

커널 레벨에서의 SSD 시뮬레이터 디자인 및 구현

장보길, 김현빈, 임승호
한국의국어대학교 디지털정보공학과
e-mail:xearo@hufs.ac.kr

SSD Simulator in Kernel-level Design and Implementation

Bo-Gil Jang, Hyunbin Kim, Seung-Ho Lim
Dept of Digital Information, Hankuk Univ. of Foreign Studies.

요 약

SSD(Solid State Drive)는 다중-채널/ 다중-웨이 방식의 NAND 플래시 메모리를 이용하는 저장장치로서 기존 HDD(Hard Disk Drive)를 대체할 차세대 보조기억장치로 주목받고 있다. 하지만 SSD 와 같은 동작을 하는 커널레벨의 시뮬레이터가 존재하지 않아, 사용자 영역에서부터 실제 NAND 플래시 칩까지의 동작 원리를 파악하기 어렵다. 이러한 문제를 해결하기 위해 본 논문에서는 SSD 시뮬레이터의 설계 및 구현내용을 기술한다. 구현한 SSD 시뮬레이터는 다중-채널/ 다중-웨이 방식의 SSD 전체적인 동작 원리를 리눅스 커널 수준에서 파악할 수 있다. 또한 FTL 개발을 위한 환경을 제공할 뿐만 아니라, 사용자가 다양한 SSD 구조를 설계하여 성능을 예측할 수 있도록 한다.

1. 서론

최근 여러 미디어 매체들의 증가로 인하여 멀티미디어 콘텐츠 데이터의 양이 많아지고 있다. 이러한 대용량의 데이터를 효율적으로 관리하기 위한 저장장치의 성능향상 또한 요구된다. 이에 따라 기존에 대표적인 저장장치인 HDD에서 보다 읽기/쓰기 성능이 향상된 SSD 로 대체될 것으로 예상된다.

SSD는 다수의 NAND 플래시 메모리칩으로 구성된 저장장치로서 NAND 플래시 메모리의 여러 가지 특성들을 지니고 있다. 또한 다수의 칩을 다중-채널 / 다중-웨이 방식으로 동시에 접근하여 읽기와 쓰기 성능을 크게 향상시킬 수 있다. 현재 SSD 의 내부 구조 및 SSD 의 동작을 제어하는 FTL과 같은 소프트웨어 등에 대한 연구가 활발히 진행되고 있다. 하지만 관련 연구는 플래시 메모리가 장착된 장치나 제품으로 연구 환경이 구성되는데, 이러한 장치는 비싸고 다루기 힘든 단점이 있어, 비용이 증가하고 개발환경 구성에 제약이 따른다.

이러한 문제를 해결하기 위해서 실제 SSD와 같은 정확한 성능과 동작을 재현할 수 있는 정교한 SSD 시뮬레이터가 필요하며, 본 논문에서는 SSD 시뮬레이터의 전체적인 구조와 설계과정에 대해 설명하고, 향후 성능 개선방안에 대해 알아본다.

설계된 SSD 시뮬레이터는 4개의 플래시 메모리칩으로 구성되며, 각각의 칩이 파일시스템에 의해 동시에 접근하여 읽기와 쓰기가 가능한 다중-채널 방식으로 구현하였으

며, 각각의 칩을 효율적으로 사용하기 위한 웨어-레벨링, 가비지 콜렉션 기능을 향상시켰다

2. 본론

연구목표는 linux open platform의 MTD를 기반으로 SSD 시뮬레이터를 만드는 것이다. 다중-채널 / 다중-웨이 방식의 구현에 앞서 제한된 다중 칩의 SSD를 구현을 우선하여 설계했다.

본 프로젝트를 진행하는데 사용된 호스트 컴퓨터의 리눅스는 Fedora core12 이며, Kernel은 linux 2.6.31.5를 사용하였다. 또한 PXA272 임베디드 보드를 연결하여 교차 개발환경에서 진행하였다. 타겟보드의 kernel 은 linux 2.6.12를 사용하였으며 64MB의 메모리를 가지고 있다.

2.1 SSD 시뮬레이터 설계 및 아이디어

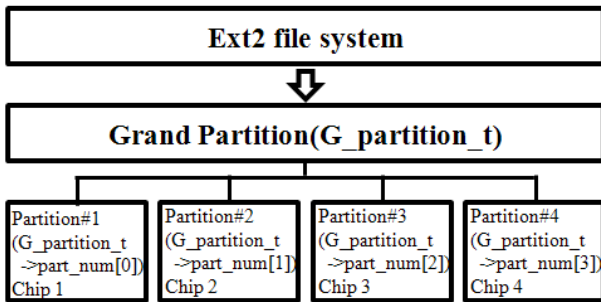
리눅스 커널 레벨에서의 SSD 시뮬레이터를 설계하는데 있어, 가장 중요한 것은 다음의 두가지로 나누어 볼 수 있다.

첫번째, linux MTD의 오픈소스는 NOR 플래시 메모리에 맞춰져 있어 NAND 플래시 메모리 환경에 맞게 수정해줘야 한다. NOR 플래시 메모리는 R/W 의 단위가 Byte 단위이며, erase가 없지만, NAND 플래시 메모리는 R/W 단위가 page 단위이며, overwrite 가 되지 않고 erase 를 블록단위로 수행한다.

