

Recordroid: 안드로이드 사용자 입력 재현 및

반응 시간 측정 도구

홍경환^o, 신동군

성균관대학교 정보통신공학부

RedCarrot@gmail.com, dongkun@skku.edu

Recordroid: A Tool for User Input Replay and

Response Time Measurement on Android

Gyeonghwan Hong^o, Dongkun Shin

School of Information and Communication Engineering, Sungkyunkwan University

요 약

모바일 단말기에서는 데이터의 처리량보다 사용자 반응 시간이 성능 지표로 가장 적합하다. 하지만, 기존의 벤치마크 도구들은 반응 시간 측정에 적합하지 않다. 본 연구에서 개발한 Recordroid는 안드로이드 기기에서 응용 프로그램 실행 중에 사용자 입력을 기록하여 재현하는 도구다. 특히, 사용자의 입력과 기기 내부의 이벤트를 동기화하여 기기의 반응 속도를 측정할 수 있다. 이 도구를 통해 모바일 기기의 서로 다른 하드웨어 설정에서 실제 사용자 입력 패턴에 대해, 최소한의 오버헤드로 반응 시간 비교 실험을 할 수 있음을 보인다.

1. 서 론

최근 몇 년간 iOS, 안드로이드 등 모바일 플랫폼 기술이 발전함에 따라 스마트 폰과 태블릿 PC의 성능은 나날이 발전하고 있다. 모바일 운영체제 및 모바일 기기 제조사들은 하드웨어와 운영체제를 최적화하여 더 나은 성능을 제공하도록 노력하고 있다. 그러므로, 모바일 단말기에서 성능 측정 문제는 가장 중요한 이슈로 부각되고 있다.

기존에는 모바일 단말기의 성능을 측정하기 위해, 쿼드런트(Quadrant)[1]와 안드로벤치(AndroBench)[2] 같이 미리 정의된 작업을 수행하고 성능을 측정하는 '벤치마크' 도구들을 사용하였다. 쿼드런트는 안드로이드 기기에서 CPU, 입출력, 그래픽스 작업 등을 수행하며 전반적인 시스템 성능을 측정하는 도구이며, 안드로벤치는 SQLite 트랜잭션을 수행하며 입출력 성능만을 측정하는 도구다.

이러한 벤치마크 도구들은 모바일 기기에서 부하가 큰 작업을 할 때의 처리량(throughput)을 측정한다. 그러나 모바일 기기에서는 대형 서버나 PC에 비해 사용자가 부하가 큰 작업을 하는 경우는 굉장히 드물고, 오히려 부하가 적은 작업을 빈번하게 한다. 정작 최종 사용자들이 느끼는 성능은 부하가 큰 동작들의 처리량이 아니라, 부하가 적은 사용자 입력 동작에 대한 반응 시간(response time)이다. 또한, 이들 벤치마크가 사용하는 시나리오가 일반 사용자의 시나리오와는

거리가 있기 때문에 다양한 실제 사용자 시나리오의 입력을 재현하고, 반응 시간을 측정하는 도구가 필요하다.

몽키러너(MonkeyRunner)[3]나 시스템센스(SystemSens)[4], 테스트드رويد(TestDroid)[5]와 같이 최종 사용자의 사용자 입력을 직접 기록하거나 재현하는 도구는 이미 존재하지만, 모바일 단말기의 성능을 측정하는 데는 적합하지 않다. 몽키러너는 안드로이드 기기에서 사용자 입력을 재현하는 도구이지만, 패턴을 임의로 스크립트 코드로써 지정해야 한다[3]. 시스템센스는 안드로이드 시스템 전반에 걸쳐 기기의 사용 양상을 기록하지만, 사용자의 입력이나 반응시간은 기록하지 않는다[4]. 테스트드roid는 최종 사용자의 입력을 기록하고 재현할 수는 있지만, 응용 프로그램 한 개에 대해서만 테스트할 수 있어서 시스템 전반의 성능을 측정하는 데는 적합하지 않다[5].

