7

CE向 가상화(Virtualization) 기술

삼성전자 ▌서상범 · 이성민 · 장경아 · 유정현

1. 서 론

반도체 하드웨어의 집적도는 급격히 증가하여 컴퓨팅 하드웨어(HW) 비용은 감소되고 있으며, CE(Consumer Electronics) 기기의 기능 향상을 위하여 컴퓨팅 기능이 탑재되고 있다. 이와 같은 환경에서, digital convergence化에 따라 CE 기기의 HW 및 소프트웨어(SW) 시스템의 복잡도와 기능은 더 한층 증가하여, 시스템의 신뢰성을 보장하는 데 더 많은 비용이 들게 되며, 또한 CE 기기에 탑재된 범용 운영체제(OS)에서 실시간 처리를 요하는 멀티미디어 SW, 씨큐리티 보장을 요하는 인터넷 뱅킹 SW와 같은 서로 다른 응용이 공존시, 서비스 요구사항을 동시에 만족시키는 것은 힘든 일이다. 그리고, 사용자에게 친숙한 컴퓨팅 환경 제공이라는 메가 트렌드에 따라, 사용자가 늘 사용하는 컴퓨팅 환경을 기기간에 자유롭게 이동하며 사용할 수 있게 하는 기술의 개발 필요성이 제기되고 있다. 이러한 CE 시장과 사용자 요구 변화에 대응하여, 기기의 하드웨어 설계 변경 없이(또는 메모리 용량 추가 정도 등의 변경만으로), 유연하게(flexible) 상기의 요구 사항을 만족시키는 요소 기술로써 가상화 기술이 있다.

본 기고에서는 하나의 "real computing machine"을 SW를 이용하여 複數로 존재하는 것처럼 만드는 컴퓨팅 가상화에 대하여 논하고자한다. 이 기술을 적용한 컴퓨팅 기기 사용자에게는, 가상의 컴퓨팅 머신들이 하나의 하드웨어에서 동작되고 있으므로, 분산 컴퓨팅 개념과 가상화 개념이 결합된 시스템처럼 보인다. 가상화 기술을 적용한 컴퓨팅 시스템의 주요 특징을 정리하면, 1) HW-SW 사이의 De-coupling, 2) OS들 사이의 Isolation, 3) 유사 HW 기능의 Consolidation이라 할 수 있다(그림 1).

첫째, De-coupling 특징을 활용하면, 컴퓨팅 기기의 상태를 정지하고 복원하는 기술의 연구 개발을 가능하게하며, MS Windows에서 동작하는 SW를 설치 없이 기기간에 이동하며 사용하는 응용이 나온다. 기존의

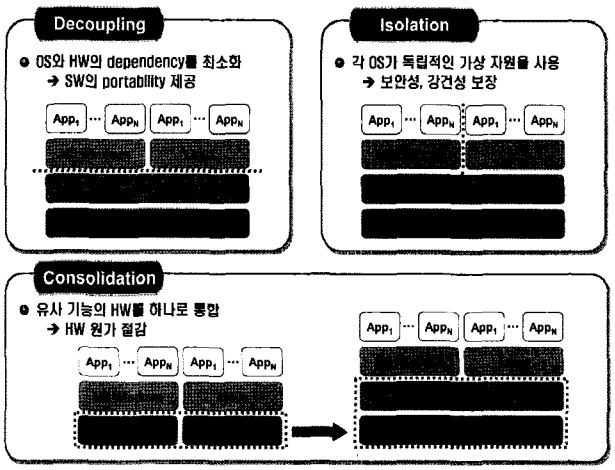


그림 1 가상화 기술의 특징

Java나 미들웨어를 활용한 SW 이동성 제공 방법과 다른 점은, Java나 미들웨어의 API(Application Programming Interface)에 상관없이 응용 SW가 작성되어도 이동성이 보장된다는 것이다. 둘째, Isolation 특징에 따라, 가상화 SW가 HW의 자원을 시간 차원과 공간 차원에서 각각의 OS에 할당하며 컴퓨팅 상태등 컨텍스트(context)를 OS 마다 분리하여 처리하므로, OS에게는 자기에게 할당받은 HW자원을 전부 사용하는 것처럼 보이게 하며, 특정 OS에 결함이 발생하여도, 다른 OS는 영향을 받지 않고 정상 동작하게 된다. 셋째, Consolidation은, 가상화 SW가 SMP/AMP(Symmetrical Multiple Processing/Asymmetrical Multiple Processing) 구조에 사용되는 CPU들을 통합하여, 하나의 CPU로 SMP나 AMP 구조를 SW적으로 동일하게 생성하는 효과를 제공하는 것을 말한다.

CE 업계의 가상화 기술 개발 수준은 시작 단계라 보이며, IT 업계에서 사용하는 가상화기술을 CE 기기의 embedded CPU에 그대로 적용하기에는 CPU 아키텍쳐 및 성능 등의 제약이 있어, CE 기기에 적합한 경량화된 가상화 기술의 개발이 필요하다. 이러한 필요성에 의하여 삼성 전자의 종합기술원에서 연구 개발한 Secure Xen on ARM(ARM CPU 기반 모바일 시스템

