Filter Driver 와 NAND FLASH Memory를 이용한 HDD 장치의 성능 개선에 관한 연구

김재경* · 김우길** · 김영길***

A Study of HDD Performance Improvement through Filter Driver & NAND FLASH Memory

Jae-kyung Kim* · Woo-Gil Kim** · Young-Kil Kim***

본 연구결과는 국토해양부의 "U 기반 해운 물류 체계 구축을 위한 기반기술 연구" 과제에서 수행된 연구결과 중 일부임을 밝히며, 연구비 지원에 감사드립니다.

요 약

본 논문에서는 NAND FLASH Memory를 HDD 의 Cache 로 사용하기 위해 I/O 용 Filter Driver의 구현을 제시 했으며, 이를 통해 HDD 저장장치의 느린 I/O 성능을 개선하기 위한 방법에 대해 연구했다. 반도체 부품으로서 빠른 I/O 성능을 보이는 NAND FLASH Memory이지만, 비싼 가격 때문에 HDD를 통째로 대체할 수 없음에서 본 개선 방법을 제안했다.

본 연구는 SSD의 빠른 I/O성능과 Filter Driver 의 Cache 관리 성능을 통해서 적은 비용으로 HDD의 높은 Performance를 이루어 냄을 목적으로 한다.

ABSTRACT

In this paper, we research the method for HDD I/O Performance improvement by Filter Driver & NAND FLASH Memory. This paper was started from NAND Flash Memory can not be replaced by HDD because of high cost.

So We consider that using NAND Flash Memory as cache for HDD. It can be achieved high HDD Performance through Filter Driver by low cost.

키워드

Nand flash memory, Cache, Filter driver, HDD performance

Key Word

Nand flash memory, Cache, Filter driver, HDD performance

* 정회원 : 아주대학교 의용공학과 박사 과정
** 준회원 : 아주대학교 전자공학과 석사과정
*** 종신회원 : 아주대학교 전자공학과 교수 (교신저자, ykkim@ajou.ac.kr)

접수일자 : 2011. 03. 16
심사완료일자 : 2011. 04. 11

Ⅰ. 서 론

I/O 디바이스들의 처리 속도가 중앙 처리 장치의 처리 속도를 따라 가지 못해 생기는 시스템의 성능 저하 현상을 줄이기 위한 노력은 계속되고 있다.

여전히 컴퓨터 시스템의 중앙 처리 장치의 동작 주파수에 비해 주변 디바이스들의 동작 속도 및 데이터 버스의 속도는 여전히 따라 주질 못하고 있으며 이는 매년 증가하는 중앙 처리 장치의 성능을 사용자가 실제로 그 차이를 느낄 수 없도록 하는 주요인으로 작용하고 있다.

특히 데이터의 주 저장장치로 쓰이는 HDD의 사용 빈도는 타 디바이스에 비해 월등히 높다. 시스템을 사용하는 사용자가 느낄 수 있는 속도의 정도는 HDD의 성능에 영향을 가장 많이 받고 있다[1].

이에 본 논문에서는 기존의 물리적 동작이 많은 HDD 저장장치의 I/O 성능상의 한계점을 뛰어넘기 위해 반도체소자 중의 하나인 NAND FLASH Memory 와 이를 HDD의 Cache로 동작시킬 수 있는 Filter Driver를 구성 하였다. Host 의 HDD로의 Read / Write 동작 시 Filter Driver 의 동작으로 HDD와 NAND FLASH Memory에 효율적으로 기입 가능함을 확인 하고자 한다. 아울러 이후의 Host 동작 시에 HDD를 통해 시스템에서 데이터를 찾고자 접근할 때, Filter Driver 의 효과적인 판단을 바탕으로 NAND FLASH Memory에 우선적으로 접근하게 하여 HDD에 접속하는 빈도를 낮추고, 보다 빠르게 데이터를 읽어 오도록 하여 궁극적으로 PC 의 Booting 시간과 Application 구동 시간을 단축을 확인 하고자 한다.

