 증가가 아닌 디스크 접근빈도를 감소 시키는 방법으로 접근해야 한다.

본 논문은 모두 5 장으로 구성되어 있다. 다음 장에서는 관련 연구들에 대해 살펴본다. 3장에서는 시뮬레이터를 이용한 실험 환경과 [6]의 논문에서 제시한 캐쉬관리 정책을 소개하고 실험 결과를 소개한다. 4장에서는 실험결과를 분석함으로써 캐쉬에 비휘발성 메모리가 추가되면 읽기, 쓰기요청에 따른 비용이 어떻게 달라지는지 알아보고 실험으로 확인한다. 마지막으로 5 장에서는 본 논문의 결론과 앞으로의 연구 방향을 제시한다.

#### 2. 관련 연구

비휘발성 메모리를 사용하여 컴퓨터 시스템의 성능향상 연구는 크게 3가지 분야로 나누어 이루어져 왔다. 첫째, 예상치 못한 전원문제 발생시 비휘발성 메모리를 이용하여 빠른 복구(fast

\* 본 연구는 한국과학재단 특장기초연구(R01-2004-000-10188-0) 지원으로 수행되었음.

recovery)를 이루려는 연구이다[1, 2]. 두번째로 비휘발성 메모리와 휘발성 메모리를 함께 사용하는 시스템에서 비휘발성 메모리에 적절한 정보를 영구히 저장함으로써, 성능향상을 꾀한 연구가 있었다[3, 4, 5]. 마지막으로, 비휘발성 메모리를 캐쉬로 사용하여 성능을 향상시킨 연구가 있었다. [6]의 연구에서는 비휘발성 메모리와 휘발성 메모리가 함께 있는 시스템에서 두 종류의 메모리를 캐쉬로 활용할 수 있는 여러가지 기법들을 소개하고 비교하였다. [7]은 Client-Server 시스템에서의 비휘발성 메모리 활용에 대한 연구이다. 디스크가 없는 Client의 경우 1~2 메가바이트 정도의 작은 비휘발성 메모리만으로 50%이상 쓰기요청을 감소시킬 수 있었고, 서버측에서는 비휘발성 메모리를 쓰기 캐쉬로 사용하여 20~90% 쓰기요청을 감소시킬 수 있었다. [8]의 연구에서는 비휘발성 메모리를 쓰기 캐쉬로 사용할 경우의 디스크 스케줄링 정책에 대해 비교하였다. LRU(Least Recently Used), STF(Shortest Access Time First), LST(Largest Segment per Track) 정책을 비교를 하였고, 또한 비휘발성 메모리에 크린블록을 확보하기 위한 single, dual threshold eviction 정책을 제안하여, 비휘발성 메모리가 없는 경우와 비교할 때 최소 70%~80%의 성능개선 효과를 냈다.

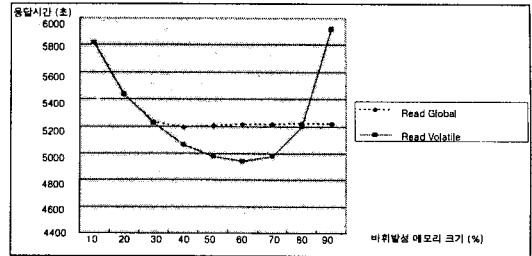
본 연구에서는 [6]의 논문 제시한 캐쉬관리 기법들을 시뮬레이터를 이용한 실험을 통하여 실험결과를 분석하였다. 그 결과 휘발성 메모리와 비휘발성 메모리가 함께 포함된 캐쉬에서는 읽기와 쓰기요청을 구분하여 캐쉬에서 적중으로 얻는 이득과 미스로드는 비용이 읽기요청인지, 쓰기요청인지에 따라 다름을 실험을 통해 확인할 수 있었다.

### 3. 실험환경 및 결과

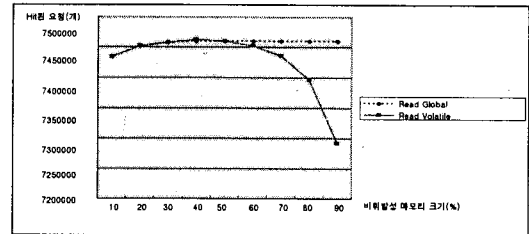
비휘발성 메모리를 버퍼캐쉬로 사용하게 되면, 두 가지의 이득을 얻을 수 있다. 먼저 더티블록을 모두 비휘발성 메모리에 캐싱함으로써 갑작스런 전원문제에도 데이터가 손실되지 않는 안정성을 확보할 수 있다. 두번째로, 비휘발성 메모리에 캐싱되는 더티 블록은 디스크로 되쓰기가 필요없기 때문에 성능적인 측면에서도 이득을 얻을 수 있다. 안정성과 성능향상을 동시에 얻기 위해서 쓰기 요청은 모두 비휘발성 메모리에 캐싱되어야 한다. 읽기 요청은 휘발성 메모리, 비휘발성 메모리 어느 곳에도 캐싱이 가능하다. 이러한 조건에서 캐쉬에서 발생하는 상황과 그 특성들을 실험을 통해 알아보았다.

