Solid-State Disk 성능에 영향을 미치는 내장 소프트웨어 설계요소 분석

포항공과대학교 ▌류준길 · 박찬익*

1. 서 론

Solid-State Disk(SSD)는 기존 HDD(Hard Disk Drive)와 동일한 인터페이스(PATA, SATA 등)를 지원하도록 여러 개의 NAND 플래시 메모리 칩들을 이용해서 구현된 대용량 저장 매체이다. SSD는 빠른 데이터 처리가 가능하며, 전력 소모, 발열, 소음 등이 낮아 빠르게 모바일 시장에서 적용되고 있다. HDD의 경우에는 모터와 같은 기계 부품에서 걸리는 소요시간이 매우 크기 때문에 상대적으로 내장 소프트웨어의 영향이 적어서 제품별 성능 차이가 크지 않는 데 반해, SSD의 경우에는 하부 NAND 플래시 메모리 관리를 담당하는 내장 소프트웨어 영향이 매우 크다. 본 논문에서는 SSD 성능을 향상시키기 위해서 내장 소프트웨어 설계시 고려해야 될 요소들을 살펴보도록 하겠다.

논문의 구성은 SSD의 저장매체로 사용되고 있는 NAND 플래시 메모리의 특성을 살펴보고, HDD를 성능 면에서 능가하기 위해서는 다수의 NAND 플래시 메모리 칩들이 요구되는데 이를 효과적으로 운영할 수 있는 내장 소프트웨어 설계요소에 대한 설명으로 이루어진다.

2. NAND 플래시 메모리의 특성과 SSD의 일반적인 하드웨어 구조

2.1 NAND 플래시 메모리 특성

SSD는 NAND 플래시 메모리 소자에 데이터를 저장하기 때문에, NAND 플래시 메모리의 I/O특성(쓰기, 읽기, 지움)에 많은 영향을 받는다. 그림 1은 NAND 플래시 메모리 칩이 페이지, 블록, 플래인 및 다이들로 구성되어 있는 것을 보여주고 있다[5]. NAND 플래시 메모리는 HDD와 달리 쓰기 위해서는 해당 영역이 미리 지워져 있어야 하기 때문에 성능을 위해서 기존 데이터 영역에 갱신되는 데이터를 덮어 쓸 수가 없다(out-of-place update). 여기서 말하는 지움 동작은 메모리 셀의 상태를 1로 변경하는 작업을, 쓰기 동작은 메모리 셀의 상태를 0으로 변경함으로써 데이터를 저장하게 하는 작업이다. 또한 NAND 플래시 메모리는 읽기와 쓰기의 기본 단위는 플래시 페이지로 이루어지지만 지움 동작은 여러 개의 플래시 페이지들로 구성된 블록 단위로 구성되며, 각 블록 지움 횟수가 제한되어 있기 때문에 지움 동작을 NAND 플래시 메모리 전체에 균등하게 분포하도록 하는 wear-leveling 기능이 중요하다. NAND 플래시 메모리 타입에는 하나의 메모리 셀에 하나의 비트만을 표현할 수 있는 SLC(Single Level Cell) 타입과 두 개 혹은 네 개의 비트를 표현하는 MLC(Multi Level Cell) 타입이 있다. 이렇게 하나의 메모리 셀에 다양한 양의 데이터를 저장하는 방법은 메모리 셀의 voltage level을 얼마나 세분화되게 구분할 수 있느냐에 따라 달려 있다. SLC

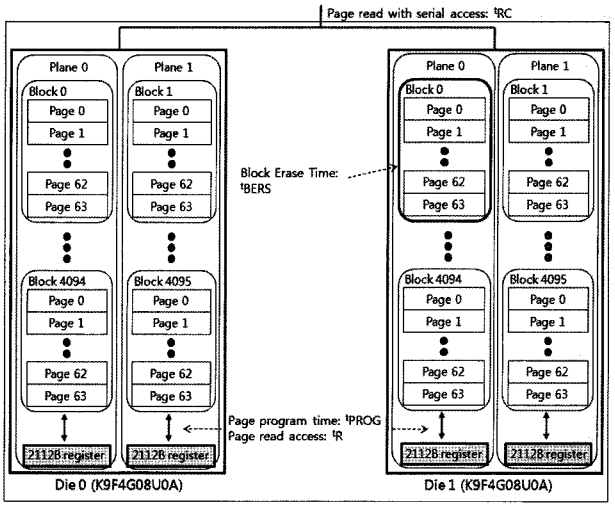


그림 1 NAND 플래시 메모리 칩(K9K8G08U0A)의 개략도 (K9K8G08U0A)는 두 개의 K9F4G08U0A로 구성되어짐

* 종신회원

의 경우 voltage level이 두 개(0, 1)만 존재하기 때문에 하나의 reference point를 이용하여 구분 할 수 있다. 그러나 MLC의 경우 voltage level이 4개(00, 01, 10, 11) 있기 때문에 3개의 reference points를 이용하여 구분 할 수 있다. 이러한 이유 때문에 SLC와 MLC 사이에는 읽기/프로그래밍 성능 차이뿐만 아니라 내구성에도 차이가 있다[1,2].

그림 2는 현재 시판되고 있는 SLC/MLC NAND 플래시 메모리 제품들의 성능 및 신뢰도 비교를 간단하게 보여준다[1,2]. 데이터 읽기 동작의 경우, 메모리 각 페이지로부터 page register로 읽혀지고, page register로부터 외부 메모리로 사이클 당 한 바이트 씩 읽혀지게 된다(그림 1). 최근 SLC NAND 플래시 메모리의 경우, page register로 읽혀지는 시간은 ^{t}R로 표시되고 페이지 당 25μs, page register에서 외부 메모리로 읽혀지는 데 걸리는 시간은 ^{t}RC로 표시되고 바이트 당 25 ns이다. 따라서 SLC NAND 플래시 메모리 칩 하나에서 읽기 성능은 26.67MB/Sec(최대 40MB/sec)이다. 하나의 블록을 지우는 데 걸리는 시간은 ^{t}BER로 표시되고 SLC NAND 플래시 메모리의 경우 1.5msec이고 MLC NAND 플래시 메모리의 경우 거의 두 배 시간이 걸리며, 데이터 쓰기 동작과 비동기적으로 메모리 블록 지움 동작을 수행한다면 쓰기 성능에 직접적인 영향을 미치지 않는다. 쓰기의 경우, 그림 2에 표시한 것처럼 외부 메모리로부터 page register로 데이터가 전송되는 시간은 읽기의 경우와 같으나 page register로부터 NAND 플래시 메모리 셀로 프로그래밍되는 시간은 ^{t}PROG로 표시되고 200μs 정도 걸린다. 따라서 최대 쓰기 성능은 10MB/Sec이다. 여기서 최대값은 2

SLC (Single Level Cell)	NAND Flash Memory	MLC (Multi Level Cell)
Features		
1	Bits per cell	2
1 or 2	Number of planes	2
2KB or 4KB	Page Size	2KB or 4KB
64	Pages per block	128
Array Operations		
25 us	^{t}R (Max), Read access	50 us
25 ns	^{t}RC, Page read with serial access	25 ns
200~300 us	^{t}PROG (typ), Program Time	600~900 us
1.5~2 ms	^{t}BERS (typ), Block Erase Time	3 ms
Reliability		
1	ECC (per 528 bytes)	4+
~100K	Endurance (Erase/Program Cycles)	~10K