Ⅱ. 본 론

2.1. NAND FLASH Memory 의 특징

최근 휴대폰, MP3, 디지털카메라, PMP와 같은 휴대용 단말기를 위한 저장 공간으로 플래시 메모리가 각광받고 있다.

NAND 플래시 메모리는 전기적으로 데이터 개서가 가능하면서 전원이 나가도 데이터를 보존하는 불휘발성 반도체 메모리로서 메모리 셀 구조가 DRAM 보다 간단하여 쉽게 고집적화를 이룰 수 있다. 메모리 셀 당 1개의 트랜지스터와 1개의 Capacitor로 구성되는 DRAM에 비해 플래시 메모리는 1개의 트랜지스터만 존재하므로 bit cost가 DRAM 의 약 0.6~0.8 배로 낮다.

따라서 대량 생산을 할 경우 가장 값싼 반도체 메모리로서 반도체 특유의 고속 동작, 높은 신뢰성 등의 장점을 무기로 보인다[2].

이러한 장점에서 착안하여 HDD 의 Cache 로 NAND FLASH Memory를 사용한 SSD (Solid State Device)를 사용하기로 결정했다.

2.2. SSD vs HDD의 성능 비교

아래 그림과 같이 Sequential Read 성능 비교 시 SSD가 HDD보다 3배정도 빠른 성능을 보이고 있다.

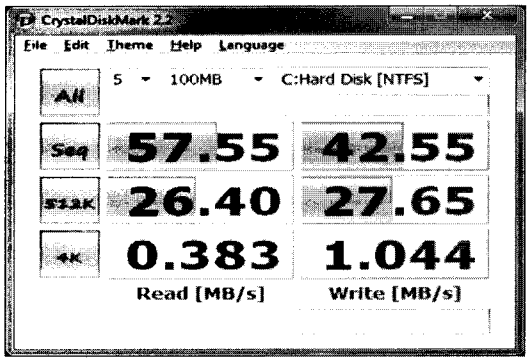


그림 1.HDD Data 전달률
(by CrystalDiskMark)
Fig 1. HDD Data Transfer Rate
(by CrystalDiskMark)

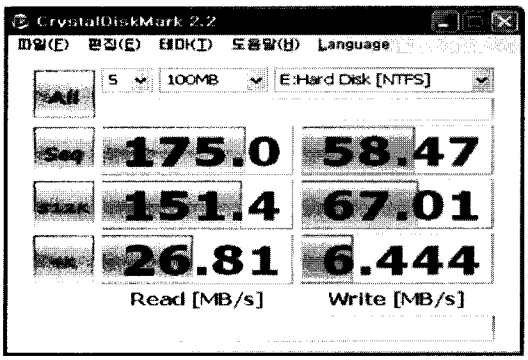


그림 2. SSD Data 전달률
(by CrystalDiskMark)
Fig 2. SSD Data Transfer Rate
(by CrystalDiskMark)

2.3. Filter Driver 의 특징

Filter Driver 는 시스템의 동작에 특정 값을 추가, 수정할 수 있는 Device Driver이다. 다른 Device Driver와 마찬가지로 OS 내에서 동작되는 kernel-mode component 이다.

하나 이상의 파일 시스템이나 파일 시스템볼륨에 대해서 입출력 동작을 Filtering 할 수 있기 때문에 Application의 성격에 따라서 로그, 계측, modify, prevent 등의 동작을 수행할 수 있다.

일반적으로 백신 프로그램, 암호화 프로그램, Hierarchical Storage Management System 등의Application 에 사용된다.

본 논문에서는 특히 Data Channel 확보, File System Control, Data Modify, Redirection 등의 역할을 할 수 있도록 구성하였다.

