
젤리빈 환경에서 응용프로그램 성능 분석을 위한 응답시간 분석 도구 구현

정서영* 허신*

*한양대학교

Implementation of response time analysis tool for application
performance analysis in Jellybean environment

Seo-Young Jung* · Shin heu*

*Hanyang University

E-mail : existor@hanyang.ac.kr

요 약

스마트기기가 대중화 되면서 그에 따른 많은 응용프로그램이 개발 되고 있다. 특히, 스마트 폰 시장에서 높은 점유율을 가지고 있는 안드로이드의 경우 많은 응용 프로그램이 개발되고 있고, 그에 따라 리소스가 사용 가능한 공간이 충분 함에도 불구하고 응용 프로그램의 서비스 프로세스가 원활히 돌아가지 않아 응답시간이 지연될 경우가 발생 한다. 본 논문에서는 응용 프로그램의 응답시간이 지연되는 상황을 분석하고 문제점을 찾아 해결하기 위한 방안을 모색 할 수 있도록 하는 응용프로그램 서비스에 대한 응답시간을 파악하는 분석도구를 구현하였다.

ABSTRACT

A lot of applications are being developed as a result of popular smart machinery. Especially, Android OS that has high market share in smart phone market, applications are being developed and resulted. response time is delayed because service process of application do not work well although the space for using resource is enough. The purpose of this paper, analysis tool analyze response time of application to grope for an answer to the question about delayed response time of application.

키워드

Android, App, Application, Jellybean, performance

1. 서 론

스마트 기기 사용자의 증가로 인하여 스마트 기기에 대한 개발이 각광받고 있는 추세이다. 특히 오픈 소스인 안드로이드 운영체제를 사용하는 비중은 스마트 기기 시장에서 약70%를 차지하고 있다. 그에 따라 안드로이드 운영체제에서 사용되는 응용프로그램개발과 그와 관련한 연구 또한 증가 하고 있다.

안드로이드를 활용한 소프트웨어 개발은 플랫폼 개발과 응용프로그램 개발로 나뉜다. 이 중 개인이 쉽게 접하고 제작할 수 있는 응용프로그램 개발은 하드웨어 사양이 충분함에도 불구하고, 제

대로 된 응용프로그램을 만들지 않아 서비스 프로세스가 원활히 돌아가지 않게 만든다. 예를 들면, 응용프로그램의 실행 시간이 길어지거나, 응용프로그램이 제대로 종료 되지 않아 백그라운드에서 계속해서 돌아가도록 만들어 두어 운영체제가 실제로 사용할 수 있는 리소스를 줄어들게 만든다. 이는 사용자가 해당 응용프로그램에 대한 신뢰성을 잃게 만들고, 불편함을 주게 한다.

따라서 본 논문에서는 해당 문제점을 해결하기 위한 방안을 모색 할 수 있도록 안드로이드 플랫폼에서 응용프로그램 서비스를 분석하여 서비스에 대한 응답시간을 파악하는 분석 도구를 제안 및 구현하였다.

II. 기존의 응용 프로그램 응답 시간 분석 방법 및 관련 연구

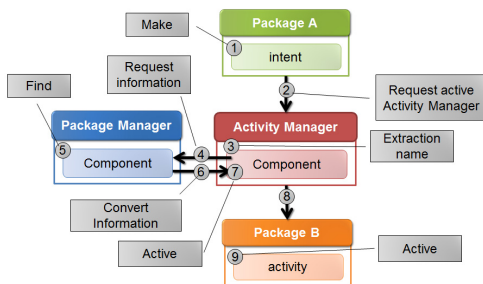
i. 응답시간 분석 방법

기존의 응용프로그램 응답시간 분석 방법으로는 프로그램을 제작하는 응용프로그램 프로그래머가 응용프로그램을 제작 할 때 각 액티비티 주기마다 Log를 작성하는 방법과, 이미 제작된 응용프로그램을 디 컴파일 하여 분석 하고, 분석 된 과일을 수정하는 방법이 있다. 첫 번째 방법은 프로그램을 만드는 프로그래머가 직접 넣어 확인 하는 방법으로, 이 방식은 일반 사용자가 확인 할 수 있는 방법이 없다. 두 번째 방법으로는, 바이트 형태의 class파일을 생성할 수 있는 JAVA 소스로 이루어진 apk파일을 디 컴파일 할 수 있는 방법이 있으나, 일반 사용자가 쉽게 디 컴파일 할 수 없다는 점이 있다

