Xen Project 4.4: Features and Futures

Russell Pavlicek
Xen Project Evangelist
Citrix Systems
About This Release

• Xen Project 4.4.0 was released on March 10, 2014.
• This release is the work of 8 months of development, with 1193 changesets.
• Xen Project 4.4 is our first release made with an attempt at a 6-month development cycle.
  – Between Christmas, and a few important blockers, we missed that by about 6 weeks; but still not too bad overall.
Xen Project 101: Basics
**Type 1: Bare metal Hypervisor**

A pure Hypervisor that runs directly on the hardware and hosts Guest OS's.

*Provides partition isolation + reliability, higher security*
Hypervisor Architectures

**Type 1: Bare metal Hypervisor**
A pure Hypervisor that runs directly on the hardware and hosts Guest OS's.

- Provides partition isolation + reliability, higher security

**Type 2: OS ‘Hosted’**
A Hypervisor that runs within a Host OS and hosts Guest OS’s inside of it, using the host OS services to provide the virtual environment.

- Low cost, no additional drivers
- Ease of use & installation
Xen Project: Type 1 with a Twist

**Type 1: Bare metal Hypervisor**

- Host HW
- Memory
- CPUs
- I/O
- Hypervisor
- Scheduler
- MMU
- Device Drivers/Models
- VM\(_n\)
- VM\(_1\)
- VM\(_0\)
- Guest OS and Apps

**Host HW**
- I/O
- Memory
- CPUs
Xen Project: Type 1 with a Twist

Type 1: Bare metal Hypervisor

Xen Architecture

Hypervisor

Scheduler

MMU

Host HW

I/O

Memory

CPUs

VM

Guest OS and Apps

Device Drivers/Models
Xen Project: Type 1 with a Twist

**Type 1: Bare metal Hypervisor**

- Host HW
- Memory
- CPUs
- I/O
- Hypervisor
- Scheduler
- MMU
- Device Drivers/Models
- Guest OS and Apps

**Xen Architecture**

- Control domain (dom0)
- Device Models
- Drivers
- Linux & BSD
- Guest OS and Apps

- Hypervisor
- Scheduler
- MMU
- Host HW
- Memory
- CPUs
- I/O
Basic Xen Project Concepts

### Control Domain aka Dom0
- Dom0 kernel with drivers
- Xen Management Toolstack

### Guest Domains
- Your apps

### Driver/Stub/Service Domain(s)
- A “driver, device model or control service in a box”
- De-privileged and isolated
- Lifetime: start, stop, kill

**Console**
- Interface to the outside world

**Trusted Computing Base**
Basic Xen Project Concepts: Toolstack+

Console

- Interface to the outside world

Control Domain aka Dom0
- Dom0 kernel with drivers
- Xen Management Toolstack

Guest Domains
- Your apps

Driver/Stub/Service Domain(s)
- "A "driver, device model or control service in a box"
- De-privileged and isolated
- Lifetime: start, stop, kill

Trusted Computing Base
Basic Xen Project Concepts: Disaggregation

Console
- Interface to the outside world

Control Domain aka Dom0
- Dom0 kernel with drivers
- Xen Management Toolstack

Guest Domains
- Your apps

Driver/Stub/Service Domain(s)
- A “driver, device model or control service in a box”
- De-privileged and isolated
- Lifetime: start, stop, kill
Xen Project 4.4 Features
Improved Event Channel Scalability

- Event channels are paravirtualized interrupts
- Previously limited to either 1024 or 4096 channels per domain
  - Domain 0 needs several event channels for each guest VM (for network/disk backends, qemu etc.)
  - Practical limit of total number of VMs to around 300-500 (depending on VM configuration)
• New FIFO-based event channel ABI allows for over 100,000 event channels
  - Improve fairness
  - Allows for multiple priorities
  - The increased limit allows for more VMs, which benefits large systems and cloud operating systems such as MirageOS, ErlangOnXen, OSv, HalVM
  - Also useful for VDI applications
PVH mode combines the best elements of HVM and PV
- PVH takes advantage of many of the hardware virtualization features that exist in contemporary hardware

