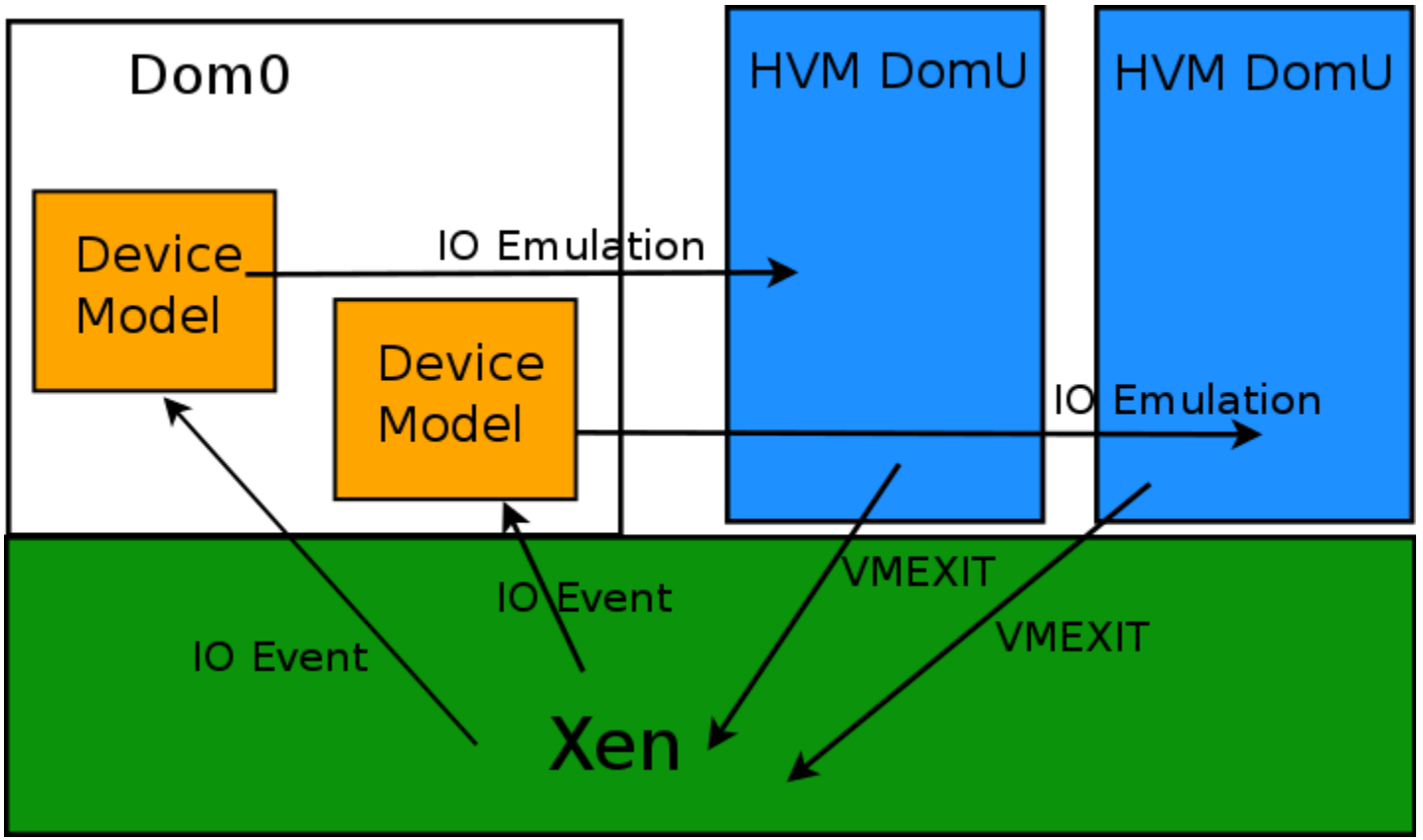
Linux based Stub-domain

why, how and benchmark

Anthony PERARD
@ Citrix

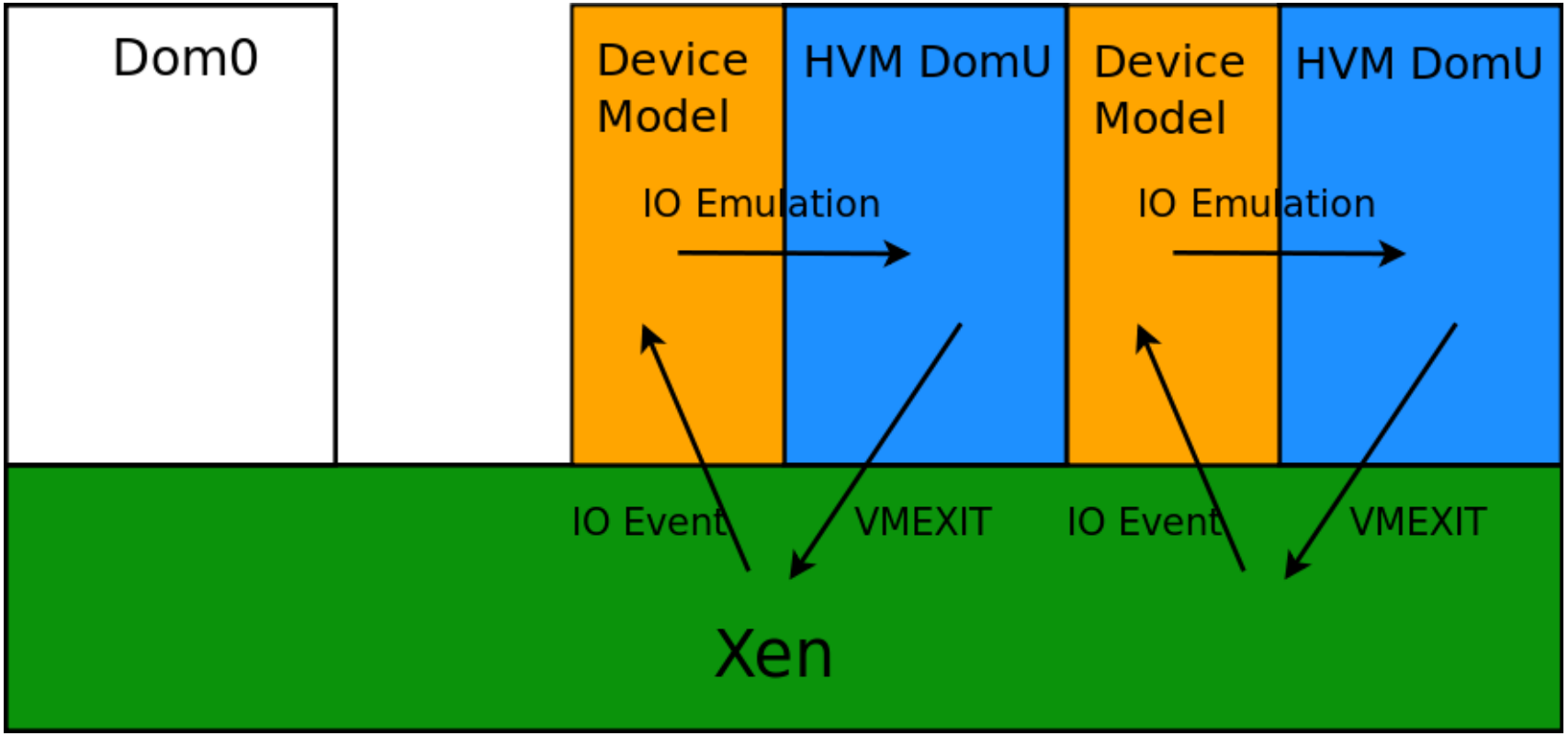
"Normal" case

- Device Model (QEMU) in dom0



Solution: stub-domain

- The Device Model in its own domain



How to do it?

- Current one based on mini-os
  - use newlib
- But QEMU upstream need more
- So, two solutions:
  - Port another libc to mini-os
  - Linux based stubdom

What do we need?