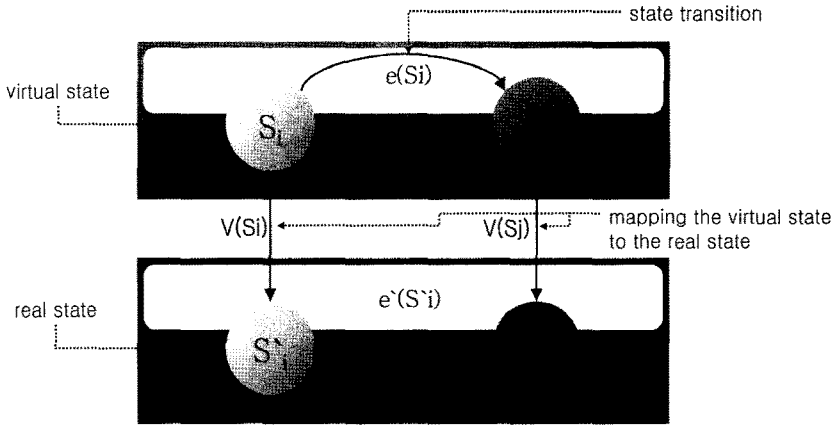


그림 2 가상화[9]

가상화 기술과 가상화 기반 보안 기술), 그리고, MobiWin (MS Windows XP OS 가상화 기반 Computing 환경 Migration 기술)을 본론에서 설명한다.

2. 가상화 이론

가상화란 물리적인 실제의 컴퓨팅 시스템을 논리적으로 재구성하여 가상의 컴퓨팅 시스템이 존재하는 것처럼 만드는 기술이다. 가상의 컴퓨팅 시스템은, 소프트웨어에 의하여 구현 가능하여, 물리적으로 한 개의 컴퓨터 하드웨어가 마치 여러 개 동작하는 것처럼 보이게 한다. 그림 2를 보면, 하단의 S'_i, S'_j 등을 포함한 박스가 물리적인 실제 컴퓨팅 하드웨어 시스템이라 하고, 상단의 S_i, S_j 등을 포함한 박스가 가상의 컴퓨팅 시스템이라 할 때, 가상의 컴퓨팅 시스템에서도 실제의 컴퓨팅 시스템에 대응되는 기능, 상태(state) 및 상태 천이(state transition)가 존재하도록 만들면 가상화되었다 할 수 있다. 이처럼 가상화 이론은 간단하지만, 가상화 시스템을 구현하는 것은 높은 기술 난이도를 갖으며, 실제 컴퓨팅 시스템을 가상화하는 데는 Linux나 Windows XP 운영 체제(OS)의 kernel을 설계 및 구현하는 정도의 기술력을 요한다.

3. Secure Xen on ARM: ARM CPU 기반 시스템 가상화 기술과 가상화 기반 보안 기술

Secure Xen on ARM은 ARM CPU 기반 시스템을 타깃으로 하는 가상화 기술로, 오픈 소스 가상화 소프트웨어인 Xen 아키텍쳐에서 보안성을 보장하도록 시스템 접근 제어(Access Control) 기능을 융합한 새로운 아키텍쳐로 설계되었으며, x86 CPU 기반 Xen의 interface와 호환되도록 개발되었다. 참고로, Xen은 고성능 서버 시스템을 위한 가상화 기술로 케임브리지 대학교 Computer Laboratory에서 처음 연구를 시작하여 2003년에 x86 CPU 기반 시스템을 가상화한 Xen SW 소스 코드가 공개되었다. Xen은 가상화시 발생하는 시스템 오버헤드를 최소화하도록, Xen SW 위에서 동작하는 Guest OS의 소스 코드를 수정하는 반가상화(Para-virtualization) 방식을 사용한다[1]. 현재까지 Xen은 IA-32/64, PowerPC로 CPU 지원을 확대하였고[3], 삼성전자가 ARM CPU 기반 HW 시스템 가상화 SW인 Secure Xen on ARM(ARM CPU 시스템 가상화 open source project name: XenARM이라 함)을 최초로 개발하여, Xen 커뮤니티에서 XenARM open source pro-

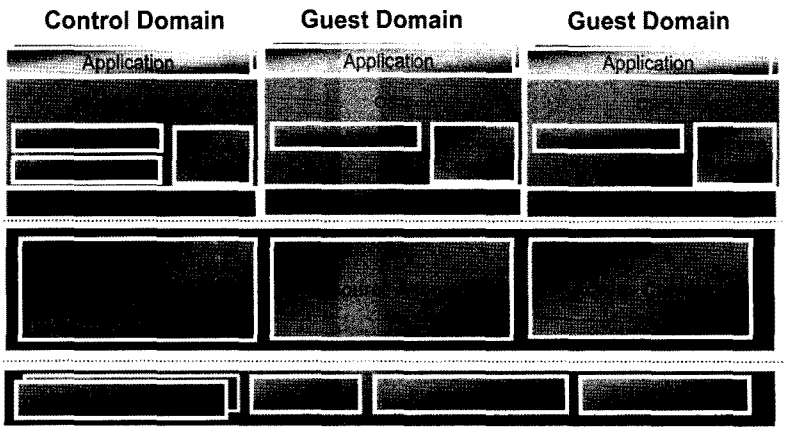


그림 3 Secure Xen on ARM 시스템 아키텍쳐

ject을 리드중이다[13].

Secure Xen on ARM은, 하나의 ARM CPU 기반 모바일 HW 플랫폼에서 여러 Guest OS들이 동시에 실행될 수 있는 환경을 제공하기 위해 ARM CPU, 메모리 및 I/O(입/출력) 장치를 가상화하였으며, 인터넷 뱅킹에 필요한 보안 수준을 제공하도록 가상화기술과 보안 기술을 융합한, 독창적 가상화 아키텍쳐이다[11]. 상위 수준의 시스템 아키텍쳐는 그림 3과 같다.

