

<http://dx.doi.org/10.7236/JIIBC.2013.13.2.115>

JIIBC 2013-2-15

어플리케이션 가상화를 적용한 안드로이드 런처 연구

A Study of Android Launcher based on Application Virtualization

임승철*

Seung-Cheol Lim

요약 최근 가상화 기술은 데스크탑 환경에서 모바일 플랫폼 환경으로 그 자리를 옮겨 개인과 기업, 그리고 학계 등 안전하지 않은 보안 환경과, 여러 사람들의 공유하는 팀 환경에서 많은 변화를 가져오고 있다. 이 환경에서 대표적인 것은 Vmware의 MVP솔루션들과 Enterproid의 Divide이다. 하지만 이러한 모바일 가상화 기술환경에서 또다른 이슈가 되고 있는 것은 한정된 자원과 성능상의 제약들인데 그것을 어플리케이션 가상화 기술로 모바일 플랫폼의 약점을 개선하려는 방안으로서 대두되고 있다. 본 논문에서는 기존에 제안되었던 로컬 기반 데스크탑용 어플리케이션 가상화 구조를 안드로이드 환경으로 가져와 데스크탑 환경 어플리케이션 가상화의 한계점인 사용자 영역구분, 프로그램 실행시 생기던 할당 자원증가를 안드로이드 어플리케이션 가상화 런처를 통해 개선하였다.

Abstract Recently, Virtualization technology is moving the desktop environment from its place a mobile platform environment, personal and businesses, academia, and lacked security and in a team environment brought about many changes are coming. In this environment, a representative of VMware's MVP solution and Enterproid Divide. But, another issue is that these mobile virtualization technologies in an environment of limited resources and performance constraints, have become its application virtualization technology to improve the weaknesses of the mobile platform. In this paper, proposed for client-based desktop application virtualization infrastructure, bring it to the Android environment, weaknesses of the existing paper was user area limitations of the desktop environment with application virtualization, program execution android application virtualization launcher was improved through increased resource allocation.

Key Words : Mobile virtualization, Application, Service management, Client virtualization, Security

1. 서론

최근의 가상화 기술 환경은 서버, 데스크탑에서 모바일로 그 자리를 옮겨오고 있는 추세다. iOS와안드로이드로 대표되는 스마트 폰의 작업영역은 데스크탑과 거의 다르지 않을 정도로 많은 발전을 이루었으며 2012년

을 기준으로 안드로이드의 점유율은 60%를 넘고 있다.^[1] 이러한 모바일시장에서 가상화 환경은 현재 개인 보다는 기업과 연구환경에서 많은 각광을 받고 있다. 데스크탑에서처럼 여러 가지 OS기반 하에 개발환경, 다중서버를 위한 환경, 개인과 작업영역 분리 등을 위한 보안환경 등 기존 데스크탑 환경에서의 가상화와는 크게

*정회원, 우송대학교 컴퓨터정보학과
접수일자 : 2013년 1월 23일, 수정완료 : 2013년 2월 25일
게재확정일자 : 2013년 4월 12일

Received: 23 Jan 2013 / Revised: 25 February 2013 /
Accepted: 12 April 2013

*Corresponding Author: sclim@wsu.ac.kr
Dept. of Computer Information Science, Woosong University,
Korea

차이가 나지 않을 정도로 그 쓰임새가 유사하다. 이 모바일 플랫폼 가상화솔루션으로 대표적으로 알려진 것은 Vmware와 Enterproid의 Divide가 있다. 하지만 모바일기반 가상화 환경에서 한정적인 자원 배분과 성능의 한계점이 대표적인 단점이다. 이를 개선하기 위해 어플리케이션 가상화가 가장 대표적이며^[2] 이 가상화 방식은 한정적인 자원과 성능을 유용하게 이용할 수 있고, 한 화면당 한 어플을 보여주는 현 iOS와 안드로이드에 어울리는 측면도 있다.