본 연구에서 제작한 Recordroid는 기존 도구들의 한계를 모두 극복하였다. Recordroid는 실제 사용자 입력을 전체 시스템 단위로 응용 프로그램 내부 이벤트까지 기록하고 재현한다. 특히 사용자 입력과 기기 내부의 이벤트를 동기화하기 때문에, 성능 비교 목적으로 사용하기에 적합하다. Recordroid는 성능 지표로서 기기의 반응 시간도 측정한다.

본 논문에서는 먼저 Recordroid의 이벤트를 정의한다. 이를 바탕으로 Recordroid의 구조와 작동 과정, 구현에 대해 설명한다. 마지막으로, Recordroid를 이용하여 하드웨어 설정이 다른 안드로이드 기기에서 동일한 사용자 패턴을 재현하여 사용자 단계의 반응 시간을 비교한다.

이 논문은 2010년 정부(미래창조과학부)의 재원으로 한국연구재단-차세대정보-컴퓨팅기술개발사업의 지원을 받아 수행된 연구임(No. 2010-0020724).

2. Recordroid

Recordroid는 안드로이드 기기의 사용자 입력 이벤트를 재현하고, 서로 다른 성능을 내는 모바일 단말기의 사용자 단계 반응 시간을 알아내는 도구다. 먼저 Recordroid 이벤트를 정의하고, 구조와 구현에 대해 다룬다.

2.1. Recordroid 이벤트 정의

Recordroid는 사용자가 모바일 기기에 가하는 입력을 기록하고 재현하여 성능을 측정하는 것이 목적이다. 이 때문에 사용자와 모바일 단말기 간의 상호작용인, 입력 이벤트와 반응 이벤트를 먼저 정의하였다.

사용자 입력 이벤트는 최종 사용자가 모바일 기기에 입력 장치를 통해 명령을 내릴 때 발생하는 이벤트다. 사용자 입력 이벤트로는 터치 이벤트, 버튼 이벤트가 있다. 터치 이벤트는 특정 부위를 터치할 때 발생하며, 버튼 이벤트는 사용자가 하드웨어 버튼을 입력할 때 발생한다.

반응 이벤트는 사용자 입력에 대한 콜백 함수들의 호출에 의해 발생하는 이벤트다. 모바일 기기가 사용자로부터 입력 장치를 통해 명령받으면, 그에 대한 콜백 함수(callback function)를 수행하여 사용자의 입력에 반응한다. 반응 이벤트로는 액티비티(activity) 불러오기 이벤트, 뷰(view) 클릭 이벤트가 있다. 액티비티 불러오기 이벤트는 응용 프로그램의 액티비티를 불러올 때 발생하며, 뷰 클릭 이벤트는 안드로이드의 최상위 그래픽 객체인 뷰가 사용자의 입력을 받을 때 발생한다. 반응 이벤트는 Recordroid가 다음 사용자 입력 이벤트를 단말기에게 전달하기 전에 필요한 이벤트이다.

2.2. 구조

Recordroid는 그림 1과 같이, 안드로이드 프레임워크 수정 부분, 클라이언트, 서버, 실행기(runner)로 구성되어 있다.

먼저, 모바일 기기에서 사용자 입력 이벤트와 반응 이벤트를 구하기 위하여 안드로이드 프레임워크에 이벤트 생성자와 반응 송신부를 추가하였다. 이벤트 생성자는 Recordroid에서 필요한 사용자 입력 이벤트와 반응 이벤트를 만들어준다. 안드로이드 프레임워크 내부의 각 이벤트 생성 조건이 되는 부분에서 이벤트 생성자를 호출하여, 이벤트를 생성해주도록 하였다. 그 중 반응 이벤트는 반응 송신부에서 Recordroid 실행기에 실시간으로 넘겨준다. 이를 통해 사용자 입력과 기기 내부의 이벤트를 동기화할 수 있다.

