

# 리눅스 커널 블록 I/O 패턴 Profiling 기법을 이용한 안드로이드 장치의 Browser I/O 패턴 분석과 개선 방안

장보길\*, 이성우\*\*, 임승호\*,  
\*한국외국어대학교 디지털정보공학과  
\*\*이에프텍  
e-mail:xearo@hufs.ac.kr

## Browser I/O Patterns of Android Devices Analysis and Improvement Using Linux Kernel Block I/O Profiling Techniques

Bo-Gil Jang\*, Sung-Woo Lee\*\*, Seung-Ho Lim\*  
\*Dept of Digital Information, Hankuk Univ. of Foreign Studies  
\*\*Elixir Flash Technology Co., Ltd.

### 요 약

현재 컴퓨터 시스템에서 대표적인 성능 저하가 발생하는 부분은 블록 I/O 시스템이다. 안드로이드와 같은 모바일 장치 또한 위와 같은 성능 이슈를 가지고 있다. 본 논문에서는 리눅스 블록 레이어의 I/O를 tracing 해주는 blktrace를 안드로이드 장치에 적용하여 SQLite를 사용하는 Web Browsing 시의 I/O 패턴 분석과 성능 개선 방안을 제시 한다.

### 1. 서론

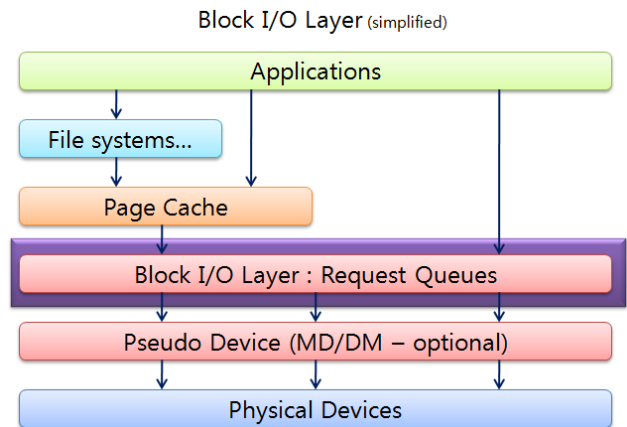
안드로이드 장치에서 Linux Kernel은 보안, 메모리 관리, 프로세스 관리, 네트워크 스택과 같은 핵심 서비스를 제공 하고 있으며 하드웨어와 소프트웨어 사이에서 추상 계층으로 동작 하고 있다. 그중 대표적으로 대부분의 모바일 저장 장치로 이용 되는 낸드 플래시 메모리에 대한 I/O역시 Linux Kernel에서 블록 장치 디바이스 드라이버로 관리 되고 있다.

낸드 플래시 메모리는 물리적인 특성상 데이터를 쓰기 전에 항상 지우기 연산을 해주어야 하는 특성이 있으며, 지우기 속도가 읽기와 쓰기 속도에 비해 현저히 느리다는 단점이 있다. 이러한 특성 때문에 낸드 플래시 메모리의 I/O 패턴에 따른 성능의 차이가 크게 발생 한다[1].

본 논문에서는 안드로이드 장치의 대표적인 성능 이슈 중 하나인 SQLite과 관련 깊은 Web Browsing 어플리케이션이 동작할 때의 블록 레이어 I/O 패턴을 분석하고, 성능 개선 방안을 제시 한다.

### 2. Linux Kernel의 블록 레이어 Tracing

블록 장치의 I/O 패턴을 분석하기 위하여 우리는 blktrace 라는 유틸리티를 사용 하였다[2]. Linux Kernel에서 수행 되며 유저 어플리케이션에서 요청 하는 실제 disk I/O 패턴을 추출 하는데 사용 된다.



(그림 1) 블록 I/O 레이어

Linux Kernel에는 블록 레이어의 I/O 이벤트를 남겨주는 인터페이스가 존재 한다. 이 이벤트를 이용 하여 blktrace는 I/O 패턴을 추출 하게 된다. 2.6.17 이상 버전의 커널이라면 설정 변경만으로도 사용 가능 하다.

