

# Xen in Mainline Linux Status Update

Jeremy Fitzhardinge  
Citrix



# Upstreaming Xen

- Xen domU support
  - Upstream and stable for well over a year
  - Shipped as standard in several distros
    - Fedora
    - Debian
    - ...?
  - (SuSE is the odd one out.)
- Dom0 support is the next frontier



# git migration

- Migrated upstream Xen work to git
- Mercurial + patchqueue was getting awkward
- Git makes is more conventional:
  - Easier for upstream developers to work with
  - Easier for Xen developers to get a working tree
- Hosted on git.kernel.org: “jeremy/xen.git”



# Git Branches

- Two main merge branches
  - xen/master – core Xen, domU
  - xen/dom0/hackery – dom0 work branch
- Lots of topic branches
  - Merged into merge branches for use
  - Will add more as needed
- Room for lots more
  - Will take anything
  - Even merged if you don't break things!



# General improvements

- Preemptable lazy mmu updates
  - Allow more use of lazy updates
  - Less scheduling effect
- /proc/xen + usermode xenbus for tools support
- New calling convention for reduced register pressure
- Still need a concerted performance-oriented push



# Paravirt performance

- Early design goal of paravirt\_ops was zero-overhead native execution
- Some benchmarking showed we were falling short
  - ~5% impact on mmap heavy benchmark
- The problem: inlines -> function calls cause lots more register pressure
- Resulting in higher cache traffic due to spills



# New calling convention

- Solution: define new calling convention
- Normal convention trashes many registers
- Define new one which preserves everything
  - Except return value
- Result:
  - compiler can generate better code
  - Inline patching more effective
- But how to call normal code?



# Generate reg-saver thunks

- Add macro to generate register saving wrapper around conventional code
- Pushes cost of call to complex code
- Simple callees – written in asm – need no wrapper
- Overall: reduce overhead to 1-2%



# Dom0

- Boot-time control domain
- Also basis for driver domains
- At heart, a normal PV Xen domain
- But with extra bits
  - APIC
  - ACPI
  - Device mappings
  - DMA / SWIOTLB



# Dom 0: APICs

- APIC = interrupt controller
- Two parts:
  - IO APICs connected to PCI slots
  - Local ACPI in each CPU
- Xen owns Local APIC, since dom0 has virtualized CPUs
- Dom0 owns IO APICs because it has the device drivers



# Dom 0: APICs (2)

- Changes surprisingly small
  - Some changes to APIC discovery to avoid local apics
  - Hook `acpi_register_gsi` to direct all setup to Xen code
  - Only support ACPI interrupt routing
  - IRQ space reserved for 1:1 GSI mapping
- Caveat: APIC and ACPI use different senses for interrupt triggering
  - Getting it wrong works surprisingly well



# Dom 0: ACPI

- ACPI = Amazingly Complex Piece of Interface
- Used for everything from device discovery, interrupt routing to power management
- To start with, mostly interested in devices and interrupts
- Main changes to APIC code were to make it map properly
- Seems well-behaved after that



# Dom 0: device mappings

- Dom0 kernel has two address spaces:
  - Pseudo-physical domain memory
  - Machine-physical memory
- Must know which is which for any given mapping
- Add `_PAGE_IOMAP` flag to mark hardware ptes
  - No pfn->mfn conversion
- VM\_IO segments mapped with `_PAGE_IOMAP`
- E820 map avoids RAM holes



# Dom 0: DMA / SWIOTLB

- Make sure devices are talking to the memory they think they are
- Hook DMA operations to
  - Convert pfn $\leftrightarrow$ mfn
  - Make memory machine contiguous
- SWIOTLB deals with a lot of the tricky cases
  - Involves copying, so should be avoided for high performance devices



**\*\*\* SUBJECT HERE \*\*\***

**\*\*\* BLURB HERE \*\*\***



# Dom 0: upstream progress

- Most core patches posted, out for review
- Looks good for next merge window (2.6.30)
  - Core support at least
  - Not necessarily every feature

- 



**DEMO**



# State: Good

- Generally very stable
- AHCI, PIIX, mpt drive controllers work
- E1000, iwlagn networking OK
- Intel graphics fully accelerated
  - Radeon X server starts OK too
- Sound fine
- USB good too
- Oh, and you can start domains



# State: Meh

- S3 suspend not done
- Cpufreq doesn't work
- ACPI hotkeys seem dead
- Need wider hardware testing for more confidence
- HVM was working, but now not (general xen-unstable issue?)



# TODO

- Host S3 suspend resume
  - Should just be a matter of bringing over patches
  - Upstreaming could be awkward
- MSI
  - Hoping its no more complex than APIC
- Pciback
- Blktap2
- Pvhvm support – started, need more work
- Wider hardware testing



# No Excuses

- Now is the time to base development on the mainline kernel
- Core Xen support is stable
- Dom0 support is fairly stable
- Remaining work is in self-contained chunks



# The Kittens are Thinking of You...



# Thanks!

## **GIT repository:**

`git://git.kernel.org/pub/scm/linux/kernel/git/jeremy/xen.git`

or...