그림 3에서, 시스템의 가상화는 Domain Manager와 Resource Allocator가 담당하며, 시스템 보안을 보장하는 것은 Access Control이 담당한다. 이와 같은 가상화 SW 상에서, 여러 개의 OS가 동작하며, 그 OS 중 가상화된 시스템의 제어에 있어서 우선권을 갖는 영역을 Control Domain이라 부르며, 기타의 OS를 Guest OS라 하며, Guest OS가 동작하는 영역을 Guest Domain이라 부른다. 예를 들면, Domain Manager와 Resource Allocator는 CPU, memory 등을 가상화하여 각각의 Domain별로 CPU bandwidth와 memory 공간을 제공하며, 이 기능들을 통합하여 Xen on ARM이라 한다. 그리고 보안을 담당하는 Access Control 부분을 Xen Security라 하며, 다음 section에서 각각에 대하여 설명한다.

3.1 Xen on ARM

3.1.1 CPU 가상화

ARM 프로세서는 모두 7개의 CPU 동작 모드(SVC, FIQ, IRQ, UND, ABT, SYS, USR)를 지원한다. 7개의 모드를 특권 수준으로 분류하면 USR 모드만 비특권 모드이고 나머지는 모두 특권 모드이다. 이러한 CPU 환경에서는, 가상화가 적용 안된 기존 Linux OS 동작 환경의 경우 OS는 특권 모드에서, 사용자 프로세스는 비특권 모드에서 실행된다. 반면 Xen on ARM을 적용한 시스템에서는, HW 자원을 우선적으로 제어 할 수 있는 CPU 특권 모드에서 Xen on ARM SW를 실행하고, Guest Domain의 Linux OS와 사용자 프로세스를 CPU 비특권 모드에서 실행하기 때문에, Linux OS kernel의 메모리 영역을 사용자 프로세스로부터 보호하는데 문제가 생긴다. 그러므로 Xen on ARM SW는 ARM CPU 비특권 모드(USR 모드)에서 두 개의 가상 ARM CPU 모드(Kernel 모드, User Process 모드)를 생성하여, Guest Domain의 OS와 사용자 프로세스를 각각 Kernel 모드, User Process 모드에서 실행하게 한다. 또한 Xen on ARM SW는 ARM CPU 특권 모드를 할당하여 동작시키며, 편의상 Xen 모드라 한다(그림 4). 이런 구조를 통해 Guest Domain 에서 동작하는 OS의 수정을 최소화하면서, OS 상에서 동작하는 응용 SW들을 소스 코드의 수정 없이 사용할 수 있는 가상화 환경을 제공한다. Xen 모드에서는 가상 모드의 CPU 컨텍스트를 저장, 복원하는 동작 등을 한다.

그림 5는 Xen on ARM에서의 익셉션(exception) 처리 예를 보여주며, 익셉션이 발생하면 Xen 모드로

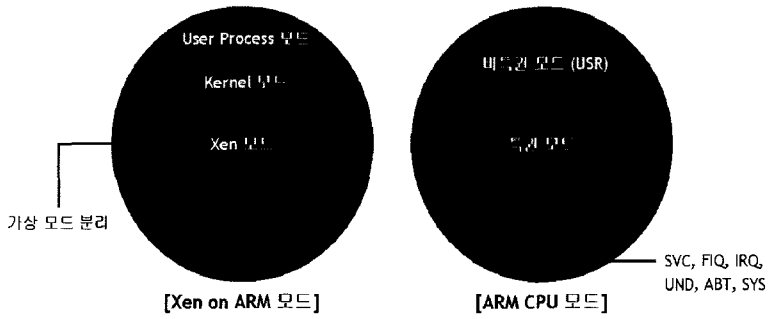


그림 4 CPU 모드 비교: Xen on ARM vs. ARM

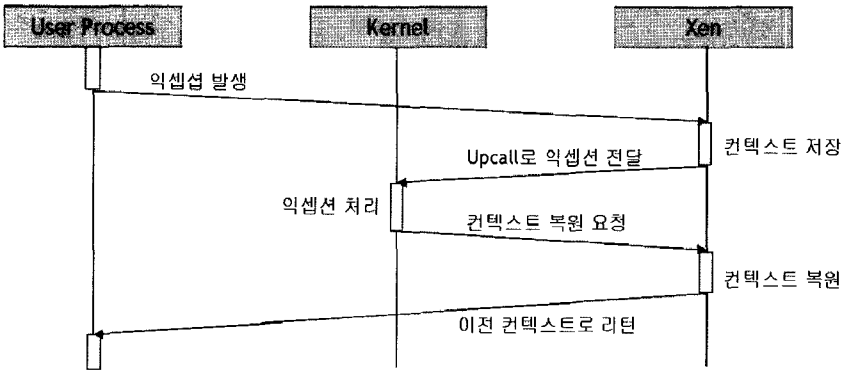


그림 5 Xen on ARM의 익셉션 처리 과정

전환되고, Xen on ARM은 Upcall 메커니즘을 통해 해당 익셉션을 Kernel 모드에 전달한다. 참고로, OS의 소스 코드 중 ARM CPU의 특권 모드에서 처리되는 sensitive instruction들은 Xen on ARM에서 제공하는 Hypercall 인터페이스로 대체된다. 이런 과정을 통하여, ARM CPU는 가상화 환경에서도 정상적으로 익셉션 처리를 하게 된다.