MTD의 수정은 Byte 단위의 FTL 을 페이지 매핑 기

반으로 했으며, character device를 FTL형태로 포맷해주는 ftl_format.c 와 FTL의 전반적인 동작에 관여하는 ftl.c 를 수정하였다. ftl_format.c 에서는 FTL형식으로 쓸 때 page단위로 적도록 바꾸었고, ftl.c 에서는 Read/Write 등의 주요 동작을 page 단위로 동작하게 하였고, NAND flash memory의 특징상 순차접근 밖에 안되는 것을 보완하였다.

두번째는 linux MTD가 하나의 플래시 메모리 칩에 대응되도록 만들어 진 것이지만, SSD는 대용량이기 때문에 이를 해결하기 위해 여러 칩들을 사용할 수 있도록 변경해 줘야 한다. 본 프로젝트는 4개의 칩을 연동하여 시뮬레이터를 설계했다.



(그림 1) G_partition_t 구조

ftl.c 파일에서는 여러 개의 칩 정보를 관장할 수 있도록 상위의 구조체를 만들었다. G_partition 이라고 명명한 이 구조체는 4개의 칩정보를 가지고 있는 4개의 파티션의 주소를 받아서 4개의 칩에 접근할 수 있게 하였다.

3. 실험결과

```

NAND device: Manufacturer ID: 0x98, Chip ID: 0x6b
flash size: 4 MiB
page size: 512 bytes
OOB area size: 16 bytes
sector size: 8 KiB
pages number: 8192
pages per sector: 16
bus width: 8
bits in sector size: 13
bits in page size: 9
bits in OOB size: 4
flash size with OOB: 4224 KiB
page address bytes: 3
sector address bytes: 2
options: 0x62
Scanning device for bad blocks
    
```

(그림 2) 생성된 NAND 플래시 시뮬레이터 정보

그림 2 과 같이 NAND 시뮬레이터인 nandsim의 모듈 수정을 통하여 4MB*4개의 NAND 플래시 시뮬레이터를 생성했다. 생성된 시뮬레이터를 그림 3 과 같이 커널에서 관리하는 MTD장치 목록에 추가한 후 4개의 각각의 칩을 FTL형식으로 포맷하여 장치를 구성했다.

시뮬레이터를 통해 생성된 장치는 /dev/ftla 의 이름을 띄고 있으며, /mnt 에 마운트 시켜 그림 4 와 같은 결과를 확인할 수 있다.

```

[root@KROBO root]#cat /proc/mtd
dev:   size  erasesize  name
mtd0: 000c0000 00040000 "Bootloader"
mtd1: 00200000 00040000 "Kernel"
mtd2: 01000000 00040000 "Ramdisk"
mtd3: 02a00000 00040000 "Usr"
mtd4: 00400000 00002000 "NAND simulator partition"
mtd5: 00400000 00002000 "NAND simulator partition"
mtd6: 00400000 00002000 "NAND simulator partition"
mtd7: 00400000 00002000 "NAND simulator partition"
    
```

(그림 3) 커널에서 관리하는 MTD장치 목록에 추가

4MB 로 생성된 각각의 단일 칩들 4개가 하나로 인식되어, ftla 의 가용 용량이 약 14MB 가 된 것을 확인할 수 있다.

```

[root@KROBO mnt]#df
Filesystem      1k-blocks  Used Use% Mounted on
/dev/ram0        35692 23421  69% /
/dev/mtdblock3  43008  2116   5% /jffs
/dev/ftla       14113  3415  26% /mnt
    
```

(그림 4) 생성된 NAND 플래시 시뮬레이터 정보

4. 결론

SSD의 성능은 칩 컨트롤러에 달려있다. 이에 따라 제조사마다 고유한 컨트롤 알고리즘을 가지고 있으며, 활발히 연구가 진행되고 있다. 본 논문의 커널레벨의 SSD 시뮬레이터는 보다 다양한 어플리케이션들을 쉽게 테스트해 볼 수 있을 것이고, 다양한 SSD 컨트롤 방식을 구현할 수 있을 것이다.

현재 설계된 SSD 시뮬레이터는 4개의 NAND 플래시 메모리를 직렬로 연결한 형태로 구현되어 있다. 보다 정교한 시뮬레이터로 발전하기 위해 다중-웨이 방식의 구현이 최우선 과제이며, 더 나아가 여러 종류의 SSD구조와 내부 동작을 모방하고, 성능비교, 에러테스트, 최적화에 기여할 수 있는 연구가 필요하다.

참고문헌

- [1] C. Jim, Micron Technology Inc, "NAND Flash 101: An Introduction to NAND Flash and How to Design It In to Your Next Product", Nov, 2006.
- [2] Intel Corporation, "Understanding the Flash Translation Layer (FTL) Specification", Dec, 1998
- [3] J. Kim, J. M. Kim, S. H. Noh, S. L. Min, and Y. Cho, "A Space-efficient Flash Translation Layer for Compact Flash Systems", 2002.
- [4] Micron Technology, "NAND Flash Translation Layer (NFTL) 4.5.0 User Guide", 2010.