Hypervisors on ARM
Overview and Design choices

ARM

Julien Grall <julien.grall@arm.com>

Root Linux Conference 2017

About me

- Working on ARM virtualization for the past 4 years
- With ARM since 2016
- Co-maintaining Xen on ARM - with Stefano Stabellini [Aporeto]

Virtualization, what is it?

Virtualization refers to the act of creating a virtual version of something

Wikipedia

Use cases

Consumer Electronics

Industrial

Automotive

Enterprise

Type of hypervisors

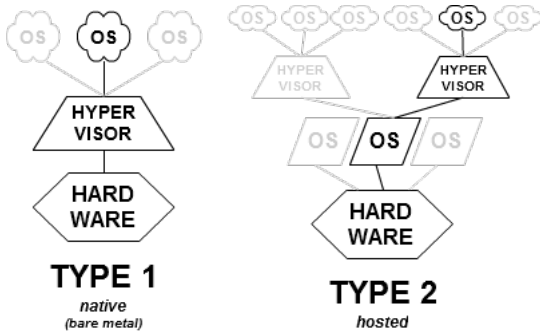


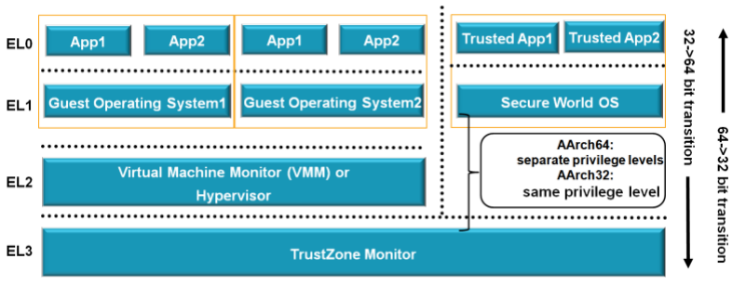
Figure: From wikipedia

Kind of virtualization

- Full hardware virtualization
  - OS is running unmodified
  - Guest I/O are either
    - emulated
    - handled by virtualization-aware hardware
- Para-virtualization
  - OS is aware of the hypervisor
  - Privilege instruction are replaced by hooks
  - Devices (network, block...) are para-virtualized
- The trend is a mix of both
  - Use as much as possible hardware-assisted virtualization
  - Devices (network, block...) para-virtualized or passthrough-ed
  - Emulation very limited

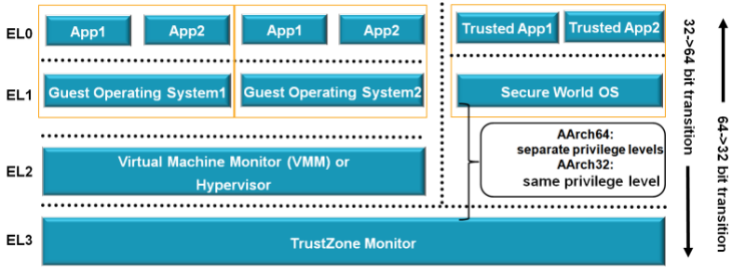
ARM virtualization

ARMv8-A Privilege Model



- Support both AArch32 and AArch64 execution modes
- 32-64bit inter-working limited to exception boundaries
- AArch64 always has a higher privilege than AArch32
- AArch64 state is a superset of AArch32 state

ARM virtualization



- Introduced with the latest version of ARMv7 architecture
- New hypervisor execution state
- Non-Secure world, higher privilege than EL1