3.1.2 메모리 가상화

Xen on ARM 아키텍쳐에서는 Xen 모드에서 Guest Domain별로 메모리 공간을 할당하고, 메모리 매핑을 수행하기에, Guest Domain의 OS는 Hypercall을 통해 페이지 테이블 업데이트를 요청하여 메모리 매핑을 변경한다. Xen on ARM은 Guest Domain들이 사용하는 메모리가 서로 분리되도록 페이지 테이블을 관리한다.

Xen on ARM CPU 모드간의 메모리 보호를 위해 만족해야 하는 조건은 다음과 같다.

1) Xen on ARM 가상화 SW의 메모리는 Guest Domain으로부터 보호돼야 한다.
2) Guest Domain의 OS 메모리는 사용자 프로세스로부터 보호돼야 한다.
3) Guest Domain의 사용자 프로세스 메모리는 다른 사용자 프로세스로부터 보호돼야 한다.

Xen on ARM 아키텍쳐에서는 ARM CPU의 페이징 메커니즘을 사용하여 Guest Domain으로부터 Xen on ARM 가상화 SW의 메모리를 보호하고, 사용자 프로세스 간의 메모리 분리를 구현한다. 그러나 페이징 메커니즘은 User Process 모드와 Kernel 모드 간의 메모리 보호를 지원하는데 한계가 있다. 두 모드는 실제로 하나의 ARM CPU 물리 모드(USR 모드)에서 실행되기 때문에 페이징 메커니즘으로는 두 모드에 대해 메모리 접근 권한을 다르게 설정할 수 없기 때문이다. User Process 모드와 Kernel 모드 간의 메모리 보호를 위해 Xen on ARM 아키텍쳐는 ARM CPU의 도메인 접근 제어 메커니즘을 사용하여 User Process 모드와 Kernel 모드 간의 메모리 보호를 구현한다. Xen 모드, Kernel 모드, User Process 모드가 사용하는 메모리 공간을 각각 도메인(D0, D1, D2)으로 설정하고, Domain Access Control Register(DACR)를 통해 각 도메인에 대한 접근 권한을 설정한다. 그림 6에서와 같이 User Process 모드 실행 시, Kernel 모드 도메인(D1)에 대한 접근 권한이 No Access로 설정되어 Guest Domain의 OS kernel 메모리는 사용자 프로세스로부터 보호된다.

3.1.3 I/O 가상화

Xen on ARM은 Xen의 I/O 가상화에 다음의 두 가지 Split/ Coordinated 디바이스 드라이버 가상화 모델을 복합하여 사용한다. I/O 가상화 아키텍쳐는 그림 7과 같다.

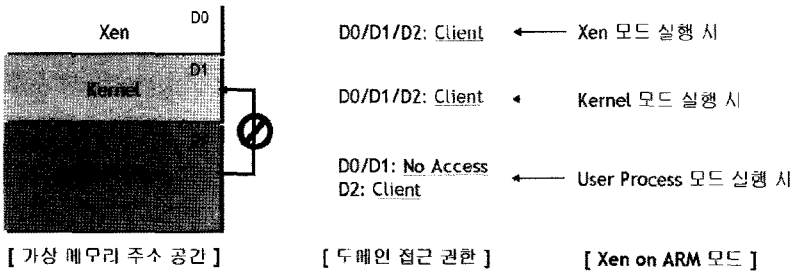


그림 6 Xen on ARM의 도메인 접근 권한 설정

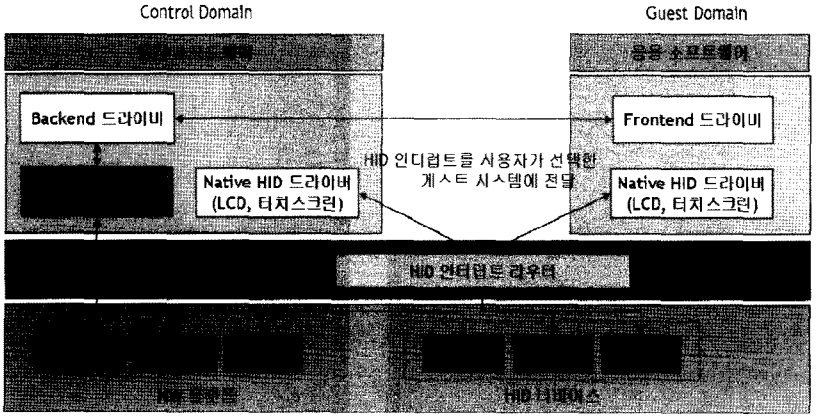


그림 7 Xen on ARM의 I/O 가상화 아키텍쳐

Split 디바이스 드라이버 모델: Xen 고유 모델

XenX86에서 사용하는 I/O 가상화 방식으로 실제 물리 디바이스에 접근하는 Native/ Backend 드라이버와 해당 I/O 디바이스에 대한 가상 드라이버 역할을 하는 Frontend 드라이버로 구성된다. Frontend 드라이버는 Guest Domain에, Backend 드라이버는 Control Domain에 위치하며, Backend 드라이버는 여러 Frontend 드라이버로부터 I/O 요청을 받아 이를 다중화하여 처리하는 기능을 한다. 네트워크와 스토리지 디바이스에 적용한다.

Coordinated Native 디바이스 드라이버 모델: Xen on ARM 모델

가상화된 기기의 사용자는 응용 SW를 자유롭게 바꿔가며 사용하는 데, 해당 응용 SW가 있는 Guest Domain이 자체적으로 갖고 있는 Native 디바이스 드라이버를 통해 I/O 디바이스에 직접 접근한다. HID(예, 터치 스크린, LCD) 디바이스에 적용한다.

