

# 플래시 메모리 기반의 파일 저장 장치에 대한 성능분석

성민영<sup>1\*</sup>

## Performance Evaluation of Flash Memory-Based File Storages: NAND vs. NOR

Min-Young Sung<sup>1\*</sup>

**요약** 본 논문은 플래시 메모리를 이용한 파일 저장 장치의 성능 분석을 다룬다. 특히 플래시 메모리의 대표적인 형태인 낸드(NAND) 플래시와 노어(NOR) 플래시에 대해 비교 분석한다. 성능 평가를 위해 마이크로소프트 PocketPC 기반의 실험 플랫폼에 두 플래시 타입을 위한 파일 저장 시스템을 각각 구성하였다. 이렇게 구성된 플랫폼을 이용하여, 버퍼 크기, 사용 용량, 커널 수준 쓰기 캐싱 등의 변수에 따른 입출력 처리량을 측정/비교하였다. 실험 결과에 따르면, 낸드 플래시 기반 저장 장치의 성능이 쓰기/읽기 처리량 관점에서 각각 4.8배, 5.7배까지 더 높은 것으로 나타났다. 본 실험 결과는 두 가지 플래시 메모리 저장 방식의 상대적인 장단점을 잘 보여주고 있으며 플래시 메모리 기반의 파일 저장장치의 설계에 유용하게 활용될 수 있다.

**Abstract** This paper covers the performance evaluation of two flash memory-based file storages, NAND and NOR, which are the major flash types. To evaluate their performances, we set up separate file storages for the two types of flash memories on a PocketPC-based experimental platform. Using the platform, we measured and compared the I/O throughputs in terms of buffer size, amount of used space, and kernel-level write caching. According to the results from our experiments, the overall performance of the NAND-based storage is higher than that of NOR by up to 4.8 and 5.7 times in write and read throughputs, respectively. The experimental results show the relative strengths and weaknesses of the two schemes and provide insights which we believe assist in the design of flash memory-based file storages.

**Key Words** : Flash memory, I/O performance, FTL (Flash Translation Layer)

### 1. 서론

플래시 메모리는 비휘발성, 낮은 전력 소모, 외부 충격 인내성 등의 매력적인 특성으로 인하여 각종 모바일 컴퓨팅 기기에 필수적인 구성 요소로 인식되고 있다 [5,6,9,11,15]. 특히, 낸드(NAND)와 노어(NOR) 타입 플래시는 두 가지 대표적인 플래시 타입으로서 프로그래밍 코드 및 데이터 저장 용도로 널리 사용되고 있다. 이렇게 플래시 메모리가 중요해짐에 따라, 두 가지 플래시 타입의 성능에 대한 완전하고 정량적인 분석이 점차 중요성을 갖기 시작했으며 이제 제품 설계에 매우 필요한 상황이 되었다. 하지만, 플래시 메모리에 대한 제품 규격 자료만으로는 그 성능을 예측하기가 매우 어렵다. 그 이유는 두 타입의 플래시 메모리는 데이터 입출력 인터페이스,

데이터 읽기/쓰기/지우기 속도 등 물리적 특성에서 큰 차이를 갖고 있으며, 이는 궁극적으로 연관된 소프트웨어의 구조에도 큰 차이를 갖도록 하였기 때문이다.

본 논문은 입출력 처리량을 성능 척도로 하여 플래시 메모리 기반 파일 저장 장치의 성능을 사용자 혹은 응용 관점에서 비교 분석한다. 이를 위해, 낸드 및 노어 플래시를 위한 두 가지 별도의 파일 저장 장치를 구성하고 다양한 값의 사용자 수준 버퍼 크기, 사용된 공간의 비율에 따른 입출력 처리량을 측정하였다. 또한 커널 수준 쓰기 캐싱이 파일 쓰기 성능에 미치는 영향을 관찰하였다. 낸드 플래시로는 삼성 낸드 플래시 메모리 및 PocketStore를 사용하는 경우를 고려하였으며 노어 플래시로는 인텔 노어 플래시 메모리 및 PSM을 사용하도록 구성하였다. 이러한 실험 결과를 이용하여 두 가지 대표적인 플래시 타입의 상대적인 장단점을 분석하였다.

본 논문은 다음과 같이 구성된다. 제 2 절은 플래시 메모리 및 플래시 기반 파일 저장 장치의 전반적 개요를 기

<sup>1</sup>상명대학교 컴퓨터소프트웨어공학과  
\*교신저자: 성민영(mysung@smu.ac.kr)

술한다. 제 3 절에서는 두 가지 타입의 플래시 기반 저장 시스템에서의 실험 결과를 제시한다. 마지막으로 제 4 절은 실험 결과를 요약하고 향후 연구에 대해 기술한다.

