# 모바일 가상화 기술 동향

성균관대학교 ▌김정한 · 김인혁 · 민창우 · 엄영익*

## 1. 서 론

최근 모바일 장치 기술의 향상과 모바일 플랫폼의 발전으로 인해 사용자들은 이전에 데스크톱 환경에서만 가능하던 많은 작업들을 모바일 환경에서 수행할 수 있게 되었다. 또한 앱스토어(App Store) 등을 통해서 필요한 어플리케이션들을 자유롭게 다운로드가 받을 수 있게 됨으로써, 기존 모바일 폰에서의 통화 기능 외에 금융거래, 기업용 어플리케이션 등도 사용 가능하게 되었다.

이와 같은 변화는 기존의 실시간성을 우선시하던 RTOS(Real-Time OS) 기반 플랫폼에서 데스크톱 수준의 범용성 및 확장성을 갖는 GPOS(General Purpose OS) 기반 플랫폼으로의 개선을 통하여 이루어졌다. 그러나 이러한 모바일 플랫폼의 변화는 새로운 문제점을 야기하고 있다. 대표적으로 복잡해진 모바일 플랫폼으로 인한 실시간성 지원 문제, 기존 소프트웨어 재사용 문제, 금융 결제 및 검증되지 않은 프로그램 설치로 인한 보안 문제 등이 있으며, 모바일 플랫폼의 발전과 관련 시장의 급속한 성장으로 인하여 문제점은 더욱더 다양해질 것이다.

가상화 기술은 물리적 장치를 추상화하여 독립된 실행환경을 제공하는 기술로써 서버 가상화, 클라우드 컴퓨팅, 데스크톱 가상화, 임베디드 가상화에까지 폭넓게 적용되고 있으며 새로운 컴퓨팅 환경을 제공함과 동시에 정보 보호, 자원 관리 등 기존의 컴퓨팅 환경의 문제점을 해결 할 수 있는 기술로 주목받고 있다[1,2]. 이러한 가상화 기술의 보급을 통하여 최근 모바일 환경에도 기존 임베디드 가상화 기술 기반의 다양한 가상화 기술이 적용되고 있으며, 현재 나타나고 있는 다양한 모바일 환경의 요구사항을 충족시킬 것으로 전망하고 있다.

본 논문에서는 모바일 가상화 기술 동향에 대해서 살펴보고자 한다. 2장에서는 모바일 플랫폼 동향에 대해서 살펴보고, 3장에서는 서버 가상화, 데스크탑 가상화, 임베디드 가상화, 모바일 가상화 등의 활용 분야를 소개한다. 4장에서는 전가상화, 반가상화 등의 가상화 구현 기술에 대해서 살펴보고, 5장에서 모바일 가상화 기술 분석을 통하여 6장에서 결론을 맺는다.

## 2. 모바일 플랫폼 동향

애플사의 iPhone으로 촉발된 스마트폰의 보급은 모바일 플랫폼에 있어서 큰 전환점을 가져왔다. 기존의 피처폰(feature phone)에서는 Mobile Java(J2ME)나 WIPI 등을 이용하여 제한적인 어플리케이션의 추가 설치만이 가능하였다. 그러나 스마트폰에서는 모바일 플랫폼이 데스크탑 환경과 같이 다양한 기능을 제공하며, 개발자들이 SDK를 이용하여 개발한 어플리케이션을 앱스토어 등을 통하여 유통하는 방식으로 변화하였다. 이러한 변화에 따라 모바일 플랫폼은 다양한 종류의 어플리케이션 제작이 가능하도록 데스크톱 이상의 기능을 요구받고 있으며, 금융거래 및 기업용 어플리케이션 등이 안전하게 수행될 수 있도록 보안성 역시 요구받고 있다.

대표적인 모바일 플랫폼인 애플사의 iPhone OS는 MaxOS X를 기반으로 하여 멀티터치, 가속도센서, 근접센서 등을 추가하여 모바일 단말에 최적화한 것으로 어플리케이션 개발을 위한 SDK를 제공하고 있으며, 앱스토어를 통하여 어플리케이션의 자유로운 배포를 허용한다[3]. 앱스토어에는 오픈 후 2년 만에 20만개 이상의 어플리케이션이 등록되었으며 40억회 이상의 다운로드 수를 보이고 있다. 또한 iPhone은 국내 출시 5개월 만에 60만대를 돌파하며 빠르게 보급되고 있다.

안드로이드(Android)는 구글이 주도하는 OHA(Open Handset Alliance)에서 2007년에 발표한 플랫폼이다. 리눅스 기반의 커널을 사용하며, Java 환경을 통하여 크로스 플랫폼 어플리케이션 환경을 제공한다. 특히 이

* 종신회원

러한 Java 환경은 기존의 JVM(Java Virtual Machine)이 아닌 새로 개발한 Dalvik VM을 제공한다. 애플사와 마찬가지로 SDK를 제공하며 플랫폼이 공개되어 있으므로 모바일 기기뿐만 아니라 태블릿, 넷북 등에서도 다양하게 활용되고 있다[4].

노키아와 인텔은 미고(MeeGo) 플랫폼을 2010년 5월에 공개했다[5]. 미고는 리눅스 기반의 플랫폼으로 노키아가 MID(Mobile Internet Device)를 타깃으로 개발하던 마에모(Maemo)와 인텔이 넷북을 타깃으로 개발하던 모블린(Mobiln) 플랫폼을 통합한 것이다. 어플리케이션 제작을 위한 SDK는 물론이고 플랫폼 역시 공개되어 있어 넷북, MID 등에 적용될 것으로 예상된다.