3.2 Xen Security: 가상화 보안

가상화 환경에서는 기본적으로 하나의 HW상에서 Control Domain과 다수 개의 Guest Domain(이하 Domains이라 함)이 구동되기 때문에, HW상의 물리적인 자원들을 공유해야 하며, Domains간의 가상자원에 대한 공유도 필요로 한다. 이러한 환경에서 허가되지 않은 Domains으로부터의 자원 접근을 차단하여 기밀성과 가용성을 보장하여야 한다. 또한 시스템의 주요 컴포넌트에 대한 무결성 보장이 필요하다. 예로써, Symbian OS에서 발견된 Cabir[12]와 같은 Malware가 자원을 소진시켜서 무선 단말의 주요 서비스를 사용할 수 없도록 공격하는 DoS(Denial of Service) attack을 방지하는 기술이 필요하다.

3.2.1 보안 아키텍처

제안한 Secure Xen on ARM [13]에서의 가상화 보안 아키텍쳐는 그림 8과 같다. ARM CPU 상에서 동작하는 가상화 소프트웨어인 Xen on ARM을 통하여 두 개의 Domains(OS)을 동시에 구동하여, 가상화된 사용자 기기의 안전성과 유연성의 보장을 위해서, Secure Domain(Control Domain)과 Normal Domain(Guest Domain)을 정의한다. Secure Domain에는 전자 상거래, 인터넷 뱅킹 등의 주요 어플리케이션 및 서비스가 구동되며, 사용자가 임의로 소프트웨어를 추가/삭제하는 것이 제한된다. 즉, CE 기기 제조자가 제공하는 최소한의 검증된 소프트웨어만이 Secure Domain내에 설치되어 실행될 수 있다. Normal Domain은 사용자가 자유롭

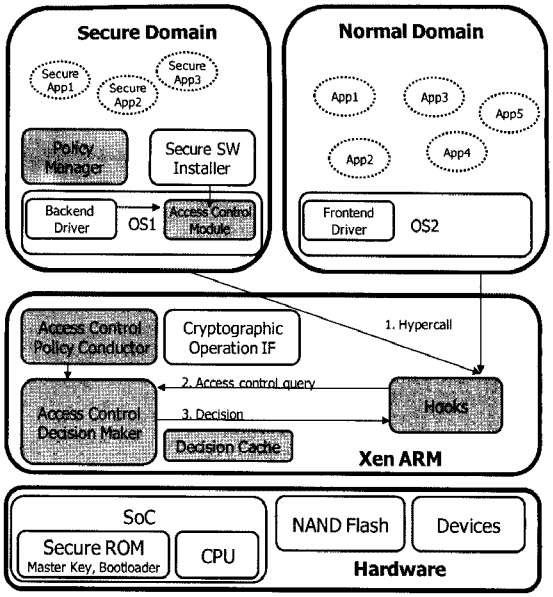


그림 8 가상화 보안 아키텍쳐

게 소프트웨어를 추가/삭제하는 것을 허용한다. 만약 Normal Domain에 Malware(바이러스 등의 악성 코드)가 설치된 경우라고 할지라도 Xen on ARM과 접근 제어 모듈에 의해서 Secure Domain 영역으로 침범할 수 없기 때문에, Domains 사이의 isolation을 보장한다.

3.2.2 Secure boot

Xen Security 아키텍쳐에서는 SoC 내부 ROM에 Master Key와 부트로더를 저장하고 Secure Xen on ARM의 bianry image, 접근 제어를 위한 access control policy 등의 주요 데이터들을 Master Key로 암호화하여 Flash 메모리에 저장한다. Secuer Xen on ARM이 CE 기기에 적용되면, 부팅시 부트로더는 Master Key를 이용하여 Flash 메모리에 저장된 데이터를 복호화하고 제조자가 전자 서명한 Secure Xen on ARM의 시그니처를 검증한 후 시그니처가 유효한 경우 Secure Xen on ARM을 메모리에 로드한다. 마찬가지로 Secure Xen on ARM은 커널을 메모리에 로드하기 이전에 커널에 대한 전자서명을 통해 커널의 변경 여부를 확인한 후 무결성이 확인된 경우 커널을 로드한다.

3.2.3 접근 제어

Xen Security 아키텍쳐에서는 가상화 SW와 Domains 각각에 적합한 Access Control Module(ACM)을 적용하는 다층 접근 제어(Multi-layer Access Control)를 함으로써, 기기의 안전성과 효율성을 높인다. ACM은 Flask architecture[10]에 기반하여 설계되어 새로운 ACM의 동적 추가가 가능하며 현재 5개의 access control models (TE, BLP, Biba, Chinese Wall, Samsung Proprietary)을 지원한다.

표 1 기존 가상화 보안 기술과의 비교

	IBM sHype	NSA XSM	Samsung Electronics ACM
현재 제공되는 Access Control Models	2가지 (TE and Chinese Wall)	5가지 (TE, Chinese Wall, RBAC, MLS, and MCS)	5가지 (TE, BLP, Biba, Chinese Wall, and Samsung Proprietary Model)
Access Control 대상	· Virtual Resources(도메인 사이의 통신시 Access Control) · Domain Management (도메인 생성 및 소멸 시)	· Physical Resources(IO memory, IRQ 접근 시 Access Control) · Virtual Resources(도메인 사이의 통신시 Access Control) · Domain Management (도메인 생성 및 소멸 시)	· Physical Resources(IO memory, IRQ 접근 시 Access Control) · Virtual Resources(도메인 사이의 통신시 Access Control) · Domain Management(도메인 생성 및 소멸 시)
DoS Attack 방지 기능	N/A	N/A	· 제한된 Resources(Memory, CPU, Battery, and Event Channels) 사용량에 대한 동적 Access Control을 수행하여 DoS Attack 방지
Access Control 위치	+ VMM (가상화 SW) 에서 Access Control 수행	· VMM에서 Access Control 수행	· Multi-Layered Access Control(성능 향상과 Fine-grained control을 위하여 VMM과 Domain에서 수행)

