Lecture 5

KVM for ARM

Christoffer Dall and Jason Nieh

Operating Systems Practical

5 November, 2014

Contents

Outline

Virtualization: reminder

- *Virtual Machine* (VM)
- Popek and Goldberg (1974)

> *A virtual computer system is a hardware-software duplicate of a real existing computer system in which a statistically dominant subset of the virtual processor's instructions execute on the host processor in native mode.*

Virtualization: reminder (2)

- Duplicate underlying hardware
  - Virtualization $\neq$ emulation
- Run (almost) as fast as the physical machine
- Abstract hardware resources
  - CPU
  - Memory
  - I/O devices

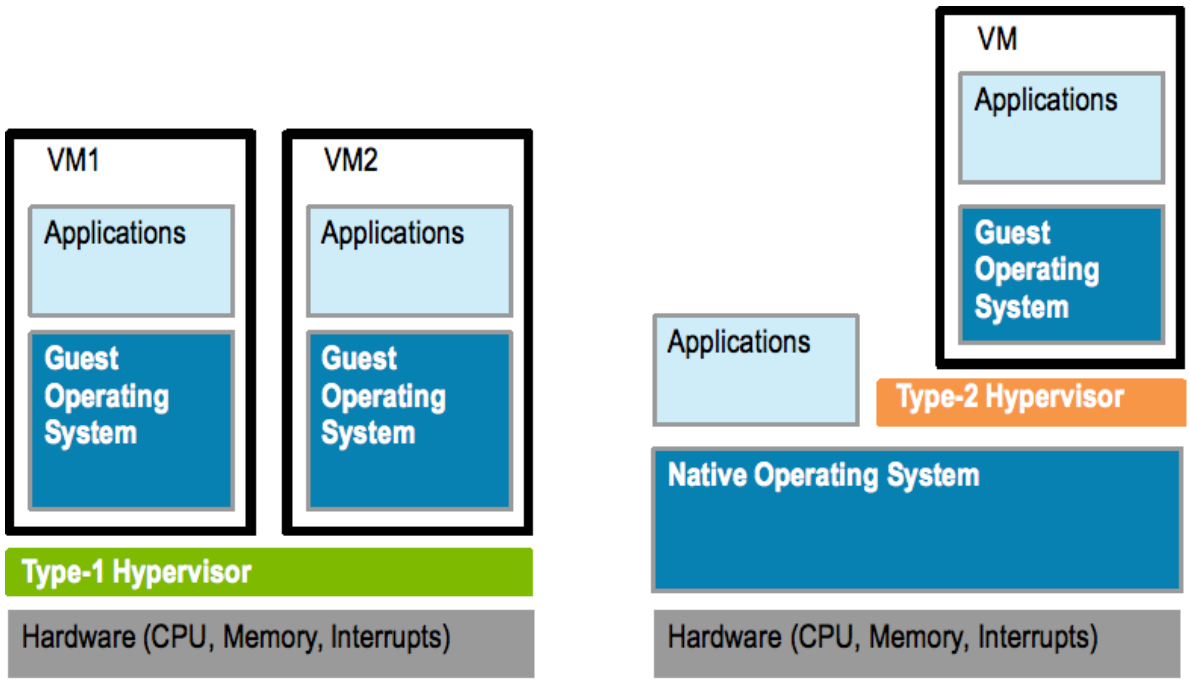
Virtualization: reminder (3)

Hypervisor, Virtual Machine Monitor (VMM)

- The two often get confused in literature
- One hypervisor per physical machine
- One VMM per VM

Virtualization architectures

- Type I: Baremetal or native
- Type II: Hosted



Source: `http://microkerneldude.wordpress.com`

Approaches to virtualization

- Binary translation
  - Translate sensitive instructions to non-sensitive ones
  - Can run unmodified guest OSes
- Paravirtualization
  - gr. para (alongside) + virtualization
  - More efficient than binary translation
  - Also more intrusive: requires modification of guest OS
- Hardware-assisted virtualization
  - Extend CPU with special "hypervisor" state
  - Trap-and-emulate
  - Similar for memory, I/O

