

윈도우즈 기반 플래시 디스크의 성능 분석을 위한 도구 개발¹⁾

박성환⁰* 유현석* 김도윤* 박원주* 오기신** 이상원** 박상원*

*한국의국어대학교 컴퓨터및정보통신공학부 **성균관대학교 정보통신공학부

{shpark⁰, hsyoo, dykim, wjpark}@islab.hufs.ac.kr, {kisunoh, swlee}@skku.edu, swpark@hufs.ac.kr

Development of Performance Analysis Tool for Flash Disk on Windows Platform

Sung-Hwan Park⁰* Hyun-Seok Yoo* Do Yun Kim* Won-Joo Park* Kisun Oh** Sang-Won Lee** Sangwon Park*

*Computer Science & Information Communication Engineering Div, Hankuk University of Foreign Studies

**Information and Communication Engineering, Sungkyunkwan University

요 약

이동 기기의 저장 장치로 플래시 메모리가 널리 사용되고 있으며 고정 하드 디스크를 대체하는 저장 매체로 부상하고 있다. 그러므로 플래시 메모리의 특성을 잘 이해하고, 최대한 효율적으로 플래시 메모리를 사용할 필요성이 있다. 이러한 역할을 해주는 것은 플래시 내부의 시스템 소프트웨어인 FTL(Flash Translation Layer)이다. FTL은 운영체제가 디스크에 전달하는 블록을 물리적인 플래시 메모리에 맵핑하는 역할을 한다. 그러므로 플래시 메모리의 성능은 FTL 알고리즘이 결정한다. 플래시를 대체한 플래시 디스크에서는 기존의 파일 시스템이 탑재되며 간단한 섹터 기반의 이동형 기기에서와는 다른 특성을 가진다. FTL 성능을 평가하기 위해서는 실제적으로 플래시 메모리가 장착된 제품들이 동작하는 시스템에서 실험을 해야 한다. 많은 플래시 디스크는 윈도우즈에서 동작하므로 윈도우즈의 디스크 I/O를 추출하여 실험을 해야만 한다. 본 논문에서는 윈도우즈에서 물리적인 디스크 I/O 패턴을 추출하여 FTL 알고리즘의 성능을 평가하기 위한 도구 개발에 대하여 설명하고 이에 대한 간단한 결과를 보인다.

1. 서론

최근 디지털 카메라, MP3 플레이어, 핸드폰과 같이 이동성이 중요한 요소로 차지하는 기기들이 많이 등장하였다. 이에 따라 소형화, 대용량화, 저 전력화, 비휘발성, 고속화 그리고 충격에 강한 메모리가 필요하게 되었다. 플래시 메모리는 이러한 요구 사항을 만족시키는 메모리로 디스크에 비해 많은 장점을 가지고 있다. 때문에 점차 디스크를 대체하여 대표적인 대용량 저장장치로도 자리매김을 할 것이다.

플래시 메모리는 임의의 읽기와 쓰기를 제공하는 디스크와 다르게 쓰기 전 소거(erase-before-write)연산을 한다. 이것은 플래시 메모리의 하드웨어적 특성으로 인해 섹터에 쓰기 연산을 하기 전에 그 섹터가 속한 블록을 소거 연산을 통해 초기화한 다음 쓰기 연산을 수행해야 한다는 것이다. 이러한 소거 연산을 한 블록에 10만 번 이상 수행하면 그 블록을 사용할 수 없게 된다. 때문에 전체 블록이 평균하게 소거 연산이 발생되도록 하는 FTL이 필요하다. 플래시 디스크를 위한 FTL 알고리즘의 성능을 평가하기 위해서는 윈도우즈의 디스크 I/O 패턴으로 실험을 해야 한다. 또한 앞으로 디스크를 대체할 보조기억장치의 성능을 평가하는데 있어서 윈도우즈의 디스크 I/O 패턴은 하나의 벤치마크 테스트 자료로 사용될 수 있다. 때문에 드라이버를 작성하여 디스크 I/O 패턴을 얻는 것은 매우 중요하다.