버퍼캐쉬의 특성을 분석하기 위해 사용한 시뮬레이터는 크게 3 부분으로 구성되어 있다. UNIX 파일 시스템의 동작을 똑같이 흉내낸 파일 시스템 시뮬레이터, 휘발성 메모리와 비휘발성 메모리로 구성된 버퍼캐쉬 시뮬레이터 그리고 실제 디스크를 정밀하게 흉내낸 DiskSim 3.0 [9]으로 구성되어 있다. 이런 시뮬레이터로 TPC-C 트레이스의 요청을 처리하였는데, DiskSim 3.0에서 지원하지 않는 하드디스크 시뮬레이터의 저장용량이 TPC-C 트레이스의 크기보다 작은 이유로 실험에서는 트레이스를 1/10으로 축소한 요청을 처리하였다. TPC-C의 쓰기 요청의 비율은 15.35%이며, 실험에서 사용한 1/10로 축소한 트레이스의 쓰기요청 비율은 15.33%로써 거의 같은 성격을 나타낸다. 이 트레이스는 전체 4.96GByte의 파일에 대해 9.26GByte의 요청이 이루어진다. 전체 파일 중 한번이라도 요청이 되었던 블록의 총 크기는 1.69GByte이다. 실험에서 버퍼캐쉬의 총 크기는 1.69GByte의 1/10인 169MByte를 사용하였다. 실험에서 사용한 버퍼캐쉬 관리 정책은 다음과 같다.

1. Read Volatile - 쓰기 요청은 항상 비휘발성 메모리에 캐싱이 되고, 읽기 요청은 휘발성 메모리에만 캐싱이 된다. ([6]의 논문에서 LRU Volatile이라 부르고 있다.)
2. Read Global - 쓰기 요청은 항상 비휘발성 메모리에 캐싱이 되고, 읽기 요청은 휘발성 메모리와 비휘발성 메모리를 합해서 교체할 페이지를 선택한다. ([6]의 논문에서 LRU



<그림 1> 정책에 따른 응답시간 비교



<그림 2> 정책에 따른 적중된 요청 횟수 비교

Global이라 부르고 있다.)

두 경우 모두 비휘발성 메모리에서 블록이 교체될 때 제거되는 블록은 디스크로 쓰고 난 후에 휘발성 메모리로 옮겨질 수 있는 기회를 가진다. (이를 [7]의 논문에서 Write Global이라 부르고 있다). 또한 휘발성 메모리에 들어있는 블록이 쓰기 요청으로 인해 비휘발성 메모리로 옮겨질 경우엔 휘발성 메모리의 블록을 삭제한다. ([6]의 논문에서 Write Purge라고 부르고 있다). 휘발성 메모리와 비휘발성 메모리는 모두 별도의 LRU정책에 의해 관리되며, 버퍼캐쉬의 총합은 항상 169MByte로 고정된 상태에서 비휘발성 메모리의 비율을 조정하여 실험하였다.

<그림 1>은 비휘발성 메모리의 비율변화에 대한 응답시간의 그래프이고, <그림 2>는 비휘발성 메모리의 비율변화에 따른 캐쉬에서 적중된 요청의 개수이다. <그림 2>의 적중횟수를 보면 Read Global이 전구간에서 Read Volatile에 비해 높은 적중횟수를 기록한다. 읽기 요청이 비휘발성 메모리에도 캐싱될 수 있기 때문에 항상 휘발성 메모리에만 캐싱되어야 하는 Read Volatile에 비해 적중횟수는 높은 결과가 나타난다. 휘발성 메모리만으로 구성된 캐쉬에서는 미스된 요청의 횟수가 응답시간과 거의 비례하였다. 그러나 이번의 실험에서 나타난 결과에서는 미스된 요청의 횟수로 응답시간을 설명할 수 없다. 이러한 결과가 나타난 이유에 대한 분석은 4장에서 이어진다.

### 4. 실험결과 분석 및 새로운 성능측정 방법

3장의 실험 결과에서 나타난 응답시간과 적중률이 일치하지 않는 이유를 분석하기 위해 비휘발성 메모리와 휘발성 메모리가 함께 있는 버퍼캐쉬의 특징을 살펴보았다. 특히, 응답시간에 직접적인 영향을 미치는 디스크 접근횟수를 고려하여 생각해 보면 다음과 같이 구분할 수 있다. 첫째, 요청이 미스되었을 경우는 다음 4가지 경우로 구분된다.