그림 2 SLC/MLC NAND 플래시 메모리 사양 비교

개의 다이로 패키징된 하나의 NAND 플래시 메모리 칩을 효과적으로 사용하였을 경우에 나오는 성능이다. 각 다이에는 명령을 따로 내릴 수 있기 때문에 외부 전송과 내부 읽기/프로그램을 상호 배치(interleaving) 할 수 있다. 그래서 NAND 플래시 메모리 내부에서 읽기/쓰기/지움 동작을 병렬적으로 수행가능하고, 언급한 최대값에 도달할 수 있는 것이다. 외부 전송 대역폭 확장 없이 여러 개의 다이들을 이용하여 패키징 하더라도 하나의 NAND 플래시 메모리 칩의 읽기 성능은 40MB/Sec (MLC NAND 플래시 메모리의 경우, 20 MB/Sec), 쓰기 성능은 20MB/Sec(MLC NAND 플래시 메모리의 경우, 5MB/Sec)를 넘지 못한다. 지움 동작의 성능은 NAND 플래시 메모리 내부에서만 수행하면 되기 때문에 읽기와 쓰기와 같은 성능 제한이 없다. 즉, 여러 개의 다이들을 패키징 할수록 지움 성능은 좋아진다.

따라서 SSD가 HDD보다 나은 성능을 보여주기 위해서는 다수의 NAND 플래시 메모리 칩들을 이용하여 효과적으로 구성하여야 한다.

2.1 SSD 하드웨어 구조

NAND 플래시 메모리기반 SSD는 그림 3처럼 호스트 인터페이스, 프로세서, SRAM, DRAM, 플래시 메모리 컨트롤러, 및 NAND 플래시 메모리 칩들로 구성되어진다. SSD는 호스트 시스템의 다양한 인터페이스(SATA, PATA, SCSI, FC)중 하나와 연결을 지원해야 하기 때문에 호스트 인터페이스 로직을 갖추고 있다. 프로세서는 디스크 논리 블록과 NAND 플래시 메모리 상의 위치 맵핑, NAND 플래시 메모리 관리(garbage collection, wear leveing)등을 처리하고, SRAM은 내부 소프트웨어 및 맵핑 데이터를 위해 사용되며, DRAM은 읽기/쓰기 요구 속도 최적화를 위해 데이터 캐시 및 버퍼로 사용한다. SSD는 성능을 위해서 여러 개의 NAND 플래시 메모리 칩들에 대한 I/Os를 병렬화하기 위해서 여러 개의 채널을 관리하는 플래시 컨트롤러를 둔다. 플래시 컨트롤러는 각 채널에 있는 NAND플래시 메모리칩들 중에서 필요한 NAND 플래시 메모리 칩을 선택하고 데이터를 전송함으로써 각 채널 별 I/O 동작을 병렬적으로 처리한다. 그림 4는 디스크 I/O 인터페이스 성능과 현재 생산되고 있는 HDD 중 디스크로 부터 호스트 시스템으로 읽기 성능이 최대인 HDD의 성능을 보여준다. HDD에서 외부 인터페이스로 주로 사용하고 있는 SCSI, SATA의 대역폭은 SCSI의 경우 320MB/sec, SATA의 경우 300MB/sec에 이르고 있고, 실제 제조사에 생산하고 있는 HDDs중에서 디스크

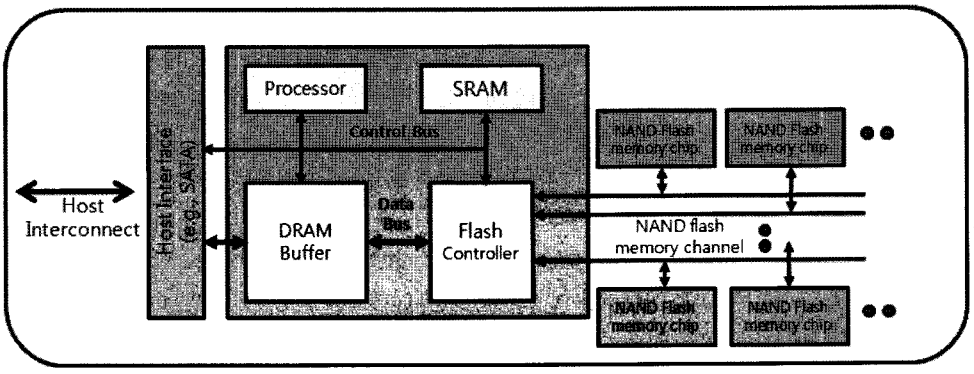


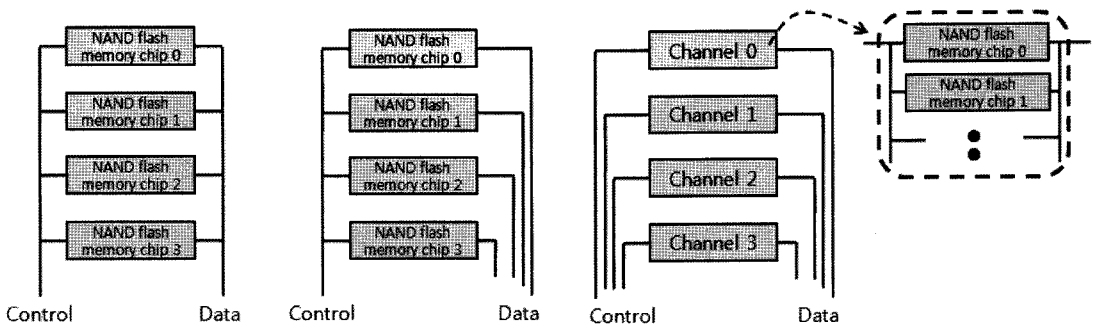
그림 3 일반적인 SSD 하드웨어 구조

Standard I/O Interface	Parallel						Serial		
	ATA			SCSI			SATA		
	ATA-5 (2000)	ATA-6 (2002)	ATA-7 (2005)	Ultra3 SCSI (1999)	Ultra-320 SCSI (2002)	Ultra-640 SCSI (2003)	SATA Rev. 1.x (2001)	SATA Rev. 2.x (2004)	SATA Rev. 3.x (2008, draft)
Maximum Data Transfer rate (MB/sec)	66.7	100	133	160	320	640	150	300	600
Maximum data transfer rate of the current HDDs (MB/sec)	120			164			120		

그림 4 HDD I/O 인터페이스 전송속도

에서 호스트로의 읽기 성능을 보게 되면 SCSI HDD의 경우 164MB/sec, SATA HDD의 경우 120MB/sec에 도달하고 있다[Seagate]. 따라서 NAND 플래시 메모리를 이용하여 HDD에 상응하는 성능을 보이기 위해서는 I/O 병렬 처리는 필수이다. NAND 플래시 메모리 칩의 최대 읽기 성능이 40MB/sec인 것을 가정하고 HDD에 상응하는 SSD를 구현하기 위해서 요구되는 NAND 플래시 메모리 채널수를 이론적으로 계산해보면 최소 4개 이상이 되어야 하며, Ultra-320 SCSI 등의 고성능 HDD 인터페이스 대역폭을 지원하기위해서는 최소 8개 이상이 되어야 한다. 물론 임의 읽기 성능을 고려한다면 NAND 플래시 메모리 칩 하나로도 SATA HDD의 성능에 상응할 수 있다.