본 연구에서는 윈도우즈에 필터드라이버를 설치하여 실제 디스크로 전송되는 I/O 패턴을 읽어내었다. 또한 실제 플래시 메모리처럼 동작하는 가상 플래시 메모리와 각 FTL 알고리즘을 구현하였고, 실험을 통하여 성능을 평가하는 프로그램을 작성하였다. 가상 플래시 메모리에서 얻어지는 정보들을 통해 어떤 FTL 알고리즘의 성능이 가장 좋은 알고리즘인지 평가할 수 있었다.

본 연구는 다른 연구들의 밑바탕이 될 것이다. 다른 연구들은 윈도우즈에 대한 추가적인 지식이 없어도 각종 디스크 I/O 패턴을 얻어 실험 할 수 있을 것이다. 또한, 연구 기간이 단축 될

것이고 연구 비용이 감소 할 것이다.

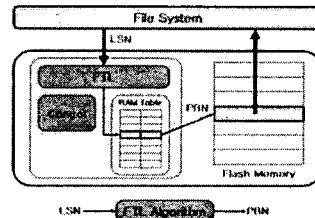


그림 1. 플래시 메모리 구조

2. 관련연구

2.1 FTL 알고리즘

그림 1은 플래시 메모리 시스템에 대한 구조도이다. FTL 알고리즘은 플래시 메모리와 파일 시스템 사이에 존재하며 파일 시스템이 플래시 메모리를 논리적 수준에서 바라볼 수 있게 해준다. FTL 알고리즘은 파일 시스템으로부터 전달받은 섹터 번호와 길이를 가지고 효율적인 맵핑을 통해 전체적으로 읽기, 쓰기, 소거연산을 줄여준다. FTL 알고리즘은 블록 소거연산을 평균화 시킨다. 또한 전체적으로 고른 소거연산을 발생시켜 플래시 메모리의 단점을 극복해 준다.

이러한 기능을 발휘하기 위해 플래시 메모리는 램을 사용한다. 램은 맵핑 정보를 가지고 있으며 이를 통하여 운영체제에서 요구하는 논리적인 섹터를 플래시 메모리의 물리적인 섹터로 맵핑할 수 있게 된다.

2.2 윈도우즈의 I/O

그림 2는 윈도우즈에서 디스크로 가는 I/O 요청의 흐름도이다. 사용자 모드의 프로그램이 읽기나 쓰기의 시스템 호출을 통해서 커널 I/O 관리자에게 I/O 요청을 한다. I/O관리자는 I/O 요청을 받으면 IRP(I/O Request Packet)를 만들어 파일 시스템에게 IRP를 전달한다. I/O 관리자는 해당 I/O 요청의 최종 목적지까지 고려하여 IRP를 작성하고, IRP의 스택에 각 계층들이 수행해야 할 정보들을 저장한다. 이렇게 전달된 IRP는 파일 시스템에서 논리 수준의 섹터 번호, 길이, 자료 지시자를 결정

1) 본 논문은 2005년도 한국의국어대학교 학술연구비 지원에 의해서 연구되었음

해 IRP를 수정한다. 디스크 드라이버는 IRP를 전달받은 뒤에 실제로 어떤 장치에게 전달해야 하는지 보고 해당 물리 장치에게 전송하게 된다.

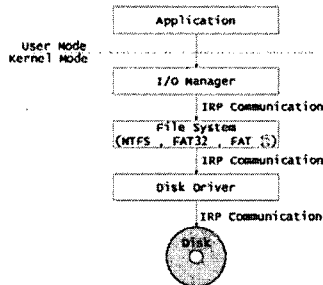


그림 2. 윈도우즈의 I/O 흐름

2.3 윈도우즈 드라이버 작성

윈도우즈 드라이버는 DDK(Device Development Kit)을 이용하여 작성할 수 있다. 사용할 수 있는 자료형, 구조체, 상수, 매크로 등은 모두 NTDDK.h라는 헤더 파일에 기술되어 있다. 드라이버에는 IRP를 받았을 때 수행해야 할 루틴들을 작성해 준다. 즉, IRP에서 읽기나 쓰기 I/O 요청할 때 이를 처리할 수 있는 루틴을 구현해 놓아야 한다.

