

차세대 메모리를 활용한 데이터베이스 기반 버퍼 관리 기법의 최신 동향

A Recent Trend of Buffer Management based on Database
using Next-generation Memory Module

이미경(Mi-kyung Lee)¹, 김재형(Jae-hyung Kim)², 박상현(Sang-hyun Park)³

요 약

최근 전원이 공급되지 않는 상태에서도 데이터를 보존하기 위한 비휘발성 차세대 메모리 소자가 각광받고 있다. 그러나 제한된 수명과 DRAM 대비 부족한 자연대기로 인해 단 기간 내에 DRAM을 완전히 대체하기란 어려울 것으로 예상되고 있다. 따라서 차세대 메모리 저장장치를 하드디스크를 대체하거나 하드디스크의 버퍼로 활용하여 현재의 환경에서 데이터베이스 시스템의 성능을 향상시키기 위한 연구들이 계속해서 진행되고 있다. 본 논문에서는 차세대 메모리 저장장치를 활용한 버퍼 관리 기법을 연구하는 것이 중요한 연구 주제라고 판단하여 버퍼 관리 기법을 위해 고려해야 할 메모리 소자의 특성과 내부 구조에 대해 설명하고 이러한 특성을 적극적으로 활용한 각각의 최신 버퍼 관리 기법들의 동향을 소개하고자 한다.

주제어: DRAM, 비휘발성 저장장치, 플래시 메모리, SSD, NVRAM, 버퍼 관리 알고리즘

1 연세대학교 컴퓨터과학과, 석사과정.

2 연세대학교 컴퓨터과학과, 박사과정.

3 연세대학교 컴퓨터과학과, 교수, 교신저자.

+ 본 연구는 2015년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행하였음.
(NRF-2015R1A2A1A05001845)

+ 논문접수: 2016년 3월 3일, 심사완료: 2016년 3월 15일, 게재승인: 2016년 3월 24일.

Abstract

Next-generation non-volatile memory modules have emerged recently in storage market. However, due to the limited life time and poor performance, these storages are predicted not to completely replace DRAM in the foreseeable future. Accordingly, there is a growing body of studies that these memories are used for an alternative to HDD or write buffer between DRAM and storage devices. In this paper, we consider the buffer management schemes using next-generation non-volatile memory module as an important research topic and also explain the characteristics of these memory modules. Finally, we describe a recent trend of the buffer management schemes using next-generation non-volatile memory modules.

Keywords: DRAM, non-volatile storage, flash memory, SSD, NVRAM, buffer management algorithm

1. 서 론

컴퓨터의 주기억장치로서 주로 사용되는 DRAM(Dynamic random access memory)은 하나의 기억 소자에 1개의 트랜지스터(Transistor)와 1개의 커패시터(Capacitor)로 구성된다. 이와 같은 간단한 구성 덕분에 DRAM은 높은 직접도로 제작될 수 있고 데이터 접근 속도가 매우 빠르며 저비용으로 대용량화할 수 있다. 그러나 DRAM은 전원이 공급되고 있는 동안 데이터를 일정한 주기로 다시 기록해 주지 않으면 데이터가 사라지는 문제점을 가지고 있다. 이러한 문제점을 극복하기 위하여 하드 디스크(Hard disk)와 같이 전원이 공급되지 않는 상태에서도 데이터를 유지하기 위한 차세대 메모리 저장장치들이 새롭게 각광받고 있다.

표 1 [1-2] 과 같이 현재까지 가장 광범위하게 사용되고 지속적으로 연구되고 있는 대표적인 메모리 소자들 중 하나인 플래시 메모리(Flash memory)는 비휘발성 특성을 갖고 있으나, DRAM 대비 읽기 및 쓰기 지연 시간이 길고 제자리 덮어쓰기(In-place update)가 불가능하며, 비트 정보가 저장된 플래시 메모리의 셀들의 지우고 쓰는 횟수가 제한되어 있어 아직까진 DRAM을 완전히 대체하지 못하고 있다. 그럼에도 불구하고 플래시 메모리는 하드 디스크와 다르게 데이터 접근 시, 기계적인 구동 없이 전기적인 신호를 사용하여 데이터에 접근하기 때문에 요청이 들어온 즉시 데이터를 접근하므로 하드 디스크보

다 전력 소모가 적다. 이와 같은 장점들 덕분에 여러 개의 플래시 메모리들을 묶음으로써 하드 디스크를 대체하기 위한 플래시 SSD(Solid state drive)가 등장했다. SSD는 플래시 메모리의 제작 공정의 발달에 따라 용량 대비 가격이 낮아지고 있어 서버 시장에까지 사용 영역을 넓히고 있다.

이외에도 보다 적극적으로 DRAM을 대체하기 위한 대표적인 비휘발성 메모리 소자를 통칭하는 NVRAM(Non-volatile random access memory)이 있다. NVRAM은 DIMM(Dual in-line memory module) 인터페이스를 통해 바이트 단위의 연산을 제공하며 제자리 덮어쓰기가 가능해 가비지 컬렉션(Garbage collection)으로 인한 오버헤드 문제를 우려할 필요가 없다. 또한, 플래시 메모리 대비 빠른 읽기 및 쓰기 지연 대기 시간을 보이고 있어 DRAM을 대체할 소자로 주목받고 있다. 그러나 DRAM 대비 느린 쓰기 지연대기와 제한적인 수명, 그리고 높은 용량 대비 가격 문제로 인해 단 기간 내에 DRAM을 완전히 대체하기란 어려울 것으로 예상되고 있다. 따라서 NVRAM을 DRAM을 대체하는 구조가 아닌, 하드 디스크 혹은 플래시 SSD와 혼합된 시스템상에서 캐시로 활용하려는 시도가 지속적으로 연구되고 있고, 이에 부응하여 각 저장장치의 특성에 적합한 버퍼 관리 기법의 연구는 앞으로도 유망할 것으로 판단된다.

본 논문에서는 차세대 메모리 소자를 활용한 버퍼

[표 1] 메모리 소자의 특성

Name	Read latency	Write latency	Non-volatility	Endurance
DRAM	Fast	Fast	NO	10^{16}
SRAM	Fast	Fast	NO	10^{16}
NAND flash	Slow	Slow	YES	$10^5 \sim 10^6$
PRAM	Fast	Slow	YES	10^8
STT-MRAM	Fast	Slow	YES	$> 10^{15}$

관리 기법을 연구하는 것이 중요한 연구 주제라고 판단하여 이와 관련된 베퍼 관리 기법들에 대해 살펴보기로 한다. 논문의 구성은 다음과 같다. 2장에서는 베퍼 관리 기법을 위해 고려해야 할 메모리 소자의 특성에 대해 설명하고, 3장에서는 2장에서 설명했던 메모리들을 활용한 최신 베퍼 관리 기법들을 소개한다. 4장에서는 차세대 메모리를 활용한 베퍼 관리 기법들을 토대로 향후 발전될 베퍼 관리 기법들의 동향 제시와 결론을 기술하도록 한다.