## 2. 플래시 메모리 기반 파일저장 시스템

### 2.1 낸드 및 노어 플래시 메모리

플래시 메모리는 몇 가지 특별한 처리를 요하는 특성을 가지고 있다[5]. 첫째, 플래시 메모리 미디어는 'erase-before-rewrite' 구조를 갖고 있다. 지움(erase) 상태에서는 플래시 종류에 따라 메모리 셀은 1 혹은 0의 값을 갖게 되며 각 데이터 비트는 새 값으로 프로그래밍될 수 있다. 하지만 일단 값이 쓰인 후에는 다시 프로그래밍하기 위해서는 지움 상태로 리셋 되어야 한다. 둘째, 지움 연산은 쓰기 연산보다 훨씬 큰 단위로만 수행될 수 있으며, 이 작업은 상당한 시간을 필요로 한다. 셋째, 각 메모리 셀에 허용되는 지움 연산의 횟수에 제한이 있다. 일반적으로 이러한 문제를 해결하기 위해, FTL (Flash Translation Layer) 이라는 중간 계층 소프트웨어를 호스트 응용 프로그램과 플래시 메모리 사이에 두게 된다 [3,4,5]. FTL의 주요 기능은 특수하게 고안된 매핑 알고리즘을 이용하여 호스트 응용으로부터 전달된 논리적 주소를 플래시 메모리내의 물리적 섹터 주소로 변환하는 것이다.

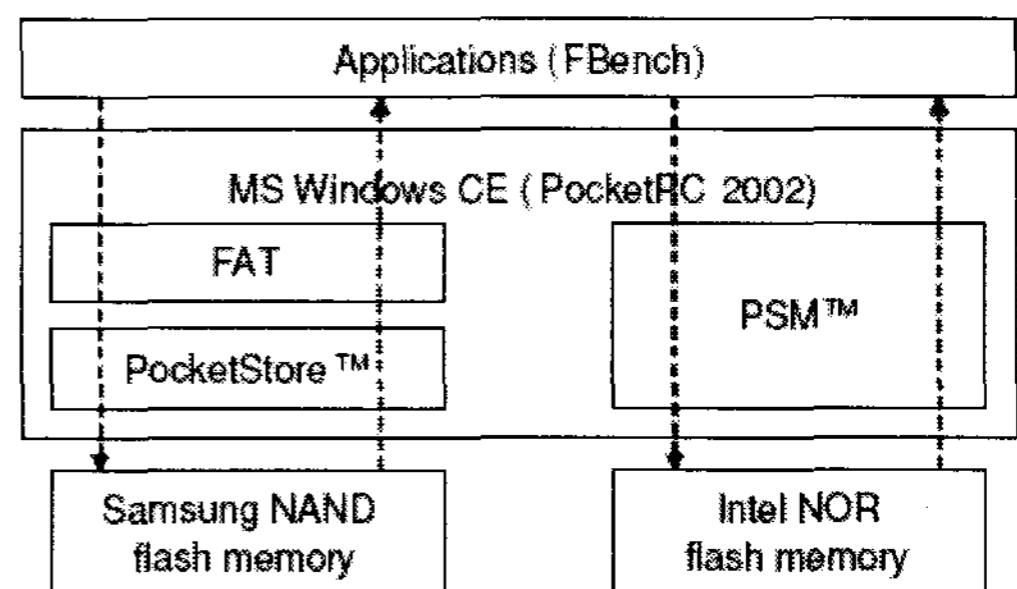
비록 낸드와 노어 플래시가 앞에서 기술된 플래시 미디어의 특성을 공유하고 있지만, 입출력 인터페이스, 구조, 동작 속도 등에서 상당한 차이를 보인다. 표 1은 이러한 특성을 요약한 것이다. 낸드 플래시는 블록 인터페이스를 제공하며 상대적으로 빠른 읽기/쓰기 속도를 갖는다[12,13]. 입출력 연산의 단위는 섹터 혹은 페이지라고 불리는데 일반적으로 528 바이트 (512 데이터 바이트와 에러 복구를 위한 추가적인 16바이트)이다. 따라서 낸드 플래시는 이동형 저장 용도를 위해 하드 디스크의 대체 용도로서 선호되어 왔다. 이에 비해, 노어 플래시는 선형적으로 주소 지정이 가능하다. 즉 각 바이트가 독립적으로 읽힐 수 있으며 쓰일 수 있다. 하지만 노어 플래시는 프로그래밍 연산 및 지움 연산 측면에서 매우 낮은 속도를 갖는다. 이러한 특성은 노어 플래시로 하여금 데이터 저장 용도보다는 프로그램 실행 장소로 더 적합하게 만들었다.

[표 1] 낸드/노어 플래시 메모리의 주요 특성

	NAND 플래시 (K9K1208U)	NOR 플래시 (28F128J3A)
I/O interface	블록 인터페이스. 528 바이트 단위 (512 바이트의 데이터 + 16 추가 데이터 영역)	Linearly addressable. SRAM 비슷한 인터페이스 XIP (Execution In Place)
Bad block	bad block 관리를 필요로 함	bad block 없이 판매됨
Erase lifetime	100,000 ~ 1,000,000 회	10,000 ~ 100,000 회
Block size	16 KB	128 KB
Program time	페이지당 200 $\mu$ s	32byte당 218 $\mu$ s
Read time	페이지당 10 $\mu$ s	32byte당 0.12 $\mu$ s
Erase time	블록당 2 ms	블록당 1000 ms

그러나, 기술이 발달함에 따라 이 두 가지 타입의 플래시는 거의 모든 응용 분야에서 서로 경쟁하고 있다. 일반적으로 플래시 메모리에 있는 프로그램을 실행하기 위해서는 낸드 플래시가 노어 플래시보다 더 많은 SDRAM을 필요로 한다. 하지만, 최근에는 "demand paging" 메커니즘을 이용하여 이러한 필요 SDRAM 용량을 줄이고 있다 [10,11]. 한편, 노어 플래시는 입출력 성능, 특히 프로그래밍 및 쓰기 속도에서 낮은 성능을 가지고 있다. 하지만, 노어 플래시 진영에서는 대부분의 모바일 기기에서 플래시 메모리보다는 네트워크가 주된 데이터 이동 병목으로 작용하므로 노어 플래시도 데이터 저장 용도로 효과적으로 사용될 수 있다고 주장하고 있는 상황이다[8].

