

모바일 응용의 사용자 중심 반응 시간 분석 도구

성노섭, 송욱^o, 김지홍

서울대학교 컴퓨터공학부

nssung@davinci.snu.ac.kr, wooksong@davinci.snu.ac.kr, jihong@davinci.snu.ac.kr

A User-Centric Response Time Analysis Tool for Mobile Applications

Nosub Sung, Wook Song^o, Jihong Kim

Department of Computer Science and Engineering

Seoul National University

요 약

스마트폰을 대상으로 하는 모바일 응용은 사용자와의 빈번한 상호 작용에 의하여 작업이 진행되므로 이러한 과정에서 사용자가 느끼는 경험이 매우 중요하다. 특히 사용자 경험은 사용자의 입력에 대한 응용의 반응 시간에 의해 결정되므로 이러한 사용자 중심의 반응 시간에 대한 분석이 모바일 응용의 성능 최적화에 매우 중요한 요소이다. 한편, 현재 모바일 응용의 개발에 사용하는 성능 분석 도구들은 사용자가 느끼는 반응 시간이 아닌 일반적인 PC 환경에서 동작하는 응용에 성능 평가 기준으로 널리 사용하고 있는 작업 수행 시간에 기반하고 있기 때문에, 모바일 응용에서 실제 사용자가 느끼는 성능을 최적화하여 사용자 경험 향상을 이끌어 내기에 적합하지 않다. 본 논문에서는 이러한 기존의 성능 분석 도구의 문제점을 개선하여 사용자 중심의 반응 시간을 측정하고 분석하여 모바일 응용 성능 최적화에 활용할 수 있는 도구를 제안하고 구현하였다. 또한 개발한 도구가 평균적으로 93%의 정확도를 보임과 동시에 계산 부하는 전체의 1% 미만임을 확인함으로써 그 설계 및 동작의 우수성을 증명하였다.

1. 서 론

스마트폰은 사용자와의 빈번한 상호 작용을 통해 동작하는 모바일 장치로서 이러한 각각의 상호 작용에 대한 사용자 경험이 사용자가 실제 느끼는 성능에 가장 큰 영향을 주는 요소이다. 즉, 사용자의 연속된 입력에 대한 모바일 응용의 반응 시간에 의해 사용자 경험이 결정되므로, 모바일 응용에서 사용자가 실제로 느끼는 성능의 향상을 위해서는 이러한 반응 시간에 대한 이해와 분석이 모바일 응용의 설계 및 구현에 매우 중요한 요소이다.

한편, 전통적인 PC 환경에서의 반응 시간은 일반적으로 해당 작업의 시작부터 끝까지의 시간으로 정의된다. 하지만 모바일 장치에서의 사용자는 PC 환경에서보다 훨씬 주관적으로 작업의 완료를 결정하게 되므로 화면에 표시되는 UI(User Interface)의 갱신이 완료되는 시점에 해당 작업이 끝났다고 판단하는 경향이 강하다. 좀 더 시스템적인 측면에서 접근하면, 화면에 표시되는 UI의 갱신에 필요한 작업들이 끝나면 이러한 작업들 이외의 다른 백그라운드 작업이 현재 수행 중이라고 하더라도 사용자는 자신의 입력에 대한 반응이 완료되었다고 판단한다는 것이다. 이러한 이유로 전통적인 PC 환경에서의 계산 중심 반응 시간은 모바일 장치에서 사용자가 느끼는 성능을 표현하기에 적합한 평가 기준이 아니며, 모바일 장치에서의 사용자 경험을 제대로 평가하기 위해서는 사용자 중심의 새로운 반응 시간에 대한 정의가 필요하다.

본 논문에서는 모바일 응용의 성능을 모바일 환경에 맞게 평가할 수 있는 사용자 중심의 반응 시간을 제안하고 기존의 계산 중심 반응 시간과의 차이를 먼저 밝힌다. 또한, 이러한 제안을 바탕으로 모바일 응용의 사용자 중심 반응 시간을 동적으로 탐지할 수 있는 기법과 도구를 소개한다.