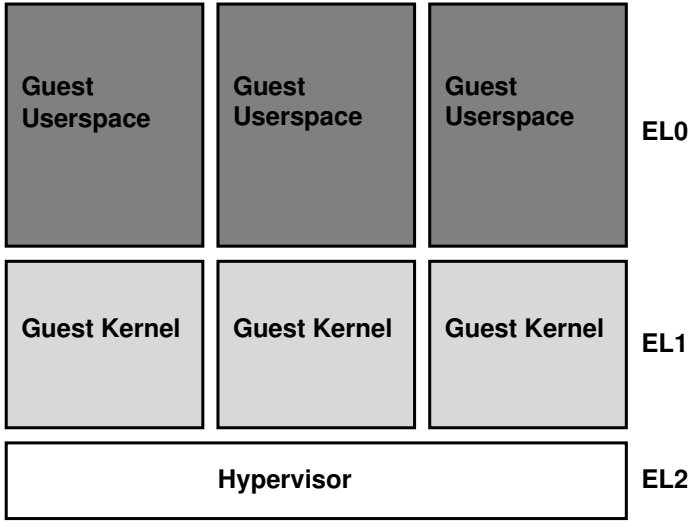
Virtualization in a nutshell

- Second stage of memory translation
  - Adds an extra level of indirection between guests and physical memory
  - TLBs are tagged by Virtual Machine ID (VMID)
- Ability to trap access of most system registers
  - The hypervisor decides what it wants to trap
- Can handle IRQs, FIQs and asynchronous aborts
  - The guest doesn't see physical interrupts firing, for example
- Guests can call into EL2 mode (HVC instruction)
  - Allows para-virtualizated services
- Standard architecture peripherals are virtualization-aware
  - GIC and timer have specific features to help virtualization

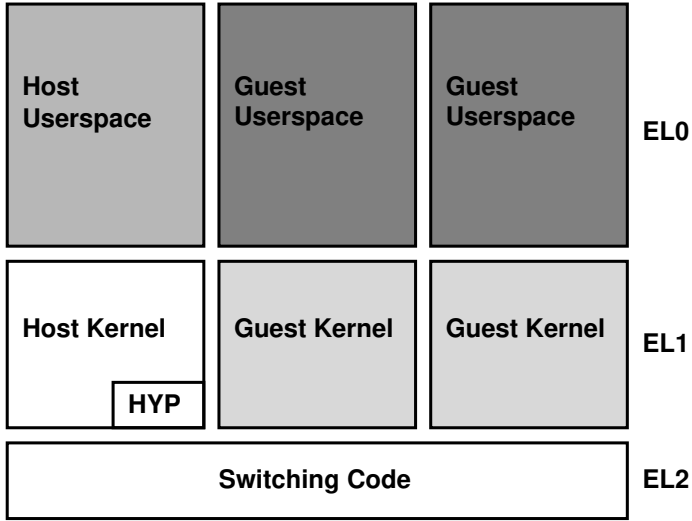
EL2: Not EL1++ (ARMv8.0-A)

- EL2 is not a superset of NS-EL1
  - Orthogonal mode to EL1
  - Allows multiplexing of NS-EL1 guests on the hardware
- Own translation regime
  - Separate Stage-1 translation, no Stage-2 translation
- It would be difficult to run Linux in EL2
  - Requires too many changes to be practical
- EL2 could be used as a ”world switch”
  - Between guests (barametal hypervisor/Type I)
  - Between host and guest (hosted hypervisor/Type II)
    This makes the host a form of specialized guest.

Hypervisor architecture - Type I



Hypervisor architecture - Type II



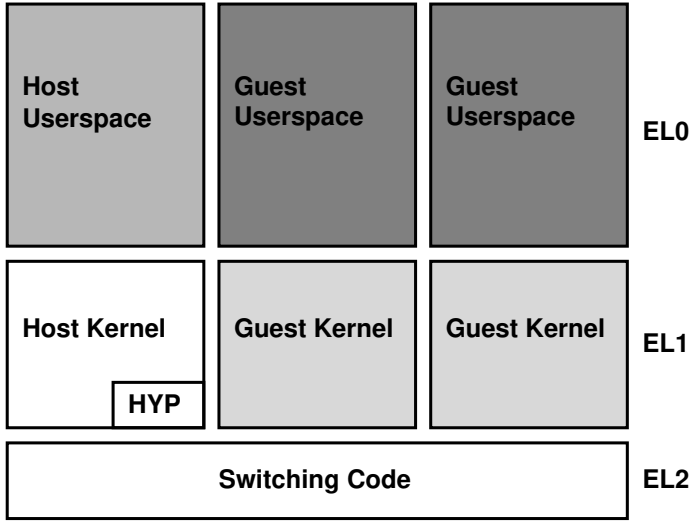
EL2 enhancement (ARMv8.1-A)

The Virtualization Host Extension (VHE) expands the capability of EL2:

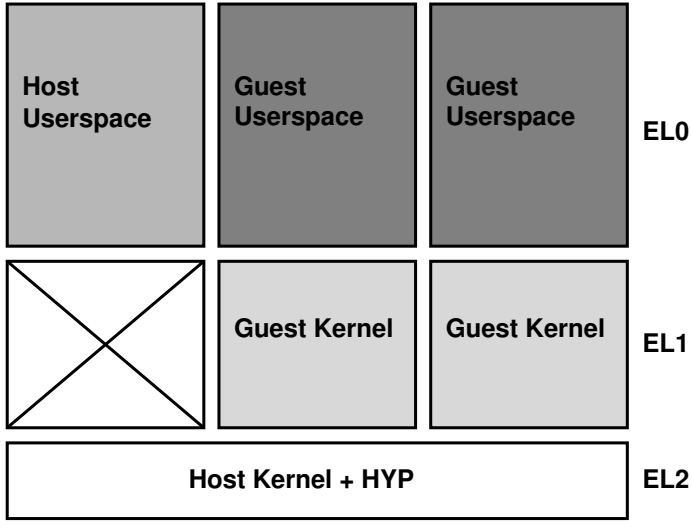
- Designed to improve the support of the Type-2 hypervisors
- Allows the host OS to be run at EL2
- The host OS requires minimal changes to run at EL2
- User-space still runs at EL0
- Host has no software running at EL1
- AArch64 specific

EL2 becomes a strict superset of EL1

Hosted hypervisor architecture on platform without VHE



Hosted hypervisor architecture on platform with VHE



Nested Virtualization (ARMv8.3-A)

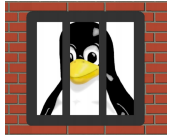
The Nested Virtualization extension allows an hypervisor in a VM.

- Unmodified guest hypervisor running in NS EL1
- Implementation of a host hypervisor required
  - Running at EL2
- AArch64 specific

Why using ARM virtualization

- Robust set of virtualization features
  - Not just about CPU virtualization
  - Covers the whole systems architecture
- Scalable architecture
  - Power to IoT-like devices ...
  - ... all the way to server-grade systems
- An architecture in motion:
  - ARMv8.1-A: `https://goo.gl/Ox4thV`
  - ARMv8.2-A: `https://goo.gl/0Ns37U`
  - ARMv8.3-A: `https://goo.gl/CJv1n0`

OpenSource Hypervisors



KVM

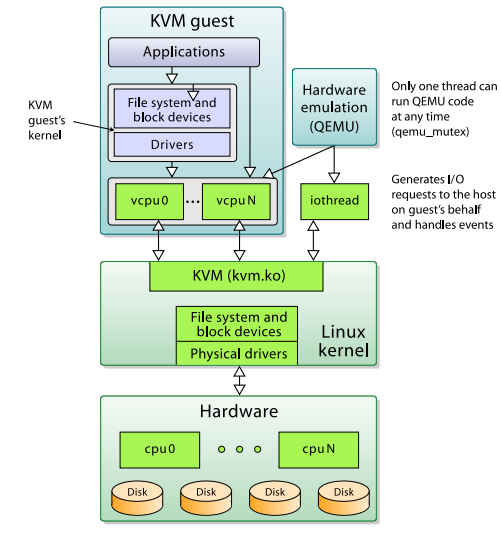
Kernel-based Virtual Machine

- First version of KVM was merged in Linux 2.6.20
  - AArch32 support merged in Linux 3.9
  - AArch64 support merged in Linux 3.11
- Source code available as GLP v2
- Hosted hypervisor

