연구를 시작하며, 하이퍼바이저 Xen의 검증

조성근
서울대학교

ROSAEC 워크샵

목표

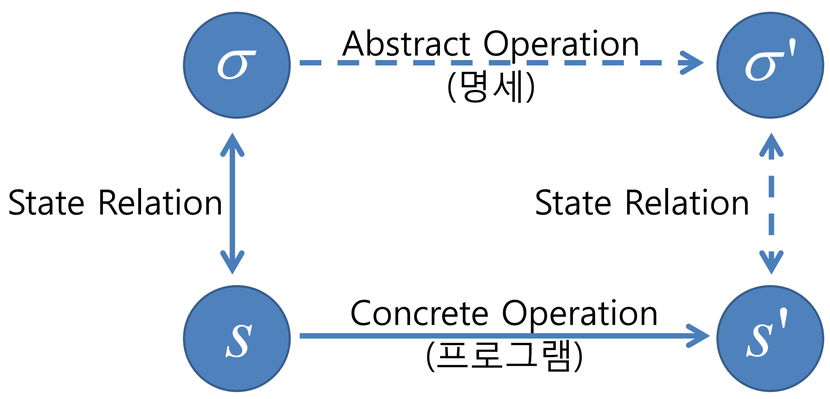
하이퍼바이저 Xen의 핵심이 되는 부분의 기능성(functionality)을 검증

프로그램의 검증이란?

프로그램이 주어진 명세(formal specification)대로 구현되었는지 증명
주어진 명세가 안전하게 디자인되었는지 증명

프로그램이 명세대로 구현되었다? Forward simulation / Refinement

명세의 실행과 프로그램의 실행에 의한 상태전이가 어떠한 상태 관계를 유지할 때, 프로그램이 명세대로 구현되었다고 할 수 있다.



증명 보조기(theorem prover)

Coq, Isabelle, Agda 등
대부분의 증명 보조기가 Curry-Howard isomorphism에 기반

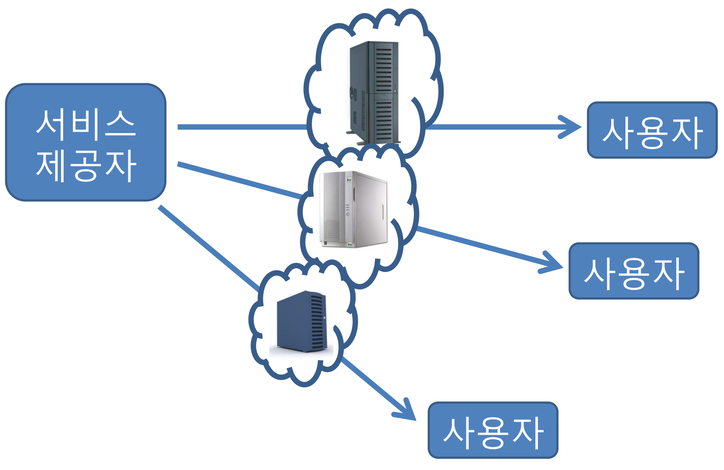
하이퍼바이저 Xen에 대해서

하이퍼바이저:

여러 운영체제가 하나의 컴퓨터에서 실행될 수 있도록 지원하는 하드웨어 가상화 기술

하이퍼바이저가 클라우드 컴퓨팅에서 왜 중요한가?

사용자: 현재 가지고 있는 코드나 실행파일을 수정할 필요가 없음
제공자: 가상기계 이주에 의한 효율적인 자원 관리가 가능
하드웨어 고장에 유연
시스템 확장이 용이



Xen

언어: C
크기: 650K lines
사용되는 곳:
아마존 EC2 (Elastic Computing Cloud)
구글 내부 클라우드 시스템, ganeti

중요한 부분들 및 성질

VCPU scheduler: 물리적 CPU를 여러 guest OS가 공유한다. 하나의 CPU 사용을 관리하는 scheduler가 올바르게 동작하는가?

Domain 0와 Domain U (Guest OS) 사이의 통신 관련:

Domain 0: 네트워크나 디스크와 같은 외부 하드웨어에 접근할 수 있는 특수한 Guest OS
Domain U: 사용자의 Guest OS

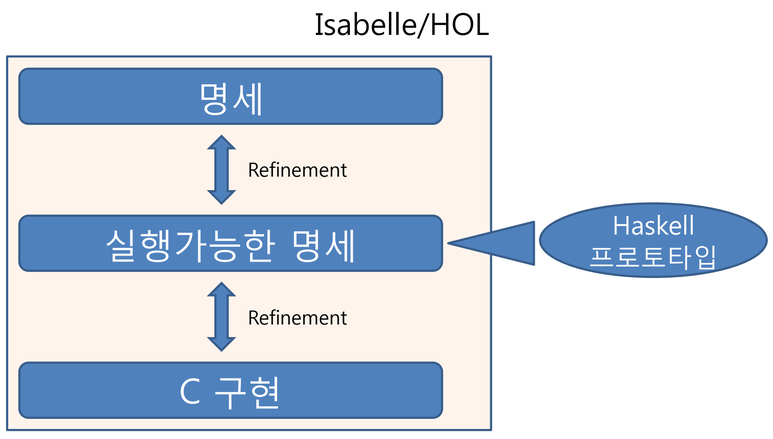
네트워크나 디스크 접근은 domain들 사이의 통신으로 이루어지고, 이 통신은 Xen에 의해 관리된다.

메모리 관리 관련: Xen은 각 guest OS에게 물리적 메모리를 할당하고 이들이 자신들에게 허가된 메모리에만 접근하도록 관리한다. 각 guest OS가 가지고 있는 page table들이 Xen에 의해서 안전하게 관리되는가?

Boot-loader의 무결성(integrity) 보장: 새로운 guest OS을 생성할 때 무결성이 보장되는가?

관련연구: 검증된 마이크로커널, seL4

seL4: Formal Verification of an OS Kernel. Gerwin Klein et al. *SOSP'09.*



특징

기존의 마이크로커널(L4)을 바로 검증하지 않고,
새로운 마이크로커널(seL4)의 구현과 검증을 동시에 수행
고수준의 프로그래밍 언어(Haskell)로 원형(prototype) 구현
개발팀과 검증팀의 타협점
검증팀: 고수준의 프로그래밍 언어를 바탕으로 명세 작성이 용이함
개발팀: C 구현을 완성하기 전에 OS의 시뮬레이션이 가능

비용: 2.2py(구현 및 모델링) + 20py(증명)

진행 상황 및 계획

관련연구 이해 / Xen의 구조 파악

Xen의 모델링 + 명세 작성
Refinement 증명
명세에 대해 특정 성질 증명