본 논문의 구성은 다음과 같다. 2장에서는 사용자 중심의 반응 시간이 필요함에 대해 밝히고, 3장에서는 제안한 사용자 중심 반응 시간 분석 도구의 구조와 동작 원리에 대하여 설명한다. 4장에서는 제안한 도구의 정확도와 계산 부하를 평가한 후, 5장에서 결론을 맺는다.

2. 사용자 중심 반응 시간

본 장에서는 먼저 앞서 언급하였던 계산 중심의 반응 시간과 사용자 중심의 반응 시간을 모바일 환경에 맞추어 명확하게 정의하였다. 또한, 실제 응용의 동작에 따른 반응 시간을 각각의 정의를 이용하여 측정 후, 이를 비교함으로써 모바일 환경에서는 사용자 중심의 반응 시간을 모바일 응용을 위한 성능 평가 기준으로 활용하는 것이 필요함을 밝혔다.

모바일 응용과 사용자 사이의 상호 작용은 사용자의 화면 터치, 키 또는 음성 등과 같은 입력과 이에 대한 모바일 응용의 반응으로 볼 수 있다. 이러한 배경을 바탕으로 계산 중심의 반응 시간을 정의하면 새로운 사용자의 입력이 인가되는 시점부터 해당 입력으로 인해 수행되는 새로운

메소드의 호출이 더 이상 일어나지 않는 시점까지로 생각할 수 있다. 본 논문에서는 이러한 정의를 따르는 계산 중심의 반응 시간을 측정하기 위해 안드로이드의 가상 머신(Dalvik) 번역기를 수정하여 모든 메소드의 호출을 감지할 수 있도록 수정하였다. 사용자 중심의 반응 시간은 여러 모바일 응용 개발 가이드[1]에서 사용자가 실제로 느끼는 성능과 가장 유사할 것으로 추정하여 권고하는 기준으로 사용자의 입력이 인가되는 시점부터 이로 인해 발생하는 UI의 갱신이 더 이상 이루어 지지 않는 시점까지로 정의한다[2]. 또한, 일반적인 응용의 사용자 중심 반응 시간을 측정하기 위하여 스마트폰의 프레임 버퍼를 동적으로 샘플링하고 이를 이전 프레임과 계속하여 비교함으로써 더 이상 화면의 변화가 발생하지 않는 시점을 감지할 수 있도록 하였다.

만약 실제 모바일 응용에 대한 계산 중심의 반응 시간과 사용자 중심의 반응 시간을 측정하였을 때, 두 반응 시간의 차이가 거의 존재하지 않거나 큰 차이를 보이지 않는다면 기존의 모바일 응용 대상 성능 평가 및 분석 도구로도 충분하다는 주장이 가능하다. 연구의 동기로서 먼저 계산 중심의 반응 시간과 사용자 중심의 반응 시간에 어떠한 정량적 차이가 존재하는지 확인하기 위해 앞서 설명한 측정 방법으로 안드로이드 4.0.4를 탑재한 Nexus S 스마트폰에서 Facebook, Web Browser, Email, Public Transportation 응용에 대한 두 가지 정의의 반응 시간을 측정하였다.