Potential for significantly increased efficiency and performance

Reduced implementation footprint in Linux, FreeBSD

Enable with "pvh=1" in your config
Xen Project Virtualization Vocabulary

- **PV** – Paravirtualization
  - Hypervisor provides API used by the OS of the Guest VM
  - Guest OS needs to be modified to provide the API
- **HVM** – Hardware-assisted Virtual Machine
  - Uses CPU VM extensions to handle Guest requests
  - No modifications to Guest OS
  - But CPU must provide the VM extensions
- **FV** – Full Virtualization (Another name for HVM)
Xen Project Virtualization Vocabulary

• **PVHVM – PV on HVM drivers**
  - Allows H/W virtualized guests to use PV disk and I/O drivers
  - No modifications to guest OS
  - Better performance than straight HVM

• **PVH – PV in HVM Container (New in 4.4)**
  - Almost fully PV
  - Uses HW extensions to eliminate PV MMU
  - Possibly best mode for CPUs with virtual H/W extensions
# The Virtualization Spectrum

<table>
<thead>
<tr>
<th></th>
<th>Disk and Network</th>
<th>Interrupts, Timers</th>
<th>Emulated Motherboard, Privileged Instructions and Page Tables</th>
<th>HVM mode/domain</th>
<th>PV mode/domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>VS</td>
<td>VS</td>
<td>VS</td>
<td>VS</td>
<td>VH</td>
<td></td>
</tr>
<tr>
<td>VH</td>
<td>P</td>
<td>VS</td>
<td>VS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>P</td>
<td>P</td>
<td>VS</td>
<td>VH</td>
<td></td>
</tr>
<tr>
<td><strong>NEW</strong> 4.4</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td></td>
</tr>
</tbody>
</table>

- **VS**: Virtualized (SW)
- **VH**: Virtualized (HW)
- **P**: Paravirtualized

<table>
<thead>
<tr>
<th></th>
<th>Disk and Network</th>
<th>Interrupts, Timers</th>
<th>Emulated Motherboard, Privileged Instructions and Page Tables</th>
<th>HVM mode/domain</th>
<th>PV mode/domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully Virtualized (FV)</td>
<td>VS</td>
<td>VS</td>
<td>VS</td>
<td>VH</td>
<td></td>
</tr>
<tr>
<td>FV with PV for disk &amp; network</td>
<td>P</td>
<td>VS</td>
<td>VS</td>
<td>VH</td>
<td></td>
</tr>
<tr>
<td>PVHVM</td>
<td>P</td>
<td>P</td>
<td>VS</td>
<td>VH</td>
<td></td>
</tr>
<tr>
<td>PVH</td>
<td>NEW</td>
<td>4.4</td>
<td>P</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>Fully Paravirtualized (PV)</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td></td>
</tr>
</tbody>
</table>
The Virtualization Spectrum

<table>
<thead>
<tr>
<th>Scope for improvement</th>
<th>Optimal performance</th>
<th>Poor performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk and Network</td>
<td>Interrupts, Timers</td>
<td>Emulated Motherboard, privileged instructions and page tables</td>
</tr>
</tbody>
</table>

- **Fully Virtualized (FV)**: VS, VS, VS, VH
- **FV with PV for disk & network**: P, VS, VS, VH
- **PVHVM**: P, P, VS, VH
- **PVH 4.4**: P, P, P, VH
- **Fully Paravirtualized (PV)**: P, P, P, P

HVM mode/domain

PV mode/domain
Linux driver domains used to rely on udev events in order to launch backends for guests

- Dependency on udev is replaced with a custom daemon built on top of libxl
- Now feature complete and consistent between Linux and non-Linux guests
- Provides greater flexibility in order to run user-space backends inside of driver domains
- Example of capability: driver domains can now use Qdisk backends, which was not possible with udev
SPICE is a protocol for virtual desktops which allows a much richer connection than display-only protocols like VNC.

Added support for additional SPICE functionality, including:
- Vdagent
- clipboard sharing
- USB redirection
In the past, Xen Project software required a custom implementation of GRUB called pvgrub.