따라서 본 논문에서는 먼저 제안되었던 로컬 데스크탑 기반의 어플리케이션 가상화 방식을 모바일 환경에서 구현 및 개선을 하고 기존 방식의 문제점이었던 로컬 기반의 독립된 가상화 환경 구성시 계층구성의 한계점과 이로 인한 보안성의 하락과 제안한 방식의 테스트에서 보였던 자원 점유율 문제를 개선한 안드로이드 기반의 어플리케이션 가상화 런처방법을 제안하고자 한다.

본 논문의 2장에서는 기존에 제안된 방법의 한계점을 세가지 측면에서 분석하고, 3장에서는 어플리케이션 가상화 런처를 서술하고 4장에서는 3장을 구현한 프로토타입과 개선결과, 결론을 통해 마무리를 맺는다.

II. 기존에 제안된 방법

1. 로컬 데스크탑의 환경상 작업 환경 분리의 한계점

제안된 논문의 방식에 따르면, 작업환경 구분이 그림 1과 같은 구조를 가지게 된다.^{[3][4]} 가상화된 사용자 영역을 만들어 내고 있으나, 기존의 유지 환경 위에 구현이 되어져 있고, 가상화 상에서 실행되는 어플이 로컬 환경에서 가져와 등록되고 동작하는 부분이 문제가 되는데, 어플리케이션 가상화가 반가상화적인 측면을 지니고는 있으나, 일반 사용자 영역과 가상화 사용자 영역을 나누었고 이를 통한다면 어플리케이션 설치 환경파일이 나뉘어져야하나 구조적으로는 기존 환경에서 벗어날 수 없다는 문제점이 있다.

또한 테스트되는 PC의 OS환경이 XP라는 점인데, 리눅스의 tty와는 달리 XP의 사용자 계정은 한 화면에 종속적이라는 특징이 있어 단순히 자원적인 측면에서만 분할이 되어 있어, 완전한 작업 환경 분할의 측면에서는 벗어나있고, 이로 인해 로컬 환경에서의 어플리케이션

가상화 구현 테스트를 윈도우로 정한 의미가 다소 퇴색된다. 따라서 윈도우 로컬 데스크탑의 경우 Vmware와 같은 전가상화나, 서버-클라이언트 구조를 사용한 공간적 시각적인 가상화 방식을 구현하였으면 적합할 것으로 생각된다.^[5]

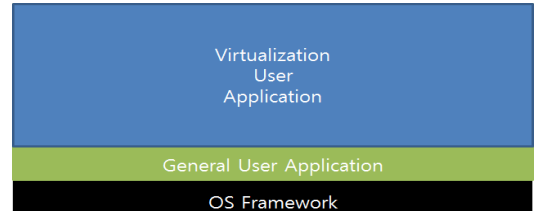


그림 1. 제안된 논문의 작업환경 구분
Fig. 1. Proposed plan of application virtualization launcher

2. 보안적인 측면의 한계점

기존 논문에서 언급한 보안적인 측면에서도 가상화된 영역에 어플리케이션을 올려도 결국은 별도의 공간에 어플리케이션을 설치한 것이 아닐뿐더러, 프로그램이 작동되는 지점이 일반환경에 종속되어져 있다.

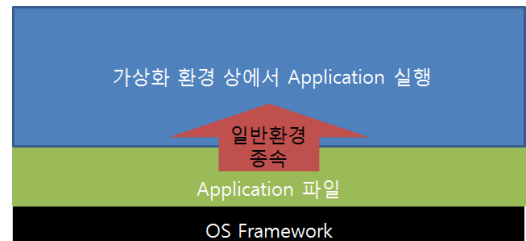


그림 2. 제안된 논문의 작업환경 종속관계
Fig. 2. Proposed paper of work environment dependency

그림 2에서와 같이 일반 환경에서의 어플리케이션이 작업환경을 불러올 수 있는 구조를 가진다면 가상화 환경에서 했던 작업이 일반 환경에 남아 이를 그대로 불러올 수 있는 구조를 갖는다. 이는 별도의 어플리케이션의 각 영역에 설치되지 않는 가상화 환경에서 생길수 있는 문제점이며, 이로 인해 보안적인 측면에서 불안정적이다.