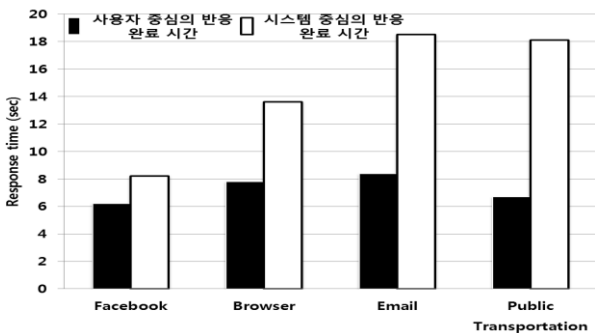


그림 1. 각 정의에 따른 여러 응용의 반응 시간

그림 1은 기 언급한 두 가지 관점에 따른 응용들의 반응 시간을 보여준다. 예상한 것과 같이 계산 중심의 반응 시간은 UI의 갱신이 모두 끝난 후에도 수 많은 새로운 함수를 호출하게 됨에 따라 모든 경우에서 사용자 중심의 반응 시간보다 길게 측정되었으며 최대 11.4 초의 차이가 존재함을 알 수 있었다. 네 가지 응용 중 유독 그 차이가 크게 나타났던 Web Browser와 Email 응용을 분석한 결과, UI의 갱신이 완료된 이후에도 웹 캐시 및 방문한 웹 사이트 목록 갱신, 첨부 파일의 백그라운드 다운 등으로 인해 계속하여 작업이 이루어졌음을 확인할 수 있었다.

이러한 결과를 통해 우리는 다음과 같은 두 가지 시사점을 확인할 수 있다. 첫째, 계산 중심의 반응 시간으로는 사용자가 실제 느끼는 성능을 평가할 수 없다. 둘째, 기존의 성능 분석 도구로 계산 중심의 반응 시간을 최적화하였다 하더라도 실제

사용자가 느끼는 성능이라는 관점에서 사용자 경험의 향상으로 이어지지 않을 수 있다. 따라서, 사용자 중심의 반응 시간의 측정과 이를 바탕으로 한 성능 분석이 가능한 도구가 필요하다.

3. 사용자 중심 반응 시간 분석 도구

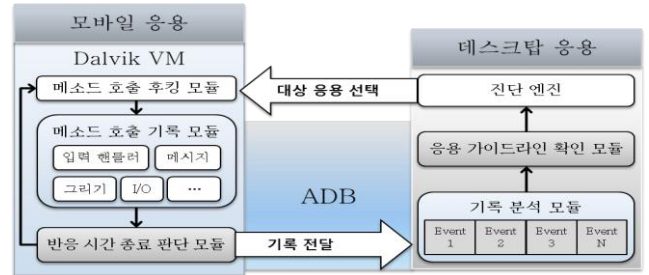


그림 2. 사용자 중심 반응 시간 분석 도구의 구조

그림 2는 본 논문에서 제안한 사용자 중심 반응 시간 분석 도구의 구조를 나타낸 도식도이다. 사용자의 입력, 사용자의 입력으로 인해 생성되는 쓰레드, UI의 갱신들을 동적으로 감지하기 위해 안드로이드의 가상 머신을 수정하여 관련 메소드들의 호출과 함께 전달되는 인자를 추적하여 이를 메소드 호출 기록 모듈을 통하여 기록하도록 하였다. 각 메소드의 호출 기록을 분석하여 새로운 사용자의 입력이 인가되었고 이와 관련되어 있는 UI의 갱신의 완료 시점을 결정하는 것은 반응 시간 종료 판단 모듈이 담당한다. 또한, 이러한 정보는 ADB를 통하여 호스트로 전달되며, 기록 분석 모듈과 응용 개발 가이드라인 확인 모듈을 거쳐 개발자에게 유용한 정보의 형태로 가공된다.

사용자의 입력과 이로 인한 UI의 갱신까지는 다양한 시나리오가 존재하며 정확한 사용자 중심의 반응 시간 측정 및 분석을 위해서는 이런 다양한 시나리오와 무관하게 사용자 입력으로부터 발생한 UI의 갱신을 추적할 수 있어야 한다. 특히 안드로이드에서는 UI의 갱신을 담당하는 단 하나의 쓰레드가 존재함과 동시에 작업 쓰레드에서도 화면 갱신을 요청할 수 있는 다양한 방법을 제공하고 있기 때문에, 하나의 메인 쓰레드와 다수의 작업 쓰레드 간의 관계 파악도 매우 중요하다.