드라이버는 계층적으로 나누어 작성할 수 있다. 한 드라이버가 해야 할 일들이 많거나, 개념적으로 다른 일을 수행해야 하면, 계층 드라이버로 나누어 작성할 수 있다. 필터 드라이버는 계층 드라이버의 한 예로서 마치 스펀지처럼 상위 계층 드라이버에서 내려온 IRP를 참조만 한 뒤 하위 계층 드라이버에게 IRP를 전달한다.

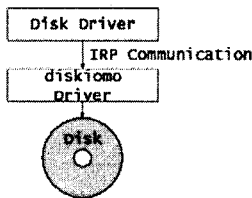


그림 3. diskio 드라이버의 부착 모습

3. 실험

3.1 디스크 I/O를 얻기 위한 드라이버 작성 및 부착

디스크로 가는 I/O를 분석하기 위해 I/O를 얻어오는 필터 드라이버를 작성하였다. 이 드라이버의 이름은 diskio(Disk I/O Monitoring Driver)로서 디스크 드라이버의 하위 계층 드라이버로 작성하였고, 실제 디스크로 가는 IRP를 스펀지하여 I/O 패턴을 얻어 낼 수 있다. 그림 3은 디스크 드라이버와 실제 물리 디스크 사이에 diskio 드라이버를 부착시킨 구조도이다. diskio 드라이버에서 얻은 I/O는 파일시스템과 디스크 드라이버를 거쳐 실제 물리 디스크에게 전송되는 정보이며 실제 플래시 메모리로 전송되는 I/O 요청과 동일하다.

3.2 FTL APAT(FTL Algorithm Performance Analysis Tool)

FTL APAT는 로그정보를 읽어 FTL 알고리즘의 성능을 평가하는 도구이다. 그림 4는 FTL APAT가 동작하는 원리를 그린 개념도이다. FTL APAT는 크게 3가지의 객체로 나누어져 있다. 2) "diskio.log" 파일에서 로그정보를 읽어오는 LogFileReader 객체와 각각의 FTL 알고리즘을 구현해 놓은 FTL_Algorithm 객체

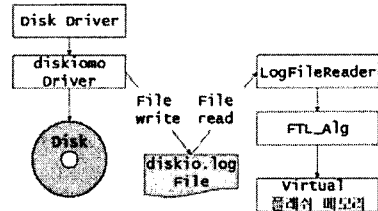


그림 4. FTL APAT 동작 원리

체가 있다. 그리고 실제 플래시 메모리처럼 동작하는 가상 플래시 메모리가 있다. FTL APAT는 각 객체로부터 정보를 수집해 결과를 도출한다.

diskio 드라이버는 I/O가 올 때마다 "diskio.log"라는 파일에 기록을 추가한다. LogFileReader 객체는 파일을 감시하면서 추가된 기록이 있는지를 확인한다. 만약 추가된 기록이 있다면 로그정보를 읽어 FTL_Algorithm 객체에게 전달해준다. 때문에 FTL APAT는 실시간으로 I/O를 얻어올 수 있다. FTL_Algorithm 객체는 LogFileReader가 전해주는 로그정보를 실행하고자 하는 FTL 알고리즘에 적용하여 연산을 한다. 가상 플래시 메모리는 FTL_Algorithm 객체에게 읽기, 쓰기, 소거연산을 제공하여 FTL_Algorithm 객체가 마치 실제 플래시 메모리에 연산하듯이 사용할 수 있게 해준다.

diskio 드라이버는 로그정보를 물리적으로 다른 디스크에 "diskio.log" 파일로 저장한다. FTL APAT의 LogFileReader 객체는 "diskio.log"파일에서 로그정보를 읽어와 FTL_Algorithm 객체에 적용시킨다. FTL_Algorithm 객체는 가상 플래시 메모리를 이용해 알고리즘을 수행한다. FTL APAT은 FTL_Algorithm 객체와 가상 플래시 메모리에서 얻어진 각종 통계 정보를 실시간으로 표시하며 결과를 특정 파일로 저장을 한다.