3. 자원적인 측면의 한계점

자원적인 측면에서는 가상화 환경에서 실행되는 어

플리케이션의 자원 점유율의 문제가 있다. 기존의 제안된 논문의 테스트 결과에서 보인 메모리 사용량은 표 1과 같다.

표 1. 기존 제안된 방식의 테스트 결과
Table 1. Test results of proposed paper

General Application	메모리 사용	Virtualizaion Application	메모리 사용
메모장	4,448KB	메모장	8,552KB
iexplorer	60,352KB	iexplorer	68,544KB

메모장의 경우 텍스트파일을 가상화 환경에서 실행했을 경우 약 2배에 달하는 메모리를 점유하고 있고, iexplorer의 경우 8192 Kilobyte 정도의 메모리를 더 점유하게 된다. 가상화 환경에서 실행되는 어플리케이션을 감안함에도 다소 높은 자원 점유율을 보이고 있으며, 가상화 환경 상에서 메모리 점유율이 1기가가 넘어가는 그래픽툴, 개발툴 등을 실행했을 경우 더 큰 메모리 점유율을 가질 가능성도 있을 것으로 보인다.

III. 제안한 어플리케이션 가상화 런처 방식

따라서 본 논문에서는 사용자 영역의 분리 측면, 보안적 측면, 자원적 측면 세가지 측면을 개선하고자한다.

먼저 사용자 영역적인 측면에서는 하나의 OS, 하나의 사용자를 갖게 되는 모바일OS 즉, 안드로이드 상에서 어플리케이션 가상화 런처를 구현하여 일반 영역과 가상화 영역을 시각, 공간적으로 나누는 구조를 가져 이를 개선하고자 한다.^[6]

1. 사용자 작업환경의 영역 분리 개선점

공간적, 시각적인 작업환경분리는 가상화 환경에서 볼 수 있는 가장 큰 특징이다. 그림 3에서 먼저 안드로이드의 특성상 OS에 런처가 구동되어 유저가 어플리케이션을 실행시킬 수 있는 UI환경을 갖추게 되며, 이는 가상화 환경 또한 같은 기능을 가지게 된다. 먼저 일반 모드 상에서는 어플리케이션 가상화가 되어있는 런처를 앱으로 등록하여 실행할 경우 가상화 런처 환경에 접근하게 되며, 두 런처는 서로 개별적인 앱을 등록하게 된다.

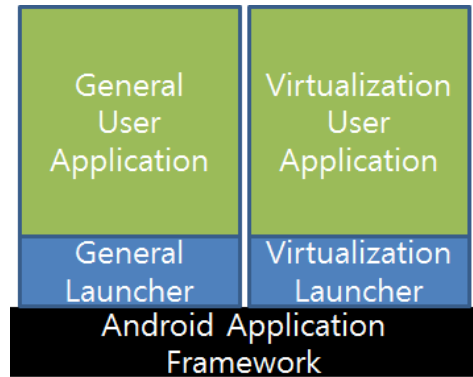


그림 3. 어플리케이션 가상화 런처 설계
Fig. 3. Plan of application virtualization launcher

따라서 서로간의 앱이 같을 수도 다를 수도 있으며, 만일 같은 앱이라 해도 공간적, 시각적으로 분리되어 있다. 따라서 기존의 방식에 비해 로컬 환경에서 사용자 영역이 일반 런치모드와 가상화 런치모드로 좀 더 확고히 구분되는 특성을 가질 수 있게 된다.^{[7][8]}

2. 보안적 측면의 개선점

그림 4에서는 보안적인 측면에서 보자면 별도의 영역을 갖고 있기 때문에 개인환경과 보안환경 두가지 측면에서 개선점을 보일 수 있다.