- Change in Linux
  - Need to use dom0 privileged command (memory mapping)
- QEMU
  - set_hvm_param DM_DOMAIN
  - no backend initialisation
- libXenLight
- initramfs

Status

- Can start a domain with both console and network
- Stubdom memory: 40MB

TODO

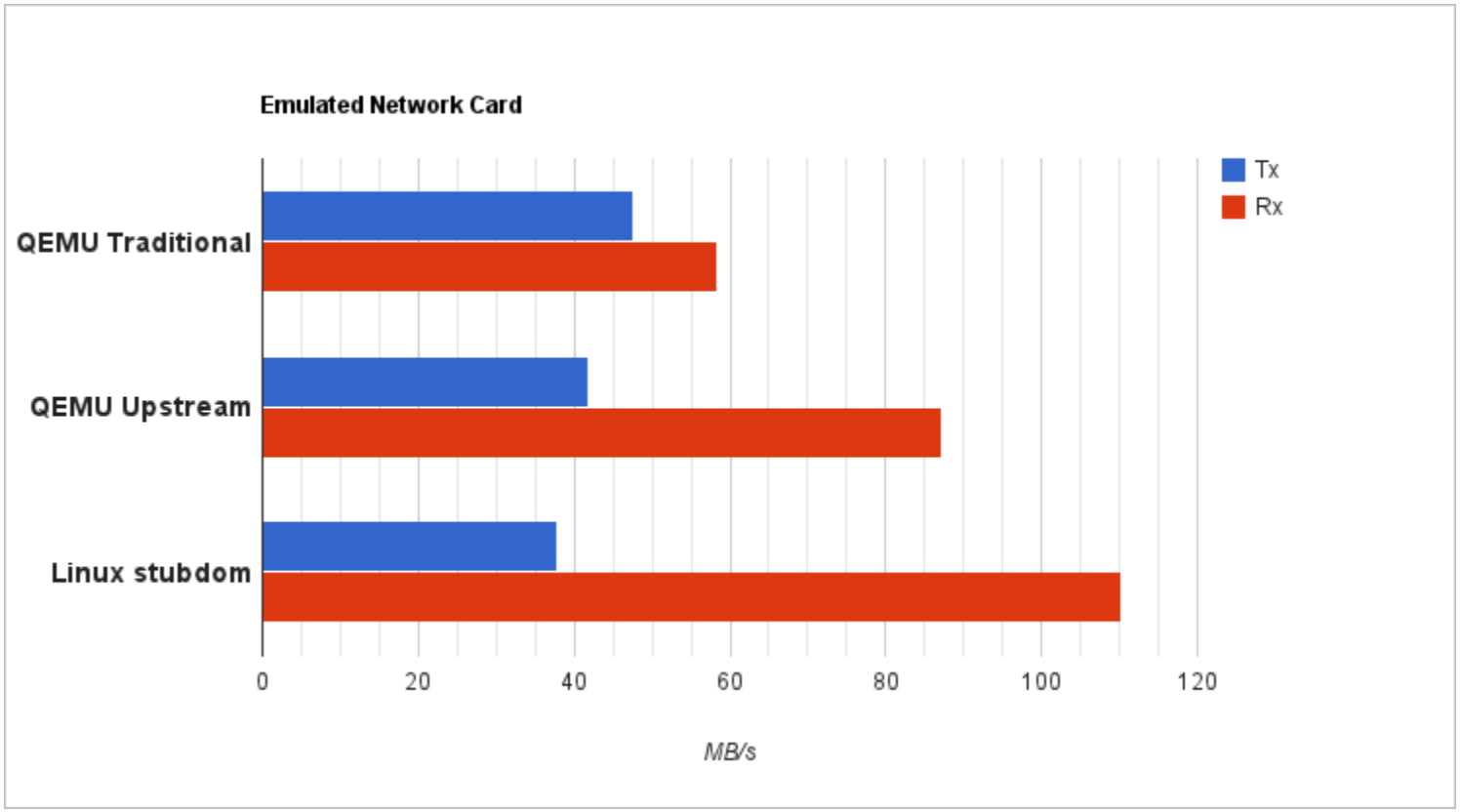
- upstream those patches
- integrate it in Xen build system
- have a video working
- could reduce the size of the stubdom