그림 5는 SSD 내부의 다수의 NAND 플래시 메모리 칩들을 구성하는 방법을 보여준다. 그림 5의 (a) Share bus gang은 SSD 내부에 있는 NAND 플래시 메모리 칩들이 데이터 버스와 컨트롤 버스를 공유하는 것이며, (b) shared control gang은 NAND 플래시 메모리 칩들이 컨트롤 버스는 공유하되 데이터 버스는 독자적으로 가지고 있는 것이다[3]. I/O 병렬 처리를 위해서는 버스 점유시간이 적은 컨트롤 버스를 공유하는 것이 낫다. 따라서 (a)보다는 (b)가 SSD의 성능 면에서는 우수하다. (c) NAND 플래시 메모리 채널은 각 채널이 각자의 데이터 버스와 컨트롤 버스를 가지는 것으로 하나의 NAND 플래시 메모리 채널에 속하는 NAND 플래시 메모리 칩들은 (a)와 같이 데이터 버스와 컨트롤 버스를 공유한다. 그러나 서로 다른 채널에 속하는 NAND 플래시 메모리 칩들 간에는 데이터 버스와 컨트롤 버



(a) Shared bus gang (b) Shared control gang (c) NAND Flash Memory Channel

그림 5 다수의 NAND 플래시 메모리 칩들을 구성하는 방법

스를 공유하지 않기 때문에 (b)의 경우보다 나은 I/O 병렬처리를 할 수 있다. 또한 (c)의 경우 보다 나은 확장성을 가지고 있기 때문에 대부분의 SSD 제품에서 채택하고 있다[3,6-8].

3. SSD 내장 소프트웨어 설계요소 분석

3.1 SSD 내부 동작원리와 성능결과의 의미

SSD를 구성하는 NAND 플래시 메모리는 성능을 위해서 이전 데이터 영역에 갱신되는 데이터를 덮어 쓰지 못하고 디스크 I/O 블록의 크기(512Byte)와 플래시 페이지의 크기(2KB 또는 4KB)가 다르기 때문에 디스크 I/O 블록과 해당 데이터 쓰여질 페이지 위치를 나타내주는 맵핑 테이블을 내부적으로 관리하여야 한다. NAND 플래시 메모리의 지움 동작의 단위가 플래시 블록(64개 또는 128개의 페이지들)이기 때문에 garbage-collection[2] 및 그에 따른 쓰기 성능을 효과적으로 하고 맵핑 테이블의 크기를 작게 하기 위해서 여러 개의 플래시 페이지들로 구성된 관리 블록을 정의하는데, 여기서는 그것을 SSD 관리 블록(SSD Management Block)이라고 부르기로 하겠다. 이러한 SSD 관리 블록은 수퍼 블럭이라고 부르기도 한다[7,13]. SSD 관리블럭은 SSD 빠른 쓰기 성능을 위한 NAND 플래시 메모리의 병렬적 구동을 효과적으로 지원하는데 핵심적인 역할을 한다. SSD 관리블록의 크기가 증가하면 매핑테이블 크기가 감소하는 반면, 플래시 페이지 단위의 쓰기 요청에도 불구하고 in-place 수정이 불가능한 NAND 플래시 메모리 특성 때문에 플래시 페이지 단위보다 훨씬 큰 SSD 관리 블록 단위로 재매핑하고 또한 해당 SSD 관리 블록 내에 있는 다른 플래시 페이지들에 대한 복사 과정이 수반되므로 결과적으로 SSD 쓰기 성능을 저하시키는 문제를 야기한다. 이를 해결하기 위해 일반적으로 SSD에서는 SSD 관리 블록을 데이터 블록과 로그 블록으로 나누고, 데이터 블록은 SSD 관리 블록 단위의 매핑을 지원하고, 로그 블록은 NAND 플래시 메모리 페이지 단위로 매핑을 관리한다. 로그 블록을 플래시 페이지 단위로 관리할 경우 쓰기 요청은 항상 플래시 페이지 단위로 매핑을 관리하고 있는 로그 블록을 이용하여 처리하고, 향후 적정 시점(해당 로그 블록이 여유페이지가 없거나, 로그 블록의 reclaiming 이 필요한 경우)에 해당 로그 블록을 데이터 블록에 저장한다. 로그 블록을 데이터 블록으로 이동시킬 때, 어떤 데이터 블록으로 이동할 것인가는 로그 블록과 데이터 블록 간에 어떤 관계를 설정하는 가에 따라 달라진다. 예를 들어, 로그블록과 데이터 블록이 1:1 관계인 경우에는 특정 로그 블록을 해당되는 데이터 블록으로 이동하는 것으로 병합동작이 지원될 수 있으나, 이 경우 로그 블록에 쓰기 동작을 통해 수용 가능한 플래시 페이지들은 해당되는 데이터 블록에서 가지고 있을 수 있는 플래시 페이지들, 즉 순차적으로 구성되어 있는 플래시 페이지들로 국한되므로 임의 쓰기에 대해서는 성능이 매우 떨어지는 단점을 가진다. 이를 해결하기 위해서 특정 로그 블록에 써지는 페이지들에 대해 아무런 제약을 가하지 않도록 하여, 즉 하나의 로그 블록이 다수의 데이터 블록을 책임지도록 하게하여 임의 쓰기에 대한 성능 저하 문제를 해결한다. 다만, 그 로그 블록을 데이터 블록으로 이동하기 위해서 로그 블록 속에 들어있는 페이지들이 소속된 데이터 블록에 따라 각각 다른 데이터 블록으로 병합되어야 하므로 병합과정의 복잡도가 증가한다. 이는 임의 쓰기 동작 직후의 SSD 성능이 불규칙적인 현상을 야기하는 문제점이 있다.

그림 6은 임의의 작은 쓰기명령을 효과적으로 처리하기 위해서 로그 블록과 데이터 블록간의 1:1 관계를 가정하고, 디스크 I/O 블록의 데이터를 플래시 페이지로 저장하는 모습을 보여주고 있다. 우선 디스크 I/O 블록을 위해서 데이터 영역에 있는 비어있는 SSD 관리 블록(데이터 블록)을 하나 선정한다. 데이터는 로그 영역에 있는 SSD 관리 블록(로그 블록)에 데이터를 쓴다(그림 6에서 (1)-1). 후에 로그 블록에 있는 데이터를 선정된 데이터 블록으로 이동하게 되는데(그림 6에서 (1)-2), 디스크 I/O 블록이 SSD 관리 블록 내에서 위치할 상대적인 플래시 페이지를 정하고 해당 페이지에 데이터를 저장한다(그림 6에서 (2) 플래시 페이지로 데이터 저장). SSD 관리 블록은 플래시 메모리 채널 상에 있는 플래시 페이지들로 구성되어지는데, 요청 받은 디스크 I/O 블록의 데이터를 해당하는 채널 상의 플래시 페이지에 쓰게 된다. 그림 6의 예는 n개의 플래시 메모리 채널을 가지고 있는 SSD에서 SSD 관리 블록을 구성하는 플래시 페이지가 각 채널 상에 있는 플래시 메모리에 하나씩 고르게 분포되어 있다는 것을 가정하였다. SSD의 내부 관리 단위가 SSD 관리 블록이기 때문에 데이터 갱신시 SSD 관리 블록 전체에 대해서 갱신이 이루어져야한다.