The upstream GRUB 2 project now has a build target which will construct a bootable PV Xen Project image:
- This ensures 100% GRUB 2 compatibility for pvgrub going forward.
- Delivered in upcoming GRUB 2 release (v2.02?).
• Modern storage devices work much better with larger chunks of data
• Indirect descriptors have allowed the size of each individual request to triple, greatly improving I/O performance when running on fast storage technologies like SSD and RAID
• This support is available in any guest running Linux 3.11 or higher (regardless of Xen Project version)
kexec allows a running Xen Project host to be replaced with another OS without rebooting
- Primarily used to execute a crash environment to collect information on a Xen Project hypervisor or dom0 crash

The existing functionality has been extended to:
- Allow tools to load images without requiring dom0 kernel support (which does not exist in upstream kernels)
- Improve reliability when used from a 32-bit dom0
- kexec-tools 2.0.5 or later is required
• XAPI and Mirage OS are sub-projects within the Xen Project written in OCaml
• Both are also used in XenServer (http://XenServer.org) and rely on the Xen Project OCaml language bindings to operate well
• These language bindings have had a major overhaul
  – Produces much better compatibility between XAPI, Mirage OS and Linux distributions going forward
Nested virtualization provides virtualized hardware virtualization extensions to HVM guests

- Can now run Xen Project, KVM, VMWare or HyperV inside of a guest for debugging or deployment testing (only 64 bit hypervisors currently)
- Also allows Windows 7 "XP Compatibility mode"
- Tech Preview not yet ready for production use, but has made significant gains in functionality and reliability
  - Enable with "hap=1" and "nestedhvm=1"

More information on nested virtualization: http://wiki.xenproject.org/wiki/Xen_nested
Experimental Support for Guest EFI boot

• EFI is the new booting standard that is replacing BIOS
  - Some operating systems only boot with EFI
  - Some features, like SecureBoot, only work with EFI
You can find a blog post to set up an iSCSI target on the Gluster blog:

A number of new features have been implemented:
- 64 bit Xen on ARM now supports booting guests
- Physical disk partitions and LVM volumes can now be used to store guest images using xen-blkback (or is PV drivers better in terms of terminology)
- Significant stability improvements across the board
- ARM/multiboot booting protocol design and implementation
- PSCI support
• Some DMA in Dom0 even with no hardware IOMMUs
• ARM and ARM64 ABIs are declared stable and maintained for backwards compatibility
• Significant usability improvements, such as automatic creation of guest device trees and improved handling of host DTBs
• Adding new hardware platforms to Xen Project on ARM has been vastly improved, making it easier for Hardware vendors and embedded vendors to port to their board

• Added support for the Arndale board, Calxeda ECX-2000 (aka Midway), Applied Micro X-Gene Storm, TI OMAP5 and Allwinner A20/A31 boards

• ARM server class hardware (Calxeda Midway) has been introduced in the Xen Project OSSTest automated testing framework
• The hypervisor can update the microcode in the early phase of boot time
  - The microcode binary blob can be either as a standalone multiboot payload, or part of the initial kernel (dom0) initial ramdisk (initrd)
  - To take advantage of this use latest version of dracut with --early-microcode parameter and on the Xen Project command line specify: ucode=scan.
  - For details see dracut manpage and http://xenbits.xenproject.org/docs/unstable/misc/xen-command-line.html
Xen Project Futures
More Fun to Come...

- Xen Automotive
  - Xen Project in the entertainment center of your car?
- XenGT
  - Virtualized GPU support
- Even More ARM Support
  - On your server, in your phone, wherever...
- PVH stability and performance
  - The new hypervisor mode to get harder and faster
  - Domain 0 support, AMD support
And Still More Fun to Come…

- Native support of VMware VMDK format
- Better distribution integration (CentOS, Ubuntu)
- Improvements in NUMA performance and support
- Additional libvirt support
- Automated Testing System
- General performance enhancements
Questions?

Russell.Pavlicek@XenProject.org
Twitter: @RCPavlicek