2. 차세대 메모리 저장 장치

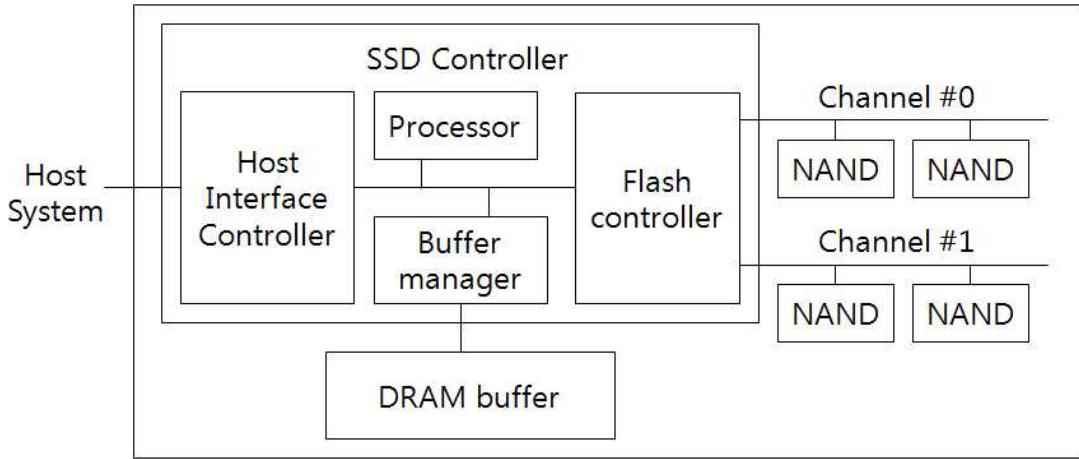
2.1 플래시 메모리

전원이 차단되어도 데이터가 그대로 보존되는 플래시 메모리는 데이터를 전기적으로 기록하고 삭제하기 때문에 하드 디스크에 비해 전력소모가 적고 데이터 접근 속도가 빠르다. 이와 같은 특성들로 인해 플래시 메모리는 차세대 메모리 저장장치로 크게 주목 받아왔다. 플래시 메모리 소자는 메모리 셀의 배열에 따라 크게 NOR 플래시 메모리와 NAND 플래시 메모리로 나눌 수 있다. 병렬로 연결된 셀의 구조로 인해 임의적 접근은 가능하지만 쓰기 연산의 수행 속도가 느리고 대용량화하기가 어려운 NOR 플래시 메모리와는 달리, 직렬로 연결된 셀의 구조를 가진 NAND 플래시 메모리는 NOR 플래시 메모리에 비해 읽기 연산의 수행 속도는 느리지만, 쓰기 연산의 수행 속도가 빠르고 비트 당 제작비가 낮아 대용량화하기가 쉽기 때문에 데이터를 저장하기 위한 용도로 적합하다. 이러한 특성을 바탕으로 하드 디스크보다 더 빠른 입출력 처리를 필요로 하는 요구에 부응하여 새로운 비휘발성 저장 장치인 SSD(Solid state disk)가 등장하였으며, 이를 기반으로 한 연

구 역시 활발히 진행되고 있다. 그러나 플래시 메모리는 쓰기 연산의 비용이 읽기 연산에 비해 확연히 크고 덮어쓰기가 불가능하며 셀 당 쓰기 및 삭제 연산의 횟수가 제한되어 있다 [3]. 이러한 한계점은 모바일 또는 서버 환경에서의 데이터베이스 시스템이나 운영체제와 같은 시스템 소프트웨어에서 기존의 하드 디스크 기반의 방법론에 SSD 혹은 NAND 플래시 메모리가 적용될 때 기대할 만한 성능 향상을 이끌어내지 못할 수 있다. 따라서 운영체제 혹은 데이터베이스 시스템의 성능을 향상시키기 위해 하드 디스크에 최적화된 기존의 방법론들을 NAND 플래시 메모리의 특성에 맞게 개선하려는 연구가 지속적으로 진행되고 있다.

2.2 플래시 SSD

NAND 플래시 메모리를 기반으로 하는 SSD는 NAND 플래시 메모리 제작 기술의 발전에 따라 용량 대비 가격 면에서 하드 디스크에 근접하게 되면서 사용 범위가 확대되는 추세다. 이와 더불어 클라우드(Cloud) 및 데이터 센터가 성장하면서 대규모 서버 시스템을 위한 대용량 데이터 저장장치로도 각광받고 있다. SSD는 그림 1에서 보는 바와 같이 SSD 컨트롤러(Controller), DRAM, 그리고 다수의 플래시 메모리들로 구성된다. 호스트(Host)로부터 읽기 또는 쓰기 명령어를 입력받으면 DRAM에 저장된 주소 변환 테이블을 통해 호스트로부터 받은 요청에 대한 논리적 주소(Logical block address)를 물리적 주소(Physical block address)로 변환한 후, 플래시 메모리로 저장한다. 이 때, 다수의 플래시 메모리들은 독립적인 채널(Channel)들로 묶여있는 다중 채널 구조를 가지며 각 채널은 입력받은 명령어가 다수의 플래시 메모리에 최대한 병렬적으로 데이터를 읽고 쓸 수 있도록 함으로써 보다 효율적인



[그림 1] SSD의 구조

입출력 연산을 수행할 수 있도록 한다. 즉 SSD는 다중 채널을 통한 내부 병렬성(Internal parallelism)을 최대한 활용하여 입출력 성능을 높인다 [4]. 한편, DRAM은 주소 변환 테이블을 임시 버퍼 공간으로 사용하기도 하는데 이러한 경우, 하드 디스크에 최적화된 방법론들을 그대로 적용시키면 하드 디스크와는 전혀 다른 성질을 지닌 SSD에 맞지 않아 기대했던 성능을 이끌어내지 못 할 가능성이 있다. 이러한 문제를 해결하고자 하드 디스크 기반의 기준 방법론들을 SSD에 최적화시켜 성능 개선을 시도하려는 연구가 많이 진행되어 왔다. 최근에는 SSD의 NAND 플래시 메모리 소자의 특성 뿐만 아니라 위에서 언급한 SSD 내부의 채널 구성 등 보다 깊은 수준의 하드웨어 특성을 다각도로 고려하여 성능 향상을 꾀하는 방법론들이 연구되고 있다.