1. Read Replace-Dirty - 읽기 요청이 미스되어 교체될 블록이 더티일 경우는 더티 블록을 저장하기 위한 디스크 접근과 요청을 처리하기 위한 디스크 접근이 일어난다.
2. Read Overwrite-Clean - 읽기 요청의 미스로 교체될 블록이 크린일 경우는 요청을 처리하기 위한 디스크 접근만 일어난다.
3. Write Replace-Dirty - 쓰기 요청이 미스되어 교체될 블록

읽기 요청			쓰기 요청			
미스		적중	미스		적중	
Read Replace Dirty	Read Overwrite Clean		Write Replace Dirty	Write Overwrite Clean	휘발성 메모리에 있을 경우	비휘발성 메모리에 있을 경우
2회	1회	0회	1회	0회	쓰기 요청의 미스로 처리	0회

<표 1> 각 요청의 적중, 미스에 따른 디스크 접근횟수

비휘발성 메모리 크기 (%)	Read Volatile		Read Global		두 정책간의 차이	
	Read Replace Dirty	Write Overwrite Clean	Read Replace Dirty	Write Overwrite Clean	Read Replace Dirty	Write Overwrite Clean
10	0	2217	0	2217	0	0
20	0	4434	0	4434	0	0
30	0	6651	116	6695	116	44
40	0	8868	33704	35855	33074	26987
60	0	13302	92216	79851	92216	66549
80	0	17736	119613	91504	119613	73768
90	0	19953	127699	91708	127699	71755

<표 2> 정책에 따른 Read Replace-Dirty, Write Overwrite-Clean의 횟수

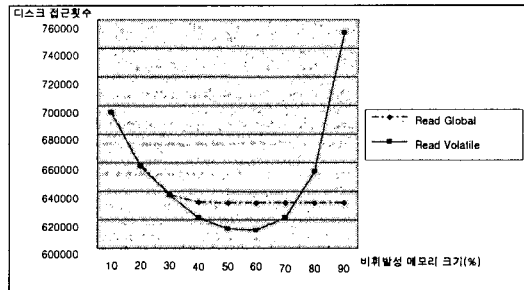
이 데이터 경우, 데이터 블록을 저장하기 위한 디스크 접근만 일어나고 요청은 비휘발성 메모리에 캐싱되기 때문에 요청을 처리하기 위한 디스크 접근은 없다.

4. Write Overwrite-Clean - 쓰기 요청의 미스로 교체될 블록이 크면 블록이 캐싱되는 요청은 미스지만 디스크 접근은 없다. 두 번째로 요청이 캐시에서 적중된 경우를 살펴보면, 읽기요청이 적중되었을 경우는 요청이 캐시에서 처리되어 완료된다. 그러나 쓰기요청의 경우는 휘발성메모리에 있는 블록이 적중되었을 경우, 그 블록은 갱신되어 데이터블록이 되기 때문에 비휘발성 메모리로 옮겨져야 한다. 즉, 쓰기요청이 미스인 경우와 동일하다. 그러므로 버퍼캐시의 성능을 측정하기 위해 디스크 접근횟수를 구하는 식은 다음과 같다.

$$\text{디스크 접근횟수} = \text{Read Replace-Dirty} \times 2 + \text{Read Overwrite-Clean} + \text{Write Replace-Dirty} \quad (1)$$

3장의 실험에서 캐시의 적중률과 응답시간이 서로 일치하지 않는 원인을 디스크 접근횟수를 바탕으로 살펴보자. Read Volatile이 Read Global에 비해 적중횟수는 적지만 전체 응답시간에서는 좋은 성능을 나타내는 구간이 나타나는 원인은 위에서 언급한 버퍼캐시의 특성에서 찾아볼 수 있다. Read Global은 비휘발성 메모리에도 읽기 요청에 대한 캐싱은 한다. 그러나, 비휘발성 메모리는 대부분이 쓰기요청에 대한 캐시로 사용되기 때문에 데이터 블록이 교체될 가능성이 크다. 반대로 쓰기 요청이 미스된 경우 비휘발성 메모리에 크리블록이 교체될 때는 디스크 접근 없이 요청을 처리할 수 있다. 즉, Read Global정책에서는 성능에 악영향을 미치는 Read Replace-Dirty와 성능향상에 도움을 주는 Write Overwrite-Clean현상이 동시에 일어난다. 그러나 <표 2>의 "두 정책간의 차이" 에서 확인할 수 있듯이 Read Replace-Dirty의 횟수가 Write Overwrite-Clean의 횟수보다 많다. 그러므로 Read Global정책이 Read Volatile 정책에 비해 항상 적중횟수는 높지만 성능은 오히려 나쁜 경우가 발생한다.