국내의 경우, 삼성전자가 자체 개발한 휴대폰용 플랫폼인 바다(bada)를 공개했다[6]. 바다는 어플리케이션 개발이 가능하도록 SDK와 API를 제공하고 있으며, 삼성에서 운영하는 앱스토어를 통하여 어플리케이션을 유통할 수 있다.

## 3. 가상화 기술의 활용

가상화 기술은 물리적인 컴퓨팅 자원을 논리적인 컴퓨팅 자원으로 추상화함으로써 컴퓨팅 자원의 활용에 있어서 유연성을 제공한다[1,2]. 이러한 장점을 제공하는 가상화 기술은 서버 컴퓨팅 환경에서 복잡했던 컴퓨팅 환경을 단순화시키고 작업처리의 분산 및 관리의 효율성을 향상시켜 비용 절감 효과를 거두고 있다. 또한 실시간성 보장을 위해서 사용되던 임베디드 가상화 기술이 최근에는 모바일 환경에도 차츰 적용되고 있는 추세이다.

### 3.1 서버 가상화

IT 산업이 발전함에 따라 다양한 서비스를 제공하기 위하여 많은 서버를 구축하였으나, 실제로는 각 서버의 활용률이 낮아서 자원 낭비가 심하고 관리 비용이 높아지고 있다. 서버 가상화는 가상화 기술이 가장 활발하게 사용되고 있는 분야로 서버 병합 및 관리의 효율성을 증대시켜 유지, 관리 비용을 감소시키는 목적으로 사용된다. 활용률이 낮은 서버들을 가상 머신을 이용하여 하나의 물리적인 서버로 통합하여 전체 활용률을 높이고, 시스템 업그레이드 및 백업에도 유용하게 사용이 가능하다. 이러한 서버 가상화 기술은 컴퓨팅 자원 활용률을 높임으로써 전력 사용을 감소시키기 때문에 그린 IT의 기반 기술로도 주목받고 있다. 대표적인 서버 가상화 솔루션으로는 Citrix사의 Xen과 VMware사의 VMware Server 등이 있다[7,8].

### 3.2 데스크톱 가상화

데스크톱 가상화는 가상화 기술을 이용하여 사용자에게 언제 어디서나 자신의 데스크톱 환경을 제공해주는 기술이다. 최근 Citrix사가 Xen 기반으로 HDX(High Definition User Experience)를 출시하였고, VMWare사가 VMware View를 출시함으로써 데스크톱 가상화 기술이 널리 활용되고 있다.

이는 자신의 데스크톱 환경을 언제 어디서나 동일하게 사용할 수 있는 Thin Client 환경을 제공한다는 장점이 있으며, 보안성이 요구되는 기업의 어플리케이션이나 데이터가 특정 가상머신에만 존재하게 함으로써 보안성을 높일 수 있는 장점이 있다. 또한 각 사용자 데스크톱의 운영체제 및 어플리케이션 설치, 유지보수 등을 중앙집중형으로 효율성 있게 관리할 수 있다는 특성을 갖는다.

### 3.3 임베디드 가상화

임베디드 시스템의 중요한 특징 중 하나는 실시간성이다[9]. 실시간성을 만족시키기 위해서 많은 임베디드 시스템들이 전용 RTOS를 사용하고 있다. 하지만 하드웨어의 발전과 디바이스의 융합으로 인하여 임베디드 시스템에서 요구되어지는 기능이 늘어남에 따라 실시간 처리를 위하여 최적화된 RTOS에서 모든 기능을 구현하는데 어려움이 있다.

임베디드 환경에서 가상화 기술을 이용하여 RTOS와 GPOS를 가상머신으로 동시에 수행함으로써 이러한 문제를 해결하는 것이 가능하다. RTOS에서는 통신 프로토콜과 같이 실시간성을 필요로 하는 작업들을 수행하고 GPOS는 다양한 사용자 서비스 및 어플리케이션을 수행함으로써 실시간성과 기능적 다양성을 모두 만족시킬 수 있다. 기존의 대표적인 임베디드 가상화 솔루션으로는 RTLinux가 있다[10].

### 3.4 모바일 가상화

최근 스마트폰의 발전에 따라 기존의 모바일 폰은 단순한 통화 기능의 실행을 넘어서 어플리케이션을 실행하기 위한 플랫폼이 되었다. 실행되는 어플리케이션도 단순 게임을 넘어서 금융거래, 기업용 어플리케이션까지로 점차 확대되고 있다. 이와 같이 모바일 장치의 기능이 다양해짐에 따라 시스템의 복잡성이 증가하고 있으며, 이에 따라 신뢰성을 보장하기가 더욱 어려워지고 있다[11,12]. 이러한 모바일 플랫폼의 보안성을 향상하기 위한 노력으로 secure execution 기능을 제공하는 모바일 가상화 기술이 주목받고 있다. 대표적으로 VMware사의 MVP(Mobile Virtualization Platform), Virtual-Logix

사의 VLX for Mobile, Open Kernel Labs의 OKL4 등이 있다[13-15]. 또한 삼성전자에서는 서버 가상화 솔루션인 Xen을 모바일 장치에 적용한 XenonARM을 개발한 바 있다[16].