그림 3은 안드로이드 응용에서 발생 가능한 화면 갱신 시나리오들을 나타낸 것으로 invalidate와 update는 각각 UI 갱신 요청과 처리를 나타낸다. 가장 기본적인 시나리오의 형태는 (a)이며 이는 메인 쓰레드에서 UI 갱신 요청과 처리가 모두 일어나는 상황을 나타낸다. 이러한 경우에는 각 UI 갱신 요청마다 View ID와 같은 식별자를 저장하고 해당 식별자에 대한 갱신 요청이 처리되는 시점에 저장한 모든 식별자가 처리되었는지를 확인함으로써 반응 종료를 확인할 수 있다. 하지만, 일반적인 응용은 이러한 시나리오보다 훨씬 복잡한 동작을 거쳐 UI 갱신이 이루어진다. (b)의 경우는 무거운 작업을 다른 비동기식 작업 쓰레드에 위임하여 처리하며 이에 대한 결과의 UI에 대한 반영만을 메인 쓰레드에 요청할 수도

있다. 또한, (c)와 같이 데몬과 유사한 작업 쓰레드가 반복적인 UI 갱신을 요청하는 경우도 가능하여 이러한 경우에는 심지어 모든 UI 갱신이 사용자로부터 나온 것이 아닐 수도 있다.

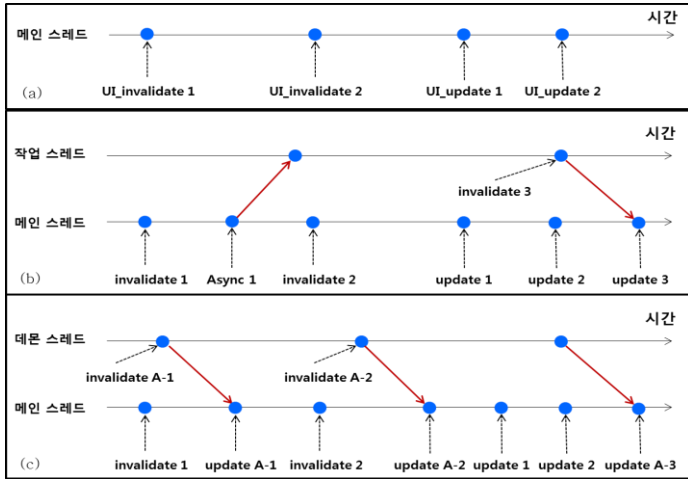


그림 3. 다양한 화면 갱신 시나리오의 예제

이러한 문제를 해결하기 위하여 그림 4와 같이 사용자 입력을 처리하는 Callback 메소드가 수행되는 동안 메인 쓰레드와 작업 쓰레드가 주고 받은 메시지를 추적할 수 있도록 하였다. 안드로이드의 경우에는 이러한 내부적인 메시지의 교환이 sendMessage/dispatchMessage 메소드로 이루어지므로 메소드의 호출과 함께 전달되는 인자를 분석함으로써 동적으로 가능하였다.