### 2.2 삼성 PocketStore와 Intel PSM



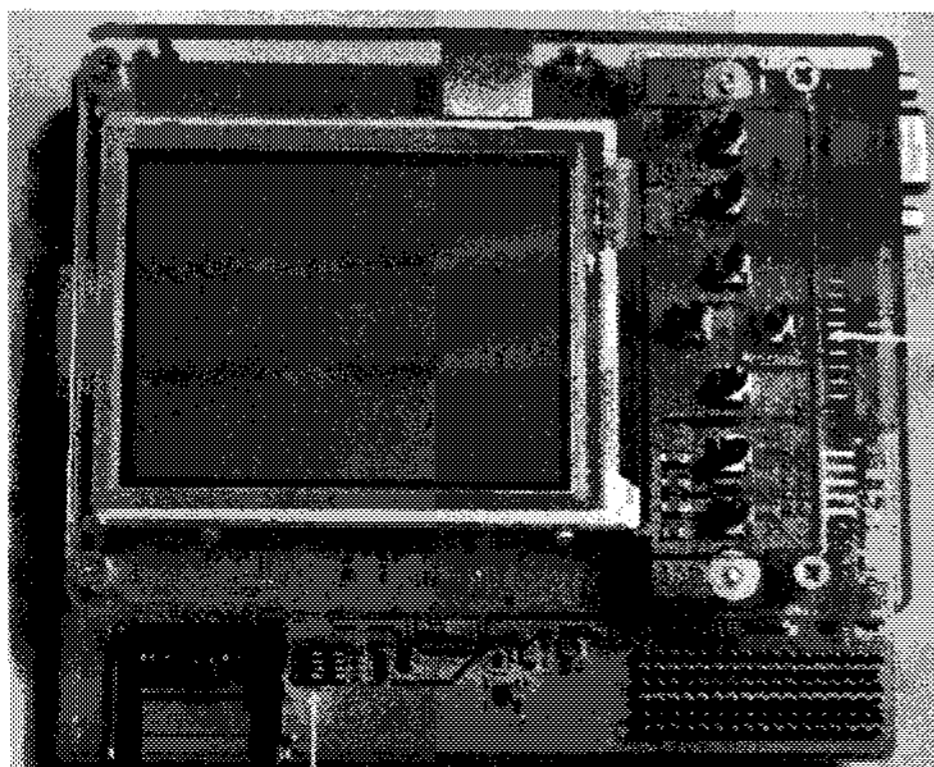
[그림 1] 플래시 메모리 기반의 저장 시스템

그림 1은 플래시 메모리 저장 장치를 탑재한 마이크로소프트 PocketPC의 전형적인 구조를 보인 것이다. 낸드 기반의 저장 장치로는 PocketStore (삼성전자 등록상표)를 이용하였다. PocketStore는 PocketPC 기반의 장치들에서 낸드 플래시 메모리를 사용하기 위한 소프트웨어 솔루션이다. PocketStore는 FAT 파일 시스템과 낸드 플래시 장치 사이에 존재한다.

PocketStore는 SSR (Simple Sector Remapper)라 불리는 FTL 방식을 포함한다[4,7]. SSR은 클러스터 모드 (clustered mode)와 스캐터 모드(scattered mode)의 두 가지 동작 모드를 가지고 있다. 섹터를 읽기 쓰기 동작의 단위라고 가정하고, 블록을 낸드 플래시 메모리에서의 지우기 동작의 단위라고 하면, 클러스터 모드와 스캐터 모드의 차이는 논리적 섹터 번호로부터 논리적 블록 번호를 계산하는 해시 함수에 있다고 할 수 있다. 클러스터 모드에서는 인접한 논리적 섹터 번호들은 보통 동일한 논리 블록 번호를 갖는 블록에 위치한다. 따라서, 인접한 논리적 섹터들은 일반적으로 같은 물리적 블록에 존재하게 된다. 반면, 스캐터 모드에서는 인접한 논리적 섹터 번호들은 여러 논리적 블록에 분산된다. 따라서 인접한 논리적 섹터들은 일반적으로 서로 다른 물리적 블록 위치에 저장된다.

노어 플래시 메모리 기반의 저장 장치를 갖는 기기에서는 일반적으로 PSM (Persistent Storage Manager)이 사용된다[1]. 낸드 메모리를 위한 구성과는 달리 PSM은 플래시 메모리를 위해 특별히 고안된 고유의 파일 시스템을 내장하고 있다. 이는 중복된 기능으로 인한 비효율을 제거하고 캐시 기능을 포함하여 성능을 높이기 위한 것으로 판단된다.