### 3. 실험 대상 소개 및 실험 방법

실험 대상으로 사용한 안드로이드 장치는 Galaxy-Tab으로 안드로이드 2.2 버전(프로요) 이다. 저장 장치에 관련된 자세한 사항은 표 1에 기록 되어 있다.

<표 1> Galaxy-tab 안드로이드2.2 저장 장치 세부 정보

Galaxy-Tab Android 2.2				
	Capacity	Mount	File System	Size
RAM	512G			
NAND	OneNAND 4G In CPU-Pack	Code		
		/system	RFS	371M
		/data	RFS	540M
MMC	MobiNAND 16G	/sdcard	VFAT	15.5G

내부 저장 장치는 OneNand 4GB를 사용 하고 둘로 파티션을 나눠서 /system, /data로 마운트 하여 사용 하고 있다. 이때 두 파티션 모두 파일 시스템은 RFS이다. 외부 저장 장치는 MobiNand 16GB를 사용 하고 /sdcard로 마운트 하여 사용 한다. 각각의 파티션에 대한 주요 기능은 표 2와 같다.

<표 2> 각각 파티션의 주요 기능

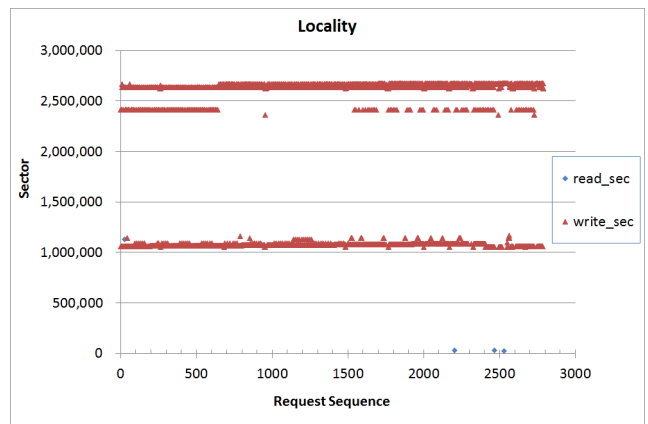
Partitions	Usage	Typical Access pattern
system	Android executables	Sequential Read
data	Sqlite files	Random R/W
sdcard	User data	Sequential R/W

blktrace는 git 저장장소에서 받을 수 있다[3]. 다운 받은 소스는 실험 대상에서 동작 할 수 있도록 툴체인 (ARM-EABI-2009q3 버전)을 사용 하여 컴파일 하였다 [4]. 컴파일된 이미지를 내장 메모리에 복사 한다. 안드로이드의 터미널을 이용하여 blktrace를 실행 시키고 Web page 접속하여 I/O 패턴을 추출 한다.

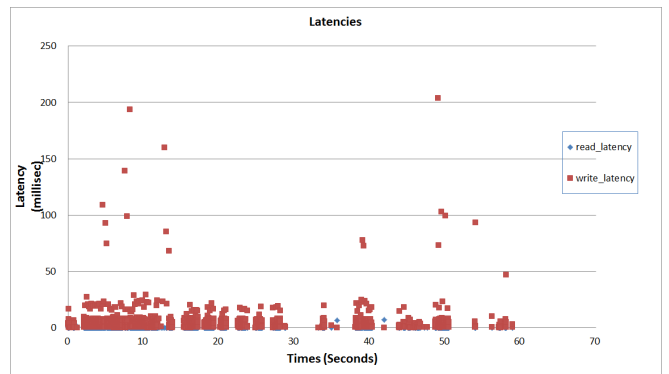
**4. 실험 결과 및 분석**

위 과정을 거쳐 얻은 결과는 그림 2와 3에서 나타내고 있다. 그림 2의 I/O 패턴을 보면 Web page를 여는 과정에서 상당히 많은 쓰기 연산이 일정 영역에 반복적으로 발생 하는 것을 볼 수 있다. 이 일정 영역은 Web Browsing시 집중적인 업데이트가 일어나는 데이터 베이스영역이라고 볼 수 있다. 낸드 플래시 메모리는 하드 디스크 드라이브와는 달리 데이터를 쓰기 전에 지워주어야 하는 물리적 특성을 가지고 있다. 그리고 읽기 / 쓰기 속도보다 지우기 속도가 현저히 느리다는 단점을 가지고 있다. 이러한 단점을 극복하기 위해 플래시 소프트웨어는 데