ii. 관련 연구

관련 연구로는 [2],[3],[4]가 있으며, 논문[2]은 플랫폼을 수정하여 일반 사용자가 손쉽게 사용할 수 있는 것은 같으나, 스마트폰 디버깅이 목적으로 쓰여진 논문으로 본 논문과의 목적이 다르다. 논문[3]에서는 Intent를 모니터링 하여 응용프로그램 설치 전 유해 응용프로그램을 차단하는 목적으로 이 역시 본 논문과 목적이 다르다. [4]에서는 테스트한 운영체제가 구 버전이고, 액티비티 주기와 캐시에 따른 응답속도만 체크 한다. 이는 응용프로그램이 실행될 때의 생명 주기만 보기 때문에 응용프로그램이 새로운 응용프로그램을 실행할 때 걸리는 응답시간을 측정하기가 어렵다.

III. 응용 프로그램 응답시간 분석 도구 제작 시 고려해야 할 사항



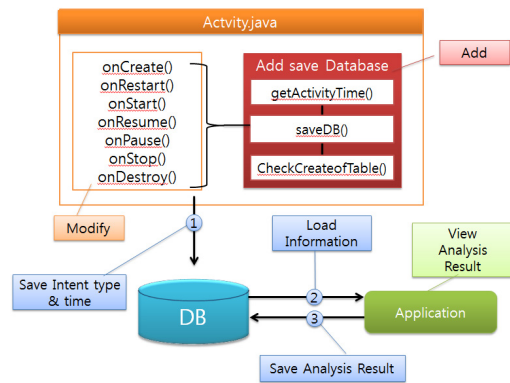
(그림 1) Component간 실행 과정

첫 번째로, (그림 1)과 같이 안드로이드는 컴포넌트간 통신을 할 때 Intent라는 비동기 메시지에

의해 활성화가 된다[5]. (그림 1)은 A라는 응용프로그램이 B라는 응용프로그램의 컴포넌트를 활성화시킬 때 생성되는 Intent의 흐름도이다. Intent는 서비스와 액티비티에게 보내지는 메시지를 가지고 있다. 이를 이용 하여 액티비티간 전환이나 컴포넌트 활성화를 요청할 때 등 여러 정보를 수집하기 위해 Intent 모니터링이 필요하다.

두 번째로, 액티비티 간의 응답시간을 수집하기 위해, 액티비티 주기가 고려 되어야 한다. 액티비티 생명주기 메소드간에 발생하는 entire lifetime이나 visible lifetime, foreground lifetime등을 알아야 정확한 액티비티간의 응답시간을 수집할 수 있기 때문이다.