Ⅲ. Filter Driver 기본 알고리즘

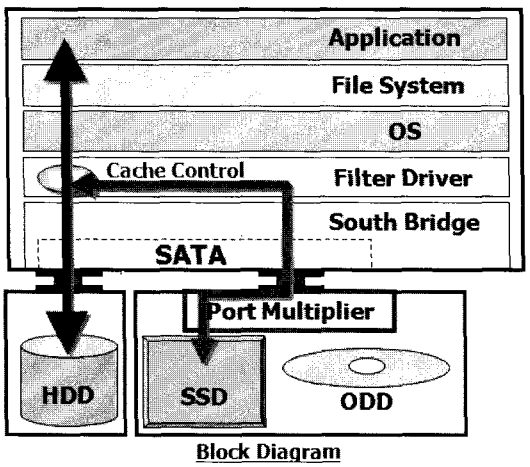


그림 3. H/W구성과 Filter Driver 할당
Fig 3. H/W and Filter Driver Allocation

그림 3.은 본 논문의 실험 진행을 위한 H/W 와 Filter Driver 의 동작 위치를 같이 나타낸 Block diagram이다.

Filter Driver 는 OS 와 SATA I/O Controller 사이에 위치하게 되며, Host 의 I/O 동작 과정에 SSD를 HDD의 Cache로 동작을 할 수 있도록 Control 한다.

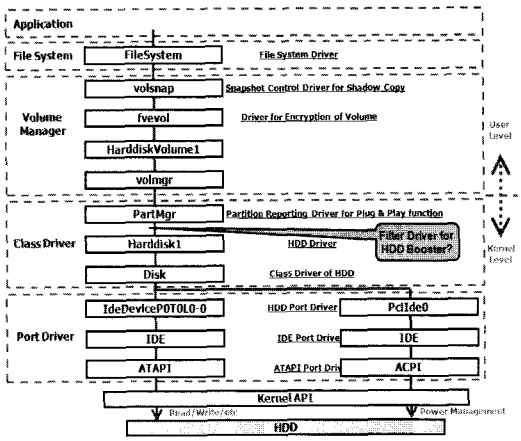


그림 4. Windows Device Driver 구성
Fig 4. Windows Device Driver Overview

그림 4.는 Windows OS상의 General 한 Device Driver 의 Hierarchy를 보여준다.

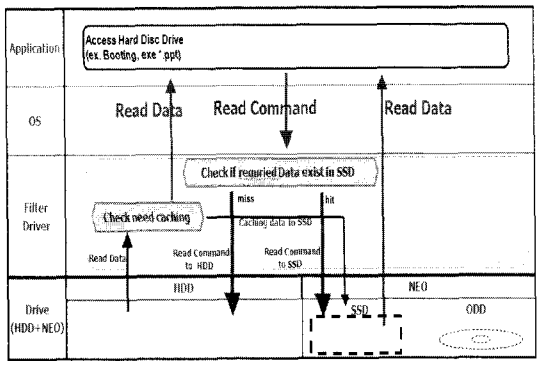


그림 5. Filter Driver 기본 알고리즘
Fig 5. Filter Driver Basic Algorithm

Filter Driver Operation 의 기본 Algorithm 은 Host (Os or Application)로 부터의 HDD 의 Data Read 시 SSD cache hit 상태에는 HDD Access를 하지 않고 SSD cached data를 Host 에 Return 하고, cache mis-hit 시는 HDD를 Access 해서 data를 Host 측에 return 함과 동시에 interactive한 동작으로 return data를 SSD 에 caching 하도록 동작한다.

이 과정의 중요한 Filter Driver 의 역할은 redirection, Filtering, 로서 필요에 따라서는 data modify 의 동작까지 처리한다.