1. Go to `git.kernel.org`
2. Look for “xen.git” on the page
3. Cut long url
4. `git clone <paste> linux-xen`



# Xen in Mainline Linux Status Update

Jeremy Fitzhardinge  
Citrix

Xen Summit at Oracle Feb 24-25, 2009



# Upstreaming Xen

- Xen domU support
  - Upstream and stable for well over a year
  - Shipped as standard in several distros
    - Fedora
    - Debian
    - ...?
  - (SuSE is the odd one out.)
- Dom0 support is the next frontier

## git migration

- Migrated upstream Xen work to git
- Mercurial + patchqueue was getting awkward
- Git makes is more conventional:
  - Easier for upstream developers to work with
  - Easier for Xen developers to get a working tree
- Hosted on git.kernel.org: “jeremy/xen.git”

# Git Branches

- Two main merge branches
  - xen/master – core Xen, domU
  - xen/dom0/hackery – dom0 work branch
- Lots of topic branches
  - Merged into merge branches for use
  - Will add more as needed
- Room for lots more
  - Will take anything
  - Even merged if you don't break things!



## General improvements

- Preemptable lazy mmu updates
  - Allow more use of lazy updates
  - Less scheduling effect
- /proc/xen + usermode xenbus for tools support
- New calling convention for reduced register pressure
- Still need a concerted performance-oriented push

## Paravirt performance

- Early design goal of paravirt\_ops was zero-overhead native execution
- Some benchmarking showed we were falling short
  - ~5% impact on mmap heavy benchmark
- The problem: inlines -> function calls cause lots more register pressure
- Resulting in higher cache traffic due to spills

## New calling convention

- Solution: define new calling convention
- Normal convention trashes many registers
- Define new one which preserves everything
  - Except return value
- Result:
  - compiler can generate better code
  - Inline patching more effective
- But how to call normal code?



## Generate reg-saver thunks

- Add macro to generate register saving wrapper around conventional code
- Pushes cost of call to complex code
- Simple callees – written in asm – need no wrapper
- Overall: reduce overhead to 1-2%

## Dom0

- Boot-time control domain
- Also basis for driver domains
- At heart, a normal PV Xen domain
- But with extra bits
  - APIC
  - ACPI
  - Device mappings
  - DMA / SWIOTLB

## Dom 0: APICs

- APIC = interrupt controller
- Two parts:
  - IO APICs connected to PCI slots
  - Local ACPI in each CPU
- Xen owns Local APIC, since dom0 has virtualized CPUs
- Dom0 owns IO APICs because it has the device drivers

## Dom 0: APICs (2)

- Changes surprisingly small
  - Some changes to APIC discovery to avoid local apics
  - Hook `acpi_register_gsi` to direct all setup to Xen code
  - Only support ACPI interrupt routing
  - IRQ space reserved for 1:1 GSI mapping
- Caveat: APIC and ACPI use different senses for interrupt triggering
  - Getting it wrong works surprisingly well

## Dom 0: ACPI

- ACPI = Amazingly Complex Piece of Interface
- Used for everything from device discovery, interrupt routing to power management
- To start with, mostly interested in devices and interrupts
- Main changes to APIC code were to make it map properly
- Seems well-behaved after that

## Dom 0: device mappings

- Dom0 kernel has two address spaces:
  - Pseudo-physical domain memory
  - Machine-physical memory
- Must know which is which for any given mapping
- Add `_PAGE_IOMAP` flag to mark hardware ptes
  - No pfn->mfnc conversion
- VM\_IO segments mapped with `_PAGE_IOMAP`
- E820 map avoids RAM holes

## Dom 0: DMA / SWIOTLB

- Make sure devices are talking to the memory they think they are
- Hook DMA operations to
  - Convert pfn<->mfn
  - Make memory machine contiguous
- SWIOTLB deals with a lot of the tricky cases
  - Involves copying, so should be avoided for high performance devices



**\*\*\* SUBJECT HERE \*\*\***

**\*\*\* BLURB HERE \*\*\***



Xen Summit at Oracle Feb 24-25, 2009

## Dom 0: upstream progress

- Most core patches posted, out for review
- Looks good for next merge window (2.6.30)
  - Core support at least
  - Not necessarily every feature

- 



**DEMO**

Xen Summit at Oracle Feb 24-25, 2009



## State: Good

- Generally very stable
- AHCI, PIIX, mpt drive controllers work
- E1000, iwlagn networking OK
- Intel graphics fully accelerated
  - Radeon X server starts OK too
- Sound fine
- USB good too
- Oh, and you can start domains

## State: Meh

- S3 suspend not done
- Cpufreq doesn't work
- ACPI hotkeys seem dead
- Need wider hardware testing for more confidence
- HVM was working, but now not (general xen-unstable issue?)

# TODO

- Host S3 suspend resume
  - Should just be a matter of bringing over patches
  - Upstreaming could be awkward
- MSI
  - Hoping its no more complex than APIC
- Pciback
- Blktap2
- Pvhvm support – started, need more work
- Wider hardware testing



# No Excuses

- Now is the time to base development on the mainline kernel
- Core Xen support is stable
- Dom0 support is fairly stable
- Remaining work is in self-contained chunks



# The Kittens are Thinking of You...



Xen Summit at Oracle Feb 24-25, 2009



# Thanks!

**GIT repository:**

`git://git.kernel.org/pub/scm/linux/kernel/git/jeremy/xen.git`

or...

1. Go to `git.kernel.org`
2. Look for “xen.git” on the page
3. Cut long url
4. `git clone <paste> linux-xen`