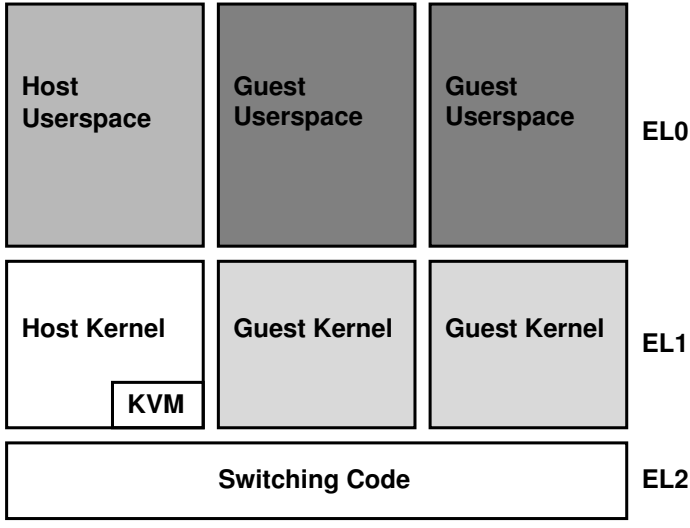
KVM virtual machine

- Use of assisted hardware virtualization
- Devices are
  - emulated (QEMU)
  - para-virtualized (VIRTIO)

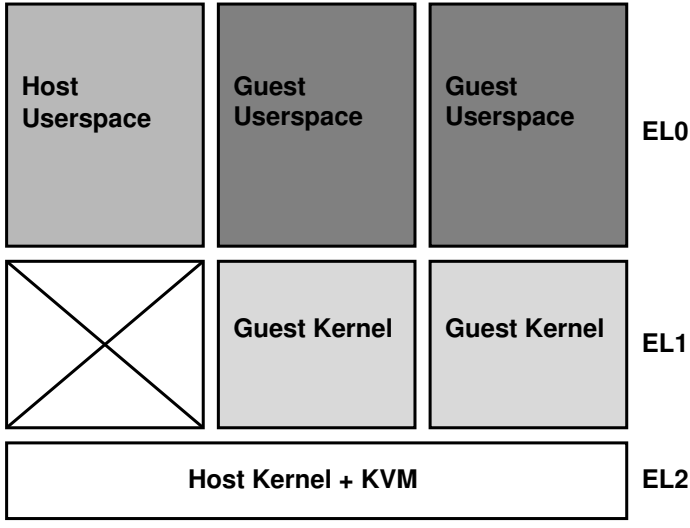
KVM architecture



KVM architecture with ARMv8.0-A



KVM architecture with ARMv8.1-A



Resource management

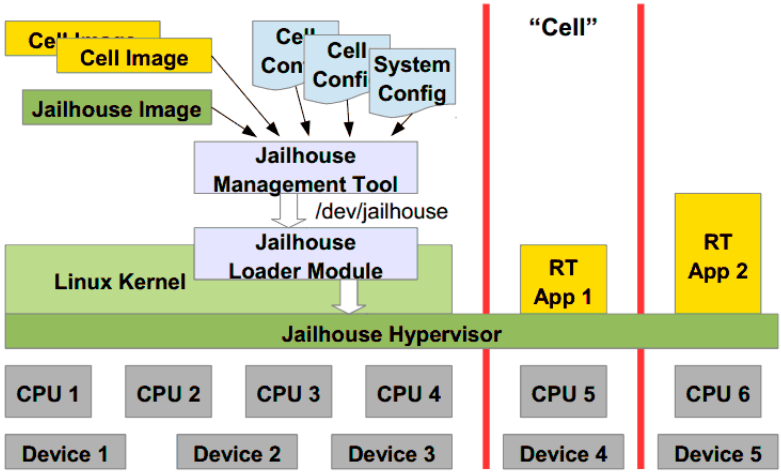
- All CPUs are using the same scheduler
- guest vCPU is a task for the host OS
- Resource management can be done using cgroup
  - Standard way in Linux to control resources

Jailhouse

Jailhouse

- Created at Siemens in 2013
- Partioning hypervisor
  - Type-I hypervisor
  - Linux will load Jailhouse
- Source code available as GPL v2
- Small code base: <10K lines