2.3 NVRAM

현재까지 주로 통용되는 컴퓨터 구조는 휘발성 메모리인 DRAM과 하드 디스크와 같은 비휘발성 저장장치로 이루어진 구조를 토대로 하고 있으며, 운영체제에서 수행하는 모든 작업들은 메모리에 1차적으로 반영이 되고 난 다음, 데이터의 영속성을 위해 하

드 디스크에 다양한 방식으로 메모리에 반영된 데이터를 저장하게 된다. 이와 같은 컴퓨터 구조에서 비휘발성 메모리 소자인 NVRAM은 보조 기억저장장치에 한정적인 NAND 플래시 메모리를 넘어, 주기억장치인 DRAM을 대체하고 있다. NVRAM은 휘발성 메모리인 DRAM과 다르게 비휘발성인 특성을 갖는 메모리 소자로서, NAND 플래시 메모리가 지닌 부족한 특성들로 인하여 기존 DRAM을 대체하지 못한 것과는 달리, DIMM 인터페이스를 통해 바이트 단위 연산을 제공하며 NAND 플래시 대비 수십 배에서 수백 배 이상 빠른 자연 대기 시간을 보장하는 DRAM을 대체할 메모리 소자들의 통칭이라고 할 수 있다. 또한 NVRAM은 플래시 메모리와 달리 제자리 덮어쓰기가 가능하여 지속적인 쓰기 연산의 수행으로 인한 가비지 컬렉션에 대한 높은 비용 문제가 존재하지 않는다. 현재까지 NVRAM으로 분류될 수 있는 메모리 소자들에는 PRAM(Phase-change RAM), STT-MRAM(Spin-transfer torque magnetic RAM), ReRAM(Resistive RAM) 등이 있으며 이들은 유력한 차세대 메모리로 부상하고 있다. 그럼에도 불구하고 PRAM을 기준으로 DRAM 대비 느린 쓰기 연산에서의 자연대기 시간 [5] 과 제

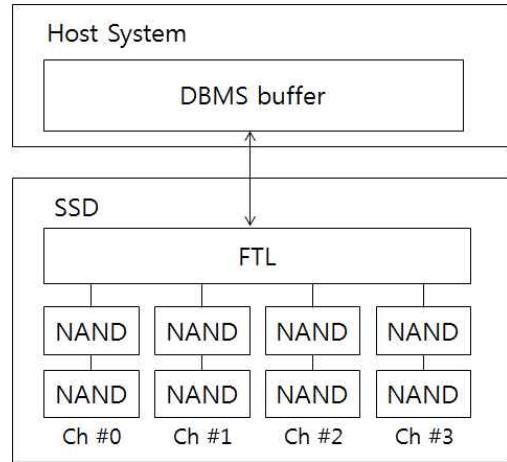
한적인 수명 문제 [6], 그리고 용량 대비 가격으로 인하여 현재까지 발표된 NVRAM이 상용 환경에서 단시일 내에 DRAM을 대체하기란 불가능할 것으로 예측되고 있다.

따라서 NVRAM을 메모리의 대체품으로 적용하기보다는 그 자체로 SSD를 대체하는 차세대 저장장치의 일환으로 사용하거나, DRAM과 하드 디스크 또는 SSD 사이의 캐시 장치로 사용하려는 시도가 연구되고 있다. 특히 캐시 장치로 사용되는 NVRAM 환경을 고려할 때 운영체제 혹은 데이터베이스의 성능을 좌우하는 버퍼 교체 기법에서 기존 하드 디스크와 SSD에 특화된 방법만을 적용할 경우, 그 특성을 살리지 못해 원하는 성능을 충분히 확보하지 못하는 문제가 있으며, 이러한 문제를 다루기 위한 다양한 연구들은 DRAM을 대체할 새로운 메모리 소자가 나오기 전까지 다양한 저장장치 아키텍처 혹은 인터페이스의 등장과 더불어 앞으로도 유망할 것으로 보인다.

3. 차세대 메모리 저장장치를 활용한 버퍼 관리 기법

3.1 플래시 메모리의 특성을 활용한 버퍼 관리 기법

플래시 메모리가 등장하기 이전에는 하드 디스크를 기반으로 한 버퍼 관리 기법들이 주로 연구되었다. 그러나 하드 디스크를 기반으로 한 기존의 버퍼 관리 기법들은 그림 2와 같은 플래시 저장장치를 보조기억장치로 사용한 데이터베이스 시스템에 적용 시켰을 때 시스템 성능이 기대에 미치지 못 할 우려가 있다. 이러한 문제를 해결하기 위하여 하드 디스크 기반의 버퍼 관리 기법들을 앞서 살펴보았던 틀



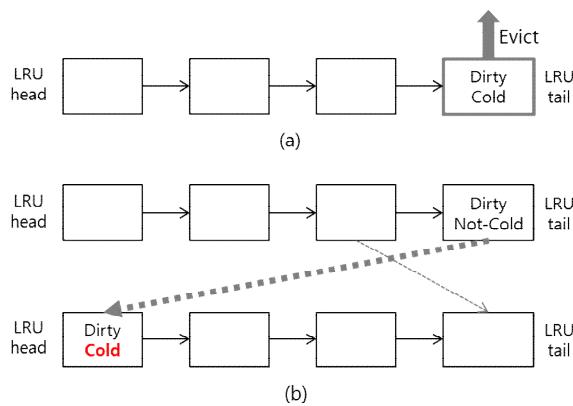
[그림 2] 플래시 메모리 기반 데이터베이스

시스템의 구조

래시 메모리만의 특성에 맞게 개선하려는 시도가 이루어지고 있다.

플래시 메모리의 부진한 쓰기 성능을 보완하기 위해 LRU(Least Recently Used) 기법[7-9]을 개선하는 연구들이 진행되고 있다. Heeseung Jo [10]은 PMP(Portable media player)의 성능을 높이기 위해 FAB(Flash-Aware Buffer management policy for portable media players) 기법을 제안하였으며, 희생자 페이지를 선정할 때 시간적 지역성(Temporal locality)보다 플래시 메모리의 우수한 순차 쓰기 성능을 고려한다. 따라서 희생자 페이지를 결정하려고 한다. 따라서 버퍼 교체 시, 페이지 개수가 가장 많은 블록을 희생자로 선택한다. Seon-yeong Park [11]은 플래시 메모리의 쓰기 연산의 비용을 줄이기 위한 CFLRU(Clean-First Least Recently Used) 기법을 제안하였다. 이를 위하여 버퍼를 최근에 참조된 페이지들이 모여 있는 워킹 영역(Working region)과 오랫동안 참조되지 않은 페이지들이 이 영역에 모여 있는 클린-퍼스트 영역(Clean-first region)으로 분류한다. 버퍼 교체 시, 클린-퍼스트 영역에 있는 페이지들 중 영역의

끝에 위치한 클린 페이지(Clean page)를 우선적으로 버퍼에서 내보내어 쓰기 연산의 비용을 절감할 수 있도록 한다. 한편, Hoyoung Jung [12] 은 콜드 탐지(Cold-detection) 기법을 사용하여 참조 빈도 수가 낮은 콜드 페이지(Cold page)를 감별하는 LRU-WSR(Integration of LRU and Write Sequence Reordering) 기법을 제안하였다. 이 기법은 버퍼 교체 시 리스트의 꼬리에 있는 희생자 후보 페이지가 클린 페이지라면 버퍼에서 내보내고, 더티 페이지이면 그림 3과 같이 해당 페이지가 콜드 페이지인지 구별하여 콜드 페이지인 경우 희생자 페이지로 간주한다. 이와 같은 과정은 클린 페이지 또는 콜드한 더티 페이지를 버퍼에서 내보내고 콜드하지 않은 더티 페이지를 버퍼에서 유지할 수 있도록 하므로 캐시 적중률의 저하를 해결하면서도 플러시되는 더티 페이지 개수를 낮출 수 있다.



