

안드로이드에서 파일 타입 별 IO 분석

김규일^o 임은령 원유집

한양대학교 컴퓨터·소프트웨어학과

gaya@hanyang.ac.kr, erlim@hanyang.ac.kr, yjwon@hanyang.ac.kr

IO analysis by each file type in Android

Kyuil Kim^o Eunryoung Lim Youjip Won

Department of Computer and Software, Hanyang University

요 약

스마트폰은 우리의 일상생활에 가장 밀접하고 혁신적인 발명품 중 하나이다. 스마트폰의 발명과 발전은 많은 사회적 이슈를 불러 일으켰고, 우리의 삶을 변화시켰으며 지금도 계속 변화하고 있다. 현재 스마트폰 사용자들은 기기의 성능, 어플리케이션의 활용도 등에 더욱 민감하게 반응하고 있다.

이 논문에서는 스마트폰의 성능하락의 원인이 되는 IO의 발생량을 알아보기 위해 실제 스마트폰 사용자들의 IO trace 정보를 수집 및 분석하였다. 분석 결과 SQLite와 SQLite journal파일들을 통해 발생하는 IO는 각각 28%와 35%로 많은 IO를 발생시키는 것을 확인할 수 있었다.

1. 서 론

스마트폰의 출시 이전에는 핸드폰, 카메라, MP3 등 용도에 따른 여러 가지의 디지털 기기를 지니고 다녔다. 하지만 스마트폰의 개발로 인하여 여러 가지의 기기들로부터 해방이 되었다. 전화통화, 사진촬영, 음악 감상 뿐만 아니라 언제 어디서든 스마트폰을 통하여 집안의 상태를 확인할 수 있고, 스마트폰과 연계하여 사용할 수 있는 스마트 시계, 구글 안경 등 많은 기기들이 출시되고 사용되고 있다. 이처럼 많은 분야에서 스마트폰을 통하여 사용자의 요구를 충분히 충족할 수 있게 됨으로써 스마트폰은 사용자들에게 없어서는 안될 꼭 필요한 생활 필수품이 되어버렸으며, 스마트폰 시장은 컴퓨터시장을 압도할 정도로 성장하였다.

스마트폰의 시장의 성장은 연구의 흐름 또한 변화시키고 있다. 이전에는 주로 기업형 서버[1]나 데스크톱 PC[2]의 성능에 대한 연구가 활발히 이루어 졌다면 최근에는 스마트폰 사용자의 스마트폰의 성능향상[3, 4]을 위한 연구가 활발히 이루어 지고 있다.

스마트폰의 하드웨어와 소프트웨어는 점점 더 강하고 빠르게 변화해가고 있으며, 스마트폰 사용자들은 예전보다 더 많은 서비스와 기능을 할 수 있는 다양한 어플리케이션들을 요구하고 있다. 스마트폰 사용자들의 이런 요구에 맞추어 어플리케이션들은 더욱 스마트하고, 빠르고, 화려하게 변화 하고 있다. 그로 인해 어플리케이션의 용량은 점점 증가하게 되었고, 현재 수백MB의 용량의 어플리케이션을 찾아보는 것은 어려운 일이 아니게 되었다. 어플리케이션의 대용량화가 되어감에 따라 대용량 어플리케이션의 실행으로 인해 어플리케이션에서 요구하는 IO의 발생량 또한 증가하고, 이는 스마트 기기의 성능저하를 유발하는 주요한 원인으로 꼽힌다. 우리는 이런 문제점을 해결하기 위해 실제 사용자가 사용하

는 스마트폰의 실시간 IO trace 정보를 수집 및 분석하여 어느 파일 타입이 많은 IO를 발생 시키고 있는지 확인해보기로 한다.

2. 설 계

기존에 스마트폰의 IO trace를 분석하는 MOST (Mobile Storage Analyzer)[5-7]라는 툴이 존재한다. 하지만 MOST는 IO trace를 분석하는 시간이 오래 걸리기 때문에 실시간으로 사용자의 IO trace정보를 수집 및 분석을 하기에 적합하지가 않다. 우리는 실시간으로 사용자의 IO trace 정보를 수집 및 분석을 위해 수정이 가능한 안드로이드의 커널 코드를 수정하여 실시간으로 사용자의 IO trace 정보를 수집할 수 있는 방법을 고안해냈다. 실시간으로 사용자의 IO trace 정보를 수집하는 메커니즘은 그림1과 같다.