클라이언트는 모바일 기기 상에서 사용자 입력 이벤트와 반응 이벤트를 실시간으로 기록하여, 호스트 PC에 있는 서버로 보낸다. 서버는 모바일 기기의 클라이언트로부터 들어온 기록을 실행기에서 다시 실행할 수 있도록 파이썬 스크립트로 번역하거나, 성능을 분석하기 위해 기록 파일로 번역한다.

실행기는 안드로이드에 기본으로 내장된 사용자 입력 자동화 도구인 몽키러너(MonkeyRunner)를 본 연구의 목적에 맞게 개량하였다. 서버가 번역한 스크립트를 실행기에 넣어주면, 실행기는 주어진 스크립트에 맞게 안드로이드 기기에 입력 명령을 내린다. 실행기가 단말기에 입력 명령을 주게 되면, 반응 수신부를 통해 단말기로부터 그에 대한 반응을 받은 후에야 다음 명령을 내린다.

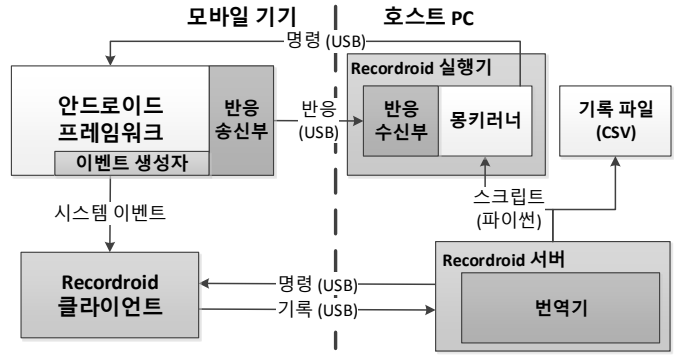


그림 1. Recordroid의 구조

2.3. 구현

클라이언트는 안드로이드 응용 프로그램 형태로 작성하였다. 서버와 실행기는 호스트 PC의 플랫폼 독립성을 위해, 자바 및 파이썬 응용 프로그램으로 제작되었다. 안드로이드 프레임워크 내부에 위치하게 될 반응 송신부와 실행기의 반응 수신부 사이, 클라이언트와 서버 사이에는 USB로 통신한다.

이벤트 생성자는 안드로이드에 프레임워크 디버깅 용도로 내장된 이벤트로그(EventLog) 모듈을 사용하여 이벤트를 생성한다. 실행기 내에 위치한 반응 수신부는 몽키러너의 플러그인 형태로 만들어, 몽키러너의 기존 기능도 사용하면서 Recordroid의 목적에 맞게 기능을 확장하였다.

3. 실험 및 분석

3.1. 실험 환경

본 연구에서는 Recordroid를 이용하여 서로 다른 하드웨어 설정에서 같은 패턴의 사용자 입력을 재현하여, 사용자 단계의 반응 속도와 Recordroid 도구 자체의 오버헤드를 측정하였다. 실험은 안드로이드 태블릿인 넥서스 10에서 수행하였다. 넥서스 10은 듀얼 코어 AP인 엑시노스 5250과 2GB의 LPDDR3 RAM을 내장하고 있으며, 운영체제는 리눅스 커널 3.4 기반의 안드로이드 4.2를 사용하고 있다. 이 기기에서 커널 매개 변수를 변경하여, AP의 코어 개수는 1~2개, 최대 메모리 크기는 1250~2000MB까지 바꾸어가면서 총 8가지의 하드웨어 설정에서 실험하였다.