[그림 3] LRU-WSR의 희생자 페이지 선정 과정

Yi Ou [13] 은 CFDC(Clean First Dirty Clustered) 기법을 제안하였으며 버퍼를 워킹 영역(Working region)과 중요도 영역(Priority region)으로 분류한다. 워킹 영역은 다시 참조될 가능성이 높은 핫 페이지(Hot page)들이 모여 있는 영역으로, 다양한 버퍼 관리 기법들을 적용시킬 수 있다. 반면,

중요도 영역은 중요도를 가진 페이지 클러스터(Cluster)들로 이루어져 있고 클러스터는 공간적으로 근접하게 위치한 페이지들로 구성된다. 중요도 영역에 있는 클린 페이지들은 LRU 리스트로, 더티 페이지들은 클러스터들이 들어있는 중요도 큐(Priority queue)로 관리되며 클러스터 탐색을 위해 해시 테이블(Hash table)이 존재한다. 버퍼 교체 시 중요도 영역의 LRU 리스트에 있는 클린 페이지가 우선적으로 희생자 페이지로 선정되고, 중요도 영역에 클린 페이지가 없으면 중요도 큐에서 중요도가 가장 낮은 클러스터를 찾아 해당 클러스터 내에서 가장 오래된 페이지를 희생자 페이지로 선택한다. 클러스터의 중요도는 페이지 개수로 결정되며 페이지 개수가 많은 클러스터일수록 클러스터의 중요도는 낮아진다. 희생자 페이지가 플러시되는 동안 공간적 지역성(Spatial locality)을 발생시켜 플래시 메모리의 쓰기 연산의 성능을 보완한다. Zhi Li[14] 는 CCF-LRU(Cold-Clean-First LRU) 기법을 제안하였으며 참조 빈도수가 낮은 콜드-클린 페이지(Cold-clean page)들을 모아둔 콜드-클린 LRU 리스트와 그 외의 핫-클린(Hot-clean), 콜드-더티(Cold-dirty), 핫-더티(Hot-dirty) 상태를 갖는 페이지들을 모아놓은 혼합 LRU 리스트로 페이지들을 관리한다. CCF-LRU 기법은 실험에 의거하여 핫-더티 페이지, 핫-클린 페이지, 콜드-더티 페이지, 콜드-클린 페이지 순으로 교체 비용이 크다는 것을 확인하였으며 이에 따라 버퍼 교체 시 콜드-클린 LRU 리스트에 있는 콜드-클린 페이지를 우선적으로 내보낸다. 따라서 쓰기 연산의 횟수와 I/O 수행 시간을 감소시킬 수 있다. 콜드-클린 LRU 리스트에 콜드-클린 페이지가 없으면 LRU-WSR 기법의 콜드-탐지 기법을 사용하여 혼합 LRU 리스트의 꼬리에 있는 페이지가 클린 페이지인지 검사한다. 해

당 페이지가 콜드-더티 페이지인 경우에는 희생자 페이지로 간주하지만, 핫-더티 페이지인 경우에는 콜드 플래그를 사용하여 콜드 페이지로 변경한 후, 혼합 LRU 리스트의 헤드로 옮긴다. 해당 페이지가 핫-클린 페이지이면 콜드 페이지로 변경한 다음, 헤드로 이동시킴으로써 읽기 연산의 비용을 절감할 수 있도록 한다. 한편, Bokyeong Kim[15]은 PMP에서 멀티미디어 데이터뿐만 아니라 메타데이터도 고려하는 LSF(Long Sequential accessed pages First) 기법을 제안하였다. 멀티미디어 데이터와 메타 데이터는 데이터를 이루는 페이지들이 순차적이며 번갈아 나타난다. 특히 멀티미디어 데이터는 순차적 패턴의 길이가 길고 데이터를 이루는 페이지들이 한번씩만 참조되지만, 메타데이터는 패턴의 길이는 짧고 자주 참조되는 경향을 가지고 있다. LSF 기법은 페이지가 새롭게 참조되면 저장되는 콜드 리스트(Cold list)와 핫 리스트(Hot list)로 분리하여 관리한다. 버퍼 교체 시 콜드 리스트에 빈 프레임이 없을 때 콜드 리스트에 있는 페이지들의 순차적인 패턴을 검사하여 긴 순차 참조 집합(Long sequential reference set)과 짧은 순차 참조 집합(Short sequential reference set)으로 분류하고, 긴 순차 참조 집합을 버퍼에서 우선적으로 내보낸다. 따라서 캐시 적중률이 높고 쓰기 연산의 개수도 줄일 수 있다. 이와 같은 연구들은 플래시 메모리가 순차쓰기 성능이 우수하다는 점을 활용하여 LRU 기법을 개선한 연구들로 플래시 메모리의 부진한 쓰기 성능을 보완하고자 하는데 의의를 두고 있다.