### 3. 성능 평가 결과



[그림 2] 실험용 하드웨어 플랫폼

### 3.1 실험 환경

실험을 위해 PDA 사양의 개발용 평가 플랫폼인 "Reindeer Plus" 개발 보드위에 마이크로소프트 PocketPC 기반의 실험 플랫폼을 구축하였다. 그림 2는 실험에 사용된 하드웨어를 보인 것이다. 테스트베드 하드웨어는 203 MHz로 동작하는 삼성 S3C2410 프로세서[14], LCD 디스플레이, 직렬 및 이더넷 인터페이스, 그리고 SDRAM 및 플래시 메모리를 위한 소켓 등을 포함한다. 특히, 소켓은 낸드와 노어 실험 모두에 동일한 하드웨어를 사용하는 것을 가능하게 했다. 낸드 플래시 실험에는 풀 섀도우 (full shadow) 정책을 사용하였다. 즉, 운영체제와 모든 응용 프로그램 이미지들은 부트 과정 중에 SDRAM에 적재되었다. 이 방식은 프로그램 코드가 플래시 메모리에서 직접 읽혀져서 실행되는 노어 플래시 메모리의 XIP(execution in place) 정책과 대비된다. 표 2는 실험 플랫폼의 사양을 요약한 것이다. 공평한 비교를 위해, 부트 후에 두 시스템이 각각 비슷한 용량의 사용 가능 RAM과 플래시 메모리를 갖도록 조치하였다.

[표 2] 실험 플랫폼 사양

	NAND 플래시 (K9K1208U[13])	NOR 플래시 (28F128J3A[2])
CPU	삼성 S3C2410 203 MHz	삼성 S3C2410 203 MHz
OS	PocketPC 2002 (with full shadow)	PocketPC 2002 (with XIP)
OS size	32 MB	16 MB
File system	FAT (with Samsung PocketStore)	Intel PSM
RAM (부트후)	64 MB (27 MB)	32 MB (30 MB)
Flash (부트후)	64 MB (14 MB)	32 MB (14 MB)

성능 척도로는 오버헤드를 고려한 I/O 처리량 (throughput)을 사용하였다. 이 척도는 파일 시스템과 함수 호출 오버헤드를 포함한 것으로서 사용자 혹은 응용 프로그램 입장에서 관찰되는 입출력 성능을 정확하게 나타낸다. 그림 3은 플래시 메모리 기반 파일 시스템에서의 입출력 처리량을 측정하도록 고안된 벤치마크 프로그램인 FBench의 전반적인 구성을 보인 것이다.

FILE\_SIZE와 NUM\_TRIALS는 각각 2MB, 13으로 설정되었다. 이 값은 플래시 지우기 연산이 벤치마크 실행마다 적어도 한번씩 일어나도록 해주는 것으로 확인되었다.

```

for (i=0; i<NUM_TRIALS; i++)
{
    remain_size = FILE_SIZE;
    start_time ← current time;

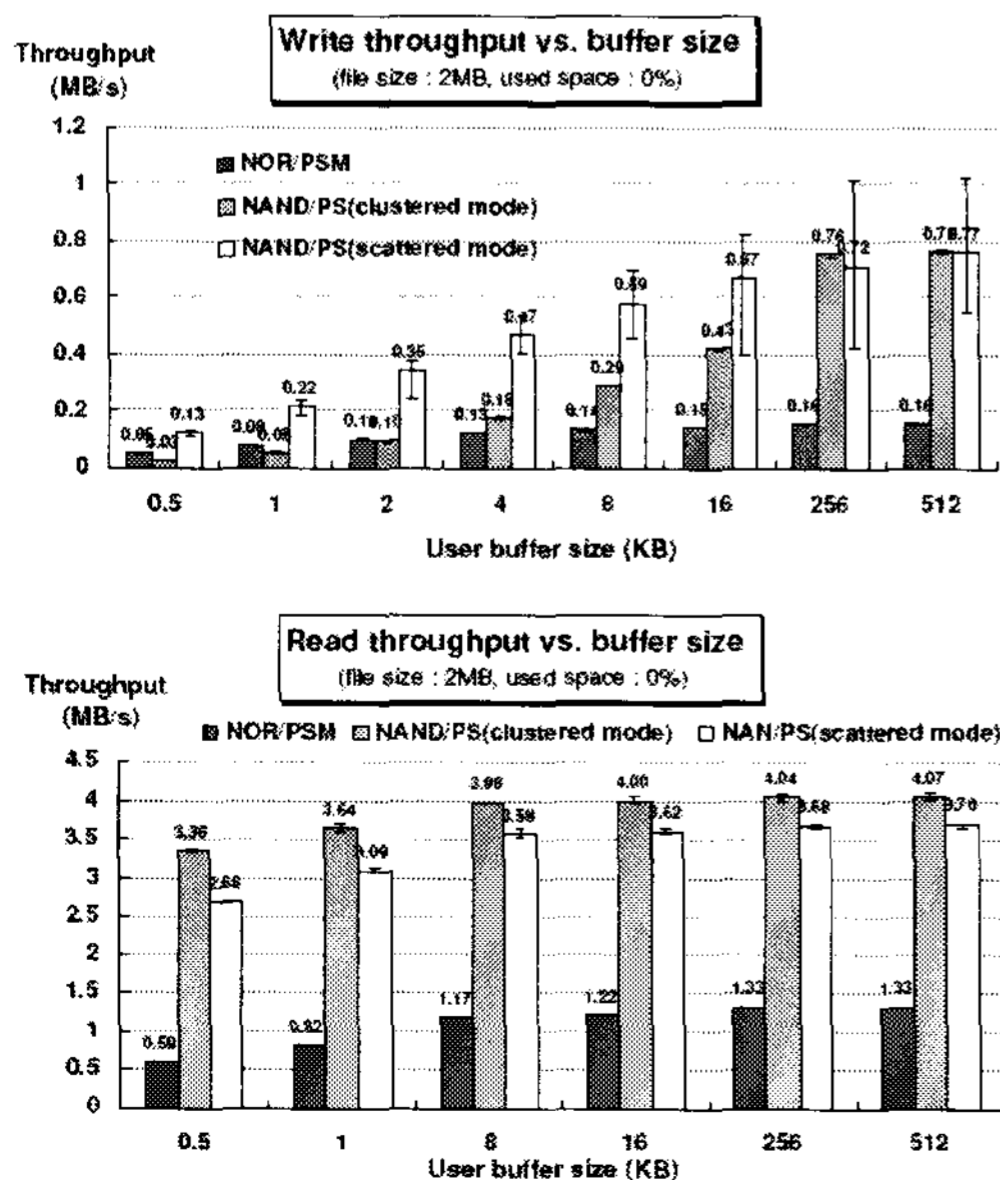
    open file for create;
    while ( remain_size > 0 )
    {
        write or read data by BUF_SIZE;
    }
    close file;

    end_time ← current time;

    throughput = FILE_SIZE /
                (end_time - start_time);
}
    
```