실험 시나리오에 사용된 응용 프로그램은 안드로이드 기본 내장 웹 브라우저, 페이스북(Facebook), 테라리아(Terraria), 계산기, 시계, 유튜브(Youtube)로 총 6개다. 응용 프로그램들을 차례로 실행하면서, 내부에서 여러 동작들을 수행하였다. 웹 브라우저에서는 두 개의 탭을 생성하여 구글, 위키피디아, 아마존, 이베이, BBC 웹 페이지에 접속하여 각 페이지들을 스크롤하였으며, 페이스북에서는 뉴스피드를 스크롤하며 새 소식들을 네트워크로부터 불러오도록 하였다. 게임인 테라리아에서는 단순한 가상 조이스틱 동작을 수행하였고, 유튜브에서는 동영상 한 개를 재생하였다. 이 워크로드는 안드로이드 기본 응용 프로그램과 구글 플레이 스토어 인기 게임 응용 프로그램으로 이루어져 있어, 벤치마크 도구와 달리 사용자들의 사용 양상을 반영할 수 있다.

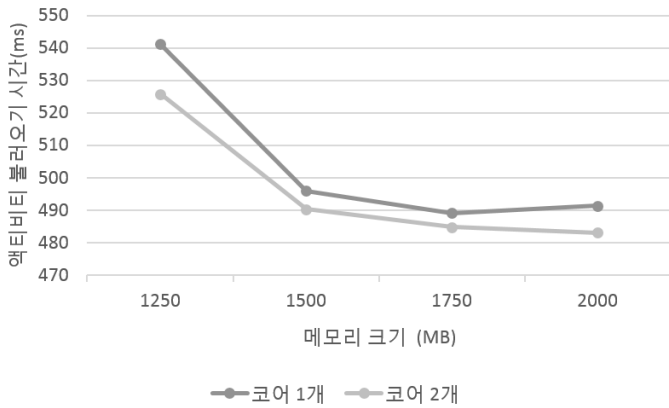


그림 2. AP의 코어 개수와 메모리 크기의 변화에 따른 액티비티 불러오기 시간의 변화

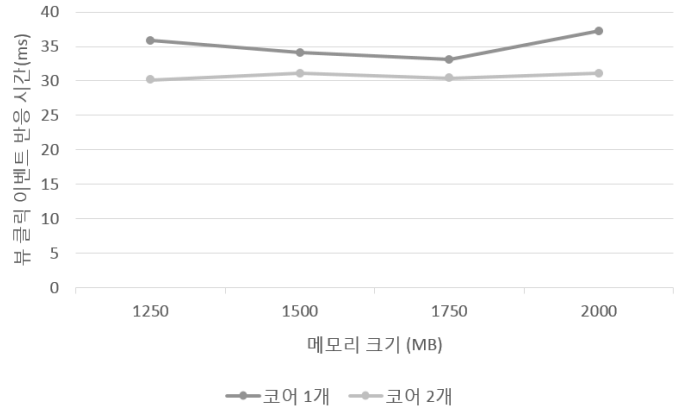


그림 3. AP의 코어 개수와 메모리 크기의 변화에 따른 뷰 클릭 이벤트 반응 시간의 변화

3.2. 하드웨어 설정에 따른 반응 시간 변화 측정

사용자 단계의 반응 속도로는 액티비티 불러오기 시간과 뷰 클릭 이벤트에 대한 반응 시간을 측정하였다. 이들은 Recordroid가 측정하는 반응 이벤트로부터 알아낼 수 있다.

그림 2에 따르면, 액티비티 불러오기 시간은 AP의 코어 개수가 늘어남에 따라 약 4~15ms(0.8~2.8%)로 소폭 감소하고, 메모리 크기가 늘어남에 따라 로그 함수 형태로 감소한다. 특히 메모리 크기가 1250MB일 때와 1500MB일 때의 액티비티 불러오기 시간 차이가 35~45ms(6~8%) 정도로 다른 경우보다 큰 차이를 보인다.

그림 3에 따르면, 뷰 클릭 이벤트 반응 시간은 메모리 크기에는 관계 없이 거의 균일하지만, 코어의 개수가 늘어남에 따라 반응 시간이 2~6ms(8~16%) 향상된다. 이 실험을 통해 액티비티 불러오기 시간은 메모리 크기에 큰 영향을 받는 반면, 뷰 클릭 이벤트 반응 시간은 AP의 코어 개수에 영향을 많이 받는다는 점을 알 수 있다.