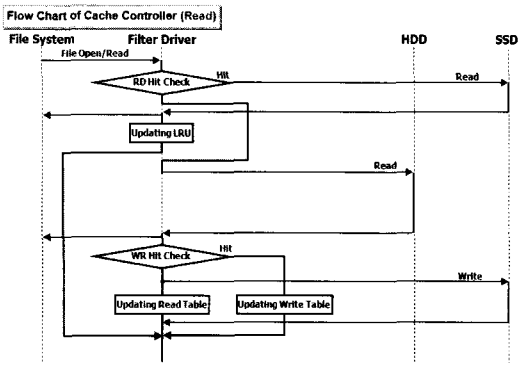


그림 6. Filter Driver Read Caching 알고리즘
Fig 6. Filter Driver Read Caching Algorithm

그림 6. 은 그림 5.의 Filter Driver 동작을 좀 더 구체적으로 도식화해서 표현한 그림이다.

Ⅳ. Filter Driver의 성능 향상을 위한 Cache 관리 Table 알고리즘 제안

기본적으로 Filter Driver라는 Device Driver를 이용하여, HDD에서 읽은 Data를 SSD에 저장해 놓고 (also update at Cache Control Table, 그림 7 참조), 다음 HDD Read시에 Filter Driver가 이미 SSD에 저장되어 있다고 판단되면(by Cache Control Table 의 cache hit 여부)해당 Data를 HDD 대신 속도가 상대적으로 빠른 SSD에서 읽도록 하여 전체적인 성능향상을 꾀하는 기능이다.

Reading Cache Control Table				
No	HDD Address (Original)		SSD Address (Caching)	
	Start LBA	End LBA	Start LBA	End LBA
1	100	10F	10000	1000F
2	200	20F	10010	1001F
3	300	30F	10020	1002F
4	400	40F	10030	1003F

그림 7. Cache 조정 Table
Fig 7. Cache Control Table

4.1. AVL Binary Tree를 이용한 검색시간 단축

Filter Driver가 얼마나 빨리 해당 Data를 SSD에서 찾아내는 것이 Filter Driver성능에 가장 큰 영향을 미치는 것을 여러 실험을 통해서 확인했다.

기존 방식은 HDD에서 읽은 순서대로 Linear하게 Record(구조는 다음 장 그림 8 참조)를 List형태로 저장하는 방식으로써 ,저장하는 Record개수가 많아질수록 검색시간이 길어져, 성능이 떨어지게 되는 현상이 나타난다.

이를 보완하고자, Record의 HDD의 LBA값을 Key로 하여 Binary Tree를 구성하여 Record검색시간을 비약적으로 줄인다.(다음 그림 9 참조)

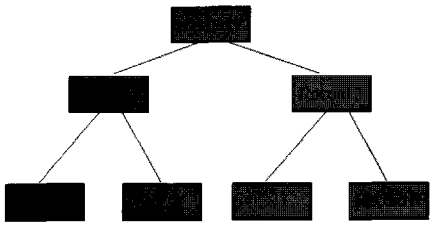
전체적인 SSD에서 부터의 read 성능 향상을 이룰 수 있었다.

<기존 Record 관리 형태 >

- SSD에 저장되는 시간 순으로 List 형태로 정렬
- List 검색 시 처음부터 끝까지 순차적으로 검색
  ➔ 찾으려는 Data가 List의 마지막에 위치한다면 검색시간이 많이 걸리게 된다.

그림 8. 기존의 Cache Table 관리 형태
Fig 8. Current Cache Table Management

< AVL Binary Tree 적용 형태 >



< 그림3 Binary Tree 구조 >

- Binary Tree 형태로 Record 관리
- Ex) 1023개의 Record가 있을 경우
  기존방식 : 최악의 경우 1023회 비교
  Binary Tree : 최악의 경우 10회 비교

그림 9. AVL Binary Tree 적용 형태
Fig 9. AVL Binary Tree Adaption

4.2. Sequential Read 성능 개선 Table 관리 방식

기존 방식은 Host 의 Read command 접수 시 Cache Control Table 의 맨 앞쪽부터 hit 여부를 판단하기 시작하여, 기존에 여러 형태의 사용으로 Control Cache Table 의 많이 caching 된 상태에서는 Searching 에 걸리는 시간이 Liner 하게 증가해서 HDD Booster 성능도 이에 준하게 떨어지는 현상이 나타났다. (다음 그림 10 참조)

이를 개선하고자, Last Read Command 의 LBA를 내부적으로 저장하고 Next Read command 접수 시 제일 먼저 참조해서 Searching 의 시간을 비약적으로 줄인다.

