Xen Hypervisor: A brief look

Apps and user stuff! →
Multiple workloads, in different OSes.
Resources isolated and protected.

← Platform stuff!
Hypervisor, tools, control domain,
device drivers, management,
host storage and networking
↓

Xen hypervisor

Scheduler, Memory Mgmt, XSM Access Control
Xen Hypervisor: A brief look

Dom0
- Xen Tools
- Linux Kernel

User Interface
- Linux Kernel

Network Driver Domain
- Network Management
- Linux Kernel
- NIC device driver

Windows VM
- Disk PV Driver
- Net PV Driver

Linux VM
- Disk PV Driver
- Net PV Driver

Xen hypervisor
Scheduler, Memory Mgmt, XSM Access Control

Powerful. Flexible. Open.
Yes, but can I run it on my®?
hmm, let’s see about that...
Bitbaking Xen for the Raspberry Pi 4

Our essential basic ingredients are in layers:

- Xen support is in meta-virtualization [source, list]
- Raspberry Pi 4 support is in meta-raspberrypi [source]
- We’ll use poky with meta-openembedded

Add in a fresh zesty sprinkle of:

- a brand-new Xen-on-Raspberry Pi 4 patch series for meta-virtualization

... along with some classic Yocto spiciness:

- configure your local.conf:

```
MACHINE="raspberrypi4-64"
DISTRO_FEATURES_append = " virtualization xen"
QEMU_TARGETS = "i386 x86_64 aarch64 arm"
```

"Bake!": `bitbake xen-image-minimal` → SD-card image!
The card image contains two partitions:

- **boot**, which includes:
  - `xen`: hypervisor binary
  - `Image`: Linux kernel binary
  - `bcm2711-rpi-4-b.dtb`: Device tree for the Raspberry Pi 4
  - `overlays`: Device tree overlays
  - `boot.scr`: Xen-specific u-boot launch script
  - `config.txt`: Raspberry Pi configuration settings
  - ...

- **root** filesystem for Domain 0
  - `poky Linux filesystem`
  - contains the familiar Xen tools
  - *does not contain* the hypervisor or the dom0 Linux kernel
Xen Hypervisor on Raspberry Pi 4

Insert the SD card, fire up the power, ... stand well back...
Xen Hypervisor on Raspberry Pi 4

Hooray!
Xen Hypervisor on Raspberry Pi 4

ok, proof that it is actually there and working
Xen Hypervisor on Raspberry Pi 4: serial

To obtain for yourself a luxury device such as this, the terms you can use to proceed are: “pl2303hx USB serial”
Xen Hypervisor on Raspberry Pi 4: serial

OK - time to wire it up! There are three wires to connect:

- Pins indicated red here: *nope, do not need those for this.*
- This one: **good**, you want it: that's **ground**: connect it to the black single wire on the USB serial thing
- These two: **also good**: RX and TX
  There's probably some way to get them the right way around, but: **don't use the red wire** off the USB thing, and then you can just try the other two either way and one way will work: **bingo!**

...there are loads more pins down here too, I just didn't draw them all
Xen Hypervisor on Raspberry Pi 4: serial

On a nearby machine with the USB serial device plugged in: minicom /dev/ttyUSB3
Xen for the Raspberry Pi 4: simple Xen commands

Testing basic Xen functionality at the console:

- `xl list` - list running VMs
- `xl info` - show data about the current hypervisor
- `ls -l /dev/xen` - examine Linux’s Xen device nodes
- `xenstore-ls` - read the contents of the XenStore tree
- `dmesg | grep Xen` - see the Linux kernel messages relating to Xen
- `xl dmesg` - see the Xen boot messages

All these should be familiar if you’ve used Xen on other systems - and now available on the Raspberry Pi 4!
Xen for the Raspberry Pi 4: building a guest VM

Let’s boot Yocto Linux inside Yocto Linux! First, build the guest filesystem image:

- `bitbake xen-guest-image-minimal`

Install the pieces needed to run a guest into the running Domain-0 of the Pi:

- **Copy in the built guest filesystem in a file:**
  - `../work/raspberrypi4_64-poky-linux/xen-guest-image-minimal/*/deploy-xen-guest-image-minimal-image-complete/xen-guest-image-minimal-raspberrypi4-64.ext3` to: `/home/root/xen-guest-image-minimal-raspberrypi4-64.ext3`

- **Copy in the guest kernel file:** *Image*
  - `../work/raspberrypi4_64-poky-linux/linux-raspberrypi/*/deploy-linux-raspberrypi/Image` to: `/home/root/Image`