2) Garbage-collection은 NAND 플래시 메모리의 out-of-place update로 인해서 발생하는 각 invalid 페이지들을 지워서 추후에 사용할 수 있도록 하는 것이다. 지움 단위가 64개 또는 128개의 페이지들로 구성된 블록이기 때문에 효과적으로 invalid 페이지들과 valid 페이지들을 구분하여 모아야 한다.

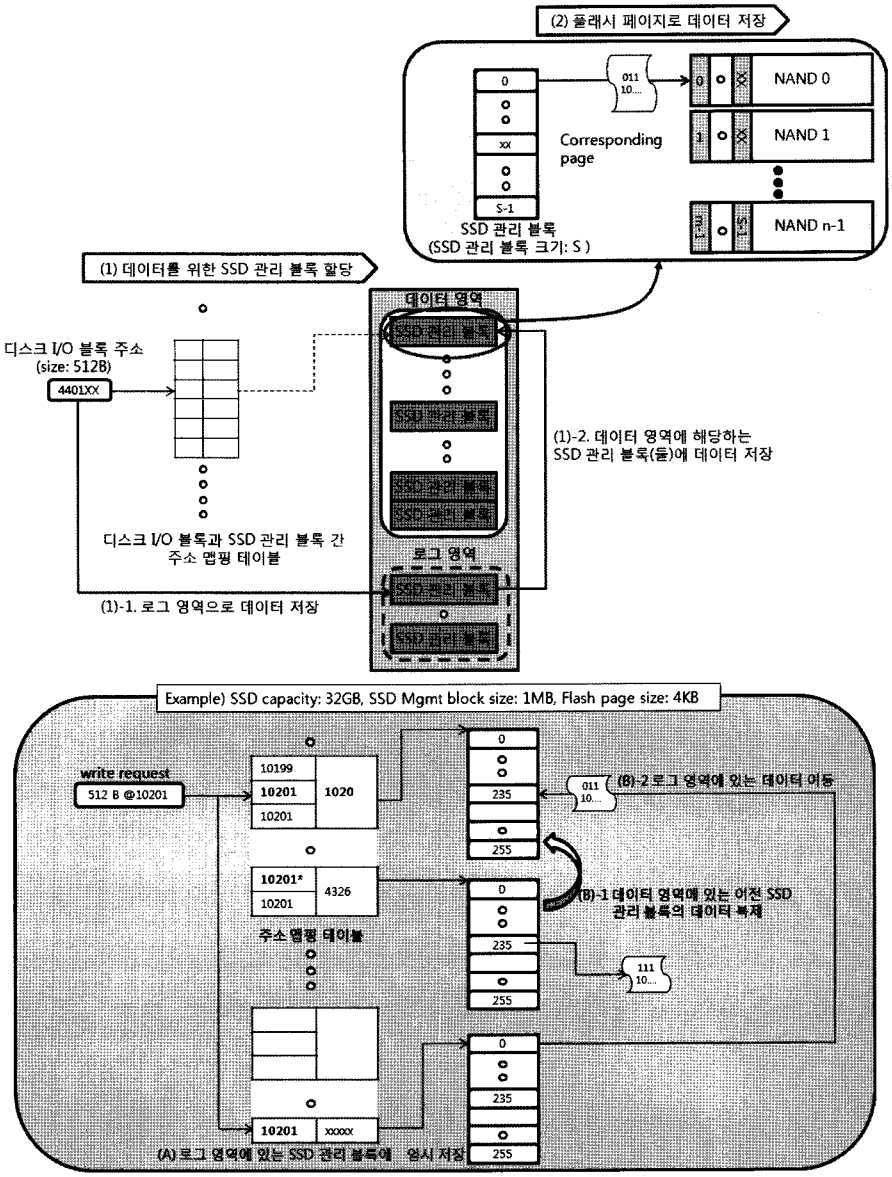


그림 6 디스크 I/O 블록, SSD 관리 블록 그리고 플래시 페이지 간의 관계 (n개의 플래시 메모리 채널을 가지고 로그블록과 데이터 블록이 1:1 관계인 SSD를 가정)

그림 6의 예를 들어 설명하면 다음과 같다. SSD의 용량이 32GB이고, SSD 관리 블록의 크기가 1MB이면 해당 SSD에는 32×1024개의 SSD 관리 블록이 존재하고, 각 SSD 관리 블록에는256개의페이지(플래시 페이지 크기가 4KB라고 가정)가 존재하게 된다. 10201 주소번지에 크기가 512B인 쓰기 명령이 온다면, SSD는 내부의 사용되지 않고 있는 임의 SSD 관리 블록 중에서 하나를 선택하고, 10201×1/2 → 5040 KB + 512 B(4MB + 944KB + 512B)이기 때문에, 해당 SSD 관리 블록에 속하는 236번째(944KB × 1/4 → 236) 플래시 페이지를 해당 쓰기 명령을 위해서 할당한다.

쓰기 요구에 해당하는 양(512B)이 하나의 플래시 페이지(4KB)보다 작기 때문에 해당 데이터만을 프로그래밍하는 partial programming을 사용할 수도 있지만 partial programming은 순차쓰기에만 가능하고 플래시 페이지에서 임의 순서로 쓰는 것은 불가능하다. 또한 플래시 페이지에 partial programming은 신뢰도를 저하시키기 때문에 권장되지 않는다. 그리고 이전에 해당 disk I/O block을 관리하던 SSD 관리 블록과 그에 속한 플래시 페이지들에 데이터가 있기 때문에 관련 플래시 페이지들(SSD 관리 블록@4326)에 있는 내용을 읽어 와서 새로이 할당받은 SSD 관리 블록(@1020)에 쓰기 명령의 데이터와 함께 프로그래밍 해야 한다(예의 (B)-1, (B)-2). 따라서 이러한 동작으로 발생하는 오버헤드를 줄이기 위해서 데이터를 로그 블록에 임시로 저장한 후에 추후에 데이터 블록으로 이동시킨다

(예의 (A)). 그런 후에 이전에 할당 받은 SSD 관리 블록(@4326)은 무효화되어 garbage collection 대상이 된다.

따라서 고성능 SSD를 위한 내장 소프트웨어 설계 시에는 다음과 같은 점들에 관심을 가져야 한다.

- SSD 관리 블록의 크기?
- 디스크 I/O 블록들을 SSD 관리 블록 내의 페이지들에 어떻게 배치할 것인가?
- SSD 관리 블록을 구성하는 페이지들을 채널 상의 NAND 플래시 메모리 채널상의 플래시 메모리 칩들을 이용하여 어떻게 할당할 것인가?

위와 같은 점들이 실제 성능에 영향을 끼치는 지를 알아보기 위해서 SSD A와 SSD C를 이용하여 구입 초기와 3개월 사용 후 성능을 iozone benchmark tool을 이용하여 측정하였다. 실험에 사용한 SSD의 사양은 그림 7과 같다[6-8].

