Stub Domains

A Step Towards Dom0 Disaggregation

Samuel Thibault, Citrix/XenSource
The Big Domain 0

- Runs a lot of Xen components
  - Domain manager
  - Domain Builder
  - Device Models
  - PyGRUB

- These are currently running as root
  - e.g. PyGRUB to access guest's disk

- Security issues
- Scalability issues
What Are Stub Domains?

- Helper domains which run Xen components
- Based on Mini-OS
- Domain Builder (Derek Murray)
- Device Model
- PV-GRUB
- ...
What Are Stub Domains?

- Helper domains which run Xen components
- Based on Mini-OS
- Domain Builder (Derek Murray)
- Device Model
- PV-GRUB
- ...
POSIX Environment on Top of Mini–OS

Application

new lib

getpid, sleep, read, select, ...

lw IP

Sched  MM  Console frontend  Network frontend  Block frontend  FS frontend  FB frontend

Mini–OS

Xen Hypervisor
New Mini-OS Features

- Disk frontend
- FrameBuffer frontend
- FileSystem frontend
  - Imported from JavaGuest
  - Remote access to some /export (e.g. of dom0)
- More advanced MM
  - Read-Only memory
  - CoW for zeroed pages
- But still keep it simple
  - Single address space, mono-VCPU, no preemption
- Bugfixes!
stubdom/

- **Makefile**
  - Download and compile a cross-compilation environment
    - binutils, gcc, newlib, lwip

- **c/**
  - 'Hello World!' C application

- **caml/**
  - 'Hello World!' Caml application

- **README**
  - Of course :)

- Of course :)
Current HVM device model

- qemu
- Linux dom0
- HVM domain
- Xen Hypervisor

IN/OUT

Xen Summit™
Current HVM dm

- Not always responsive
  - Have to wait for dom0 Linux to schedule qemu
- Eats dom0 CPU time
- Uses dom0 resources from userland
  - Disk, tap network
  - Hence runs as root
HVM dm domain

Linux

dom 0

Mini-OS

stub dom

qemu

HVM domain

Xen Hypervisor

Xen Summit™
HVM dm domain

![Bar chart showing boot time comparison between Dom0 and Stubdom.]
HVM dm domain Disk Perfs

<table>
<thead>
<tr>
<th></th>
<th>Read (MB/s)</th>
<th>Write (MB/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dom0</td>
<td>70</td>
<td>60</td>
</tr>
<tr>
<td>Stubdom</td>
<td>70</td>
<td>60</td>
</tr>
<tr>
<td>Native</td>
<td>70</td>
<td>60</td>
</tr>
</tbody>
</table>
HVM dm domain Net Perfs e1000

- **Recv (MB/s)**
  - Dom0: 45
  - Stubdom: 75

- **Send (MB/s)**
  - Dom0: 40
  - Stubdom: 70
HVM dm domain Net CPU%
e1000

Recv
stubdom

stubdom
Send

dom0

stubdom

stubdom

DomU
Dom0
Stubdom
Free

DomU
Dom0
Stubdom
Free
HVM dm domain Net Perfs
bicore

The graph shows the network performance in MB/s for two domains: Dom0 and Stubdom. The y-axis represents the performance in MB/s, ranging from 0 to 120. The graph compares the receive (Recv) and send (Send) speeds for both Dom0 and Stubdom.

- **Recv (MB/s)**:
  - Dom0: Approximately 65 MB/s
  - Stubdom: Approximately 105 MB/s

- **Send (MB/s)**:
  - Dom0: Approximately 55 MB/s
  - Stubdom: Approximately 90 MB/s
HVM dm domain Net CPU% bicore

Recv

stubdom

dom0

stubdom

Send

dom0

DomU
Dom0
Stubdom
Free

DomU
Dom0
Stubdom
Free
HVM dm domain

- Almost unmodified qemu
  - Disable e.g. sound support, plug Mini-OS PV drivers
- Relieves dom0
- Provides better CPU usage accounting
  - Can charge HVM domain with dm domain time
- A lot safer
  - Only privilege is having the HVM dom as target
  - Uses same resource access as PV guests
- More efficient
  - Let the hypervisor schedule it directly
  - More lightweight OS
PyGRUB

Xen Hypervisor

PyGRUB

xend

Linux

dom 0

PV domain

menu.lst
vmlinuz
initrd
PyGRUB

- Needs to be root to access guest disk
  - Security issues
- Does not currently provide network boot
- Reimplements GRUB
PV–GRUB start

xend

Linux
dom 0

GRUB
libxc

Mini-OS

Xen Hypervisor

menu.lst
vmlinux
initrd
PV–GRUB loading

Xen Hypervisor

xend
Linux
dom 0

PV kernel
initrd
GRUB
libxc
Mini OS
blkfront
netfront

menu.lst
vmlinux
initrd
PV–GRUB loaded

xend

Linux
dom 0

Xen Hypervisor

PV kernel
initrd

GRUB
libxc

MINI-OS

Kexec!
PV–GRUB

Xen Hypervisor

xend

Linux
dom 0

PV kernel

initrd

PV domain

Xen Hypervisor

Xen Summit™
PV–kexec

initrd

PV kernel

boot

ekexec

libxc

GRUB

Mini-OS

Mini-OS

virtual memory
PV–kexec

PV kernel
initrd
boot
kexec
libxc
GRUB
Mini-OS

stack
pgtable
initrd
PV kernel

0xc0000000

Mini-OS virtual memory
Target PV guest virtual memory
PV-kexec

Min-i-OS

Virtual memory
PV-kexec

PV kernel
initrd
boot
PV kernel
boot
ekexec
libxc
GRUB
Mini-OS

stack
boot
pgtable
initrd
PV kernel

0xc0000000

Mini-OS
virtual memory

Target PV guest
virtual memory

Xen Summit™
PV-kexec

### Structure

- **PV kernel**
- **initrd**
- **boot**
- **libxc**
- **GRUB**
- **Mini-OS**

### Virtual Memory

- Stack
- Boot page table
- initrd
- PV kernel

### Target PV guest

- Virtual Memory

---

**Xen Summit™**
PV-kexec

initrd
boot
PV kernel

boot
ekexec
libxc
GRUB
Mini-OS

stack
boot
pgtable
initrd
PV kernel

0xc0000000

Mini-OS virtual memory

Target PV guest virtual memory
PV–GRUB

- Executes upstream GRUB
  - Replace native drivers with Mini–OS drivers
  - Add PV kexec implementation
- Just uses the target PV guest resources
- Supports network
- Supports graphical menu
Conclusion

- Dm domain
  - Improves security
  - Improves accounting
  - Improves scalability
  - Improves performances

- PV-GRUB
  - Improves security
  - Provides network boot

- Mini-OS also being tested at Cisco for IOS

- Available in the unstable tree
Future Work

- **Dm domain**
  - *Live* migration, PCI PT
  - IA-64 support
  - Group scheduling with HVM domain

- **PV-GRUB**
  - Kexec 64bit guest from 32bit PV-GRUB
  - PVFB shutdown/restart

- **OCaml support**
  - 'Hello World!' works
  - Needs runtime rebuild to properly hook into POSIX layer