[그림 3] 입출력 처리량 벤치마크를 위한 의사코드

시간 측정에는 WinCE에서 제공하는 GetTickCount() 함수를 사용하였다. 이 함수는 시스템이 부트된 후부터 지나간 시간을 millisecond 단위로 리턴하는데, 측정 결과, 실험 시스템에서는 약 10ms 내외의 정확도 (resolution)를 갖는 것으로 판단되었다. 플래시 메모리 기반 저장 공간에 2MB 정도의 작은 파일을 입출력하는데도 약 1.5초 이상의 시간이 걸리기 때문에 GetTickCount()로도 큰 오차 없이 측정할 수 있다고 판단했다.



[그림 4] 사용자 수준 버퍼 크기에 따른 쓰기/읽기 처리량

### 3.2 사용자 수준 입출력 버퍼 크기

사용자 수준 입출력 버퍼란 파일 읽기 쓰기를 위해 응용 프로그램에 의해 제공되는 메모리 버퍼를 지칭한다. 이는 그림 3의 BUF\_SIZE에 해당한다. 그림 4는 다양한 버퍼 크기에 따른 평균, 최소, 최대 쓰기/읽기 처리량을 보인 것이다. 낸드 플래시의 평균 처리량은 0.03 ~ 0.77 MB/s이었으며 노어 플래시의 평균 처리량은 0.05 ~ 0.16 MB/s이었다.

