Xen virtualization on FreeBSD

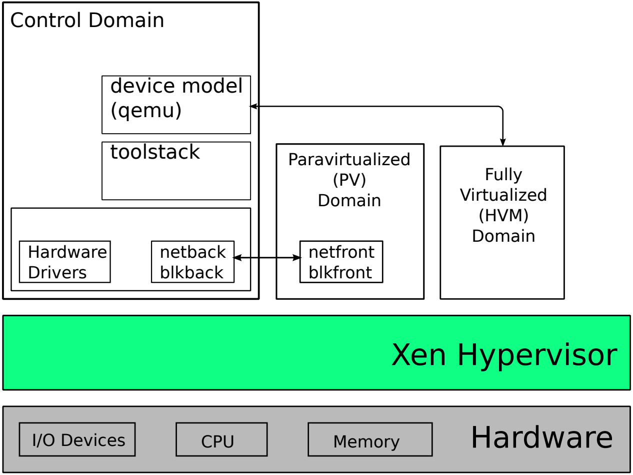
Roger Pau Monné

Tokyo – March 12, 2015

Goals of this presentation

- Description of Xen.
- A peek into Xen's new features.
- Recent work done in FreeBSD to improve Xen support.
- Demo of a FreeBSD/Xen Dom0.

Xen Architecture



Paravirtualization

- Virtualization technique developed in the late 90s.
- Designed by:
  - XenoServer research project at Cambridge University.
  - Intel.
  - Microsoft labs.
- x86 instructions behave differently in kernel or user mode, options for virtualization were full software emulation or binary translation.
  - Design a new interface for virtualization.
  - Allow guests to collaborate in virtualization.
  - Provide new interfaces for virtualized guests that allow to reduce the overhead of virtualization.
- The result of this work is what we know today as paravirtualiztion.

Paravirtualization

- All this changes lead to the following interfaces being paravirtualized:
  - Disk and network interfaces
  - Interrupts and timers
  - Boot directly in the mode the kernel wishes to run (32 or 64bits)
  - Page tables
  - Privileged instructions

Full virtualization

- With the introduction of hardware virtualization extensions Xen is able to run unmodified guests
- This requires emulated devices, which are handled by Qemu
- Makes use of nested page tables when available.
- Allows to use PV interfaces if guest has support for them.

The virtualization spectrum

VS	Software virtualization
VH	Hardware virtualization
PV	Paravirtualized

🟥	Poor performance
🟨	Room for improvement
🟩	Optimal performance

	Disk and network	Interrupts and timers	Emulated motherboard	Privileged instructions and page tables
HVM	🟥 VS	🟥 VS	🟨 VS	🟩 VH
HVM with PV drivers	🟩 PV	🟥 VS	🟨 VS	🟩 VH
PVHVM	🟩 PV	🟩 PV	🟨 VS	🟩 VH
PV	🟩 PV	🟩 PV	🟩 PV	🟨 PV

Xen's new features

- Recent Xen changes:
  - Improved support for running Xen on ARM.
  - New virtualization mode: PVH.
  - As usual, improvements/bugfixes across all components.

Xen on ARM

- Started on 2011, focused on bringing Xen into ARM boards with virtualization extensions.
- Xen 4.5 is the recommended release for Xen on ARM.
- Has support for both 32 and 64bit ARM chips.
- More information can be found at `http://www.xenproject.org/developers/teams/arm-hypervisor.html`.

New x86 virtualization mode: PVH

- PV in an HVM container.
- PVH should use the best aspects from both PV and HVM:
  - No need for any emulation.
  - Has a ”native” MMU from guest point of view.
  - Has access to the same protection levels as bare metal.
- Written by Mukesh Rathor @ Oracle.
- Significant revisions by George Dunlap @ Citrix.