Outline

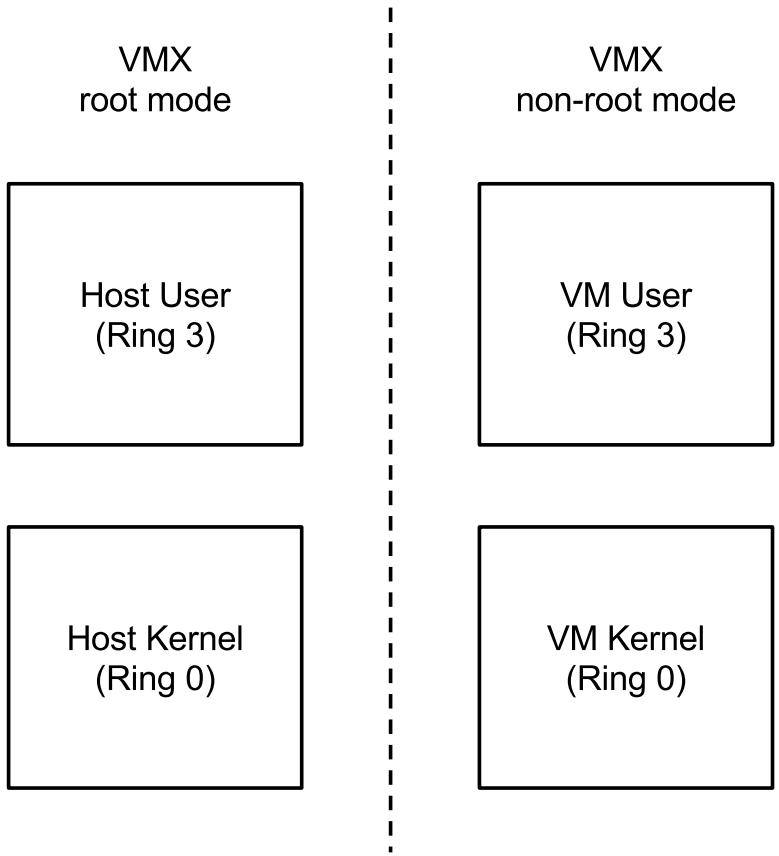
KVM

- *Kernel-based Virtual Machine*
- Module for the Linux kernel that turns it into a hypervisor
  - Part of the Linux mainline kernel (since 2.6.20)
- Exposes `/dev/kvm` to user space
- Requires a user space host to run
  - e.g. QEMU

KVM/QEMU on x86

- `kvm.ko` module provides virtualization interface via `/dev/kvm`
- QEMU sets up the guest VM using `/dev/kvm`
- This requires hardware assisted virtualization (VT-x or AMD-V)
- Otherwise QEMU will fall back to software emulation
- KVM architecture is **strongly** dependent on Linux/x86

Virtualization on x86 (Intel VT-x)



- AMD-V Secure Virtual Machine looks similarly

Outline

ARMv7-A Instruction Set Architecture

- Base ISA is **not** virtualizable
- CPU modes/Privilege Levels:
  - User mode (PL0)
  - Supervisor mode (PL1)
  - Monitor mode (Secure PL1)
  - ...