## 4. 가상화 기술

시스템 전체를 가상화하기 위해서는 프로세서와 메모리를 비롯하여 각종 장치들에 대한 가상화가 선행되어야 한다. 각각의 가상화 기술은 다양한 세부 기술로 나뉘어지며, 가상화 목적에 따라 선택이 가능하다. 본 절에서는 현재까지 연구·개발된 가상화 기술들을 소개한다.

### 4.1 프로세서 가상화

프로세서 가상화는 가상 머신 내에서 일반 명령어(normal instruction), 특수 권한 명령어(privileged instruction), 실행 모드 전환, 인터럽트 및 예외 처리 등을 다루는 기술이다. 호스트(host)는 가상 머신 안에서 동작하는 게스트(guest) 운영체제가 마치 실제 하드웨어를 독점적으로 사용하고 있는 것처럼 보이기 위해 다양한 기술들을 사용한다. 대표적으로는 게스트 운영체제를 수정하여 최적의 성능을 제공하는 반가상화(paravirtualization) 기술과 코드 변환 기술 등을 이용하여 게스트 수정 없이 가상화를 지원하는 전가상화(full-virtualization) 기술이 있다. 최근에는 INTEL과 AMD 등에서 개발한 하드웨어 지원 프로세서 가상화 기술도 널리 사용되고 있다.

#### 4.1.1 반가상화

반가상화 기술은 운영체제 위에서 동작하는 일반 프로세스가 시스템 자원에 접근할 때 시스템 콜(system call)을 이용하는 것과 유사한 형태를 가진다. 아래 그림 1에서 보이는 바와 같이 실행 모드 전환, 인터럽트/예외 처리 등 특수 권한이 필요한 명령어들을 사용하는 운영체제 코드를 직접 수정하여 시스템 콜과 비슷한 형태의 하이퍼콜(hyper-call)로 대체함으로써 게스트 운영체제에서 특수 권한이 필요한 명령어들을 모두 제거한다. 이를 통해 일반 명령어만으로 구성된 게스트에서 에뮬레이션이나 코드 변환 기술 없이 직접 특수 권한 코드가 수행되게 하므로 다른 가상화 기술에 비해 성능 저하가 적다는 장점을 갖는다.

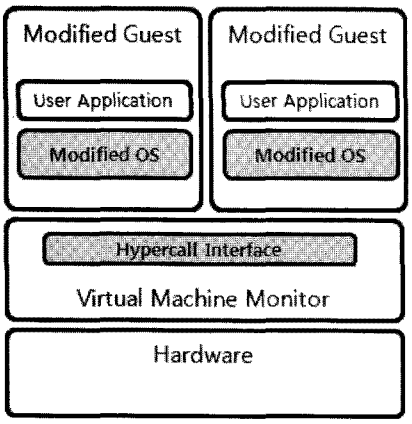


그림 1 반가상화 구조

#### 4.1.2 전가상화

전가상화 기술은 게스트 운영체제의 수정 없이 가상 머신 내에서 실행하는 것을 목적으로 한다. 이를 위해 하드웨어 전체를 소프트웨어로 구현하는 에뮬레이션 기법과 실행 코드를 실행 시에 동적으로 변환하여 특수 권한 명령어를 실행하는 코드 변환(binary translation) 기술 등이 있다.

에뮬레이션의 경우, 일반 명령어와 특수 권한 명령어가 모두 소프트웨어로 구현되기 때문에 속도가 매우 느리다. 그러나 하드웨어 의존성이 없기 때문에 다른 프로세서나 하드웨어 플랫폼에서도 사용이 가능하며, 이에 따라 임베디드 장비의 개발 환경을 구성하는데 주로 사용된다.

코드 변환 기술은 수정되지 않은 게스트에서도 특수 권한 명령어를 실행 할 수 있도록 동적으로 모든 코드를 변환하는 기술이다. 가상 머신은 게스트의 코드를 직접 실행하지 않고 기본 블록(basic block)으로 나누어 코드를 관리한다. 실제 실행은 변환된 코드 블록을 코드 캐시(code cache)에 저장한 후 이루어지며, 프로그램의 높은 지역성(locality)에 따라서 효과적인 성능을 제공한다. 코드 변환 기술을 이용하는 전가상화 기술은 반가상화 기술에 비해서는 느리지만, 에뮬레이션에 비해서는 월등히 빠르고 또한 게스트의 수정이 필요 없다는 장점을 갖는다.

#### 4.1.3 하드웨어 지원 프로세서 가상화

전가상화 기술의 경우 게스트를 수정할 필요가 없다는 장점이 있지만 구현이 어렵고 반가상화에 비해 성능이 낮다는 단점을 갖는다. 이를 개선하기 위해서 Intel과 AMD에서는 서로 유사한 동작 방식을 갖는 하드웨어 지원 가상화(hardware-assisted virtualization) 기술을 발표하였다. Intel에서 제공하는 하드웨어 지원 프로세서 가상화 기술인 VT-x는 기존의 4단계 실행 권한 위에 VMX root/non-root 모드를 추가하여 호스트와 게스트가 독립적으로 4단계 실행 권한을 사용하도록 지원한다[17]. 그리고 VMX non-root 모드에서 동

작하는 게스트가 특수 권한이 필요한 명령어를 수행하면 자동으로 예외를 발생시켜 VMX root 모드에서 이를 처리한다. 그 결과 복잡한 소프트웨어적인 구현 기술 없이 전가상화를 지원하며, 구현이 쉽고 성능 저하가 적다는 장점을 갖는다.

