FreeBSD/Xen update

Roger Pau Monné roger.pau@citrix.com

Tokyo – 11th of March, 2016
Xen Architecture (type-1 hypervisor)

Control Domain

Linux, FreeBSD or NetBSD

Guest 1

Guest 2

Hardware

CPU

MMU

Xen

syslog

xterm

xorg
Type-2 hypervisor architecture

- **syslog**
- **xterm**
- **xorg**
- **...**
- **Guest 1**
- **Guest 2**

**Operating System**

**Hardware**
- **CPU**
- **MMU**
- **...**
Xen architecture in detail

Control Domain (VM₀)

- kernel
  - blkfront
  - netfront
  - blkback
  - netback

- user-space
  - QEMU
  - xenstored
  - xl (toolstack)
  - xenconsoled

Hardware

- CPU
- MMU
- ...

Part of FreeBSD

Part of the Xen package

VM₁

VM₂

VMₙ

Tokyo – 11th of March, 2016
FreeBSD/Xen update
xen-block related improvements

- Remove broken block protocol extensions (r284296).
- Unmapped IO support for blkfront (r290611).
- Indirect descriptors support, by Colin Percival (r286062).
Dom0 improvements

- Multiple fixes for the multiboot support in the loader (r277291, r277418, r280953, r280954).
- Improved PIRQ handling (r278854, r278855).
- Indirect descriptors support, by Colin Percival (r286062).
- Improvements to foreign memory mapping (r282634).
- Added save, restore and live migration support to the Xen package (r398918).
EC2 specific improvements

- Allow creating EC2 AMIs from the FreeBSD build system (r280928) by Colin Percival.
- Support for SR-IOV (A.K.A EC2 Enhanced Networking) for FreeBSD guests.
Generic fixes and improvements

- Xenstore device fixes (r278844).
- Add a handler for the debug interrupt (r280838).
- Update Xen headers to 4.6, previous version was 4.2 (r288917) by Julien Grall.
- Cleanup and unification of Xen files (r289685, r289686) by Julien Grall.
- New PV console driver (r289033) by Julien Grall.
- Add run-time options to disable PV devices (r286999).
- Removal of the i386 UP PV port (r282274) by John Baldwin.
xen-net related improvements

- Fix initial ARP sending on restore from migration (r282908).
- Preserve configured options across migrations (r285098).
- Fix PF to work with netfront (r289316) by Kristof Provost
- Clean-up and new feature
  - Remove obsolete page flipping mode (r289583)
  - Implement multiqueue (r294442)
  - Throughput from guest to host with iperf: 1 queue 5.8 Gb/s, 4 queues 11.2 Gb/s (with WITNESS and INVARINANTS)
## The full virtualisation spectrum

<table>
<thead>
<tr>
<th>VS</th>
<th>Software virtualisation</th>
<th>Poor performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>VH</td>
<td>Hardware virtualisation</td>
<td>Room for improvement</td>
</tr>
<tr>
<td>PV</td>
<td>Paravirtualized</td>
<td>Optimal performance</td>
</tr>
</tbody>
</table>

**Disk and network**
- 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

**Interrupts and timers**
- 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

**Emulated motherboard 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
Why PVH?

- Performance: use hardware feature as much as possible
- Security
  - No emulation eliminate a main class of security bugs
  - No PVMMU etc, a lot less complex code for both guest kernel and Xen toolstack
- Maintenance
  - No PVMMU etc, a lot less code
- Easier to port new OSes
Gory details about PVH

- PVH-classic vs HVMLite
- PVH-classic is first attempt for the design, to make PV guest look like HVM guest
- HVMLite is the new approach, to make HVM guest look like PV guest
- They will converge at some point, the agreed upon road map is to make HVMLite canonical ”PVH”
- End users probably won’t notice the difference
## Guest support

- List of OSes and virtualisation support:

<table>
<thead>
<tr>
<th></th>
<th>PV</th>
<th>PVH*</th>
<th>PVHVM</th>
<th>HVM with PV drivers</th>
<th>HVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Windows</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>NetBSD</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>FreeBSD</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>OpenBSD</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>DragonflyBSD</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
</tr>
</tbody>
</table>
Better scalability

- Finer grained locks in hypervisor: per-vcpu maptrack free lists, per-cpu rwlock
- Fairer locks in hypervisor: queue rwlock
- Should benefit all guests, especially Xen virtual devices with multiqueue support (net, block)
  - 2-socket Haswell-EP systems, Linux 16 queues inter-VM network throughput jumped from 15 Gb/s to 48 Gb/s
Virtual Performance Monitoring Unit

- Fully implemented in Xen 4.6, works for both PV and HVM
- Intended for non-production use
- Use dtrace(1) or pmcstat(8) to profile your VM
xSplice - hypervisor hot-patching

- **Rationale:**
  - Rebooting hypervisor to fix security bugs are not desirable
  - A large number of security bugs require very simple patch to fix
- Phase one goal is to handling patching functions, patching structure is yet to come
- User space tooling is not tied to particular operating system
Thanks

Questions?