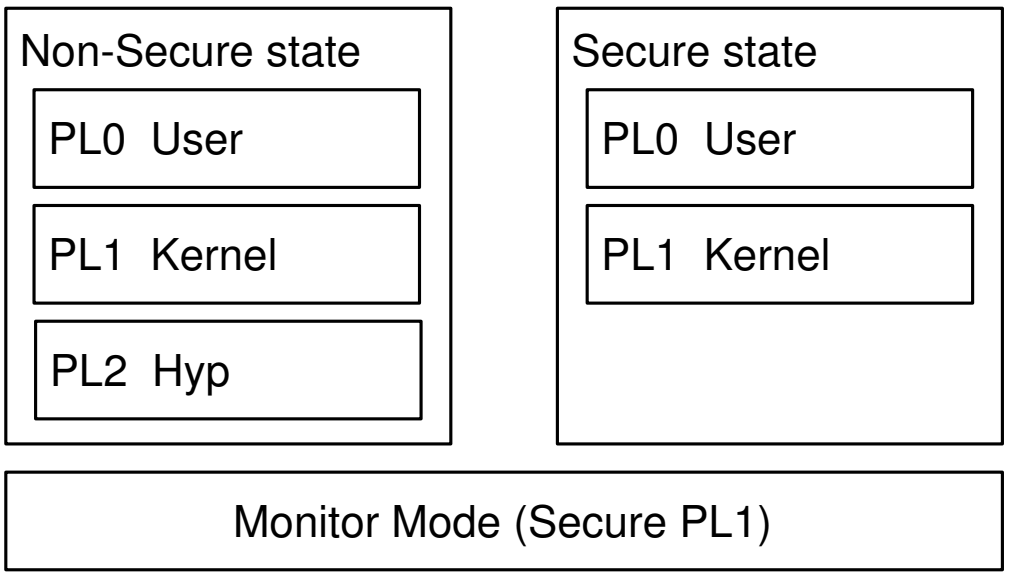
ARM TrustZone

- TrustZone/Secure Monitor does **not** provide virtualization
  - No support for trap-and-emulate logic
  - Secure boot, chain of trust
  - Memory isolation from a rich OS
  - Running trusted/proprietary software (e.g. Digital Rights Management)

ARM Virtualization Extensions

- Extensions for CPU and MMU virtualization
- Implemented in Cortex-A15, Cortex-A7
- New processor mode: *Hyp mode* (PL3)

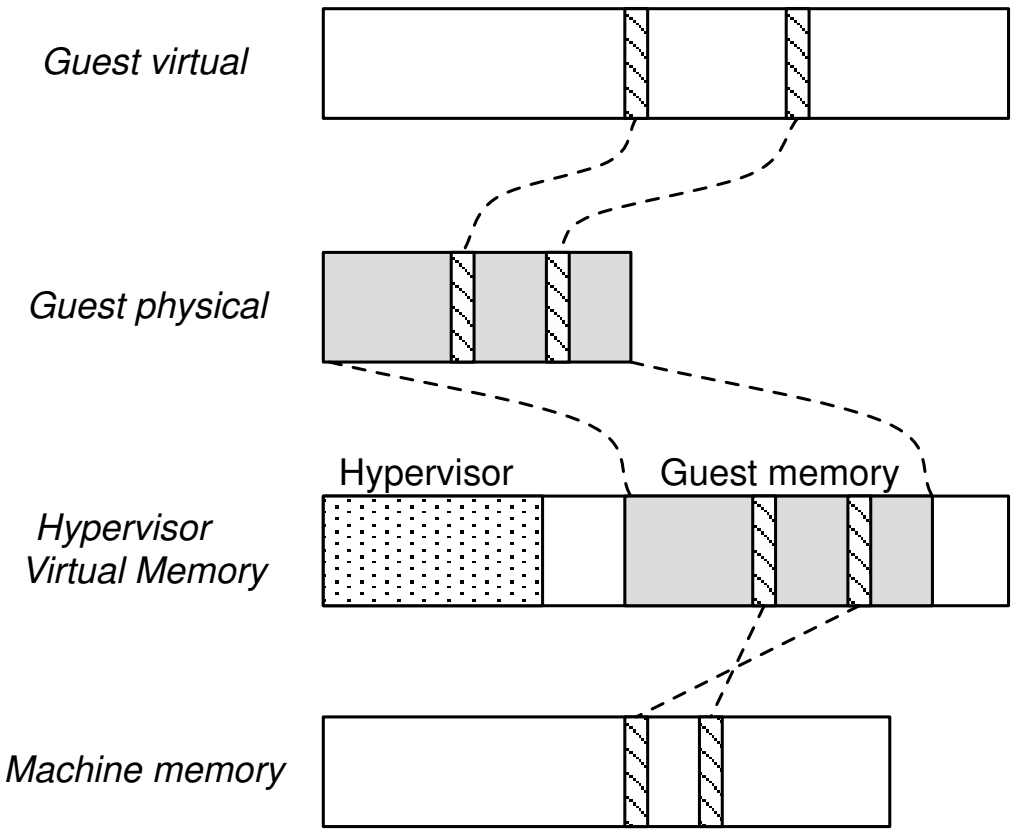
ARM VE: CPU virtualization



ARM VE: CPU virtualization (2)

- Traps are configurable
- Sensitive instructions may trap to
  - Supervisor mode
  - Hyp mode
- Hyp mode has a reduced number of registers compared to Supervisor mode

ARM VE: Memory virtualization



ARM VE: Memory virtualization (2)

MMU supports two stages of translation:

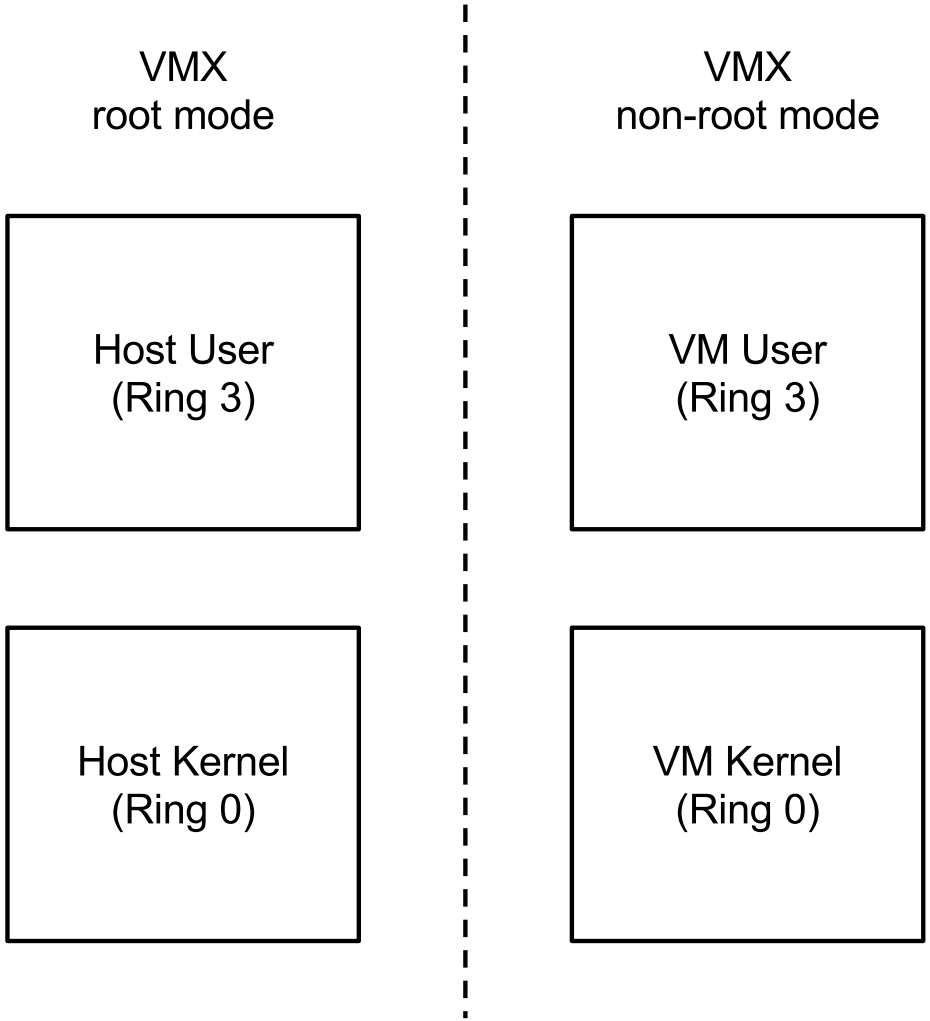
- Stage 1: Virtual (VA) to Intermediate Physical (IPA)
  - Managed by guest OS
- Stage 2: Intermediate Physical (IPA) to Host Physical (PA)
  - Managed by hypervisor

ARM VE: I/O virtualization

- Interrupts
  - Generic Interrupt Controller (GIC)
  - VGIC
  - Memory-mapped interface (MMIO)
  - Inter-Processor Interrupts (IPI)
- Timers
  - Generic Timer Architecture
  - Virtual counter, virtual timer