또한, 플래시 메모리의 쓰기 성능을 개선하기 위하여 하드 디스크 기반의 CLOCK 기법[16]을 개선하려는 연구들도 존재한다. Debnath B.[17]은 SSD를 위한 LB-CLOCK(Large Block CLOCK) 기법을 제안하였으며, 페이지들을 SSD의 소거 블록(Erase

block)과 동일한 크기의 논리적 블록 단위로 유지하고 모든 논리적 블록은 CLOCK 기법처럼 참조 비트를 가지고 있다. 페이지 교체 시 참조 비트가 0인 블록들을 희생자 후보 집합(Victim candidate set)에 포함시키고, 집합에서 페이지 개수에 가장 많은 블록을 희생자 블록으로 선정하여 블록에 속한 모든 페이지들을 버퍼에서 내보내므로 쓰기 연산의 횟수를 감소시킬 수 있다. Hyojun Kim[18]은 스마트폰을 위한 Spatial Clock 기법을 제안하였으며 버퍼에 페이지를 추가할 때마다 페이지들을 논리적인 섹터 번호 순서로 정렬하기 때문에 다른 기법보다 순차쓰기 연산들을 많이 생성한다. Donghyun Kang[19]은 TS-CLOCK(Temporal and Spatial locality aware CLOCK) 기법을 제안하였는데, 이 기법은 페이지들을 블록 단위로 묶고 블록 내 페이지들을 논리적 섹터 번호를 기준으로 정렬한다. 이 기법은 버퍼 교체 시, t-hand 포인터가 가리키는 페이지의 참조 횟수가 0일 때 해당 페이지가 클린 페이지이면 버퍼에서 내보내고, 더티 페이지이면 s-hand 포인터를 이용하여 해당 페이지가 속한 블록에 들어있는 페이지들의 참조 횟수를 순차적으로 검사하여 희생자 페이지를 결정한다. 클린 페이지를 희생자 페이지로 우선적으로 고려하였기 때문에 쓰기 연산의 횟수를 줄일 수 있고 다른 기법에 비해 순차적 쓰기 패턴을 많이 생성한다는 장점을 가지고 있다.

이와 같이 NAND 플래시 메모리를 활용한 버퍼 관리 기법들은 페이지를 교체하는 과정에서 더티 페이지를 디스크에 쓰는 비용을 줄이는 데 중점을 두고 있다. 이를 위하여 플래시 메모리를 활용한 버퍼 관리 기법들을 크게 세 가지 형태의 기법들로 분류할 수 있다. 쓰기 연산의 횟수를 줄이기 위하여 클린 페이지를 우선적으로 버퍼에서 내보내는 데 중점을 둔 기법, 플래시 메모리의 순차쓰기 성능을 이용함

으로써 부진한 쓰기 성능을 보완하는 기법, 시간적 지역성을 고려하여 캐시 적중률을 높이고 이로 인해 페이지 교체 횟수 자체를 감소시키려는 기법이 이에 해당된다. 그러나 메모리 직접도 등 플래시 메모리의 물리적인 구조를 변화시키는 연구가 계속해서 다루어지고 있기 때문에 제시한 기법들 외에도 다양한 방법들을 활용한 버퍼 관리 기법들이 제시될 수 있다.

3.2 플래시 SSD의 다중 채널 구조를 이용한 버퍼 관리 기법

NTFS, FAT과 같은 파일 시스템이나 운영체제, 그리고 이를 기반으로 하는 프로그램들은 기본적으로 섹터 단위로 작업을 수행한다. 그러나 플래시 SSD는 블록과 페이지 기반으로 작업을 수행한다. 이러한 구조적 차이 때문에 파일 시스템과 운영체제는 직접적으로 플래시 메모리를 사용할 수 없게 되어있다. 그러나 플래시 SSD는 자체 컨트롤러(Controller) 단계에서 FTL(Flash Translation Layer)을 통해 파일 시스템과 운영체제 기반의 논리적인 주소를 플래시 메모리에 맞는 물리적인 주소로 변환함으로써 기존의 파일 시스템과 운영체제, 그리고 이를 기반으로 하는 프로그램들은 이에 대해 특별히 고려할 필요 없이 SSD를 사용할 수 있도록 한다. 그 외에도 FTL은 SSD의 다중 채널 구조를 이용하여 논리적인 페이지들을 여러 개의 채널에서 병렬적으로 처리할 수 있도록 하고, 가비지 컬렉션(Garbage collection)을 수행하여 플래시 메모리가 덮어쓰기가 불가능하다는 점을 해결한다. 그러나 가비지 컬렉션은 유효 페이지의 복사 작업과 블록의 소거 작업과 같은 추가 연산을 필요로 하기 때문에 이로 인해 입출력 처리에 대한 지연 시간이 길어지거나 대역폭이 변동하는 현상(Bandwidth fluctuation)

이 발생하기도 한다.

이와 같은 문제를 해결하기 위하여 SSD의 내부 버퍼를 쓰기 버퍼로 활용하여 플래시 메모리의 부진한 쓰기 연산 성능을 보완하고 이를 통해 가비지 컬렉션의 오버헤드를 감소시키려는 연구가 수행되고 있다. 특히 Sung Kyu Park [20]은 쓰기 버퍼에 대하여 버퍼 교체가 발생한 경우, I/O 병렬성을 활용하기 위해 버퍼 교체 시 각 블록을 하나의 채널에 할당하여 하나의 블록에 속한 페이지들이 동일한 플래시 메모리 블록에 저장되는 기법을 제안하였다. 이로 인해 가비지 컬렉션 발생 빈도가 감소된다. 한편, 희생자 페이지의 개수가 아니라 희생자 페이지가 선택되는 순서를 고려하는 연구를 역시 수행되어 왔다. Jinho Seol [21]은 플래시 메모리의 연산 단위가 호스트 인터페이스의 연산 단위보다 커짐에 따라 SSD는 때때로 부분적으로 채워진 페이지(Partially-filled page)에 대한 쓰기 작업이 수행될 수 있음을 주장하였다. 이와 같은 쓰기 작업은 요청된 섹터에 해당되는 페이지를 읽어오고 해당 페이지에서 요청된 섹터로 수정한 다음, 수정된 페이지를 다시 쓰는 읽기-수정-쓰기(Read-modify-read) 연산을 수행하게끔 한다. 그리고 이러한 연산은 칩-웨이팅(Chip-waiting) 문제로 이어지면서 시스템의 성능 저하로 이어질 수 있다. 이를 해결하기 위하여 부분적으로 채워진 페이지를 쓰게 되는 경우, 이웃한 페이지를 희생자 페이지로 선정하여 SSD의 멀티 칩을 병렬로 처리하며 쓰기 성능을 개선하고 있다. Xufeng Guo[22]은 SSD 내부 버퍼의 병렬성을 좀 더 극대화하는 PAB 기법을 제안하였다. 이 기법은 버퍼를 워크 존(Work zone)과 파라 존(Para zone)으로 분류하며 워크 존은 일반적인 버퍼 관리 기법을 사용하여 반복된 쓰기 연산을 줄이고 워크 존이 가득 차면 워크 존에서 페이지들을 파라 존으

로 내보낸다. 파라 존에서는 페이지들을 다중 채널에 맞게 병렬 단위(Parallel unit)를 설정하여 페이지들을 재배치하여 관리한다. 한편, Ronghui Wang [23] 은 버퍼 교체 시, 희생자 페이지의 개수와 순서를 모두 고려한 CAWR(Channel-Aware Write Reordering mechanism for SSDs) 기법을 제안하였다. 공간적 지역성을 고려하기 위해 페이지들을 클러스터링(Clustering)하고 각 클러스터는 SSD의 각 채널과 대응된다. 따라서 버퍼 교체가 발생한 경우, 각 클러스터의 페이지를 각 채널로 동시에 내보내어 병렬성을 높이고 있다. 이와 같은 연구들은 SSD 내부 버퍼를 쓰기 버퍼로 활용하여 SSD의 쓰기 성능을 보완하고자 한 연구들에 해당되며 SSD의 다중 채널 구조를 활용하여 가비지 컬렉션에 대한 오버헤드를 줄이는데 의의를 두고 있다. 또한 희생자 페이지를 선정할 때 희생될 페이지의 개수나 순서를 고려하고 있다.