그림 8과 그림 9는 SSD A와 C의 성능 결과이다. 채널의 수, 플래시 메모리 칩의 수, 메모리 버퍼의 크기 차이를 고려하더라도 다음과 같은 점들에 주목하여야 한다.

- SSD A의 순차 쓰기와 임의 쓰기 간의 성능 차이가 큼 참고로 HDD는 기계적 특성으로 인해서 디스크 헤더가 해당 주소로 찾아가는 데 걸리는 오버헤드가 크기 때문에 임의 쓰기의 성능이 순차 쓰기의 성능보다 매우 낮다. 그러나 SSD의 경우 해당 주소로 찾아가는데 오버헤드가 거의 없지만 순차 쓰기와 임의 쓰기간 성능 차이가 크다.
- SSD A와 달리 SSD C의 경우 순차 쓰기와 임의 쓰기 간의 성능 차이가 작음

	SSD A	SSD B	SSD C
NAND flash memory type	SLC	SLC	MLC
Memory buffer size	32 MB	16 MB	16 MB
Physical page size	4KB	2KB	4KB
Number of channels	4	4	10
Number of flash memory chips	8	16	20

그림 7 SSD A, B, C의 하드웨어 사양

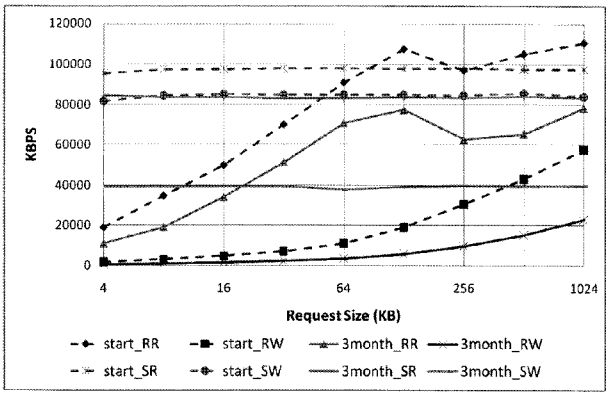


그림 8 SSD A의 성능

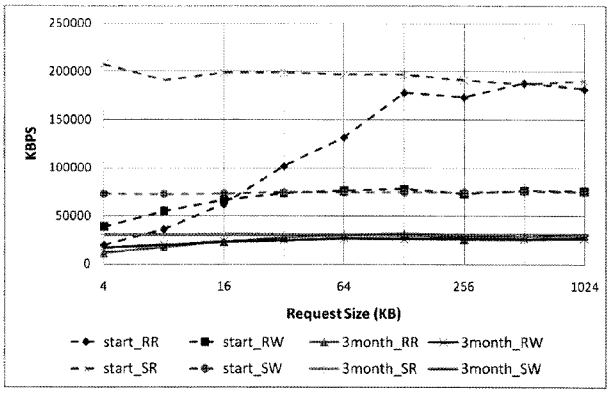


그림 9 SSD C의 성능

- SSD A의 경우, (순차/임의) 읽기 성능이 구입 초기와 사용 3개월 후에 성능을 비교했을 때, 전체적으로 감소하지만 SSD C에 비해서는 크게 감소하지 않았다. SSD C의 읽기 성능은 심각할 정도로 매우 감소한 것을 알 수 있다.

3.1 SSD 관리 블록 크기

그림 10은 SSD 관리 블록으로 인해서 발생하는 오버헤드를 보여주는 것이다. 그림 10의 (1)은 SSD 관리 블록의 크기보다 작은 크기의 데이터 갱신 요청이 SSD에 내려올 때, 해당 갱신 요청이 하나의 SSD 관리 블록 안에 해당할 경우를 보여주는 것이고, (2)는 해당 갱신 요청이 두 개의 SSD 관리 블록들에 걸치는

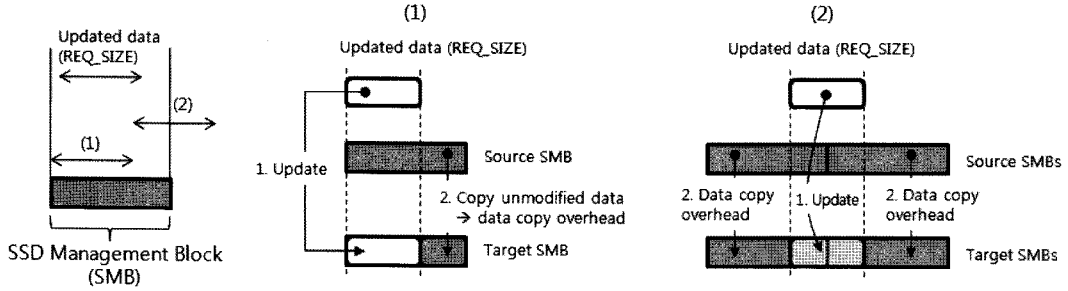


그림 10 데이터 갱신시 SSD 관리 블록으로 인해서 발생하는 오버헤드

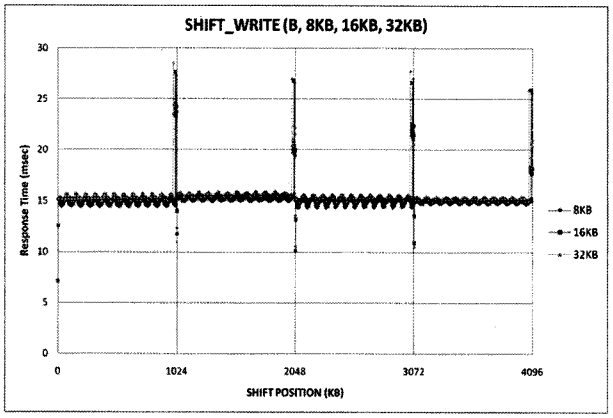


그림 11 SSD B의(8, 16, 32KB) 크기의 이동 쓰기에 따른 반응시간

경우를 보여주는 것이다. 해당 SSD 관리 블록(들)에 속하는 갱신되지 않는 데이터 부분들도 SSD 관리 블록이 SSD 내부 관리 단위이기 때문에 복제를 해야 한다. 이것이 SSD 관리 블록의 크기가 쓰기 요청의 크기보다 크기 때문에 발생하는 오버헤드이다. 여기서 알 수 있는 것으로는 첫째, SSD 관리 블록의 크기가 클수록 쓰기 요청에 따른 데이터 복제 오버헤드가 증가한다는 것이고 둘째, (2)의 경우가 (1)의 경우에 비해서 2배 이상의 오버헤드가 발생하기 때문에 이를 이용하여 SSD 관리 블록의 크기를 추정해 볼 수 있다는 것이다.

이를 이용하여 SSD B의 SSD 관리 블록의 크기를 알아보기 위해서 일정한 크기의 쓰기 요청을 일정한 크기로 이동해가면서 반응시간을 측정하면 그림 11과 같은 결과를 얻을 수 있다. 실험은 오차를 줄이기 위해서 SSD를 다수의 파티션으로 구분한 후에 각 파티션에서 동일한 실험들을 반복한 것이다.