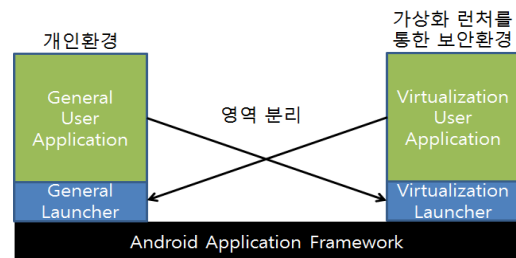


그림 4. 가상화 런처를 통한 보안환경과 개인환경의 분리
Fig. 4. Separation of the personal environment and security environment through virtualization launcher

시각적으로는 한 화면에 하나의 런처만이 기동되어 보여지기 때문에, 다른 런처가 기동중일 경우 서로 간에 작업환경을 알 수 없고 별도의 앱을 가지고 있기 때문이다. 공간적인 측면에서는 각 런처 당 별개의 앱 저장과 별개의 작업환경을 갖기 때문에 각 영역의 침범이 이루어 질 수 없다.

따라서 두 런처는 완전히 개별적이며, 별도의 관리를 할 수 있게 되어 보안적인 측면이 기존의 방식보다 강화되어 있다.

3. 자원적 측면의 개선점

자원적인 측면에서 그림 5의 안드로이드 런처와 가상화 런처의 흐름 특성상 같은 각각의 런처는 고유의 어플리케이션을 설치할 수 있고, 서로 다른 환경의 런처 상에서 실행시킬 수 있다.

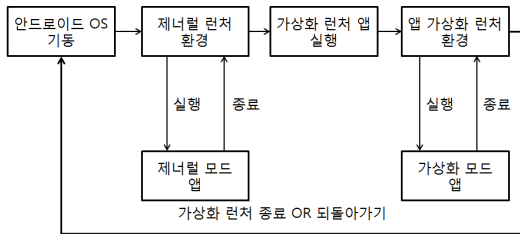


그림 5. 안드로이드의 런처 및 가상화 런처의 흐름도
Fig. 5. Flowchart of the android launcher and virtualization launcher

안드로이드의 OS관점에서 보면 한 런처 위에 한 앱 이 기동되기 때문에 메모리 점유율에 별다른 차이가 나 올 수 없다. 따라서 자원적인 측면에서 기존 방식보다 개선될 것으로 보인다.^[9]

IV. 구현된 프로토타입

개발환경은 안드로이드2.3.3을 기준으로 eclipse와 안드로이드 SDK를 통해 데스크탑 환경에서 제작하였으며, 가상화 런처는 안드로이드의 베이직 런처의 기능을 활용해 별다른 기능을 추가하지는 않았다. 가상화 런처의 경우 카테고리라 스택의 분리를 통해 일반 모드와 완전히 영역을 분리하여 앱 설치영역을 나누어 서로간의 공유할 수 있는 영역을 분리시켜 서로 다른 환경에서 폰을 사용하는 것처럼 가상화 환경을 구현하였다.

그림 6은 안드로이드 일반 모드 런처 화면과 가상화 모드 런처의 화면 비교로 일반모드에는 어플리케이션 가상화 런처가 설치되어 있고, 안드로이드에 기본적으로 주어지는 어플리케이션들이 설치되어있다. 이 어플리케이션들은 가상화 런처에서는 볼수 없으며, 실행 시킬수 없다. 실행 방법은 일반모드런처상의 Vm런처 아

이콘을 클릭하거나, 메뉴화면상에서 우송런처 아이콘을 이용하여 가상화 환경으로 새로운 런처를 띄우도록 처리하였다.

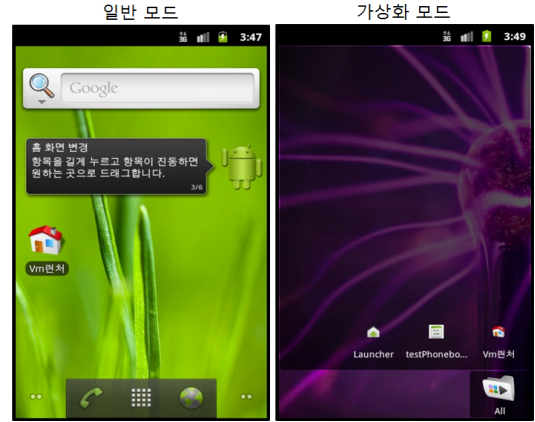


그림 6. 일반 모드 런처와 가상화 모드 런처 비교
Fig. 6. Comparison of general mode launcher and virtualization mode launcher