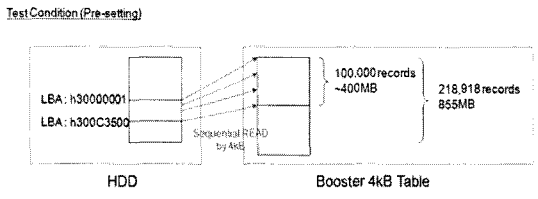


그림 10. 도입부의 Control Table searching
Fig 10. Control Table searching at first position

위 실험을 통해서, Cache Control Table 의 Caching 상태가 1st Read 시도 (LBA h30000001 ~h300C3500) 후 왼쪽 booster 4KB Table 의 파란색과 같이 이미 채워져 있는 상태에서 동일한 Address (LBA h30000001 ~h300C3500)를 다시 내려서 HDD Booster 의 성능을 확인하는 과정에서 Cache Control Table Searching 에만 필요한 시간을 표시한 그림이다.

결과적으로, Sequential 한 LBA 의 Read 시 Cache Control Table Searching Time 은 Liner 한 증가를 나타냈다.

이 알고리즘의 적용을 통해서 Sequential Read 의 성능을 향상 시킬 수 있었다.

V. 성능 분석 결과

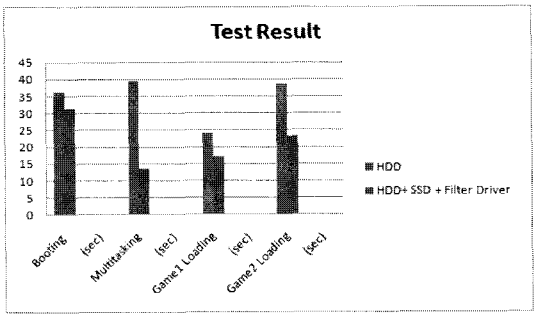
그림 11.은 구성된 SSD (H/W)를 기반으로 위 제안된 알고리즘을 적용한 Filter Driver를 동작한HDD의 성능 향상을 측정한 결과 이다.

PC Booting 시간, Multitasking loading 시간, Game loading 시간등의 3가지 항목에 대해서 측정을 하였다.

1) PC Booting 시간 : 동일 PC 조건에서 Win7 OS 의 Booting 시간 측정 방식
→ HDD only 동작 대비 13% 향상

2) Multitasking loading 시간 : 동일 PC 조건에서 Multi-Task : Photoshop, Notepad, explorer, MsPaint, Calc, WordPad, PowerPoint, MsWord, excel, MPlayer, FreeCell 등의 일반적인 상용툴의 loading 시간
→ HDD only 동작 대비 65% 향상

3) Game loading 시간 : Turok 이라는 상용 게임 loading 시간
→ HDD only 동작 대비 36% 향상



Item	HDD	HDD + SSD + Filter Driver
Booting (sec)	36.25	**31.5**
Multitasking (sec)	39.47	**13.61**
Game Loading (sec)	62.73	**40.34**

그림 11. Test1 Performance 측정 결과
Fig 11. Test1 Performance Result

그림12. 는 PC Industry standard bench mark tool 인 PCmark Score를 측정한 결과이다.