### 4.2 메모리 가상화

메모리 가상화는 가상 머신 안에서의 주소 변환을 다루는 기술이다. 실제 머신에서는 페이징을 지원하는 메모리 관리 장치(MMU)를 이용해 가상 주소를 물리 주소로 자유롭게 변환할 수 있지만, 가상 머신 내에서는 메모리 관리 장치에 접근이 불가능하기 때문에 다른 방식으로 이와 같은 문제를 해결해야 한다. 메모리 가상화 기술로는 게스트 운영체제를 직접 수정하여 실제 물리 주소 공간을 직접 접근 및 관리하는 다이렉트 페이징(direct paging) 기술과 수정되지 않은 게스트를 지원하는 섀도우 페이징(shadow paging) 기술 등이 있다. 최근에는 하드웨어 지원 가상화 기술을 이용한 메모리 가상화 기술도 널리 사용되고 있다.

#### 4.2.1 다이렉트 페이징

다이렉트 페이징은 반가상화 기술과 같이 게스트 운영체제의 메모리 관리 부분을 수정하여 게스트가 실제 머신의 물리 주소 공간에 직접 접근할 수 있게 하는 기술이다. 다이렉트 페이징을 지원하는 대표적인 가상 머신인 Xen의 경우, 게스트 초기화 시 필요한 물리 주소 공간을 미리 할당하여 게스트에게 알려준다. 게스트는 이 할당받은 주소를 이용하여 다음 그림 2에서 보이는 바와 같이 페이지 디렉토리(PGD), 페이지 테이블(PTE) 등을 생성하고 관리한다. 이와 같이 다이렉트 페이징은 게스트를 수정해야 하기 때문에 리눅스와 같이 공개된 운영체제에만 적용이 가능하다는 단점이 있지만 그에 반해 구현이 비교적 쉽고 성능 저하가 적다는 장점을 갖는다.

#### 4.2.2 섀도우 페이징

섀도우 페이징은 수정되지 않은 게스트를 지원하는 메모리 가상화 기술이다. 섀도우 페이징을 사용하는 게스트는 마치 자신만의 메모리 관리 장치를 갖고 있는 것처럼 동작한다. 호스트는 다음 그림 3에서 보이는 바와 같이 섀도우 페이지 디렉토리(SPGD)와 섀도우 페이지 테이블(SPTE)를 생성하여 게스트가 사용하는 메모리 공간을 관리하고 실제 물리적인 주소 변환을 돕는다. 이를 위해 호스트는 게스트가 사용하는 모든 페이지 디렉토리와 테이블을 수정이 불가능하도록 설정하고, 이에 대한 수정 요청 시 자동으로 페이지 폴트가 발생하여 호스트가 해당 요청을 섀도우 페이지 테이블에 반영할 수 있도록 한다. 이와 같은 기술은 수정되지 않은 게스트를 지원할 수 있다는 장점이 있지만 구현이 어렵고 성능 저하가 따르기 때문에 현재 마이크로소프트사의 윈도우즈와 같이 소스가 공개되어 있지 않는 게스트를 지원하기 위해 사용된다.

#### 4.2.3 하드웨어 지원 페이징

섀도우 페이징의 단점을 보완하기 위해 INTEL과 AMD에서는 게스트에서도 독립적인 하드웨어 주소 변환 기능을 제공할 수 있도록 하는 기술을 개발하였다. 이는 게스트에서 주소 변환에 의해 나온 게스트 물리 주소를 호스트에서 마련해 놓은 추가적인 변환 테이블을 이용하여 실제 물리 주소로 변환한다. 이를 위해 호스트는 섀도우 페이징 기법처럼 복잡한 처리 과정 없이 오직 게스트 물리 주소를 실제 물리 주소로 변환해 주기 위한 테이블을 만들어 주면 된다. 이 기술은 메모리

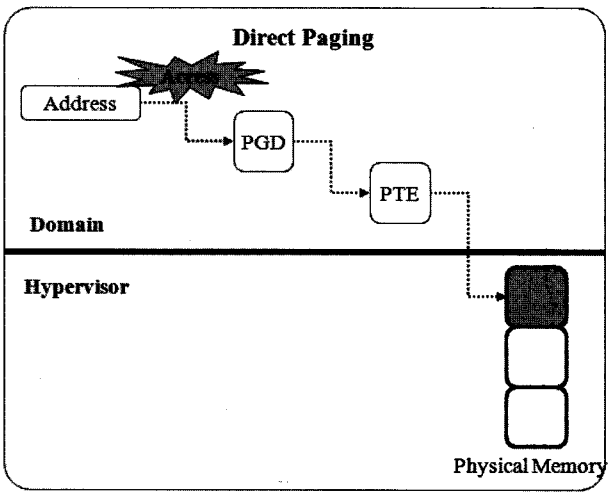


그림 2 다이렉트 페이징

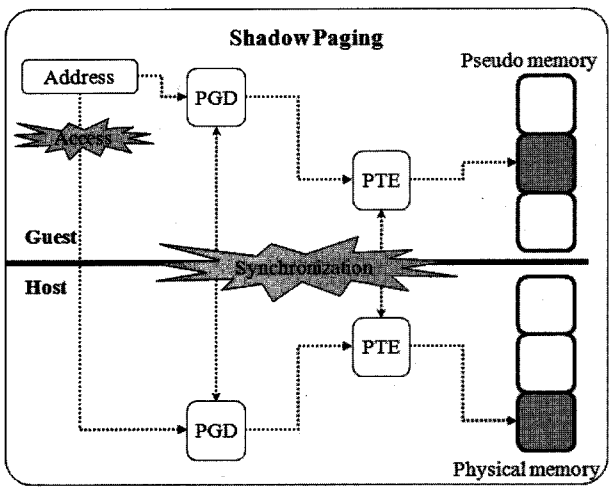


그림 3 섀도우 페이징

변환이 모두 하드웨어에서 이루어지기 때문에 성능 저하가 적고, 섀도우 페이징과 같이 게스트의 메모리 사용을 일일이 관리 및 추적할 필요가 없다는 장점을 갖는다.