그림 6의 오른쪽 가상화 모드 런처는 어플리케이션 가상화를 통해 구현된 가상화 런처로 그림 6의 왼쪽 일반모드 런처와는 다르게 전혀 다른 구성으로 런처가 작동중이며, 두 런처 간의 설치되어있는 어플은 서로 다른 영역에서 존재하고 있다.

따라서 사용자 작업 환경이 분리되어있으며, 서로 다른 어플리케이션 관리 등을 통해 개인환경과의 차이점을 두어 협업, 기업용 모드 등에 있어 차이점을 주어 시각적, 공간적인 보안성을 나타내고 있다.

일반 모드		가상화 모드	
testPhonebook 버전 1.0	0.00 B	testPhonebook 버전 1.0	0.00 B
저장공간	0.00 B	저장공간	0.00 B
전체	684 KB	전체	540 KB
애플리케이션	676 KB	애플리케이션	528 KB
데이터	8.00 KB	데이터	12.00 KB
데이터 지우기	SD 카드로 이동	데이터 지우기	휴대전화로 이동
캐시	0.00 B	캐시	0.00 B
캐시 지우기	캐시 지우기	캐시 지우기	캐시 지우기
기본으로 실행	기본값이 설정되지 않았습니다.	기본으로 실행	기본값이 설정되지 않았습니다.
기본 작업 지우기	기본 작업 지우기	기본 작업 지우기	기본 작업 지우기

그림 7. 일반환경과 가상화환경의 어플리케이션 자원비교
Fig. 7. Resource comparison of general and virtual environments applications

그림 7은 일반환경에서 설치된 테스트용 프로토타입 어플리케이션과 가상화 환경에서 설치된 테스트용 프로토타입 어플리케이션 간의 자원 비교를 나타내는 스크린 샷으로 이를 표 2로 정리하면 다음과 같다.

표 2. 제안한 방식의 프로토타입 어플리케이션 자원 비교
Table 2. Prototype application resource comparison of the proposed method

General Application	자원 사용량	Virtualizaion Application	자원 사용량
전체	684KB	전체	540KB
어플리케이션	676KB	어플리케이션	528KB
데이터	8KB	데이터	12KB

테스트용 프로토타입 어플리케이션은 간단한 주소록 프로그램으로 내부에는 DB를 이용해 주소록이 저장되어 있으며, 주소록 데이터는 가상화 환경에만 존재한다.

전체와 어플리케이션 자원 사용량이 차이가 나는 것은 액티비티가 런처의 UI를 따라가는 문제로 가상화 환경의 런처가 더 가벼운 기능을 탑재하기 때문에 전체와 어플리케이션의 용량이 일반환경보다는 작을 수 밖에 없다. 하지만 이 부분은 런처의 무게에 따른 문제로 같은 UI를 구현하였을 경우 그 자원 사용량이 비슷한 선에서 소모될 것으로 예상된다.

또한 데이터 사용량이 가상화 환경이 큰 것은 어플리케이션 내부에 데이터가 존재하기 때문이며, 일반 환경에선 아무런 내용이 존재하지 않아 그 용량이 작을 수 밖에 없다.

가상화시의 용량을 기준으로 기존에 제안한 방식에서는 13%의 자원 증가가 있었으나, 제안한 가상화 방식에서는 21.06%의 자원 감소효율을 가져왔으며, 이는 위에서 언급한 안드로이드의 런처 특성을 적용한 결과로 플랫폼이 다른 점도 염두에 두어야 할 것이다.

