

# 안드로이드 응용 프로그램의 런칭시간 실시간 측정 도구 구현

김학봉, 임제헌, 김지홍  
서울대학교 컴퓨터공학부  
e-mail:{haknalgae, brownrabbit, jihong}@davinci.snu.ac.kr

## Development of a Tool for Measuring the Launching Time of Android Applications in Real-time

Hakbong Kim, Jehun Lim, Jihong Kim  
\*Dept. of Computer Science and Engineering, Seoul National University

### 요 약

응용 프로그램의 런칭시간에 관한 많은 연구를 통해 런칭시간이 길수록 사용자가 불편을 느낀다는 것이 확인되었으며, 이에 런칭시간을 최적화하는 다양한 연구가 있어왔다. 이런 연구의 일환으로 본 논문에서는 응용 프로그램의 런칭시간을 실시간으로 측정하는 도구를 개발했다. 이 도구는 안드로이드 프레임워크 내에 존재하며 응용 프로그램이 화면에 모두 그려지는 순간을 런칭의 끝으로 보고 런칭시간을 실시간으로 측정한다. 이 도구는 안드로이드 시스템의 응용 프로그램 런칭시간 최적화 기법에 사용될 수 있다. 또한 이 도구를 이용하여 모바일 시스템에서 “미리 읽기”(prefetching) 기법이 안드로이드 시스템에서 얼마나 효과가 있을지를 확인하기 위한 실험을 수행하였다. 실험 결과, “마리 읽기” 기법은 안드로이드 시스템에서 효과가 작은 것을 확인하였다.

▶ Keyword : 런칭타임(launching time), 안드로이드(android), 측정 도구(measurement tool)

## I. 서 론

응용 프로그램의 런칭시간은 사용자의 만족에 큰 영향을 주기 때문에 이를 줄이고자 하는 다양한 연구가 있었다. 하지만 모바일 환경에의 런칭시간 최적화 관련 연구는 아직 없는 상황이다.

이전 연구에서 호스트 컴퓨터를 이용하여 런칭시간을 측정하는 도구를 개발하였다.<sup>[1]</sup> 하지만 시스템이 자체적으로 런칭시간을 측정하고 최적화를 할 수 있어야만 상용 디바이스에도 이를 적용할 수 있기 때문에 실시간으로 런칭시간을 측정하는 도구가 필요하다. 이에 본 연구는 안드로이드 응용 프로그램의 런칭시간을 실시간으로 측정하는 도구를 개발하였다.

또한 이 도구를 이용하여 런칭시간을 어떻게 최적화해야 하는지를 탐색하였다. 런칭시간을 줄이는 대표

적인 방법 중 “미리 읽기”(prefetching) 이 있는데 이 방법을 적용하였을 때 얼마나 런칭시간을 줄일 수 있는지를 실험을 통하여 알아보고, 실험결과가 가지는 의미에 대해서 논의하였다.

## II. 관련 연구

### 1. 모바일 디바이스에서의 런칭시간 측정<sup>[1]</sup>

이 연구에서는 호스트 컴퓨터와 안드로이드 디바이스가 연결된 환경에서 런칭시간을 측정하는 도구를 개발하였다. 언급했듯이 이 도구는 호스트컴퓨터가 필요하기 때문에 런칭시간 최적화에는 사용될 수 없다.

## III. 본 론

## 1. 도구의 구현

이 도구의 구조는 그림 1. 과 같다. 이 도구는 안드로이드 내의 프레임워크(Framework) 계층에 존재하며 Information Collector, Launching Time Calculator 두 개의 모듈로 구성되어 있다. Information Collector 는 프레임워크 내의 다른 모듈들과 통신하여 언제 사용자가 응용 프로그램을 실행하고, 터치하고, 종료하는 지 등의 대한 정보를 수집한다. 또한 이 도구는 런칭된 응용 프로그램이 모두 화면에 그려졌을 때를 런칭이 끝나는 시간으로 기준을 삼기 때문에, UI(User Interface) 가 그려지는 정보도 수집한다. 수집한 정보를 바탕으로 Launching Time Calculator 가 런칭이 되고 있는 응용 프로그램에 해당되는 정보만을 필터링하여 런칭시간을 계산한다.