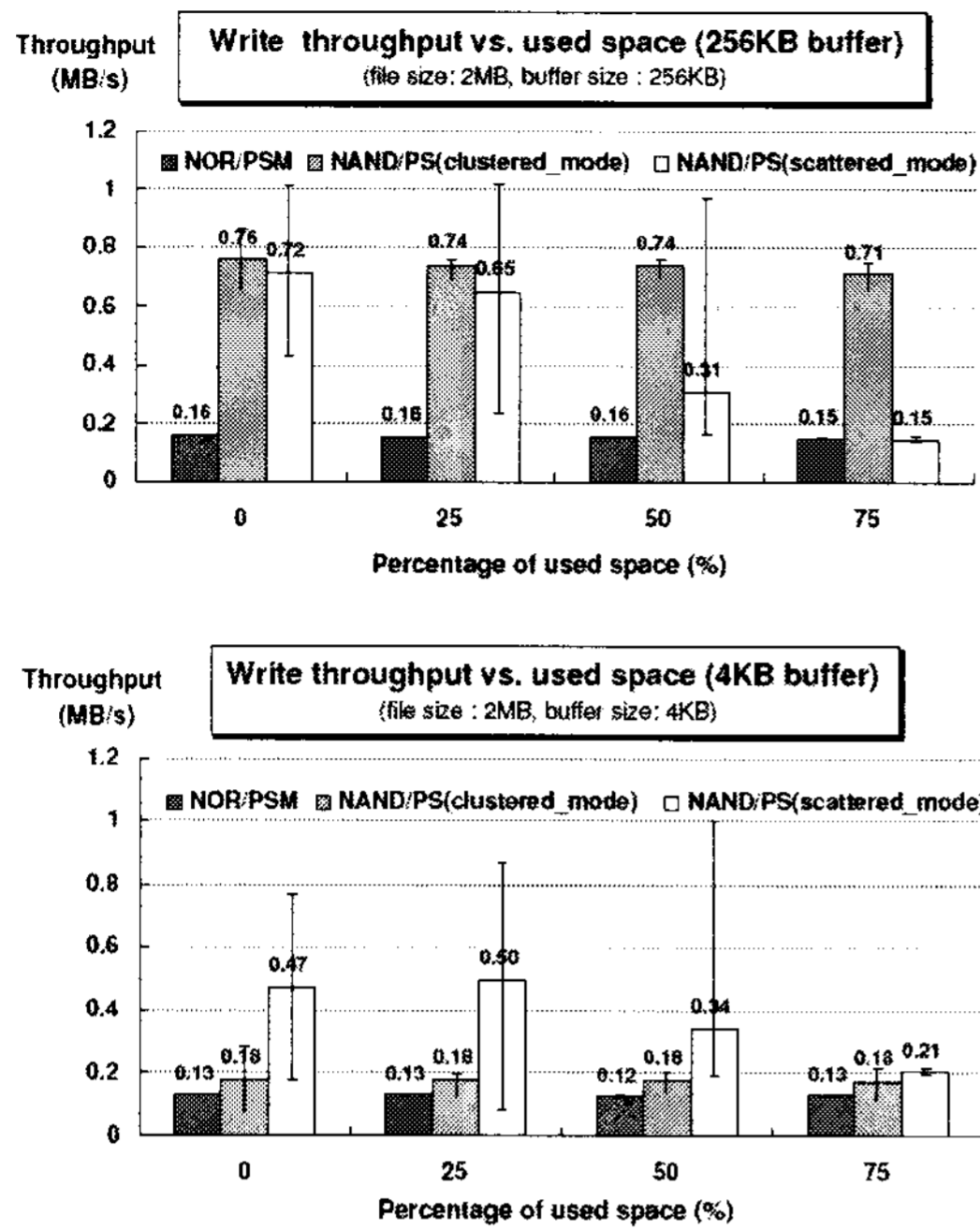
그림 4에서 보는 바와 같이, 작은 버퍼 크기는 낸드, 노어 모두에 대해 심각한 쓰기 처리량 감소를 가져온다. 특히, NAND/PocketStore 조합은 버퍼 크기에 매우 민감한 것으로 나타났다. PocketStore는 클러스터 모드에서 512 바이트 버퍼 크기에 대해 쓰기 처리량이 불과 0.03 MB/s에 불과했다. 이는 256 KB이상의 버퍼 크기에 대해 0.76 MB/s의 쓰기 처리량을 갖는 것과 크게 대비된다. 작은 버퍼 크기를 갖는 경우, FAT 테이블에 대한 많은 갱신 요청이 발생하며, 클러스터 모드에서의 쓰기 연산은 많은 지움 연산을 유발한다. 이는 인접한 논리적 섹터들이 보통 동일한 물리적 블록에 위치하기 때문이다. 반면, 스캐터 모드는 다양한 버퍼 크기에 대해 높은 쓰기 성능을 보이기는 하지만, 처리량의 편차가 매우 크다는 단점을 갖는 것을 알 수 있다. 이러한 현상은 다수의 블록에 걸쳐서 플래시 지움 연산을 유발시키는 물리적 페이지 할당 정책에서 그 원인을 찾을 수 있다. NOR/PSM 조합은 PocketStore에 비해 상대적으로 버퍼 크기에 덜 민감한 것을 알 수 있었다.

읽기 처리량에 대해서는 PocketStore가 높은 성능을 보였다. 특히, 클러스터 모드에서의 읽기 처리량은 3.36 ~ 4.08 MB/s에 달했다. 이는 노어 플래시 쓰기 처리량의 3 배가 넘는 수준이다.

요약적으로 기술하면, 전반적으로 읽기/쓰기 연산 모두에 대해 NAND/PocketStore 조합이 NOR/PSM 조합에 비해 높은 성능을 보였다. 낸드 조합의 처리량은 읽기, 쓰기에 대해 각각 최대 5.7 배, 4.8 배 높음을 알 수 있다.

### 3.3 사용 중 공간 크기

그림 5는 256 KB와 4 KB의 버퍼 크기에 대해 사용된 공간의 크기에 따른 쓰기 처리량을 보인 것이다. 단편화 (fragmentation) 효과를 성능에 반영하기 위해 처리량을 측정하기 전에 플래시 파일 시스템에 스트레스 테스트를 가하였다. 즉, 플래시 저장장치를 다양한 크기의 파일로 채운 후 무작위로 파일을 선택하여 삭제하는 작업을 반복하여 사용 중 공간의 비율이 의도한 수준에 달하도록 조치하였다.



[그림 5] 사용 중인 공간의 비율에 따른 쓰기/읽기 처리량

실험결과에 따르면 스캐터 모드인 경우 가용 공간이 작아짐에 따라 쓰기 처리량이 현저하게 줄어드는 것을 알 수 있다. 사용 중 공간의 크기가 75%일 때, 쓰기 처리량은 0.15 MB/s까지 감소하였는데 이는 최대 가능한 처리량의 1/5 수준에 불과한 것이다. 스캐터 모드의 이러한 특성은 처리량의 큰 편차를 갖는 특성과 함께 스캐터 모드를 제한된 환경에서만 사용하도록 하는 요인이 되고 있다. 스캐터 모드와 달리, 클러스터 모드와 PSM은 사용 중 공간의 크기에 거의 영향을 받지 않았다.