3.3. Recordroid의 오버헤드 측정

Recordroid를 적용하려면 사용자 단계 프로그램에 해당하는 Recordroid 클라이언트를 설치하고, 안드로이드 플랫폼 펌웨어 코드 수정하여 그림 1의 반응 송신부에 해당하는 시스템 서비스를 삽입해야 한다. 따라서 반응 송신부와 Recordroid 클라이언트가 단말기에서 오버헤드를 일으킨다.

넥서스 10에서 동작하는 두 개의 모듈의 PSS(Proportional Set Size)를 덤프시스(dumpsys)라는 안드로이드 시스템 정보 추출 도구로 측정하였다. Recordroid 클라이언트의 PSS는 유휴 상태에서 약 4MB이며 기록 중일 때는 약 16MB까지 늘어난다. 반응 송신부의 PSS는 유휴 상태에서는 1MB이며 재생 중일 때는 3MB까지 늘어난다. 즉, Recordroid로 인한 메모리 오버헤드는 유휴 상태에서 약 5MB, 기록 중일 때는 17MB, 재생 중일 때는 7MB다. 2GB의 메모리를 지닌 넥서스 10에서 메모리 오버헤드는 전체의 0.25~0.85%에 불과하다.

CPU 오버헤드는 거버너(governer)를 최대 클럭을 내도록 하는 퍼포먼스(performance)로 설정한 후, proc 파일 시스템을 통해 측정하였다. 기록 중일 때와 재생 중일 때

모두 유휴 상태보다 CPU 사용량이 10% 정도 상승하였다. Recordroid가 기록하는 사용자 패턴이 부하가 적은 것을 위주인 점을 감안하면, 기록 중이거나 실행 중인 동작에 영향을 미칠만큼의 CPU 오버헤드는 아니다.

4. 결론 및 향후 계획

본 연구에서는 안드로이드 기기의 사용자 입력을 재현하고, 서로 다른 성능을 내는 모바일 기기의 사용자 단계 반응 시간을 알아내는 도구인 Recordroid를 제작하였다. 이를 바탕으로 서로 다른 하드웨어 설정에서 똑같은 사용자 입력 패턴을 실행했을 때, 액티비티 불러오기 시간은 메모리 크기의 변화에 따라 최대 8%의 향상되며, 뷰 클릭 이벤트는 AP의 개수의 변화에 따라 8~16% 향상됨을 보였다. Recordroid의 메모리 오버헤드는 0.25~0.85%에 불과하고 CPU 사용량은 10% 정도 상승하여, 다른 동작에 큰 영향을 미치지 않음을 보였다.

앞으로는 Recordroid에서의 성능 측정 대상을 사용자 단계 뿐만 아니라 운영체제 커널 단계까지 포함하고 사용자 입력 및 반응 이벤트의 종류를 늘려, 통합적인 성능 측정 도구를 개발할 예정이다. 또한 최적화된 안드로이드 메모리 및 스토리지 관리 기법을 개발하고, Recordroid를 바탕으로 이들의 효과를 실제 사용자 입력에 기반하여 입증할 것이다.

참고 문헌

- [1] Quadrant, <http://www.aurorasoftworks.com>, 2009.
- [2] Je-min Kim. et. al., "AndroBench: Benchmarking the Storage Performance of Android-Based Mobile Devices", *Frontiers in Computer Education*, p.667-674, 2012.
- [3] Milano, Diego Torres. "Android application testing guide", 2011.
- [4] Hossein Falaki, et. al., "Systemsens: a tool for monitoring usage in smartphone research deployments.", *Proceedings of the sixth international workshop on MobiArch*, 2011.
- [5] Jouko Kaasila, et al. "Testdroid: automated remote UI testing on Android." *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*, 2012.