가상화 SW안에 있는 ACM은 Domains이 Memory, IRQ, IOMAP 등의 물리 자원(physical resources)에 대한 접근시 access control policy에 따라서 접근 허용 여부를 결정하고, event channel이나 grant table operation과 같은 Domains간의 통신에 필요한 가상 자원(virtual resources)에 대한 접근 제어를 담당한다. 반면, Secure Domain에서의 ACM은 file system, network 등의 자원에 대한 fine-grained access control을 제공한다.

기존의 가상화 보안을 위한 접근 제어 기술은 PC/Server 보안을 위해 개발된 IBM sHype[8]와 NSA XSM [4]이 존재하며, 삼성전자에서 무선 단말기의 보안을 위해 제안한 접근 제어 기술과는 표 1과 같은 차이점들이 있다.

4. MobiWin: MS Windows OS 가상화 및 Computing 환경 Migration (이동) 기술

4.1 PC-to-PC Computing 환경 Migration

Computing 환경 Migration은 소프트웨어 및 데이터를 이동식 저장 매체나 네트워크를 통해, PC간에 이동시켜 언제 어디에서나 계속적인 컴퓨팅(seamless computing)을 가능하게 하는 기술로, 단순 사용자 데이터뿐만 아니라 개인화된 작업 환경의 이동성을 지원하도록 하여 일반 PC가 있는 곳이면 어디에서나 프로그램 설치 없이 소프트웨어를 이용할 수 있으므로 기존 사무용 Note PC 등을 대체할 수 있는 新computing paradigm의 한 형태로 발전하고 있다.

이와 관련하여 종래에는 첫째, 특정 API로 이동하며 사용하는 응용 SW을 새로 작성(例: SanDisk社 U3[14]) 또는 OS의 소스코드를 수정하는 방식으로 이동성을 제공하였다. 둘째, 응용 SW나 OS의 수정 없이, 가상화 기술을 통해 OS와 사용자의 작업 환경을 함께 가상머신 형태로 저장하여 이동하며 사용하는, 가상 머신 모니터(Virtual Machine Monitor)를 활용하는 방식이 있다(例: EMC社 VMWare[15]). 세째, 엔터프라이즈 환경의 서버-클라이언트 구조에서는 응용 SW 및 OS의 수정 없이 서버관리자가 설치 과정 전후의 차이를 포함한 응용 SW 패키지를 생성하여 이를 스트리밍 방식으로 서비스하고 있다.

그러나, 첫째 방식은 소스코드가 공개된 응용프로그램이나 OS에 한하여 이동성을 지원하는 제한이 있으며, 둘째 방식은 OS를 포함한 가상머신을 이동시 수 GB의 저장 장치 공간을 요구하고, 부팅시간이 길며, 응용 SW 실행시 PC 성능 저하가 발생할 수 있다.

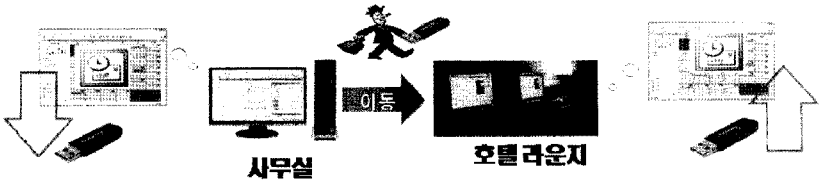


그림 9 Computing 환경 Migration 시나리오

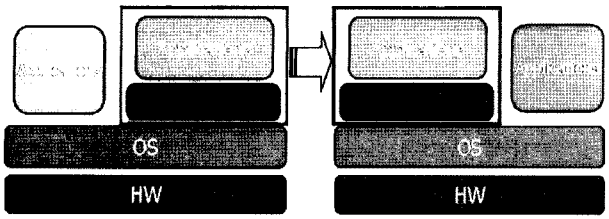


그림 10 OS 가상화 기반 시스템 아키텍쳐

세째 방식에서는, 응용 SW의 패키징시 전문성이 요구되기 때문에 일반 사용자가 해당 과정을 수행하는 것이 쉽지 않는 제약이 있다.

이에 대해 본 연구는 OS 가상화 기술을 기반으로 이동성을 지원함에 있어, OS 및 응용 SW의 수정없이 일반 PC에서 구동 가능하며, 사용성을 고려하여, 패키징 과정을 제거하였으며, 경량의 최적화된 가상화 SW를 제공하도록 하였다.

추상화된 OS 가상화 기반 시스템 아키텍쳐는 그림 10과 같다. 가상화 SW는 OS와 응용(applications) 사이에 위치하여 가상화 SW상의 응용들이 요구하는 시스템 자원 - 레지스트리, 파일시스템, 동적 링크 라이브러리(DLL), 서비스, 데스크 탑 등 - 을 제공하기 위하여, 가상화 SW의 자원과 실제 PC상의 OS 자원을 동적으로 결합하여, 하나의 독립된 OS처럼 동작하는 역할을 한다.

그림 11은 MS Windows XP OS 가상화 시스템 아키텍쳐로, 가상화를 통하여 만든 가상 OS(Virtual OS)와 실제 host OS 구조를 보여주며, 가상 OS는 Process Management, Execution Environment Management, File System Management, Profile Management, Security Management 등으로 구성 된다.