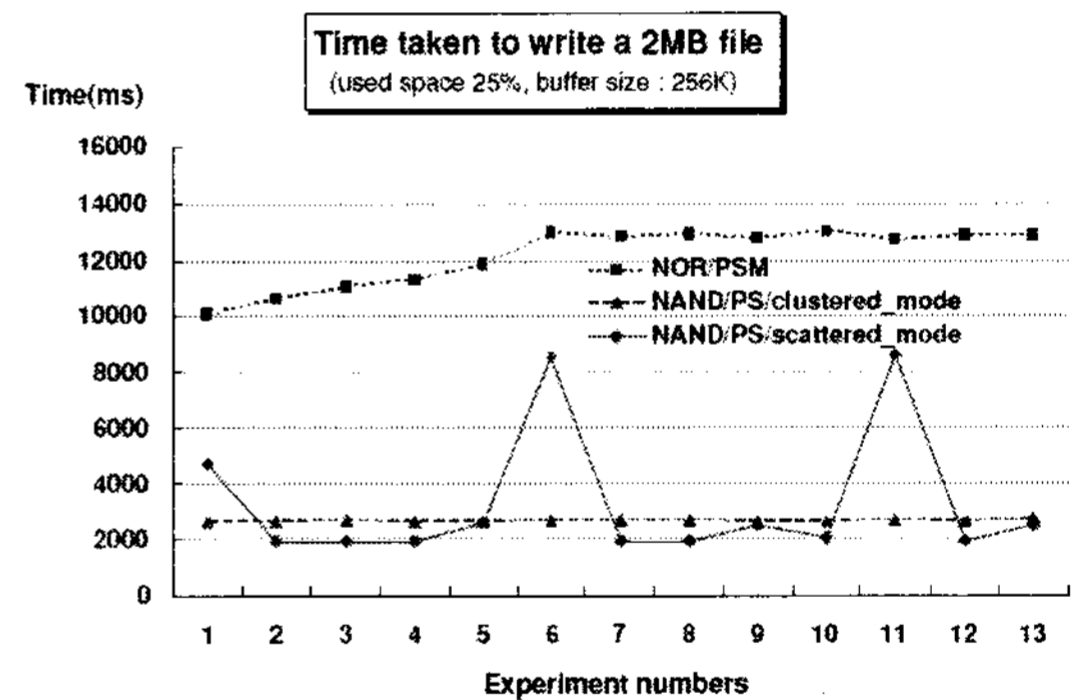
### 3.4 커널 수준 쓰기 캐시의 효과

실험 결과, PocketPC 커널은 플래시 기반 파일 시스템에 대해 쓰기 캐시를 제공하지 않는 것으로 드러났다. 커널-수준 캐시 메커니즘의 효과를 분석하기 위해 파일을 오픈할 때 FILE\_FLAG\_WRITE\_THROUGH 플래그를 지정할 때와 지정하지 않을 때 각각에 대해 벤치마크 프로그램을 실행하였다. 개발자 매뉴얼에 따르면 이 플래그는 시스템으로 하여금 중간 캐시를 거치지 않고 바로 디스크에 쓰도록 강제하는 역할을 한다. 하지만 기대와는 달리, 이 동기화 플래그는 플래시 메모리 기반 파일 쓰기에 대해 아무런 차이를 보이지 않았다.

반면 PSM은 고유의 쓰기 캐시를 갖추고 있는 것으로 추측된다. 그림 6은 벤치마크를 13회 반복하는 동안 2MB의 파일을 쓰는데 걸린 시간을 보인 것이다. PSM의

경우 초기 6회의 실험 동안에는 파일 쓰기 시간이 점차적으로 증가하는 것을 볼 수 있다. 이는 PSM이 쓰기 요청을 직접 플래시 메모리로 전송하기 보다는 요청의 일부를 RAM에 캐싱하기 때문이다. 즉, 실험이 진행됨에 따라 RAM이 부족하여 플래시에 더 많이 쓰게 되어 시간이 늘게 된 것이다. 실험 결과로부터 우리는 간헐적인 파일 쓰기의 경우 약 24%의 성능 개선을 얻을 수 있음을 알 수 있다.

마찬가지로 낸드의 경우에도 파일 시스템 (FAT) 혹은 FTL (PocketStore) 계층에 쓰기 캐시를 도입함으로써 상당한 성능 개선을 기대할 수 있다. 그러나, 버퍼 캐시를 FTL 계층 내에 포함하는 것을 고려할 때는 데이터의 일관성에 주의를 기울여야 한다. FTL 계층이 블록 장치 드라이버로서 동작하기 때문에 파일 시스템은 이 계층에 쓰여진 데이터가 커밋(commit)되었다고 가정한다. 이는 갑작스런 전원 중단과 같은 상황에서 데이터 일관성에 문제를 야기할 가능성을 갖는다. 아마도 이러한 문제 때문에 PSM의 설계시 FAT을 채용하기보다 고유의 파일 시스템을 고안하기로 결정했을 수도 있다.



[그림 6] 사용 중 공간의 비율에 따른 쓰기 처리량

3.2 절에서 기술된 바와 같이, PocketStore는 스캐터 모드에서 상당한 쓰기 처리량 편차를 보인다. 그림 6에 표현된 두 개의 피크는 그러한 편차를 잘 설명해준다. 피크는 FTL에 의한 페이지 회수 과정에 의해 유발된다. 즉 긴 시간이 소요되는 지움 연산과 쓰기 연산 때문이다. 클러스터 모드에서는 페이지 환수 연산이 전체 실험 회차에 걸쳐 두루 일어남에 비해 스캐터 모드에서는 페이지 환수가 매번 대규모로 일어난다.