3.3 실험 구조

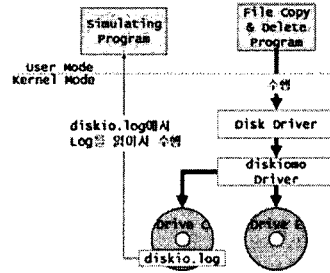


그림 5. 실험 구조

파일 복사와 소거를 수행하는 프로그램이 diskio 드라이버가 부착된 드라이브 E에 강제로 I/O를 발생시킨다. 이때 diskio 드라이버는 이에 따른 로그정보를 만들어 드라이브 C에 "diskio.log"파일을 만들어 기록을 하게 되고, FTL APAT는 "diskio.log"파일에서 로그정보를 실시간으로 읽는다. FTL APAT는 로그정보를 FTL 알고리즘에 적용해 각종 정보를 얻어내고 통계를 계산해 FTL 알고리즘의 성능을 평가한다.

3.4 실험 환경

표 1은 diskio 드라이버와 FTL APAT가 동작한 환경이다. 운영체제는 Windows 2000 Professional을 사용하였고 파일시스템은 FAT와 NTFS 모두에서 실험을 했다. 디스크는 100M 바이트 크기로 실험을 했고, 신뢰성 있는 결과를 얻기 위해 약 130M바이트 정도의 자료를 이용해 3.5G바이트정도의 I/O를 발생시켰다. I/O를 발생시키는 패턴에는 순차적인 I/O와 임의적인 I/O가 있고, 이는 성능평가의 중요한 요소가 된다.

2) 자세한 내용은 <http://islab.hufs.ac.kr/flash>를 참고.

표 1. 실험 환경

운영체제	Windows 2000 Professional
파일시스템	FAT , NTFS
사용 디스크	드라이브 C : "diskio.log" 파일 저장 드라이브 E : diskio모 드라이버를 부착
드라이브 E 크기	100M바이트
발생 I/O의 크기	130M바이트 정도의 자료를 이용해 약 3.5G바이트 정도의 I/O를 발생시킴

표 2는 드라이브 E에 강제로 발생시키는 I/O의 패턴별로 정리가 되어 있다. JPG는 200K바이트~400K바이트 정도의 작은 파일이며 작은 길이의 순차 로그정보가 발생한다. MP3는 3M 바이트~4M바이트 정도의 비교적 큰 파일이며 긴 길이의 순차 로그정보가 발생한다. 혼합은 다중 프로세스가 접근하여 다량의 분할 I/O가 발생한다. DB는 TPC-H 벤치마크 테스트를 하면서 I/O를 얻었다.

표 2. I/O 발생 패턴

JPG	200K바이트~400K바이트의 작은 크기의 파일
MP3	3M~5M의 큰 크기의 파일
혼합	JPG와 MP3를 혼합하여 다중 프로세스를 통해 다량의 분할 발생
Database	TPC-H 벤치마크 테스트

휴대용 기기에서는 단일 프로세스가 순차적인 I/O를 발생시킬 것으로 예상되며, 차후 플래시 메모리가 디스크를 대체하게 되면 혼합과 DB와 같은 임의적인 I/O가 발생 될 것으로 예상된다.

4. 실험결과

4.1 diskio모 드라이버 부착 모습

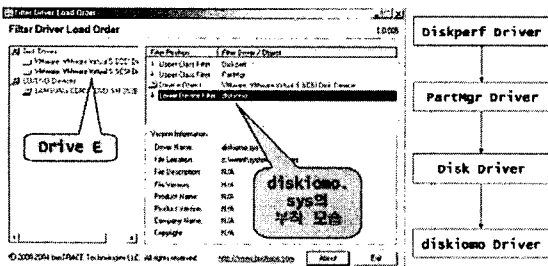


그림 6. Filter Driver Load Order 실행 모습

Filter Driver Load Order 프로그램은 앞에서 설명한 필터 드라이버의 부착 모습을 계층적으로 보여주는 프로그램이다. 그림 6의 왼쪽에는 실제 디스크를 나타낸다. 오른쪽에는 diskio모 드라이버가 계층적으로 부착되어 있는 모습이다.