앞의 연구와는 다르게 SSD 내부 버퍼를 읽기 버퍼로 활용한 연구도 시도되고 있다. 특히 Zhiguang Chen [24] 은 SSD 내부 채널들 사이에서 수행되는 워크로드의 불균형을 고려하였다. 여러 개의 채널들 중 특정 채널들은 읽기 연산이 집중된 핫 데이터(Hot data)를 가지고 있을 수 있는데 여기에서 제안한 SAC 기법은 채널들을 제외한 나머지 채널들에서 희생자 페이지를 결정함으로써 읽기 연산을 수행하는데 평균적으로 소요되는 지연시간을 감소시키고자 한다. 이와 같이 SSD를 기반으로 한 버퍼 관리 기법들은 주로 플래시 메모리의 쓰기 및 지우기 성능과 이로 인해 발생되는 가비지 컬렉션에 대한 오버헤드를 줄이는데 목적으로 두고 있으며 SSD의 다중 채널 구조를 이용한 버퍼 관리 기법들은 플래시 메모리의 직접도 기술과 다중 채널 기술이 끊임없이 발전되므로 이를 고려한 다양한 버퍼 관리 기법들이

제시될 수 있음을 알 수 있다.

3.3 NVRAM의 특성을 이용한 버퍼 관리 기법

사실 NVRAM의 개념은 매우 오래된 것으로, NVRAM을 쓰기 버퍼로 활용해 느린 하드 디스크의 I/O 성능을 높이려는 시도는 새로운 것이 아니다. 1988년 발표된 Menon [25] 의 논문에서 이미 유닉스 환경에서 파일 시스템의 쓰기 연산 성능을 향상시키기 위해 NVRAM을 사용, DRAM과 하드 디스크 사이의 캐시로 구성하고, NVRAM에 대한 쓰기 연산이 완료되면 하드 디스크에 대한 쓰기가 완료되지 않더라도 사용자 어플리케이션 혹은 데이터베이스에 완료 프로토콜을 전송하게 된다. 이를 네트워크 파일 시스템에 적용한 Lyon [26] 의 논문은 비교적 작은 크기의 블록을 쓸 때도 동기화된 쓰기 연산을 수행해야만 하는 네트워크 파일 시스템의 프로토콜적 특성으로 인한 성능 저하를, 파일 시스템을 구동하고 있는 원격 서버에 NVRAM을 위치시킴으로써 해결하고 있다. Baker [27] 가 발표한 논문에서는 로그 기반 파일 시스템(Log-structured file system, LFS)에서 발생하는 세그먼트의 단편화 현상으로 인해 증가하는 하드 디스크로의 접근을 줄이기 위해 NVRAM을 통해 일정 크기의 세그먼트를 채운 후에 디스크로 전송하여 충분한 대역폭을 활용할 수 있게 하였다. 지금까지의 연구는 하드 디스크를 기반으로 NVRAM을 활용한 쓰기 버퍼 기법의 연구에 해당하는 것으로, 2000년대 들어 등장한 SSD에는 하드웨어적 특성으로 인해 곧바로 적용하기는 비효율적인 면이 있었다.

하드 디스크 기반의 쓰기 버퍼 기법에서 고려하는 요소는 NVRAM에서 하드 디스크로 전달되는 쓰기 연산의 수를 줄이기 위한 시간적 지역성의 활용과 평균적인 접근 비용의 감소를 위한 공간적 지역성의

극대화로 나타낼 수 있다. 시간적 지역성은 SSD에 대한 쓰기 버퍼 기법에서도 여전히 유효한 요소가 될 수 있으나, 공간적 지역성의 경우 디스크 상에 데이터가 저장된 위치가 접근 시간에 영향을 미치지 않는 SSD의 특성상 고려되어야 할 요소라고 볼 수 없게 되었다. 대신 SSD의 특성 중 하나인 FTL에서 제자리 덮어쓰기의 불가능으로 인해 발생되는 삭제 연산과 그로 인해 파생되는 유효 페이지들의 병합 연산을 효율적으로 다루기 위해, 버퍼 교체는 해당 삭제 연산의 단위인 블록 단위와 동일하게 구성되어야 하며, 순차 쓰기 성능이 무작위 접근에 비해 매우 빠른 경향을 고려하여 순차 쓰기가 가능한 버퍼 영역을 먼저 내리도록 고안할 필요가 있음을 전제로 한다. Park[28]는 이러한 특성을 고려하여 NVRAM에 위치한 페이지들을 하나의 클러스터 단위로 묶어 SSD로 전달할 수 있도록 하며, 이를 통해 FTL에서 병합 연산 발생을 최소화하고, 혹 병합 연산이 발생하더라도 부분 병합(Partial merge) 혹은 교환 병합(Switch merge) 연산이 발생할 수 있도록 가능한 한 순차적 쓰기 연산의 형태가 되는 희생자 클러스터를 구성하도록 하는 CLC(Cold and Largest Cluster) 정책을 제안했다. Kang [29]은 한발 더 나아가 페이지를 클러스터로 구성하는 정책과 더불어 다양한 종류의 버퍼 교체 정책을 구현하고 이를 웹 서핑, 전자메일 등에서 발생하는 워크로드를 통해 평가하여 페이지 클러스터 기반 정책의 유효성을 입증하였다.

더 나은 성능이 보장됨에도 불구하고 최근 개발된 NVRAM들이 상용적으로 발전하지 못한 이유는 가격과 수명 문제인데, Kim [30]은 이러한 문제점을 해결하는 것이 예측 가능한 미래에는 불가능할 것으로 보고, 갑작스런 전원 공급 중단 사태에도 일정 시간 전력을 공급하여 데이터의 지속성

(Durability)을 보장하는 전통적인 접근법을 통해 DRAM을 논리적으로 비휘발성인 영역과 휘발성인 영역으로 구분하여 비휘발성 영역을 NVRAM과 같이 간주하는 방법론을 제시했다. 이와 동일한 접근법을 사용한 Yoo [31]는 플래시 메모리를 저장장치로 사용하는 가장 활성화된 사용자 단말인 스마트폰 환경에서의 플래시 메모리에 대한 쓰기 버퍼 전략을 평가하고, 추가적으로 차세대 NVRAM 중의 하나인 PRAM을 직접적으로 사용했을 때의 성능 평가를 통해 NVRAM의 미래를 점치고 있다. 또한 SQLite의 WAL(Write-Ahead Logging) 동작으로 인해 발생되는 작은 크기의 I/O로 인한 부하를 더 빠른 자연 대기 성능을 보이는 PRAM을 통해 완화하고, 플래시 메모리에 대한 쓰기 횟수를 줄여 수명 문제를 해결하려 한다.