이터를 덮어 쓰는 작업 대신 지워진 프리 영역에 새롭게 쓰고 데이터의 논리 주소와 플래시 메모리의 물리 주소를 매핑해주는 매핑 테이블을 변화 시키는 작업을 한다. 만약 프리 영역이 일정 기준 이하로 떨어진 다면 기존에 데이터가 써진 영역 중 한 블록을 추출하여 유효한 데이터만 복사를 한 뒤, 해당 블록을 지워줌으로써 프리 영역을 확보한다. 이러한 과정을 가비지 콜렉션(GC, Garbage Collection)이라고 한다[5]. 사용자가 체감하는 속도에 많은 영향을 미치는 지표 Latency패턴을 측정한 그림 3을 보면 어느 시점에서 값이 커지는 것을 볼 수 있다. Latency값이 갑자기 증가하는 몇몇 구간은 GC가 발생했다고 볼 수 있다. GC수행은 사용자가 체감하는 가장 큰 성능 저하의 원인이 된다.



(그림 2) Web Browsing시 나타나는 Block I/O 패턴



(그림 3) Web Browsing시 나타나는 Latency 패턴

**5. 성능 개선 방안**

SQLite는 단일 파일로 데이터 베이스정보를 저장 하므로, 위에서 일어나는 집중적인 쓰기 연산은 단일 파일에 집중 되어 있는 것을 가정 할 수 있다. 낸드 플래시 메모리를 사용하는 대부분의 장치들은 쓰기 횟수가 줄어들게 된다면 GC의 횟수 또한 줄어들게 된다. GC횟수의 감소는 성능 향상을 의미 한다. 쓰기 횟수를 줄이는 대표적인 방법으로 나중 쓰기 방법이 있다. Web browsing시에 일어나는 블록 I/O는 단일 파일에 집중 되어 있다. 파일 시스

템에서 단일 파일에 집중 되는 쓰기 연산은 비교적 쉽게 알아차릴 수 있기 때문에 나중 쓰기 메커니즘을 적용해 볼 수 있을 것으로 판단된다.

## 6. 결론

모바일 플랫폼의 성능 개선 연구는 많은 분야에서 활발히 진행 중이다. 본 논문은 많은 모바일 장치에서 보조 기억 장치로 사용하고 있는 낸드 플래시 메모리의 특성과 실제 리눅스 커널에서 발생 하고 있는 블록 I/O 패턴을 tracing한 결과를 분석하여 개선 방안을 제시 하였다. 이를 Web Browsing상황 외에 여러 가지 종류의 어플리케이션들의 I/O 패턴을 분석 하여 공통점을 찾아내고 분석한다면 모바일 기기에 적합한 FTL, 파일시스템 등의 연구에 공헌할 수 있을 것이라 생각한다.

## 참고문헌

- [1] Intel Corporation, "Understanding the flash translation layer(FTL) specification", <http://developer.intel.com/>
- [2] Jens Axboe, Alan D. Brunelle "blktrace User Guide"
- [3] [git://git.kernel.org/pub/scm/linux/kernel/git/axboe/blktrace.git](http://git.kernel.org/pub/scm/linux/kernel/git/axboe/blktrace.git)
- [4] <https://opensource.samsung.com/>
- [5] 임승호, "가변 할당 방식을 이용한 플래시 메모리 저장장치의 데이터 중복 방지 방법", 대한전자공학회 2010년 하계종합학술대회