그림 8은 FTL APAT가 실제로 디스크 I/O를 얻어 실험하고 있는 모습이다. FTL APAT는 실시간으로 로그정보를 읽어 통계치 및 결과를 보여준다. 표 3은 FTL APAT가 실시간으로 보여주는 정보표이다. 이중에 플래시 메모리 연산 정보와 물리 블록 소거 정보는 해당 FTL 알고리즘의 성능을 평가하는 중요한 정보이다. 플래시 메모리 에러 확률과 사용 블록 수는 알고리즘을 디버깅하기 위한 정보이다.

본 논문에서의 결과물로 diskio모 드라이버와 사용법에 관한

내용과 실제 디스크 I/O 패턴을 뽑은 데이터 및 실험 결과는 <http://islab.hufs.ac.kr/flash>에서 얻을 수 있다.

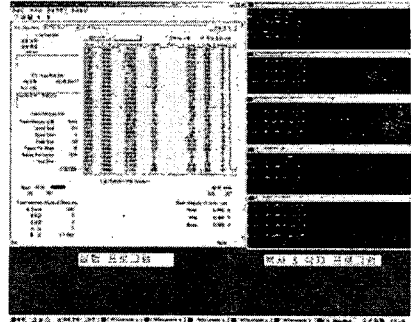


그림 7. 실험 프로그램 동작 모습

표 3. FTL APAT가 제공하는 정보

Log 파일 정보	실험 날짜, 실험 환경, 메모
FTL 알고리즘 정보	알고리즘 이름, 특이사항, RAM 사용 크기
플래시 메모리 정보	플래시 메모리 이름, 섹터 크기, 스페어 크기, 페이지 크기, 블록 당 섹터 수, 총 블록 개수, 총 메모리 크기
파일 시스템 Log 정보	읽기 쓰기 비율
플래시 메모리 연산 정보	한 I/O당 실제 플래시 메모리 읽기, 쓰기, 소거 연산 호출 횟수
물리 블록 소거 정보	총 소거 연산 횟수, 최대 소거 횟수, 최소 소거 횟수, 차이 값, 평균, 표준편차
플래시 메모리 에러 확률	읽기, 쓰기, 소거 오류 확률
사용 블록 수	전체 블록 중 사용 중인 블록 수

5. 결론

최근 다양한 휴대용 장치가 등장함에 따라 플래시 메모리가 주목받고 있다. 그동안 보조기억장치로 자리매김을 해왔던 디스크보다 장점이 많기 때문이다. 하지만, 플래시 메모리를 효율적으로 사용하기 위해서는 FTL 알고리즘이 필수적이다. 플래시 디스크에서 여러 FTL 알고리즘을 평가하기 위해서는 디스크 I/O 패턴을 얻어야 한다. 본 논문에서는 디스크 드라이버의 필터 드라이버를 작성하여 디스크 I/O 패턴을 얻었으며, 이를 실시간으로 FTL 알고리즘에 적용해 보며 성능을 평가할 수 있는 FTL APAT를 개발하였다.

실험을 위해 몇 가지의 디스크 I/O 패턴만을 얻었지만, 다양한 플래시 메모리 제품의 성능을 평가하기 위해서는 다양한 디스크 I/O 패턴이 필요하다. 이는 앞으로 벤치마크 테스트 자료가 될 것이며, diskio모 드라이버는 벤치마크 테스트 자료를 만들 때 필수적인 드라이버가 될 것이다. 그리고 FTL APAT는 FTL의 성능을 평가하는 필수적인 프로그램이 될 것이다.

참고문헌

[1] Art Baker, Jerry Lozano, "Windows 2000 Device Driver Book : A Guide for Programmers 2nd Edition", Prentice Hall, 2000
 [2] O'Reilly, "Windows NT File System Internals, Developer's Guide", Rajeev Nagar