### 4.3 장치 가상화

장치 가상화 기술은 키보드, 그래픽카드, 하드디스크, 네트워크 장치 등의 물리 장치를 게스트에서 사용할 수 있도록 지원하는 기술이다. 여러 게스트들이 하나의 물리 장치를 서로 나누어 사용해야 하기 때문에 호스트는 다양한 방식으로 각각의 장치들을 효과적으로 사용할 수 있도록 돕는다. 대표적인 장치 가상화 기술로는 호스트와 게스트가 IPC를 통해 장치 관련 정보를 주고받는 프론트/백 엔드 드라이버 모델과 실제 물리 장치와 동일하게 동작하는 가상의 장치를 소프트웨어로 구현하는 에뮬레이션 방식이 있다. 또한, 장치에서 하드웨어적으로 가상화 환경을 인식하여 동작하게 하는 하드웨어 지원 장치 가상화 기술이 있다.

#### 4.3.1 프론트/백 엔드 드라이버 모델

프론트/백 엔드 드라이버 모델은 주로 반가상화 기술에서 사용되며, 실제 물리 장치를 제어하는 백 엔드와 각각의 가상 머신 내에서 동작하는 프론트 엔드로 나뉜다. 다음 그림 4에서 보이는 바와 같이 게스트는 프론트 엔드 드라이버를 통하여 장치에 접근하며 이는 호스트의 백 엔드 드라이버를 경유하여 실제 물리 장치에 접근하게 된다. 호스트는 백 엔드에서 처리해야 할 게스트들의 장치 요청에 대해 우선순위 부여 등 여러 가지 정책을 통하여 조율한다.

#### 4.3.2 에뮬레이션

장치 에뮬레이션은 특정 물리 장치와 동일하게 동작하는 가상 장치를 소프트웨어로 구현하는 기술이다.

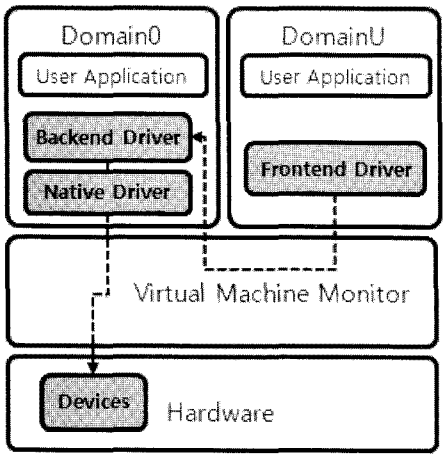


그림 4 프론트/백 엔드 드라이버 모델

호스트는 게스트가 가상 장치의 포트 혹은 메모리 주소 영역에 접근하려는 모든 요청을 가로채 가상 장치로 전달한다. 이는 게스트 입장에서는 기존의 드라이버를 그대로 사용할 수 있다는 장점을 갖지만 물리 장치 에뮬레이션 비용 및 성능 저하 등의 문제점도 가지게 된다.

#### 4.3.3 하드웨어 지원 장치 가상화

가상화 기술이 일반화되면서 프로세서, 메모리뿐만 아니라 각종 장치들에도 가상화 기술이 적용되고 있다. 가상화를 지원하는 장치는 여러 게스트가 동시에 하나의 장치에 접근할 수 있도록 도움으로써 기존의 가상 장치 지원 방식과 달리 기존의 드라이버를 그대로 활용할 수 있게 한다. 앞으로 다양한 형태의 가상화 기술을 지원하는 장치가 출시될 것으로 기대되고 있다.

## 5. 모바일 가상화

지금까지의 임베디드 가상화 기술은 범용 OS의 실시간 성능을 높이기 위한 용도로 주로 활용되어 왔다. 그러나 최근 모바일 환경으로의 변화에 따라 임베디드 가상화 기술은 모바일 가상화 기술로 발전하고 있으며, 이는 실시간성 지원이 아닌 보안성 및 관리 효율성 향상에 목적을 두고 있다.

### 5.1 Virtual Logix

Virtual-Logix사는 임베디드 시스템을 위한 가상화 솔루션을 주력으로 개발하고 있다. 최근 출시한 VLX for Mobile은 모바일 전용 가상머신으로 독일 Grundig 사의 피처폰인 U900에 적용 되었다[14,18]. VLX for Mobile은 기존의 임베디드 가상화 영역에서 축적해온 실시간성 제공 기술을 바탕으로 ARM, PowerPC 등 임베디드용 프로세서에 최적화된 구조를 통하여 MMU가 없는 임베디드 환경에서도 동작이 가능하며 임베디드 시스템 환경에서도 높은 가용성과 신뢰성을 보장한다. 최근에는 기존의 피처폰뿐만 아니라 스마트폰에도 적용이 가능하여 안드로이드, 윈도우즈 모바일 등의 GPOS도 지원하고 있으며, 이를 바탕으로 스마트폰을 위한 모바일 가상화 관련 연구 및 개발에 주력하고 있다.