이와 같이 NVRAM을 활용한 버퍼 관리 전략은 메모리 소자의 발전에 따라 다양한 특성을 고려하여 특화된 형태로 발전하고 있음을 확인할 수 있다. 이제까지의 연구에서 제시한 기법들이 플래시 메모리의 특성에 입각해 연구되었음에 반해, 현재의 차세대 NVRAM들 또한 수명문제를 본질적으로 갖고 있으므로, 쓰기 버퍼로 활용 시 특정 비트에 대한 덮어쓰기 연산 발생 시 해당 비트의 수명이 먼저 다하는 문제가 발생할 수 있다. 이러한 메모리 소자의 특성을 다각도로 고려한 버퍼 관리 연구는 수명 문제와 쓰기 속도 문제가 극적으로 개선되지 않는 한 지속적으로 유망할 것으로 판단된다.

4. 결론 및 향후 과제

현대 컴퓨터 구조의 확립 아래로 메인 메모리와 하드 디스크의 성능 차이를 메우기 위한 노력의 일환으로 NAND 플래시 메모리, 그리고 PRAM,

STT-MRAM, ReRAM과 같은 NVRAM 계열의 메모리 소자 등 다양한 종류의 비휘발성 메모리 소자들이 등장하고 있다. 그럼에도 불구하고 제한된 수명과 부족한 성능 등으로 인하여 아직까지 DRAM을 대체할 수 있는 비휘발성 메모리 소자는 개발되지 않고 있으며, 예측 가능한 미래에도 DRAM을 완전히 대체하기란 불가능할 것으로 예상되고 있다. 그러나 플래시 메모리 또는 NVRAM 계열의 차세대 저장 장치가 발전함에 따라 하드 디스크 대비 빠른 성능을 끌어내기 위하여 제한된 수명 문제를 다양한 방식으로 억제하면서 시스템에 적용하려는 시도가 최근에도 계속되고 있다. 또한, 운영체제나 데이터베이스 시스템 상에서 앞서 언급한 차세대 저장장치가 하드 디스크를 대체하여 사용되거나 쓰기 베퍼로 활용하려는 등의 시도를 통해 성능 향상을 꾀하고 있다.

현재까지 개발된 메모리 소자들의 경우, 각각이 갖는 물리적 혹은 논리적 특성으로 인해 하드 디스크에 최적화된 방식을 그대로 적용할 경우, 원하는 수준의 성능 향상을 보이기 어려울 뿐만 아니라 수명이 단축되는 문제가 있었다. 플래시 메모리는 쓰기 연산의 수행 속도가 읽기 연산에 비해 확연히 느리고 셀마다 쓰기 및 삭제 연산의 횟수가 제한되어 있다는 단점을 가지고 있다. 이러한 문제를 해결하기 위하여 메인 메모리를 베퍼로 할당하여 플래시 메모리의 장점을 이용하고 단점을 개선하려는 연구가 계속해서 진행되어 왔다. 최근에는 플래시 메모리 기반 SSD의 물리적, 논리적 특성을 더 다양한 측면에서 고려한 베퍼 교체 기법들이 연구되고 있다. 이러한 예로는 SSD의 내부 병렬성과 FTL의 동작 방식을 고려하여 성능을 향상시키려는 시도가 대표적이다. NVRAM의 경우에는 바이트 단위의 연산을 제공하고 NAND 플래시 메모리보다 지연대기 시간

이 빠르다는 점에서 주목 받았으나, DRAM에 비해 쓰기 연산이 느리고 수명이 제한적이기 때문에 근시일 내에 DRAM을 대체하기는 힘들 것으로 예측된다. 이로 인해 DRAM을 대체하기 보다는 SSD와 DRAM 사이의 쓰기 베퍼로 활용하여 DRAM에 근접한 수준의 쓰기 속도를 보이기 위한 기법들이 발표되고 있는 추세이다.

이 외에도 각 CPU 별로 특정 메모리가 속한 채널을 통해 SSD로 접근하는 메모리 채널 플래시 저장장치(Memory channel flash storage)와 DIMM 모듈에 NAND 플래시 메모리나 PRAM과 같은 비휘발성 메모리를 결합한 NVDIMM(Non-volatile dual in-line memory module)이 새로운 저장장치로 주목받고 있다. NVDIMM은 전원 공급을 차단하면 데이터가 사라지는 DRAM과 달리, 데이터를 영구 저장할 수 있는 비휘발성 메모리가 결합되었으므로 예기치 못한 전원 손실이 발생했을 경우에도 데이터가 안전하게 저장되고 복구할 수 있다. 또한, CPU와 직접 통신하는 DIMM 규격을 가지며 DRAM이 일종의 캐시 메모리 역할을 수행하기 때문에 많은 비용을 들이지 않고서도 데이터 처리 속도를 높이는 것이 가능하다. 그러나 이러한 연구 흐름은 수명 문제 및 쓰기 지연 속도를 극복하고 DRAM을 완전히 대체하기 전까지는 지속적으로 연구될 것으로 판단된다.

참고 문헌

- [1] Mu-Tien Chang, Rosenfeld, P, Shih-Lien Lu, Jacob, B, "Technology comparison for large last-level caches (L 3 Cs): Low-leakage SRAM, low write-energy SiT-RAM, and refresh-optimized eDRAM", In