디스크 접근횟수와 응답시간은 직접적인 비례관계가 있는지를 확인하기 위한 실험을 3장에서 언급한 시뮬레이터와 캐시 관리 정책을 이용하여 하였다. 디스크 접근횟수는 (1)의 식을 이용하여 계산했다. <그림 3>은 각 정책에 대한 디스크 접근횟수를 나타낸 그래프이다. <그림 1>의 응답시간 그래프와 비교할 때 디스크 접근횟수는 응답시간과 직접적인 비례관계가 있음을 확인할 수 있다. 이상의 결과에서 확인할 수 있듯이 비휘발성 메



<그림 3> 정책에 따른 디스크 접근횟수

모리가 추가된 버퍼캐시는 적중률 보다는 디스크 접근횟수를 이용하여 성능을 정확히 측정할 수 있다.

### 5. 결론 및 향후 연구계획

DRAM과 유사한 접근 속도를 가지며, 영속적인 데이터 저장이 가능한 비휘발성 메모리 (FRAM, PRAM, MRAM)를 휘발성 메모리와 함께 버퍼캐시로 사용하면 안정성과 성능향상을 동시에 이룰 수 있다. 비휘발성 메모리와 휘발성 메모리를 혼용한 캐시관리 정책에 관한 기존의 연구는 실험결과에 대한 분석이 미흡했다[6]. 본 연구에서는 시뮬레이터를 이용하여 기존의 연구에서 제시한 캐시관리 정책을 실험하고 실험 결과를 분석하여 비휘발성 메모리와 휘발성 메모리가 함께 사용되는 캐시의 새로운 특성을 밝혀냈다. 비휘발성 메모리가 캐시에 포함되면 읽기, 쓰기의 요청의 종류, 미스되었을 경우 캐싱될 블록의 데이터부, 읽기 요청이 적중되었을 때, 적중된 메모리의 종류에 따라 각각의 요청을 처리하기 위한 디스크 접근횟수가 달라지는 특성을 나타낸다. 이러한 특성으로 인해 비휘발성 메모리가 추가된 버퍼캐시의 성능, 즉 응답시간은 적중률보다는 디스크 접근횟수를 이용하여 정확한 측정이 가능하다. 그러므로, 비휘발성 메모리가 추가된 버퍼캐시의 성능향상을 위해서는 적중률을 높여주는 노력에서 디스크 접근을 줄이려는 노력으로 바뀌어야 한다.

본 연구팀은 앞으로 비휘발성 메모리가 추가된 버퍼캐시에서 디스크 접근을 최소화로 줄일 수 있는 캐시관리 정책에 대한 연구를 진행할 예정이다.

### 6. 참고문헌

- [1] Mary Baker and Mark Sullivan, "The Recovery Box: Using Fast Recovery to Provide High Availability in the Unix Environment", 1992 USENIX Summer Conference, pp. 31-44.
- [2] R. Ohmura, N. Yamasaki, and Y. Anzai, "Device State Recovery in Non-volatile Main Memory Systems", International Computer Software and application Conference, 2003.
- [3] E. Miller, S. Brandt, and D. Long, "HeRMES: High-Performance, Reliable MRAM-Enabled Storage", In The Eighth Workshop on Hot Topics in Operating Systems (HotOS VIII), Schloss Elmau, Germany, May 2001.
- [4] A.B. Szczurowska "MRAM-Preliminary Analysis for File System Design" MS Thesis, University of California, Santa Cruz, 2002.
- [5] A. A. Wang, P. Reiter, G. J. Popek, and G. H. Kuenning, "Conquest: Better Performance Through a Disk/Persistent-RAM Hybrid File System", Proceedings of the USENIX Annual Technical Conference (USENIX 02).
- [6] Sedat Akyurek and Kenneth Salem, "Management of Partially Safe Buffers", IEEE Transactions on Computers, Vol. 44 No. 3 pp. 394-407, 1995.
- [7] Mary Baker, Satoshi Asami, Etienne Deprit, John Ousterhout, and Margo Seltzer, "Non-volatile memory for fast, reliable file systems", Proceedings of the 5th International Conference on Architectural Support for Programming Languages and Operating System, Oct. 1992.
- [8] T. Haining and D. Long, "Management policies for non-volatile write caches", In 1999 IEEE International Performance, Computing and Communications Conference, pp. 321-328, Feb 1999.
- [9] The DiskSim Simulation Environment (Version 3.0). <http://www.pdcl.cmu.edu/DiskSim/index.html>