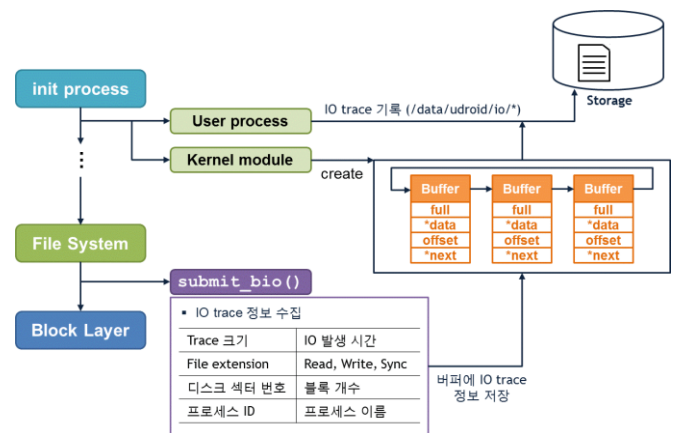


Figure 1. IO trace 수집 알고리즘

스마트폰의 부팅 시 버퍼를 관리하는 커널 모듈과 IO trace정보를 스토리지에 저장하는 유저프로세스를 실행시킨다. 커널 모듈은 임시로 IO trace정보를 저장하는

버퍼를 3개 생성 및 초기화 작업을 수행한다. 버퍼는 do_mmap()함수를 사용하여, 각각 4KB크기를 갖는다. 작업을 수행하고, 버퍼가 가득차게 되면 유저프로세스에게 가득찬 버퍼의 주소값을 전달한다. 유저 프로세스는 커널 모듈에게 3개의 버퍼들 중 가득찬 버퍼에 수집된 IO trace 정보를 요청하고 요청이 완료 될 때까지 대기한다. 만약 버퍼들 중 하나라도 가득차게 된다면 커널 모듈은 유저 프로세스가 요청한 가득찬 버퍼의 IO trace 정보를 전달한다. 커널 모듈로부터 IO trace정보를 전달 받은 유저 프로세스는 대기상태에서 깨어나 버퍼의 IO trace 정보를 스토리지에 파일 형식으로 저장하게 된다. 이때 저장되는 파일의 크기는 최대 1MB를 넘지 않는다.

만약 power off 시 버퍼에 저장된 IO trace 정보는 파일에 저장하지 않는다.

submit_bio()함수는 파일시스템에서 블록 레이어로 접근 시 모든 IO정보들이 접근하는 함수로 실질적으로 사용자들의 IO trace정보를 추출 및 버퍼에 IO trace 정보를 저장하는 작업을 수행한다. 이때 수집되는 데이터는 MOST를 통해 수집되는 데이터와 비슷하며 수집된 데이터 표 1과 같다.

표 1. IO trace 수집 정보

Index	비고
1	IO trace의 크기
2	IO가 발생한 시간(ms)
3	IO가 발생한 파일 타입(표 2참고)
4	Read, Write, Synchronous 표시
5	IO가 발생한 디스크 섹터 번호
6	IO가 발생한 블록의 개수
7	IO가 발생된 프로세스 ID
8	IO를 발생한 프로세스 이름

IO는 어플리케이션에 따라 다양한 파일을 통해 발생을 한다. 어플리케이션에서 IO를 발생시키는 파일의 타입을 크게 7개의 그룹으로 나누어 볼 수 있다. 각각의 그룹은 표2에 나타나 있으며, 해당 그룹에 따라 사용자들이 사용하는 어플리케이션들의 어떤 파일 타입이 IO를 발생시키는지 쉽게 확인할 수 있다.

표 2. 파일 타입 그룹

File extension	비고
db	SQLite 파일
db-journal	SQLite journal 파일
db-mjxxxx, db-wal	SQLite-temp 파일
jpg, 3gp, mp3, thumb, local	멀티미디어파일
so, dex, odex, apk	실행파일
localstorage, dat, xml, thumbdata3, cache	리소스파일
ttf등	나머지파일

3. 평가

이제 우리는 스마트폰 사용자들에게 IO trace 정보를 수집하도록 수정을 한 넥서스 5를 제공해주고 하루 동안 실제로 사용한 정보를 수집하여 분석을 할 것이다. 스마트폰 사용자들이 발생하는 IO trace들 중 우리가 중점적으로 볼 사항은 각 파일 타입 그룹 별 IO 발생 빈도 이다. IO 파일 타입 그룹을 분석함으로써 어느 파일 그룹이 잦은 IO를 발생시켜 성능을 저하시키는지 알 수 있다.

그림 2는 파일의 타입별로 IO 발생 횟수를 나타낸 그래프이다. 각각 Read, Write, ReadSync, WriteSync 별로 구분을 하였으며, ReadSync에서 IO가 발생하지 않았기에 ReadSync를 제외하였다.