### 5.2 OKL4

OKL4는 Open Kernel Labs에서 개발한 마이크로 커널 기반 모바일용 가상머신으로써 2009년 모토로라 피처폰 Evoke QA4에 적용되었다[15,19]. OKL4의 경우 가상머신 상에서 Linux와 RTOS를 모두 수행가능하게 하는 구조를 제공한다. OKL4는 최신 모바일 플랫폼과 기존의 피처폰에서 사용하는 RTOS를 모두 지원함으로써 기존의 소프트웨어 재사용성을 보장한다. 이를 통하여

모바일 폰의 핵심적인 기능 등은 RTOS에서 담당하며, 그 외의 추가 기능들은 GPOS에서 담당하게 한다. OKL4는 마이크로커널을 사용하여 안정성 측면에서 보다 개선된 구조를 갖고 있다.

### 5.3 VMware MVP

VMware는 2008년 말 실시간 임베디드 가상화 업체인 TRANGO를 인수하여 2009년 말 MVP(Mobile Virtualization Platform)을 출시하였다[13]. MVP는 안드로이드, 심비안, WinCE 등과 같은 다양한 최신 모바일 운영체제를 지원한다. VMware MVP는 VMware가 기존의 VMware ESX Server에서 사용하던 반가상화 기술을 이용하여 하나의 모바일 단말기에서 두 개의 가상 머신을 운용하여 하나는 개인 용도로 사용하고, 다른 하나는 업무 용도로 활용이 가능하다. VMware는 기존 축적된 서버 가상화 기술을 바탕으로 모바일 가상화 시장 확보에 주력하고 있다.

### 5.4. Xen on ARM

Xen on ARM은 2008년에 삼성전자에 의해 공개된 Xen 기반의 모바일 가상화 솔루션이다[16]. 이는 모바일 플랫폼의 보안성 향상을 위한 secure execution, secure boot, secure storage 등 다양한 보안 기술들을 제안하였다. 다음 그림은 Xen on ARM의 소프트웨어 구조이다.

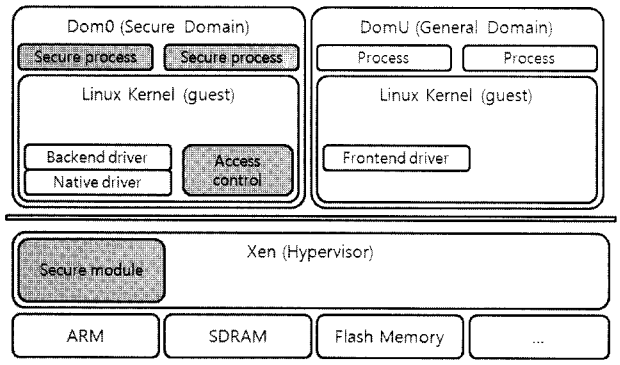


그림 5 Xen on ARM의 소프트웨어 구조

위 그림과 같이 Xen on ARM은 기존의 Xen의 구조를 그대로 따르고 있다. 하이퍼바이저(hypervisor)는 게스트들 간의 IPC와 스케줄링을 담당한다. Dom0는 프론트/백 엔드 드라이버 모델에 따라 실제 물리 장치를 제어한다. DomU는 프론트엔드 드라이버를 장치 요청을 Dom0로 전달한다. Dom0는 백엔드 드라이버로 들어온 모든 디바이스 요청을 관리하여 실제 물리 장치에 반영한다. 기존의 임베디드 가상화 기술들과는 달리 실제 물리 장치에 대한 제어를 하이퍼바이저에서 하지 않고 Dom0에서 하기 때문에 장치 드라이버를 새로 구현할 필요 없이 리눅스의 장치 드라이버를 그대로 활용할 수 있다는 장점이 있다. 그러나, 기존의 임베디드 가상화 기술들에 비해 성능상 오버헤드가 크다는 단점이 있다.

Xen on ARM은 하나의 물리머신에서 생성된 여러 개의 가상머신은 각각 격리(isolation)되어 있어 서로 임의의 접근이 불가능하다는 점을 이용하여 secure execution기능을 제공한다. 이를 위해 가상머신을 보안 정책에 따라 secure domain(Dom0)과 general domain(DomU)으로 구분한다. secure domain에는 전화, 주소록 등의 꼭 필요한 기본 프로그램과 인터넷 뱅킹 등의 보안에 민감한 프로그램을 사용하며, general domain에는 사용자 임의로 사용이 가능하도록 한다. 이를 통하여 만약 general domain이 악의적인 공격에 노출되더라도 가상화 기술에 의해 완벽하게 분리되어 있는 secure domaion은 보안 데이터를 안전하게 보관할 수 있게 된다.

### 5.5 MyAV

MyAV는 커널/사용자 주소 공간 분리 기술을 제안한 새로운 형태의 가상화 기술이다[20]. 기존의 가상화 기술들은 하드웨어 지원 없이 가상 메모리를 지원하기 위해 호스트와 게스트가 독립적인 가상 주소 공간을 사용하였다. 이로 인해 게스트가 동작 중일 때 하드웨어 인터럽트가 발생하는 등 호스트의 개입이 필요할 때는 호스트의 가상 주소 공간으로 전환한 뒤 필요한 작업들을 처리해야 한다. 이러한 빈번한 가상 주소 공간 변환은 매번 TLB 및 캐시를 비우는 추가 비용이 발생하여 전체적인 성능에 많은 영향을 주게 된다. 이를 해결하기 위해 MyAV에서는 다음 그림 6에서 보이는 바와 같이 커널/사용자 주소 공간 분리 기술을 제안하였다.