- Proceedings of the 19th International Symposium on High Performance Computer Architecture, pp. 143-154, 2013
- [2] Y.Xie, "Emerging NVM Memory Technologies", Presentation, Department of Computer Science and Engineering, Pennsylvania State University, USA, 2010.
- [3] Roberto Bez, Emilio Camerlenghi, Alberto Modelli, Angelo Visconti, "Introduction to Flash memory", IEEE, Vol.91, no.4, pp.489-502, 2003
- [4] Yang Hu et al., "Exploring and exploiting the multilevel parallelism inside SSDs for improved performance and endurance", IEEE Transactions on Computers, Vol.62, no.6, pp.1141-1155, 2013
- [5] Hyojun Kim et al., "Evaluating phase change memory for enterprise storage systems: A study of caching and tiering approaches", In Proceedings of the 12th USENIX Conference on File and Storage Technologies. pp.33-45, 2014
- [6] Dmytro Apalkov et al., "Spin-transfer torque magnetic random access memory (STT-MRAM)", ACM Journal on Emerging Technologies in Computing Systems (JETC), Vol.9, no.2, 2013
- [7] James L. Peterson, Abraham Silberschatz, "Operating system concepts (2nd ed.)", Addison-Wesley Longman Publishing Company, pp.201-256, 1985
- [8] BELADY, L.A, "A study of replacement algorithms for virtual storage computers", IBM Systems, Vol.5, no.2, pp.78-101, 2010
- [9] Mr.C.C.Kavar, Mr. S.S.Parmar, "Performance Analysis of LRU Page Replacement Algorithm with Reference to different Data Structure", Engineering Research and Applications, Vol.3, no.1, pp.2070-2076, 2013
- [10] Heeseung Jo et al., "FAB: Flash-aware Buffer management policy for portable media players", IEEE Transactions on Consumer Electronics, Vol.52, no.2, pp.485-493, 2006
- [11] Seon-yeong Park et al., "CFLRU: A Replacement Algorithm for Flash memory", In Proceedings of the 2006 international conference on Compilers, architecture and synthesis for embedded systems(CASES), pp.234-241, 2006
- [12] Hoyoung Jung et al., "LRU-WSR: integration of LRU and writes sequence reordering for flash memory", IEEE Transactions on Consumer Electronics, Vol.54, no.3, pp.1215-1223, 2008
- [13] Yi Ou, Theo Harder, Peiquan Jin, "CFDC: a flash-aware replacement policy for database buffer management", In Proceedings of the 5th International Workshop on Data Management on New Hardware, pp.15-20, 2009
- [14] Zhi Li et al., "CCF-LRU: a new buffer replacement algorithm for flash memory", IEEE Transactions on Consumer Electronics, Vol.55, no.3, pp.1351-1359, 2009
- [15] Bo-Kyeong Kim, Dong-Ho Lee, "LSF: a new buffer replacement scheme for flash memory-based portable media players", IEEE Transactions on Consumer Electronics, Vol.59, no.1, pp.130-135, 2013
- [16] Victor F. Nicola, Asit Dan, Daniel M. Dias, "Analysis of the generalized clock buffer replacement scheme for database transaction processing", In Proceedings of the 1992 ACM

- SIGMETRICS joint international conference on Measurement and modeling of computer systems, pp.35-46, 1992
- [17] Biplob Debnath, Sunil Subramanya, David Du, David J. Lilja, "Large Block CLOCK (LB-CLOCK): A write caching algorithm for solid state disks", In Proceedings of Modeling, Analysis&Simulation of Computer and Telecommunication Systems, pp.1-9, 2009
- [18] Hyojun Kim, Moonkyung Ryu, Umakishore Ramachandran, "What is a good buffer cache replacement scheme for mobile flash storage?", In Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE joint international conference on Measurement and Modeling of Computer Systems, pp.235-246, 2012
- [19] Dong Hyun Kang, Changwoo Min, Young Ik Eom, "TS-CLOCK: temporal and spatial locality aware buffer replacement algorithm for NAND flash storages", In Proceedings of the ACM international conference on Measurement and modeling of computer systems, pp.581-582, 2014
- [20] Sungkyu Park, Youngwoo Park, Gyudong Shim et al., "CAVE: channel-aware buffer management scheme for solid state disk", In Proceedings of the 2011 ACM Symposium on Applied Computing, pp.346-353, 2011
- [21] Jinho Seol et al., "A buffer replacement algorithm exploiting multi-chip parallelism in solid state disks", In Proceedings of the 2009 international conference on Compilers, architecture, and synthesis for embedded systems, pp.137-146, 2009
- [22] Xufeng Guo, Jianfeng Tan et al., "PAB: parallelism-aware buffer management scheme for NAND-based SSDs", In Proceedings of the 21st International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems(MASCOTS), pp.101-110, 2013
- [23] R Wang et al., "CAWR: Buffer Replacement with Channel-Aware Write Reordering Mechanism for SSDs.", ETRI, Vol.37, no.1, pp.147-156, 2015
- [24] Z Chen, N Xiao, F Liu, "Sac: rethinking the cache replacement policy for ssd-based storage systems", In Proceedings of the 5th Annual International Systems and Storage Conference, 2012
- [25] Menon J., Hartung M., "The IBM 3990 Disk Cache", In Proceedings of COMPCON, pp.146-151, 1988
- [26] Lyon B., R. Sandberg, "Breaking Through the NFS Performance Barrier.", SunTech, pp.21-27, 1989
- [27] Baker Mary et al. "Non-volatile memory for fast, reliable file systems.", ACM SIGPLAN Notices. Vol.27. no.9. 1992
- [28] Park, S et al., "Using non-volatile ram as a write buffer for nand flash memory-based storage devices", In Proceedings of Modeling, Analysis and Simulation of Computers and Telecommunication Systems (MASCOTS), 2008
- [29] Kang S. et al., "Performance trade-offs in using NVRAM write buffer for flash memory-based storage devices.", IEEE Transactions on Computers, pp.744-758, 2009
- [30] Kim D., S. Kang, "Dual region write buffering: making large-scale nonvolatile buffer using small capacitor in SSD", In Proceedings of the 30th Annual ACM Symposium on Applied Computing, pp.2039-2046, 2015

[31] Ungjoo Yoo et al., "Reducing Mobile Storage Writes: Non-Volatile Write Buffer and PRAM-based Journaling." 정보과학회지, Vol.33, no.2, pp.52-59, 2015



이 미 경

2014년 동덕여자대학교 컴퓨터학과 학사
2014년~현재 연세대학교 컴퓨터과학과 석사

관심분야 : 데이터베이스, 버퍼 관리 기법, 플래시메모리



김 재 혼

2007년 한양대학교 컴퓨터공학과 학사
2009년 한양대학교 컴퓨터공학과 석사
2013년~현재 연세대학교 컴퓨터과학과 박사

관심분야 : 데이터베이스, 스토리지, 빅데이터



박 상 현

1989년 서울대학교 컴퓨터공학과 학사
1991년 서울대학교 대학원 컴퓨터공학과 공학석사

1991년 UCLA 대학원 컴퓨터과학과 공학박사
1991년~1996년 대우통신 연구원
2001년~2002년 IBM T. J. Watson Research Center Post-Doctoral Fellow

2002년~2003년 포항공과대학교 컴퓨터공학과 조교수
2003년~2006년 연세대학교 컴퓨터과학과 조교수
2006년~2011년 연세대학교 컴퓨터과학과 부교수
2011년~현재 연세대학교 컴퓨터과학과 교수

관심분야 : 데이터베이스, 데이터마이닝, 바이오인포매틱스, 적응적 저장장치 시스템, 플래시 메모리 인덱스, SSD