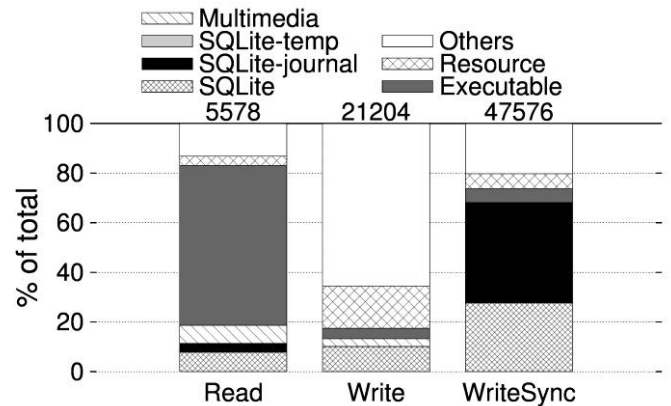


Figure 2. File type 별 IO 발생 비율

File type 별 IO 발생 비율 중 타입을 구분 짓지 않은 나머지 파일로 인하여 발생한 IO의 발생량이 전체 IO 발생 횟수 중 약 33%로 가장 많이 발생하였다. 이중 Write를 통해 발생하는 IO 발생 비율에서 타입을 구분 짓지 않은 나머지 파일로 인한 IO의 비율 약 65%로 어플리케이션 실행을 위해 필요한 파일 들이 많아졌으며, 이로 인해 발생하는 IO의 양이 증가됨을 알 수 있다. 또한 WriteSync를 보면 SQLite파일과 SQLite의 저널 파일(SQLite-journal)을 통해 발생하는 IO의 발생 비율이 각각 28%와 35%로 많은 IO를 발생시키고 있음을 확인할 수 있다. 이는 어플리케이션에서 WriteSync시 SQLite와 SQLite의 저널링으로 인해 발생하는 파일들의 IO 발생량이 무시하지 못할 정도로 큰 비중을 차지함을 알 수 있다.

4. 결론

이 논문에서는 실제 스마트폰 사용자들의 IO trace 정보를 수집 및 분석을 하였다. 스마트폰 사용자들은 자신이 실행시키는 다양한 어플리케이션으로 인하여 스마트폰의 성능에 영향을 끼치는 많은 IO를 발생시키고 있었다.

하루 동안 발생하는 IO를 file type별 발생 횟수를 나

누어 분석한 결과 타입을 구분 짓지 않은 나머지 파일로 인해 발생하는 IO가 하루 동안 발생하는 IO중 약 33%로 가장 많으며, SQLite와 SQLite 저널 파일로 인해 발생하는 IO의 양 또한 47%로 무시하지 못할 정도로 많이 발생함을 알 수 있었다.

스마트폰의 어플리케이션들은 사용자들의 다양한 요구를 만족하기 위해 더욱더 큰 크기와 많은 IO를 발생시킬 것이다. 어플리케이션들이 발생시키는 IO들 중 SQLite, SQLite 저널파일로 인해 발생하는 IO를 줄일 수 있다면 스마트폰의 성능향상을 기대할 수 있을 것이다.

5. Acknowledgement

본 연구는 지식경제부 및 한국산업기술평가관리원의 산업원천 기술개발사업(정보통신)의 일환으로 수행하였음. [No.10041608, 차세대 메모리 기반의 스마트 디바이스용 임베디드 시스템 소프트웨어]

참고문헌

- [1] A. Riska and E. Riedel, "Disk Drive Level Workload Characterization," in *USENIX Annual Technical Conference, General Track*, 2006, pp. 97-102.
- [2] T. Harter, C. Dragga, M. Vaughn, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau, "A file is not a file: understanding the I/O behavior of Apple desktop applications," *ACM Transactions on Computer Systems (TOCS)*, vol. 30, p. 10, 2012.
- [3] S. Jeong, K. Lee, S. Lee, S. Son, and Y. Won, "I/O stack optimization for smartphones," in *Presented as part of the 2013 USENIX Annual Technical Conference*, 2013, pp. 309-320.
- [4] Y. Nakamura, K. Nagata, S. Nomura, and S. Yamaguchi, "I/O scheduling in Android devices with flash storage," in *Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication*, 2014, p. 83.
- [5] S. Jeong, K. Lee, J. Hwang, S. Lee, and Y. Won, "Androstep: Android storage performance analysis tool," in *ME13: In Proc. of the First European Workshop on Mobile Engineering, Aachen, Germany*, 2013, pp. 327-340.
- [6] S. Jeong, K. Lee, J. Hwang, S. Lee, and Y. Won, "Framework for Analyzing Android I/O Stack Behavior: From Generating theWorkload to Analyzing the Trace," *Future Internet*, vol. 5, pp. 591-610, 2013.

- [7] *Mobile storage analyer (most)*. Available: http://dmclab.hanyang.ac.kr/sub/main_most.htm