그림 6에서 보인 바와 같이, 기존의 범용 OS들은 커널과 사용자 공간이 동일한 가상 주소 공간 안에 존재하였다. 리눅스는 가상 주소 공간의 하위 3GB 영역을 사용자가 사용하게 하고, 상위 1GB 영역은 커널이 사용하고 있다. 이로 인해, 호스트와 게스트 모두가 리눅스인 Xen의 경우에는 호스트-게스트 전환을 위해 반드시 가상 주소 공간을 전환해야 하는 문제가 발생한다.

MyAV의 경우에는 게스트 커널이 상위 1GB 영역을 사용하지 않고 게스트 사용자와 동일한 하위 3GB 영역을 사용한다. 이로 인해 호스트 커널은 항상 상위 1GB 영역에 존재하면서 호스트-게스트 전환이 필요할 때 언제든지 가상 주소 공간 변환 없이 이의 처리를 지원한다. 하지만 기본적으로 커널이 상위 1GB 영역을 사용하는 리눅스 커널을 위와 같은 형태로 동작시키기 위해서는 몇 가지 해결해야 할 문제들이 존재한다. 첫

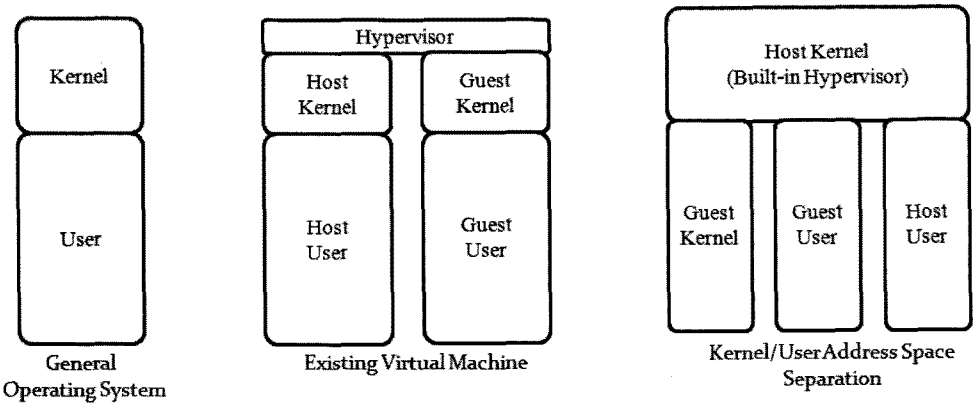


그림 6 커널/사용자 주소 공간 분리

번째는 게스트의 사용자 영역에서 시스템콜 혹은 예외 발생 시 호스트 커널은 이를 인식하여 게스트의 커널로 요청을 전달해야 한다는 것이다. 그리고 시스템콜 처리 과정 중 게스트 커널에서 게스트 사용자 영역에 접근이 필요할 때는 직접 접근이 불가능하기 때문에 접근이 필요한 사용자 영역을 임시로 매핑하여 접근하도록 해야 한다. 이는 매번 시스템콜 호출시마다 추가 비용을 발생시키지만 상대적으로 발생빈도가 높은 호스트-게스트 전환 비용을 제거함으로써 전체적으로는 더 좋은 성능을 보이게 된다. 또한 호스트와 게스트, 게스트 커널과 게스트 프로세스의 주소 공간을 분리함으로써 하드웨어 상의 추가적인 메모리 보호 메커니즘 없이 secure execution이 가능하다는 장점이 있다. 다음 그림 7은 MyAV의 소프트웨어 구조를 보인다.

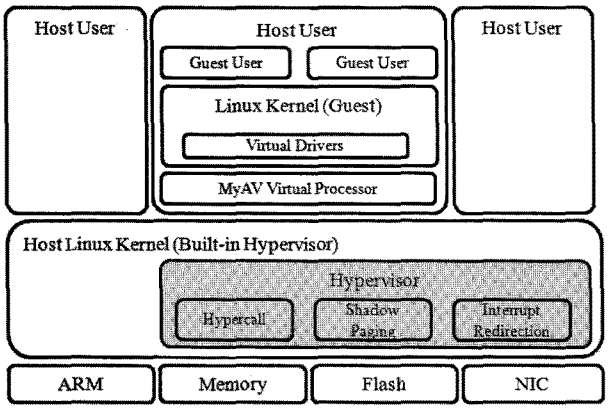


그림 7 MyAV 구조

위 그림과 같이 MyAV는 기존의 리눅스 커널에 내장된 형태의 하이퍼바이저이다. 하이퍼바이저에는 하이퍼콜 처리, 섀도우 페이징, 시스템콜/예외 전달 등과 관련된 부분이 구현되어 있다. 게스트 커널에는 커널/사용자 주소 공간 분리 기술이 적용된 MyAV 가상 프로세서 및 가상 장치 관련 부분이 추가되었다. 호스트 커널과 게스트 커널에 추가 및 수정된 부분이 모두 4000줄 미만이며, 특히 ARM 프로세서에 종속적인 코드는 600라인이 채 안된다. MyAV는 모바일 환경에 적용이 용이한 심플한 구조를 갖고 있으며 프로세서 종속성을 상당부분 개선하여 다양한 임베디드 환경에서도 적용이 가능하다.