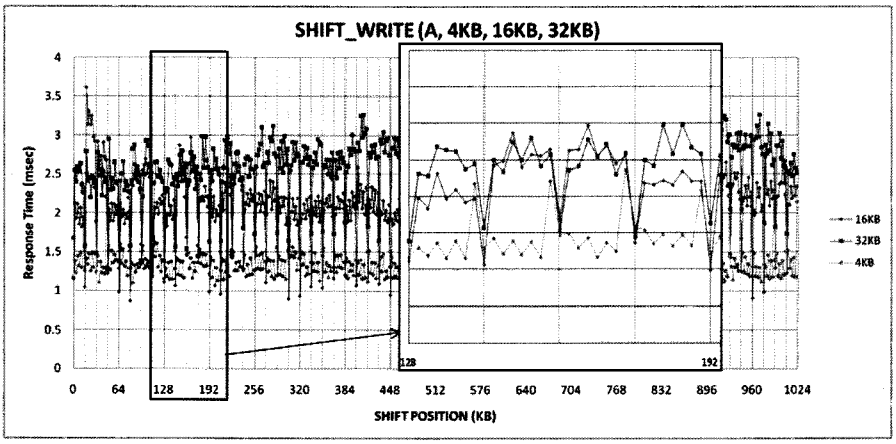


그림 12 SSD A의 (4, 16, 32 KB) 크기의 이동 쓰기에 따른 반응 시간

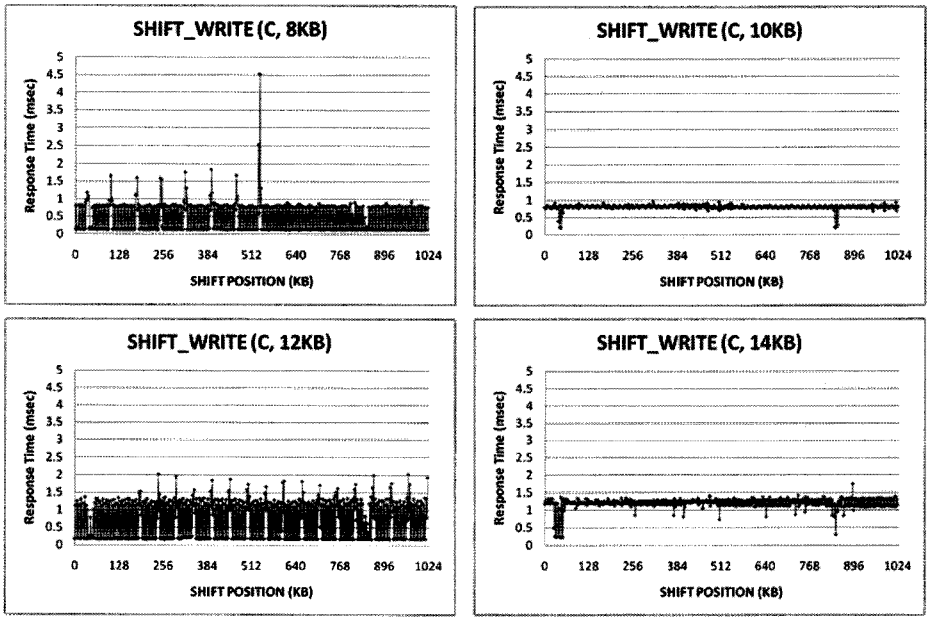


그림 13 SSD C의 (8, 10, 12, 14 KB) 크기의 이동 쓰기에 따른 반응 시간

	Random Read					Random Write				
Request Size (KB)	4	8	16	32	64	4	8	16	32	64
A	19045	34822	49910	70103	90915	1699	2962	4734	7108	10941
B	22828 (1.20)	35904 (1.03)	50179 (1.01)	62035 (0.88)	70064 (0.77)	607 (0.36)	1225 (0.41)	2445 (0.52)	4752 (0.67)	8965 (0.82)
C	19685 (1.03)	35940 (1.03)	61933 (1.24)	101541 (1.45)	131501 (1.45)	38756 (22.81)	55070 (18.59)	66421 (14.03)	73405 (10.33)	75905 (6.94)

그림 14 작은 크기 읽기/쓰기 요청의 성능: 괄호 안의 값은 SSD A의 성능에 대한 상대적인 성능임

그림 11을 보게 되면 매 1MB마다 반응 시간이 크게 오르는 것을 알 수 있다. 이것은 SSD B의 관리 블록 크기가 1MB임을 보여주는 것이다. 이와 같은 실험을 SSD A와 C에도 수행했을 경우 SSD A는 16KB, SSD C는 4KB 크기의 SSD 관리 블록을 갖는 다는 것을 알 수 있었다(그림 12, 13).

앞에서 SSD 관리 블록의 크기가 클수록 데이터 복제 오버헤드가 커질 것이라고 말을 하였다. 이를 알아보기 위해서 iozone benchmark tool을 이용하여 각 SSD의 성능을 측정하였다.

그림 14는 성능 측정 결과이다. 임의 쓰기 성능을 보게 되면 SSD C, A, B순으로 나타남을 알 수 있다. SSD C은 채널수가 10개(20개의 플래시 메모리 칩)이기 때문에 직접적인 비교를 할 수 없지만 쓰기에 걸리는 시간이 SLC NAND 플래시 메모리에 비해서 거의 2~3배인 MLC NAND 플래시 메모리를 사용한다는 점을 고려해 볼 때, 작은 크기의 SSD 관리 블록으로 인해서 상대적인 성능 향상 효과를 보이는 것을 알 수 있다. SSD B의 임의 쓰기 성능은 SSD A에 비해서, 각 SSD 관리 블록의 크기(1MB/16KB)를 고려할 때에 생각보다 성능 저하가 크지 않음을 알 수 있다. 이를 통해서 작은 임의 쓰기를 임시로 쓸 수 있는 로그 영역에 SSD 관리 블록을 할당하여 임의 쓰기로 인한 성능저하를 최소화하려고 하였다는 것을 알 수 있다[9,10]. 그림 6은 작은 임의 쓰기로 인한 성능 저하를 최소화하기 위해서 로그 영역을 사용하는 것을 보여주고 있다. 작은 크기의 임의 쓰기가 오면 로그 영역에 SSD 관리 블록(로그 블록)을 할당하고 후에 데이터 영역에 있는 SSD 관리 블록(데이터 블록)으로 데이터를 이동시킨다. 여기서 고려해야 될 중요한 문제는 로그 영역에 있는 SSD 관리 블록과 데이터 영역에 있는 SSD 관리 블록 간의 관계이다. 로그 블록과 데이터 블록간의 관계를 정하지 않으면 후에 로그 블록에 있는 데이터를 이동시에 오버헤드가 발생할 수 있다. 이와 달리 다수의 데이터 블록들과 로그 블록을 일정한 관계를 가지게 한다면 로그 블록 할당 및 로그 블록에 있는 데이터를 추후 데이터 블록으로 이동시키는데 효과적일 수 있다. 그러나 이에 따라 로그 블록들의 공간이 낭비될 수도 있다.

참고로 작은 크기의 읽기 명령에서 SSD B의 임의 읽기 성능이 SSD A의 읽기 성능보다 나은 것은 SSD B를 구성하는 플래시 페이지의 크기가 2 KB로 SSD A의 4 KB에 비해서 작기 때문에 더 나은 I/O 병렬처리에 기인하는 것으로 보인다.