V. 결 론

본 논문에서 제안하는 어플리케이션 가상화 런처 방식과 적용플랫폼의 변경을 통해 기존에 제안되었던 방식에 비해 작업환경을 분리가 확실히 이루어져 같은 OS라 할지라도 전혀 다른 작업환경을 갖추도록 작업영역을 일반환경과 가상화 환경과의 분리를 하였고, 보

안적인 측면에서는 서로 다른 어플리케이션 설치와 서로 다른 어플리케이션 관리를 통해 어플리케이션의 작업 내용이 공유될 수 있는 문제점을 해결하였다. 자원면에서도 안드로이드의 런처 실행방식을 이용하여 비록 가상화 환경 상에서 실행되더라도 그 자원소모량을 최대한 억제하였으며, 작업영역의 분리가 이루어졌어도 내부에 쓰여진 자원소모량에는 큰 차이가 없는 결과를 나타내어 기존 방식에서 문제되었던 세 가지 문제점을 개선하였다.

최근 모바일 시장에서 안드로이드는 많은 발전을 하고 있으며, 구글에서는 사용자 계정의 추가 등을 통해 작업영역을 분리하는 보안 기능을 새 버전에 추가하는 등 가상화가 아니라도 작업영역의 분리와 보안성을 염두에 두는 업데이트를 이루고 있다.^{[10][11]} 이렇듯 가상화 기능은 매우 매력적이며, 하나의 OS에서 여러 작업환경을 만들어 내는 것은 개발환경, 연구 환경에 있어 물리적인 자원소모와 여러 테스트환경 구축과 그 보안에 있어 많은 도움이 되므로 많은 발전 가능성을 보이고 있다.

추후 연구내용은 가상화 앱을 다수 설치하여 다중 가상화에 대한 테스트와 그 자원사용량을 억제하는 부분이 필요할 것으로 보이며, 가상화 런처 진입 시에 비밀번호나 보안패드 등의 기능을 추가하여 그 보안성 강화가 필요하다.

참 고 문 헌

- [1] Sun-Hyang Jang., Su-ji Hwang, Young-Hyun Chang, Min-jeong Koo "Study on the Enhancement of the Credibility of Android OS based Applications", Journal of IWIT, Vol.11, No.3, pp.382~384, fall 2012.
- [2] Nelson Ruest, Danielle Ruest "Virtualization, A Beginner's Guide", McGraw-Hill 2009
- [3] Se-jung Lim, Gwang-jun Kim, Tae-geun Kang "Application Program Virtualization based on Desktop Virtualization", The Korea Institute of Electronic Communication Sciences, Vol.5, No.6, 2010
- [4] SILBERSCHATZ "Operating System Principles 7/E", Wiley 2005
- [5] VMware, Inc. "Understanding Full Virtualization,

- Paravirtualization, and Hardware Assist“, White Paper, Nov. 2007
- [6] Mark Murphy “Beginning Android3”, Apress 2012
- [7] Barret Rhoden, Kevin Klus, David Zhu, Eric Brewer “Improving Per-Node Efficiency in the Datacenter with New OS Abstractions“, ACM 2011
- [8] B Weinberg, L Pundit “Designing and Deploying with Mobile Virtualization“, Whitepaper. Linux Pundit 2009
- [9] Mohammad Nauman, Sohail Khan “Design and implementation of a fine-grained resource usage model for the android platform“, The International Arab Journal of Information Technology 2011
- [10] Google, Inc. “Android 4.2 API: Multiple Users”, <http://developer.android.com/about/versions/android-4.2.html> 2012
- [11] S.J. OH, “Design of a Middleware for Android-based Smart Phone Applications”, Journal of Korean Institute of Information Technology, vol. 12, issue 2, pp. 111-117, Apr 2012.

저자 소개

임 승 철(정회원)



- 1985년 : 한양대학교 전자공학과 학사
 - 1994년 : 전북대학교 정보통신과 석사
 - 2003년 : 전북대학교 영상공학과 박사
 - 2006년 ~ 현재 : 우송대학교 컴퓨터 정보학과 교수
- <주관심분야 : 이동통신, 컴퓨터네트워크, 임베디드시스템소프트웨어>