Lies, damned lies, and benchmarks

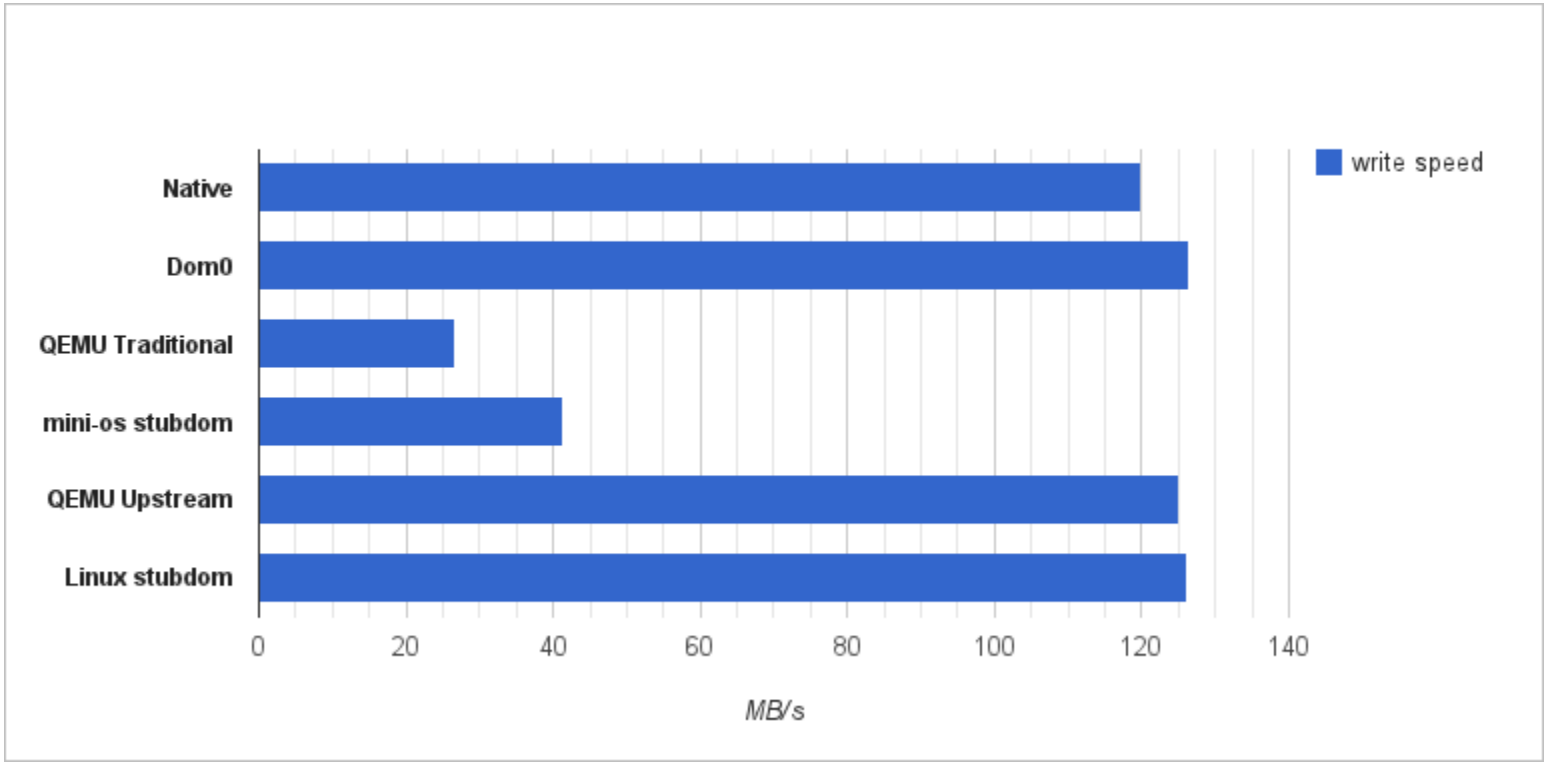
- With 8 CPU AMD Opteron, 8GB
- dom0: Arch Linux 64bit
- domU: Arch Linux 64bit, 2GB of RAM
- (kernel Linux 3.4.8)