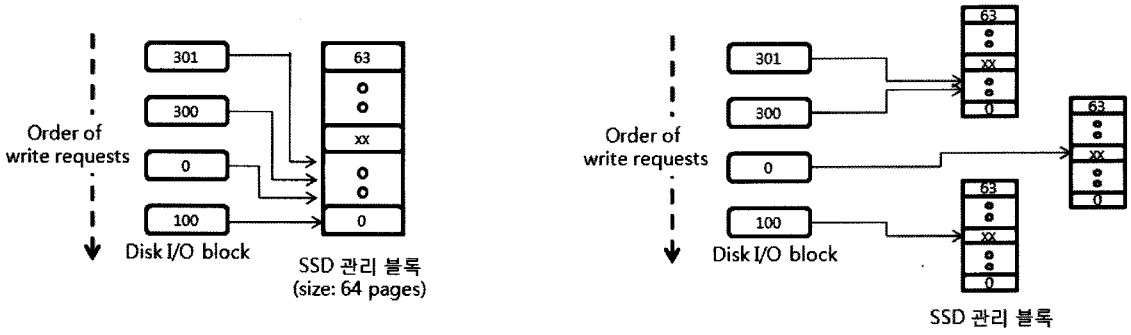
SSD 관리 블록의 크기가 크게 되면 맵핑 테이블이 간소화되고 읽기 명령에 해당하는 주소를 검색이 빨라지기 때문에 읽기 성능이 향상된다. 또한 garbage-collection이나 wear-leveling 기법이 효과적이며 간소화되기 때문에 시간에 따른 성능 저하현상이 적어지게 된다. 그림 8, 9와 같은 성능 저하 현상은 SSD B에서는 거의 발생하지 않았다. 그러나 SSD 관리 블록이 큰 것이 반드시 나은 것은 아니다. MLC NAND 플래시 메모리의 경우 SLC에 비해서 쓰기 성능이 떨어지기 때문에 SSD 관리 블록 크기로 인한 오버헤드는 더욱 크게 보여질 것이다. SSD 관리 블록의 크기를 크게 할 경우, 로그 버퍼관련 크기 및 알고리즘에 신중을 기하여야 한다.

3.2 디스크 I/O 블록들을 SSD 관리 블록의 페이지에 배치하는 방법

디스크 I/O 블록들을 SSD 관리 블록에 배치하는 방법은 그림 15와 같다.

(a) 쓰기 명령 순서처럼 쓰기 명령이 내려오는 순서대로 배치하는 방법으로 로그기법[12]과 동일하다. 그림 6에서처럼 로그 영역을 따로 설정할 필요가 없다. 이 방법의 장점은 SSD 관리 블록의 크기가 크더라도 데이터 복제 오버헤드를 최소화시키기 때문에 쓰기 성능을 향상시킬 수 있다는 것이다. 그러나 이에 따라 맵핑 테이블이 커져서 해당 주소를 검색하는 시간이 오래 걸리거나 읽기 동작을 위한 병렬처리가 최적화 되어 있지 않기 때문에 읽기 성능이 저하될 것이다.

(b) 쓰기 명령 주소처럼 해당 디스크 I/O 블록이 위치해야 될 곳이 정해진 경우에는 데이터 복제 오버헤드로 인해서 쓰기 성능은 저하되겠지만 읽기 성능 SSD



(a) 쓰기 명령 순서 (b) 쓰기 명령 주소

그림 15 디스크 I/O 블록들의 배치 방법

관리 블록 디자인에 따라 안정적으로 될 것이다. 그림 6처럼 로그 영역 설정이 성능 향상을 위해서 반드시 필요하다.

3.1에서 언급한 작은 임의 크기를 임시로 쓸 수 있는 로그 버퍼영역을 고려한다면 두 가지 방식의 성능 차이가 없을 것이라고 생각 할 수 도 있지만, (a)의 경우에는 쓰기 명령의 순서대로 플래시 페이지를 배치하고 유지하지만, (b)의 경우에는 로그 버퍼영역에 임시로 쓰기를 하고서 정해진 위치로 데이터를 위치시키기 때문에 로그 버퍼의 크기에 따라 쓰기 성능 저하는 외부로 늦게 보일 수 있다. 읽기 성능을 볼 때, (a)의 방법을 사용한 경우에는 임의/순차 읽기의 성능은 크게 저하될 수 있지만, (b)의 방법을 사용한 경우에는 임의/순차 읽기 성능은 일정한 수준으로 지속적으로 유지될 수 있을 것이다. 이는 그림 8, 9를 통해서 확인할 수 있다. SSD C의 3개월 후의 순차/임의 읽기 성능은 저하되고 순차 쓰기와 임의 쓰기의 성능이 거의 같다는 점을 볼 때 (a)와 같은 방법을 사용할 가능성이 크다는 것을 알 수 있고, 상대적으로 SSD A는 순차 쓰기와 임의 쓰기의 성능차이가 크기 때문에 (b)와 같은 방법으로 사용한다는 것을 알 수 있다. 그림 16은 가장 큰 SSD 관리 블록을 지니고 있는 SSD B에 EasyCo에서 개발한 MFT(Managed Flash Technology)를 적용하여 성능을 측정한 것이다[11]. MFT는 작은 크기의 임의 쓰기를 순차 쓰기 형태로 변환해

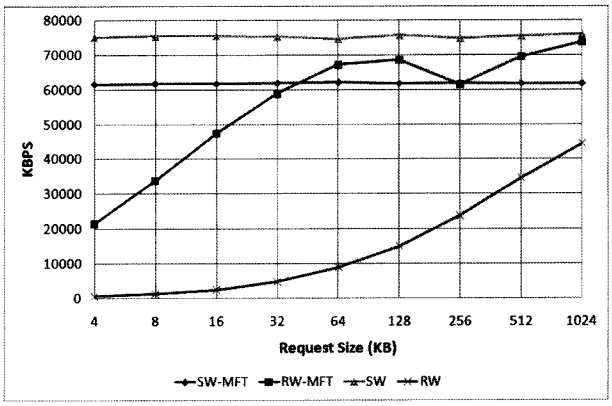


그림 16 SSD B에 MFT를 적용했을 때 순차/임의 쓰기 성능: SW는 순차 쓰기, RW는 임의 쓰기

서 SSD로 하여금 순차 쓰기 형태만을 처리하도록 도와준다. 따라서 (b)와 같은 방법을 유지하고 있는 SSD B로 하여금 (a)와 같은 방법이 적용되어 임의 쓰기 성능을 향상시키도록 도와준다. 문제는 이러한 방법이 그림 9와 같이 읽기 성능을 저하시킨다는 것이다.

3.3 SSD 관리 블록을 구성하는 플래시 페이지들 배치

다중 플래시 메모리 채널을 지니고 있는 SSD에서는 I/O 병렬처리를 효과적으로 달성하기 위해서 SSD 관리 블록을 구성하는 플래시 페이지들이 플래시 메모리 채널 간에 고르게 분포하는 것이 중요하다. 그림 17은 SSD 관리 블록을 구성하는 플래시 페이지들을 할당하는 가능한 방법을 보여준다.