- **Create a new file:** *guest.cfg*
  - `kernel = "/home/root/Image"
cmdline = "console=hvc0 earlyprintk=xen sync_console root=/dev/xvda"
memory = "256"
name = "rpi4-xen-guest"
vcpus = 1
serial="pty"
disk = [ 'phy:/dev/loop0,xvda,w' ]
vif=[ 'mac=00:11:22:66:88:22.bridge=xenbr0.type=netfront' ]`
Xen for the Raspberry Pi 4: prepare for a guest VM

Networking - so the guest can get its own network access:

- Create a bridge and move the eth0 physical device onto it:
  - killall -SIGUSR2 udhcpc # release your existing DHCP lease
  - brctl addbr xenbr0 # create a new bridge called “xenbr0”
  - brctl addif xenbr0 eth0 # put eth0 onto xenbr0
  - killall udhcpc # terminate the DHCP client daemon
  - udhcpc -R -b -p /var/run/udhcpc.xenbr0.pid -i xenbr0 # restart the DHCP client daemon on the new bridge

Disk:

- Loopback mount the ext3 guest filesystem file to make it available as a device
  - losetup /dev/loop0 /home/root/xen-guest-image-minimal-raspberrypi4-64.ext3
Xen for the Raspberry Pi 4: running a guest VM

xl create -c guest.cfg
Xen for the Raspberry Pi 4: the current patch series

OK, so what’s in this new meta-virtualization patch series to make this work?

- A new “dynamic layer” for settings that are specific to Xen-on-Raspberry-Pi-4
  - A series of Linux kernel patches: DMA fixes from the Xen ARM maintainer
  - Enabling the Yocto kernel cache with the Raspberry Pi kernel to build with meta-virtualization
  - Enabling the hardware interrupt controller that Xen needs in the rpi-config
  - Custom Xen-specific bootloader script:
    - Loads the Xen hypervisor binary
    - Loads the Linux kernel
    - Amends the device tree, that the Raspberry Pi has already processed before u-boot
    - Sets the Dom0 kernel command line, to include Xen config settings eg. for the console
  - A Xen hypervisor “defconfig”, with settings specific for the Raspberry Pi 4 hardware
  - A Xen-specific SD-card class, to include the Xen binary on the first partition of the image
- A new ARM assembly Xen patch to implement an atomic primitive for spinlocks with the latest gcc in Yocto
- Xen version upgrade to 4.13 (and 4.14 should be coming soon)
- A new method of engaging Xen-specific config settings when DISTRO_FEATURES includes ‘xen’
Xen in meta-virtualization: beyond the Pi!

Other example platforms for running Xen with Yocto, using meta-virtualization:

- **Intel x86-64**: ubiquitous!
  - meta-virtualization has wic tool integration
    - enables simple production of a bootable image:
      - wic create directdisk-xen -e xen-image-minimal
      - dd the output file to your hard disk and boot it!

- **PCEngines APU2**: low power, low cost (~$100), very open hardware
  - open hardware schematics, has coreboot support
  - add meta-pcengines and set MACHINE = “pcengines-apu2”
    then use the wic image - dd it to a drive and boot into Xen
  - hardware supports D-RTM - that’s a big deal: see the OpenXT community for more!

- **runqemu**
  - runqemu xen-image-minimal nographic slirp
    - launch Xen at your command prompt!
      - nb: is currently pretty experimental - has worked with MACHINE = “genericx86-64”; may need some work.
Thanks

- **Xen Community**
  - for the Xen hypervisor and Linux kernel work to make this possible
    - Stefano Stabellini @ Xilinx, Julien Grall @ Amazon
  - for the interest in the Raspberry Pi 4
    - Roman Shaposhnik @ Eve Project, Zededa
  - hey! See you at the [Xen Design and Developer Summit](#) next week!

- **Yocto and OpenEmbedded meta-virtualization Community**
  - for the first Xen on Raspberry Pi 4 meta-virtualization patch submissions
    - Corey Minyard @ MontaVista, Stewart Hildebrand @ DornerWorks
  - for the support for bringing Xen work in, in a maintainable way
    - Bruce Ashfield @ Xilinx, Bertrand Marquis @ ARM
  - for this talk opportunity!
    - Tim Orling @ Yocto Project, Intel, David Reyna @ Wind River

- **OpenXT Community**
  - for supporting my involvement with Xen and OpenEmbedded

- **Raspberry Pi Community**
  - for developing and promoting accessible hardware with Open Source software