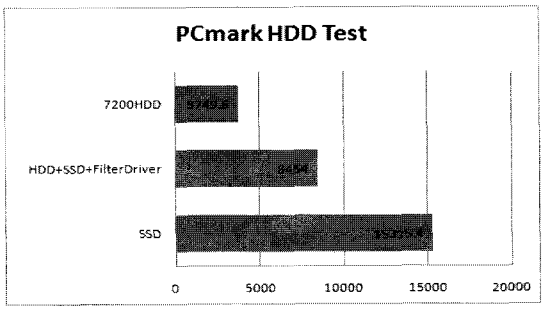


그림 12. Test2 Performance 측정 결과
Fig 12. Test1 Performance Result

7200 rpm 의 HDD 대비 4700 점 이상의 높은 결과를 나타내고 있다.

VI. 결 론 및 향후 계획

본 논문에서는 SSD를 HDD의 성능 향상을 위한 Cache로 사용할 목적으로 Filter Driver 와의 결합을 시도한 방법을 제안하였다. 제안한 방법은 HDD 와 SSD 간의 I/O 처리를 HDD 에 접근하는 방법을 최소화하는 형태로 동작하도록 Filter Driver를 구현함으로 HDD 의 성능 향상의 결과를 유도하였다.

특히 Cache Control Table의 Inserting, Deleting, Searching등의 성능 향상이 성능과 직접적으로 관여된다는 사실을 파악했으며, 이 부분의 향상을 위한 연구를 향후 중요한 topic 으로 삼을 예정이다.

보다 나은 성능을 위한 Demerit 원인 도출과 해결을 위한 추가 연구도 진행할 계획이다.

감사의 글

본 연구결과는 국토해양부의 "U 기반 해운 물류 체계 구축을 위한 기반기술 연구" 과제에서 수행된 연구결과 중 일부임을 밝히며, 연구비 지원에 감사드립니다.

참고문헌

[1] 최병윤, "하드 디스크 드라이브 성능 향상을 위한 NAND메모리 적용 시스템 설계 및 성능평가" 아주대학교 대학원, 2007.

[2] 서강덕, "고용량 플래시 메모리의 응용 및 전망"삼성전자 반도체 메모리 本部, 1997

[3] Eyee Hyun Nam, Jin Hyuk Yoon, Yoon Jae Seong, Hong Seok Kim, Jin-yong Choi, and Soo Kwan Lee School of Computer Science and Engineering, Seoul National University, Seoul, Korea "Hydra: A High Performance Flash Memory Solid State Disk Architecture", 2008

[4] ERAN GAL and SIVAN TOLEDO School of Computer Science, Tel-Aviv University "Algorithms and Data Structures for Flash Memories", 2004

[5] CHANIK PARK, WONMOON CHEON, JEONGUK KANG,KANGHO ROH, and WONHEE CHO Samsung Electronics and JIN-SOO KIM Korea Advanced Institute of Science and Technology "A Reconfigurable FTL (Flash Translation Layer) Architecture for NAND Flash-Based Applications" , 2008

[6] SILBERSCHATZ, GALVIN, GAGNE "Operating System Concepts"

저자소개

김재경(Jae-kyung Kim)

1988. 2 아주 대학교 전자공학과 학사
1990. 2 아주대학교 전자공학과 석사
1989.12 ~ 2011.2 메디슨 연구소
2010. 8 ~ 아주대학교 의용공학과 박사 과정
2011. 4 ~현재 : 지멘스 (SLS) 초음파 상무이사
※관심분야 : 의료기기(초음파진단기), 신호처리

김우길(Woo-Gil Kim)

2000 명지대학교 전기전자공학과학사
2000-2001 LG전자 설계실 연구원
2001-현재 Hitachi-LG Data Storage 책임연구원
아주대학교 전자공학과 석사과정
※관심분야 : 데이터 통신, 컴퓨터 공학, Embedded System

김영길(Young-kil Kim)

1978 고려대학교 전자공학과 학사
1980 한국과학기술원 석사
1984 ENST(프랑스) 박사
1984-현재 아주대 전자공학과 교수

※관심분야: 의용 생체 공학, Embedded System