Jailhouse architecture

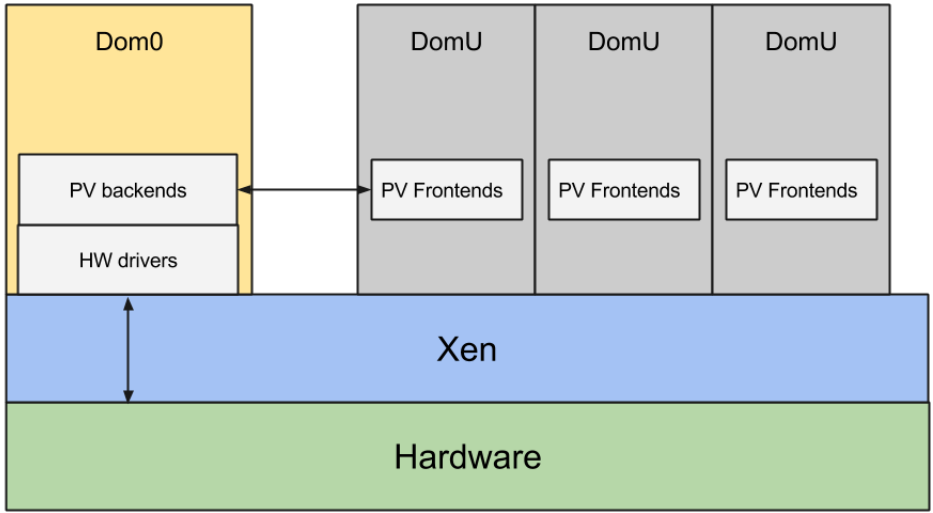


Xen

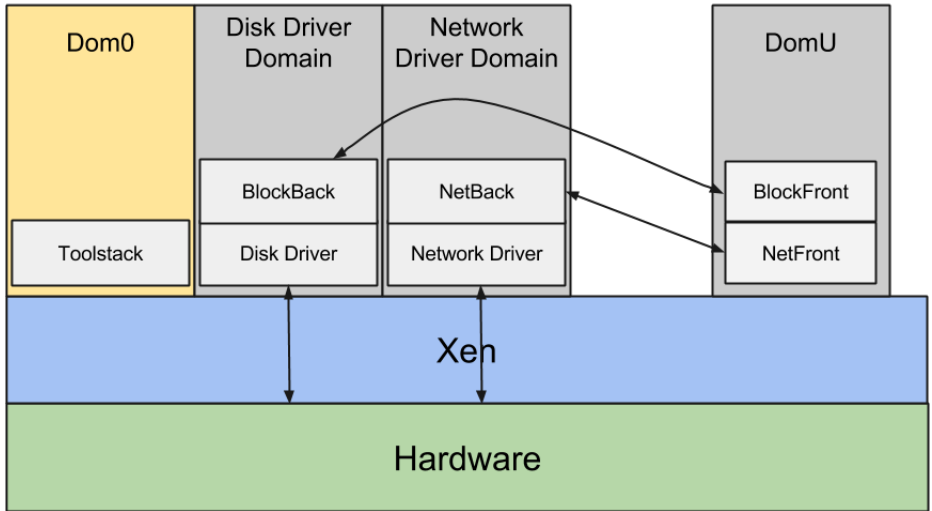
Xen

- First released in 2003
  - ARM officially supported since Xen 4.4
- Source code available as GPL v2
- Small code base: 30K
- Bare-metal hypervisor

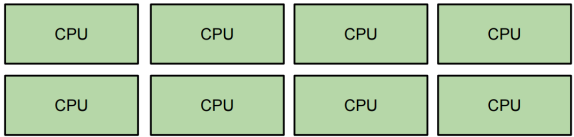
Xen architecture



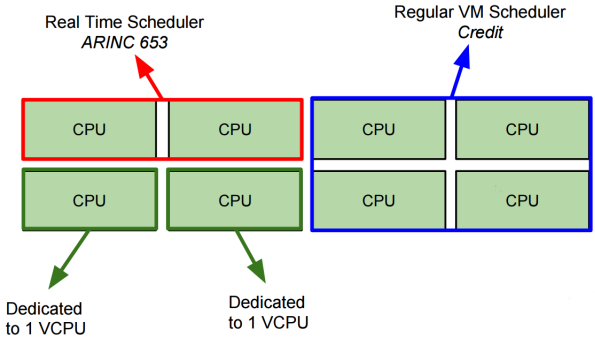
Xen architecture - 2



Xen schedulers

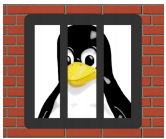


Xen schedulers



Summary

https://www.linux-kvm.org/



https://github.com/siemens/jailhouse

https://xenproject.org/

Questions?

ARM