OS 가상화 기반 Computing 환경 Migration 시스템은 이동성을 보장하기 위해 가상화 초기화 단계, 가상화 실행 단계 및 가상화 종료 단계로 구동된다.

4.2 가상화 초기화 단계

이동식 저장매체 또는 네트워크를 통해 시스템 구동 전 사용자는 인증 과정을 수행하고, 이를 통해 관련된 로컬 host 자원과 가상화 계층의 자원들을 연결짓도록 하여 사전 점검한다.

4.3 가상화 실행 단계

가상화 초기화 수행 완료 후 우선적으로 사용자는 로컬 host OS와는 다른 가상 데스크 탑을 제공받게 되며, 이 가상 데스크탑은 사용자에게 친숙한 SW의 icon과 Windows OS 바탕화면으로 언제 어디에서나 이동하며 사용할 수 있다. 또한, 사용자의 응용 SW는 가상 OS 환경에서 설치되며, 설치 정보는 가상화 환경에 저장되고 로컬 host OS에는 아무런 영향을 끼치지 않도록 하며, 이때, 주요 모듈의 특징은 다음과 같다.

4.3.1 File System Management(Virtual File I/O)

File I/O와 관련된 시스템 호출이 가상화 대상 프로세스에서 발생하는 경우에, 해당 호출은 가상화된 시스템 드라이브로 경로가 변경되어 수행하게 된다. 예를 들어 가상화 대상 프로세스에서 C:₩Data₩a.txt라는 파일에 대한 접근을 요청하는 경우, 이러한 접근 경로는 가상화 계층을 거치면서(외부 저장장치의 디스크 볼륨이 F:인 경우) F:₩Data₩a.txt로 변경되어 해당 파일에 접근이 이루어지게 된다.

4.3.2 Process Management(Process Classification)

가상화 대상 프로세스들은 별도의 데이터베이스로 관리하여 가상화 대상 프로세스가 다른 프로세스에

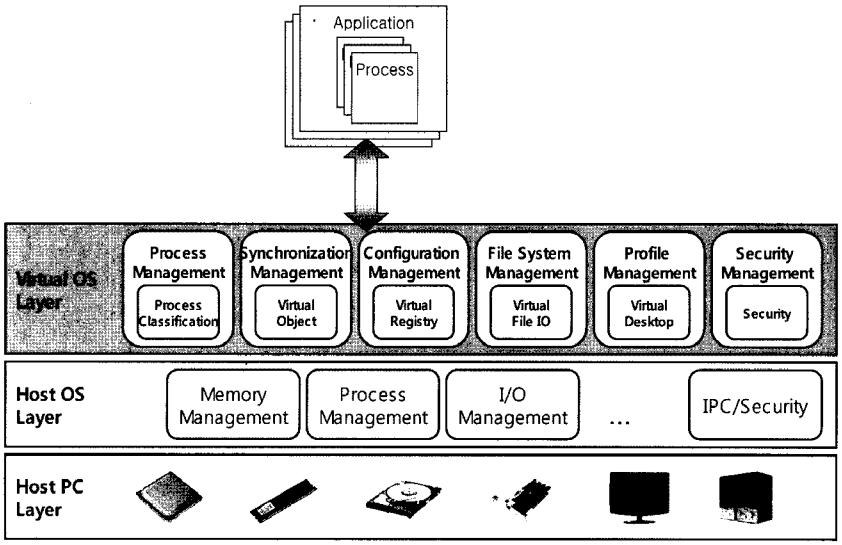
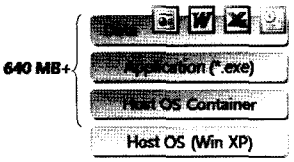
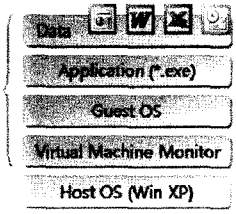
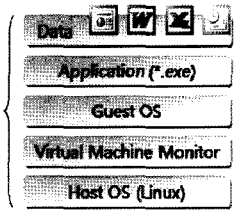


그림 11 MS Windows OS 가상화 시스템 아키텍쳐

표 2 가상화 기반 Migration 기술 비교

	OS 가상화	시스템 가상화	
이동 대상	 (오피스 설치 기준 data 크기)		
특징	- Virtual OS(Host OS Container) 기반 - Guest OS 필요 없음 - Host OS에 의존성 가짐 - CPU Overhead 작음 - Storage overhead 작음	- Virtual Machine Monitor 기반 - 다양한 Guest OS 지원 - Suspend/ Resume 지원 - CPU overhead 큼 - Storage overhead 큼	- Virtual Machine Monitor 기반 - Dummy PC 환경에 적합 - Suspend/ Resume 지원 - 부팅 속도 느림 - CPU overhead 큼 - Storage overghead 큼
회사	Samsung Electronics社 MobiWin RingCube社 MojoPac[7]	MS社 VirtualPC/ KeyChain[6] EMC社 VMware ACE	IBM社 Soul Pad[2,5]

대한 검색이나 접근 등을 위한 시스템 호출의 경우 해당 프로세스가 가상화 대상 프로세스인지 확인 작업을 거쳐 해당되는 프로세스에만 접근이 가능하도록 제어 한다. 이로써 가상화 OS 계층 위에서 동작되는 프로세스가 임의로 로컬 host의 프로세스에 접근하는 것을 차단할 수 있으며 역으로 로컬 host의 프로세스가 임의로 가상화 대상 프로세스에 접근하는 것을 차단할 수 도 있다.