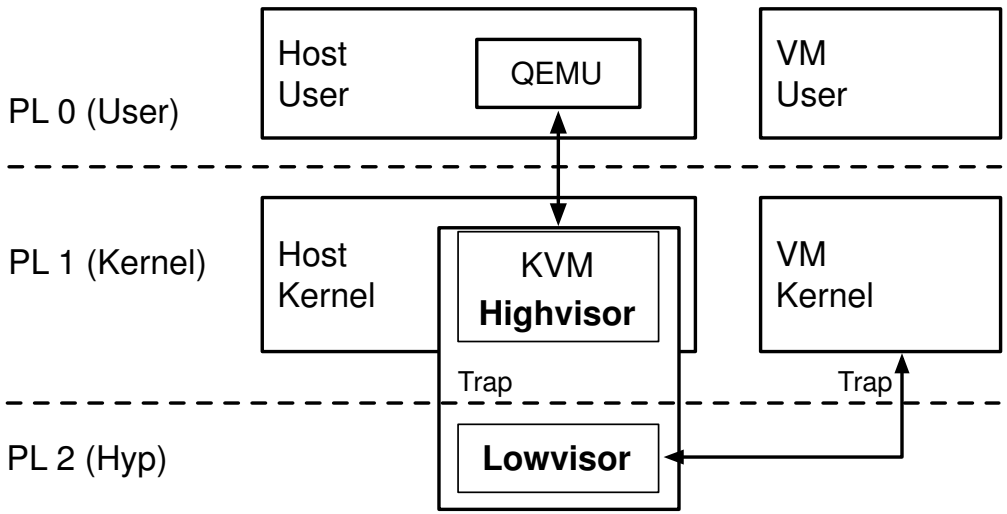
Outline

Intel VT-x: CPU virtualization



KVM on ARM: Challenges

- KVM architecture is **strongly** dependent on Linux/x86
- Difficult to migrate Linux to Hyp mode
  - Modify a lot of code
  - Maintain separate branch
  - Don't get included in upstream
- Hyp mode lacks the features of VMX root mode
  - Difficult run an entire OS there



- Factor KVM into **Highvisor** and **Lowvisor**

Split-mode virtualization

- Lowvisor
  - Switches between Host (Highvisor) and VMs
- Highvisor
  - Leverages KVM/Linux's mechanisms
  - Manages both its page tables and the Lowvisor's
- **Note**: trapping to the Highvisor requires double-switching

Outline

World switches

- VMs and the Host/Highvisor run at the same level of privilege
- All world switches are intermediated by the Lowvisor
- The Host controls switching policy

VM and Host state

Action	Nr.	State
Context Switch	38	General Purpose (GP) Registers
	26	Control Registers
	16	VGIC Control Registers
	4	VGIC List Registers
	2	Arch. Timer Control Registers
	32	64-bit VFP registers
	4	32-bit VFP Control Registers
Trap-and-Emulate	-	CP14 Trace Registers
	-	WFI Instructions
	-	SMC Instructions
	-	ACTLR Access
	-	Cache ops. by Set/Way
	-	L2CTLR / L2ECTLR Registers

Switching from Host to VM

1. Store Host state
2. Configure and load VM state
3. Enable traps from Supervisor to Hyp
4. Enable Stage 2 translation
5. Trap to Supervisor or User

Switching from VM to Host

1. Store VM state
2. Configure and load VM state
3. Disable (most) traps from Supervisor to Hyp
4. Disable Stage 2 translation
5. Trap to Host

Outline

Memory virtualization

- Provided by enabling Stage 2 translation
- Stage 2 page tables are managed by the Highvisor
- Stage 2 translation is enabled/disabled by the Lowvisor

Outline

I/O virtualization

- Direct access to I/O memory is disabled
  - Accesses by the VM will result in Stage 2 pagefaults
  - ... and trap to the hypervisor
  - Device access policy is controlled by QEMU
- Pass-through devices are an exception

Interrupt virtualization

- Interrupts are emulated through VGIC
- VGIC state is saved/restore on context switches
- GIC distributor accesses will trap to the hypervisor
  - Virtual distributor routes IPIs

Timer virtualization

- Uses virtual timers
- Only Hyp mode can access physical timers
- Virtual timer interrupts trap to hypervisor
  - Hypervisor forwards interrupts to VMs
  - Hypervisor performs ACK and EOI operations
- Per-CPU timers are multiplexed using the Host's software timers

Outline

Keywords

- virtualization
- hardware-assisted virtualization
- hypervisor
- virtual machine monitor
- split-mode virtualization
- highvisor
- lowvisor
- world switch

Resources

- http://landley.net/kdocs/ols/2010/ols2010-pages-45-56.pdf
- http://systems.cs.columbia.edu/files/wpid-asplos2014-kvm.pdf
- http://systems.cs.columbia.edu/projects/kvm-arm/
- http://www.linux-kvm.org/page/Status
- http://www.linux-kvm.org/page/Main_Page

Outline

Questions

?