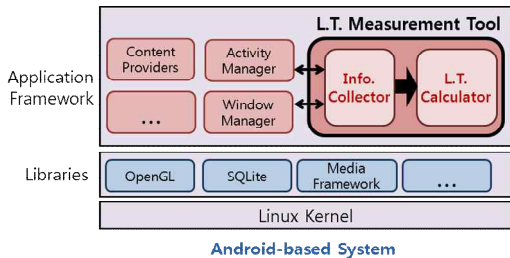


그림 1. 실시간 런칭시간 측정도구의 구조  
Fig 1. An overview of the tool

이 도구가 제대로 런칭시간을 재는지를 확인해 본 결과, 사용자가 응용 프로그램이 완전히 런칭되기 전에 응용 프로그램을 사용하는 경우를 제외하면 런칭시간을 정확히 측정하는 것을 확인하였다.

## 2. 모바일에서의 런칭시간 최적화 방향 탐색

모바일 환경에서 어떤 기법을 사용하는 것이 런칭시간을 줄이는데 효과가 있는지 알아보기 위한 실험을 수행하였다.

### 2.1 배경지식

실험에 앞서 응용 프로그램의 런칭을 분류하는 기준을 설명하고자 한다. 일반적으로 런칭을 cold start, warm start 이렇게 2가지로 분류한다. cold start 는 응용 프로그램이 런칭될 때 필요한 모든 페이지가 메모리에 없는 경우에 런칭이 되는 상황을 의미하며, 이 때가 가장 런칭시간이 긴 경우가 된다. warm start 는 필요한 모든 페이지가 메모리에 있는 경우 런칭이 되는 상황을 의미한다. “미리 읽기”를 하면 cold start 가 될 수 있는 상황을 warm start 로 바꿀 수 있기 때문에 런칭시간을 감소시킬 수 있다.

### 2.2 실험 목적 및 결과

실험은 “미리 읽기”의 효과를 알아보는 실험으로 cold start 와 warm start 일 때의 런칭시간을 구하여 얼마나 차이가 있는지를 알아보았다. cold start

와 warm start 의 차이가 클수록 “미리 읽기”의 효과가 큰 것이다.

20개의 응용 프로그램에 대하여 cold start 와 warm start를 각각 측정하여 평균을 낸 결과를 그림 2. 에서 보여준다.

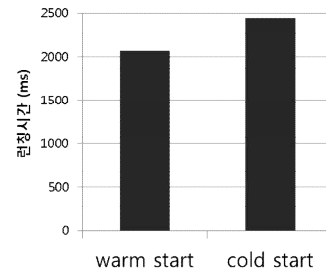


그림 2. cold start 와 warm start 의 런칭시간  
Fig 2. The launching time of cold start and warm start

## 2.3 실험 결과 분석

그림 2. 를 보면 cold start 와 warm start 의 차이가 20% 미만인 것을 볼 수 있다. 이렇게 차이가 많이 나지 않는 이유를 2가지로 볼 수 있다. 첫째, 메모리에 필요한 페이지를 로드하는 시간보다 다른 작업이 상대적으로 훨씬 오래 걸린다. 둘째, 완벽한 cold start 를 만들 수가 없다. cold start 를 재현하기 위해 메모리를 비우는 리눅스 명령어를 이용한다. 안드로이드 시스템 특성상 프로그램 간 공유하는 라이브러리가 많은데, 공유 페이지는 이 명령어에 의해 쫓겨나지 않기 때문에 완벽한 cold start 를 재현할 수 없는 것이다. 하지만 이 실험에서 측정한 cold start 런칭시간이 실제 시스템에서의 최악수행시간이라고 볼 수 있다. 왜냐하면 공유 페이지의 경우, 웬만해서는 실제 사용 중에 쫓겨나지 않기 때문이다.

## IV. 결론

본 논문에서는 안드로이드 응용 프로그램의 런칭시간을 실시간으로 측정하는 도구를 개발하였고, 이 도구를 이용한 실험을 통해 “미리 읽기” 기법이 안드로이드 시스템에서는 효과가 작은 것을 볼 수 있었다.

## 참고문헌

- [1] Jehun Lim, Hakbong Kim, Wook Song, and Jihong Kim, "LTmeter: An App Launching Time Analyzer for Personal Smart Devices," In *Proceeding of the 6th International Conference on Ubiquitous Information Technologies & Applications*, 2011