- iperf for network
- dd for disk

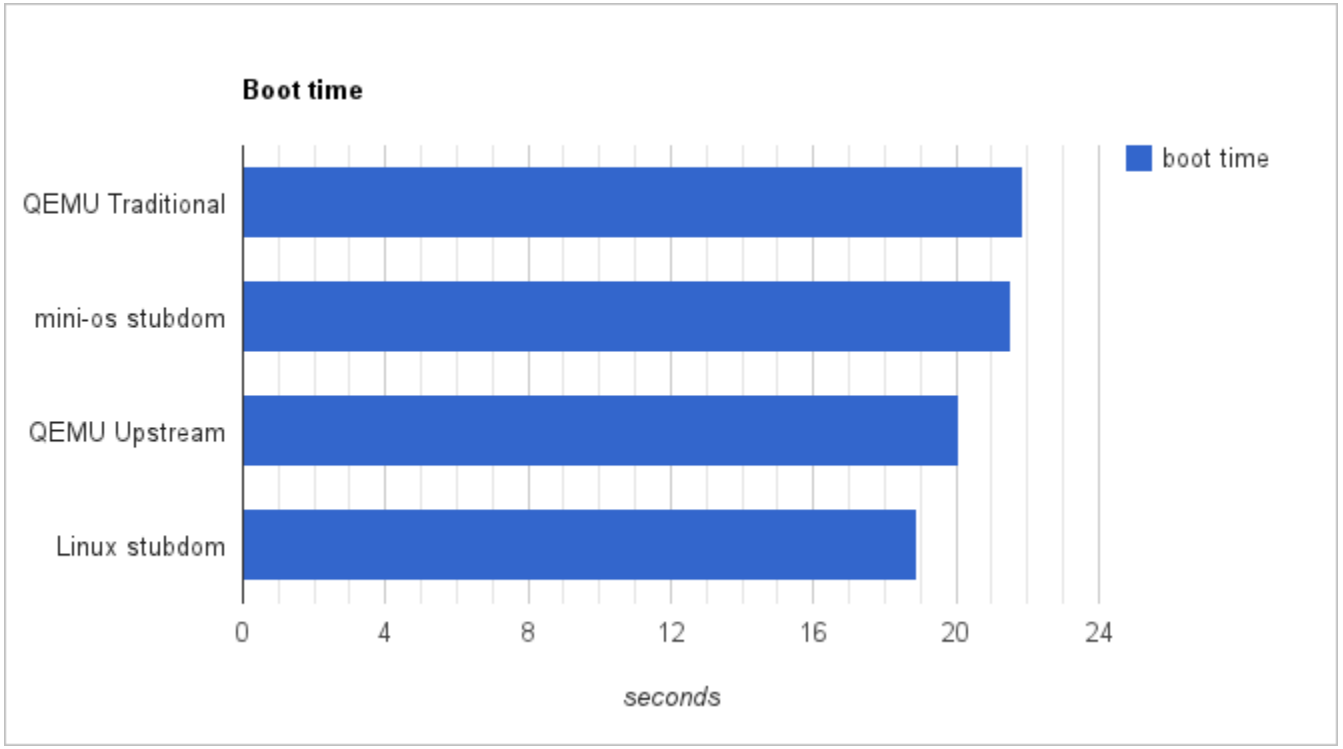
benchmark - Network



benchmark - Disk



benchmark - Boot time



Between `xl create vm` and `ssh guest true`

Conclusion

- No more competition, priority inversion in dom0
- Extra layer against security vulnerability in QEMU
- Little impact on performance
- Should it be the default?

Question ?