## 4. 결론

본 논문은 낸드/노어의 두 가지 대표적인 플래시 메모



리 방식에 대해 파일 기반 입출력 성능을 비교 평가하였다. 즉, NAND/PocketStore와 NOR/PSM 이라는 두 가지의 대표적인 PocketPC 기반 플래시 파일 저장시스템의 성능을 비교하였다. “오버헤드를 고려한 입출력 처리량”을 성능 척도로 하여 입출력 버퍼 크기, 사용된 공간의 비율, 커널-수준 쓰기 캐싱 등의 관점에서 각각의 성능을 분석하였다. 다음은 주요 실험 결과를 요약한 것이다.

- NAND/PocketStore 구성은 NOR/PSM 구성에 비해 읽기/쓰기 처리량 성능에서 각각 최대 5.7 배, 4.8 배 높은 성능을 보인다.
- 낸드, 노어 구성 모두에 대해 쓰기 처리량은 사용자 수준 입출력 버퍼 크기에 크게 의존적이다. 특히, 버퍼 크기는 NAND/ PocketStore의 쓰기 처리량에 상당한 영향을 미친다.
- NAND/PocketStore의 경우, 클러스터 모드는 버퍼 크기에 민감한 반면 스캐터 모드는 가용 공간의 크기에 민감한 특성을 갖는다.
- PSM은 내부 쓰기 캐시를 포함하고 있는 것으로 추측된다. 이는 간헐적인 파일 쓰기에 대해 약 24%의 성능 향상을 가져오는 것으로 예상된다.

현재, PocketStore의 개선된 버전이 릴리즈될 예정이다. 이는 새로운 FTL 방식과 고유의 파일 시스템을 포함할 것이다. 이 FTL은 버퍼 크기와 사용된 공간의 크기에 대한 성능 편차를 줄이도록 특별히 고안되었다. 또한 새 버전의 PocketStore는 내부 파일 시스템에 의한 큰 성능 향상을 기대하고 있다. 이 파일 시스템은 FAT 호환성을 가지며 쓰기 캐시 및 갑작스런 전원 중단에 대한 복구 기능을 포함하고 있다.

## 감사의 글

저자는 삼성전자 소프트웨어연구소의 이광윤, 윤송호, 윤남현, 권철, 정현모 연구원에게 본 논문의 실험 및 분석에 있어서 베풀어준 아낌없는 지원과 조언에 감사의 말을 전하고 싶다.

## 참고문헌

- [1] Intel Corp., "Intel Persistent Storage Manager user's guide," <http://www.intel.com/design/flash/embedded.htm>.
- [2] Intel Corp., "Intel StrataFlash memory (J3)," <http://www.intel.com/design/flash/embedded.htm>.
- [3] Intel Corp., "Understanding the flash translation layer (FTL) specification," <http://www.embeddedfreebsd.org/Documents/Intel-FTL.pdf>.
- [4] B. Kim and G. Lee, "Method of driving remapping in flash memory and flash memory architecture suitable therefore," US Patent, no. 6,381,176, 2002.
- [5] J. Kim, et al., "A space-efficient flash translation layer for compactflash systems," IEEE Trans. Consumer Electronics, vol.48, no.2, pp.366-375, 2002.
- [6] H. Jo, et al., "FAB: Flash-Aware BufferManagement Policy for Portable Media Player," IEEE Trans. Consumer Electronics, vol.52, no.2, pp.485-493, 2006.
- [7] G. Lee, et al., "SSR (Simple Sector Remapper): the fault tolerant FTL algorithm for NAND flash memory," Int'l Tech. Conf. on Circuits/Systems, Computers and Communication, 2002.
- [8] M. Leinwander, "Memory subsystem for wireless systems," MEMCON, 2003.
- [9] S. Lim and K. Park, "An Efficient NAND Flash File System for Flash Memory Storage," IEEE Trans. Computers, vol.55, no.7, pp.906-912, 2006.
- [10] C. Park, et al., "Cost-efficient memory architecture design of NAND flash memory embedded systems," Int'l Conf. on Computer Design, pp.474-479, 2003.
- [11] C. Park, et al., "Energy-Aware Demand Paging on NAND Flash-based Embedded Storages," Proc. of the 2004 International Symposium on Low Power Electronics and Design (ISLPED'04), pp.338-343, 2004.
- [12] Samsung Electronics, "256M x 8 bit/ 128M x 16 bit NAND flash memory," <http://samsungelectronics.com>.
- [13] Samsung Electronics, "64Mx 8 bit, 32Mx 16 bit NAND flash memory," <http://samsungelectronics.com>.
- [14] Samsung Electronics, "S3C2410X: 32-bit RISC microprocessor user's manual," <http://samsungelectronics.com>.
- [15] G. Lawton, "Improved flash memory grows in popularity," IEEE Computer, vol. 39, no. 1, pp. 16-18, Jan. 2006.

**성 민 영**(Min-Young Sung)

[정회원]



- 1995년 2월 : 서울대학교 컴퓨터 공학과 (공학사)
- 1997년 2월 : 서울대학교 컴퓨터 공학과 (공학석사)
- 2002년 8월 : 서울대학교 전기컴퓨터공학부(공학박사)
- 2002년 9월 ~ 2006년 8월: 삼성전자 소프트웨어연구소 책임연구원
- 2006년 9월 ~ 현재 : 상명대학교 컴퓨터소프트웨어공학과 전임강사

<관심분야>

실시간시스템, 임베디드시스템, 운영체제, 멀티미디어 시스템 등