IV. 구현 및 실험 결과

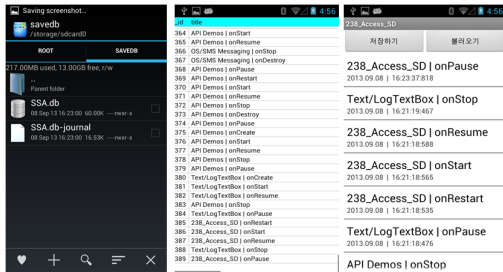


(그림 2) 구현된 분석 도구 시나리오

3장 (그림 1)에서 살펴 보았던 것처럼 Intent 모니터링과 액티비티 생명주기 간의 응답시간을 받아 처리해주는 과정을 분석하고, 해당 클래스를 수정한다. Intent가 가장 먼저 전달이 되는 Activity클래스를 보면 각 액티비티가 실행시키기 위해 사용 된다는 것을 알 수 있다. 그리 하여, (그림 2)와 같이getActiveTime() 메소드를 추가하여 각 액티비티 생명주기 마다 getActiveTime() 메소드를 호출하고, getActiveTime() 메소드에서 Intent type과 각 액티비티 생명주기마다 실행되는 시간을 얻어 온다. 이를 통해 각 생명주기 메소드 사이에서 발생하는entire lifetime나 visible lifetime의 시간 까지 얻어 올 수 있다. 해당 정보를 얻어 오게 되면, saveDB() 메소드를 사용하여 정보들을 저장한다. CheckCreateofTable()메소드는 해당 Database가 제대로 생성 됐는지 확인 하는 메소드 이다. 저장된 메소드는 sql DB로 저장 되어 정보를 읽어 오는 응용프로그램을 통하여 볼 수 있다.

실험을 위해 기존의 있던 안드로이드 플랫폼을 수정하였다. 수정된 플랫폼에서 각 응용프로그램

그램을 실행 하여 액티비티 마다 응답시간을 받아오는지 실험을 하였다. 해당 플랫폼에 설치된 응용프로그램을 실행 시키고, 사용을 한 후 새로운 응용프로그램을 실행 시키거나, 사용 중인 응용프로그램을 종료 하였다. 이후 DB정보를 읽어 오는 응용프로그램을 제작하여, 실험 결과를 확인 하였다.



(그림 3) 실험 결과

(그림 3)을 보면 응용프로그램의 응답 시간을 알아보기 위해 분석 툴을 사용한 결과이다. 첫 번째 그림은 DB가 외부 저장장치에 저장되어 있음을 알 수 있다. 두 번째 그림은 해당DB를 열어 본 결과 화면이다. 세 번째 그림은 사용자가 쉽게 DB를 읽어 올 수 있도록 응용프로그램을 제작 하여 확인 한 화면이다. 실행된 응용프로그램 이름, Intent 정보, 실행된 시간을 알 수 있다.

V. 결론 및 향후 과제

본 논문에서는 지금까지 제안한 모델을 토대로, 기존의 안드로이드 플랫폼을 수정하여 응답 시간 분석 도구를 구현하였다. 일반 사용자에게 보다 빠르고 편리한 환경을 제공하기 위해서는 제작된 응용프로그램의 문제점을 빠르고 정확하게 찾는 것이 중요하다. 하지만, 기존의 논문은 한 개의 응용프로그램이 실행 될 때의 생명 주기만 보기 때문에 컴포넌트가 활성화 되어 액티비티간 전환을 할 때 걸리는 응답시간을 구하지 않는다. 이는 결국 자세한 분석이 되지 않음을 의미하고, 조금 더 정확하게 문제점을 찾기 힘들다는 것을 의미한다. 그리고 더 나아가 앞서 이야기한 사용자의 불편함을 만드는 계기가 된다. 이는 곧 프로그램에 대한 신뢰도를 떨어트리고 해당 응용프로그램을 기피하게 만들기 때문에, 이를 해소 하기 위한 응답시간 분석 도구를 구현하였다.

향후 연구로는 이 연구를 토대로 Intent를 정밀하게 분석 하여 사용자가 실행하는 응용프로그램의 정밀한 분석 및 성능 개선을 계획하고 있다.

참고문헌

- [1] J. Kyong, S. Lim, M. Lee, S. Her, “USB-based Debugging Method for Android Smart Phones” Poc. of the 6th KIISE Symposium, vol.39, no.2, Apr., 2012. (in Korean)
- [2] 여세환, 이진, 김정선 “안드로이드 플랫폼에서 인턴트 모니터링을 통한 유해 응용프로그램 차단 방법” 한국정보과학회 제19권, 제5호, 2013년 5월
- [3] Kyosuke Nagata Saneyasu Yamaguchi, “An Android Application Launch Analyzing System” in Proc. IEEE Volume: 1, pp. 76-81, 2012
- [4] Google API Guide and API documentation [Online]
<http://developer.android.com/guide/components/intents-filters.html>, 2013