## 6. 앞으로의 전망 및 결론

최근 모바일 단말의 성능 향상과 고기능성 모바일 플랫폼들의 등장으로 사용자들은 기존 데스크탑과 유사한 컴퓨팅 환경을 모바일 단말에서도 사용할 수 있게 되었다. 이러한 변화는 다양한 장점과 더불어 보안 문제, 관리 문제 등 다양한 새로운 문제들을 야기하게 되었다. 이러한 가운데 가상화 기술은 다양한 컴퓨팅 환경의 문제점을 개선하는 기술로써 주목받고 있으며 최근에는 모바일 환경에도 적용되어 다양한 가능성을 제시하고 있다. 모바일 가상화는 스마트폰의 보급과 함께 앞으로 모바일 환경에서 더욱 널리 사용될 것이며 이에 따라 다양한 관련 연구가 필요하다.

## 참고문헌

[1] R. P. Goldberg, "Survey of Virtual Machine Research," IEEE Computer Magazine, pp. 34-45, 1974.
[2] M. Rosenblom and T. Garfinkel, "Virtual Machine Monitors: Current Technology and Future Trends," IEEE Computer, Vol. 38, No. 5, pp. 39-47, 2005.
[3] App Store, http://en.wikipedia.org/wiki/App_Store
[4] Andriod Project, http://www.android.com/
[5] MeeGo Project, http://meego.com/
[6] Samsung Bada Project, http://www.bada.com/
[7] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Wareld, "Xen and the Art of Virtualization," In Proc.

of the 19th ACM Symposium on Operating System Principles (SOSP), pp. 164-177, 2003.
[8] VMware Server, http://www.vmware.com/products/server/
[9] G. Heiser, "The Role of Virtualization in Embedded Systems," In Proc. of the 1st Workshop on Isolation and Integration in Embedded Systems, pp. 11-16, 2008.
[10] V. Yodaiken, "The RTLinux Manifesto," Tech. Rep., Department of Computer Science, New Mexico Institute of Technology, 1999.
[11] J. Brakensiek, A. Droge, H. Hartig, A. Lackorzynski, and M. Botteck, "Virtualization as an Enabler for Security in Mobile Devices," In Proc. of the 1st Workshop on Isolation and Integration in Embedded Systems, 2008.
[12] M. Hypponen, "Malware Goes Mobile," Scientific American, Vol. 295, No. 5, pp. 70-77, 2006.
[13] VMware MVP (Mobile Virtualization Platform), http://www.vmware.com/products/mobile/
[14] VirtualLogix VLX for Mobile Handsets, http://www.virtuallogix.com/products/vlx-for-mobile-handsets.html
[15] G. Heiser, Virtualization for Embedded Systems, http://www.oklabs.net/_assets/download_library/OK_Virtualization_WP.pdf
[16] J. Hwang, S. Suh, S. Heo, C. Park, J Ryu, S. Park, and C. Kim, "Xen on ARM: System Virtualization using Xen Hypervisor for ARM-based Secure Mobile Phones," In Consumer Communications and Networking Conference, 2008. pp. 257-261, 2008.
[17] Intel VT Technology, http://www.intel.com/technology/virtualization/
[18] Grundig Mobile U900, http://www.linuxfordevices.com/c/a/News/Singlecore-Linux-phone-hitsthe-market
[19] G. Heiser, "The Motorola Evoke QA4 - A Case Study in Mobile Virtualization," http://www.oklabs.com/_assets/image_library/evoke.pdf, 2009.
[20] E. Ryu, I. Kim, J. Kim, Y. I. Eom, "MyAV: An All-round Virtual Machine Monitor for Mobile Environments," In Proc. of the 8th IEEE International Conference on Industrial Informatics 2010 (INDIN 2010), (To Be Appeared)

## 약 력

**김 정 한**

2008 세종대학교 컴퓨터소프트웨어학과 졸업
2010 성균관대학교 전자전기컴퓨터공학과 석사
2010~현재 성균관대학교 전자전기컴퓨터공학과 박사과정 재학중
관심분야 : 운영체제, 가상화, 정보보호
E-mail : junghan@ece.skku.ac.kr

**김 인 혁**

2006 성균관대학교 전자전기컴퓨터공학과 졸업
2008~현재 성균관대학교 전자전기컴퓨터공학과 석사과정 재학중
관심분야 : 운영체제, 가상화, 정보보호
E-mail : kkojiband@ece.skku.ac.kr

**민 창 우**

1996 숭실대학교 컴퓨터학부 졸업
1998 숭실대학교 컴퓨터학과 석사
1998~2005 한국 IBM UCL(Ubiquitous Computing Lab.) 전문과장
2005~현재 삼성전자 DMC연구소 책임연구원
2010~현재 성균관대학교 휴대폰학과 박사과정 재학중
관심분야 : 시스템 소프트웨어, 운영체제, 가상화, 모바일 플랫폼
E-mail : multics69@ece.skku.ac.kr

**엄 영 익**

1983 서울대학교 계산통계학과 학사
1985 서울대학교 전산과학과 석사
1991 서울대학교 전산과학과 박사
2000~2001 UCI-ICS 방문교수
1993~현재 성균관대학교 정보통신공학부 교수
관심분야 : 시스템 소프트웨어, 운영체제, 미들웨어, 가상화, 시스템 보안
E-mail : yieom@ece.skku.ac.kr