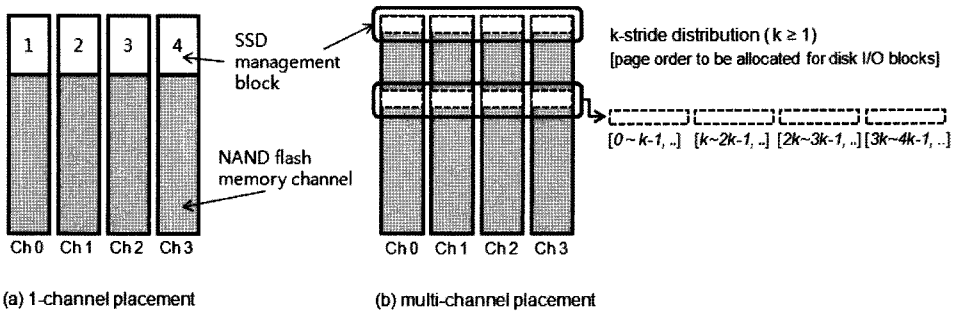


그림 17 SSD 관리 블록을 구성하는 플래시 페이지 할당 방법

(a) 1-channel placement 방법은 하나의 SSD 관리 블록을 하나의 플래시 메모리 채널에 있는 플래시 페이지들을 이용하여 할당하는 방법이다. 이 방법이 I/O 병렬처리에 부정적으로 보일 것 같지만 SSD 관리 블록의 크기를 적당하게 하고 3.2의 쓰기 명령 순서 기법을 이용한다면 추후 읽기 명령을 병렬 처리하는데 도움이 될 수 있을 것이다.

(b) mutil-channel placement 방법은 하나의 SSD 관리 블록을 다중 플래시 메모리 채널에서 고르게 플래시 페이지들로 구성하는 방법이다. 이 방법과 3.2의 쓰기 명령 순서 기법을 이용한다면 추후 읽기 명령을 병렬 처리하는데 최악이 될 수도 있지만 쓰기 명령 주소 기법으로 사용하게 되면 읽기 성능을 최적화 시킬 수 있을 것이다.

4. 결 론

SSD는 저장 매체인 NAND 플래시 메모리의 고유한 I/O 특성으로 인해서, 디스크 I/O 블록을 내부 플래시 페이지로의 맵핑 및 Garbage-collection이나 wear-leveling 등 NAND 플래시 메모리 관리를 위한 내장 소프트웨어의 설계가 매우 중요하다. SSD의 중요한 내장 소프트웨어 설계요소는 다음과 같다.

디스크 I/O 블록과 내부 플래시 페이지 간 맵핑 테이블 및 관련 알고리즘을 결정하는데 중요한 역할을 하는 것은 SSD 관리 블록이다. SSD 관리 블록의 크기, SSD 관리 블록 내의 데이터 배치, SSD 관리 블록을 위한 플래시 페이지 구성 방법은 성능이나 I/O 특성에 영향을 끼친다.

SSD 관리 블록의 크기가 클수록 주소 맵핑 테이블이 작아지기 때문에 관리 오버헤드가 적어질 수 있다. 그러나 SSD 관리 블록의 크기보다 작은 쓰기의 경우 많은 오버헤드를 발생시킬 수 있다. 이에 따라 로그 영역관리가 중요해 진다.

SSD 관리 블록에 데이터 배치는 쓰기 및 읽기 성능에 매우 중요한 영향을 미칠 수 있다. 로그 기법으로 데이터를 배치할 경우 쓰기 성능은 매우 향상되지만, 읽기 성능은 장담할 수 없고, 플래시 메모리 관리 오버헤드도 증가할 수밖에 없다. 이와 달리 정해진 위치에 데이터 배치는 읽기 성능을 보장할 수 있지만 쓰기 성능 저하를 피할 수 없다. 따라서 데이터를 임시로 버퍼링할 수 있는 로그 영역이 필요하다.

I/O 병렬처리 최적화를 위해 디스크 I/O 블록의 SSD 관리 블록내의 플래시 페이지로의 배치와 동시에 SSD 관리 블록을 구성하는 플래시 페이지 할당하는 방법은 중요하다.

향후 SSD 내부의 DRAM 쓰기 버퍼가 SSD 성능에 미치는 영향에 대해 분석이 필요하며, 이를 이용하여 호스트 I/O 요구 형태를 SSD 내부 DRAM 쓰기 버퍼를 이용하여 하위 SSD 관리 블록의 효용성을 높이는 형태의 연구가 필요하다.

참고문헌

[1] J. Cooke, Flash memory technology direction. In Proceedings of Microsoft WinHec 2007.

[2] J. Cooke, Introduction to flash memory. In Proceedings of Flash Memory Summit 2008.

[3] N. Agrawal, V. Prabhakaran, T. Wobber, J. D. Davis, M. Manasse, and R. Panigrahy, Design tradeoffs for ssd performance. In Proceedings of 2008 USENIX Annual Technical Conference.

[4] IOZONE Iozone filesystem benchmark tool. http://www.iozone.org.

[5] Samsung corporation, SLC NAND flash memory (k9xxg08uxa) specification. http://www.samsung.com/global/business/semiconductor/products/flash/ProductsNANDFlash.html.

[6] Samsung corporation, ssd products. http://www.samsungssd.com

[7] Mtron corporation, ssd products. http://www.mtron.net.

[8] Intel corporation, ssd products. http://www.intel.com/design/flash/nand/ extreme/index.htm.

[9] J. Kim, J. M. Kim, S. H. Noh, S. L. Min, and Y. Cho. "A space-efficient flash translation layer for compact flash systems", IEEE Transactions on Consumer Electronics, vol. 48, no. 2, pp. 366-375, 2002.

[10] S. W. Lee, D. J. Park, T. S. Chung, W. K. Choi, D. H. Lee, S. W. Park, and H. J. Song. "A log buffer based flash translation layer using fully associative sector translation", ACM Transactions on Embedded Computing Systems, vol. 6, no. 3, 2007.

[11] Managed Flash Technology, http://www.easyco.com

[12] Mendel Rosenblum and John K. Ousterhout: The Design and Implementation of a Log-Structured File System, ACM Transactionson Computer Systems, vol. 10, no. 1, pp. 26-52, 1992

[13] Y. H. Bae, "Design of A High Performance Flash Memory-based Solid State Disk," Journal of Korean Institute of Information Scientists and Engineers, Vol. 25, Issue 6, 2007.

류 준 길

2002 포항공과대학교 학사
2002~현재 포항공과대학교 컴퓨터공학과 석박사 통합과정 재학중
관심분야 네트워크 스토리지 QoS, 고신뢰도 스토리지 시스템
E-mail : lancer@postech.ac.kr

박 찬 익

1983 서울대학교 전자공학과 학사
1985 한국과학기술원 전자공학 (컴퓨터공학 전공) 석사
1988 한국과학기술원 전자공학 (컴퓨터공학 전공) 박사
1989~현재 포항공과대학교 컴퓨터공학과 교수
관심분야 : 지능형 스토리지 시스템, 내장형 실시간 운영체제, 시스템 보안
E-mail : cipark@postech.ac.kr