The extended virtualization spectrum

VS	Software virtualization
VH	Hardware virtualization
PV	Paravirtualized

🟥	Poor performance
🟨	Room for improvement
🟩	Optimal performance

	Disk and network	Interrupts and timers	Emulated motherboard	Privileged instructions and page tables
HVM	🟥 VS	🟥 VS	🟨 VS	🟩 VH
HVM with PV drivers	🟩 PV	🟥 VS	🟨 VS	🟩 VH
PVHVM	🟩 PV	🟩 PV	🟨 VS	🟩 VH
PVH	🟩 PV	🟩 PV	🟩 PV	🟩 VH
PV	🟩 PV	🟩 PV	🟩 PV	🟨 PV

PVH technical overview

- Runs inside of an HVM container.
  - No PV MMU.
  - Runs with normal privilege levels.
- Disable HVM emulated devices.
- Uses PV start sequence.
  - Start with basic paging setup.
- Uses the PV path for several operations:
  - vCPU bringup.
  - PV hypercalls.
  - PV e820 memory map.
- Uses the PVHVM callback mechanism.

FreeBSD 9.x Xen support

- i386 PV port.
- HVM with PV drivers (both i386 and amd64).
  - Xenstore and grant-table implementations.
  - Event channel support.
  - PV Disk and Network front and backends.
  - Suspend and resume.

FreeBSD 10.x Xen support

- PVHVM.
  - Vector callback support.
  - Unified event channel code with the i386 PV port.
  - PV timer.
  - PV IPIs.
  - PV Suspend and resume.

Ongoing work in HEAD

- PVH DomU support.
- PVH Dom0 support.

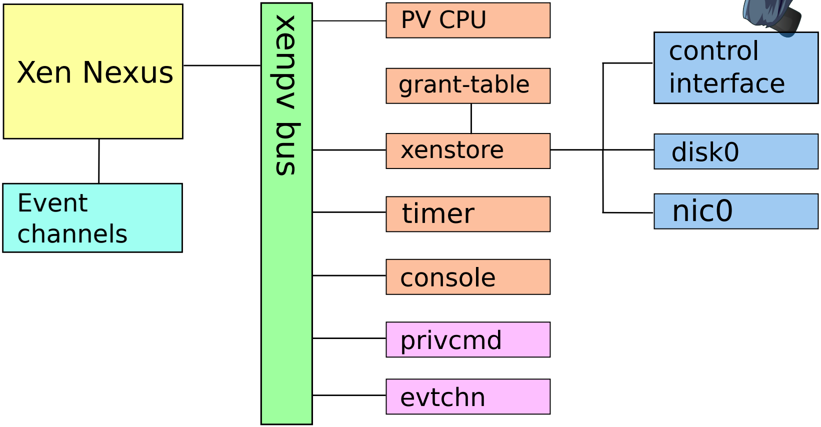
PVH DomU

- PV entry point into the kernel.
- Wire the PV entry point with the rest of the FreeBSD boot sequence.
- Fetch the e820 memory map from Xen.
- PV console.
- Get rid of the usage of any previously emulated devices (serial console, timers).
- PV vCPU bringup for APs.
- Hardware description comes from xenstore, not ACPI.

PVH Dom0

- Builds on top of DomU PVH support.
- Has access to physical hardware devices.
- Parses ACPI tables and notifies Xen about the underlying hardware.
- Special user-space devices are needed, so the toolstack can interact with Xen.

Architecture overview



Pending work items

- Improve robustness and compatibility of if_xn/xnb (PV nic).
- Add some additional user-space devices to interact with Xen:
    - gntdev: allows user-space applications to map grants.
    - gntalloc: allows user-space applications to share memory using grants.
- Add a FreeBSD Dom0 to the Xen automatic test system (OSSTest).
- Test on different hardware.

Conclusions

- FreeBSD/Xen support is evolving from HVM → PVHVM → PVH.
- Initial FreeBSD PVH Dom0 support committed to HEAD.
- Using Xen allows to provide a fully featured virtualization platform based on FreeBSD.

Q&A

Thanks

Questions?