4.4 가상화 종료 단계

사용자의 가상 데스크 탑 종료시 가상화 계층 상의 모든 프로세스는 함께 중단되고 로컬 host와 연결된 가상화 자원들은 분리하여 종료 과정을 수행한다. 이 때, 가상화 구동 단계에서 발생된 모든 시스템 변동 사항은 가상화 환경에 저장하여 이동 후 계속적으로 사용될 수 있게 한다.

본 연구는 OS 가상화를 기반으로 Computing 환경 Migration을 진행함에 있어 다른 연구 방식과 표 2처럼 비교할 수 있겠다.

5. 결 론

지금까지 모바일 컴퓨팅 환경에 적합하게 삼성전자가 연구 개발한, ARM CPU 기반 CE 시스템 가상화 기술 및 가상화 보안 기술, 그리고, MS Windows OS 가상화 기반 컴퓨팅 환경 Migration 기술에 대하여 설명하였다. 가상화 기술은 IT 업계에서는 대형 서버의 컴퓨팅 시스템 자원 활용성(utilization)을 높이는 데 응용되었으며, 최근 네트웍 상에 분산된 컴퓨팅 자원을 논리적으로 통합하며, 가상의 컴퓨팅 머신을 생성하는 기술로써 클라우드 컴퓨팅에 적용되고 있다. CE 업계의 경우는 최근에 와서 컴퓨팅 성능과 자원이 상대적으로 부족한 CE 기기에 적합하게 가상화 기술의 연구 개발이 활발하게 이루어지고 있는 중이다. Gartner에서 2008년에 발표한 2012년까지의 IT 분야 10대 유망 기술에도 포함되어 있는 가상화 기술은 앞으로도 많은 연구 및 응용이 이루어질 것으로 기대된다.

참고문헌

[1] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," in Proceedings of the nineteenth ACM symposium on Operating systems principles(SOSP '03). New York, NY, USA: ACM, 2003, pp. 164-177.

[2] Ramon Caceres, Casey Carter, Chandra Narayanaswami, M. T. Raghunath, Reincarnating PCs with Portable SoulPads, ACM/USENIX MobiSys 2005, pp. 65-78.

[3] David Chisnall, "The Definitive Guide to the Xen Hypervisor," Prentice Hall, 2007.

[4] G. Coker, "Xen Security Modules(XSM)," 3rd Xen Summit, IBM T.J. Watson, September 2006.

[5] IBM, IBM Research support, http://www.research.ibm.com/WearableComputing/SoulPad/soulpad.html

[6] Microsoft, Microsoft Virtualization support, http://www.microsoft.com/virtualization/default.mspx

[7] Mojopac, Mojopac-take your pc in your pocket, http://www.mojopac.com/portal/content/splash.jsp.

[8] R. Sailer, E. Valdez, T. Jaeger, R. Perez, L. van Doorn, J. L. Griffin, and S. Berger, "sHype: A

secure hypervisor approach to trusted virtualized systems," IBM Research Report, 2005.

[9] J. Smith, and R. Nair, Virtual Machines, chapter 1, Morgan Kaufmann, 2005.

[10] R. Spencer, S. Smalley, P. Loscocco, M. Hibler, D. Andersen, and J. Lepreau, "The Flask Security Architecture: System Support for Diverse Security Policies," In Proceedings of the 8th USENIX Security Symposium, 1999.

[11] Sang-bum Suh, "Secure architecture and implementation of xen on arm for mobile devices," 4th Xen Summit at IBM TJ Watson, Yorktown Heights, NY, April 2007. http://www.xensource.com/files/xensummit4/Secure_Xen_ARM_xen-summit-04%_07_Suh.pdf

[12] Symantec Corporation, "SymbOS.Cabir," http://security-response.symantec.com/avcenter/venc/data/epoc.cabir.html

[13] Secure Xen on ARM project, http://wiki.xensource.com/xenwiki/XenARM

[14] U3 smart drive, Bring the power of portable software to your usb flash drive. http://www.u3.com/smart/default.aspx

[15] VMware, Take control of virtual desktops across the enterprise. http://www.vmware.com/products/ace.

서 상 범

인하대학교 공학사
연세대학교 공학 석사
University of Cambridge 전산학 박사
대우통신/ 대우전자 중앙 연구소(1985~1998)
현재 삼성 전자 기술 총괄 종합기술원 수석연구원 Virtualization Project Lead, XenARM open source project의 maintainer
관심분야 : Virtualization, Embedded System Architecture, Cloud Computing, Wireless QoS
E-mail : sbuk.suh@samsung.com

이 성 민

2001 고려대학교 컴퓨터학과 이학박사
2003 동양SYSTEMS 기술연구소 주임연구원
현재 삼성전자 종합기술원 전문연구원
관심분야 : Virtualization Security, Home Network Security, Usable Security, Cryptographic Protocol 등
E-mail : sung.min.lee@samsung.com

장 경 아

2001 고려대학교 컴퓨터학과 박사
현재 삼성전자 기술총괄 종합기술원 전문연구원
관심분야 : OS 가상화, 시스템 보안, 클라우드 컴퓨팅
E-mail : kachang@samsung.com

유 정 현

1998 서울대학교 기계설계학과 학사
2000 서울대학교 기계설계학과 석사
2000~2002 대우전자 디지털신호처리 연구소
현재 삼성전자 종합기술원 전문 연구원
관심분야 : 임베디드 SW
E-mail : yjhyun.yoo@samsung.com