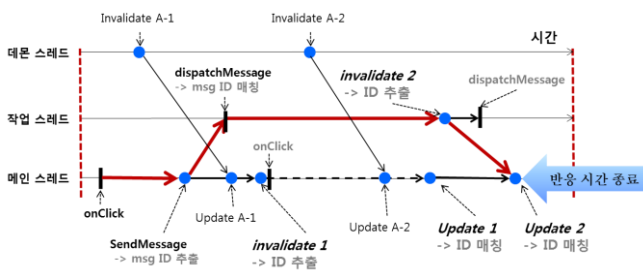


그림 4. 사용자 중심 반응 시간 판별의 예

4. 실험 결과

제안한 도구의 효용성을 판단하기 위해 본 논문에서는 안드로이드 4.0.4가 탑재된 갤럭시 넥서스 스마트폰에서 여러 유명 응용들에 대한 정확도와 계산 부하를 평가하였다. 대상 응용의 선정은 마켓에서 백만 건 이상 다운로드 된 응용들 중에서 선정한 것이며 각 응용의 사용 시나리오는 런칭과 주요 작업 중 하나를 선정하였다. 정확도의 평가는 제안한 도구를 사용하여 측정한 사용자 중심의 반응 시간과 카메라를 통해 응용의 동작을 촬영한 영상을 프레임 별로 분석하여 얻은 사용자 중심의 반응 시간을 비교하였다.

그림 5는 제안한 도구의 정확도 평가의 결과를 나타낸 그래프이다. 실험 결과 약 93% 정도의 정확도를 보였으며,

오차는 평균적으로 약 100 ms 정도로 매우 정확함을 확인할 수 있었다. 계산 부하를 평가하기 위해 추가한 모듈들이 반응 시간에 미치는 비중 또한 평가해보았는데, 전체 반응 시간에서 약 1% 미만의 계산 부하만을 나타낸 것을 확인하여 응용의 수행에 큰 문제 없음 또한 확인할 수 있었다.

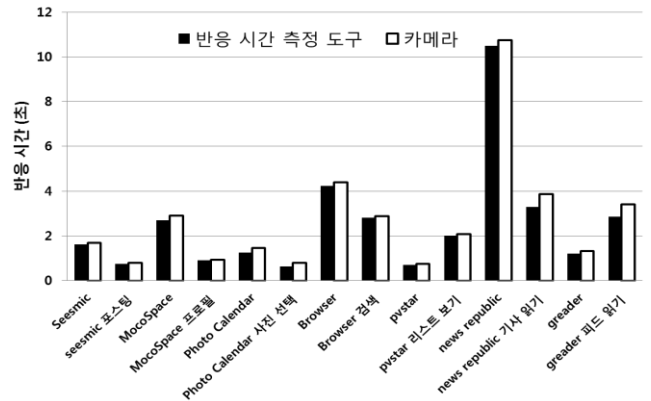


그림 5. 제안한 도구의 정확도 평가

5. 결론

본 논문에서는 모바일 응용에서 사용자 중심의 반응 시간을 성능 평가 기준으로 활용하는 것이 필요함을 보이고 기존의 모바일 응용 대상 성능 평가 도구들이 이러한 사용자 중심의 반응 시간을 고려하고 있지 않음을 인지하여 사용자 중심 반응 시간 측정 및 분석 도구를 제안하였다. 개발한 도구는 정확도가 높은 반면 그 계산 부하가 낮음을 보임으로써 개발자들이 실제 모바일 응용을 개발하는데 사용하기에 부족함이 없음을 증명하였다. 하지만, 가상 머신 수준에서 메소드의 호출을 추적하는 방식으로 반응 시간을 추적하기 때문에 네이티브 메소드로 구성된 응용에 대해서는 태생적 한계가 존재한다. 향후, 이러한 한계를 보완하여 도구의 활용성을 높일 계획이다.

감사의 글

이 연구를 위해 연구장비를 지원하고 공간을 제공한 서울대학교 컴퓨터 연구소에 감사 드립니다. 이 논문은 2014년도 정부(미래창조과학부)의 재원으로 한국연구재단-차세대정보·컴퓨팅기술개발사업의 지원을 받아 수행된 연구임 (No. 2010-0020724). 또한, 이 논문은 2014년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(NRF-2013R1A2A2A01068260).

참고 문헌

[1] Apple Inc., Launch Time Performance Guidelines, <http://developer.apple.com/documentation/Performance/Conceptual/LaunchTime/LaunchTime.pdf>
 [2] L. Ravindranath, et al., "AppInsight: Mobile App Performance Monitoring in the Wild," in *Proc. International Conference on Operating System Design and Implementation*, 2012